

ADDIS ABABA INSTITUTE OF TECHNOLOGY
ENERGY CENTER



**Model Based Design of HAWT and Its Control System under
Simulink Environment**

By: Tewodros Walle

Advisor: Dr.-Ing. Abebayehu Assefa

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Masters of Science in Energy Technology.

Addis Ababa Institute of Technology

Addis Ababa, Ethiopia

August, 2011

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
ENERGY CENTER
ENERGY TECHNOLOGY PROGRAM

By Tewodros Walle _____

Student *Signature* *Date*

Approved by:

1. *Dr.-Ing. Abebayehu Assefa.* _____

Advisor *Signature* *Date*

2. _____

Chairman, Dept.'s *Signature* *Date*

Graduate Committee

3. _____

Chairman, Faculty's *Signature* *Date*

Graduate Committee

4. _____

Dean, Graduate school *Signature* *Date*

ACKNOWLEDGMENT

This project would be impossible without the help of God and his Mother. I would like to express my sincere gratitude to my advisor **Dr.-Ing Ababayehu Assefa** for giving me the opportunity to work on this project and for his guidance and encouragement without which this work would not have been completed. He has been a constant source of inspiration throughout my study period.

I am also grateful to **Mr. Shetaye Leulseged** for helping me with material and inspiration starting from the beginning of my MSc program and for all tutorial teachers in Mathwork for the kind support rendered to me on different occasions.

Last but not least, I would like to thank my family and friends Yidnekachew Melesse and Alazar Tadele who were always beside me and played a great role in the completion of my work.

Table of Contents

List of Figure	v
List of Tables.....	viii
Abstract	ix
Acronyms	x
Chapter One.....	1
1.1 Background	1
1.2 Literature Review	2
1.3 Objectives	2
1.4 Methodology	3
1.5 Limitation.....	3
Chapter Two	4
2.1 Introduction.....	4
2.1.1 What is Wind?.....	4
2.1.2 Wind Availability.....	4
2.1.3 Significant Impact of Wind Speed on the Power Production	4
2.1.4 Wind Energy	4
2.1.5 Where Does Wind Energy Come From?.....	4
2.1.6 Wind Turbine and Its Working Principles.....	4
2.1.7 The Power of the Wind.....	5
2.1.8 Wind Speed Measurement: Anemometers	5
2.2 Aerodynamics of Wind Turbines	6
2.2.1 Airfoils and General Concepts of Aerodynamics	6
2.2.2 Lift, Drag and Non-dimensional Parameters	7
Chapter Three.....	9
3.1 Simulink.....	9
3.1.1 How Simulink Software Interacts with the MATLAB Environment	9
3.1.2 Starting Simulink Software.....	9
3.1.3 Opening an Existing Model.....	10
3.1.4 Running the Simulink model	10

3.2 Model-Based Design	11
3.2.1 Modeling Process	11
3.2.2 Tool for Model-Based Design	13
3.2.3 Tool for Simulation	13
3.2.4 Tool for Analysis	13
3.3 Comparison of Model Based Design with Traditional Development Process.....	13
Chapter Four	15
4.1 Modeling Wind Turbine Block Set Under Simulink Environment.....	15
4.1.1 General Structure	15
4.1.2 Mechanical Components Library.....	16
4.1.3 Electrical Machinery Library.....	16
4.1.4 Power Converters Library	17
4.1.5 Common Bocks Library	18
4.1.6 Transformations Library.....	18
4.1.7 Measurements Library.....	19
4.1.1 Control Library	19
Chapter Five	20
5.1 Overview of the Wind Turbine Simulink Model.....	20
Chapter Six	26
6.1 Modeling Subsystem of the Wind Turbine.....	26
6.2 Wind Input to the System (Turbine Input).....	26
6.3 Aerodynamic Loads (Blade Lift and Drag)	28
6.4 Blade.....	33
6.4.1 Aerodynamics Design	33
6.4.2 Structural Design.....	36
6.4.3 Defining a Body with Mass, Coordinate System and Visualization Properties	
Mass Properties	37
6.4.4 Modeling the Blade Pitch Linkage.....	40
6.4.5 Modeling the Blade as Flexible Body	41
6.4.6 Comparing Methods and Results	46
6.5 Nacelle	48
6.6 Gear Train.....	49

6.7 Generator.....	53
6.7.1 Power Output of the Turbine to the Grid.....	55
6.8 Tower.....	56
6.9 Transmission to Grid	57
6.10 Machine Environment	58
Chapter Seven.....	60
7.1 Control System Design of the Wind Turbine.....	60
7.2 Main Controller	61
7.3 Pitch Control	65
7.4 Pitch controller	66
7.3.1 PI and PID controllers	68
7.3.2 Pitch Actuation System Simulink Model	69
7.3.3 Modeling the pitch control system based on the ideal system.....	73
7.3.4 Initial design of the actual pitch actuator system.....	74
7.3.5 Compensator Design	76
7.5 Yaw System	87
Chapter Eight	94
8.1 Conclusions and Recommendations for Future Work.....	94
Reference.....	96
Appendix A: Wind Turbine System Requirements.....	98
Appendix B: Wind Turbine Parameters.....	104
Appendix C: Embedded System of Aerodynamic Load.....	111
Appendix D: The NACA 4 Digit Airfoil.....	117
Appendix E: Integrating Airfoil of Blade with the Model.....	121
Appendix F: NACA 8412 Airfoil.....	122
Appendix G: Lift and Drag Coefficient	123
Appendix H: Automatic Report Generation.....	129

