Accepted Manuscript

Load mitigation of a class of 5-MW wind turbine with RBF neural network based fractional-order PID controller

A.H. Asgharnia, A. Jamali, R. Shahnazi, A. Maheri

PII: S0019-0578(19)30296-4
DOI: https://doi.org/10.1016/j.isatra.2019.07.006
Reference: ISATRA 3259

To appear in: *ISA Transactions*

Received date : 25 October 2018
Revised date : 1 July 2019
Accepted date : 2 July 2019

*Title page showing Author Details

Load Mitigation of a Class of 5-MW Wind Turbine with RBF Neural Network based Fractional-Order PID Controller

A.H. Asgharnia

Faculty of Mechanical Engineering, University of Guilan, Rasht, Iran. P.O. Box: 3756

E-mail: asgharnia4@yahoo.com; Tel: +981333690274

A. Jamali

Faculty of Mechanical Engineering, University of Guilan, Rasht, Iran. P.O. Box: 3756

E-mail: ali.jamali@guilan.ac.ir; Tel: +981333690274

R. Shahnazi

Department of Electrical Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran

E-mail: shahnazi@guilan.ac.ir; Tel: +981333690274

A. Maheri

School of Engineering, University of Aberdeen, Aberdeen AB24 3UE, UK

E-mail: alireza.maheri@abdn.ac.uk, Tel: +447956661417

*Highlights (for review)

- A gain-scheduling fractional-order PID pitch controller is proposed
- The controller is designed to mitigate the mechanical loads
- A database controller parameters are evaluated via chaotic differential evolution
- The proposed controller method has shown to have superior performance
- The results are validated via FAST simulator

Load Mitigation of a Class of 5-MW Wind Turbine with RBF Neural Network based Fractional-Order PID Controller

Abstract- In variable-pitch wind turbines, pitch angle control is implemented to regulate the rotor speed and power production. However, mechanical loads of the wind turbines are affected by the pitch angle adjustment. To improve the performance and at the same time alleviate the mechanical loads, a gain-scheduling fractional-order PID (FOPID), where a trained RBF neural network chooses its parameters is proposed. The database, which the RBF neural network is trained based on, is created via optimization of a FOPID in several wind speeds with chaotic differential evolution (CDE) algorithm. The simulation results are compared to an RBF based PID controller that is designed via the same method, a conventional gain-scheduling baseline PI controller developed by NREL, an optimal RBF based PI controller, and a FOPI controller. The simulations indicate that the RBF based FOPID improves the control performance of the benchmark wind turbine in comparison to the other controllers, while the applied loads to the structure are mitigated. To validate the performance and robustness, all controllers are implemented on FAST wind turbine simulator. The superiority of the proposed FOPID controller is depicted in comparison to the other controllers.

Keywords: Gain-scheduling fractional-order PID, Wind turbine pitch control, Chaotic differential evolution, RBF neural network, FAST

1 Introduction

In past decades, more attention has been paid to developing and economizing renewable sources of energy. Among them, wind energy has received noticeable attention. Installed wind energy conversion systems (WECSs) have increased by 40% in the 2000s [1]. Until now, many countries have installed WECSs, and the capacity of installed WECSs is going to pass 700 GWs by 2020 [2]. It should be noted that developing control algorithms has played an essential role in this rise [3].

It is conventional to use more than one strategy to operate a wind turbine in different wind speeds, which is based on *rated-speed*. While in speeds below rated-speed the goal is to keep the captured power as high as possible via torque control, in above rated-speed the point is to regulate the rotor speed via pitch angle and torque control, simultaneously [4].

Research is abundant in the performance of controllers of each kind in wind turbine pitch angle and torque adjustment. For instance, in [5], by combining a radial basis function (RBF) neural network and PI controller, a gain-scheduling PI controller is developed. Therefore, by measuring the wind speed, the RBF neural network selects suitable gains for the PI controller. The proposed method has shown better performance in regulating rotor speed and power in a stochastic wind condition over a constant-gain PI controller. In [6], two controllers are designed for pitch actuator based on MLP and RBF neural networks. In the article, RBF had slightly better performance in rotor speed regulation. The performance of a nonlinear PI (N-PI) controller is studied in [7], in which by designing an extended-order state and perturbation observer to estimate the nonlinearities, an external signal is added to the output of a PI controller. The results showed the effectiveness of N-PI in decreasing the RMS (Root Mean Square) of error and mechanical loads in comparison to a gain-scheduled PI. Meanwhile, the results are also validated via the FAST simulator. In [8], to overcome the effect of the unknown delays caused by hydraulic pressure driven units a PI controller is optimized for an ideal system, while a delay estimator is designed to estimate the perturbation caused by the delay. Using this estimation as a compensation signal the effect of the delay in the output is removed. The technique was tested on wind turbines with different rated powers, and it is observed that the performance of a 4.8-MW wind turbine has been improved.

The quality of adjusting the controlling parameters of a wind turbine has significant effects on the mechanical loads of the drivetrain, tower, and blades [7]. Pitch regulation, changes the direction of the airfoil, so as the vector of applied forces on the blades. These changes and wind speed fluctuations, cause cyclic motion and vibration in the blades and tower. Hence, the control methods play an essential role in limiting the loads and fatigue damages and as a result lowering the maintenance cost and increasing the efficiency of wind turbines. These are the motivations to search for suitable approaches in operation. Although one of the manners is to redesign the blades with respect to fatigue reduction [9], or implementing new sensors and mechanical equipment [10], a fast and viable way in order to response these demands are changing the control algorithms and software, in which, the requirement for new sensors and design would be relaxed.

Therefore, more research is done recently to decrease the mechanical loads. For instance, in [11], several control algorithms are presented to alleviate loads of a wind turbine. These methods consist of installing new sensors to measure the loads, using individual pitching control (IPC), providing a joint control between power production and the loads, and using the torque control to alleviate the torsional resonance. IPC is a technique that every blade rotates along its longitude axis separately. An experimental study on IPC is conducted in [12] to reduce the loads. The controller is designed based on the linear state-space model, and the gains are calculated via linear quadratic regulator (LQR) method. The controller demonstrates better performance in lowering the loads while maintaining the error and pitch actuator usage. However, in these kinds of model-based methods, a complete model of the system is needed. A combination of IPC and fuzzy controllers is studied to reduce mechanical loads [13]. To do this, a fuzzy controller is designed to control the rotor speed by adjusting the pitch angle and generator reference torque, while the other two fuzzy controllers are responsible for controlling the mechanical loads (blade moments) by adding an extra signal to the output of the first controller. The control performance shows a reduction in fatigue loads. A robust H_∞ method is examined in [10] for tower and drivetrain load mitigation in a 5-MW wind turbine, where two H_∞ controllers are designed at above the rated speed; one controller is for adjusting the pitch angle, and another is designed to tune the generator torque. The inputs of the controllers were generator speed, tower, and blade tip accelerations. The method has superiority in load reduction in comparison to a baseline controller. In [14], it is shown that how optimization of a pitch and torque controller can affect the loads. In the method, a hybrid cost function is defined, which includes the fatigue and ultimate loads of blades, tower and drivetrain and the rate of pitch angle. Then the variation of cost in different proportional and integral gains is studied. A reduction of 2% was achieved in load effect in particular wind speed. In [15], a comparison is made between SISO and MIMO active flow control in a wind turbine. It is shown that in a wind turbine equipped with active flow control, a MIMO controller can be decomposed into simpler SISO controllers, which is highly efficient in load reduction.

In the past years, the fractional order controllers have received many interests. Fractional order controllers have more parameters to set so that the controller designer can apply more consideration to account. A motivation to study this kind of controller is its particular structure: If their extra parameters, which are their orders, are set to 1, they act as a simple PID controller. On the other hand, albeit their nonlinear figure ($PI^{\lambda}D^{\mu}$), they are usually approximated via linear transfer functions that are similar to high order linear controllers. In several cases, fractional-order controllers have shown a better control performance than their integer order counterparts: In [16], the performance of an automatic voltage regulator is investigated under control of an optimized FOPID. In [17], a multi-objective optimization is accomplished to control a hydraulic turbine. Besides, in [18], a multi-objective design process is suggested to design a FOPID and PID for plants with parametric uncertainty. In [19], a fractional order PI controller is investigated for a 4.8 MW wind turbine, while its gains are constant during the operation. In [20], a gain-scheduling PID and a gain/order-scheduling FOPID are designed via optimization. The simulation results show significant superiority of schedule-gain/order FOPID in decreasing control signal fluctuations.

In this paper, to mitigate the mechanical loads in a wind turbine and maintain its performance, simultaneously, a new method, which is a combination of FOPID and RBF neural network, is proposed. In the process, the wind turbine equipped with a simple FOPID controller undergoes several wind profiles with fixed average speed. Then, employing chaotic differential evolution (CDE), the optimal gains and orders are found. The primary goal of this design is to alleviate the tower and blade moments, which are critical in the wind turbine lifespan. With the optimal dataset, an RBF neural network is trained to choose the best parameters and put them into the controller. To study the effectiveness of FOPID, an RBF neural network based PID is also designed within the same framework. Then several fluctuated wind speeds are applied to the wind turbine model, and the results are compared with a conventional gain-scheduling PI controller (NREL baseline PI controller) [21], the RBF PI controller [5], and the FOPI controller [19]. It is known that the validation of a proposed controller is of utmost importance. To this end, to validate the simulation results, all controllers are applied to the FAST (Fatigue, Aero-elastic, Structure, Turbulence) as a detailed wind turbine simulator.

The motivation of this paper is twofold: 1) Proposing controllers to investigate the load mitigation of a wind turbine and comparing it via a conventional controller in the industry. 2) Since the load mitigation and performance in wind turbines conflict with each other, another motivation is that the controllers should present satisfactory performance. It should be noted that, although a controller with more coefficient may demonstrate a better achievement in some control objectives, its effects on different aspects should be studied. The contributions of this paper, to accomplish those motivations, are as follows:

1) Proposing a cost function to decrease the mechanical loads.
2) Considering the performance of a conventional gain-scheduling PI controller (NREL baseline PI controller) as a constraint.
3) Proposing an RBF neural network that can predict the gains of the PID/FOPID controllers without any demand to measure the wind speed.
4) Validation the control performance of the proposed controllers via a standard wind turbine simulator (FAST).
5) In the proposed methods, unlike IPC related papers, there is no demand for new mechanisms [11-13].
6) The need for sensors to measure the wind speed or the tower/blades acceleration is relaxed [5, 6, 10].

This paper is organized as follows: Section 2 is a brief description of the wind turbine model. In Section 3, the baseline controller and the proposed methods are presented. Section 4 demonstrates the process of deriving the parameters, test scenarios, and validation. Finally, Section 5 concludes the paper by discussing the main advantages of the proposed method.

2 Wind Turbine Dynamic Model

A wind turbine (WT) dynamics can be divided into several parts: Aerodynamics, drivetrain, generator, pitching system, and flexible tower. The wind turbine that is presented in this study as the benchmark is a land-based 5 MW class horizontal wind turbine, which is proposed by NREL [21].

2.1 Aerodynamics

The captured energy crucially depends on blade shape. However, it is also affected by wind speed and pitch angle. The captured power is calculated as:

$$P_a = \frac{1}{2}\,\rho\pi R^2 C_P(\lambda,\beta)v^3 \qquad (1)$$

where P_a is the captured aerodynamics power, ρ is the air density, and R is the radius of blades plus hub radius. C_P is power coefficient and v is the wind speed. β is the pitch angle and $\lambda = {}^{R\omega_r}/_{v}$ is called the tip speed ratio (TSR).

The captured torque from wind is calculated as follows:

$$P_a = T_a\omega_r \qquad (2)$$

where T_a is the aerodynamic torque.

C_P is an experimental coefficient, which is nonlinear and dependent on blade shapes, TSR, and pitch angle. Here the coefficient is adopted from a look-up table of NREL 5-MW wind turbine [21].

