Advanced Pitch Angle Control Based on Genetic Algorithm and Particle Swarm Optimisation on FAST Turbine Systems

Goksu Gorel*, Mahdi O. Abdi
Department of Electrical and Electronics Engineering, Cankiri Karatekin University, Uluyazi Campus, 18100, Fatih, Cankiri, Turkey
goksugorel@karatekin.edu.tr, mahdiosman97@gmail.com

***Abstract*—In this paper, the increase in the quality of the rotor speed of wind turbines and the decrease in mechanical loads on the turbines are investigated. Adjusting the angle of the blade to the nominal wind speed, the rotor speed of the wind turbine is maintained at its nominal value. Using control methods (such as proportional integral (PI), genetic algorithms (GAs), and particle swarm optimisation (PSO)), different results can be recovered. In addition, individual control of the blade tilt angle allows us to reduce the mechanical loads on the turbine with the control methods. The wind turbine was modelled in Matlab/Simulink. The simulation results show that individual control of the blade tilt angle ensures the quality of the rotor speed of the wind turbine and reduces the balanced periodic loads on the wind turbine. In the first part, we study the wind turbine in a global way, as well as the method used to calculate them. Then, we discuss the FAST system, which was used to model the wind turbine, as well as the design of individual pitch angle control. As a result, it is possible to reduce the fatigue of the mechanical wind turbine parts. According to the study, the mechanical load for all three blades was reduced by an average of 44 % compared to the PI and PSO methods and by 1 % compared to the PI and GA methods. The control of the pitch angle in wind energy systems is performed with different control methods. The study analysis of the mechanical loads found that they are largely balanced. Winds that blow perpendicular to the turbine blades on the x-axis provide these loads.**

***Index Terms*—Wind power generation; Genetic algorithms; Particle swarm optimisation; PI control; FAST system.**

I. INTRODUCTION

The wind turbine is a device that transforms the kinetic energy of the wind into mechanical energy, known as wind energy, which is then transformed into electrical energy. Wind energy existed before the invention of the windmill in the 9th century. Over the decades, the wind turbine has been refined to the point that it may now be used in industrial settings. In 1890, Dane Paul La Cour invented the first "industrial" wind turbine to generate energy, which he used to manufacture hydrogen through electrolysis. To reduce the load of the wind turbine, anticipate the load to be reduced. This is achieved by individual pitch control [1].

The National Renewable Energy Laboratory (NREL) must ensure that the energies of its wind turbines are stabilised without too much loss of energy, hence the real-time dynamic control of the wind turbines is of great importance. This research examines a simple wind turbine model with a depth control system with a nonlinear receding horizon [2], [3]. For the development of the system, the comparison work on wind turbine dynamics is based on the NREL 5MW reference wind turbine model, where the results reveal that a nonlinear real-time receding horizon control on the wind turbine regulates the dynamics (in real time). This illustrates the importance of selecting a specialised controller that allows for dynamic control. This paper provides a detailed examination of a small wind turbine model that was validated before being used to create a nonlinear receding horizon control system [3]–[6].This shows the difference in the usefulness of using a specific controller for dynamic control. The purpose of pitch angle control is to achieve sufficient power. In this case, the cost is low and the performance is efficient [7], [8].

The harsh environmental conditions to which wind turbine components are exposed, particularly those located offshore, cause significant fatigue. The blades of the wind turbine are subjected to considerable gravitational, inertial, and aerodynamic loads, which cause fatigue and degradation over the lifetime of the turbine. The current study investigates modelling this phenomenon utilising blade root moment data from a sensor in a high-fidelity simulator of an industrial-scale wind turbine. In recent years, there has been a surge in interest in combining control with fatigue load minimisation. The research reported in this paper investigates the use of a model predictive control (MPC) in conjunction with a fatigue-based prognostic strategy to reduce damage to wind turbine components (blades). The wind turbine now has a mechanism to run securely and optimise the trade-off between component life and energy production thanks to the integration of a system health management module with the MPC control. The controller objective was changed by including a new criterion that considers cumulative damage. The findings reveal that there is a trade-off between maximal power and accumulating damage [9], [10].

These control algorithms boost the blade pitch actuation, primarily to lower the 1P (once per revolution) aerodynamic load component. However, it is well recognised that blade pitch system failure is a primary cause of turbine downtime. The pitch actuation will only be exacerbated by control

Manuscript received 12 March, 2023; accepted 14 June, 2023.

algorithms that increase the pitch actuation [11], [12]. As a result, increasing the pitch actuation of the blade to minimise mechanical stresses may not always be the best option. Individual pitch control allows for reduced pitch actuation while following the rotor speed. Individual pitch control (IPC) can regulate rotor speed at higher speeds with less pitch actuation. In addition, the IPC can limit forward and backward movement, as well as rotation. Compared to the basic controller, the suggested IPC can maintain the nominal rotor speed with less pitch actuation. With increasing wind speeds, the performance improves. Reduced blade pitch actuation is proposed to improve rotor speed control. Furthermore, the displacement of blades out-of-plane, the displacement of the fore-aft tower, and the rotation of the pitch of the platform can be reduced [13], [14].

The purpose of this research is to develop two control techniques for a variable pitch wind turbine system and compare their performance in real-world conditions [15]–[17]. The wind turbine calculation code FAST is used to implement two linear and nonlinear models in SIMULINK. FAST is a wind turbine simulator made by the American Renewable Energy Laboratory. FAST can be run in Matlab/Simulink and written in Fortran. FAST programmes can be linked to standard Matlab programmes. Thus, due to FAST, equations of motion written with S-functions can be combined with the Simulink model. This provides great flexibility throughout simulation in wind turbine control applications. It allows modules for generator torque control, yaw control, and pitch angle control to be designed using FAST. Thus, all nonlinear wind turbine equations can be used in simulations. A linear model and a nonlinear model, both created in SIMULINK with the FAST wind turbine computation package, are used to develop the control methods. Subsequently, they are installed in the wind turbine control module, and operating data are recorded for around a month for each. A proportional integral (PI) control technique is the first control strategy [18]. It was chosen for its ease of design and application, as well as its high performance. Gain scheduling was required to obtain good angular velocity control. The behaviour of the nonlinear model is validated using the operating data of the PI strategy.