List of Figure

Figure 2- 1:	Power vs wind speed.....	5
Figure 2- 2:	Anemometer	6
Figure 2- 3:	Airfoil Nomenclature.....	7
Figure 2- 4:	Forces and moments on an airfoil section, angle of attack and chord.....	7
Figure 4- 1	Structure of the Simulink Toolbox for Wind Turbine Simulations	15
Figure 4- 2:	Mechanical Components Library	16
Figure 4- 3:	Electrical Machinery Library	17
Figure 4- 4:	Content of the Power Converters Library.....	17
Figure 4- 5:	Common Blocks Library	18
Figure 4- 6:	Content of the Transformations Library.....	19
Figure 4- 7:	Measurements Library	19
Figure 4- 8:	Content of the Control Blocks Library.....	19
Figure 5- 1:	Wind Turbine Overall System.....	22
Figure 5- 2:	3D Animation	23
Figure 5- 3:	Wind Speed and Direction	24
Figure 5- 4:	Rotor Speed (RPM)	24
Figure 5- 5:	Pitch Angle and Command (Deg)	24
Figure 5- 6:	Nacelle Yaw	25
Figure 6- 1:	Signal Builder Block.....	26
Figure 6- 2:	Wind Speed and Direction	27
Figure 6- 3:	Wind Speed and Direction (Full)	27
Figure 6- 4:	Wind Direction Indicator Diagram.....	27
Figure 6- 5:	Blade as Single Element	29
Figure 6- 6:	Blade as Multiple Elements	29
Figure 6- 7:	Blade Loads.....	29
Figure 6- 8:	Simulink Representation of Blade Loads	30
Figure 6- 9:	Simulink Representation of the Angle of Attack	30
Figure 6- 10:	Lift and Drag Coefficient.....	31
Figure 6- 11:	Lift and Drag force	31
Figure 6- 12:	Moments on the Blade	32
Figure 6-13:	Blade Actuator.....	32
Figure 6-14:	Embedded Systems of Aerodynamic Loads	33
Figure 6- 15:	NACA four-digit airfoil profile.....	34
Figure 6-16:	Generated NACA 8412 Airfoil Profile.....	36
Figure 6- 17:	Blade	36

Figure 6- 18: Blade body block parameter.....	37
Figure 6- 19: Position of the Blade.....	38
Figure 6- 20: Orientation of the Blade.....	39
Figure 6- 21: Visualization of the blade	40
Figure 6- 22: Pitch Actuation Linkage	40
Figure 6- 23: 3D Animation of Blade.....	41
Figure 6- 24: Flexible Cantilever Lumped Parameter Approach.....	41
Figure 6- 25: Simulink Model of Lumped Parameter Approach	42
Figure 6- 26: Discretization of a Beam into Generalized Beam Elements.....	42
Figure 6- 27: Joint Spring and Damper.....	43
Figure 6- 28: Flexible Cantilever FEA Approach	44
Figure 6- 29: Using Finite Element Analysis.....	44
Figure 6- 30: Embedding State-Space Flexible Body Model	45
Figure 6- 31: Embedded state-space flexible body model in SimMechanics.....	45
Figure 6- 32: Motion of Body-loaded Beam, Modeled by SimMechanics with 10 GBEs.....	46
Figure 6- 33: Motion of Body-loaded Beam, Modeled by FEA and State Space Dynamics	46
Figure 6- 34: Comparison of the Two Methods.....	47
Figure 6- 35: Gear train input torque.....	48
Figure 6- 36: Subsystem of Nacelle.....	49
Figure 6- 37: Gear train.....	49
Figure 6- 38: Simulink Gear Train	50
Figure 6- 39: Subsystem of Brake	50
Figure 6- 40: Clutch Block Parameter	51
Figure 6- 41: Gear train ratio.....	52
Figure 6- 42: Generator power versus generator speed	53
Figure 6- 43: Gear train input torque.....	53
Figure 6- 44: Subsystem of the Generator	54
Figure 6- 45: Generator at Full Load Model.....	55
Figure 6- 46: Rotor and generator speed.....	56
Figure 6- 47: Power to the grid	56
Figure 6- 48: Subsystem of tower.....	57
Figure 6- 49: Transmission Line to Grid	58
Figure 6- 50: Environment Block Parameter	59
Figure 7- 1: Subsystem in the main controller (Supervisory control)	62
Figure 7- 2: Subsystem in turbine state machine.....	63
Figure 7- 3: Controllable frictional clutch in brake sub system	64
Figure 7- 4: Display of turbine state machine	65
Figure 7- 5: Control structure for pitch control.....	66
Figure 7- 6: Overall model of the pitch control of the wind turbine model.....	67

Figure 7- 7:	Simulink representation for determining the pitch command.....	67
Figure 7- 8:	Subsystem of the pitch controller	68
Figure 7- 9:	PI controller.....	69
Figure 7- 10:	Subsystem of pitch actuation system.....	69
Figure 7- 11:	Revolute Motion of Follower (blue) Relative to Base (red).....	70
Figure 7- 12:	Spring & Damper block	70
Figure 7- 13:	Hydraulic subsystem.....	71
Figure 7- 14:	Subsystem of hydraulic actuator system.....	72
Figure 7- 15:	Pitch actuation system	73
Figure 7- 16:	Ideal pitch control system	74
Figure 7- 17:	Ideal pitch angle and force	74
Figure 7- 18:	Initial design pitch angle and force.....	75
Figure 7- 19:	Pitch angle and pitch actuation force.....	75
Figure 7- 20:	3D animation of blade.....	76
Figure 7- 21:	Control and estimation tool manager.....	77
Figure 7- 22 :	Closed loop signals	78
Figure 7- 23:	Operating point.....	78
Figure 7- 24:	Design configuration wizard	79
Figure 7- 25:	Root Locus	79
Figure 7- 26:	Step response plot.....	80
Figure 7- 27:	LIT viewer.....	81
Figure 7- 28:	LIT viewer.....	81
Figure 7- 29:	Pitch command	82
Figure 7- 30:	Force command	82
Figure 7- 31:	Initial design value of proportional and integral gain.....	83
Figure 7- 32:	Better design value of proportional and integral gain	83
Figure 7- 33:	Optimized design value of proportional and integral gain.....	84
Figure 7- 34:	Compensation design results.....	85
Figure 7- 35:	Comparison of the ideal and optimized design	86
Figure 7- 36:	Yaw control system	88
Figure 7- 37:	Subsystem of yaw motor.....	88
Figure 7- 38:	Subsystem of the servo motor	89
Figure 7- 39:	Subsystem of the Yaw gearbox.....	89
Figure 7- 40:	Top view of Nacelle.....	90
Figure 7- 41:	Yaw controller	90
Figure 7- 42:	Control structure for Yaw control	91
Figure 7- 43:	Simulink control structure for Yaw control.	91
Figure 7- 44:	Nacelle Yaw	92
Figure 7- 45:	Nacelle Yaw rate	92
Figure 7- 46:	Torque Command	92
Figure 7- 47:	Yaw actuator torque.....	93

List of Tables

Table 1:	Representative Angle of the Wind Direction.....	28
Table 2:	NACA 8412.....	34
Table 3:	Comparison of FEA and Lumped Parameter.....	48
Table 4:	Generator Requirement	52
Table 5:	Environment Requirements.	58

Abstract

This project addresses the model based design of a horizontal axis wind turbine simulation using MATLAB/Simulink. Wind turbine consists of different subsystems that use different technologies. Usually different teams develop each subsystem in separate environments, integrate and test it after developing a prototype.

This project, however, starts with specifications and requirement of the model to model the whole system, integrate the system and test it in single environment before a prototype is developed.

