Save 50% with the IET Summer Sale

Use code **SUMMER23** to save on over 650+ selected engineering and technology books.

*Discount only available on selected print books between 21 August to 15 September

The Institution of Engineering and Technology

Browse the sale titles

IET Renewable Power Generation

Research Article

Proportional–integral–derivative parameter optimisation of blade pitch controller in wind turbines by a new intelligent genetic algorithm

ISSN 1752-1416
Received on 14th January 2016
Revised 18th April 2016
Accepted on 19th April 2016
E-First on 28th July 2016
doi: 10.1049/iet-rpg.2016.0029
www.ietdl.org

Zafer Civelek[1] ✉, Ertuğrul Çam[2], Murat Lüy[2], Hayati Mamur[3]
[1]*Technical Business College, Çankırı Karatekin University, Çankırı, Turkey*
[2]*Faculty of Engineering, Kırıkkale University, Kırıkkale, Turkey*
[3]*Faculty of Engineering, Çankırı Karatekin University, Çankırı, Turkey*
✉ *E-mail: zafercivelek@karatekin.edu.tr*

Abstract: Output powers of wind turbines (WTs) with variable blade pitch over nominal wind speeds are controlled by means of blade pitch adjustment. While tuning the blade pitch, conventional proportional–integral–derivative (PID) controllers and some intelligent genetic algorithms (IGAs) are widely used in hot systems. Since IGAs are community-based optimisation methods, they have an ability to look for multi-point solutions. However, the PID parameter setting optimisation of the IGA controllers is important and quite difficult a step in WTs. To solve this problem, while the optimisation is carried out by regulating mutation rates in some IGA controllers, the optimisation is conducted by altering crossover point numbers in others. In this study, a new IGA algorithm approach has been suggested for the PID parameter setting optimisation of the blade pitch controller. The algorithm rearranging both the mutation rate and the crossover point number together according to the algorithm progress has been firstly used. The new IGA approach has also been tested and validated by using MATLAB/Simulink software. Then, its superiority has been proved by comparing the other genetic algorithm (GAs). Consequently, the new IGA approach has more successfully adjusted the blade pitch of a WT running at higher wind speeds than other GA methods.

1 Introduction

Wind energy (WE) has been used for various purposes in the past. However, it was not very popular for electricity generation. In recent years, it has become more attractive because of both the increase in technological innovations and the reduction in costs of WE systems. Nowadays, it has been used for electricity generation from small household consumers to big factories. Since large WE systems are more efficient than small and medium ones, investors have started to prefer the establishment of the large and powerful WE systems due to their cost, performance and power requirements advantages.

Output power of wind turbine generators (WTGs) is proportional to the cube of the wind speed. Therefore, wind speed data and their analysis are very important for such systems. In WE systems, two generator models have been utilised as a WTG; variable speed WTG and the fixed speed WTGs. Particularly in WE systems with large scale, blade pitches have to be controlled in order to prevent damage to the system over nominal wind speeds [1]. For this reason, the variable speed WTGs provides an opportunity to generate more power than the fixed speed WTGs.

On the other hand, the output power of the variable speed WTGs varies due to fluctuations in wind speeds. Ultimately, this causes changes of the voltage and the frequency. However, a fixed voltage and a fixed frequency are two of the most important criteria for a quality of electrical energy. In order to improve quality of the output power, appropriate control methods have to be applied on WE systems [2]. The most well-known methods are passive stall control and blade pitch control mechanisms for a WE [3]. Most effective control method among them is the blade pitch control method [2].

In some studies concerning the WT blade pitch control through intelligent genetic algorithms (IGAs) methods, first of all, Perales *et al.* recommended a fuzzy logic control (FLC) method based-on estimated wind speed so as to bring a steady state to the wind turbine (WT) system and obtain maximum power from the system. In the blade pitch control, Kong *et al.* [4] proposed a method that combines a non-linear sliding mode control with a fuzzy set theory to provide a stable output power in WTs with high power such as several MWs. In order to derive energy from a WT system, two FLC methods, one of which controls the blade pitch and the other manages the WTG speed using a field oriented FLC, were carried out by Amendola and Gonzaga [5]. According to the simulation results of their method, even though a WT conversion system was under a strong and turbulent wind speed condition, a stable and a smooth dynamic behaviour were achieved. Muhando *et al.* [6] realised a learning adaptive controller using a self-tuning regulator (STR). In their suggestion, the STR combines a linear quadratic Gaussian with linear parameter estimator. The maximisation of the energy harvesting and a damage minimisation due to mechanical fatigue were fulfilled successfully. Moreover, in a study executed by Yao *et al.* [7], an adaptive fuzzy sliding mode control method was applied to a pitch executor activated by a pitch control unit. In this study, problems including non-linear, parameter time variations, anti-disturbance and time-lag were reduced when inspected the simulation results. A blade pitch system was also operated in a manner that would give a good static and dynamic response. In addition, Zhang *et al.* [8] performed a blade pitch control based on fuzzy–proportional–derivative for large WTs operating above-rated power. By means of the control strategy, better power regulation and less blade pitch movement were acquired. In addition to these, a new control method that was based on FLC was asserted in order to reach a stability position to output powers of WTs working in low and high wind speeds by Jian-Jun *et al.* [9]. According to their simulation results, they obtained a quicker response in output power time about 4 s. In another study, Dou *et al.* [10] suggested an adjustable pitch control system based on a fuzzy self-tuning proportional–integral–derivative (PID) control for large power WTs. In their study, an optimum drive torque was achieved by the blade pitch angle control with specific blade parameters. Jiao and Wang [11] improved an artificial neural network with recurrent back propagation forward that gave good results in terms of the blade pitch control. For large power WTs, a new self-tuning PID algorithm that used an actor-critic learning strategy was introduced by Kim *et al.* [12]. Katsigiannis *et al.* [13] and Sundareswaran *et al.* [14] offered a non-dominating sorting genetic algorithm (GA) to solve multiobjective optimisation problems. A GA was used to unravel optimisation problems such as main parts, important technique properties and electrical system design of wind fields by Zhao *et al.* [15]. In addition to, a standard GA was employed for aerodynamic optimisation in a WT blade design by Yassin *et al.* [16]. They achieved a remarkable increase of 5.9% in annual energy production. Likewise, Jiang *et al.* [17] used a GA that optimises the parameters of a hybrid estimation method written for wind fields. In case of a power grid wind disaster emergencies, a GA enhanced fitness function was applied for power grid adaptation by Zhang *et al.* [18] as well. Furthermore, Chen *et al.* [19] practised a multiobjective GA for wind field design. The fitness function of the algorithm was formed to both maximise efficiency and minimise energy cost of wind fields. In addition these, some GAs were accomplished for wind field optimisations in [20, 21]. Eventually, when these studies were examined in detail, each of them solved a problem.

When attention is paid to aforementioned studies, PID controllers have been used in a great deal of control applications because of their simple structures, high safety and robustness [22, 23]. Therefore, PID controllers have been employed in order to control system outputs of particular systems, which have precise mathematical models [24, 25]. However, setting of PID parameters is quite difficult. Especially today, as complexities of industrial systems having non-linear behaviours, high order and long response time increase, some problems and performance decrease in the controlled of the type industrial systems via the conventional PID controllers emerge. Thus, so as to adjust the PID parameters, a great deal of modern intelligent optimisation methods has been utilised. The PID control parameters can be determined by Ziegler–Nichols method [26]. However, it is difficult to achieve a good system performance with this method. In this method, the best system performance is usually obtained with the experience of a designer. For this reason, the systems with artificial intelligence, neural networks, fuzzy logic (FL), neural-FL and so on control algorithms have been used to adjust the PID parameters as well [27, 28]. However, the analyses of these control algorithms are more complicated. Recently, GAs and particle swarm optimisation (PSO) algorithm methods have been handled in order to adjust the parameters [29, 30].