II. MATERIALS AND METHODS

A wind turbine captures the kinetic energy of the wind and converts it into a torque that turns the rotor blades. Three factors determine the ratio between the wind energy and the mechanical energy recovered by the rotor: the density of the air, the area swept by the rotor, and the wind speed [19]–[21].

This captured energy from the air density (moving air with energy) is shown in Fig. 1. The following equation shows relation between captured energy and air density, power coefficient, turbine swept area

$$P = 0.5\rho A v^3 A, \tag{1}$$

where P – mechanical power in moving air (W), ρ - air density (kg/m^3), A – area swept by rotor blades (m^2), v – velocity of air (m/s), and C_p – power coefficient.

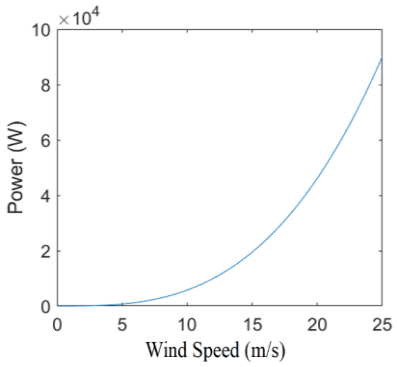

Fig. 1. Wind speed wind power curve.

According to a scientist named Betz, the maximum power that an ideal infinite-blade turbine rotor can extract from the wind under perfect conditions is 59.26 % of the power available in the wind. For structural and economic reasons, wind turbines are designed with two or three blades and can generate nearly 50 % of the power [22], [23].

If (1) is added to (2), (3) is obtained and shown:

$$P_{\omega t} = PC_p(\beta, \lambda), \tag{2}$$

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

where C_p (β, λ), β, and λ are the power coefficient, the pitch angle of the blade, and the tip speed ratio (TSR) of the blade of the turbine, respectively, in Fig. 2.

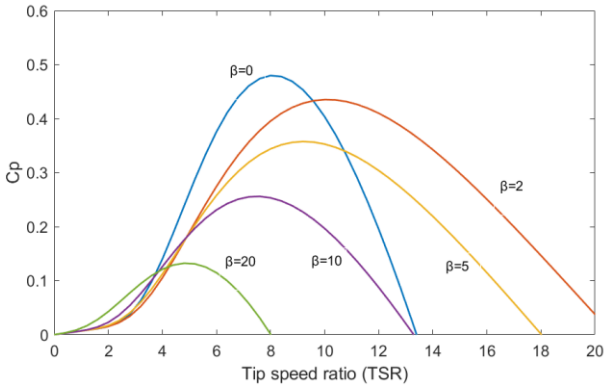

Fig. 2. Cp-λ plot for different values of β.

The power coefficient C_p of the wind turbine is given in (4)

$$C_p = 0.5176\left(\frac{116}{\lambda} - 0.4\beta - 5\right)e^{\frac{-21}{\lambda_i}} + 0.0068\lambda, \tag{4}$$

where C_p is the power coefficient of the wind turbine, β is the pitch angle of the blade, and λ is the speed ratio of the tip. The value of C_p is very nonlinear, varying with wind speed, turbine rotation speed, and turbine blade parameters such as pitch angle [23], [24].

FAST is a wind turbine simulator built by the US Renewable Energy Laboratory. The simulator is written in FORTRAN and can be run as an S-function in Matlab/Simulink. Standard Matlab subroutines can be linked to FAST subroutines, allowing the FAST equations of motion written in S-functions to be combined with the Simulink model. This provides great flexibility throughout simulation in wind turbine control applications. It allows the

design of generator torque control, motor ground yaw control, and pitch angle control modules in Simulink. Thus, the wind turbine equations for all nonlinear airborne forces in FAST can be used in simulations [25], [26].

FAST also analyses the mechanical load of the wind turbine and allows individual pitch angle control. The Simulink model of FAST is shown in Fig. 3 and the OpenLoop model is shown in Fig. 4.

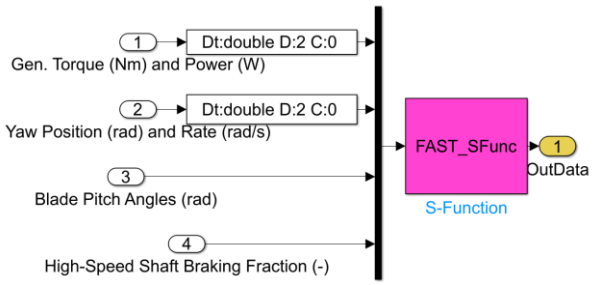

Fig. 3. FAST wind turbine block.

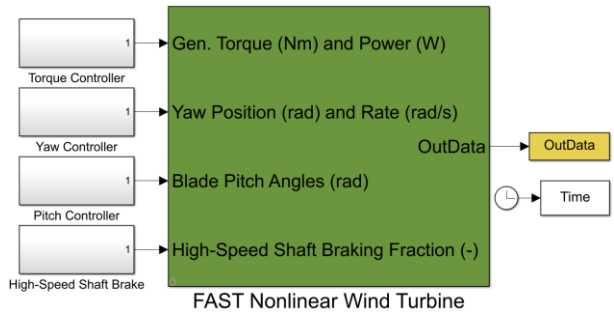

Fig. 4. Simulink model of OpenLoop.mdl.