In order to analyze the dynamic and/or steady state behavior of a wind turbine, the basic components of a wind turbine are structured in several libraries. Mechanical components, electrical machinery, power converters, common models, transformers, measurements and control, for studying the whole system and the effect of the change of one of the wind turbine components in the wind turbine efficiency are presented.

Basic blade geometry, force and moment equation of the wind turbine model component are presented as embedded system in the model. Each equation of the control system converted into block diagram for integrating with Simulink environment for each control system of the model compensator design optimization is done. The NACA four-digit airfoil profile generated based on the recommended analytical equations of NACA aerodynamics blade profile. The profile is changed into MATLAB code to BE embedded in Simulink environment.

3D animations of wind turbine and its component models are developed to show its motion and response for the various operating condition of the system.

A graphical user interface (GUI) is built which makes any user of this model to test the power output and the various operating condition for the data of selected site that a user wants. For each control system of the model compensator design optimization is done.

Results of the system simulation are presented in graphical form suitable for system performance determination. From the power output, pitch actuator force, pitch actuator torque, Yaw actuator torque and wind turbine parameters, the useful energy and power output are determined.

Acronyms

HAWT	Horizontal Axis Wind Turbine
VCS	Voltage Source Converter
NACA	The United States National Advisory Committee for Aeronautics
CG	Center of Gravity
CSs	Coordinate Systems
GBEs	Generalized Beam Elements
FEA	Finite Element Analysis
GBEs	generalized beam elements
LTI	linear, time-invariant
PI	proportional and integral
PID	proportional and integral differential
CS	Function
V	velocity of air
P	density of air
C_L	Lift Coefficient
C_D	Drag Coefficient

Chapter One

1.1 Background

A wind turbine is a machine that converts the wind's kinetic energy into rotary mechanical energy, which is then used to do work. In more advanced models, the rotational energy is converted into electricity, the most versatile form of energy, by using a generator.

According to Carlin, P.W [1], horizontal axis wind turbines may have been invented as early as the twelfth century. These turbines were used for several operations, such as milling wheat into flour and would be driven at a variable rate. In the 19th century wind turbines were largely used across the United States to pump water, charge batteries and run farm equipment. In fact, in 1888, Brush Wind Turbines in Cleveland, Ohio was producing up to 12 kW of direct current (DC) power for charging batteries. Up to this point in time, for the majority of uses for wind turbines, varying speed would only impact the rate at which work would be accomplished, or the voltage produced. Even though one can easily produce DC power from turbines rotating at a variable speed, due to the high voltage that is needed to efficiently distribute DC power through long transmission lines, it was very difficult to transport this power to where it was needed most, in cities. Due to this fact, alternating current (AC) quickly became the choice for power distribution and was standardized at 60 Hz.

Variable-speed wind turbines have become more common than traditional fixed-speed turbines [2]. In a variable-speed wind turbine with full-power converter, the wind turbine is directly connected to the generator and the generator is completely decoupled from the electric grid. The advantages of using a variable speed wind turbine design versus a fixed speed wind turbine are now commonly accepted. Depending on location and wind profile a variable speed wind turbine can produce up to 20 % more electrical power and increase the life of its mechanical components [3]. In order to achieve the variable speed operation several solutions have been devised, almost all of which deal with power electronics

The classical analysis of the wind turbine was originally developed by Betz and Glauert (Glauert, 1935) in the 1930s. Subsequently, the theory was expanded and adapted for solution by digital computers (see Wilson and Lissaman, 1974; Wilson et al., 1976; de Vries, 1979). In all of these methods, momentum theory and blade element theory are combined into a strip theory that enables calculation of the performance characteristics of an annular section of the rotor. The characteristics for the entire rotor are then obtained by integrating, or summing, the values obtained for each of the annular sections [3].

1.2 Literature Review

David James Burnham [4] has discussed about Control of Wind Turbine Output Power. The software package PSCAD is used for the wind turbine simulations. PSCAD solves the differential equations representing an electrical network and mechanical components in the time domain. By modifying the generator torque-speed characteristic, it shows and discusses that it is possible to maintain a flat power curve between the rated and cut-out wind speeds to achieving a higher power capture.

Richard Gagnon, et. al. [5] has described modeling and real-time simulation of a doubly-fed induction generator driven by a wind turbine. The paper presented the modeling and real-time simulation of a generic wind-turbine doubly-fed induction generator (WT_DFIG) in a power system. MATLAB/Simulink software is used to develop the WT_DFIG model. The generated code of the Simulink model is linked to the Hypersim digital real-time simulator [4] in order to simulate the power system together with wind turbines. The paper is divided into three parts. The MATLAB/Simulink model is developed in the first part. The implementation of the model in Hypersim is presented in the second part. The third part presents a case study of real-time simulation of wind turbines in a generic power system. A case study example of real-time simulation of wind turbines in a generic series compensated power system is presented.

Nicholas W. Miller, et. al. [6] have discussed dynamic modeling of GE 1.5 and 3.6 MW wind turbine-generators for stability simulations. The dynamic models presented here are specific to the GE WTGs. The implementation is structured in a fashion that is similar to other conventional generators. To construct a complete WTG model, three device models are used: Generator/converter model, Electrical control model and Turbine and turbine control model. A fourth, user-written model can be used to simulate a wind gust by varying input wind speed to the turbine model. The user can also input wind speed vs time sequences, derived from field measurements or other sources.

Leithead has presented in [8], [9] and [10] a detailed description of the wind turbine mechanical parts. The structural dynamics of the whole drive train was thoroughly described and analyzed using MATLAB/Simulink software. The aerodynamic interaction between free wind and the turbine itself was incorporated into an overall model in the form of various filters applied to the free wind data.

1.3 Objectives

The main objective of this thesis work is to develop a detailed MATLAB Simulink model based and control model of a general wind turbine applicable for any site. For this goal, the following specific objectives are set:

- Studying the characteristics of different wind turbines;
- Mathematical modeling of different components of a typical wind turbine;
- Studying behavior of different control mechanisms for the selected wind turbine;
- Selecting the best control mechanism;
- Mathematical modeling of the selected control mechanism;
- Developing Simulink equivalent of each component of a wind turbine;
- Developing Simulink equivalent of the selected control mechanism;
- Construct the whole wind turbine and its control mechanism with MATLAB-Simulink;
- Simulating the different characteristics of the wind turbine and control mechanism by varying input variables; and
- Observing the effect of optimization compensator design of different components on the overall performance and control of the turbine

1.4 Methodology

The methods to be employed to achieve the objectives of the research are:

- Literature review;
- Wind speed simulation;
- Building wind turbine model blocks;
- Mathematical models of wind turbines;
- Simulink model of the whole system;
- Investigate the variation of the output power of the wind turbine based on variable speed nature of the wind speed; and
- Give explanation to the control feedback.