In this study, the blade pitch control of a large WT system with the conventional PID controller that is employed in various components of WE systems is firstly attempted. However, as before mentioned, adjustment and optimisation of the PID parameter have been extremely difficult due to system complexity. So as to facilitate the PID parameter adjusting and to overcome these challenges, a new IGA approach changing both a crossover and a mutation method together according to algorithm progress is recommended and validated by MATLAB/Simulink software. Thus, in the blade pitch and the output power control, significant improvements have been obtained by means of the new IGA approach.

2 Method

2.1 Conventional PID controller

Conventional PID controllers have a control structure with feedback control. After a system error is passed from proportional, integral and derivative operations, the error is applied again to the system input in accordance with and the system output is controlled like as desired. Description of the continuous form of the PID controller is as follows:

$$u(t) = K_{\rm p}e(t) + K_{\rm i}\int_0^t e(t)\,{\rm d}t + K_{\rm d}\frac{{\rm d}e(t)}{{\rm d}t} \tag{1}$$

where $u(t)$ is the controlled output, $K_{\rm p}$ is the proportional gain, $K_{\rm i}$ is the integral gain, $K_{\rm d}$ is the derivative gain and $e(t)$ is also the error signal between the system output and the system reference value. A discrete form of PID controller is as below:

$$\Delta u(k) = K_{\rm p}[e(k) - e(k-1)] + K_{\rm i}T_{\rm s}e(k) + \frac{K_{\rm d}}{T_{\rm s}}[e(k) - 2e(k-1) + e(k-2)] \tag{2}$$

where $T_{\rm s}$, k and $e(k)$ are the time constant, the number of iterations and the error between the output and the reference value in k iteration.

2.2 Standard GA

As is known, a binary code structure is generally used in GA systems. An initial population is also formed randomly. Flow diagrams of standard both GA and IGA are shown in Figs. 1*a* and *b*.

2.2.1 Selection: The purpose of selection is to define the members of a new generation from a current population in accordance with the chosen fitness function and the selection method. Parents are selected according to their fitness values for mating. Transfer possibility of the individual with higher fitness to the new generation is higher. There are a whole of selection operators such as roulette wheel, tournament, steady state and permutation. In the standard GA, the roulette wheel has been used.

2.2.2 Crossover: The crossover whose mission is to produce new individuals by using the structures of two individuals is one of the most significant processes in the GA. The individuals with the highest capability values from the individuals with higher values are struggled. The crossover rate refers to the rate of the entire population to the individuals to be subjected to the crossover. While the crossover varieties change with respect to an application, one-point crossover, two-point crossover, multi-point crossover and uniform crossover are the most common methods. When a classic GA was designed in the study, a one-point crossover is preferred. In addition, the crossover rate was taken as 90%.

2.2.3 Mutation: The mutation is random changes on chromosome genes at a certain rate. Through this process, during the search of the best solution, the GA tries the best or the worst solution away from the next solution search space. As a result of the mutation process, even though the found chromosome is worse, the GA would gather himself and then could approach the best value because of it being population-based. The mutation preserves the local maximisations and minimisations. For this reason, the mutation rate has to be well chosen. A high mutation rate may affect the best convergence of the algorithm or obscure it while a low mutation rate may cause the inability to get rid of the local maximisations and minimisations of the algorithm. In this study, the optimal mutation rate was defined as 8% for the classic GA through experiments.

2.2.4 Fitness function: It is a function selected to give a measure of closeness of an element to other searching element in a population. There is absolutely needed a coherent function to indicate how the desired solution for the each of the community chromosome that round in different points of the decision space close. The GA fitness function has not been explained in here because the GA fitness function is same to the IGA fitness function.

2.3 Proposed IGA

In the proposed study, the first members of the initial population were chosen as zero to speed up the approach. The algorithm defines an iteration number for convergence to an optimal value of population. The iteration number may show some varieties according to system functions or system applied a GA. For this reason, a personal defines the value according to the applied system or system functions. In this study, the value has been taken as 20. After the iteration number given for convergence is passed, the algorithm agrees that there are the local minima or maxima. In

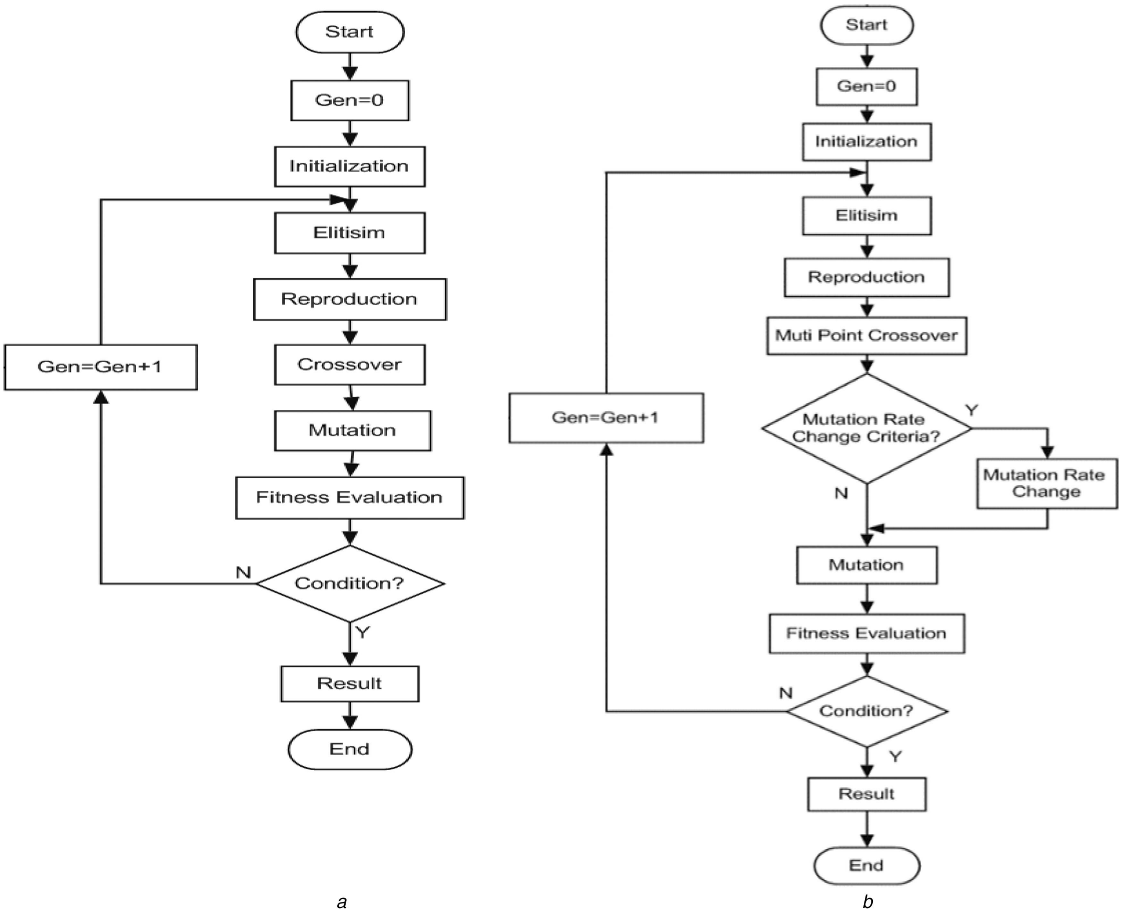


Fig. 1 *Flow diagrams of*
(a) GA, ***(b)*** IGA

order to recover the local minima or maxima, the algorithm implements two operations.