2.2 Drivetrain

The drivetrain is a complex component that transmits the captured power to the generator. In a large-scale wind turbine, the drivetrain can have severe effects on the performance, because of its flexibility. It is more common to simplify the model to separated masses. In [22], several separated mass models in the transient period, such as 2-mass, 3-mass, and 6-mass are compared. It is studied that 2-mass model is accurate and yet simple enough to be chosen for simulation and controller design. A two-mass simplified model for drivetrain is shown in Figure 1.

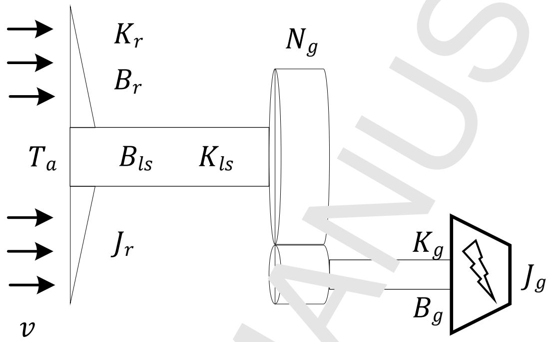


Figure 1 Two-mass simplified drivetrain model

The drivetrain equations are derived as follows

$$J_r\dot{\omega}_r = T_a - T_{ls} - B_r\omega_r \tag{3}$$

where J_r is the inertia of blades, hub and low-speed shaft. T_{ls} is the low-speed shaft torque and B_r is the rotor damping coefficient. T_{ls} can be calculated as follows

$$T_{ls} = K_{ls}(\psi_r - \psi_{ls}) + B_{ls}(\omega_r - \omega_{ls}) \tag{4}$$

where K_{ls} is low-speed shaft stiffness and B_{ls} is low-speed shaft damping. ω_{ls} is the speed of low-speed shaft while ψ_r and ψ_{ls} are the rotor and low-speed shaft angular deviation, respectively.

The gearbox transmission ratio is defined as:

$$N = \frac{T_{ls}}{T_{hs}} \tag{5}$$

where N is gearbox ratio and T_{hs} is the high-speed shaft torque.

In the generator side, the following equations exist:

$$J_g\dot{\omega}_g = T_{hs} - T_g - B_g\omega_g \tag{6}$$

In (6), J_g is the generator inertia, T_g is the generator torque and B_g is the generator damping.

According to (3)-(6), the drivetrain differential equations are derived as follows

$$\begin{bmatrix}\dot{\omega}_r\\ \dot{\omega}_g\\ \dot{T}_{ls}\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}\begin{bmatrix}\omega_r\\ \omega_g\\ T_{ls}\end{bmatrix} + \begin{bmatrix}b_{11}\\ b_{21}\\ b_{31}\end{bmatrix}T_a + \begin{bmatrix}c_{11}\\ c_{21}\\ c_{31}\end{bmatrix}T_g \tag{7}$$

where

$$a_{11} = -\frac{B_r}{J_r} \quad a_{12} = 0 \quad a_{13} = -\frac{1}{J_r} \quad b_{11} = \frac{1}{J_r} \quad c_{11} = 0$$

$$a_{21} = 0 \quad a_{22} = -\frac{B_g}{J_g} \quad a_{23} = \frac{1}{N_g J_g} \quad b_{21} = 0 \quad c_{21} = -\frac{1}{J_g}$$

$$a_{31} = K_{ls} - \frac{B_{ls}B_r}{J_r} \quad a_{32} = \frac{1}{N_g}\left(\frac{B_{ls}B_g}{J_g} - K_{ls}\right) \quad a_{33} = -B_{ls}\left(\frac{J_r + N_g^2 J_g}{N_g^2 J_r J_g}\right) \quad b_{31} = \frac{B_{ls}}{J_r} \quad c_{31} = \frac{B_{ls}}{N_g J_g}$$

2.3 Generator

The generator is supposed to convert the kinetic energy of the wind to electrical power. In this paper, a simple first order generator is chosen, and its differential equation is as follows

$$\dot{T}_g = \frac{1}{\tau_g}(T_{ref} - T_g) \tag{8}$$

$$P_g = \eta_g T_g \omega_g \tag{9}$$

where τ_g is the generator time constant, P_g is the generated power and η_g is generator efficiency. It should be noted that there is also a limitation in both torque and torque rate in generators dynamics. T_g is limited between 0 to 47,402.91 N.m whereas its rate is limited between -15 to 15 KN.m/s [21].

Since the main contribution of this paper is to study the mechanical loads and pitch control, the turbine is considered to be an off-grid; thus, a first order generator is reasonable [21]. However, a more advanced model for the generator is needed when the turbine is connected to the grid. Usually, the doubly-fed induction generator, along with a back-to-back converter, is utilized [23]. To control the connection of WT to the grid, one of the effective methods is to use a back-to-back converter to control the frequency. Besides, since doubly-fed induction generators consume reactive power, the back-to-back converter can also be used as a capacitor bank to compensate power factor [24].

2.4 Pitch actuator

Pitch actuator rotates the blades around their longitude axis. In this research, a simple first order actuator is implemented. The differential equation is as follows

$$\dot{\beta} = \frac{1}{\tau_\beta}(\beta_{ref} - \beta) \tag{10}$$

In (10), β_{ref} is the reference pitch angle, generated by the controller and τ_β is the time constant of the actuator. In a pitch actuator, the limitations are playing a crucial role.

β is usually limited between 0º and 90º while the rate limitation is considered to be between -8 to +8 º/s.

2.5 Tower

Rising wind through wind turbine caused vibration in the tower. In tall wind turbines, tower vibration caused an additional fluctuation in wind speed. In this paper, the tower is approximated via a mass-spring-damper system. The differential equation of the tower can be derived as follows:

$$\ddot{z} = \frac{1}{m_{tow}}\left(F_{tow} - K_{tow}z - B_{tow}\dot{z}\right) \tag{11}$$

where z is the displacement of the tower top. K_{tow} and B_{tow} are the tower stiffness and damping coefficient, respectively. F_{tow} is the applied force to the tower and has a nonlinear relation with wind speed and pitch angle [21].

Although the effect of the flexible tower is usually neglected in many papers, in this paper, it is considered by its impact on wind speed fluctuations. In other words, the tower tip speed is added to the wind speed. It is noticeable that the blade motion, like tower motion, could also affect the WT performance by changing the power curve. However, the effect of blade motion on power production and the interaction between the drivetrain, tower, and blade is neglected in the two-mass model.

Table 1 exhibits some of the leading wind turbine parameters.

Table 1 Wind turbine parameters [21]

Parameter	Value
Power capacity	5 MW
Cut-in, Cut-out and rated speed	3 m/s, 25 m/s and 11.4 m/s
Rotor radius	63 m
Tower height	87.6 m
Rated generator angular speed	122.9 rad/s
Rated generator torque	43093.55 N.m
Gearbox ratio	97:1
Maximum power coefficient	0.482

3 Control Designs

3.1 Baseline controller

In this part, the baseline PI controller, which is proposed by [25] and designed for a 5-MW wind turbine by NREL [21] is described. Baseline PI controller is a gain-scheduling PI controller and developed based on the simple single degree of freedom wind turbine model. Based on the free body of a simple drivetrain, the rotor equation of motion can be written as follows

$$T_a - NT_g = \left(J_r + N_g^2 J_g\right)\frac{d}{dt}(\omega_r + \Delta\omega_r) = J_d\Delta\dot{\omega}_r \tag{12}$$

where J_d is the drivetrain inertia.

Since the Generator torque changes are ignorable in the region above rated-speed, it can be calculated by

$$T_g(N_g\omega_r) = \frac{P_0}{N_g\omega_r} \tag{13}$$

where P_0 is the rated mechanical power. On the other hand, by assuming that the change in the captured aerodynamic force is ignorable:

$$T_a(\beta) = \frac{P(\beta, \omega_{r-rated})}{\omega_{r-rated}} \tag{14}$$

where P is the mechanical power and $\omega_{r-rated}$ is the nominal rotor speed.

By using first-order Taylor expansion of (13) and (14) around ω_r and β, respectively, two equations can be written as:

$$T_g \approx \frac{P_0}{N_g\omega_{r-rated}} - \frac{P_0}{N_g\omega_{r-rated}^2}\Delta\omega_r \tag{15}$$

$$T_a \approx \frac{P_0}{\omega_{r-rated}} + \frac{1}{\omega_{r-rated}}\left(\frac{\partial P}{\partial\beta}\right)\Delta\beta \tag{16}$$

where $\Delta\beta$ is a small deviation of blade pitch angle about its operational point. A PID controller scheme, which its input is deviation of rotor speed and its output is defined as the deviation of blade pitch angle can be written as:

$$\Delta\beta = K_P N_g \Delta\omega_r + K_I\int_0^t N_g\Delta\omega_r dt + K_D N_g \Delta\dot{\omega}_r \tag{17}$$

where K_P, K_I and K_D are proportional, integral, and derivative gains, respectively.

Now by assuming $\Delta\beta = \varphi$, and combining (12) and (15)-(17), the equation of motion for rotor-speed will be calculated as follows

$$\left[J_d + \frac{1}{\omega_{r-rated}}\left(-\frac{\partial P}{\partial\beta}\right)N_g K_D\right]\ddot{\varphi} + \left[\frac{1}{\omega_{r-rated}}\left(-\frac{\partial P}{\partial\beta}\right)N_g K_P - \frac{P_0}{\omega_{r-rated}^2}\right]\dot{\varphi} + \left[\frac{1}{\omega_{r-rated}}\left(-\frac{\partial P}{\partial\beta}\right)N_g K_I\right]\varphi = 0 \tag{18}$$

Eq. (18) bears a striking resemblance to an ordinary second-order system with following the differential equation

$$M_{eq}\ddot{\varphi} + C_{eq}\dot{\varphi} + K_{eq}\varphi = 0 \tag{19}$$

In (19), natural frequency and damping ratio can be defined as

$$\omega_\varphi = \sqrt{\frac{K_{eq}}{M_{eq}}} \tag{20}$$

$$\zeta_{\varphi} = \frac{C_{eq}}{2\sqrt{K_{eq}M_{eq}}} \tag{21}$$

In [25] it is suggested to neglect the K_D and assume the natural frequency to be 0.6 rad/s and the damping ratio to be 0.6 - 0.7. Therefore, the gains can be calculated with the following equations

$$K_P = \frac{2\, J_d\, \omega_{r-rated}\, \zeta_{\varphi}\, \omega_{\varphi}}{N_g\left(-\frac{\partial P}{\partial \beta}\right)} \tag{22}$$

$$K_I = \frac{J_d\, \omega_{r-rated}\,\ \omega_{\varphi}^2}{N_g\left(-\frac{\partial P}{\partial \beta}\right)} \tag{23}$$

In the above equations, $-\frac{\partial P}{\partial \beta}$ is the blade pitch sensitivity and is dependent on the wind speed, pitch angle, and rotor speed. In [21], the blade pitch sensitivity curve is driven for wind speed. With blade pitch sensitivity, both proportional and integral gains can be calculated. Figure 2 shows the gains for operation in the region above rated-speed.

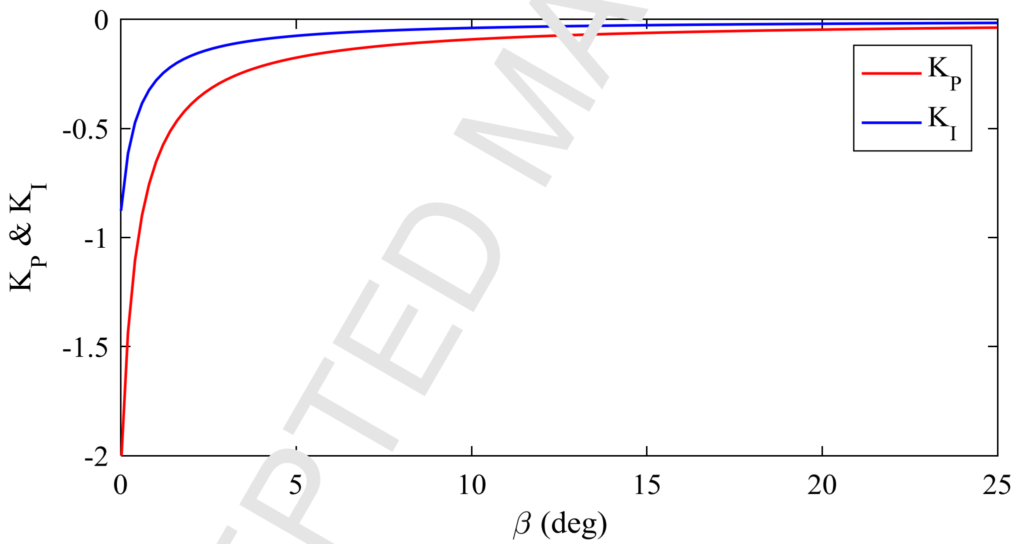


Figure 2 K_P and K_I in the baseline PI controller that is designed for a 5-MW wind turbine [21]

In this method, the gains are chosen based on the pitch angle. Thus, the speed measurement is needed. However, it is a cost-effective suggestion since wind speed measurement is not an easy or accurate task [26]. The anemometer that is usually installed on the wind turbine can only measure the wind speed in the installed point, which does not give proper information about the other parts of the wind turbine.