1.5 Limitation

The signal builder of the HAWT model was modeled to work with directly connected data logger to study the control system. Unfortunately a data logger could not be obtained to test the model by connecting with it.

Chapter Two

2.1 Introduction

2.1.1 What is Wind?

Wind is simply air in motion. It is caused by the uneven heating of the earth's surface by the sun. Since the earth's surface is made up of land, desert, water and forest areas, the surface absorbs the sun's radiation differently.

2.1.2 Wind Availability

Whether constructing a wind turbine is economically viable at our home or farm depends most strongly on the quality of our wind resource. Generally, average annual wind speeds of at least 4.0-4.5 m/s are needed for a small wind turbine to produce enough electricity and to be cost-effective.

2.1.3 Significant Impact of Wind Speed on the Power Production

The wind turbine doesn't generate the same amount of power all the time. The wind speed at a single location may vary considerably and this can have a significant impact on the power production from a wind turbine.

2.1.4 Wind Energy

Wind energy is the kinetic energy associated with the movement of atmospheric air. It has been used for hundreds of years for sailing, grinding grain and for irrigation. (Water pumping) Wind energy system converts this kinetic energy to more useful form of power. Wind energy system for irrigation and milling has been in use since ancient times and since the beginning of the 20th century it is being used to generate electric power. Windmills for water pumping have been installed in many countries particularly in the rural areas.

2.1.5 Where Does Wind Energy Come From?

All renewable energy (except tidal and geothermal power) and even the energy in fossil fuels, ultimately comes from the sun. The sun radiates 100×10^{12} kilo watt hours of energy to the earth per hour. In other words, the earth receives 10 to the 18th power of watts of power. About 1 to 2 % of the energy coming from the sun is converted into wind energy.

2.1.6 Wind Turbine and Its Working Principles

A wind turbine obtains its power input by converting the force of the wind into torque (turning force) acting on the rotor blades by means of converting the kinetic energy of the wind into kinetic energy of a rotating shaft.

Energy extraction from wind, by wind energy converters, is always related to a certain time difference as wind and operational conditions are usually subject to constant changes. This is why in most cases the instantaneous energy value. Wind energy generators are equipped with rotors to extract wind power and consist of one or several rotor blades. The extracted wind power generates rotation and is thereby converted into mechanical power at the rotor shaft. Mechanical power is taken up at the shaft in the form of a moment at a certain rotation and is transferred to generator. It is physically impossible to technically exploit the entire wind energy, as in this case air flow would come to a standstill. In this case, air would fail to enter the swept rotor area and wind power would no longer be available. There are two different physical principles to extract power from wind.

- The less efficient airfoil drag method is based on the wind drag force incident on wind-blown surface.
- The second principle also referred to as aerodynamic airfoil lift principle, which is based on flow deviation inside the rotor – is predominantly applied for wind energy conversion. When compared to the drag principle, double or triple the power output is achieved for a given cross-section area.

2.1.7 The Power of the Wind

The wind speed is extremely important for the amount of energy a wind turbine can convert to electricity: The energy content of the wind varies with the cube (the third power) of the average wind speed.

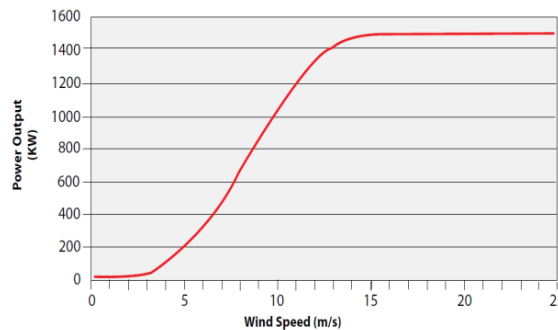


Figure 2- 1: Power vs wind speed

2.1.8 Wind Speed Measurement: Anemometers

The measurement of wind speeds is usually done using a cup anemometer, such as the one in Figure 2- 2 below. The cup anemometer has a vertical axis and three/four cups which capture the wind. The number of revolutions per minute is registered electronically.

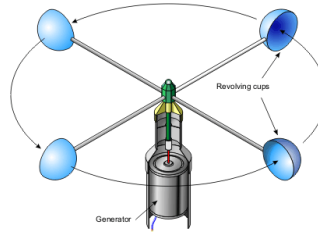


Figure 2- 2: Anemometer

2.2 Aerodynamics of Wind Turbines

Wind turbine power production depends on the interaction between the rotor and the wind. The wind may be considered to be a combination of the mean wind and turbulent fluctuations about that mean flow. The major aspects of wind turbine performance (mean power output and mean loads) are determined by the aerodynamic forces generated by the mean wind. Periodic aerodynamic forces caused by wind shear, off-axis winds and rotor rotation and randomly fluctuating forces induced by turbulence and dynamic effects are the source of fatigue loads and are a factor in the peak loads experienced by a wind turbine.

Practical horizontal axis wind turbine designs use airfoils to transform the kinetic energy in the wind into useful energy.

2.2.1 Airfoils and General Concepts of Aerodynamics

Airfoils are structures with specific geometric shapes that are used to generate mechanical forces due to the relative motion of the airfoil and a surrounding fluid. Wind turbine blades use airfoils to develop mechanical power. The cross-sections of wind turbine blades have the shape of airfoils. The width and length of the blade are functions of the desired aerodynamic performance, the maximum desired rotor power, the assumed airfoil properties and strength considerations.

A number of terms are used to characterize an airfoil, as shown in Figure 2.3. The mean camber line is the locus of points halfway between the upper and lower surfaces of the airfoil. The most forward and rearward points of the mean camber line are on the leading and trailing edges, respectively. The straight line connecting the leading and trailing edges is the chord line of the airfoil and the distance from the leading to the trailing edge measured along the chord line is designated the chord, c , of the airfoil. The camber is the distance between the mean camber lines and the chord line, measured perpendicular to the chord line. The thickness is the distance between the upper and lower surfaces, also measured perpendicular to the chord line. Finally, the angle of attack, α , is defined as the angle between the relative wind and the chord line.