One of them is that the mutation rate is increased. In the case, the mutation rate is increased depending on a predetermined mutation step when the algorithm passes the iteration number. The increase continues until the maximum mutation limit. The mutation value returns the starting mutation value after the maximum mutation limit. The maximum mutation limit value is generally selects a great value. In this study, the value has been selected as 250. If the algorithm gets rid of the local maxima and minima, the mutation rate is returned the starting value.

Other is that the crossover point value is increased one in order to enrich the population when the iteration limit value is passed. The algorithm begins with one-point crossover when it operates. If the fitness function repetitive value being same exceeds the iteration limit, the crossover point is raised two. In case of the each iteration exceeding, the value is coming until the round value by one, chromosome/10. It then returns again to one. If the algorithm does not get rid of the local maxima or minima, the loop continues. In Fig. 7*a*, the compliance value variations as a function of mutation rate and crossover point number during iteration are illustrated.

2.3.1 Selection: In this study, the roulette wheel selection operator was preferred. For this reason, each array was chosen a probability value being proportional with the fitness value defined. The surface of the roulette wheel was marked in proportional to the fitness value of the arrays. Every time the roulette wheel was rotated, an array was thrown into the matching pool. The election changes of the arrays having with better fitness values were higher owing to the more places in the roulette wheel.

2.3.2 Crossover: For the IGA, a variable multi-point crossover was applied, that is, starting from a one-point crossover in the program algorithm, the multi-point crossover was utilised until reaching the chromosome number to the n values expressed as below:

$$n = \text{rounded}(\text{chromosome number}/10) \quad (3)$$

As the chromosome number was increased one, the crossover point number was increased one until reaching the n value in (3) like that the second chromosome of the population was the one-point crossover, the third chromosome of the population was the two-point crossover and such that. Here, the first chromosome of the population was the best and it was not included in the crossover process and it was not changed. When the iteration was reached n value, it was returned the one-point crossover. The purpose of the iteration has been to observe very rich exchange among the chromosomes by means of the multi-point crossover. The approach possibility of the reach exchanges to the optimum value was higher.

2.3.3 Mutation: In the IGA, the mutation rate was altered between the lower and upper limits by a software algorithm. An altering criterion of the mutation rate was the repeating number of the best found solution. The iteration was started from the lower limit rate. When the best found solution was repeated at a certain number, the mutation rate was increased to a certain value by the program algorithm. After the mutation rate reached to the upper limit, at this point, it was expanded to the lower limit as long as there was the altering creation of the mutation rate since the upper mutation rate prevented the optimum convergence of the population. After the upper mutation rate recovered from the local minimisations and maximisations to the algorithm, it returned back to the former state. Otherwise, the population did not continue the global minimisation and maximisation convergence.

2.3.4 Fitness function: Two criteria were taken into consideration when determining the fitness function in the study. One of them

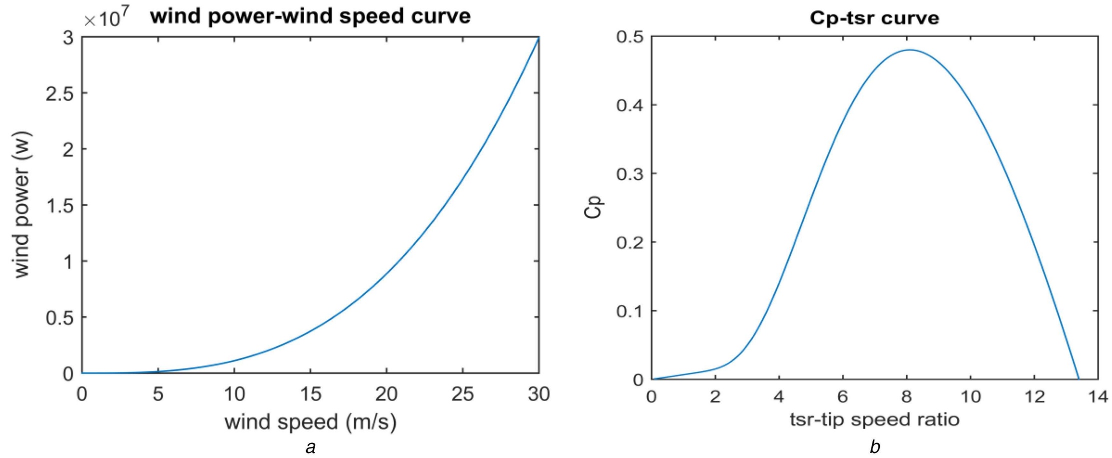


Fig. 2 *Wind power and TSR curves*
(a) Graph of change in wind power as a function of wind speed, ***(b)*** Curve of the power coefficient as a function of the TSR

was the total error of the system being as small as possible. The other was the acceptable maximum overshoot value. According to the defined criteria, the fitness function was consisted of two parts as follows:

$$F(t) = c_1 \cdot F_1(t) + c_2 \cdot F_2(t) \tag{4}$$

where $F(t)$, $F_1(t)$, $F_2(t)$, c_1 and c_2 are the fitness function sum, first fitness function, the second fitness function and constant coefficients, respectively. The first fitness function $F_1(t)$ is introduced so as to reduce the minimum total error as shown below:

$$F_1(t) = 10^{-5(e_m - sp)/sp} \tag{5}$$

where e_m is the average error rate and sp is the adjusting point. The second fitness function $F_2(t)$ was used as follows and then the rules were defined:

$$F_2(t) = 10^{-5(e_{\max} - o_{\max})/o_{\max}} \tag{6}$$

where $o_{\max}$ is the acceptable overshoot value and $e_{\max}$ is also the overshoot value of the working system. The defined rules are:

i. In the regions above the acceptable overshoot value, $F_2(t)$ was zero.
ii. Since the acceptable overshoot value, $F_2(t)$ value, which was increased towards the adjusting point, was taken.

In Fig. 1*b*, a general flow diagram of IGAs is given. When Fig. 1*b* is considered, IGAs apply the multi-point crossover. Moreover, for the mutation process, the IGAs are to change the mutation rate to obtain better solutions from the algorithm through the above mentioned criteria.

2.4 Used power equations of WTs

Wind power is proportional to the cube of wind speed and is expressed as below:

$$P = 0.5\rho A v^3 \tag{7}$$

where P is the wind speed (W), ρ is the air density (kg/m^3), A is the area swept by the blades (m^2) and v is the wind speed (m/s). The graph of change in wind power as a function of wind speed is presented in Fig. 2*a*.

WTs can convert save of the wind power into energy. The conversion rate does not exceed 59% and the rate is limited by Betz criteria [31]. The amount of power derived from WTs is defined by power coefficient C_p that is a function of the blade pitch angle β and the tip speed ratio (TSR) rate λ [32]. In Fig. 2*b*, the curve of the power coefficient as a function of the TSR is given.