III. METHODS OF CONTROL

1. *PI control*

The L and T values used in computing the necessary coefficients for the PI and proportional-integral-derivative (PID) controllers were found using the output power curve data in the Matlab environment, as shown in Fig. 5. The wind turbine output power curve is broken into four zones in the Matlab environment by looking at the system without applying it to the controller. The x-axis in the cut-out part between regions I and II is indicated by the L value. The L and T values are always different when using a PI or PID controller. The wind system values calculated or stated in this section must have a pitch angle of zero. Subsequently, using the methods in Table I, K_p (Proportional gain), K_i (Integral gain), and K_d (Derivative gain) were easily derived [27].

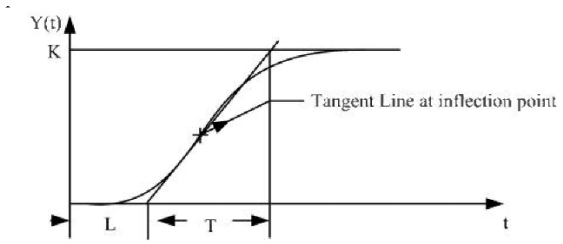

Fig. 5. Method for determining coefficients for PID controller.

To determine K_p, K_i, and K_d, the PI and PID coefficients were derived using the closed-loop system according to Table I. Looking at the Matlab simulation, yields the L and T values. The coefficients K_p, K_i, and K_d were determined after identifying the L and T values of the closed-loop system according to Fig. 5 by looking at the output power curve in Matlab in simulation. This approach is not used in this study because the system does not contain an open-loop. The closed-loop approach is shown below. The closed-loop approach of the PI and PID controller has been determined as shown in Table I. The coefficients K_p, K_i, and K_d were derived after identifying the L and T values of the closed-loop system according to Fig. 5 by looking at the output power curve in Matlab in simulation. [28].

TABLE I. CALCULATION OF PI AND PID CLOSED-LOOP COEFFICIENTS.

Control type	K_p	K_i,	K_d,
P	0.50, K_c	0	0
PI	0.45, K_c	1.2 $K_p dT/P_c$	0
PID	0.60, K_c	2 $K_p dT/P_c$	$K_p P_c/(8dT)$

Applying the Ziegler-Nichols closed-loop method and looking at the Ku and Tu values in Matlab, Ku = -1/300 and Tu = 400, so K_p, K_i, and K_d are calculated.

Displayed as PI regulator's transfer function. The input signal to the controller is generated by an intrinsic portion of the controller. The PI controller reduces the steady-state error of the system to some extent.

The PI controller transfer function is described in (5). The Ziegler-Nichols method is used in this study to optimise the output power of the wind turbine and keep it at the setpoint. Only K_p and K_i are present in the PI controller

$$G_c(s) = K_p + \frac{K_i}{s}. \quad (5)$$

The PID controller's transfer function is described in (6). In this case, the Ziegler-Nichols approach optimises the wind turbine output power set point more effectively than the PI controller. K_p, K_i, and K_d are the three components of the PID controller

$$G_c(s) = K_p + \frac{K_i}{s} + K_d S. \quad (6)$$

2. *Genetic algorithm*

As is well known, GA systems typically employ a binary coding format. Additionally, a starting population is created at random. Figure 6 displays the GA flow diagrams.

The fitness of each chromosome is measured, and a survival of the fittest approach is used. The error value is utilised to assess the fitness of each chromosome in this study. A genetic algorithm has three main operations: breeding, crossover, and mutation. Figure 6 depicts the series of operations involved in GA.

Measure fitness: The fitness of a chromosome is assessed using a fitness metric. A chromosome with a higher fitness value has a higher chance of contributing to one or more offspring in the future generation, according to the survival principle of the fittest. The efficiency criterion is linked to the fitness function via a genetic algorithm, and the ideal PID settings are determined by minimising an ideal that incorporates a balanced mixture of integral absolute error (IAE), integral squared error (ISE), and integral time-

weighted absolute error (ITAE).

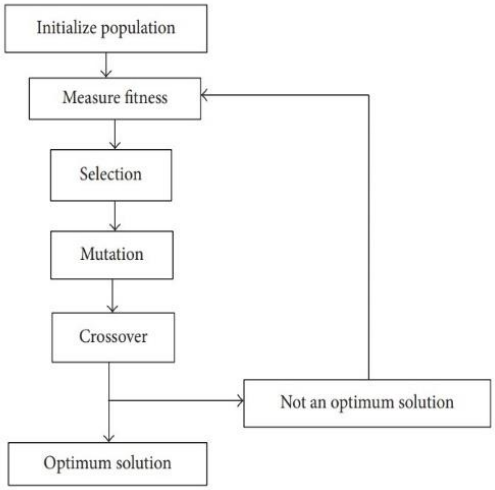

Fig. 6. Genetic algorithm (GA).

Selection: The goal of selection is to identify the individuals who will make up a future population from the present population using the fitness function and the selection technique of choice. Parents are chosen on the basis of their suitability for coupling. Individuals with higher fitness have a better chance of passing on to the next generation. There are many different selection generators, including permutation, steady-state, tournament, and roulette wheel. The roulette wheel has been used in traditional GA.

Crossover: The basic operator of the genetic algorithm is breeding. It is run on the principle of "survival of the fittest". The chromosome of the present population is reproduced or copied in the next generation in each generation, according to the probability of reproduction P_{ri}, which is defined by (7)

$$P_{ri} = \frac{F_i(\theta)}{\sum_{i=1}^{P_i} F_i(\theta)}. \quad (7)$$

where P_i – population size.

The genetic algorithm is directed to the best individuals for breeding. The crossover operation is used to share information between any two chromosomes in the mating pool via a probabilistic decision and to give a mechanism for mixing chromosomes at the splice point.

Mutation: However, in genetic algorithms, the gene pool tends to become increasingly homogeneous as a better gene emerges after several generations, resulting in the convergence of a nonoptimal solution too soon. To avoid this unwanted convergence, the GA includes a probability-based mutation of the third genetic operator. The mutation is a once-in-a-while change in the gene from zero to one or one to zero, with the mutation location chosen at random.

