

# **SPEED CONTROL OF VARIABLE SPEED WIND TURBINE USING FOPID CONTROLLER**

A PROJECT REPORT

submitted by

**AHNA NOUSHAD**

**(Reg. No. TKM20EEII01)**

to

the APJ Abdul Kalam Technological University  
in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology

in

Electrical and Electronics Engineering

with specialisation in

*Industrial Instrumentation and Control*



**Department of Electrical and Electronics Engineering**

**TKM College of Engineering**

**Kollam - 691005**

**KERALA**

**JULY 2022**

## DECLARATION

I undersigned hereby declare that the project report entitled "**Speed Control of Variable Speed Wind Turbine Using FOPID Controller**", submitted for partial fulfillment of the requirements for the award of degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of *Prof. Salil N*, Assistant Professor, Department of Electrical and Electronics Engineering. This submission represents my ideas in my own words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam  
July 01, 2022

**AHNA NOUSHAD**

# DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

TKM COLLEGE OF ENGINEERING

Kollam - 691005



## CERTIFICATE

This is to certify that the report entitled " **Speed Control of Variable Speed Wind Turbine Using FOPID Controller** " submitted by **AHNA NOUSHAD** , (Reg. No. **TKM20EEII01**) of fourth semester to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electrical and Electronics Engineering with specialisation in Industrial Instrumentation and Control, is a bonafide record of the project work done by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

**Prof. Salil N**  
Project Supervisor  
Assistant Professor  
Department of EEE  
TKM College of Engineering

**Prof. Sumayya Jaleel**  
Project Coordinator  
Assistant Professor  
Department of EEE  
TKM College of Engineering

**Prof. Shanavas T N**  
PG Coordinator  
Associate Professor  
Department of EEE  
TKM College of Engineering

**Dr. Sabeena Beevi K**  
Head of the Department  
Associate Professor  
Department of EEE  
TKM College of Engineering

# Acknowledgement

A lot of effort and hard work has been put into this project in course of its presentation. However, it would not have been possible without the kind support and help of many individuals and other sources. I would like to extend my sincere thanks to all of them. I take this opportunity to express my deep sense of gratitude and sincere thanks to all who helped me to complete this project report successfully.

I express my sincere thanks *Dr. T A Shahul Hameed*, Principal, TKM College of Engineering for his encouragement in the completion of my project.

I thank *Dr. Sabeena Beevi K*, Head of the Department, Department of Electrical and Electronics Engineering, *Dr. Imthias Ahamed T P*, Professor, Department of Electrical and Electronics Engineering and *Prof. Shanavas T N*, P G Coordinator, Department of Electrical and Electronics Engineering for their support and cooperation.

I am greatly thankful to my Project Supervisor *Prof. Salil N*, Assistant Professor, Department of Electrical and Electronics Engineering for his supervision, assistance and helpful suggestions.

I am deeply indebted to *Prof. Sumayya Jaleel*, Assistant Professor, Project Coordinator and *Prof. Amal A*, Assistant Professor, Department of Electrical and Electronics Engineering, for their excellent guidance, positive criticism and valuable comments.

Finally I thank my parents, friends, near and dear ones who directly and indirectly contributed to the successful completion of my project.

AHNA NOUSHAD

# Abstract

Wind power generation systems, which can be run at constant speed or variable speed, are gaining a lot of attention as clean and secure renewable energy sources. Due to an increase in the production of wind energy, variable speed generation systems are more appealing than fixed speed systems. A variable speed wind turbine should run as close as practical to its ideal power coefficient to maximize the harvest of wind energy. Here, the speed of wind turbine with dual mass is controlled using a Fractional Order PID (FOPID) controller whose parameters are tuned using computational intelligence approaches like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). A comparison with the two computational tuning approaches is carried out in different wind profiles using MATLAB software. Simulation results shows the higher performance of FOPID controller design technique when compared with conventional PID in terms of multiple performance evaluation indices.

# Contents

**Abstract**

**List of Tables** **i**

**List of Figures** **ii**

**Abbreviations** **iii**

**Notations** **iv**

**1 INTRODUCTION** **1**

1.1 General Background . . . . . 1

1.2 Objectives . . . . . 2

1.3 Organisation of the Report . . . . . 2

**2 LITERATURE REVIEW** **4**

2.1 Introduction . . . . . 4

2.2 Basic Structure of Wind Turbine . . . . . 4

2.3 Ideal Power Curve . . . . . 6

2.4 Conclusion . . . . . 8

**3 WIND TURBINE DYNAMICS** **9**

3.1 Introduction . . . . . 9

3.2 Modelling . . . . . 9

3.3 Conclusion . . . . . 12

**4 DESIGN OF CONTROLLER** **13**

4.1 Introduction . . . . . 13

4.2	Fractional Calculus . . . . .	13
4.2.1	PID Controller with Fractional Order . . . . .	14
4.2.2	Tuning . . . . .	16
4.3	Conclusion . . . . .	16
<b>5</b>	<b>TUNING USING COMPUTATIONAL INTELLIGENCE</b>	<b>17</b>
5.1	Introduction . . . . .	17
5.2	Computational Intelligence . . . . .	17
5.2.1	Techniques . . . . .	18
5.2.2	Fitness Assessment . . . . .	19
5.3	Genetic Algorithms . . . . .	20
5.3.1	Foundation . . . . .	21
5.3.2	Phases . . . . .	21
5.3.3	Advantages . . . . .	23
5.3.4	Limitations . . . . .	24
5.4	Particle Swarm Optimization . . . . .	24
5.4.1	Foundation . . . . .	24
5.4.2	Phases . . . . .	25
5.4.3	Advantages . . . . .	27
5.4.4	Limitations . . . . .	27
5.5	Conclusion . . . . .	27
<b>6</b>	<b>SIMULATION RESULTS</b>	<b>28</b>
6.1	Introduction . . . . .	28
6.2	Graphical Results . . . . .	28
6.2.1	Input Wind . . . . .	28
6.2.2	Open Loop Results . . . . .	30
6.2.3	Results Based on GA . . . . .	31
6.2.4	Results Based on PSO . . . . .	32
6.3	Analysis of result . . . . .	33
6.3.1	Open Loop Response . . . . .	33
6.3.2	Comparison Based on GA . . . . .	33
6.3.3	Comparison based on PSO . . . . .	34

6.4 Conclusion . . . . .	34
<b>7 CONCLUSION AND FUTURE SCOPE</b>	<b>35</b>
<b>REFERENCES</b>	<b>35</b>
<b>List of Publications</b>	<b>39</b>

# List of Tables

3.1	Parameters of dual mass wind turbine . . . . .	12
6.1	Comparison Based on GA . . . . .	33
6.2	Comparison Based on PSO . . . . .	34

# List of Figures

2.1	Illustration of a working wind turbine . . . . .	5
2.2	Optimum power curve . . . . .	6
2.3	Power coefficient at various pitch angles . . . . .	7
3.1	Dual mass model . . . . .	10
3.2	Schematic of dual mass system . . . . .	11
4.1	FOPID controller block diagram . . . . .	14
4.2	Controller FOPID converge . . . . .	15
4.3	General FOPID controller block . . . . .	15
5.1	Computational intelligence based tuning . . . . .	19
5.2	Flow chart of genetic algorithm . . . . .	22
5.3	Flow chart of particle swarm optimization . . . . .	26
6.1	Gust wind . . . . .	29
6.2	Base with step change wind . . . . .	29
6.3	Turbine speed without controller for gust wind . . . . .	30
6.4	Turbine speed without controller for base with step change wind . . . . .	30
6.5	Response for gust wind using GA . . . . .	31
6.6	Response for base with step change wind using GA . . . . .	31
6.7	Response for gust wind using PSO . . . . .	32
6.8	Response for base with step change wind using PSO . . . . .	32

# Abbreviations

CI	Computational Intelligence
FOPID	Fractional Order Proportional Integral Derivative
GA	Genetic Algorithm
ITAE	Integral of Time multiplied by Absolute Error
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization

# Notations

$B_{ls}$	Shaft damping coefficient, $Nm/rad/s$
$C_p$	Power coefficient
$J_g$	Generator inertia, $kg.m^2$
$J_t$	Turbine inertia, $kg.m^2$
$k_1, k_2$	Electromagnetic torque coefficients
$K_{ls}$	Shaft stiffness coefficient, $Nm/rad$
$\lambda$	Tip speed ratio
$n_g$	Gearbox ratio
$R$	Turbine radius, $m$
$\rho$	Air density, $kg/m^3$
$T_{em}$	Electromagnetic torque, $Nm$
$T_{hs}$	High speed shaft torque, $Nm$
$T_{ls}$	Low speed shaft torque, $Nm$

# Chapter 1

## INTRODUCTION

### 1.1 General Background

The constant rise in energy consumption, as well as the strong need to reduce the harmful consequences of climate change caused by global warming, have sparked a variety of research efforts in recent years to capture renewable energy sources. As a result of the major issues related to the rapidly depleting reserves and rising costs of fossil energy like oil, coal, and natural gas, emerging renewable energy applications like wind, hydro, solar, and biomass are growing and becoming more popular. Wind power is one of the most attractive clean and renewable sources of energy because of the rapid development of wind technology, as well as its simplicity and safety [1]. Designing a wind turbine now has optimization within the operational region and market context as its primary focus instead of following tradition.

Variable speed, gear less drive system and pitch control, are replacing fixed speed, gear boxed drive system and stall control in wind turbine design principles. On the other hand, due to their complexity, non-linearity, unpredictability, and unknown disturbances, wind turbines provide severe hurdles. The goal of ongoing study is to lower mechanical stress and boost energy efficiency [2]. Making greater use of the turbine's resources, extending the lifespan of its electrical and mechanical components, and generating higher returns are all possible solutions. One of them is to apply modernised controls that boost the turbine's performance. Because of the benefit of having controllers that can be used in a variety of wind turbine sizes, the dynamics of the systems that generate energy from wind turbines are frequently modelled as two-mass systems.

Changes in speed will be sent to the generator since the axes of the wind generator system are connected by shafts and a gearbox. This demonstrates the significance of dual mass wind turbine speed control [3] in reducing extreme overshoot and rotor fluctuations [4].

Over the past few decades, a variety of control tactics have been investigated; both conventional and artificial intelligence-based methods have made significant progress in current industry. Proportional-Integral-Derivative (PID) controllers are practical, offer good stability, and have a quick reaction, they are widely accepted and the most frequently used controller in industrial applications. To provide the best reaction and control, the coefficients of this controller that is proportional, integral and derivative actions are adjusted for each application [5]. For the objective of tuning PID controller parameters in the context of complex modifications, heuristic approaches have been presented. In order to get the most reliable performance of the systems, several scientists are currently using fractional order controllers widely [6]. The fractional order integro-differential operators are a part of the generalised fractional calculus, which is a progression from the common integer-order calculus.