The mechanical power of a WT obtaining from wind is given as below:

$$P_{\omega t} = PC_p(\beta, \lambda) \tag{8}$$

When (7) and (8) are combined, (9) is derived

$$P_{\omega t} = 0.5\rho A v^3 C_p(\beta, \lambda) \tag{9}$$

where $C_p(\beta, \lambda)$ is the turbine power coefficient and β is the blade pitch and λ is the TSR. The power coefficient C_p having highly non-linear and changing wind speeds is given as follows:

$$C_p = 0.5176\left(\frac{116}{\lambda_i} - 0.4\beta - 5\right)\mathrm{e}^{(-21/\lambda_i)} + 0.0068\lambda \tag{10}$$

C_p value is found by λ_i value in (11) substituted in (10). Here, λ_i value that is an intermediate variable is used to simplify the calculations and expressed as below equation:

$$\frac{1}{\lambda i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{3\beta + 1} \tag{11}$$

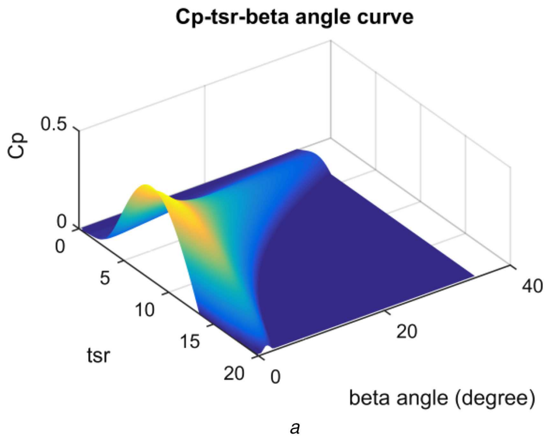
The TSR is the mechanical angular speed of blades to the wind speed and is given as follows:

$$\lambda = \frac{\omega_{\omega t} R}{v} \tag{12}$$

where $\omega_{\omega t}$ is the angular speed of the turbine rotor (rad/s) and R is the blade radius of turbine (m). The curve of the power coefficient and the TSR for the various pitch angles values are given in Fig. 3*a*.

Any change in the rotor speed or the wind speed in WTs varies the TSR and thus the power coefficient varies. As a result of these changes, the obtained amount of power also varies. According to (11) and (12), the power coefficient is changed via the blade pitch. This is an essential control strategy for the power control of large WTs.

The mechanical output power in a WT with variable speed also varies [33]. As shown in Fig. 3*b*, there are four operating zones in the output power curve of the WTs with variable blade pitch and variable speed. In the first zone of them, the wind speed is smaller than the cut-in speed and here, the output power is 0. The second zone is between the cut-in and the nominal speed. The third zone is between the nominal speed and the cut-in speed. The fourth zone is beyond the cut-out speed and the WT is stopped in terms of security in this region [34]. A maximum power point tracing is desired in the second zone. In order to derive maximum energy



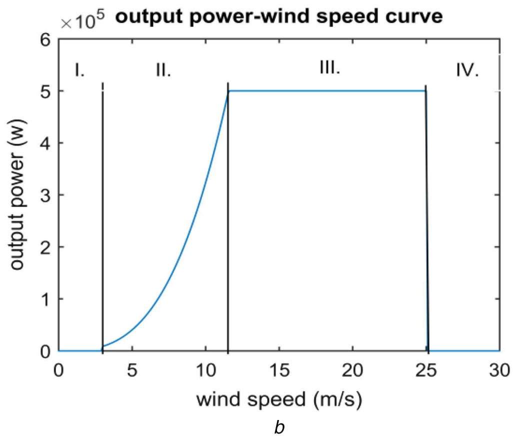


Fig. 3 *C_p, TSR and WT operation zone curves*
(a) Curve of C_p and the TSR for the various β values, *(b)* WT operation zones

Table 1 Some widely used Benchmark function equations

Function name	Function	Limits
Sphere function	$f(x) = \sum_{i=1}^{n} x_i^2$	$f(x_1, \ldots, x_n) = f(0, \ldots, 0) = 0$
Ackley's function	$f(x,y) = -20\,\mathrm{e}^{-0,2\sqrt{0,5(x^2+y^2)}} - \mathrm{e}^{0,5(\cos(2\pi x)+\cos(2\pi y))} + e + 20$	$f(1, 1) = 0$
Lévi function	$f(x,y) = \sin^2(3\pi x) + (x-1)^2(1+\sin^2(3\pi y))$ $+(y-1)^2(1+\sin^2(2\pi y))$	$f(1, 1) = 0$
Beale's function	$f(x,y) = (1,5 - x + xy)^2 + (2,25 - x + xy^2)^2 + (2,625 - x + xy^3)^2$	$f(3, 0, 5) = 0$
Goldstein–Price function	$f(x,y) = (1 + (x+y+1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2))$ $(30 + (2x-3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2))$	$f(0, -1) = 3$
Booth's function	$f(x,y) = (x + 2y - 7)^2 + (2x + y - 5)^2$	$f(1, 3) = 0$

obtained from wind, power electronic equipment supply a constant frequency and a voltage, has to be inserted into a grid and a WT [35]. At the beginning of the third zone, at the nominal wind speed, when the WT reach to the nominal power, if the wind speed increase, the output power will increase. Therefore, in order to stabilise the output power within the design limits, a control system is needed. The blade pitch control changing the power coefficient and then varying the output power is carried out by the control system. The output power is struggled through by increasing the blade pitch [32].

3 Performance analysis of algorithms

Currently, combinations of the modern optimisation methods with conventional PID controllers have become more important in terms of speed and control of a system. Therefore, in this study, the performance of parameter adjusting the PID controller was compared to the standard GA with the IGA above the nominal wind speeds in the blade pitch control of WTs.

While the performance analysis of the algorithms was being carried out in this study, some widely used Benchmark functions in Figs. 4*a–f*, which are the graphics of them, and in Table 1, which are equations of them, were employed [36].

During the tests, the population number and the crossover rate of each algorithm were defined as 50 and 70%, respectively. The mutation rate of the standard GA was taken as 8% and a one-point crossover was preferred. According to the algorithm approach, the mutation rate of the IGA was defined between 8 and 200% by the written program. In addition, the IGA was utilised from one-point crossover to n crossover in (3). For each algorithm result, 1000 iterations were fulfilled, the iterations were repeated 10 times and then the averages of them were taken.

Data obtained from the implementation of some Benchmark functions were given in Table 2. The expected results in Table 2 show the global minimum value of the function. The standard GA and IGA columns also show how the results approach to the expected results when the algorithm is worked 1000 iterations. The results in Table 2 show that the IGA exhibited a better performance than the standard GA for each function. Although the standard GA in such as Goldstein-Price has not approached the optimum result, the optimum results have been achieved with a great success by the IGA.

For example, the expected result was 0 for sphere function. Namely, the minimum point of the function was 0. At the end of 1000 iterations, while the obtained result of the standard GA was $4.608798059342836 \times 10^{-6}$, the obtained result of the IGA was $6.072540845517372 \times 10^{-10}$. The IGA ensured a better approach to the optimum result.

In order to understand and verify the accuracy and performance of results, some similar studies have been investigated in the literature [29], it was seen that similar results were obtained. According to this, the test functions discussed in [29] were operated by our IGA optimisation method functions. These functions are given in Table 3.