Remark 1: As it is shown in Figure 2, K_P and K_I are negative parameters. As it is indicated in [21], the relationship between control signal (which is pitch angle) and controller input (which is the error of generator speed) is inverse. On the other hand, the existence of the torque controller adds negative damping to the system. Therefore, for the stability of the system, it is needed to use negative gains.

3.2 Proposed controller

The proposed controller is a gain-scheduling fractional-order PID, which uses the subtraction of generator and nominal speeds as the input and a reference pitch angle as the output. Although FOPID is used in this paper, by this method, any controller with adjustable gains or parameters can be designed. Eq. (24) shows a fractional-order PID in the time domain.

$$\beta_{ref} = K_P e(t) + K_I \int_0^t e(\tau) d\tau^\lambda + K_D \cdot \frac{d^\mu e(t)}{dt^\mu} \tag{24}$$

where μ and λ are two fractional numbers.

Remark 2: Fractional-order controllers are usually approximated via specific expansions, among them, Oustaloup approximation, which recently received many attentions, is slightly simpler to be implemented by hardware [27, 28]. In this paper, due to its effectiveness, the Oustaloup approximation is used. To perform a fractional controller, many tools can be utilized. Although electrochemical systems [29] and electronic circuits [16] can be used, microprocessors and PLCs are the most viable and practical methods.

To choose the optimal parameters, they are first derived by solving a suitable optimization problem. This procedure gives a set of optimal parameters for different wind speeds. This optimal set is used to train an RBF neural network. Thus, the trained neural network can select the proper parameters in each wind speed. However, due to reasons mentioned in Subsection 3.1, the wind speed should not be measured directly. Thus, in our method, the wind speed is estimated by using measurable quantities of the wind turbine. In the following subsections, the technique is explained.

3.2.1 Gains Calculation

To calculate the gains of (24), different wind speeds are considered. Then using an optimization algorithm, a suitable cost function will be minimized, and thus a set of optimal parameters for (24) is derived for each wind speed. With this method, an optimal dataset for gains and orders will be found.

To optimize the controller, the following cost function is considered.

$$Cost = \int_0^{T_{max}} |\dot{u}(t)| \cdot dt \tag{25}$$

where T_{max} is the maximum simulation time and $u(t)$ is the control signal (i.e., the pitch angle reference) at the time t.

Minimizing (25), leads to minimization of the surface below $|\dot{u}|$ over time. There are many reasons for choosing (25) as the primary cost function. $\dot{u}$ is highly related to the rate of pitch angle, which means the rate of force vector changes on the blades. Thus, it is highly correlated with the blades and the tower mechanical loads. One other suggestion instead of (25) is the integral absolute error (IAE) of the rotor speed [5]. However, making the error as small as possible may not generally be a good choice concerning load reduction. Besides, reducing cost function (25) will lower the risk of wind-up and saturation in pitch angle actuators, which because of the minor time constant is probable. It is noticeable that although utilizing (25) can mitigate the loads; it may jeopardize the performance, i.e., generator speed error. Thus, a constraint is needed to determine suitable performance. In this paper, the constraint is defined as the maximum generator speed error of a wind turbine with a PI controller, which its gains are equal to the baseline in each wind speed. Eq. (26) introduces the constraint

$$\int_0^{T_{max}} |e_{PS}^v(t)| \cdot dt \leq \int_0^{T_{max}} |e_{PI}^v(t)| \cdot dt \tag{26}$$

where $e_{PS}^v(t)$ and $e_{PI}^v(t)$, are the error at wind speed v for the proposed controller and PI controller with the baseline gains, respectively. It should be noted that the constraint makes a suitable background in comparing the controllers in the sequel. Considering the above discussions, the following optimization problem can be defined

$$\begin{gathered} \min_{\text{Controller Parameters}} \int_0^{T_{max}} |u(t)| \cdot dt \\ s.t. \int_0^{T_{max}} |e_{PS}^v(t)| \cdot dt \leq \int_0^{T_{max}} |e_{PI}^v(t)| \cdot dt \end{gathered} \tag{27}$$

Selecting the RMS or variance of the signals may be another choice for the cost function. However, it is observed that it does not necessarily minimize the signal frequency; although the RMS or variance is decreased, there might be more cycles. Thus, the derivative of the pitch actuator is not necessarily decreased, and in an uncertain situation, it leads to more loads on the structure.

Now, a gain-scheduling mechanism should be implemented, so that in every wind speed, suitable gains will be assigned to the controller. This mechanism is discussed in the following.

It is essential to consider the difference between gain-scheduling and order-scheduling problems. In (24), the control signal is linear concerning the parameters K_P, K_I, and K_D. However, changing the fractional orders in (24) needs recalculating of Oustaloup approximation, which for each time step, new operators should be calculated. Although Oustaloup approximation can approximate fractional operators, it is not accurate for the first time steps. Thus, changing the order of the fractional operator will cause the controller to give inaccurate results. Figure 3 shows this effect. The figure demonstrates a Sine wave, its full derivative, and its half derivative. As it is shown, the half-derivative behavior in the first few moments is different: In the first half cycle, the amplitude is less than the steady state. However, after a few time steps, the half derivative of Sine is reached to its steady state.

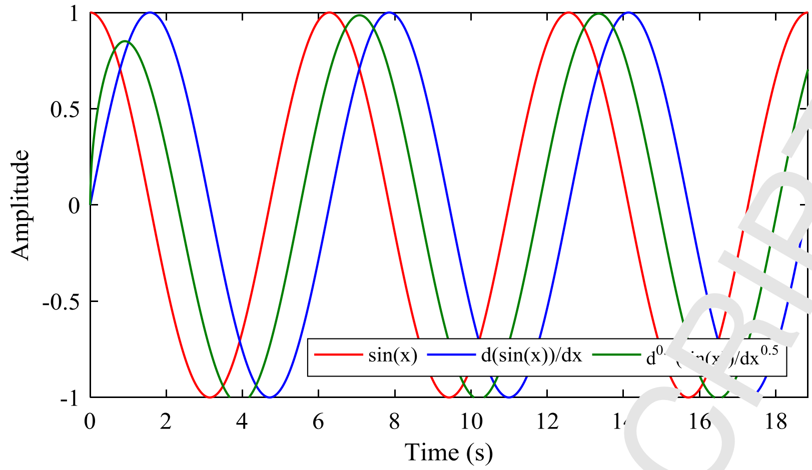


Figure 3 Sin(x), its full derivative, and its half derivative

To solve the problem above, we will assume that the orders of FOPID do not change during operation, and they are equal to the average of optimized orders of the optimization results. Now by considering the orders of (24) to be constant values, another optimization is done to recalculate the three gains of FOPID.

3.2.2 Wind speed estimation

The Newton-Raphson method [7] and artificial neural network [30] have been used in wind speed prediction. Although the estimation tools may be different, the principle of all is the same and based on extracted aerodynamics power. In fact by measuring the P_a in any time and considering (1), the wind speed v can be estimated.

P_a can be calculated via (1) for different values of β, ω_r, and v, to provide a database for the relation between the variables and actual wind speed. It should be noted that it is impossible to measure the captured power (P_a). Instead, the generator power is measured and divided into generator and drivetrain efficiency. The generator efficiency is 94.4%, and the drivetrain is considered to be frictionless [21]. In addition, since the drivetrain model is deemed to be unknown, the ω_r is calculated by dividing the ω_g to the gearbox ratio.

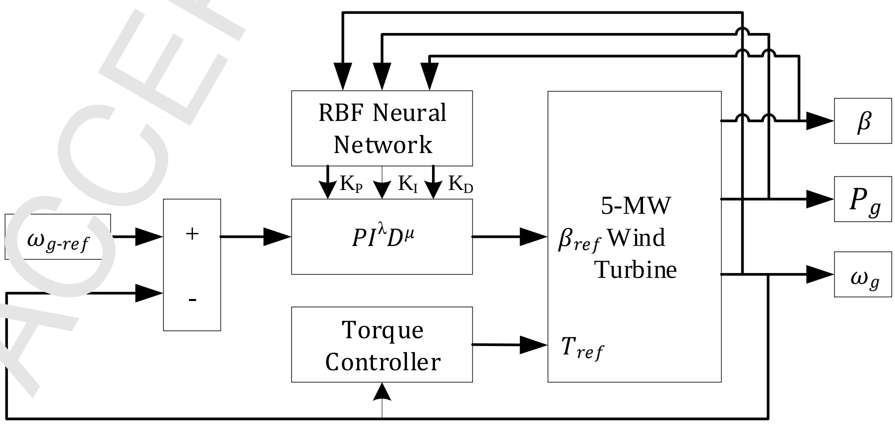


Figure 4 Proposed controller structure

Then, a prediction method can be evaluated that its input vector is β, ω_g and P_g and its output vector is the wind speed (v). However, since the goal is to set the gains in each situation, instead of v, we consider estimating the gains vector in each wind speed, directly. Regarding the discussions in Subsections 3.2.1 and 3.2.2, the structure of the proposed method can be depicted in Figure 4.

4 Simulation

In this section, the proposed controller in section 3 will be designed for the model in section 2, and then test scenarios will be studied. In the sequel, the performance of the proposed controller is compared with the gain-scheduling PID controller designed using the proposed method, NREL baseline PI controller described in Subsection 3.1, RBF PI controller proposed in [5], and a FOPI controller [19], which is tuned based on [31, 32]. For the subsequent discussions, these controllers are respectively denoted as proposed FOPID, proposed PID, baseline PI, RBF PI, and FOPI. It should be noted that all controllers are designed based on the two-mass model and are validated via the FAST simulator.

4.1 Tools

4.1.1 Chaotic differential evolution

Differential evolution (DE) is one of the oldest; however, the strongest optimization algorithms. In this paper, a *rand/2/best* mutation is considered as [33].

$$V_{id}^t = X_{id}^t + F^t(X_{id}^t - XBest_d^t) + F^t(A_d^t - B_d^t) \tag{28}$$

where X_{id}^t is the d^{th} dimension of i^{th} population among generation t. A and B are two random members from X^ts. V'_d is the d^{th} dimension of the i^{th} mutated vector in generation t, $XBest_d^t$ is the d^{th} dimension of the best solution in generation t. Meanwhile, F^t is a value called the *scaling factor*. In this paper the F^t is generated via a Gaussian chaotic map as

$$x_{n+1} = \exp(-b \cdot x_n^2) + c \tag{29}$$

where x is the representative of the chaotic random number [34]. The map features a chaotic behavior for many values of b and c. In this study b and c are considered to be 6.2 and -0.5, respectively. Since, the value of x is in the interval of [-0.2878, 0.5000], it is mapped to the interval of [0.5, 1] [35].

In the crossover, the same dimension of some members is exchanged with another one. The crossover that is used in this study is precisely the same as the ordinary DE in [33]. Table 2 indicates the parameters as well as the chaotic map used to calculate the mutant factor.

391 **Table 2** CDE parameters

Parameter	**Value**
Maximum iteration	50
Population	10 times of variables
F	rand(0.5, 1)*
Cr	0.6
Chaotic map	$x_{n+1} = \exp(-b \cdot x_n^2) + c$ $b = 6.2, c = 0.5$

392 *Random number is created via Gaussian chaotic map

393 **Remark 3:** In this study, any kind of optimization algorithm is applicable. However,
394 chaotic DE is selected since it is simple and at the same time powerful. Besides, its
395 dominance over ordinary DE and PSO is shown in [35].