Fitness function: It is a function that has been chosen to quantify how closely related a given element is to other searching elements in a community. Unquestionably, a coherent function is required to show how the desired outcome is reached for each community chromosome that rounds in a distinct region of the decision space.

A genetic algorithm is used to control PID. The following is a summary. GA creates a random population to begin with, which is then implemented with a modest population size to allow the controller to be optimised and converge quickly. The PID parameters Kc, TI, and Td are encoded in binary strings called "chromosomes" to define the initial population. The fitness of each chromosome is determined by translating its binary text into values in the real world that represent the PID parameters. The PID controller receives each set of PID parameters. Individual cost functions such as ISE, IAE, and ITAE, as well as a weighted mixture of these three cost functions, are used to determine the complete system response for each value of the PID parameter and its initial fitness value. Phases 2 and 3 will be repeated untill the process is completed. Generations have acquired the highest level of fitness. The final purpose of GA is to find global PID values (Kc, TI, and Td) with a minimal fitness value that will allow the continuous stirred-tank reactor (CSTR) plant to run through the entire range [29]–[33].

3. *Particle swarm optimization*

Due to their simplicity and functional performance, PI and PID controllers are used to control quantities such as voltage, power, and frequency in most industrial systems (robust and reliable). Several ways have been presented to tune the PI/PID controller settings in the loop of electrical systems. According to the literature, the performance of the system is influenced by both the controller and the objective function used to set the controller gains. The frequency control method has incorporated a wide variety of nature-inspired algorithms and artificial intelligence techniques in recent years. Particle swarm optimisation is a population-based stochastic optimisation method inspired by the social behaviour of birds or schools of fish developed by Drs. Eberhart and Kennedy in 1995 [34]–[37]. The optimisation technique of the PSO algorithm is utilised for the optimum values of K_p and K_i in real-time operation to reduce the transient response, eliminate time overshoot, and obtain low steady-state error owing to load changes. PSO has been shown to generate superior results more quickly and cheaply than previous approaches. It can also be parallelised. In addition, it ignores the gradient of the problem to be solved. Put it another way, PSO does not require a differentiable problem, in contrast to traditional optimisation methodologies. We have not yet discussed the inertia, cognitive, or social coefficients, as you may have noted. The levels of exploration and exploitation are controlled by these coefficients. The ability of particles to target the best solutions found so far is known as exploitation. The ability of particles to evaluate the entire research space is called "exploration". The remainder of the challenge of the paper would be to define the impact of these coefficients to find an appropriate equity position between exploration and exploitation. The assumption of an equilibrium for exploration and exploitation only makes sense if both are quantifiable, and such an equilibrium is neither necessary nor sufficient from an efficacy point of view. However, we shall use these terms in this post for the sake of clarity

$$P_i^t = \lfloor x_0^t, i, x_1^t, i, x_2^t, i, x_3^t, i, \ldots\ldots, x_n^t \rfloor. \quad (8)$$

In a search space, a group of particles have (possible solutions) the global minimum. In this search space, there is only one global minimum. No one of the particles knows the location of the global minimum, but they all have fitness values that must be optimised using the fitness function

$$V_i^t = \lfloor v_0^t, i, v_1^t, i, v_2^t, i, v_3^t, i, \ldots\ldots, v_n^t \rfloor. \quad (9)$$

Each of these particles has a velocity that allows them to update their position over time to discover the global minimum:

$$P_i^{t+1} = P_i^t + V_i^{t+1}, \quad (10)$$

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_{best(i)}^t - P_i^t) + c_2 r_2 (P_{bestglobal}^t - P_i^t). \quad (11)$$

The speed of each particle is stochastically accelerated towards its best position (personal best) and the best solution of the group in all iterations of the search space (global best). Put another way, the velocity of each particle is modified at each iteration in Fig. 7.

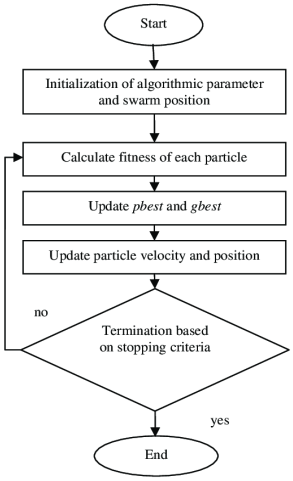

Fig. 7. Particle swarm optimisation.

This velocity is governed by the two best values found so far and is subject to inertia. The first value is the greatest personal solution to date for the particle. The second is the best global answer, which the swarm of particles has discovered thus far. As a result, each particle has its best personal answer, as well as the best global solution stored in its memory [38]–[40].

IV. SIMULATION MODEL

As shown in Fig. 8, the FAST Matlab/Simulink wind turbine simulator model is used for the simulation. As a wind turbine model, to be in real time, three blades with adjustable parameters in the FAST programme, 70 m in diameter, and a 1500 kW wind turbine model were prepared. A PI, GA, and PSO controller for rotor speed control is used. At wind speeds higher than the nominal wind speeds, the rotor speed of the PI, GA, and PSO controllers at the nominal rotor speed produces a pitch angle value to maintain it.

Information on mechanical loads on the wind turbine blades was obtained from the fast wind turbine simulator during simulation [36]. Information about these mechanical loads is passed through a low-pass filter to filter out high-frequency components. There is a 120 ° phase difference between the low-frequency periodic loads on each blade. These loads are added to the collective pitch angle value at a certain ratio, and an individual pitch angle value is produced for each wing. Thus, the amount of periodic mechanical load on the blades of the wind turbine is reduced by changing the blade angle where the load is high in Fig. 9.

V. RESULTS AND DISCUSSION

According to the simulation results, as shown in Fig. 10, the rotor speed of the PSO control showed better stability than that of the PI and GA control, except for some small instantaneous variations. Figures 11–13 show the individual pitch angles PI, GA, and PSO together. The individual pitch angles show an oscillation around each of them. The magnitude of the oscillation is proportional to the coefficient by which the RootMyc moment is multiplied. The optimal adjustment of this coefficient is a separate research topic. Figure 11 shows in more detail the variation of the different tilt angles of each control method.