Table 2 The results in the implementation applying the GAs to the benchmark functions

Function name	Expected results	Standard GA	IGA
Sphere function	0	$4.608798059342836 \times 10^{-6}$	$6.072540845517372 \times 10^{-10}$
Ackley's function	0	0.108638094119376	$1.199019946451330 \times 10^{-4}$
Lévi function	0	0.068785432352108	$4.799099684833500 \times 10^{-6}$
Beale's function	0	0.360277722638943	0.050242789240740
Goldstein–Price function	3	$1.090268794572235 \times 10^{7}$	3.000000074160503
Booth's function	0	0.736320222869314	0.014548004549668

Table 3 Some test functions [29]

Functions	Equations	Intervals
function 1	$f_1\text{: } \max f(x,y) = 1 + x \cdot \sin(4\pi x) - y \cdot \sin(4\pi y + \pi) + \frac{\sin\left(\sqrt[6]{x^2 + y^2}\right)}{\sqrt[6]{x^2 + y^2 + 10^{-15}}}$	$x, y \in [-1, 1]$
function 2	$f_2 = \max f(x,y) = 3 \cdot (1-x)^2 \cdot e^{[-x^2-(y+1)^2]} - 10 \cdot \left(\frac{x}{5} - x^3 - y^5\right) \cdot e^{(-x^2-y^2)} - \frac{1}{3} \cdot e^{[-(x+1)^2 - y^2]}$	$x, y \in [-3, 3]$
function 3	$f_3 = \max f(x,y) = \sin(5,1 \cdot \pi \cdot x)^{30} \cdot e^{\left[-4.\mathrm{lg}2(x-0,0667)^2/0,64\right]} \cdot \sin(5,1 \cdot \pi \cdot y)^{30} \cdot e^{\left[-4.\mathrm{lg}2(x-0,0667)^2/0,64\right]}$	$x, y \in [0, 1]$
function 4	$f_4 = \max\ f(x,y) = -\left[20 + x^2 - 10 \cdot \cos(2\pi x) + y^2 - 10 \cdot \cos(2\pi y)\right]$	$x, y \in [-5, 5]$

Table 4 The results applying the test functions to the algorithms and compared results with [29]

	NFO: Number of finding the optimum					
	SGA	BSGA	DMGA	AGA	SOGA	Proposed IGA results
f_1	53	1000	997	1000	1000	1000
f_2	38	1000	969	998	998	1000
f_3	0	998	861	905	997	1000
f_4	0	11	150	10	1000	996
mean	22.75	752.25	744.25	728.25	998.75	**999**

The standard GA (SGA), the adaptive GA (AGA) [37], the improved GA or abbreviated GA (BSGA), the GA with adaptive mutation rate or abbreviated GA (DMGA) and the self-organising GA (SOGA) [29] functions were used for tests in this study. As above, the IGA was also added to Table 4 that demonstrates how many the optimal result of the GA was found at the conclusion 1000 iterations. These obtained results are given in Table 4. According to Table 4, though the IGA performed worse performance than SOGA recommended in [29] in f_4 function, it performed better performance in f_2 and f_3 functions. According to the overall average, the IGA was slightly better than SOGA and was better than the others.

4 Simulation results and discussion

First of all, a WE system was designed by MATLAB/Simulink software in this study. In Fig. 5, a WT MATLAB/Simulink model was demonstrated to simulate a WT system. Here, a wind system turbine and generator block was embedded the equations from (7)–(12). The simulated WT system parameters were presented in Table 5. There are two MATLAB functions that are a MATLAB function and MATLAB function 1. The blocks employed for similar tasks. One of them started up the control system over nominal wind speeds; other cut-out the control system above nominal wind speeds. The servo motor block was utilised for an angular position formula. To be simple, this was expressed by $1/(s(s+1))$. A desired operating power value of the WT was entered a sp block. A wind speed variable in Fig. 6*a* was applied to the designed MATLAB/Simulink system by means of a wind speed block. An error signal between the desired operating power value and the produced value was obtained. If wind speed was above at nominal wind speeds, the error was given a PIC control block, then a pitch angle was calculated and after that the calculated value was entered a servo motor block. The servo motor that adjusts the blade pitch was operated by the signal. Thereby, so as to limit the output power of the system in the set point, the blade pitch was adjusted by the controller as shown in Fig. 6*b*. The servomotor output sent the pitch angle value to the wind energy system. Thus, the output power was reduced the desired value. Ultimately, the error signal was transferred MATLAB/Workspace to use the GA and the IGA by taking the error signal absolute value. The values were evaluated by the GA the IGA and so a fitness value was generated according to the PID parameter values. These results were given in Fig. 7*b*. The fitness values of the IGA in the results of iterations were produced better than the GA results.

The WT MATLAB/Simulink model was activated by the optimised PID parameters tuned both the GA and the IGA. According to these activated algorithms, both of their WT output powers were shown in Fig. 8. Three enlarged statuses of the WT output powers in some regions were indicated in Figs. 8*a–c*. Due to the fact that the IGA was achieved better a fitness value than the GA, the output power stability of the system was obtained a better result as expected. For example, when Fig. 8*a* was analysed, while the GA overlap value was seen as 5.1198×10^5, the IGA overlap value was achieved as 5.1032×10^5. According to the results, here, an appreciable improvement of 17% was calculated. While the improvement would correspond to a power of 8.5 kW in a system of 0.5 MW, it would correspond to a power of 85 kW in a system of 5 MW. Namely, the improvement would mean that a system optimised by the IGA would give a better efficiency about 17% than a system optimised by the GA at each wind speed change being at about nominal wind speeds.

Furthermore, although the GA stability time was taken as 15.40 s, the IGA stability time was succeeded as 14.61 s. As a good result, the response stability time difference between the system optimised by IGA and the system optimised by GA was calculated 0.79 s. Consequently, even though the response time depends on receiving wind speeds and wind direction changes, the achieved results would give an idea that the superiority of the IGA than the GA when taking into account suddenly wind speeds and direction changes in wind.

5 Conclusion

In this study, in order to define PID parameters used for the blade pitch control, which controls the output power of a WTG, a newly developed IGA structure was employed. A new IGA algorithm structure, which perceives the local minimums and maximums, then changes simultaneously both the mutation rate and crossover point values in the IGA structure, was suggested and verified. After proving the superiority of the IGA structure to the standard GA in the test functions, it was applied to a WT system. According to the MATLAB/Simulink software simulation results, it was observed that the IGA was better than Ziegler–Nichols GA being classic PID parameter calculation method in the PID parameter optimisation. According to the optimisation results, the system optimised by the IGA gave a better efficiency about 17% than the system optimised by the GA. During the simulations, since MATLAB/Simulink model was directly used, optimisation of PID parameters of other systems was facilitated by means of the IGA structure. As a result,