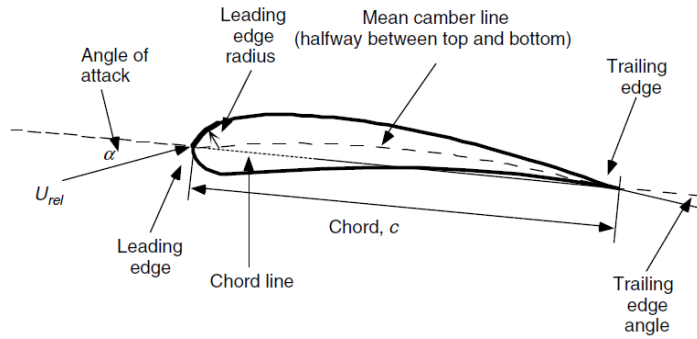


Figure 2- 3: Airfoil Nomenclature

The span of the airfoil is not shown in the figure, which is the length of the airfoil perpendicular to its cross-section. The geometric parameters that have an effect on the aerodynamic performance of an airfoil include: the leading edge radius, mean camber line, maximum thickness and thickness distribution of the profile and the trailing edge angle.

2.2.2 Lift, Drag and Non-dimensional Parameters

Air flow over an airfoil produces a distribution of forces over the airfoil surface. The flow velocity over airfoils increases over the convex surface resulting in lower average pressure on the ‘suction’ side of the airfoil compared with the concave or ‘pressure’ side of the airfoil. Meanwhile, viscous friction between the air and the airfoil surface slows the air flow to some extent next to the surface.

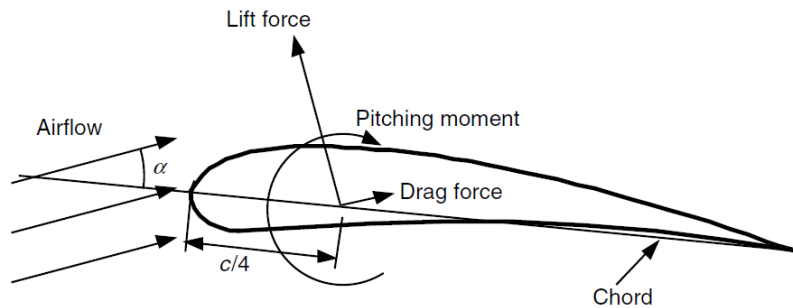


Figure 2- 4: Forces and moments on an airfoil section, angle of attack and chord.

The direction of positive forces and moments is indicated by the direction of the arrow. As shown in Figure 2- 4. The resultant of all of these pressure and friction forces is usually resolved into two forces and a moment that acts along the chord at a distance of $c/4$ from the leading edge.

- Lift force: defined to be perpendicular to direction of the oncoming air flow. The lift force is a consequence of the unequal pressure on the upper and lower airfoil surfaces.
- Drag force: defined to be parallel to the direction of the oncoming air flow. The drag force is due both to viscous friction forces at the surface of the airfoil and to unequal pressure on the airfoil surfaces facing toward and away from the oncoming flow.
- Pitching moment: defined to be about an axis perpendicular to the airfoil cross-section.

Chapter Three

3.1 Simulink

Simulink is software that models, simulates and analyzes dynamic systems. It enables to raise a question about a system, model the system and see what happens. Simulink can be used to easily build models from scratch, or modify existing models to meet the needs. Simulink supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multi rate, having different parts that are sampled or updated at different rates. Scientists and Engineers around the world use Simulink to model and solve real problems in a variety of industries, including:

- Aerospace and Defense
- Automotive
- Communications
- Electronics and Signal Processing
- Medical Instrumentation
- Power Production


3.1.1 How Simulink Software Interacts with the MATLAB Environment

Simulink software is tightly integrated with the MATLAB environment. It requires MATLAB to run, depending on it to define and evaluate model and block parameters. Simulink can also utilize many MATLAB features. For example, Simulink can use the MATLAB environment to:

- Define model inputs.
- Store model outputs for analysis and visualization.
- Perform functions within a model, through integrated calls to MATLAB operators and functions.

3.1.2 Starting Simulink Software

To start the Simulink, first start the MATLAB, then start the Simulink software in two ways:

- On the toolbar, click the Simulink icon.  or
- Enter the Simulink command at the MATLAB prompt.

The Library Browser appears. It displays a tree-structured view of the Simulink block libraries installed on the system. Build models by copying blocks from the Library Browser into a model window.

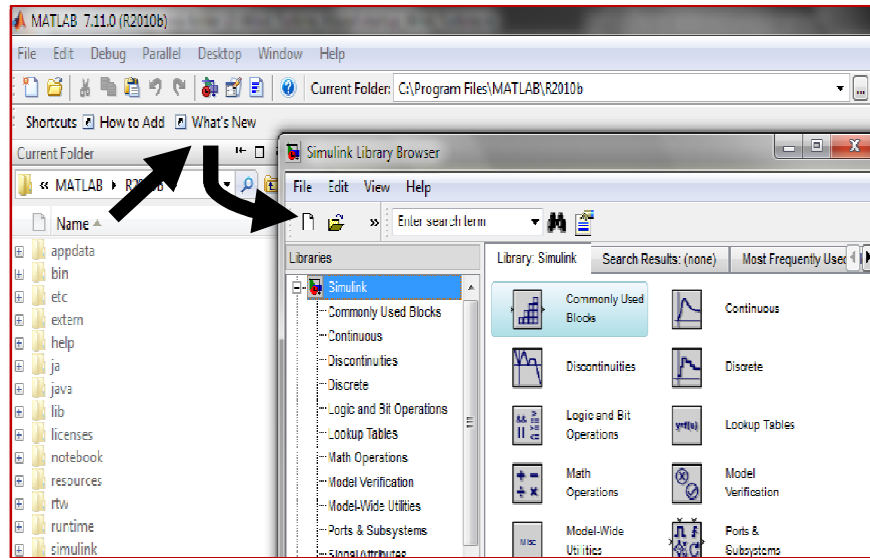


Figure 3- 1: Starting Simulink

3.1.3 Opening an Existing Model

To open an existing model diagram, either:

- Click the Open button on the Library Browser toolbar or select Open from the Simulink library window’s File menu and then choose or enter the file name for the model to edit.
- Enter the name of the model (without the .mdl extension) in the MATLAB software Command Window. The model must be in the current folder or on the path.

3.1.4 Running the Simulink model

Consider a sample model which display the value of sine wave as shown in Figure 3- 2 below.