## **1.2 Objectives**

The main goals of this thesis are:

1. To represent the wind turbine dynamically.
2. To implement a suitable wind turbine speed management system that lessens the effects of wind turbulence and excessive overshoot and permits greater performance.
3. MATLAB software will be used to test the methods for designing controllers based on computational intelligence in various wind profiles.

## **1.3 Organisation of the Report**

The report is organised as follows:

1. The background study of wind turbine components and its power curves are discussed in chapter 2.

2. Chapter 3 shows the dynamic modelling of wind turbine system.
3. Fractional controller design is done in chapter 4.
4. Chapter 5 constitute various computational intelligence approaches for tuning the controllers.
5. Simulation results are presented and discussed in chapter 6.
6. Chapter 7 discusses the conclusion and future scope of the work.

# Chapter 2

## LITERATURE REVIEW

### 2.1 Introduction

Various wind turbine parts are covered in this section. In order to comprehend the area where the wind turbine should operate to achieve optimal efficiency, an ideal power curve of a wind turbine is shown.

### 2.2 Basic Structure of Wind Turbine

A wind turbine consists of five main components and numerous auxiliary ones depicted in Figure 2.1. The tower, generator, rotor, nacelle and foundation make up the key components. The operation of a wind turbine is dependent on each of them.

The actual structure that houses the wind turbine is the tower. It holds up the rotor, nacelle, blades, and other parts of a wind turbine. Commercial wind towers typically range in length from 50 to 120 m and are made of reinforced steel or concrete. When operating with a wind that range from 3 to 15 m/s, blades are physical system that are aerodynamically tuned to help catch the most power from wind. Based on the level of power, each blade typically has a length of 20 m or more. The wind turbine generator, gearbox, and internal components are enclosed in the nacelle. It shields the internal turbine parts from the outside environment. The wind turbine's revolving component is known as the rotor. It transmits the wind's energy to the shaft. The gearbox and rotor hub are connected by the low-speed shaft, which also

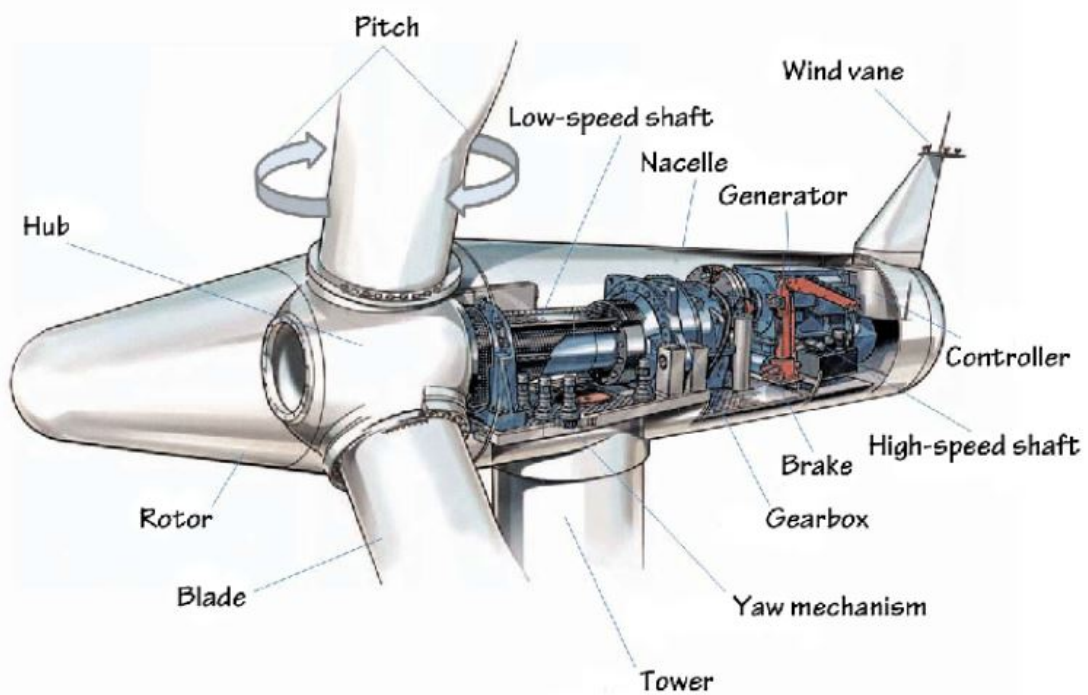


Figure 2.1: Illustration of a working wind turbine

secures the wind turbine blades. The mechanism for changing the rotor blade's angle of attack is known as pitch. Blades are rotated along their longitudinal axis to adjust the angle of attack in response to the wind directions. There are two sorts of shafts: high speed and low speed. Low-speed shafts transfer mechanical power from the rotor to the gearbox while the high-speed shafts transfer mechanical energy from the gearbox to the generator. Yaw is the turbine's horizontally moving component. To face the wind, it rotates either clockwise or counterclockwise. The yaw drive and yaw motor are the two primary components of the yaw. Whenever the wind direction changes, the yaw drive maintains the rotor towards the wind. The yaw is moved by the yaw motor.

The brake is a mechanical component attached to the high-speed shaft that lowers rotational speed, stops the wind turbine from over speeding, or stops it in an emergency. A mechanical device that is used to change the rotational speed called a gearbox. The gearbox in wind turbines is used to regulate the generator's rotational speed. The generator is responsible for transforming the rotor's mechanical energy into electrical energy. The most common types of generators are permanently magnet synchronous generators and doubly fed induction generators which is utilised in wind turbines. The controller is the head of the wind turbine. To maximise the ben-

efits of the wind, it continuously analyses the wind turbine’s health and regulates the pitch and yaw systems.

## 2.3 Ideal Power Curve

The wind turbine machine’s mechanical and electrical components can be distinguished. The aeroturbine, gearbox, slow shaft, and fast shaft are the mechanical components of the wind turbine machine. The behaviour of a wind turbine’s aerodynamics is its most crucial feature. The conversion of wind energy into kinetic energy is explained by aerodynamic behavior. Transmission shafts carry this energy to the generator, where it is converted to electric energy. Understanding the connection between wind speed and power is crucial for selecting the right kind of regulation, optimization, or constraint. The power curve [7] shows how much energy can be extracted from the dominant wind. Wind turbine’s optimal power curve is depicted in Figure 2.2.

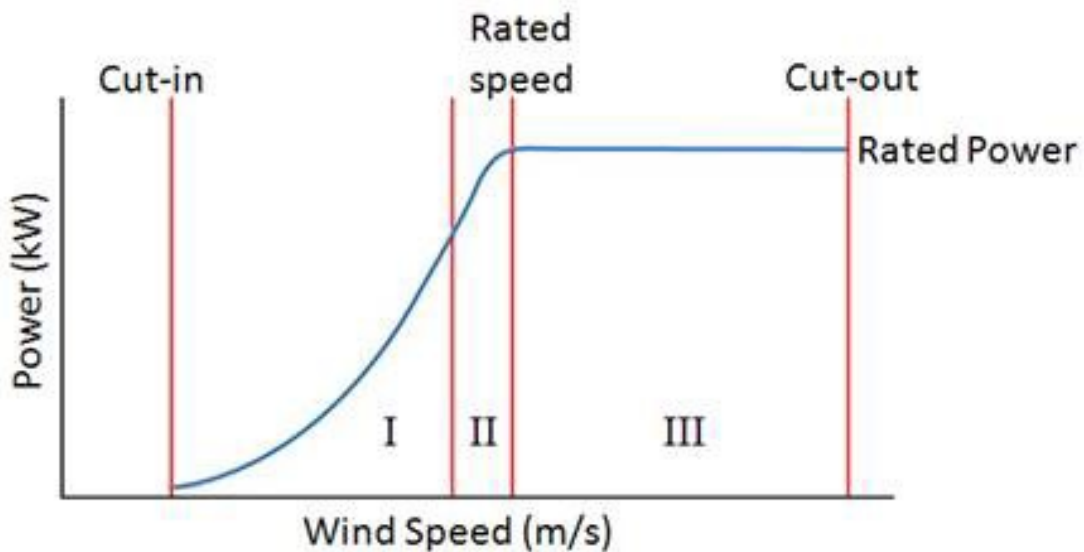


Figure 2.2: Optimum power curve

The cut-in and cut-out speeds define the turbine’s operational boundaries. By staying within this range, it is made sure that the amount of energy that is available is higher than the threshold that is necessary to preserve structural health. Cost and energy are taken into account at the manufacturer’s point, also referred to as the rated power. Additionally, the estimated wind speed was chosen since wind speeds exceeding this point are uncommon.

The turbine in Region I is operated at its maximum efficiency to harness all the electricity because of the low wind speeds and output power levels that fall just below rated turbine capacity. In other words, the turbine controls are made to be as effective as possible. The reduction of rotor torque and disturbance is a major priority in Region II, the last transitional zone. In contrast, Region III experiences strong winds and is using all of its available turbine power. The turbine then regulates when working in this region by taking into account the power output. The electric system of the wind turbine has a much faster temporal responsiveness than the other parts. By separating the control designs for the generator and turbine sides, this enables the use of two control loops (aeroturbines). While the outside control loop manages the turbine side to force the rotor speed to its ideal value, the inside control loop controls the electric generator through the converters.