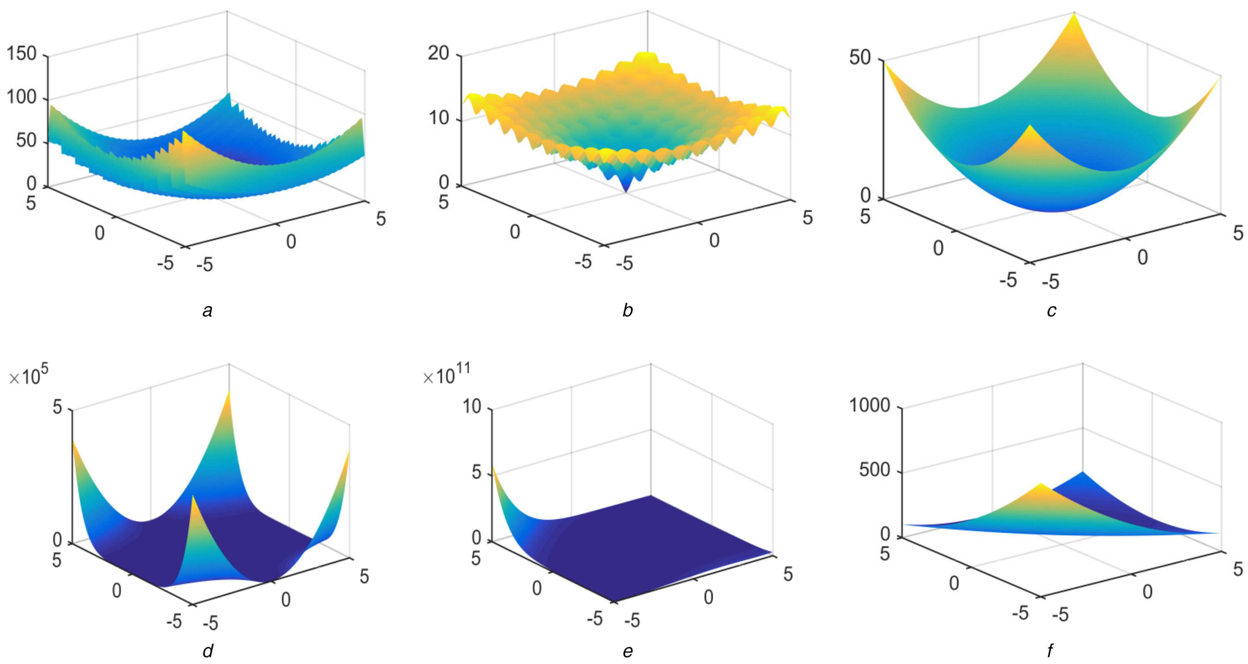


Fig. 4 *Some widely used Benchmark function graphics*
(a) Lévi function, ***(b)*** Ackley's function, ***(c)*** Sphere function, ***(d)*** Beale's function, ***(e)*** Goldstein–Price function, ***(f)*** Booth's function

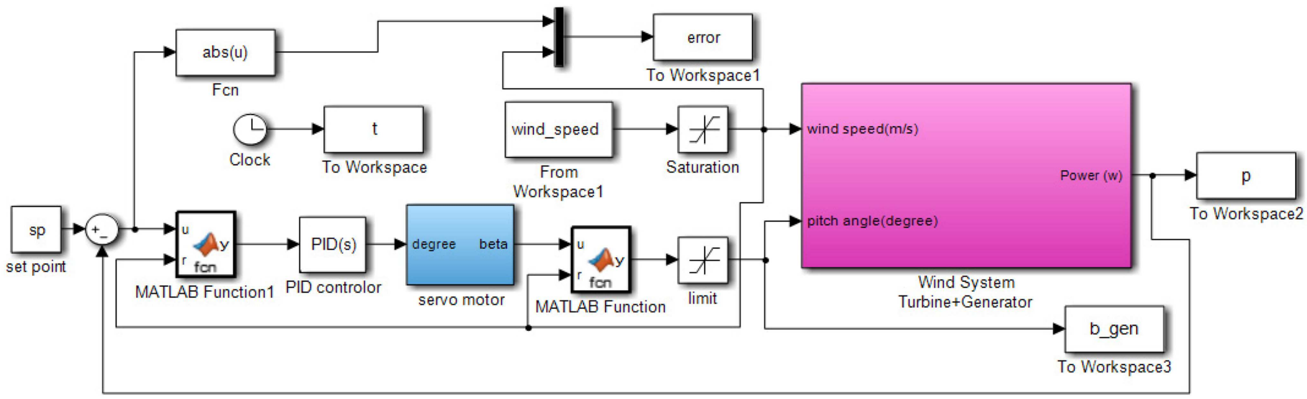


Fig. 5 *WT MATLAB/Simulink model*

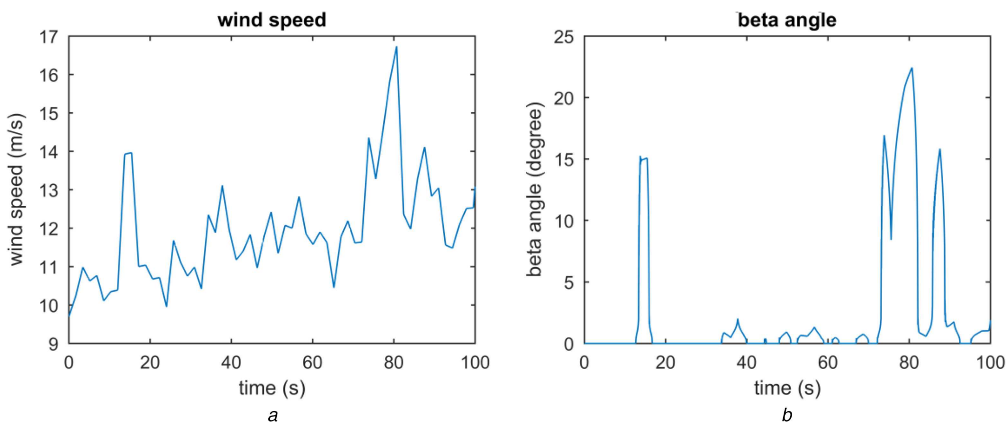
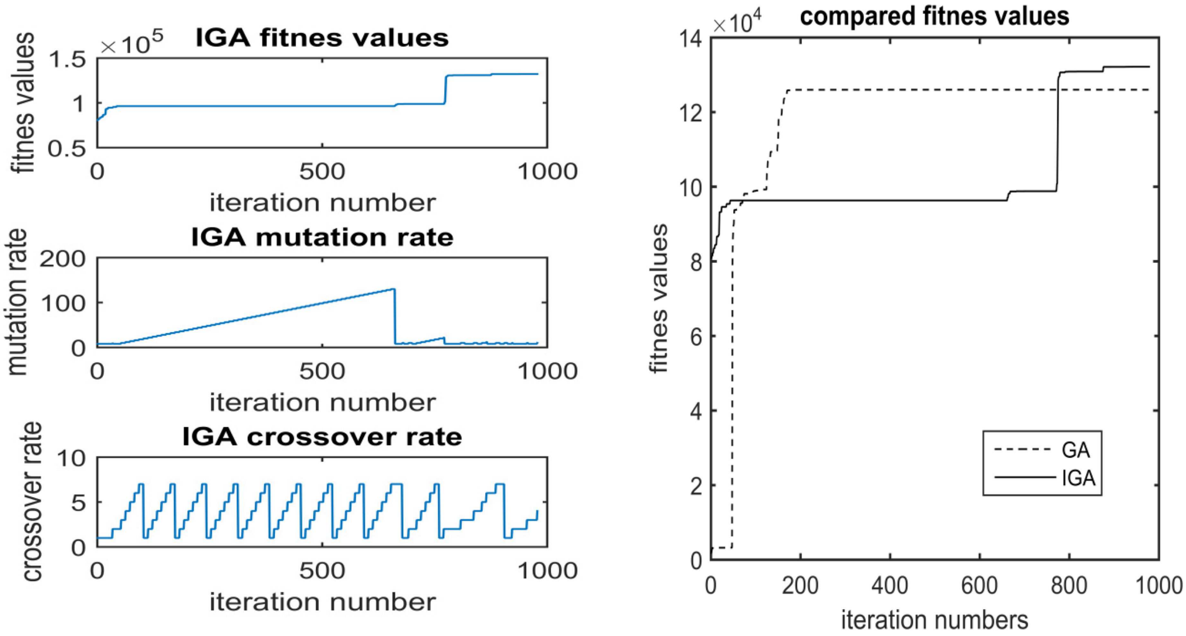


Fig. 6 *Wind speed and blade pitch change curves*
(a) Curve of wind speed change, ***(b)*** Curve of blade pitch change

any system having the MATLAB/Simulink model could be optimised by the IGA structure with small modifications.