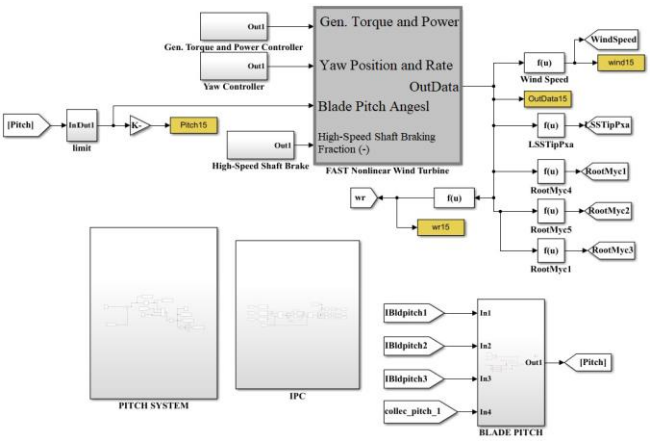

Fig. 8. Simulink model of the whole system.

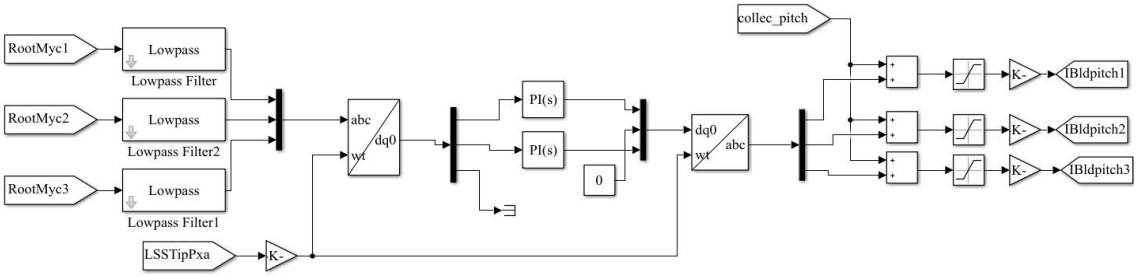

Fig. 9. Individual pitch angle production layer of the wind energy system.

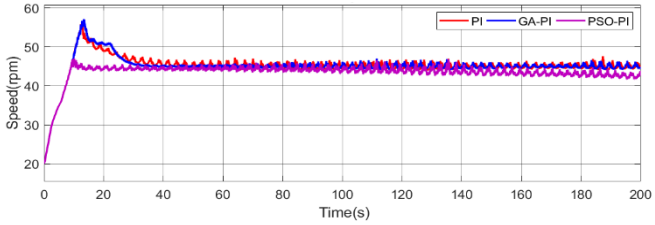

Fig. 10. Rotor speed.

As can be seen in Figs. 11–13, there is a phase difference of 120 ° between the individual pitch angles. This phase difference is due to the phase difference of the blade moments.

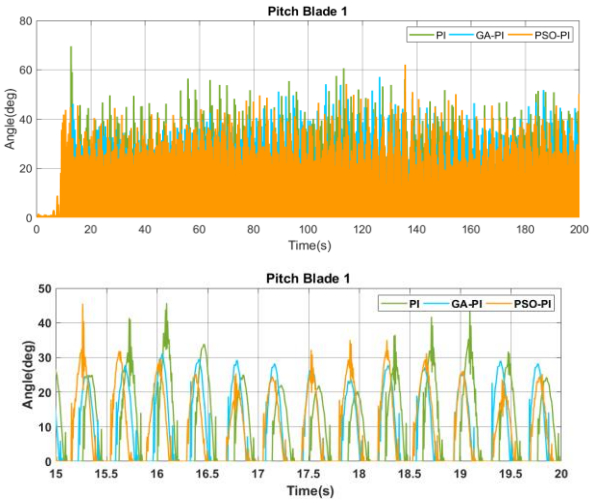

Fig. 11. Pitch blade 1.

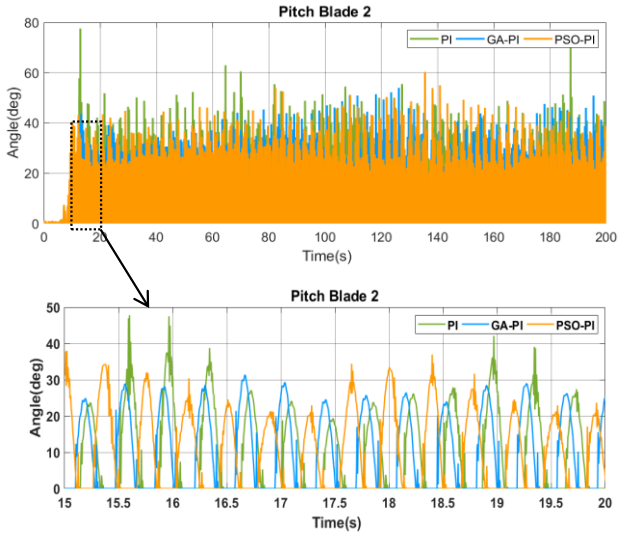

Fig. 12. Pitch blade 2.

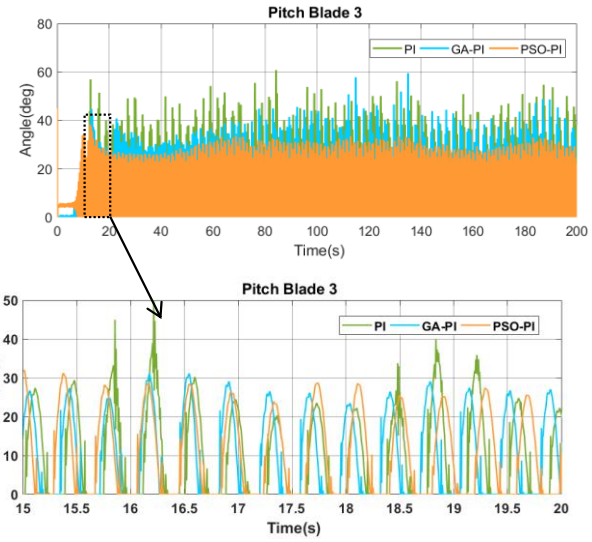

Fig. 13. Pitch blade 3.