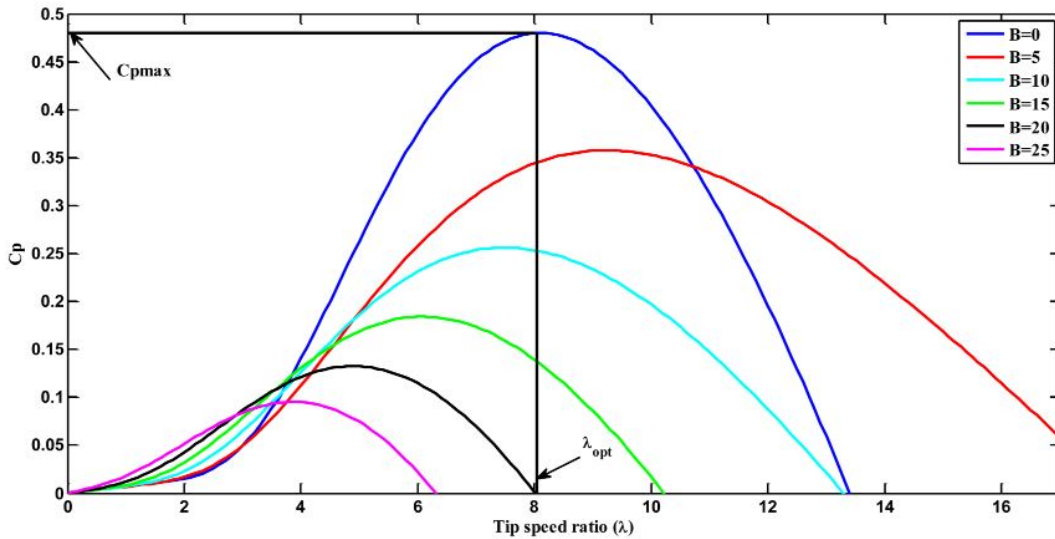


Figure 2.3: Power coefficient at various pitch angles

In this work, the electrical loop is assumed to be well-controlled, and only the outside control loop will be considered. Under the rated wind speed, the mechanical power need to be increased. The wind turbine power coefficient [8], or  $C_p$ , is computed as:

$$C_p(\lambda, \beta) = c_6\lambda + c_1 \left( \frac{c_2}{\lambda_i} - c_3\beta - c_4 \right) e^{-\frac{c_5}{\lambda_i}} \quad (2.1)$$

Here  $c_1, c_2, c_3, c_4, c_5$  and  $c_6$  is given by 0.5176, 116, 0.4, 5, 21 and 0.0068 respectively. For an optimum pitch angle of  $0^\circ$  and a tip speed ratio of 8.1, the power coefficient is at its highest

value at  $C_{p\max}(\lambda_{opt}, \beta_{opt}) = C_{opt} = 0.48$ . The turbine operates at its highest efficient level at this value, producing the maximum energy.

## **2.4 Conclusion**

The essential parts of a wind turbine system are explained in this chapter, as well as the power curve, which shows how much electricity a wind turbine should generate at any given wind speed. The next chapter displays a dual mass wind turbine system's detailed modelling.

# Chapter 3

## WIND TURBINE DYNAMICS

### 3.1 Introduction

This chapter covers how a dual mass model is commonly used to describe the mechanics of wind turbines. Models of the mechanical and dynamic drive trains are provided.

### 3.2 Modelling

The mechanics of wind turbines are typically described by a dual mass model [9]. A dual mass model is being used to generate controllers since the control rules it generates are more general and may be applied to wind turbines of different sizes. These controllers work particularly well with variable wind turbines that are difficult to adequately model using a one mass model [10].

The formula below can be used to calculate the mechanical power [8] produced by a wind turbine:

$$P_t = \frac{1}{2} \rho \pi R^2 C_p(\lambda, \beta) v^3 \quad (3.1)$$

Here density, radius, power coefficient and wind speed are given by  $\rho$ ,  $R$ ,  $C_p(\lambda, \beta)$  and  $v$  respectively. It is possible to estimate the power coefficient  $C_p$  of wind turbine blade efficiency by using the method [8] represented as:

$$C_p(\lambda, \beta) = 0.0068\lambda + 0.5176 \left( \frac{116}{\lambda_i} - 0.4\beta - 5 \right) e^{-\frac{21}{\lambda_i}} \quad (3.2)$$

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1} \quad (3.3)$$

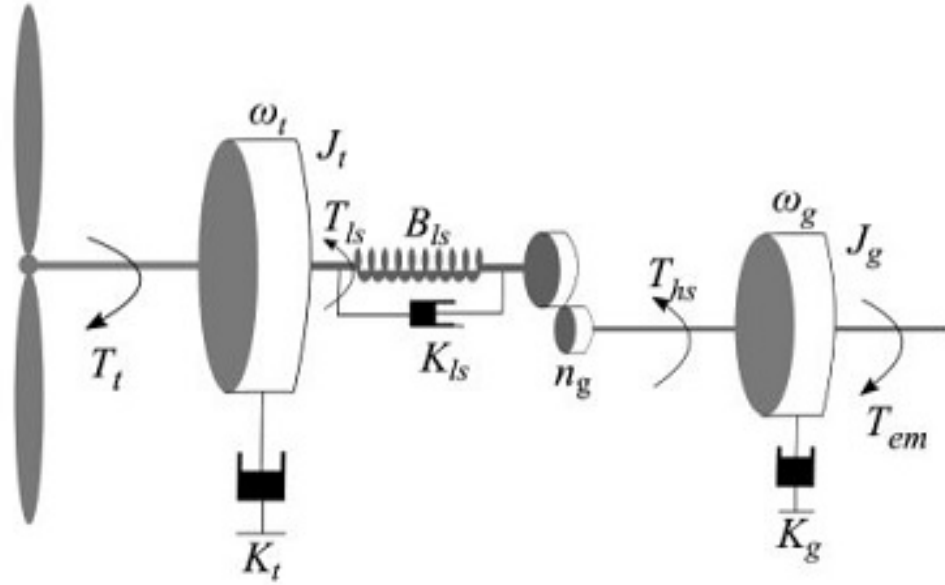


Figure 3.1: Dual mass model

The effectiveness of the rotor blades is influenced by the tip speed ratio  $\lambda$  and the blade pitch angle  $\beta$ . The parameters  $\lambda$  and  $\beta$  affect how well the system works. The formula for the tip speed ratio is

$$\lambda = \frac{\omega_t R}{v} \quad (3.4)$$

Where the wind turbine shaft's angular speed is  $\omega_t$ . The wind turbine's  $T_t$  torque [11] is determined by,

$$T_t = \frac{P_t}{\omega_t} = \frac{\rho \pi R^3 v^2 C_p(\lambda, \beta)}{2\lambda} \quad (3.5)$$

The essential parts of a wind turbine with variable speed are a generator, a gearbox, and an aeroturbine. The vitality of the rotor is rotated by the wind around two or three propellers. Due to the primary shaft, which in turn is attached to the rotor, the high speed shaft transfers aerodynamic power to the gearbox, which then drives the generator.

Most often, a dual mass model is used to simulate the dynamics of a wind turbine as in Figure 3.1. The two-mass wind turbine type is distinguished by the benefit of their controllers generic design, which makes them useful for wind turbines of various sizes.

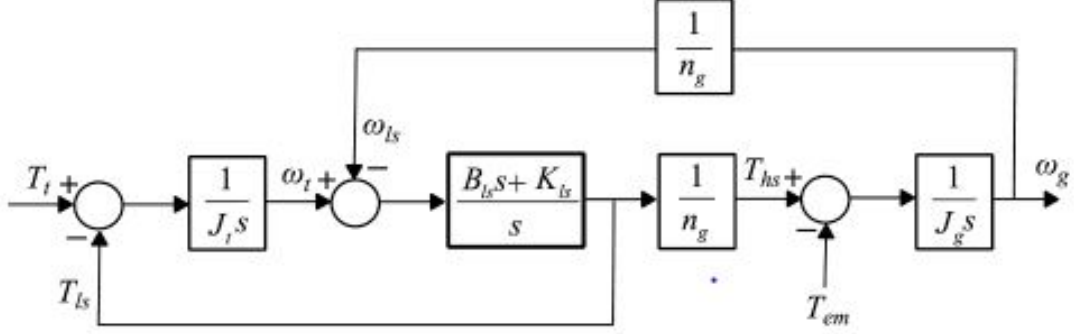


Figure 3.2: Schematic of dual mass system

The model of a dual mass wind turbine is depicted in Figure 3.2, where  $J_t$  stands for the side inertia of the turbine driven at speed  $\omega_t$  using a torque  $T_t$  which is given as follows:

$$J_t \frac{d\omega_t}{dt} = T_t - T_{ls} - K_t \omega_t \quad (3.6)$$

External dampening of the wind turbine is provided by  $K_t$  and  $T_{ls}$  denotes the low speed shaft torque as seen in the following:

$$T_{ls} = K_{ls} (\theta_t - \theta_{ls}) + B_{ls} (\omega_t - \omega_{ls}) \quad (3.7)$$

Low speed shaft stiffness and damping coefficients are represented by  $K_{ls}$  and  $B_{ls}$ , respectively. The low speed shaft is denoted by  $\omega_{ls}$ , and the angular deviations of the wind turbine and low speed shaft are represented by  $\theta_t$  and  $\theta_{ls}$ , respectively. The electromagnetic torque  $T_{em}$  brakes the high speed shaft's torque  $T_{hs}$ , which in turn causes the generator's inertia  $J_g$  to produce the following:

$$J_g \frac{d\omega_g}{dt} = T_{hs} - T_{em} - K_g \omega_g \quad (3.8)$$

Table 3.1: Parameters of dual mass wind turbine

Parameter	Value
Air density	1.29 kg/m <sup>3</sup>
Turbine radius	21.62 m
Gearbox ratio	43.165
Turbine inertia	325000 kg.m <sup>2</sup>
Generator inertia	34.4 kg.m <sup>2</sup>
Shaft stiffness coefficient	269100 Nm/rad
Shaft damping coefficient	9500 Nm/rad/s
Power coefficient	0.48
Tip speed ratio	8.1
Electromagnetic torque coefficients	$k_1=0.1082, k_2=0.2147$

$$T_{em} = K_1\omega_g^2 - K_2\omega_g \quad (3.9)$$

Where  $K_g$  and  $K_1, K_2$  represent the external damping of a generator and electromagnetic torque coefficients. The transmission ratio of the optimal gearbox is given by

$$n_g = \frac{T_{ls}}{T_{hs}} = \frac{\omega_g}{\omega_{ls}} = \frac{\theta_g}{\theta_{ls}} \quad (3.10)$$

The transfer function of the dual mass wind turbine system from Figure 3.2 is given by,

$$G_t(s) = \frac{\omega_t(s)}{T_t(s)} = \frac{n_g^2 J_g s^2 + B_{ls} s + K_{ls}}{n_g^2 J_t J_g s^3 + B_{ls} (J_t + n_g^2 J_g) s^2 + K_{ls} (J_t + n_g^2 J_g) s} \quad (3.11)$$

### 3.3 Conclusion

This chapter covered the mechanical and dynamic modelling of wind turbines. The design parameters for a dual mass wind turbine are given. The following chapter explores fractional order controllers.

# Chapter 4

## DESIGN OF CONTROLLER

### 4.1 Introduction

A fractional controller is suggested in this chapter for the speed control of a dual mass wind turbine. The fundamental benefit of a fractional-order controller would be that it gives the control system more flexible time and frequency responses, enabling better and more reliable performance.