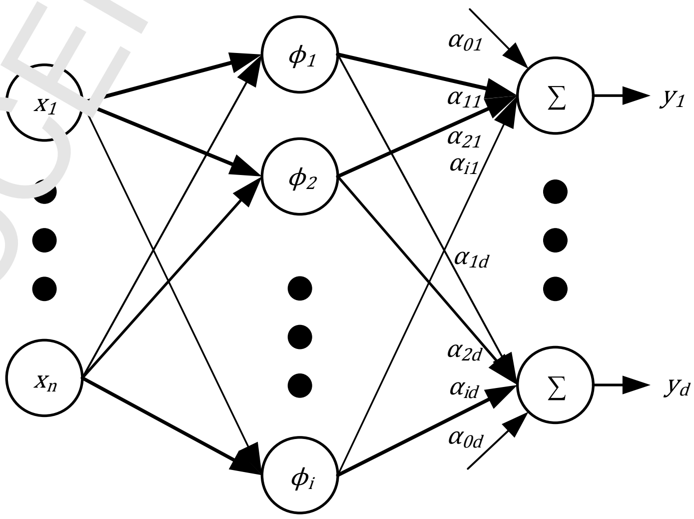
396 ### 4.1.2 RBF neural network

397 The basis of artificial neural networks is the human brain mechanism of learning and
398 producing knowledge. RBF neural networks, which its structure is presented in Figure 5
399 uses a single array of radial basis functions in the hidden layer, and the output layer is
400 usually considered as a linear function [36]. Thus, it has less parameter in comparison to
401 MLP and GMDH, which makes RBF more straightforward tool for function
402 approximation. RBF can be trained in a shorter time, and it works best if there are many
403 training vectors available [37].

404 The activation function in RBF neural network hidden layer is a Gaussian function as
405 follows:

$$\phi_i(x) = \exp\left(-\frac{\|x - C_i\|^2}{\sigma_i^2}\right), i = 1,2, \dots, k, \quad (30)$$

406 where C_i, which in the form of $C_i = [C_{i1}, C_{i2}, \dots, C_{in}]$ is the center of Gaussian radial
407 function $\phi_i(x)$, and $\sigma_i = [\sigma_1, \sigma_2, \dots, \sigma_k]$, is called spread and determines the width of
408 each Gaussian radial function. To train the neural network, the procedure proposed in
409 [37] is considered.



410

411 **Figure 5** RBF neural network structure

Remark 4: To predict the gains, the goal is to make a relation between the three inputs and the three outputs. It should be noted that any modeler, such as different kinds of artificial neural networks and regression models are applicable. However, artificial neural networks have shown better performance in wind energy related applications such as power curve estimation and fault detection [38, 39]. On the other hand, slightly better performance has been reported for RBF against MLP as a direct pitch controller [6].

4.2 Optimization and training

The optimization is done for 26 wind speeds between 12 m/s up to 24.5 m/s with the step of 0.5 m/s. To challenge the robustness, all the wind speeds have minimal fluctuation with a maximum frequency of 10 Hz (Figure 6) [40]. All the wind profiles are created via Kaimal wind model based on IEC 61400-3 [41]. The optimization problem is considered in (27). To calculate the IAE of the baseline PI controller, firstly the gains of the baseline PI are obtained from Figure 2. Then, by the constant gains, the IAE of the baseline PI controller is calculated for each wind speed profiles of Figure 6. Thus, during the optimization, the IAE of FOPID will be compared to IAE of the baseline PI controller, and if the constraint does not meet, a penalty function is applied. Table 3 shows the equivalent pitch angle, gains, IAE, and the cost function (25) for the baseline PI controller in some wind speeds.

It should be noted that the same method can be easily applied to an ordinary PID. Table 4 shows some of the optimal parameters of FOPID and PID.

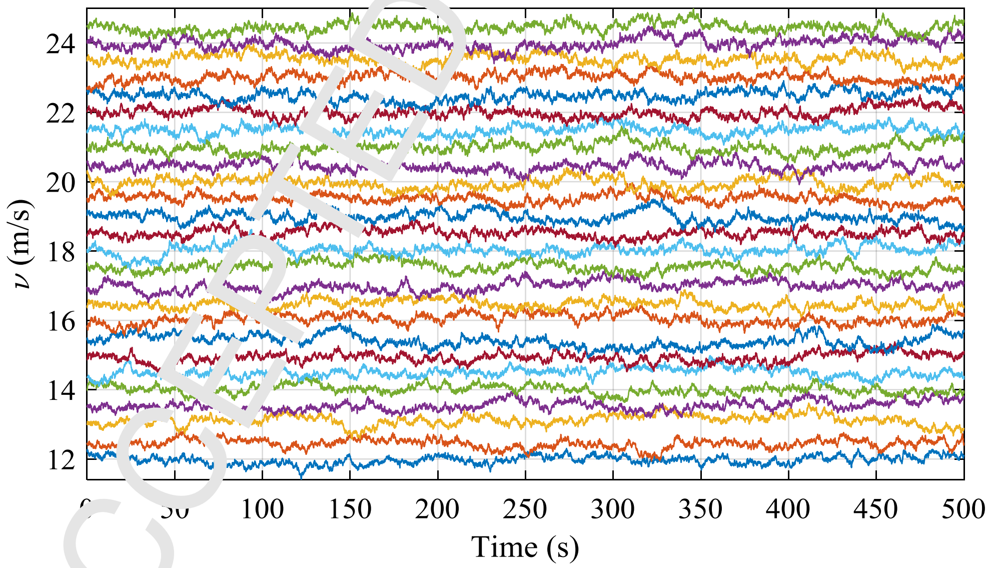


Figure 6 Wind speed profiles used for the optimization process

Table 3 Parameters of baseline PI controller

Wind speed	Equivalent pitch angle (deg)	K_p	K_i	IAE	Cost in (25)
14	8.7	-0.6298	-0.2699	144.449[illegible]	68.8277
15.5	9.6	-0.4912	-0.2105	171.0[illegible]49	62.6868
17	10.4	-0.4119	-0.1765	185.[illegible]42[illegible]	56.0194
18.5	11.3	-0.3593	-0.1540	207.5071	55.8160
20.5	12.0	-0.3108	-0.1332	251.5365	55.7465
22	12.8	-0.2838	-0.1216	267.6865	56.2760
24	13.5	-0.2559	-0.1097	31[illegible].1463	59.3156

Remark 5: Unlike many related kinds of literature [5, 7], in this paper, a fluctuated wind speed is used for optimization. The amplitude of these fluctuations is minimal. Therefore the values can be used instead of nominal constant wind speed. However, the variations can affect the performance significantly, since the behavior of the wind turbine varies in different wind frequencies. Therefore, to put the optimization in a more realistic condition, it is more appropriate to accomplish the optimization process in wind speed with real fluctuation frequencies.

Remark 6: The Oustaloup fractional-order approximation, which is used in this paper, is assumed to be a 5th order. The band frequency also is considered to be in the interval of [0.01,100] Hz, which is suitable for most of the industrial purposes [17].

Table 4 The optimized parameters of PID and FOPID

Controller	Wind speed	K_P	K_I	K_D	λ	μ	IAE	Cost in (25)
PID	14	-0.7103	-0.1695	-0.063244	1	1	144.4440	56.5092
	15.5	-0.5244	-0.1684	-0.062084	1	1	171.0113	54.3429
	17	-0.4567	-0.152[illegible]	-0.040277	1	1	185.3392	50.6492
	18.5	-0.365[illegible]	-0.1459	-0.043744	1	1	207.5067	50.9266
	20.5	-0.3009	-0.1325	-0.035805	1	1	251.5078	51.6529
	22	-0.265[illegible]	-0.1222	-0.033855	1	1	267.6456	51.1509
	24	-0.2290	-0.1112	-0.032115	1	1	312.6179	53.6180
FOPID	14	-0.4179	-0.2090	-0.3967	0.9368	0.4982	144.3742	53.2665
	15.5	-0.4[illegible]57	-0.1746	-0.2450	0.9850	0.5917	170.9041	51.0409
	17	-0.[illegible]685	-0.1930	-0.3316	0.9284	0.3962	185.2877	47.0226
	18.5	-0.3009	-0.1544	-0.1612	0.9724	0.6240	207.4520	48.3941
	20.5	-0.2807	-0.1346	-0.1086	0.9926	0.7014	251.5317	48.9513
	22	-0.2422	-0.1263	-0.1072	0.9843	0.7423	267.3900	47.4482
	24	-0.1978	-0.1200	-0.1020	0.9787	0.7150	312.4545	49.5714

It should be noted that it is observed that if the system is optimized for a fractional PI, the λ will converge toward 1, and the result is the same as integer-order PI.

As it is expressed in Subsection 3.2.1, another optimization is done in which; the fractional orders remain constant, equal to the average of the first optimization. The fact that μ and λ are nearly the same in all wind speeds validates this simplification. In this study, the average value for λ is 0.9607, and the average for μ is 0.6062. Table 5 shows these parameters for the new optimization for some wind speeds.

Table 5 The optimized parameters of FOPID

Wind speed	K_P	K_I	K_D	λ	μ	IAE	Cost in (25)
14	-0.5128	-0.1962	-0.3067	0.9607	0.6062	144.3939	53.2921
15.5	-0.4025	-0.1825	-0.2612	0.9607	0.6062	170.99[illegible]1	51.1794
17	-0.3135	-0.1724	-0.2053	0.9607	0.6062	185.2608	47.2882
18.5	-0.2861	-0.1594	-0.1731	0.9607	0.6062	20[illegible].45[illegible]8	48.4363
20.5	-0.2416	-0.1469	-0.1475	0.9607	0.6062	25[illegible].4[illegible]53	49.0613
22	-0.2081	-0.1363	-0.1329	0.9607	0.6062	2[illegible]7.6402	47.7781
24	-0.1635	-0.1306	-0.1355	0.9607	0.6062	[illegible]12.922	49.9602

To train the RBF neural network, the database is created for the wind speeds between 12 to 24.5 m/s with the step of 0.5 m/s, for the ω_r (which will be converted to ω_g) between 1 to 1.5 rad/s with the step of 0.0025 rad/s and for the pitch angle from 0° to 25° with the step of 1°. However, the entries that lead the power to become less than 4MWs and higher than 6MWs are eliminated, since the wind turbine does not see these conditions in the region above rated-speed. In this way, 10295 entries are created. Then the neural network is trained via the method that is discussed in subsection 4.1. For the RBF neural network, 10 neurons and 3 outputs are considered, so there are 30 weights and 3 biases that should be calculated via the training method. However, instead of ν as the output vector, the equivalent optimal gains are set. Thus, as it is shown in Figure 4, the outcome is an RBF neural network for each proposed controller, which it's input vector is β, ω_g and P_g, and its output vector is K_P, K_I, and K_D.

Remark 7: Since the inputs are not in the same order, all of them are normalized and mapped to the interval of [0, 1].

To determine the best spread value for the RBF neural network, the mean squared error (MSE) of several situations is considered. The training was conducted for ten times with different spreads (between 0.5 to 3 with the step of 0.1) for 70% of the database as train data, and then the best spread is chosen based on the MSE of remaining 30%. Table 6 demonstrates the average MSE for different spreads in test data.

Table 6 The average MSE of 10 RBF training for validation data with different spreads

Spread (σ)	0.5	0.7	1.0	1.5	2.4	2.9
For PID database	0.001900	0.001043	0.0008051	**0.0006553**	0.0007088	0.0009962
For FOPID database	0.001326	0.0007988	0.0006554	**0.0005505**	0.0005822	0.0006997

Based on Table 6, the spread for training the RBF neural network for both PID and FOPID is considered to be 1.5.