Figure 14 shows the applied wind speed. As the nominal wind speed is assumed to be 12 m/s, the average wind speed is chosen above the nominal speed to show the effect of the controller. The wind speed is applied along the x-axis, perpendicular to the turbine, since it is applied for balanced loads.

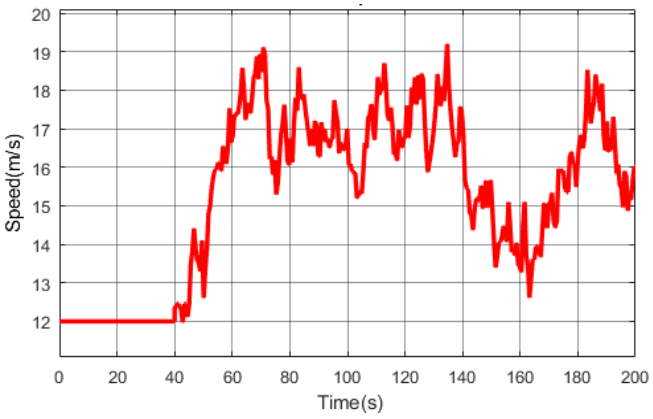

Fig. 14. Wind speed.

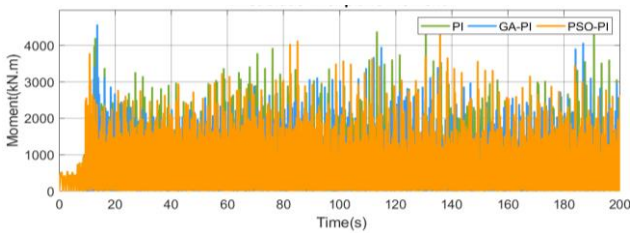

Fig. 15. First blade in-of-plane moment.

TABLE II. MOMENT DESCRIPTIONS.

Moments	Description
RootMxb1	1. Lateral edgewise moment at the wing root of the wing
RootMyb1	1. Pitch-flapwise moment of the wing at the wing root
RootMzb1	1. Pitch moment of the wing at the wing root
RootMxc1	1. Plane moment in plane at the wing root of the wing
RootMyc1	1. Out-of-plane of plane at the wing root of the wing moment
RootMxc2	2. Plane moment of the wing at the wing root
RootMyc2	2. Out-of-plane of plane at the wing root of the wing moment
RootMzc2	2. Pitch moment of the 2nd wing at the wing root
RootMxb2	2. Laterally edgewise at the wing root of the wing moment
RootMyb2	2. Pitch flapwise at the wing root of the wing moment
RootMxc3	3. Plane moment of the wing at the wing root
RootMyc3	3. Out-of-plane of plane at the wing root of the wing moment
RootMzc3	3. Pitch moment of the 2nd wing at the wing root
RootMxb3	3. Laterally edgewise at the wing root of the wing moment

TABLE III. MOMENT RESULTS OF INDIVIDUAL PITCH ANGLES WITH DIFFERENT CONTROLS ARE COMPARED.

Moment Measurement	Individual pitch angle (Average moment)			Variation (%)	
	PI	PSO	GA	PI\PSO	PI\GA
RootMxb1	133.95356	193.876739	135.780444	44.73429368	1.363819089
RootMyb1	-2032.05638	-2259.43321	-2104.71947	11.18949453	3.575840304
RootMzb1	6.69044711	7.30673	6.45872691	9.211385703	3.463448639
RootMxc1	14.1493801	13.6005646	16.2038511	3.878724565	14.51986568
RootMyc1	-1973.34693	-2211.40841	-2047.1397	12.06384352	3.73947285
RootMzc1	6.69044711	7.30673	6.45872691	9.211385703	3.463448639
RootMxc2	16.4892063	14.6235413	11.8308178	11.31446193	28.25113833
RootMyc2	-1976.70951	-2213.04534	-2037.80413	11.95602253	3.090723472
RootMzc2	6.78244133	7.38748383	6.37130644	8.920718549	6.061753776
RootMxb2	135.746938	193.815903	132.804979	42.77736654	2.167237936
RootMyb2	-2036.76938	-2260.81537	-2095.49509	11.00006643	2.883277012
RootMzb2	6.78244133	7.38748383	6.37130644	8.920718549	6.061753776
RootMxc3	12.0307229	15.4466927	14.756165	28.3937198	22.65401746
RootMyc3	-1968.25904	-2215.90503	-2046.55032	12.58198182	3.977692242
RootMzc3	6.59856044	7.4552228	6.44961312	12.98256447	2.257269887
RootMxb3	130.559511	193.660961	134.702561	48.33156159	3.173304414
RootMyb3	-2027.03279	-2262.47148	-2105.20852	11.61494246	3.856658809
RootMzb3	6.59856044	7.4552228	6.44961312	12.98256447	2.257269887

Figure 15 shows the in-of-plane moment of the first wing at the root of the wing. As can be seen in Fig. 15, the individual control of the pitch angle has reduced the mechanical loads. Table I shows the values and reduction rates of the mechanical load between the PI, PSO, and GA controls. Explanations of the moment abbreviations are given in Table II. For example, if we look at the first row of Table III, we can see that the average value of the flanking moment at the root of the 1st wing, denoted by rootmxb1, is 133.95 kNm for the PI controller, 193.87 for the PSO controller, and 135.78 kNm for the GA controller. This means a reduction of 44.73 % in moment load compared to PI and PSO, and 1.36 % compared to PI and GA.