### 4.2 Fractional Calculus

The advanced fractional calculus, an extension of the conventional integer-order calculus, includes the integro-differential processes of fractional orders. Recent years have seen a lot of interest in fractional-order calculus since so many processes in physics, chemistry, and engineering display a memory effect and are best described by fractional-order dynamics [12]. The precise and quantitative answers of fractional-order integro-differential equation have received a great deal of attention as a result of the rise in system applications.

The behaviour of fractional-order systems can be modified using integer or fractional-order control techniques. In many cases, it has been demonstrated that fractional-order controllers perform better than their integer-order equivalents [13]. The Proportional-Integral-Derivative (PID) controller is one such controller that has been successfully used in industrial applications for many years. The PID controller's attractiveness is due to how simple the design procedures are and how well it performs. On the other hand, a Fractional-Order PID (FOPID) controller

added two new parameters to the integral and derivative components of the conventional PID controller, making it more challenging to tune these parameters.

### 4.2.1 PID Controller with Fractional Order

For many years, industries have used PID controllers for control systems. They have a simple design, which contributes to their widespread appeal, and outstanding performance, which includes little percentage overshoot and a quick settling period for slower process plants as shown in Figure 4.1. The traditional PID controller is expanded upon by a fractional PID controller. Changes in a controlled system's and controller's parameters are less noticeable to fractional order controllers. Iso-damping is a fairly simple feature that can be attained with a fractional order controller. This fractional PID controller's simplified transfer function is given by:

$$C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s^\lambda} + K_D s^\mu, (\lambda, \mu \geq 0) \quad (4.1)$$

Here  $C(s)$  is the controller output. The control signal is represented by  $U(s)$ , the error signal by  $E(s)$ , the proportional gain by  $K_P$ , the integration gain by  $K_I$ , the derivative gain by  $K_D$ , the order of integration by  $\lambda$ , and the differentiator order by  $\mu$ .

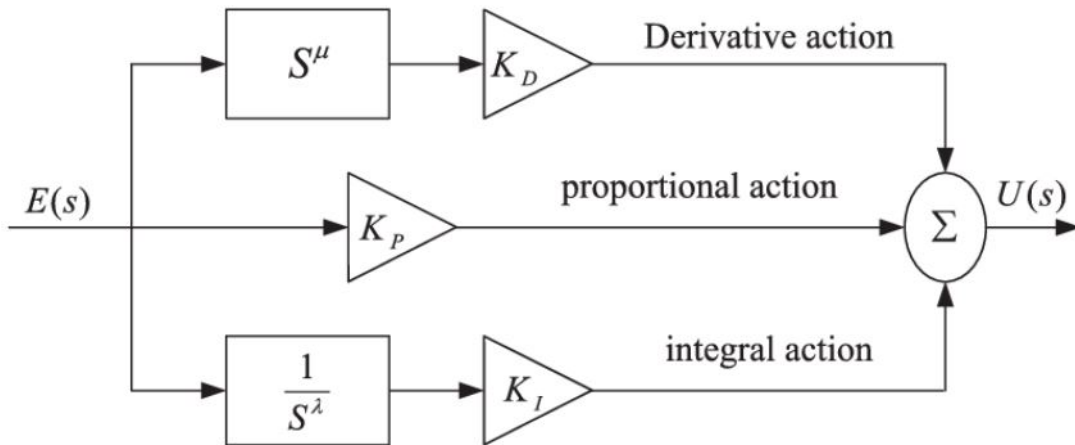


Figure 4.1: FOPID controller block diagram

Referring to Figure 4.2, all classical PID controllers are certain kinds of fractional controllers, when  $\mu$  and  $\lambda$  are both 1. In terms of the PID plane, this implies that one can move constantly

within it as opposed to hopping between its four fixed locations. The range of fractional order often taken into account is 0 to 2.

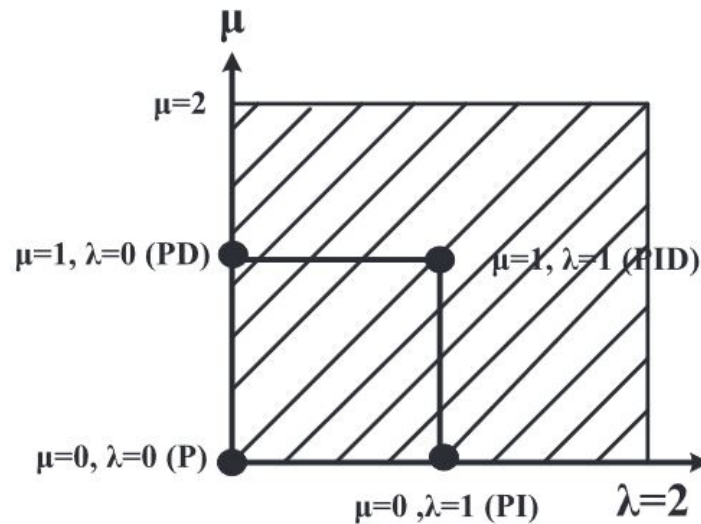


Figure 4.2: Controller FOPID converge

The general design of a closed loop system using a fractional order PID controller is shown in Figure 4.3. The effectiveness of the entire system can be increased since the fractional controllers have two more parameters than the traditional PID controllers, allowing for the fulfilment of two more standards.

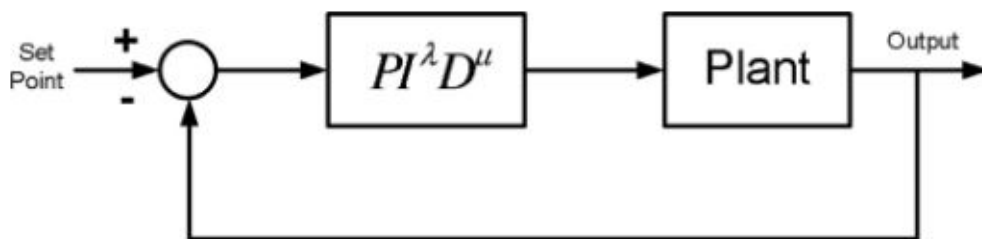


Figure 4.3: General FOPID controller block

The following situations/applications benefit more from the use of a fractional PID controller:

1. It is possible to obtain five different specifications, which is not possible with a traditional PID controller.
2. Comparatively to the conventional PID controller, the FOPID may easily achieve the iso-damping property.

3. The performance of a classical PID controller degrades for higher order systems, while a fractional PID controller performs better.
4. FOPID performs better than a classical PID controller for a system with a long time delay.
5. One fractional order PID controller is typically sufficient for non-linear systems, although non-linear systems are typically linearized at various operational points, and controllers are built for different operating points.

### 4.2.2 Tuning

The majority of controllers tuning is a difficult process to accomplish. There are more parameters that must be tuned when developing a fractional PID controller. It is difficult to optimize the controller because of this [14].

1. **Based on rules:** One of the first strategies for tuning PID controllers was the Ziegler-Nichols method. It typically enhances the control performance, but it is always a possibility to use it to select the first tuning point. Step responses in the S-shaped curve are necessary for this procedure.
2. **Analysis method:** This approach's effectiveness was compared to that of the conventional PID controller. Equations that were found for the desired requirements were solved to acquire the parameters of this controller. Additionally, a comparison of the step input responses of the heating furnace's integer and fractional-order models was carried out.
3. **Numerical approach:** The adjustment of fractional controllers using various optimization techniques like PSO, GA etc. Different performance indices, including overshoot, rising time, settling time, and control input, have been taken into consideration for this tuning procedure.

## 4.3 Conclusion

In this chapter, fractional calculus and PID controllers with fractional orders were covered. Since tuning controllers is challenging, several ways are suggested. The following chapter will examine computational tuning techniques.

# Chapter 5

## TUNING USING COMPUTATIONAL INTELLIGENCE

### 5.1 Introduction

The approaches of computational intelligence that enable automatic decision-making and effective complicated problem resolution are covered in this chapter. The details of computational intelligence methods such as the genetic algorithm and particle swarm optimization are provided.

### 5.2 Computational Intelligence

Soft computing is another name for computational intelligence, a field of research where the goal is to teach computers how to solve complicated or real-world issues using data from experiments or other observations. A collection of strategies or approaches are employed in computational intelligence to tackle complicated or real-world issues. Because of their complexity, uncertainty, or lack of a clear description, real-world problems are typically too difficult or unpredictable to be solved using conventional computing techniques [15].

The possibility for intelligent, probabilistic decisions and the resolution of complicated issues in the actual world exists in computational intelligence. The organisation of industry processes, disease diagnosis, video game visualisation, intelligent robotics, intelligent chatbots, autonomous cars, and translational systems like Alexa and Siri that really can interpret human

language and function accordingly all benefit greatly from CI. The theories, perspectives, algorithms, and implementations of computational intelligence enable or assist proper behaviour (smart behaviour) in complex and dynamic situations.

### 5.2.1 Techniques

1. **Fuzzy logic:** This method frequently works in a variety of areas, including control, image processing, and decision-making. However, it is also extensively used in the sector of home appliances, including washing machines and microwaves. Fuzzy logic lacks learning capacities, a skill that humans are highly valued for having and is only good for approximative thinking. They are able to become better versions of themselves by reflecting on their past errors.
2. **Neural networks:** On the basis of biological neural networks, which can be divided into three basic parts: the cell body, which processes data, the axon, which is a mechanism enabling the conduct of signals, as well as the synapse, which regulates signals, CI professionals work to construct artificial neural networks. As a result, distributed data processing systems are dotted with artificial neural networks, managing the process and the training from experiential data. One of the key advantages of this theory is that it functions like humans do, which is fault tolerance.
3. **Evolutionary computation:** The evolutionary computation, which is based on the natural selection process, involves using the power of natural evolution to develop new artificial evolutionary approaches. It also covers other topics like problem-solving evolutionary algorithms and evolutionary strategy. The principal applications of this theory include fields like optimizing and multi-objective optimization, where conventional mathematical one-step techniques are no longer sufficient to solve a variety of issues.
4. **Learning theory:** Learning theory describes the concept and method of learning and describes how a learner puts information together, processes it, and retains it in order to make wise decisions about difficult situations. Theory of learning focuses on various ways in which the learning might occur and aims to examine and comprehend various learning strategies.

5. **Probabilistic methods:** These are non-constructive and non-deterministic procedures, are extensively used to establish an object's existence.

The difference between reference operating point and the actual speed is sent to the computationally tuned fractional order PID controller as given in Figure 5.1.