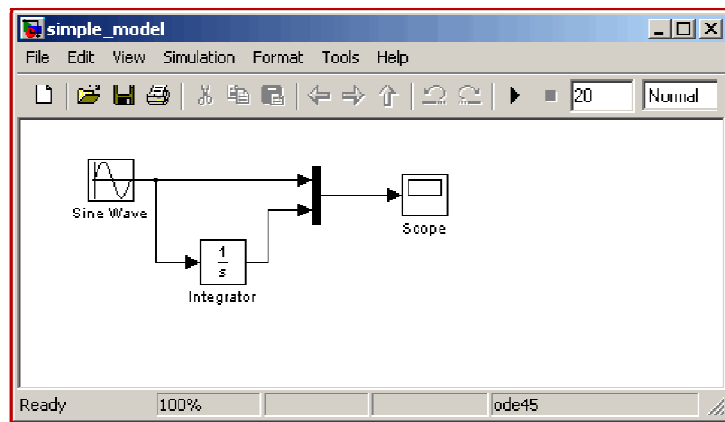



Figure 3- 2: Running the Simulink

On the toolbar, click the Simulink icon  to run the model.

3.2 Model-Based Design

Model Based Design is a process that enables faster, more cost effective development of dynamic systems including control systems, signal processing and communications systems. In Model-Based Design, a system model is at the center of the development process, from requirements development, through design, implementation and testing. The model is an executable specification that is continually refined throughout the development process. After model development, simulation shows whether the model works correctly.

When software and hardware implementation requirements are included, such as fixed point and timing behavior, the user can automatically generate code for embedded deployment and create test benches for system verification, saving time and avoiding the introduction of manually coded errors.

Model-Based Design allows improving efficiency by:

- Using a common design environment across project teams;
- Linking designs directly to requirements;
- Integrating testing with design to continuously identify and correct errors;
- Refining algorithms through multi domain simulation;
- Automatically generating embedded software code;
- Developing and reusing test suites;
- Automatically generating documentation; and
- Reusing designs to deploy systems across multiple processors and hardware targets.

3.2.1 Modeling Process

There are six steps to modeling any system. It needs to perform the first three steps of this process outside of the Simulink software before beginning to building a model.

1. Defining the System

The first step in modeling a dynamic system is to fully define the system. When modeling a large system it need to broken into parts and should model each subcomponent on its own. Then, after building, each component is integrated into a complete model of the system.

2. Identifying System Components

The second step in the modeling process is to identify the system components. There are three types of components to define a system:

- Parameters: System values that remain constant unless the user change them;
- States: Variables in the system that change over time; and
- Signals: In Simulink, parameters and states are represented by blocks, while signals are represented by the lines that connect blocks.

3. Modeling the System with Equations

The third step in modeling a system is to formulate the mathematical equations that describe the system. For each subsystem, the list of system components are used to describe the system mathematically. The model may include:

- Algebraic equations;
- Logical equations;
- Differential equations, for continuous systems; and
- Difference equations, for discrete systems.

These equations are used to create the block diagrams in Simulink.

4. Building the Simulink Block Diagram

After defining the mathematical equations that describe each subsystem, building a block diagram of that model in Simulink is started. Block diagram is built for each of subcomponents separately and then they are integrated into a complete model of the system.

5. Running the Simulation

After building the Simulink block diagram, the model is simulated and the results analyzed. Simulink allows interactively defining system inputs, simulating the model and observing changes in behavior. This allows quick evaluation of the model.

6. Validating the Simulation Results

Finally, the model is validated whether or not it accurately represents the physical characteristics of the system. The linearization and trimming tools available from the MATLAB command line, plus the many tools in MATLAB and its application Toolboxes are used to analyze and validate the given model.

3.2.2 Tool for Model-Based Design

With Simulink, the model moves beyond idealized linear models to explore more realistic nonlinear models, factoring in friction, air resistance, gear slippage, hard stops and the other things that describe real world phenomena. Simulink turns computers into a laboratory for modeling and analyzing systems that would not be possible or practical otherwise. Simulink provides the tools to model and simulate almost any real world problem. Simulink provides a graphical user interface (GUI) for building models as block diagrams, allowing drawing models as would with pencil and paper. Simulink also includes a comprehensive block library of sinks, sources, linear and nonlinear components and connectors. If these blocks do not meet the designers need, however, they also create their own blocks. The interactive graphical environment simplifies the modeling process, eliminating the need to formulate differential and difference equations in a language or program.

3.2.3 Tool for Simulation

After defining a model, one can simulate it, using a choice of mathematical integration methods, either from the Simulink menus or by entering commands in the MATLAB Command Window. Using scopes and other display blocks, the simulation results can be seen while the simulation runs. Changing of parameters is possible to see what happens for “what if” exploration. The simulation results can be put in the MATLAB workspace for post-processing and visualization.

3.2.4 Tool for Analysis

Model analysis tools include linearization and trimming tools, which can be accessed from the MATLAB command line, plus the many tools in MATLAB and its application Toolboxes because MATLAB and Simulink are integrated. Therefore, simulating, analyzing and revising models in either environment at any point is possible.

3.3 Comparison of Model Based Design with Traditional Development Process

Wind turbine consists of different subsystems that use different technologies. Usually different teams develop each subsystem and integrate.

In traditional development process, engineers start with specification and requirement and develop design. In order to test the developed system, it is integrated with the other built subsystems, a prototype produced and finally tested. With the traditional development process, there are a few difficulties.

1. It is difficult for engineers to compare the design they have with the requirement and specification. Requirement may be unclear, incomplete and

not integrated in design process. It is difficult for the engineers to improve and iterate the design to meet their design requirements.

2. The second problem that traditional development processes have is that individual teams work separately from one another. It is difficult for them to test and see what will happen when different designs are integrated; because they simulate and design in different environments. This leaves with the only option that the test and integration issues are verified in hardware prototype.

But in model based design process, there are some changes.

1. The whole model is simulated in a single environment. In this process, the test and integration issues are detected earlier, because different teams communicate with one another.

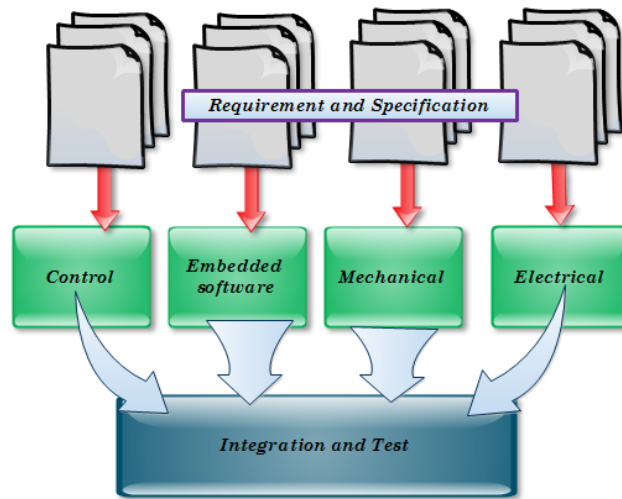


Figure 3- 3: Creating single Simulink Environment

2. The engineer makes sure that the design requirements are met. This procedure generates better design by continuously comparing design and specification and also integration and testing.