VI. CONCLUSIONS

In this study, the control of pitch angle in wind energy systems is carried out with different control methods. For this purpose, data are obtained from the FAST software used in the determination of mechanical loads and data for the wind turbine installed in Matlab/Simulink. All results are obtained through simulation.

Making the simulation realistic and demonstrating consistency between the results of the various control techniques applied to each specific pitch angle are the goals here. To reduce some of the mechanical demands on the wind turbine, individual adjustment of the pitch angle can be suggested as a solution. The study analysis of the mechanical loads found that they are largely balanced. Winds that blow perpendicular to the turbine blades on the x-axis provide these loads. The pitch angles of the turbine blades are individually regulated along with the mechanical loads on the turbine, and the low-frequency components of these loads are decreased. As a result, it is possible to reduce the fatigue of the mechanical wind turbine parts. According to the study in Table I, the mechanical load for all three blades was reduced by an average of 44 % compared to the PI and PSO methods and by 1 % compared to the PI and GA methods. The importance of the individual pitch control of the system becomes apparent at this speed. As a result, the significance of the study is also made clear. Future research will also use various control methods that will increase the sensitivity. The simulation study will also simulate the features of a real wind turbine.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] P. A. Galvani, F. Sun, and K. Turkoglu, "Aerodynamic modeling of NREL 5-MW wind turbine for nonlinear control system design: A case study based on real-time nonlinear receding horizon control", *Aerospace*, vol. 3, no. 3, p. 27, 2016. DOI: 10.3390/aerospace3030027.

[2] L. Yu, M. Chen, F. Yang, and X. Liu, "A fuzzy control strategy for pumped storage system of the microgrid with wind power", *Recent Advances in Electrical & Electronic Engineering*, vol. 7, no. 1, pp. 37–46, 2014. DOI: 10.2174/2213111607666140516234304.

[3] X. Yin and X. Zhao, "Composite hierarchical pitch angle control for a tidal turbine based on the uncertainty and disturbance estimator", *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 329–339, 2020. DOI: 10.1109/TIE.2019.2896261.

[4] M. Rezkallah, S. Sharma, A. Chandra, and B. Singh, "Implementation and control of small-scale hybrid standalone power generation system employing wind and solar energy", in *Proc. of 2016 IEEE Industry Applications Society Annual Meeting*, 2016, pp. 1–7. DOI: 10.1109/IAS.2016.7731835.

[5] J. G. Stuart, A. D. Wright, and C. P. Butterfield, "Wind turbine control systems: Dynamic model development using system identification and the FAST structural dynamics code", *prepared for 1997 ASME Wind Energy Symposium*, National Renewable Energy Laboratory, 1996. DOI: 10.2514/6.1997-975.

[6] M. Soliman, O. P. Malik, and D. T. Westwick, "Multiple model predictive control for wind turbines with doubly fed induction generators", *IEEE Trans. Sustain. Energy*, vol. 2, no. 3, pp. 215–225, 2011. DOI: 10.1109/TSTE.2011.2153217.

[7] S. Mishra, S. Chatterji, S. L Shimi, and S. Shukla, "Modeling and control of standalone direct-driven PMSG WECS for grid compatibility at varying wind speeds", *International Journal of Engineering Trends and Technology*, vol. 17, no. 10, pp. 495–501, 2014. DOI: 10.14445/22315381/IJETT-V17P297.

[8] S.-H. Song, B.-C. Jeong, H.-I. Lee, J.-J. Kim, J.-H. Oh, and G. Venkataramanan, "Emulation of output characteristics of rotor blades using a hardware-in-loop wind turbine simulator", in *Proc. of Twentieth Annual IEEE Applied Power Electronics Conference and Exposition*, 2005, pp. 1791–1796, vol. 3. DOI: 10.1109/APEC.2005.1453290.

[9] H. Sanchez, T. Escobet, V. Puig, and P. F. Odgaard, "Health-aware model predictive control of wind turbines using fatigue prognosis", *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 1363–1368, 2015. DOI: 10.1016/j.ifacol.2015.09.715.

[10] J. Adhikari, R. Sapkota, and S. K. Panda, "Impact of altitude and power rating on power-to-weight and power-to-cost ratios of the high