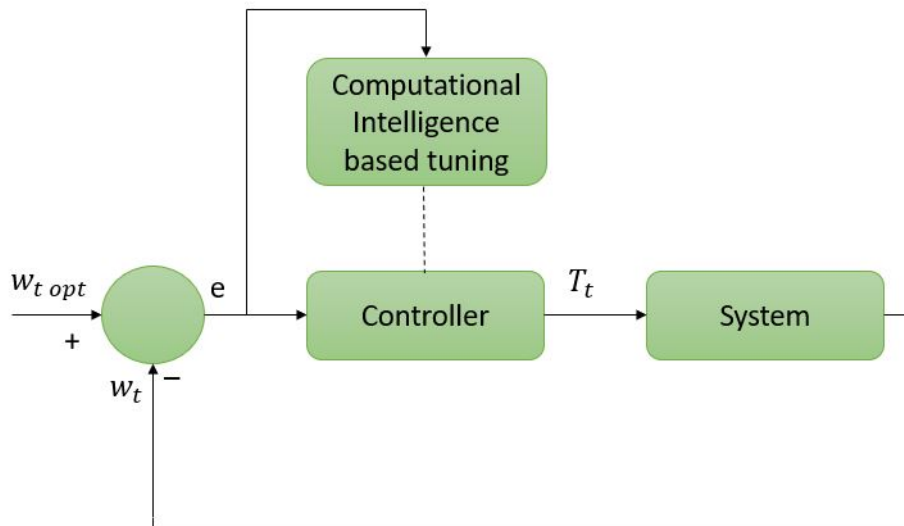


Figure 5.1: Computational intelligence based tuning

### 5.2.2 Fitness Assessment

The most important step in assessing computational intelligence algorithms is choosing the objective function which will be used to judge each agent's fitness. Integral of Time multiplied by Absolute Error given by equation (5.1) is the objective function that is used in this literature to minimize the error signal between the reference and the measured output.

$$ITAE = \int_0^{\tau} |error(t)|t dt \quad (5.1)$$

When a time-domain problem is indicated by error(t). A fitter an individual is, the smaller its fitness value will be and vice versa. The goal of the fitness function is to reduce the value of the selected objective function.

## 5.3 Genetic Algorithms

Population of potential solutions to an optimization issue is developed toward better solutions in a genetic algorithm. Solutions are generally expressed in binaries as strings of 0s and 1s, although other encodings are also feasible. Each optimal solution has a set of characteristics (its chromosomes or genotype) that can be modified and updated [16].

A generation is the term used to describe the population in each iteration of the evolution, which typically begins with a community of randomly generated individuals. Every member of the population has their fitness assessed once every generation; the fitness is typically the magnitude of the objective function in the optimization method being addressed. A new generation is created by stochastically selecting the fittest people from the existing population, recombining their genomes, and maybe introducing genetic changes. The following algorithm iteration uses the fresh generation of candidate solutions. When the population reaches a desired fitness level or the algorithm has produced the maximum repetitions, the process normally ends. In a conventional genetic algorithm, the solution domain must be genetically represented and evaluated using the fitness function.

Each potential answer is typically represented as an arrays of bits. The use of arrays of various types and structures is fundamentally the same. The fundamental benefit of these genetic representations is that because of their fixed size, their pieces are simple to align and allow for straightforward crossover procedures. It is also possible to employ representations of variable length, but crossover execution is more difficult in this situation.

In gene expression programming, a combination of the both linear chromosome and trees are examined. Genetic programming explores representations in the form of trees, while evolutionary programming explores representations in the form of graphs. GA initiates a population of solutions after defining the genetic representations and the fitness value, and then improves it by repeatedly using the mutation, crossover, inversion, and selection operations.

### 5.3.1 Foundation

The population's chromosome genetic structure and behaviour serve as the foundation for the genetic algorithm. Each chromosome represents a potential answer. Consequently, the population is a group of chromosomes. Each person in the population has a fitness function that describes them. The answer is therefore to get more fit. The population's best individuals are chosen to reproduce the children of the following generation. The result of a mutation, the offspring will have traits from both parents. A mutation is a minor alteration to the gene's structure.

### 5.3.2 Phases

1. **Population Initialization:** The population is the first group of people in the process. Everyone is the answer to the issue. Genes are a collection of parameters that define an individual. The gene is combined with a wire to form a chromosome [17].
2. **Fitness Function:** The ability to compete with other people's people is determined by a person's fitness level, or function. It assigns a fitness rating to each person. Depending on his fitness score, a person may be chosen for reproduction.
3. **Selection Process:** The purpose of the selection phase is to choose the best candidates so that they can pass on their genes to the following generation. The fitness score of the parents determines which of two sets of parents are chosen. High fitness individuals are more likely to be chosen for reproduction. Boltzman selection, tournament selection, roulette wheel selection, rank selection are some methods for selection.
4. **Crossover:** The importance of crossover in genetic algorithms is crucial. A crossover point is chosen at random from the DNA for each set of parents. Single point, two point and uniform are different types of crossovers [18].

One point crossover is a random location where both parents chromosome is chosen and referred to as the "crossover spot". The different parent chromosomes switch bit to the right of the that location. Two children are born as a result, and each one carries some genetic material from both parents.

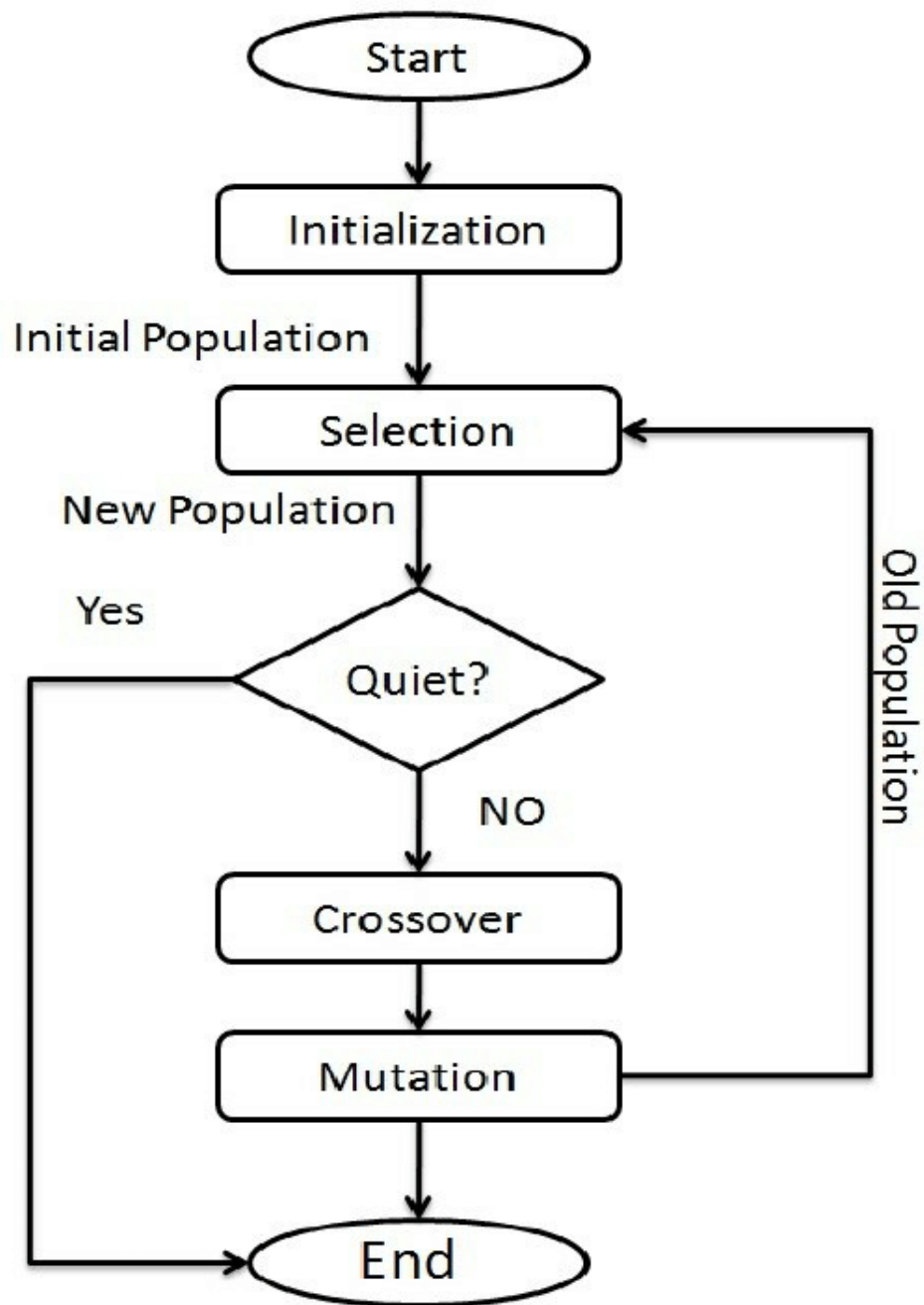


Figure 5.2: Flow chart of genetic algorithm

Two point crossover is a particular application of the N-point Crossover method. The genetic information is exchanged at two randomly selected locations along the individual chromosomes (strings).

Each gene (bit) is randomly chosen from one of the parent chromosome's corresponding genes for uniform crossover. The flow chart for GA is given in Figure 5.1.

5. **Mutation:** A community of evolutionary algorithm chromosomes uses the genetic operator of mutation to preserve genetic variety from one generation to the next. Similar to biological mutation, it is comparable.

The probability that each bit in a DNA sequence will alter from its starting state is included in the typical illustration of a mutation operator. It is normal practice to implement the mutation operator by assigning a random value to each bit in a sequence. This random variable predicts whether a specific bit will be flipped or not. Single point mutation is the name of the mutation process that is based on biological point mutations. Reversal and float point mutation are additional types. Switches, reversals, and scrambles are the mutations that occur so when gene encoding is restricted, as in permutation issues.

The goal of mutation in GAs is to provide diversity to the population that was sampled. By stopping the populations of chromosome from becoming overly similar to one another and so delaying or even blocking converging to the global optimum, mutation operators are employed to try to avoid local minima. This logic also influences many GA systems to choose a randomized (or semi-random) group with a weight toward the fitter individuals rather than just selecting the population's fittest individuals to create the subsequent generation.

6. **Termination:** If the population changes, the algorithm stops working. The genetic algorithm is thus said to have provided a solution for the problems [19].

### 5.3.3 Advantages

Genetic algorithms have the best parallel capabilities. It aids in the optimization of a variety of issues, including continuous functions, multi-objective issues, and discrete function issues. It offers a remedy to a problem that becomes better with time. Derivative knowledge is not necessary for a genetic algorithm.

### **5.3.4 Limitations**

For simple issue solving, genetic algorithms are ineffective. It does not ensure the calibre of the problem's ultimate resolution. Calculating fitness values repeatedly may present some computing difficulties.