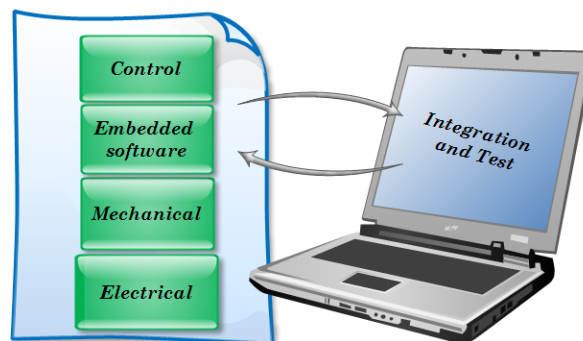


Figure 3- 4 Integration and Testing

Chapter Four

4.1 Modeling Wind Turbine Block Set Under Simulink Environment

4.1.1 General Structure

To analyze the dynamic and steady state behavior of a wind turbine, the basic components of a wind turbine have been modeled and structured in seven libraries: Mechanical Components, Electrical Machinery, Power Converters, Common Models/Blocks, Transformations, Measurements and Control as shown in Figure 4- 1.

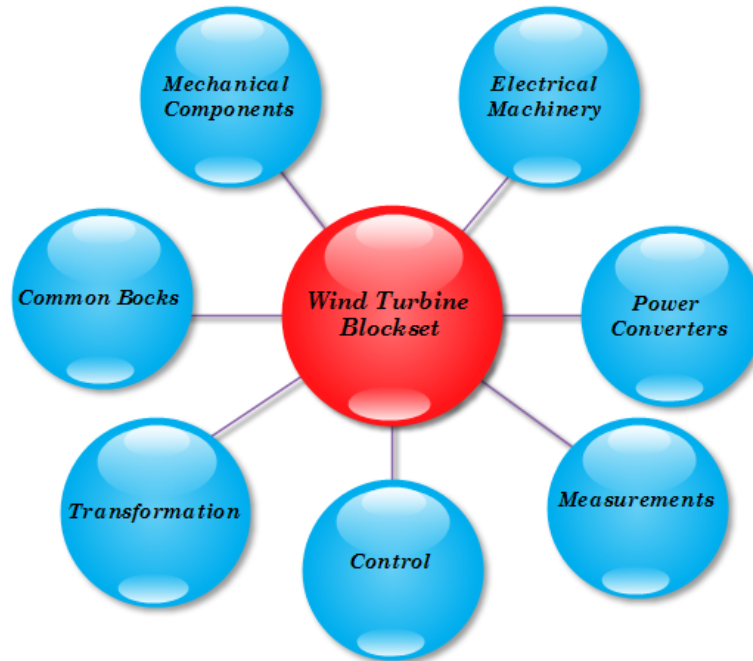


Figure 4- 1 Structure of the Simulink Toolbox for Wind Turbine Simulations

These models are built based on Simulink blocks as well as using CS Function. Some of the features of this developed toolbox can be summarized as follows:

- All the developed models use basically only Simulink blocks;
- It uses the matrix support in order to minimize the number of blocks and connection lines;
- All models which involve a great number of differential equations (e.g. electrical machines, drive-trains and transmissions) are also available as C S-Functions for high-speed simulations.

4.1.2 Mechanical Components Library

The first, Mechanical Components Library contains: wind model, aerodynamic models of the wind turbine rotor and different types of drive train models as shown in Figure 4- 2.

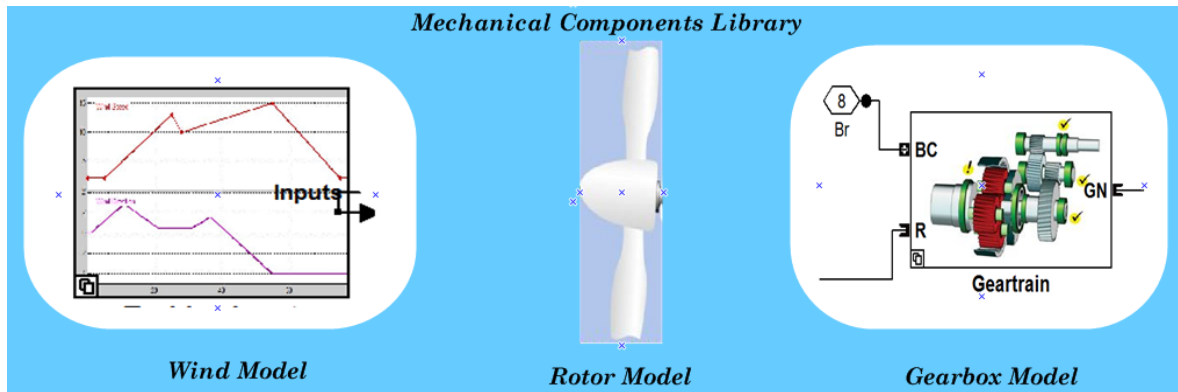
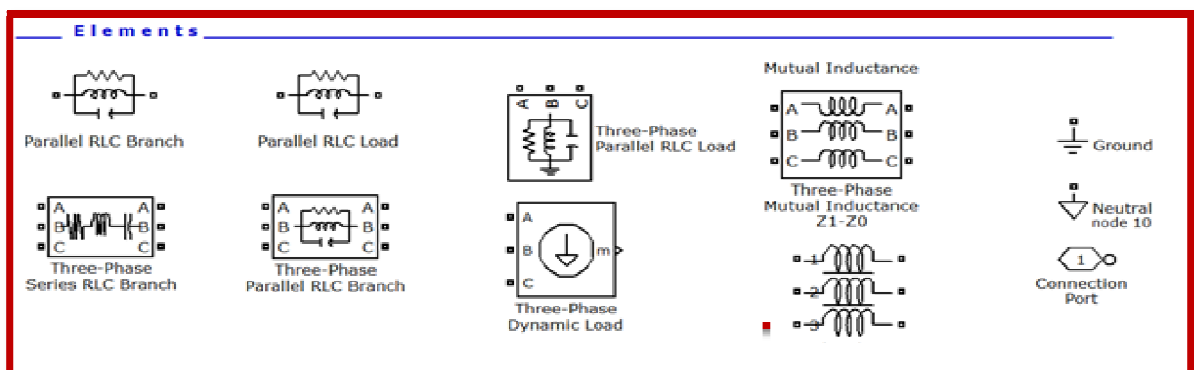