One of the most critical stages in design is to guarantee the performance mathematically. However, providing analytical proof in the wind turbine (even in the two-mass model) is not a straightforward task, because the aerodynamical equations and the structure of the controllers are highly nonlinear. On the other hand, in a more real condition, when the wind fluctuations are high and stochastic, the linear models for stability analysis do not provide a suitable background in the design. Thus, two test scenarios are brought in following. In the first one, the two-mass model is implemented, and the performance in different wind fluctuations is studied. In the next, the same controller that is designed for the two-mass model is implemented on a more detailed simulator; therefore the performance and robustness of the proposed controller is studied under different wind fluctuations in a more realistic situation.

4.3 Test on the two-mass model

In this Subsection, the proposed FOPID, proposed PID, baseline PI, RBF PI (Ref. [5]), and FOPI (Ref. [19]) are compared on the two-mass model. Eighteen wind speed profiles are generated based on the Kaimal wind model, adopted from the IEC 61400-3 [41], which includes different wind speeds average and different standard deviations.

The presented controller in [5], is an RBF based PI controller, which is trained based on an optimized dataset of PI controllers in different steady wind speeds. The IAE of generator speed is considered as an optimization cost function, and a sensor for wind speed measurement is assumed. Thus, this paper is a good example to study the effect of our proposed method.

The performance criteria, which are chosen to compare the controllers at the first step, will be RMS of generator speed error and RMS of control force rate. However, this is not satisfactory enough since different loads on the structure of the wind turbine should also be considered. The most critical loads on a wind turbine are the tower fore-aft moment and the blade root out-of-plane motions. The first one is the torque caused by movements of the tower to its front and back, and the second one is the motion of blades out of rotation plane. To compare the loads, their RMS around their mean value is calculated [42]. It is noticeable that the blade out-of-plane deflection refers to the deflection of the blade that is caused by wind and push the blade outside the rotation plane. Meanwhile, the blade in-plane deflection refers to a deflection inside the rotation plane. The moments caused by these deflections are called out-of-plane and in-plane moments, respectively. Figure 7 depicts these two blade deflections in a cross section of the rotation plane.

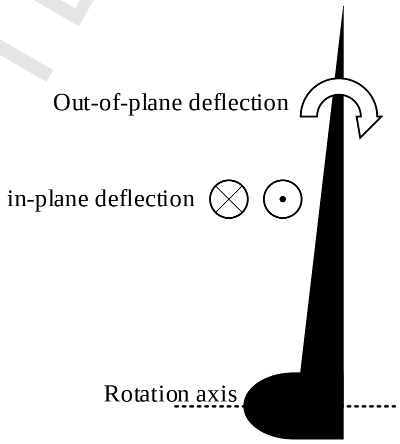


Figure 7 Cross section of a blade rotation plane

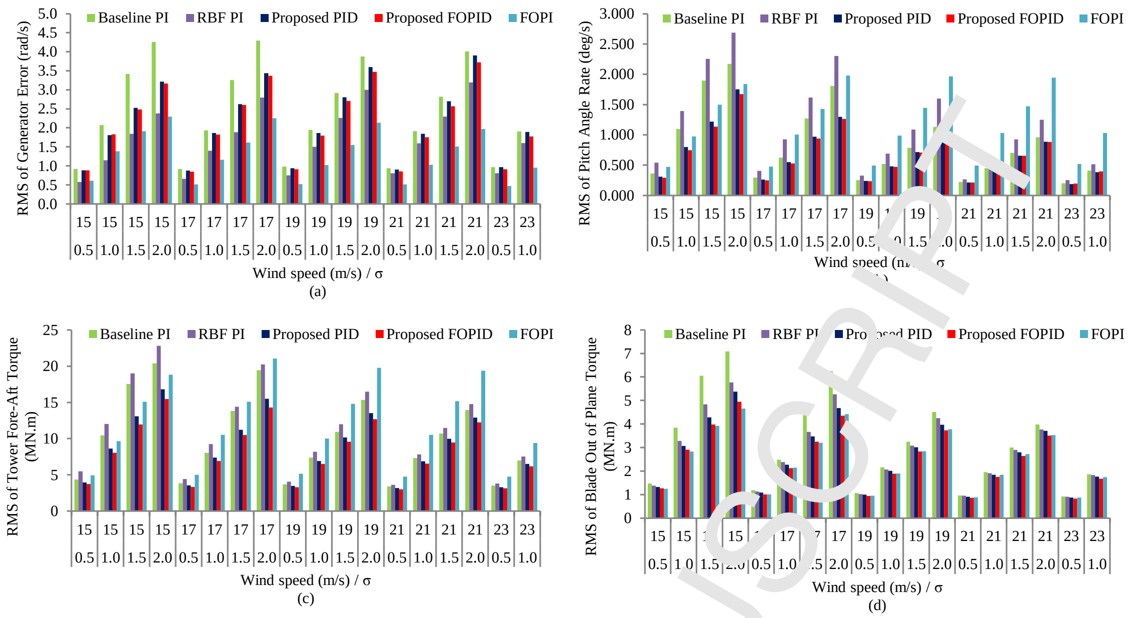


Figure 8 (a) RMS of generator speed error in 2-mass model. (b) RMS of pitch actuator rate in 2-mass model. (c) RMS of pitch actuator rate in 2-mass model. (d) RMS of the out-of-plane moment of blade root in 2-mass model

Figure 8 shows the performances of five controllers. To have a better comparison, the simulation time is considered 900 seconds. It should be noted that the absolute percentages are calculated via

$$A \text{ respect to } B = 100\frac{B-A}{B} \tag{31}$$

Figure 8 (a) shows the RMS of generator speed error. The FOPI has performed almost the best among all controllers by 38.0 % better performance comparing the proposed FOPID. On the other hand, the performance of RBF PI is 27.7% better than the proposed FOPID. The performance of the proposed FOPID is slightly better than the proposed PID in this figure, and the average error in the proposed FOPID is 3.1% better than the proposed PID. However, the baseline PI controller shows the weakest performance. The figure depicts that the proposed FOPID is minimizing the RMS by 11.2% in comparison to the baseline PI. Less value in RMS of the generator speed error means the rotor is under less torque variation.

Figure 8 (b) demonstrates the RMS of the pitch actuator rate. It can be seen that controllers are performing differently at different wind speeds. The proposed PID and proposed FOPID have less variation in pitch angle rate. Although, the proposed PID and the proposed FOPID has had better performance than the baseline PI by 13.8% and 15.0% in average, respectively, in some cases the baseline PI controller has been acted better than the other two controllers. However, the proposed FOPID is working better in minimizing pitch angle rate; it has reduced pitch actuator rate by 1.7% on average, in comparison to the proposed PID. The RBF PI has the weakest performance in lower wind speeds in pitch angle rate, while The FOPI had the most inferior performance in higher wind speeds. The FOPI performed 32.3% worse than the baseline PI, by average. On the other hand, RBF PI has achieved 31.0% worse than the baseline PI, mainly because there was no trace of u in the cost function.

Figure 8 (c), depicts that the proposed FOPID has reduced the RMS of the tower fore-aft moment by 6.0% and 16.3% in comparison to the proposed PID and the baseline PI, respectively. The proposed PID, on the other hand, has acted 11.0% better than the baseline PI in this survey. However, again in this part, the response of the baseline PI and the RBF PI controllers get better in higher wind speeds. Besides, the performance of the RBF PI and FOPI are respectively 9.9% and 17.3% weaker than the baseline PI controller. Thus, the proposed FOPID controller is more capable of reducing the cyclic loads to the wind turbine tower in comparison to the other controllers.

Figure 8 (d) demonstrates the RMS of the out-of-plane moment of the blade root, which directly affects the fatigue damages to the blades. Blades have the most risk of damages, among other components. Therefore, reducing the variations of this parameter is essential. It can be seen from Figure 8 (d) that the proposed FOPID acts the best among all controllers. Although in all cases, the proposed FOPID is working better than the proposed PID with the average of 5.9%, the behavior of the baseline PI is changing in different wind speeds in comparison to the proposed PID. The baseline PI controller is acting 17.1% worse than the proposed FOPID, while the proposed PID is performing 12.0% better than the baseline PI, on average. In this case, the performance of FOPI is the best among lower wind speeds, but it gets slightly worse than the proposed FOPID at higher wind speeds.

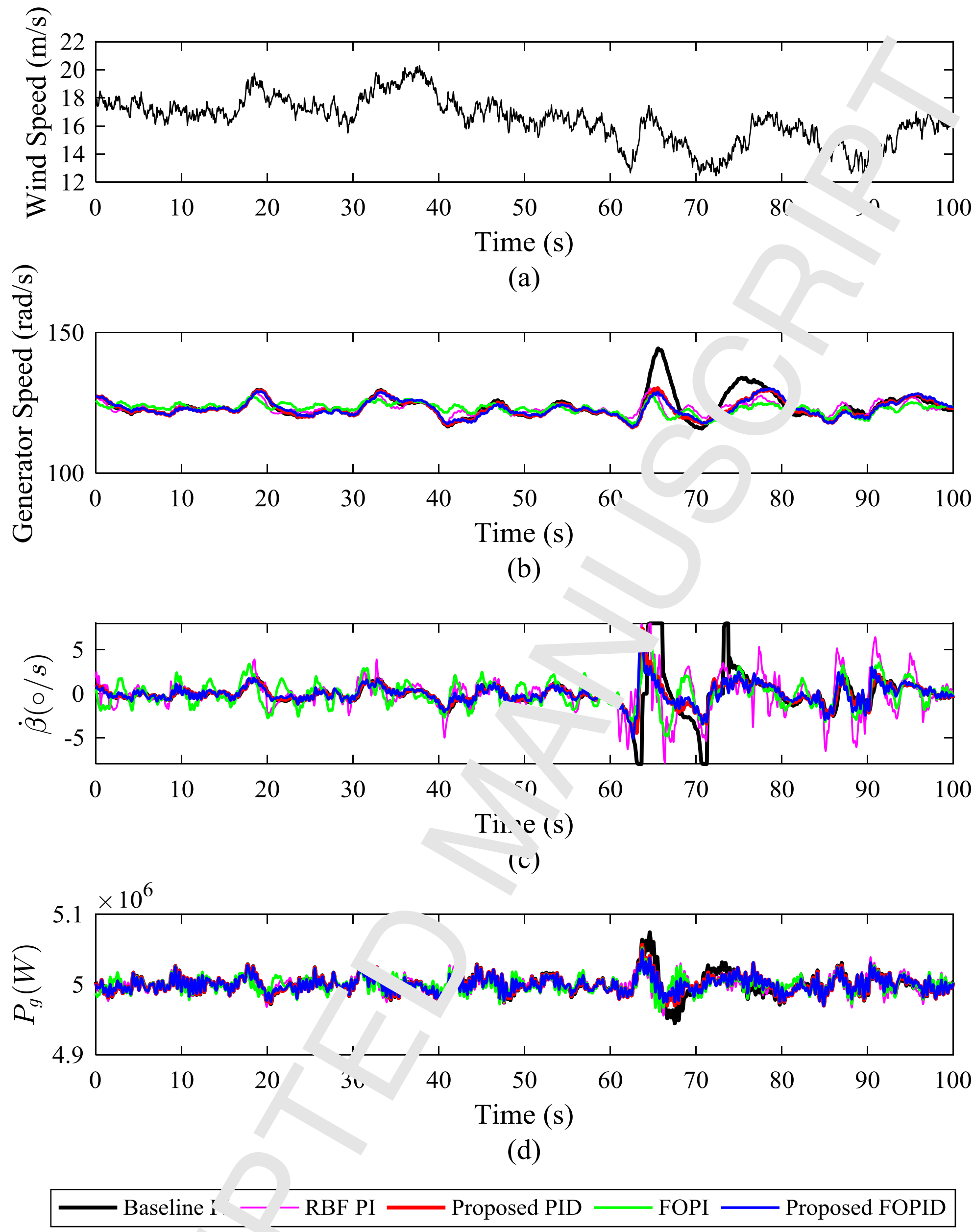
Remark 8: It is noteworthy that comparing the above values to Tables 4 and 5 reveals that the performance of the controllers varies in the presence of higher wind perturbation. Although the difference in cost function between the proposed PID and FOPID is small in the table, they differ higher in the test section. In addition, although the difference between RMS of generator error and control signal in test scenarios are small, the difference between the RMS of loads is much higher. It means that by a slight reduction in the (25) and even keeping the (26) near the same as the baseline PI, the proposed controllers are more capable of mitigating the loads. Besides, although the aim of this paper was not to decrease the IAE from the beginning, and IAE was only the optimization constraint, the proposed controllers showed a better performance in reducing the generator speed error.