altitude wind power generating system", *Renew. Energy*, vol. 115, pp. 16–27, 2018. DOI: 10.1016/j.renene.2017.08.015.
[11] X. Jin, Y. Wang, W. Ju, J. He, and S. Xie, "Investigation into parameter influence of upstream deflector on vertical axis wind turbines output power via three-dimensional CFD simulation", *Renew. Energy*, vol. 115, pp. 41–53, 2018. DOI: 10.1016/j.renene.2017.08.012.
[12] B. Kroposki, R. Lasseter, T. Ise, S. Morozumi, S. Papathanassiou, and N. Hatziargyriou, "Making microgrids work", *IEEE Power Energy Mag.*, vol. 6, no. 3, pp. 40–53, 2008. DOI: 10.1109/MPE.2008.918718.
[13] S. Sarkar, B. Fitzgerald, and B. Basu, "Nonlinear model predictive control to reduce pitch actuation of floating offshore wind turbines", *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 12783–12788, 2020. DOI: 10.1016/j.ifacol.2020.12.1936.
[14] E. Koutroulis and K. Kalaitzakis, "Design of a maximum power tracking system for wind-energy-conversion applications", *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 486–494, 2006. DOI: 10.1109/TIE.2006.870658.
[15] R. Palanisamy and K. Vijayakumar, "Wind-PV hybrid energy source fed three level NPC with Quasi Z source network", *Inter. J. Power Electron. Drive Syst.*, vol. 8, no. 3, pp. 1285–1293, 2017. DOI: 10.11591/ijpeds.v8.i3.pp1285-1293.
[16] A. Beddar, H. Bouzekri, and B. Babess, "Control of grid connected wind energy conversion system using fractional order PI controller: Real time implementation", *Recent Adv. Electr. Electron. Eng.*, vol. 9, no. 2, pp. 132–141, 2016. DOI: 10.2174/2352096509666160512115408.
[17] H. S. K. El-Goharey, W. A. Omran, A. T. M. Taha, and S. M. El-Samanoudy, "Voltage stability investigation of the Egyptian grid with high penetration level of wind energy", *Recent Adv. Commun. Netw. Technol.*, vol. 4, no. 2, pp. 78–89, 2015. DOI: 10.2174/2215081105666160118234913.
[18] F. Dubé, "Conception et comparaison des performances de stratégies de commande PI et DAC appliquées au calage variable d'une eolienne de 10kW", M.S. thesis, University of Quebec, 2014.
[19] S. H. E. Abdel Aleem, A. Y. Abdelaziz, and A. F. Zobaa, "Egyptian grid code of wind farms and power quality", in *Handbook of Distributed Generation*. Springer, Cham, 2017, pp. 227–245. DOI: 10.1007/978-3-319-51343-0_7.
[20] F. Díaz-González, A. Sumper, O. Gomis-Bellmunt, and R. Villafáfila-Robles, "A review of energy storage technologies for wind power applications", *Renew. Sustain. Energy Rev.*, vol. 16, no. 4, pp. 2154–2171, 2012. DOI: 10.1016/J.RSER.2012.01.029.
[21] T. Ackermann and L. Söder, "Wind energy technology and current status: A review", *Renew. Sustain. Energy Rev.*, vol. 4, no. 4, pp. 315–374, 2000. DOI: 10.1016/S1364-0321(00)00004-6.
[22] W. Tong, *Wind Power Generation and Wind Turbine Design*. WIT Press, 2010.
[23] F. Blaabjerg and K. Ma, "Future on power electronics for wind turbine systems", *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 1, no. 3, pp. 139–152, 2013. DOI: 10.1109/JESTPE.2013.2275978.
[24] P. O. Ochieng, A. W. Manyonge, and A. O. Oduor, "Mathematical analysis of tip speed ratio of a wind turbine and its effects on power coefficient", *Int. J. Math. Soft Comput.*, vol. 4, no. 1, pp. 61–66, 2014. DOI: 10.26708/IJMSC.2014.1.4.07.
[25] S. Guntur *et al.*, "FAST v8 verification and validation for a MW-scale wind turbine with aeroelastically tailored blades", in *Proc. of 34th Wind Energy Symposium*, 2016, pp. 1–18. DOI: 10.2514/6.2016-1008.
[26] A. Larsson, "Flicker emission of wind turbines during continuous operation", *IEEE Trans. Energ. Convers.*, vol. 17, no. 1, pp. 114–118, 2002. DOI: 10.1109/60.986447.
[27] A. J. Moshayedi *et al.*, "Simulation and validation of optimized PID controller in AGV (Automated Guided Vehicles) model using PSO and BAS algorithms", *Computational Intelligence and Neuroscience*, vol. 2022, art. ID 7799654, 2022. DOI: 10.1155/2022/7799654.
[28] K. K. Jha, M. N. Anwar, B. S. Shiva, and V. Verma, "A simple closed-loop test based control of boost converter using internal model control and direct synthesis approach", *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 10, no. 5, pp. 5531–5540, 2022. DOI: 10.1109/JESTPE.2022.3157515.
[29] A. Jayachitra and R. Vinodha, "Genetic algorithm based PID controller tuning approach for continuous stirred tank reactor", *Advances in Artificial Intelligence*, vol. 2014, art. ID 791230, 2014. DOI: 10.1155/2014/791230.
[30] A. Jooyayeshendi and A. Akkasi, "Genetic algorithm for task scheduling in heterogeneous distributed computing system", *Inter. J. Sci. Eng. Res.*, vol. 6, pp. 1338–1345, 2015.
[31] F. Xhafa, J. Carretero, and A. Abraham, "Genetic algorithm based schedulers for grid computing systems", *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 5, pp. 1053–1071, 2007.
[32] Z. Civelek, E. Çam, M. Lüy, and H. Mamur, "Proportional–integral–derivative parameter optimisation of blade pitch controller in wind turbines by a new intelligent genetic algorithm", *IET Renewable Power Generation*, vol. 10, no. 8, pp. 1220–1228, 2016. DOI: 10.1049/iet-rpg.2016.0029.
[33] A. Thakkar and K. Chaudhari, "Information fusion-based genetic algorithm with long short-term memory for stock price and trend prediction", *Applied Soft Computing*, vol. 128, art. 109428, 2022. DOI: 10.1016/j.asoc.2022.109428.
[34] S. M. G. Kumar, D. Jayaraj, and A. R. Kishan, "PSO based tuning of a PID controller for a high performance drilling machine", *International Journal of Computer Applications*, vol. 1, no. 19, art. 3, 2010. DOI: 10.5120/410-607.
[35] J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proc. of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942–1948, vol. 4. DOI: 10.1109/ICNN.1995.488968.
[36] A. J. Carlisle, "Applying the particle swarm optimizer to non-stationary environments", Ph.D. dissertation, Auburn University, Auburn, AL, USA, 2002.
[37] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems", *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, 2008. DOI: 10.1109/TEVC.2007.896686.
[38] G. Sripriyanka and A. Mahendran, "Bio-inspired computing techniques for data security challenges and controls", *SN Computer Science*, vol. 3, art. no. 427, 2022. DOI: 10.1007/s42979-022-01292-w.
[39] N. S. Sukanya and P. R. J. Thangaiah, "Enhanced differential evolution and particle swarm optimization approaches for discovering high utility itemsets", *International Journal of Computational Intelligence and Applications*, vol. 22, no. 1, 2023. 10.1142/S1469026823410055.
[40] E. A. Bossanyi, "Further load reductions with individual pitch control", *Wind Energy*, vol. 8, no. 4, pp. 481–485, 2005. DOI: 10.1002/we.166.