## **5.4 Particle Swarm Optimization**

Particle swarm optimization (PSO) is a computer technique that seeks to solve a problem more effectively by repeatedly attempting to make a candidate solution better in terms of a specified quality metric. By using a population of potential solutions, here referred to as particles, and moving them across the search space in accordance with a straightforward math equation so over particle's position and velocity, it solves problems [20]. In addition to being led towards to the better remembered positions in the search area, which are updated as other particles find better positions, every particle's movement is also impacted by its own best known position. The swarm should move toward the better answers as a result of this.

A population-based search approach that mimics the social behaviour of flocks of birds. In order to uncover the patterns that govern how well birds can fly in unison, change directions abruptly, and recover in an ideal arrangement, the primary purpose of the swarm based concept was to visually reflect the lovely and startling motions of a flock of birds. From this basic goal, the idea developed into a straight forward and effective optimization procedure.

### **5.4.1 Foundation**

This concept was inspired by research on social connection and animal behaviours like fish schooling and bird flocking. The collaboration and rivalry among the entire population reflects how they get food. The several unknown parameters which need to be improved are represented by each of the particles that make up a swarm. A population of randomly selected solutions is used to initialise the "swarm". Particles move throughout a PSO system in a multidimensional search space, altering their position based on their own and their neighbour's experiences. With the idea of finding better solutions along the way and ultimately settling on a single minimal or maximum solution, the objective is to efficiently search the solution space by directing the

particles towards the best-fitting solution found in earlier iterations. The effectiveness of each particle is assessed using a fitness function that has been predetermined and is related to the nature of the problem. Many of the most recent papers in this sector have mentioned the use of PSO. Its simplicity, cheap computing cost, and effective performance have led to its reputation as a promising optimization approach.

### 5.4.2 Phases

The crucial terms are particle  $i$  represent any possible solution, population  $N$  group of all particles, search space  $[a, b]$  which has all possible solution to the problem [5]. It is assumed that each particle has the following two properties: A position  $x_i$  and velocity  $v_i$ . Each particle keep track for  $p_{\text{best}}$  (personal best) and  $g_{\text{best}}$  (global best). The phases considered is:

1. **Initial Population:** Having initial population  $x$  and initial velocity  $v$ , there are  $N$  numbers of particles. In accordance with the objective function, assign  $p_{\text{best}}$  and  $g_{\text{best}}$ .
2. **Update:** Each particle's position and velocity are [21] updated which is given by,

$$v_i(t) = c_1 r_1 (p_{\text{best},i} - x_i(t-1)) + \theta v_i(t-1) + c_2 r_2 (g_{\text{best}} - x_i(t-1)) \quad (5.2)$$

here  $\theta v_i(t-1)$  ensure that the particle doesn't abruptly change direction. The particle's optimum position on the track is created by  $c_1 r_1 (p_{\text{best},i} - x_i(t-1))$ , particle follow the best location that the group has determined according to the term  $c_2 r_2 (g_{\text{best}} - x_i(t-1))$ . Inertia weight  $\theta$  is considered to vary linearly from 0.9 to 0.4 and is a proportional agent that correlates with the pace of the most recent improvement [22]. The rates of cognitive (individual) and social (group) learning are  $c_1$  and  $c_2$ . The random numbers  $r_1$  and  $r_2$  have a uniform distribution and fall between 0 and 1. The position is represented by,

$$x_i(t) = v_i(t) + x_i(t-1) \quad (5.3)$$

$$\text{If } x_i(t) > x_{ub} \rightarrow x_i(t) = x_{ub} \quad (5.4)$$

$$\text{If } x_i(t) < x_{lb} \rightarrow x_i(t) = x_{lb} \quad (5.5)$$

The  $p_{\text{best},i}$  and  $g_{\text{best}}$  are updated.

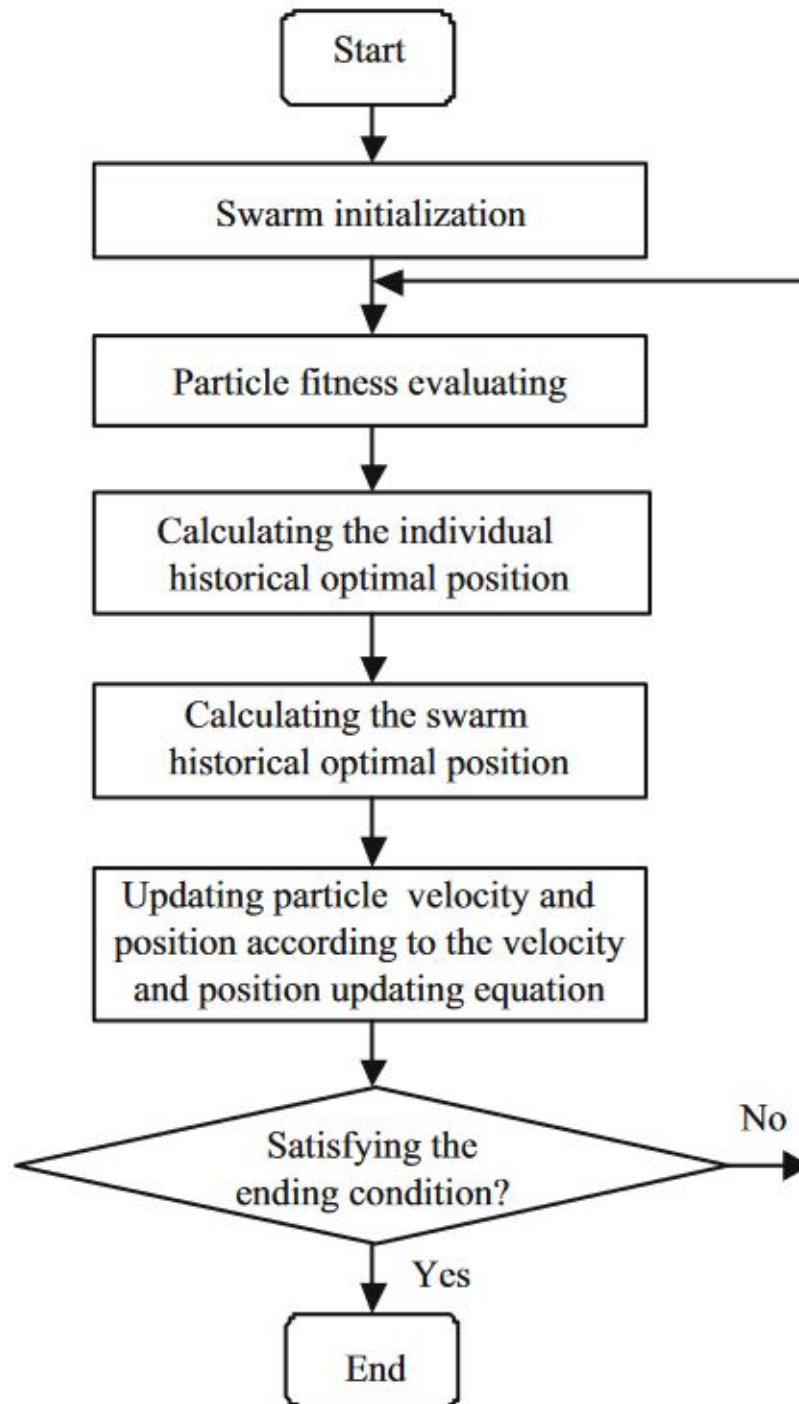


Figure 5.3: Flow chart of particle swarm optimization

3. **Termination:** The PSO algorithm's phases are iteratively repeated until the allotted number of iterations is reached or a cutoff criteria is met. Convergence is the assumption that the procedure has converged occurs when all particle positions converge to the very same set of values. The flow chart is given in Figure 5.2.

### **5.4.3 Advantages**

The mathematical PSO is built on intelligence. It can be used in engineering as well as scientific research. PSO do not calculate mutations or overlap. The particle's speed can be used to conduct the search. Only the most optimistic particle may pass knowledge to the other particles over the evolution of several generations, and research moves quite quickly. The PSO computation is pretty straightforward. When compared to other development calculations, it takes up more optimization space and is simpler to finish. PSO uses a real numeric code, and the decision is made immediately by the answer. The dimension's number is the same as the solution's constant.

### **5.4.4 Limitations**

The method is unable to solve the scattering and the approach is incapable of solving non-coordinate system issues, such as how to deal with the energy field and how the particles move within the energy field.

## **5.5 Conclusion**

In-depth discussions of the genetic algorithm and particle swarm optimization are covered in this chapter, which covered the fundamentals of computational tuning approaches. The results are displayed in the following chapter.

# Chapter 6

## SIMULATION RESULTS

### 6.1 Introduction

This chapter analyses the findings for variable speed wind turbine systems utilising various wind profiles and presents the graphical results.

### 6.2 Graphical Results

The dual mass wind turbine system is simulated using the Matlab/Simulink environment, and Table 3.1 lists the key parameters for system simulation. To evaluate the efficiency and performance of the suggested FOPID controller, the dynamic behaviour of the dual mass system is analyzed under different wind speed profiles given as base with step change and gust wind.

#### 6.2.1 Input Wind

The gust wind speed is depicted in Figure 6.1, varies over time. The controller's tolerance to sudden variations in wind speed is tested using the stated gust model, to check controller's resilience to rapid variations in wind. The base with step change in wind speed are depicted in Figure 6.2. In order for a wind turbine to operate, the base wind must always exist. The fundamental wind speed typically remains constant with time, hence it can be said to be invariable and has a value of 10 m/s. A step-down from 10 to 8 m/s at 0.5 seconds and a step-up from 8 to 10 m/s at 1 seconds may be seen as the step change in the wind's profile.