Remark 9: While it seems trivial that by proposing a more sophisticated controller, better performance is achievable in some control desirables, in reality, the other aspects of designs might remain neglected. For instance, surely fuzzy controllers have much more parameters to set (membership functions and rule base), but in spite of better performance in regulating the rotor speed, the control signal becomes higher in comparison to a simple PI/FOPI controller. Thus, although more advanced controllers might reduce IAE, they do not necessarily resolve all the demands [43].

Figure 9 shows the above comparison of five mentioned controllers for an average wind speed of 17 m/s and gust of 1.5 m/s. Figure 9 (a) depicts 100 seconds of the wind speed that the simulation is done. Figure 9 (b) demonstrates the performance of five controllers in generator speed adjustment. As can be seen in time between 60 seconds to 80 seconds, the proposed FOPID and proposed PID were more capable of keeping the performance near the desired value (122.9 rad/s) in comparison to the baseline PI, but the FOPI has the best performance overall in this section. Figure 9 (c) shows the rate of pitch actuator. Interestingly, unlike the baseline PI controller, none of the other controllers have led the actuator to become saturated between 60 seconds to 80 seconds. Besides, the peak of the rate of pitch angle on the proposed FOPID is less in comparison to the other controllers. The figure depicts that the RBF PI and FOPI controllers have more fluctuation in their performance. Figure 9 (d) shows the generated power. Based on this figure, the proposed FOPID has superiority against the proposed PID, the baseline PI, RBF PI, and FOPI controllers in adjusting the generated power on its nominal (5 MWs). Figures 10 (a) and 10 (b) show the tower fore-aft moment and out-of-plane blade root moment of five controllers, respectively. It can be seen that the proposed FOPID reaches the smallest moments and thus, mitigates the mechanical loads the most.

4.4 Validation via the FAST

In this paper, FAST code is utilized to predict a more realistic performance of the wind turbine. This code is a powerful tool, which is capable of simulating the loads and control performance of wind turbine if the structural properties, such as blade and tower configurations, are entirely defined [7, 44]. This code cooperates with the aerodynamic subroutine AeroDyn, which provides a detailed analysis of aerodynamics by blade element momentum theory (BEM) and dynamic stall [45]. Since the baseline NREL 5-MW wind turbine is fully defined in FAST V8.0; it is implemented to validate the control performance in this paper.



608

609 **Figure 9** The performance of five controllers in a wind speed of 17 m/s with a standard deviation of 1.5
610 m/s. (a) The wind speed profile. (b) The generator speed. (c) rate of pitch angle. (d) The generated power

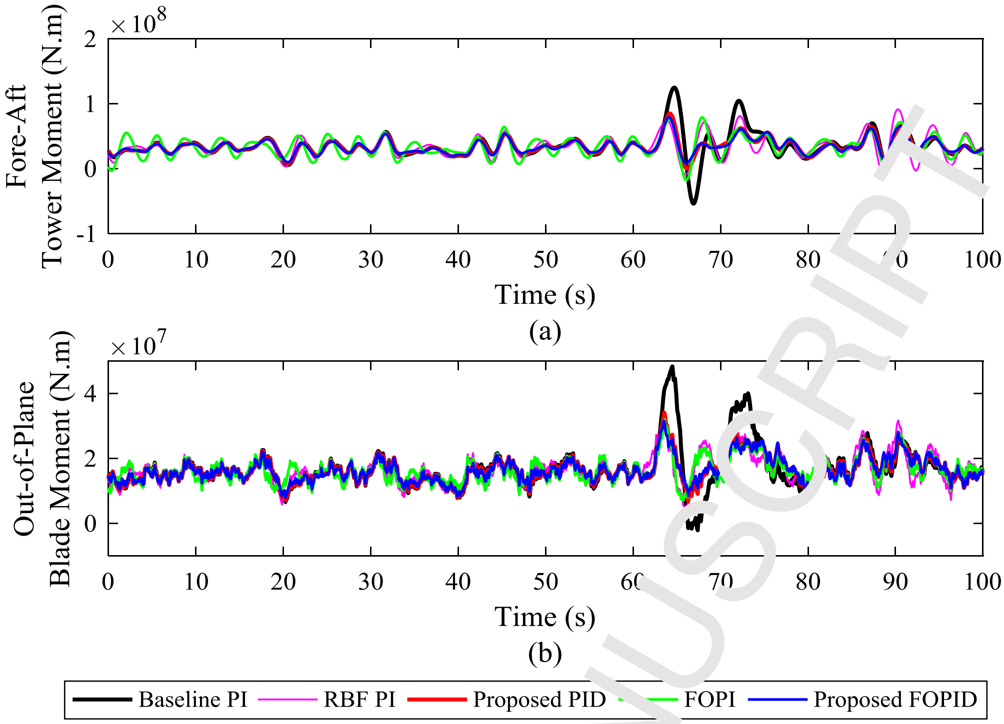


Figure 10 The applied loads in five controllers in a wind speed of 17 m/s with a standard deviation of 1.5 a) The fore-aft tower moment b) the out-of-plane blade moment

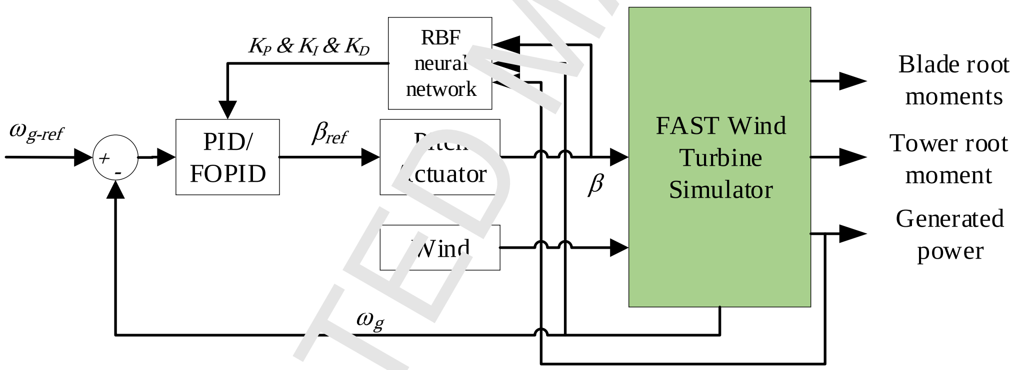


Figure 11 Scheme of implementation of FAST code

Nature always is more complicated than our constructed models and simulations. Thus, to make a better comparison and challenge the robustness, a more detailed model is implemented. The model that is used to derive the parameters (which was discussed in Section 2) had many neglected dynamics, such as the side-side movements and the blades both in-plane, out-plane deflections and the interaction between blades and tower. These deflections can affect performance and cause unexpected behavior or even instability. However, with the FAST code, the designer will be able to anticipate many of this ignorance. Although FAST is only a simulator and not a real setup, it makes our proposed controller one more step nearer to a real situation. FAST is also capable of predicting extreme loads and fatigue damages in different wind speeds [44]. In this study, the first blades edgewise mode, the first and second blade flapwise modes, the first and second tower side-to-side and fore-aft mode, the drivetrain flexibility and the generator DOFs are simulated. Remarkably, FAST is not equipped with a pitch actuator model; thus, the same differential equation in (10) is considered for following simulations.

Figure 11 depicts a schematic block diagram of FAST code in our proposed method. It should be noted that many studies have used the FAST to validate their results [7, 35, 42]. To show the effectiveness of the proposed method, the controllers (Proposed PID/FOPID, baseline PI, RBF PI, and FOPI) that were designed for the simplified two-mass system and tested in the previous section are applied to the FAST simulator.

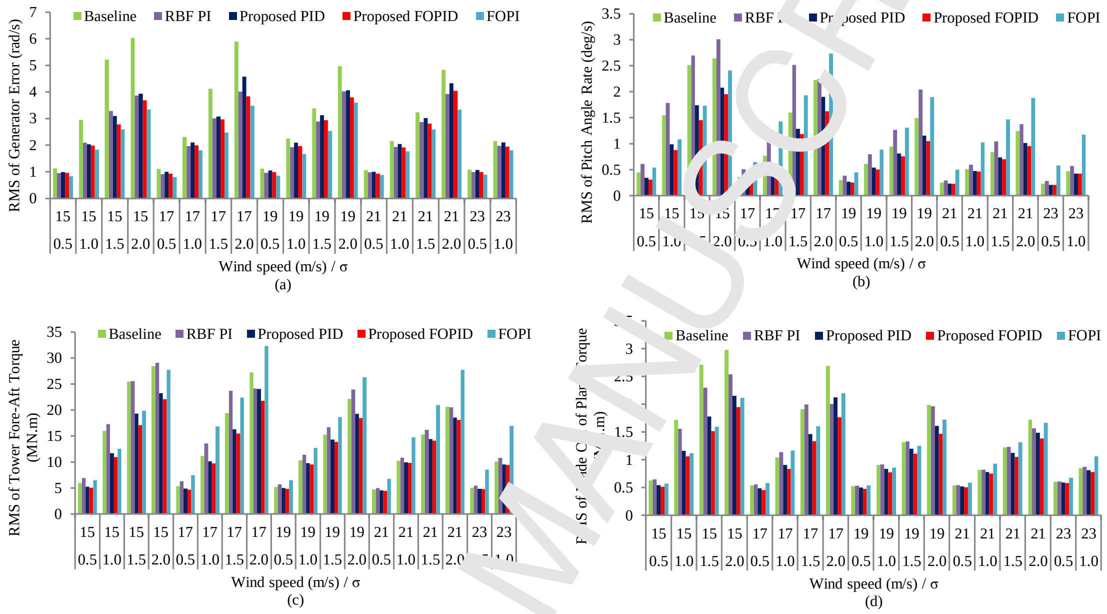


Figure 12 (a) RMS of generator speed error in FAST. (b) RMS of pitch actuator rate in FAST. (c) RMS of tower fore-aft moment in FAST. (d) RMS of the out-of-plane moment of blade root in FAST

Thus, in this section, the controllers will be faced with some unmodeled dynamics as well as the wind fluctuations. The wind models are precisely the same as wind profiles in Subsection 4.3 and are created via Kaimal wind model [41]. The same criteria of Subsection 4.3 are used in part as well: The RMS of generator speed error, RMS of pitch angle rate and RMS of tower root and out of plane blade root moments.

Figure 12 compares the performance of five controllers in different aspects. Figure 12 (a), shows the RMS of the generator speed error of five controllers. It is observed that in all of the cases, the FOPI has the best control performance. The proposed FOPID has 19.3% and 6.6% better performance in comparison to the baseline PI and the proposed PID, respectively. However, the proposed PID has acted 13.6% better than the baseline PI. The RBF PI controller has performed 18.7% better than the baseline PI, but its performance was slightly weaker than the proposed FOPID on average. Besides, FOPI has shown 10.6% better than the proposed FOPID. As it is seen in this part, the difference between the IAE of five controllers is increased in comparison to the previous subsection.

Figure 12 (b), compares the actuator rate among five controllers. Like what it is observed in the two-mass model, the performance of baseline PI and the RBF PI improve as the wind speed rises. However, the difference between rates of pitch actuator is more sensible in the FAST model. In many cases, the proposed FOPID showed less actuator rate in comparison to the proposed PID, which is 7.2%, on average. However, the proposed FOPID shows 22.4% less actuator rate in contrast to the baseline PI controller. On the other hand, the proposed PID has a 10.7% less actuator rate than the baseline PI controller. Like the previous section, the pitch angle rate in FOPI is the worst in higher wind speeds, and it is worked 25.4% worse than the baseline PI controller. Besides, RBF PI has performed 19.8% worse than the baseline PI. Figures 12 (a) and 12 (b) demonstrate that the proposed FOPID achieved to the least RMS of the generator speed error and actuator rate.