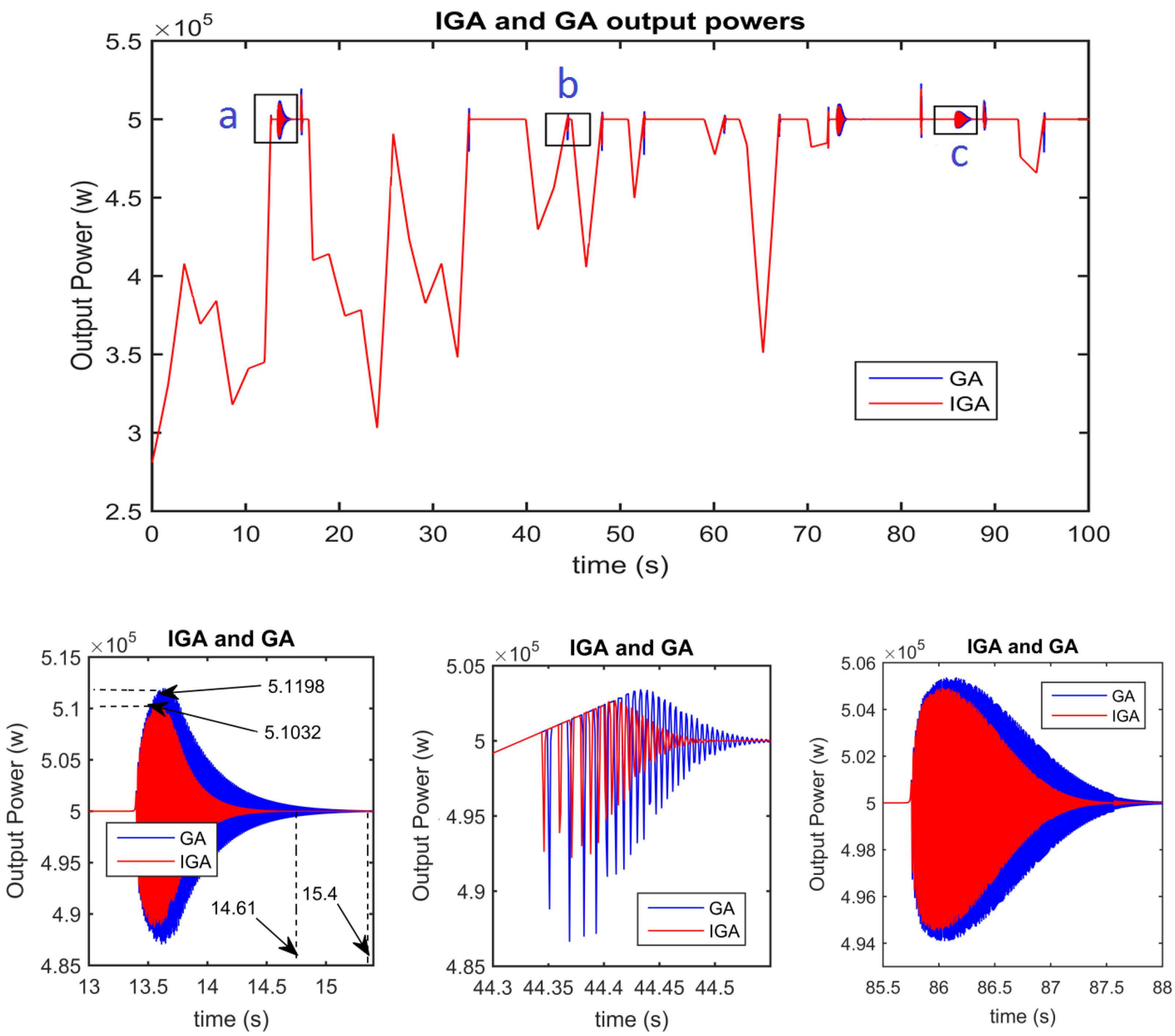
6 Acknowledgment

This project was supported by the Scientific and Technological Research Council of TURKEY (TUBITAK) under Grant 114E419 ('Design, optimization and experimental verification of a permanent magnet synchronous generator with external rotor, maximum power and minimum cogging torque').



a *b*

Fig. 7 *Compliance and fitness value variations*
(a) Compliance value variations as a function of mutation rate and crossover point number, ***(b)*** Variations of the fitness values of the GA and the IGA



a *b* *c*

Fig. 8 *Optimised WT output powers of IGA and GA*
(a) Zoomed figure region a, ***(b)*** Zoomed figure region b, ***(c)*** Zoomed figure region c

Table 5 The simulated WT system parameters

Simulated system parameters	
nominal output power	500 kW
working mode	network connection
cut in wind speed	3 m/s
nominal wind speed	12 m/s
cut out wind speed	25 m/s
rotor diameter	48 m
sweep area	1810 m^2
blade number	3
nominal rotor speed	30 rpm
rotor speed range	10–30 rpm
gear box rate	01:50
generator number	2
generator type	PMSG
generator nominal output	250 kW
generator nominal cycle	1500 rpm
generator voltage	690 V

7 References

[1] Burton, T., Sharpe, D., Jenkins, N.*, et al.*: '*Wind energy handbook*' (John Wiley & Sons Press, 2001, 1st edn.)

[2] Jelavic, M., Petrovic, V., Peric, N.: 'Estimation based individual pitch control of wind turbine', *Automatika*, 2010, **51**, (2), pp. 181–192

[3] Qi, J., Liu, Y.: 'PID control in adjustable-pitch wind turbine system based on fuzzy control'. Proc. 2010 Second Int. Conf. on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pp. 341–344

[4] Kong, Y., Wang, Z., Yuan, H.: 'A new fuzzy sliding-mode control for MW variable speed-variable pitch wind turbine'. Proc. Seventh IET Int. Conf. on Advances in Power System Control, Operation and Management (APSCOM 2006), Hong Kong, China, October–November 2006, pp. 256–260

[5] Amendola, C., Gonzaga, D.: 'Fuzzy-logic control system of a variable-speed variable-pitch wind-turbine and a double-fed induction generator'. Proc. Seventh Int. Conf. on Intelligent Systems Design and Applications (ISDA), Rio de Janerio, Spain, October 2007, pp. 252–257

[6] Muhando, E.B., Senjyu, T., Yona, A.*, et al.*: 'Disturbance rejection by dual pitch control and self-tuning regulator for wind turbine generator parametric uncertainty compensation', *IET Control Theory A*, 2007, **1**, (5), pp. 1431–1440