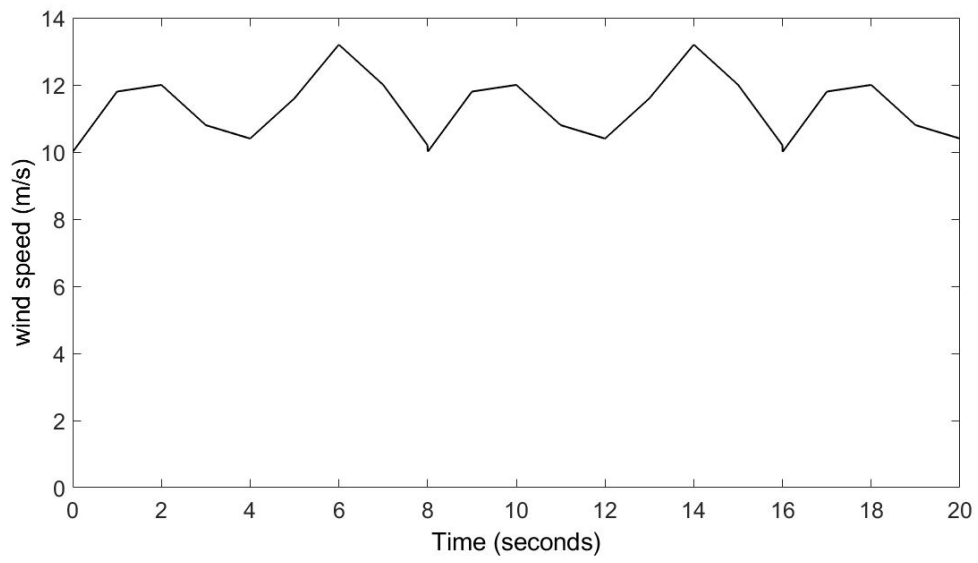


Figure 6.1: Gust wind

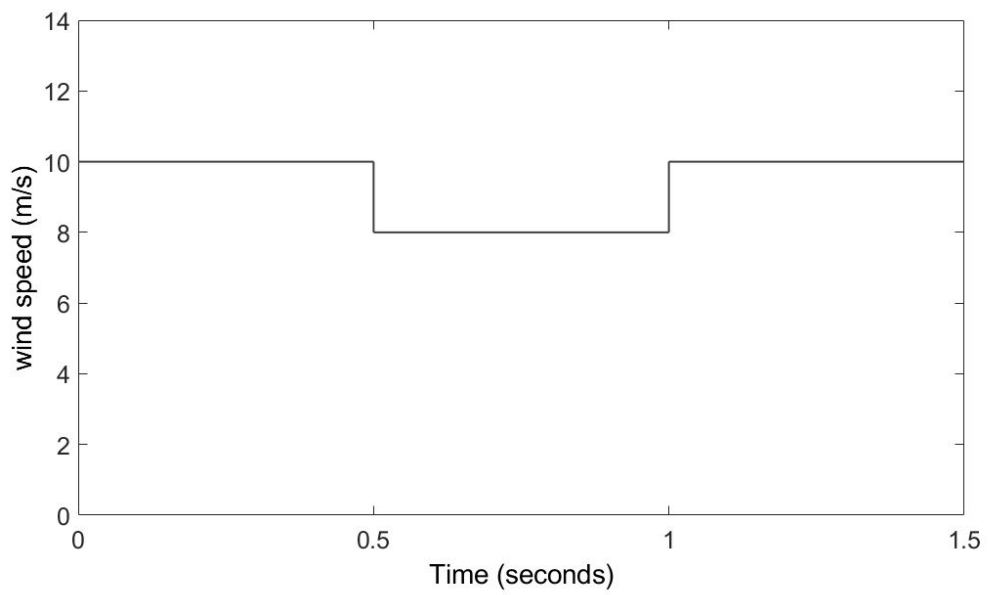


Figure 6.2: Base with step change wind

## 6.2.2 Open Loop Results

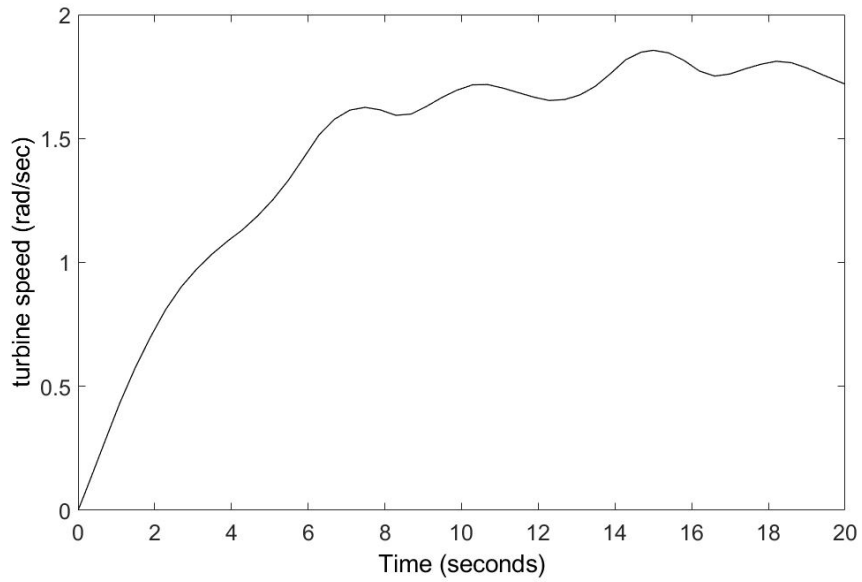


Figure 6.3: Turbine speed without controller for gust wind

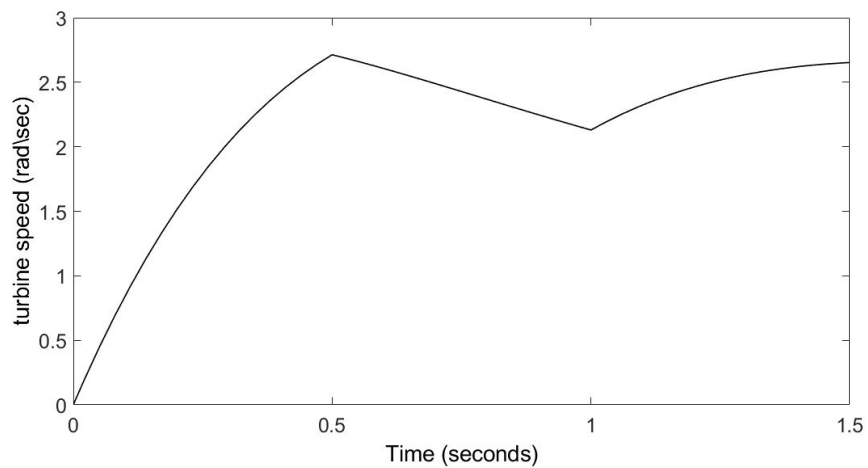


Figure 6.4: Turbine speed without controller for base with step change wind

### 6.2.3 Results Based on GA

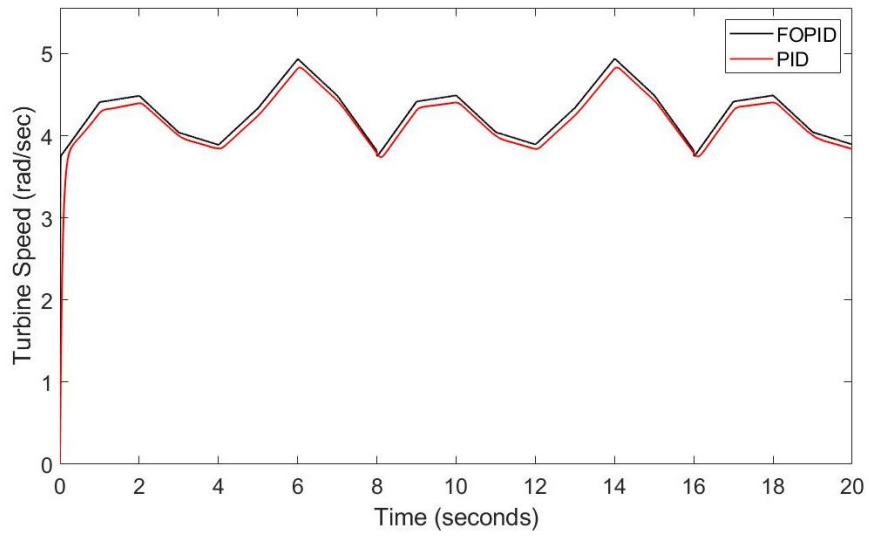


Figure 6.5: Response for gust wind using GA

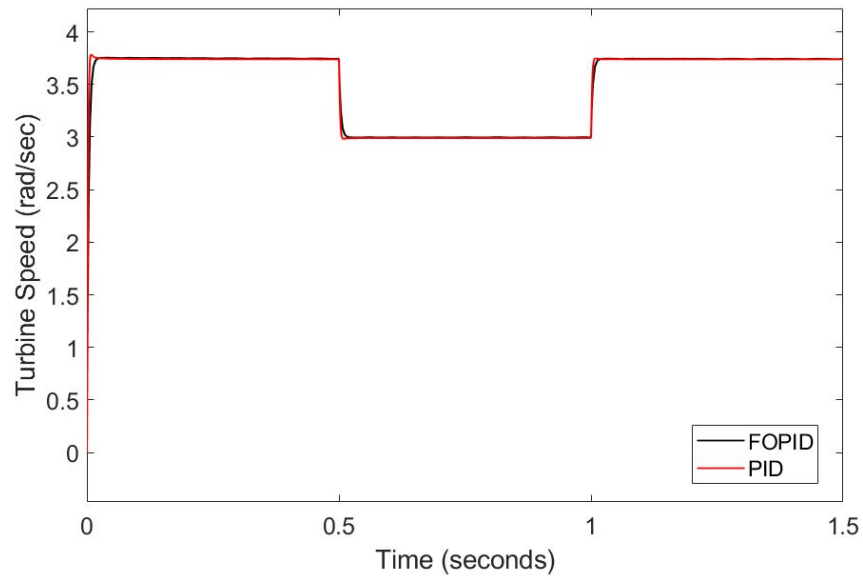


Figure 6.6: Response for base with step change wind using GA

## 6.2.4 Results Based on PSO

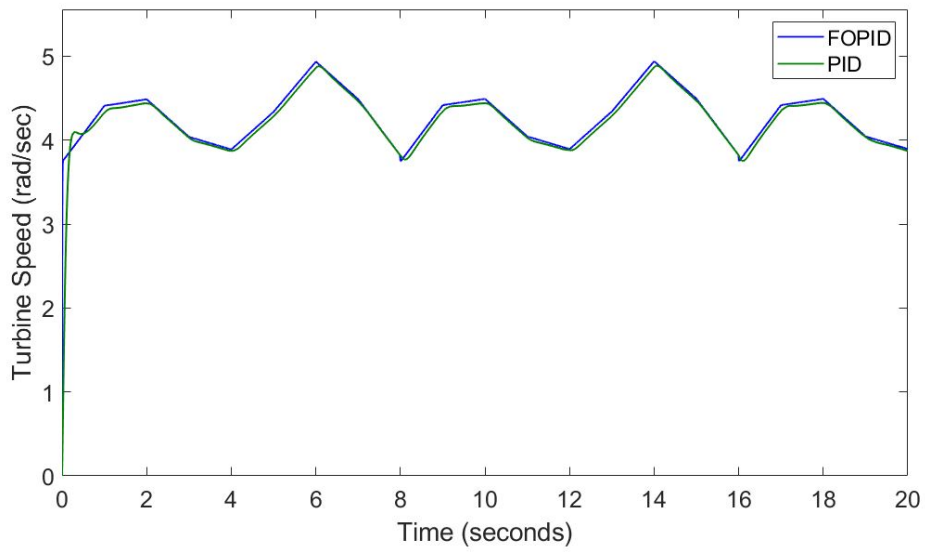


Figure 6.7: Response for gust wind using PSO

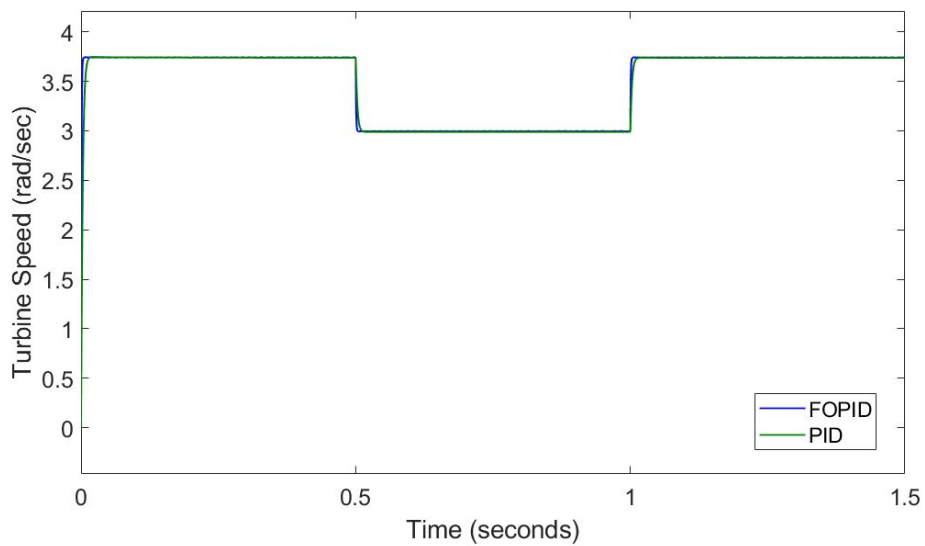


Figure 6.8: Response for base with step change wind using PSO

## 6.3 Analysis of result

### 6.3.1 Open Loop Response

From Figure 6.3 and Figure 6.4 it is evident that the turbine's speed does not vary according to the input wind. This causes the generator speed to fluctuate which results in low efficiency. Variations in the output must be controlled in order to attain desired outcomes.

### 6.3.2 Comparison Based on GA

Different wind profiles are used to evaluate and compare different genetic algorithm-based controller design methodologies. Figure 6.5 shows the responses of the wind turbine's dual mass system in the case of gust wind for both FOPID and PID controllers. Figure 6.6 illustrates the reaction of GA based controlled output for base with step-change wind speed, demonstrating how the turbine speed fluctuates in response to the input. The optimised controller parameters and the outcomes of the closed loop responses in terms of settling time, overshoot, and rise time are shown in Table 6.1.

Table 6.1: Comparison Based on GA

Controller	Settling time (s)	Rise time (s)	overshoot %
FOPID	0.0054	0.0036	0.43
PID	0.0128	0.0071	1.19

In this case, Integral of Time multiplied by Absolute Error (ITAE) is taken into consideration while using the objective function to decrease the error signal between the reference and measured output. ITAE is given as 0.0025 for the GA tuned FOPID controller and 0.0028 for the GA tuned PID controller. Due to a low fitness value, the GA adjusted FOPID controller is the fittest in this case.

### 6.3.3 Comparison based on PSO

The response of PSO-tuned FOPID and PID controllers is shown in Figure 6.7 for gust and Figure 6.8 for base step change wind. With respect to both gust and base step change wind speeds, the output varies based on the input. Because of this, the controlled output is more stable and can therefore produce the necessary turbine speed. Table 6.2 illustrates the optimised controller parameters as well as the results of the closed loop responses for PSO based FOPID and PID controllers, demonstrating the advantage of FOPID controllers in terms of settling time, overshoot, and rise time.

Table 6.2: Comparison Based on PSO

Controller	Settling time (s)	Rise time (s)	overshoot %
FOPID	0.0026	0.0015	0.15
PID	0.0104	0.0063	0.23

Here, ITAE is 0.0008 for the PSO-tuned FOPID controller and 0.0016 for the PSO-tuned PID controller. The PSO tuned FOPID is thought to be the fittest controller because to its low fitness value compared to all other controllers.

## 6.4 Conclusion

This section analysed the results of variable speed wind turbine speed control. Different wind profiles are used to test and compare the computational intelligence based controller design methodologies. The conclusions and future scope are presented in the following chapter.

## Chapter 7

# CONCLUSION AND FUTURE SCOPE

A fractional order PID controller is used to regulate the speed of a dual mass wind turbine system. Better transient performance is made possible by creating a wind turbine speed management technique that minimises the consequences of wind disturbance and severe overshoots. A typical PID controller and a FOPID controller are examined. Since controller tuning is a challenging task, computational intelligence techniques are taken into consideration. For tuning, CI techniques like the genetic algorithm and particle swarm optimization are used. This work examined the controller's closed-loop reactions like settling time, rising time and overshoot. Simulation results employing various wind profiles demonstrate that PSO tuned FOPID controllers outperform all other controllers in terms of closed loop performance. When compared to the various tuning techniques that have been explored in terms of stability and a variety of performance assessment metrics, the fractional order design technique produces the best results. In the future, more advanced computational intelligence techniques could be used to control the speed of variable speed wind turbines.