Figure 12 (c) shows the RMS of the tower root moments. This figure depicts that, as the wind rises, the performance of the baseline PI and the RBF PI controller get better. By average, the proposed FOPID reduces the moment by 3.9% in comparison to the proposed PID. On the other hand, the proposed FOPID has acted 13.3% better than the baseline PI controller. Besides, the proposed PID has worked 9.8% better than the baseline PI. The performance of FOPI is 17.2% worse than the baseline PI. On the other hand, the RBF PI controller has performed almost 7% worse than the baseline PI controller.

Figure 12 (d) demonstrates the difference of controllers for the out-of-plane moment of the blade root. It is shown that the proposed FOPID has superiority in all cases over the other controllers. RMS of the out-of-plane moment of blade root for the proposed FOPID is 7.4% better than the proposed PID, whereas it has 19.7% better performance in comparison to baseline PI. The proposed PID has also acted 13.6% better than the baseline PI controller. The RBF PI and FOPI controllers have performed just 2.6% and 4.8% better than the baseline PI, respectively.

Figure 13, depicts loads and performances for one of the wind profile cases. Figure 13 (a) shows 100 seconds of 17 m/s wind speed with a standard deviation of 1.5 m/s. Figure 13 (b), demonstrates the errors of the baseline PI, the proposed PID, the proposed FOPID, RBF PI, and FOPI controllers. As it is seen in the figure, the FOPI has slightly better performance in speed regulation. The difference is more vivid in the times between 60 seconds to 80 seconds. Figure 13 (c) depicts the rate of pitch angle in five controllers. In this survey, a small superiority in the proposed FOPID against the proposed PID is observed. Although four out of five controllers have led the actuator to its limits, it is shown that the proposed PID and proposed FOPID have reached the nominal values sooner. Although the plant with FOPI is not saturated, the fluctuation in its operation is much more. Figure 13 (d) shows the generated power. Based on this figure, the proposed FOPID has got superiority against the proposed PID and the baseline PI controllers in adjusting the generated power. Figures 14 (a) and 14 (b) show that the amplitudes of tower fore-aft and the blade out of the plane moment in the proposed FOPID, the proposed PID, the baseline PI, the RBF PI, and the FOPI. From Figures 14 (a) and 14 (b), it can be seen that the proposed FOPID is able to mitigate the mechanical load most effectively since it can decrease the tower and blade moments, the most.

Using the FAST simulator, it can be seen that not only the proposed method is robust enough to tolerate more real conditions, but also the performance that is achieved in the previous subsection remains, relatively.

Remark 10: For more clarification, Figure 15 depicts the overall design process of the proposed method as a flowchart. It should be noted that the optimization (using chaotic DE) and training of neural network are offline procedures. Then, the trained neural network is used (without any online optimization) to tune the parameters of the fractional-order PID controller making a gain-scheduling fractional-order PID controller.

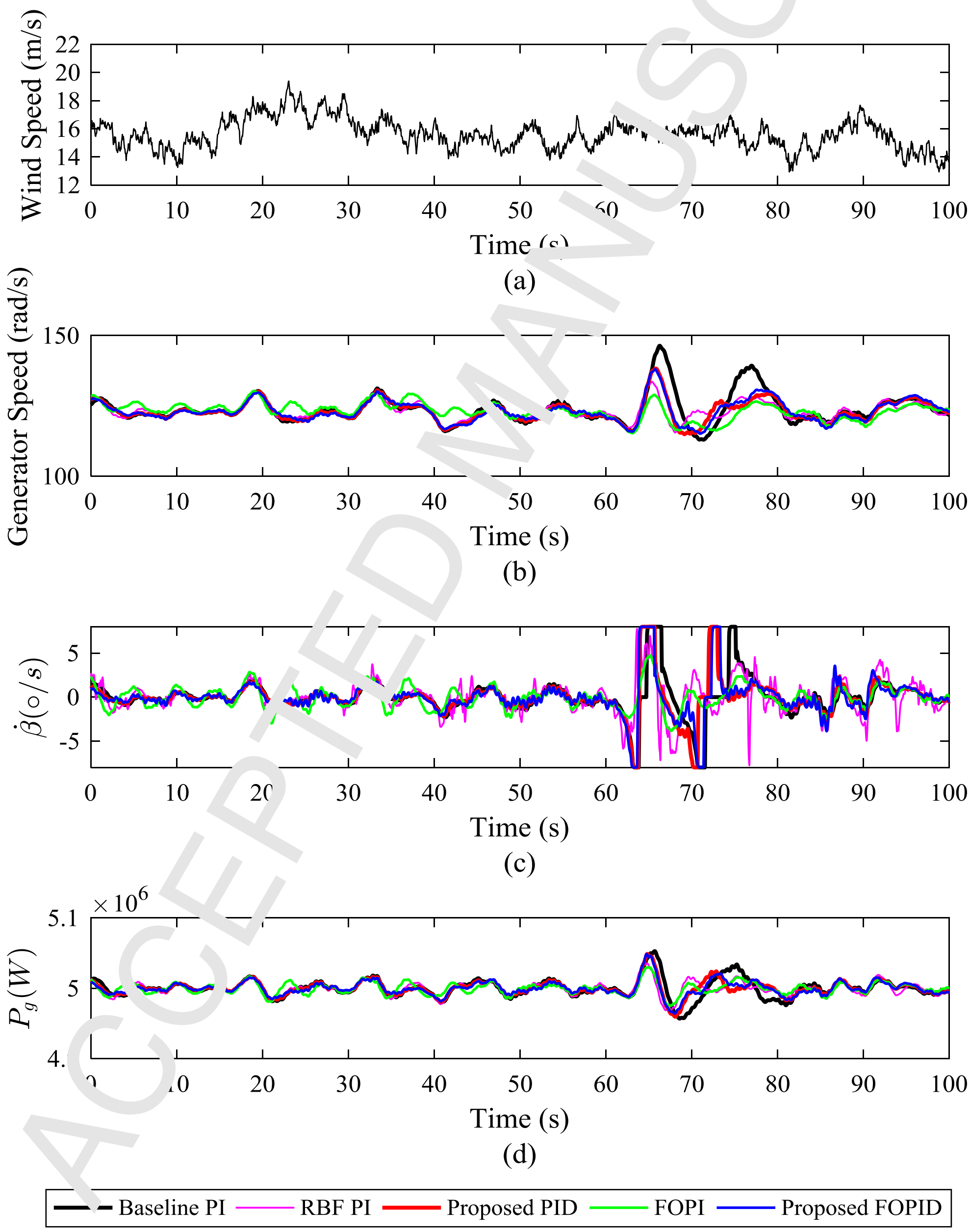


Figure 13 The performance of five controllers in a wind speed of 17 m/s with a standard deviation of 1.5 in the FAST simulator (a) The wind speed profile. (b) The generator speed (c) The rate of pitch angle (d) The generated power

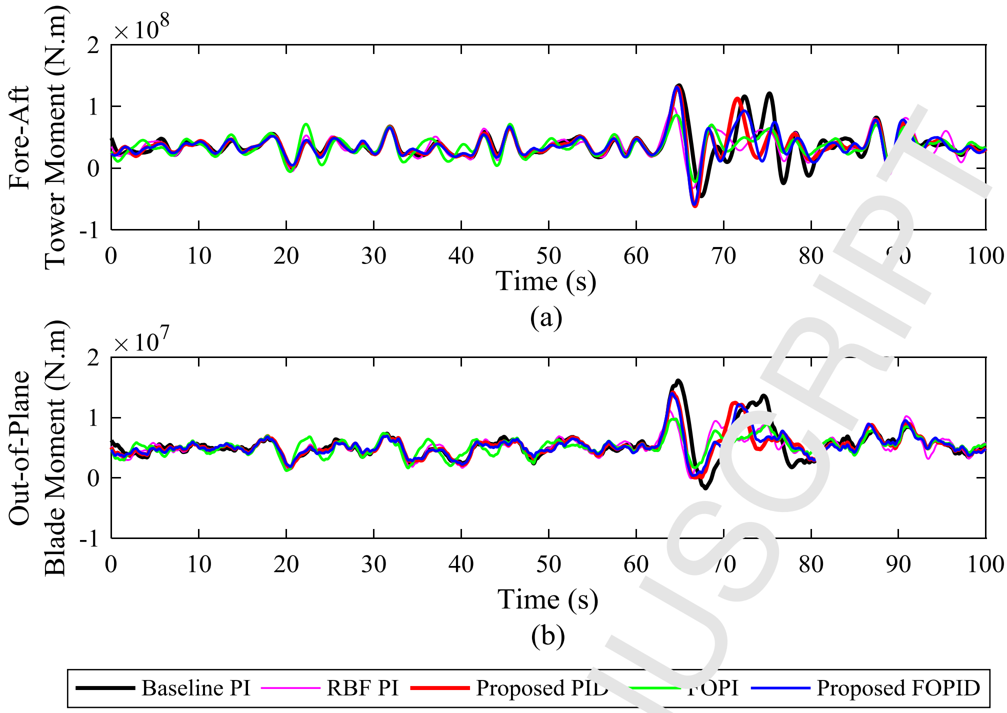


Figure 14 The applied loads in five controllers in a wind speed of 17 m/s with a standard deviation of 1.5 in the FAST simulator (a) The fore-aft tower moment. (b) the out-of-plane blade moment

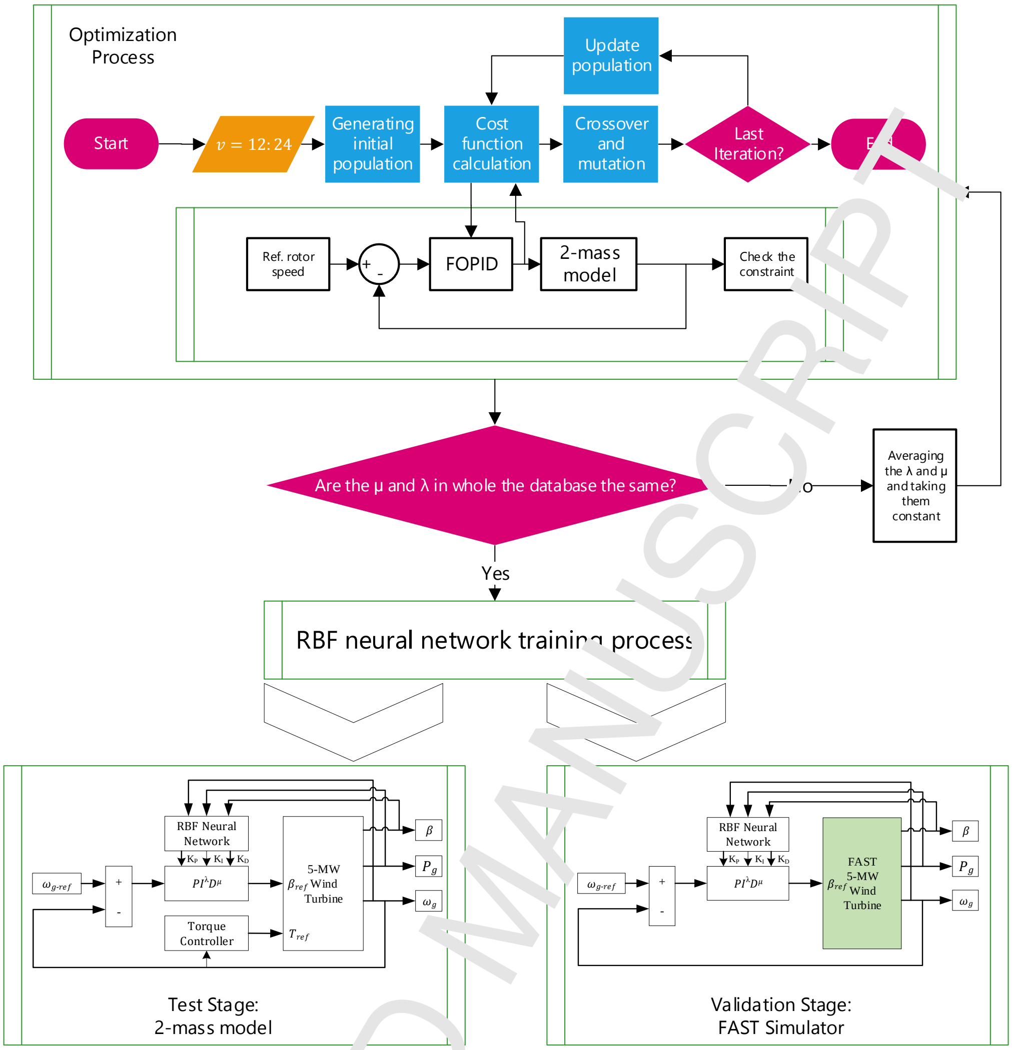


Figure 15 The proposed controller design process

5 Conclusion

In this study, an RBF based fractional-order PID (FOPID) has been applied to control the pitch angle concerning mitigation of mechanical loads. To train the RBF neural network, a dataset of optimal gains and orders is provided for several wind speeds by solving a suitable optimization problem using chaotic differential evolution (CDE) algorithm. Since by changing the direction of the force vector on blades, the pitch angle rate has a significant effect on the loads. Thus, the cost function for this optimization problem has been considered the rate of the control signal. Meanwhile, to maintain the performance, a constraint on error has been defined. To compare the performance a simplified two-mass model has been used with different wind speeds and fluctuations. The simulation has shown that a better performance is achievable in the proposed FOPID, comparing to the other controllers. In the second scenario, the controllers, which have been designed for the simplified model, have been tested on a more realistic standard simulator called FAST. It has been shown that in many cases the proposed FOPID has reached better performance and robustness with less actuator rate, in comparison to the other controllers. Besides, it was observed that the proposed FOPID controller is more capable of alleviating mechanical loads in comparison to the same structure PID, the baseline PI controllers, the RBF PI, and the FOPI.