[7] Yao, X., Liu, Y., Guo, C.: 'Adaptive fuzzy sliding-mode control in variable speed adjustable pitch wind turbine'. Proc. 2007 IEEE Int. Conf. on Automation and Logistics, Jinan, China, August 2007, pp. 313–318

[8] Zhang, L., Li, H., Chunliang, E.*, et al.*: 'Pitch control of large scale wind turbine based on fuzzy-PD method'. Proc. 2007 IEEE Int. Conf. on Automation and Logistics, Jinan, China, August 2007, pp. 2447–2452

[9] Jian-Jun, X., Li-Mei, Y., Xiao-Ning, Q.*, et al.*: 'Study of variable-pitch wind turbine based on fuzzy control'. Proc. 2010 Second Int. Conf. on Future Computer and Communication (ICFCC), Wuhan, China, May 2010, pp. 235–239

[10] Dou, Z., Cheng, M., Ling, Z.*, et al.*: 'An adjustable pitch control system in a large wind turbine based on a fuzzy-PID controller'. Proc. 2010 Int. Symp. on Power Electronics Electrical Drives Automation and Motion (SPEEDAM), Pisa, Italy, June 2010, pp. 391–395

[11] Jiao, B., Wang, L.: 'RBF neural network sliding mode control for variable-speed adjustable-pitch system of wind turbine'. Proc. 2010 Int. Conf. on Electrical and Control Engineering, Wuhan, China, June 2010, pp. 3998–4002

[12] Kim, J.S., Jeon, J., Heo, H.: 'Design of adaptive PID for pitch control of large wind turbine generator'. Proc. 2011 Tenth Int. Conf. on Environment and Electrical Engineering (EEEIC), Rome, Italy, May 2011, pp. 1–4

[13] Katsigiannis, Y., Georgilakis, P., Karapidakis, E.: 'Multiobjective genetic algorithm solution to the optimum economic and environmental performance problem of small autonomous hybrid power systems with renewables', *IET Renew. Power Gener.*, 2010, **4**, (5), pp. 404–419

[14] Sundareswaran, K., Vigneshkumar, V., Palani, S.: 'Development of a hybrid genetic algorithm/perturb and observe algorithm for maximum power point tracking in photovoltaic systems under non-uniform insolation', *IET Renew. Power Gener.*, 2015, **9**, (7), pp. 757–765

[15] Zhao, M., Chen, Z., Blaabjerg, F.: 'Optimisation of electrical system for offshore wind farms via genetic algorithm', *IET Renew. Power Gener.*, 2009, **3**, (2), pp. 205–216

[16] Yassin, K., Diab, A., Ghoneim, Z.: 'Aerodynamic optimization of a wind turbine blade designed for Egypt's Saharan environment using a genetic algorithm', *Renew. Energy Sustain. Dev.*, 2015, **1**, (1), pp. 106–112

[17] Jiang, P., Li, X., Dong, Y.: 'Research and application of a new hybrid forecasting model based on genetic algorithm optimization: A case study of Shandong wind farm in China', *Math. Probl. Eng.*, 2015, **2015**, pp. 1–14

[18] Zhang, B., Li, X., Wang, S.: 'A novel case adaptation method based on an improved integrated genetic algorithm for power grid wind disaster emergencies', *Expert Syst. Appl.*, 2015, **42**, (21), pp. 7812–7824

[19] Chen, Y., Li, H., He, B.*, et al.*: 'Multi-objective genetic algorithm based innovative wind farm layout optimization method', *Energy Convers. Manage.*, 2015, **105**, (November), pp. 1318–1327

[20] Gao, X., Yang, H., Lin, L.*, et al.*: 'Wind turbine layout optimization using multi-population genetic algorithm and a case study in Hong Kong offshore', *J. Wind Eng. Ind. Aerodyn.*, 2015, **139**, (April), pp. 89–99

[21] Khatua, K., Yadav, N.: 'Voltage stability enhancement using VSC-OPF including wind farms based on Genetic algorithm', *Int. J. Electr. Power*, 2015, **73**, (December), pp. 560–567

[22] Åström, K.J., Hägglund, T.: 'The future of PID control', *Control Eng. Pract.*, 2001, **9**, (11), pp. 1163–1175

[23] Li, Y., Ang, K.H.: 'PID control system analysis and design', *IEEE Control Syst. Mag.*, 2006, **26**, (1), pp. 32–41

[24] Civelek, Z., Lüy, M., Çam, E.*, et al.*: '*Control of pitch angle of wind turbine by fuzzy PID controller*' (Intell. Autom. Soft Co., 2015), pp. 1–9

[25] Mamur, H.: 'Design, application and power performance analyses of a micro wind turbine', *Turk. J. Electr. Eng. Comput. Sci.*, 2015, **23**, (6), pp. 1619–1637

[26] Hasanien, H.M.: 'Design optimization of PID controller in automatic voltage regulator system using Taguchi combined genetic algorithm method', *IEEE Syst. J.*, 2013, **7**, (4), pp. 825–831

[27] Qi, S.R., Wang, D.F., Han, P.*, et al.*: 'Grey prediction based RBF neural network self-tuning PID control for turning process'. Proc. 2004 Int. Conf. on Machine Learning and Cybernetics, Shangai, China, August 2004, pp. 802–805

[28] Rubaai, A., Young, P.: 'EKF-based PI-/PD-like fuzzy-neural-network controller for brushless drives', *IEEE Trans. Ind. Appl.*, 2011, **47**, (6), pp. 2391–2401

[29] Zhang, J., Zhuang, J., Du, H.: 'Self-organizing genetic algorithm based tuning of PID controllers', *Inf. Sci.*, 2009, **179**, (7), pp. 1007–1018

[30] Soltoggio, A.: 'An enhanced GA to improve the search process reliability in tuning of control systems'. Proc. the 7th Annual Conf. on Genetic and Evolutionary Computation, Seattle, Washington, USA, July 2005, pp. 2165–2172

[31] Tong, W.: '*Wind power generation and wind turbine design*' (Wit Press, 2010, 1st edn.)

[32] Hemami, A.: '*Wind turbine technology*' (Cengage Learning, 2012, 1st edn.)

[33] Chen, C.H., Hong, C.M., Ou, T.C.: 'Hybrid fuzzy control of wind turbine generator by pitch control using RNN', *Int. J. Ambient Energy*, 2012, **33**, (2), pp. 56–64

[34] Hwas, A.M.S., Katebi, R.: 'Wind turbine control using PI pitch angle controller'. Proc. IFAC Conf. on Advances in PID Control PID'12, Brescia, Italy, March 2012

[35] Wenjing, Z., Hongze, X.: 'Active disturbance rejection based pitch control of variable speed wind turbine'. Proc. 2011 30th Chinese Control Conf. (CCC), Yantai, China, July 2011, pp. 5094–5098

[36] Bhushan, B., Pillai, S.S.: 'Particle swarm optimization and firefly algorithm: performance analysis'. Proc. 2013 IEEE Third Int. Advance Computing Conf. (IACC), Ghaziabad, India, February 2013, pp. 746–751

[37] Zhang, J., Chung, H.S.H., Lo, W.L.: 'Clustering-based adaptive crossover and mutation probabilities for genetic algorithms', *IEEE Trans. Evol. Comput.*, 2007, **11**, (3), pp. 326–335