# References

- [1] C.-M. Hong, C.-H. Chen, and C.-S. Tu, “Maximum power point tracking-based control algorithm for pmsg wind generation system without mechanical sensors,” *Energy conversion and management*, vol. 69, pp. 58–67, 2013.
- [2] J. Mérida, L. T. Aguilar, and J. Dávila, “Analysis and synthesis of sliding mode control for large scale variable speed wind turbine for power optimization,” *Renewable Energy*, vol. 71, pp. 715–728, 2014.
- [3] M. L. Frikh, F. Soltani, N. Bensiali, N. Boutasseta, and N. Fergani, “Fractional order pid controller design for wind turbine systems using analytical and computational tuning approaches,” *Computers and Electrical Engineering*, vol. 95, p. 107410, 2021.
- [4] B. Boukhezzar and H. Siguerdidjane, “Comparison between linear and nonlinear control strategies for variable speed wind turbines,” *Control Engineering Practice*, vol. 18, no. 12, pp. 1357–1368, 2010.
- [5] M. Rahimian and K. Raahemifar, “Optimal pid controller design for avr system using particle swarm optimization algorithm,” in *2011 24th Canadian conference on electrical and computer engineering (CCECE)*, pp. 000337–000340, IEEE, 2011.
- [6] N. Fergani and A. Charef, “Process step response based fractional  $pi\lambda d\mu$  controller parameters tuning for desired closed loop response,” *International Journal of Systems Science*, vol. 47, no. 3, pp. 521–532, 2016.
- [7] R. F. Manrique Piramanrique and J. Sofrony Esmeral, “Data driven fault detection and isolation: a wind turbine scenario,” *Tecnura*, vol. 19, no. 44, pp. 71–82, 2015.

- [8] S. Ghasemi, A. Tabesh, and J. Askari-Marnani, "Application of fractional calculus theory to robust controller design for wind turbine generators," *IEEE transactions on energy conversion*, vol. 29, no. 3, pp. 780–787, 2014.
- [9] P. Novak, I. Jovik, and B. Schmidtbauer, "Modeling and identification of drive-system dynamics in a variable-speed wind turbine," in *Proceedings of the third IEEE conference on control applications*, vol. 1, pp. 233–238, 1994.
- [10] P. Cross and X. Ma, "Nonlinear system identification for model-based condition monitoring of wind turbines," *Renewable Energy*, vol. 71, pp. 166–175, 2014.
- [11] Z. Xin-Fang, X. Da-Ping, and L. Yi-Bing, "Adaptive optimal fuzzy control for variable speed fixed pitch wind turbines," in *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*, vol. 3, pp. 2481–2485, IEEE, 2004.
- [12] O. P. Agrawal and D. Baleanu, "A hamiltonian formulation and a direct numerical scheme for fractional optimal control problems," *Journal of Vibration and Control*, vol. 13, no. 9–10, pp. 1269–1281, 2007.
- [13] I. Petras, "The fractional-order controllers: Methods for their synthesis and application," *arXiv preprint math/0004064*, 2000.
- [14] C. A. Monje, B. M. Vinagre, V. Feliu, and Y. Chen, "Tuning and auto-tuning of fractional order controllers for industry applications," *Control engineering practice*, vol. 16, no. 7, pp. 798–812, 2008.
- [15] A. P. Engelbrecht, *Computational intelligence: an introduction*. John Wiley & Sons, 2007.
- [16] S. M. Thede, "An introduction to genetic algorithms," *Journal of Computing Sciences in Colleges*, vol. 20, no. 1, pp. 115–123, 2004.
- [17] D. Meena and A. Devanshu, "Genetic algorithm tuned pid controller for process control," in *2017 International Conference on Inventive Systems and Control (ICISC)*, pp. 1–6, IEEE, 2017.
- [18] L. Haldurai, T. Madhubala, and R. Rajalakshmi, "A study on genetic algorithm and its applications," *International Journal of Computer Sciences and Engineering*, vol. 4, no. 10, p. 139, 2016.

- [19] M. S. Hoque, M. Mukit, M. Bikas, A. Naser, *et al.*, “An implementation of intrusion detection system using genetic algorithm,” *arXiv preprint arXiv:1204.1336*, 2012.
- [20] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” *Soft computing*, vol. 22, no. 2, pp. 387–408, 2018.
- [21] R. Rajesh, “Optimal tuning of fopid controller based on pso algorithm with reference model for a single conical tank system,” *SN Applied Sciences*, vol. 1, no. 7, pp. 1–14, 2019.
- [22] B. Al-Kazemi and C. K. Mohan, “Multi-phase generalization of the particle swarm optimization algorithm,” in *Proceedings of the 2002 congress on evolutionary computation. CEC’02 (cat. No. 02TH8600)*, vol. 1, pp. 489–494, IEEE, 2002.

# List of Publications

- [1] Ahna Noushad and Salil N, "Computational Intelligence based Speed Control of Variable Speed Wind Turbine using Fractional Order Controller," communicated to *International Conference on Edge Computing and Applications (ICECAA)*, 2022.
- [2] Ahna Noushad and Salil N, "Speed Control of Variable Speed Wind Turbine Using FOPID Controller," communicated to *IEEE 19th India Council International Conference (INDI-CON)*, 2022.