For future research, since many possible faults can easily affect the wind turbine operation, such as blade damages, actuator failures or natural accidents such as bird strike a study on the fault tolerance characteristics of the proposed controllers is suggested. One other suggestion is to do the same framework, with a multi-objective optimization instead of the single-objective. Meanwhile, more parameters can be taken into accounts, such as direct consideration of blades and tower mechanical loads.

References

[1] Camblong, H. (2008). Digital robust control of a variable speed pitch regulated wind turbine for above rated wind speeds. *Control Engineering Practice, 16*(8), 946-958. doi:10.1016/j.conengprac.2007.11.004

[2] GWEC, G. W. E. C. (2017). Global wind report. 2015. *Brussels: GWEC*.

[3] Moradi, H., & Vossoughi, G. (2015). Robust control of the variable speed wind turbines in the presence of uncertainties: A comparison between H∞ and PID controllers. *Energy, 90*, 1508-1521. doi:10.1016/j.energy.2015.06.100

[4] Novak, P., Ekelund, T., Jovik, I., & Schmidtbauer, B. (1995). Modeling and control of variable-speed wind-turbine drive-system dynamics. *IEEE Control systems, 15*(4), 28-38.

[5] Poultangari, I., Shahnazi, R., & Sheikhan, M. (2012). RBF neural network based PI pitch controller for a class of 5-MW wind turbines using particle swarm optimization algorithm. *ISA Trans, 51*(5), 641-648. doi:10.1016/j.isatra.2012.06.001

[6] Yilmaz, A. S., & Özer, Z. (2009). Pitch angle control in wind turbines above the rated wind speed by multi-layer perceptron and radial basis function neural networks. *Expert Systems with Applications, 36*(6), 9767-9775.

[7] Ren, Y., Li, L., Brindley, J., & Jiang, L. (2016). Nonlinear PI control for variable pitch wind turbine. *Control Engineering Practice, 50*, 84-94. doi:10.1016/j.conengprac.2016.02.004

[8] Gao, R., & Gao, Z. (2016). Pitch control for wind turbine systems using optimization, estimation and compensation. *Renewable Energy, 91*, 501-515. doi:10.1016/j.renene.2016.01.057

[9] Macquart, T., & Maheri, A. (2019). A stall-regulated wind turbine design to reduce fatigue. *Renewable Energy, 133*, 964-970. doi:https://doi.org/10.1016/j.renene.2018.10.089

[10] de Corcuera, A. D., Pujana-Arrese, A., Ezquerra, J. M., Segurola, E., & Landaluze, J. (2012). H∞ Based Control for Load Mitigation in Wind Turbines. *Energies, 5*(12), 938-967. doi:10.3390/en5040938

[11] Bossanyi, E. (2003). Wind turbine control for load reduction. *Wind energy, 6*(3), 229-244.

[12] Stol, K. A., Zhao, W., & Wright, A. D. (2006). Individual blade pitch control for the controls advanced research turbine (CART). *Journal of solar energy engineering, 128*(4), 498-505.

[13] Han, B., Yang, F., Xiang, Z., & Zhou, L. (2016). Individual pitch controller based on fuzzy logic control for wind turbine load mitigation. *IET Renewable Power Generation, 10*(5), 687-693. doi:10.1049/iet-rpg.2015.0320

[14] Tibaldi, C., Hansen, M. H., & Henriksen, L. C. (2014). *Optimal tuning for a classical wind turbine controller.* Paper presented at the Journal of Physics: Conference Series.

[15] Macquart, T., Maheri, A., & Busawon, K. (2017). A decoupling control strategy for wind turbine blades equipped with active flow controllers. *Wind energy, 20*(4), 569-584.

[16] Zamani, M., Karimi-Ghartemani, M., Sadati, N., & Parniani, M. (2009). Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Engineering Practice, 17*(12), 1380-1387. doi:10.1016/j.conengprac.2009.07.005

[17] Chen, Z., Yuan, X., Ji, B., Wang, P., & Tian, H. (2014). Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II. *Energy Conversion and Management, 84*, 390-404. doi:10.1016/j.enconman.2014.04.052

[18] Hajiloo, A., Nariman-zadeh, N., & Moeini, A. (2012). Pareto optimal robust design of fractional-order PID controllers for systems with probabilistic uncertainties. *Mechatronics, 22*(6), 788-801. doi:https://doi.org/10.1016/j.mechatronics.2012.04.003

[19] Viveiros, C., Melício, R., Igreja, J., & Mendes, V. M. F. (2015). Performance assessment of a wind energy conversion system using a hierarchical controller structure. *Energy Conversion and Management, 93*, 40-48. doi:https://doi.org/10.1016/j.enconman.2015.01.002

[20] Pan, I., Das, S., Mukherjee, A., & Gupta, A. (2011). Gain and Order Scheduling of Optimal Fractional Order PID Controllers for Random Delay and Packet Dropout in Networked Control Systems. *Advanced Materials Research, 403-408*, 4814-4820. doi:10.4028/www.scientific.net/AMR.403-408.4814

[21] Jonkman, J., Butterfield, S., Musial, W., & Scott, G. (2009). *Definition of a 5-MW reference wind turbine for offshore system development* (NREL/TP-500-38060). Retrieved from Golden, CO:

[22] Muyeen, S. M., Ali, M. H., Takahashi, R., Murata, T., Tamura, J., Tomaki, Y., Sakahara, A., & Sasano, E. (2007). Comparative study on transient stability analysis of wind turbine generator system using different drive train models. *IET Renewable Power Generation, 1*(2), 131-141. doi:10.1049/iet-rpg:20060030

[23] Abdeddaim, S., & Betka, A. (2013). Optimal tracking and robust power control of the DFIG wind turbine. *International Journal of Electrical Power & Energy Systems, 49*, 234-242. doi:10.1016/j.ijepes.2012.12.014

[24] Pena, R., Clare, J. C., & Asher, G. M. (1996). A doubly fed induction generator using back-to-back PWM converters supplying an isolated load from a variable speed wind turbine. *IEE Proceedings - Electric Power Applications, 143*(5), 380-387. doi:10.1049/ip-epa:19960454

[25] Hansen, M. H., Hansen, A. D., Larsen, T. J., Øye, S., Sørensen, P., & Fuglsang, P. (2005). *Control design for a pitch-regulated, variable speed wind turbine* (8755034098). Retrieved from

[26] Muhando, E. B., Senjyu, T., Uehara, A., & Funabashi, T. (2011). Gain-Scheduled H∞ Control for WECS via LMI Techniques and Parametrically Dependent Feedback Part II: Controller Design and Implementation. *IEEE Transactions on Industrial Electronics, 58*(1), 57-65. doi:10.1109/TIE.2010.2045414

[27] Pan, I., & Das, S. (2013). *Intelligent fractional order systems and control: an introduction* (Vol. 438). Berlin: Springer.

[28] Efe, M. Ö. (2011). Fractional order systems in industrial automation—a survey. *IEEE Transactions on Industrial Informatics, 7*(4), 582-591.

[29] Monje, C. A., Chen, Y., Vinagre, B. M., Xue, D., & Feliu-Batlle, V. (2010). *Fractional-order systems and controls: fundamentals and applications*: Springer Science & Business Media.

[30] Wei, Q., Wei, Z., Aller, J. M., & Harley, R. G. (2008). Wind Speed Estimation Based Sensorless Output Maximization Control for a Wind Turbine Driving a DFIG. *IEEE Transactions on Power Electronics, 23*(3), 1156-1169. doi:10.1109/tpel.2008.921185

[31] Odgaard, P. F., Stoustrup, J., & Kinnaert, M. (2013). Fault-Tolerant Control of Wind Turbines: A Benchmark Model. *IEEE Transactions on Control Systems Technology, 21*(4), 1168-1182. doi:10.1109/TCST.2013.2259235

[32] Maione, G., & Lino, P. (2007). New tuning rules for fractional PIα controllers. *Nonlinear dynamics, 49*(1), 251-257. doi:10.1007/s11071-006-9125-x

[33] Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation, 13*(2), 398-417.

[34] Hilborn, R. C. (2000). *Chaos and nonlinear dynamics: an introduction for scientists and engineers*: Oxford University Press on Demand.

[35] Asgharnia, A., Shahnazi, R., & Jamali, A. (2018). Performance and robustness of optimal fractional fuzzy PID controllers for pitch control of a wind turbine using chaotic optimization algorithms. *ISA Transactions, 79,* 27-44. doi:https://doi.org/10.1016/j.isatra.2018.04.016

[36] Lu, Y., Sundararajan, N., & Saratchandran, P. (1998). Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. *IEEE Transactions on Neural Networks, 9*(2), 308-318. doi:10.1109/72.661125

[37] Demuth, H. B., Beale, M. H., & MathWorks Inc. (1994). *MATLAB : neural network toolbox*. Natick, MA: MathWorks.

[38] Li, S., Wunsch, D. C., O'Hair, E., & Giesselmann, M. G. (2001). Comparative analysis of regression and artificial neural network models for wind turbine power curve estimation. *Journal of solar energy engineering, 123*(4), 327-332.

[39] Schlechtingen, M., & Santos, I. F. (2011). Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mechanical systems and signal processing, 25*(5), 1849-1875.

[40] Seixas, M., Melício, R., Mendes, V. M. F., & Couto, C. (2016). Blade pitch control malfunction simulation in a wind energy conversion system with MPC five-level converter. *Renewable Energy, 89,* 339-350. doi:10.1016/j.renene.2015.12.005

[41] Commission, I. E. (2005). IEC 61400-1: Wind turbines part 1: Design requirements. In *International Electrotechnical Commission*.

[42] Jeong, J., Park, K., Jun, S., Song, K., & Lee, D.-H. (2012). Design optimization of a wind turbine blade to reduce the fluctuating unsteady aerodynamic load in turbulent wind. *Journal of mechanical science and technology, 26*(3), 827-838.

[43] Viveiros, C., Melício, R., Igreja, J. M., & Mendes, V. M. (2014). *Fuzzy, integer and fractional-order control: Application on a wind turbine benchmark model.* Paper presented at the Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference On.

[44] Jonkman, J. M., & Buhl Jr, M. L. (2005). *FAST User's Guide-Updated August 2005* (No. NREL/TP-500-38230). Retrieved from

[45] Moriarty, P. J., & Hansen, A. C. (2005). AeroDyn theory manual. In. Golden, CO (US): National Renewable Energy Lab., Golden, CO (US).

*Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this article