Figure 4- 2: Mechanical Components Library

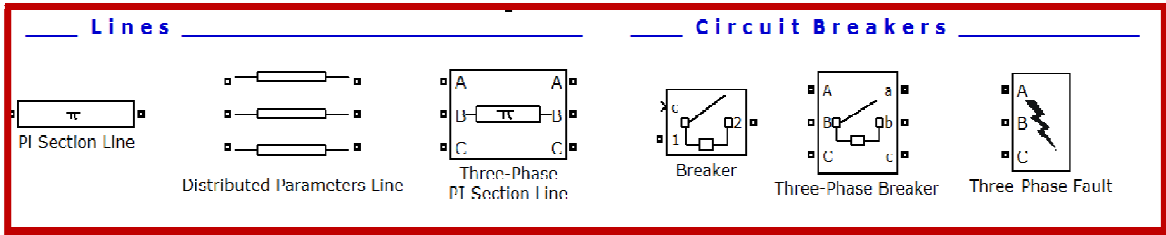
4.1.3 Electrical Machinery Library

The Electrical Machinery Library contains models with different level of detailing for electrical machines used currently in wind turbine systems as shown in Figure 4- 3. The standard models as well as the new models for squirrel cage induction machine, wound rotor induction machine, salient-poles synchronous machine and permanent magnet synchronous machines have been included in this library.

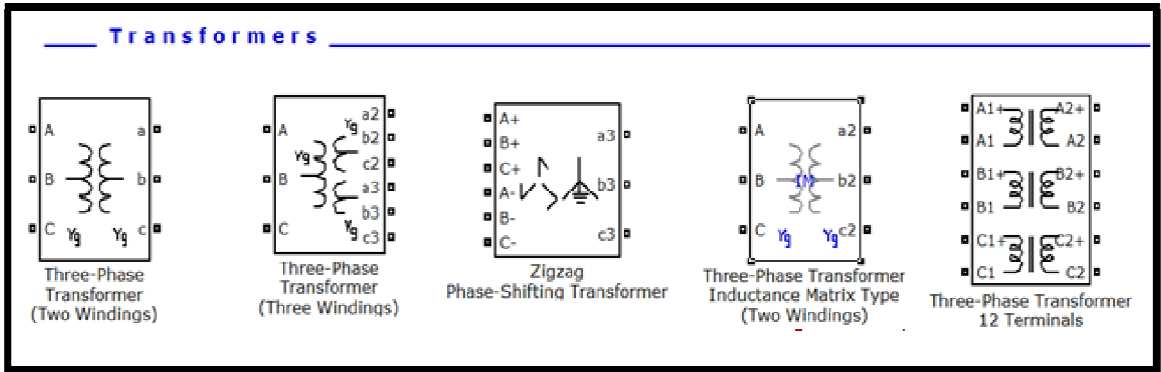
Since all these electrical machines models involve a relatively big number of derivatives for the state variables, two methods have been chosen to implement the models in Simulink: using Simulink blocks and C S-Function.



(a) Electrical Elements



(b) Lines and Circuit Breakers



(c) Transformers

Figure 4- 3: Electrical Machinery Library

4.1.4 Power Converters Library

The third library contains models for power converters based on switching functions. At the moment the following models are available: 3-phase diode bridge rectifier, voltage source converter (VCS), soft-starters and different modulation strategies for power converters, as shown in Figure 4- 4.

Using two VSC blocks connected via a DC-link circuit a back-to-back converter topology can be obtained. The connection type of the soft-starter can be selected from the mask interface, e.g. star connection, delta or branch delta connection.

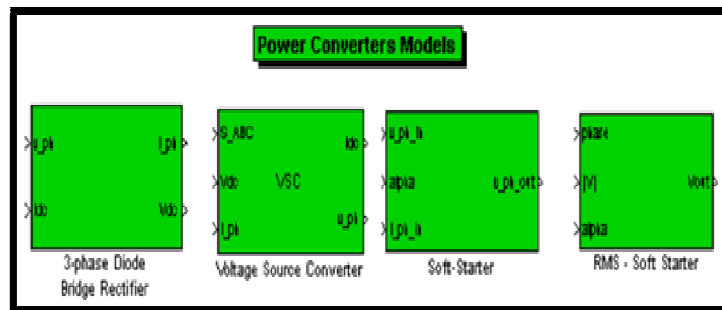


Figure 4- 4: Content of the Power Converters Library

Two types of modulation strategy for the Voltage Source Converter are available, namely sinusoidal pulse width modulated (PWM) and Space Vector both average and switching models. The switching models include minimum pulse width, dead time compensation and pulse dropping.

4.1.5 Common Blocks Library

This library contains models related with the grid connection of the wind turbines as shown in Figure 4-5. Some relevant models from this library are:

- Transformer model, which takes the core geometry as well as the iron losses into account;
- Distribution line model is based on the dynamic equations of the short line without distortion, which is a relatively simple model with good results in the case of a distribution line;
- Switch model (on-off) of the circuit breaker, which takes different opening time moments for poles into account;

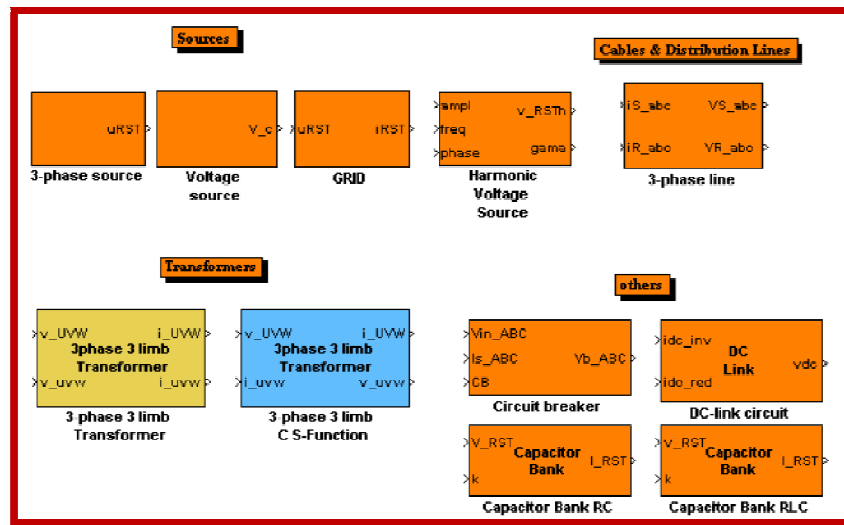


Figure 4- 5: Common Blocks Library

4.1.6 Transformations Library

As in a dynamic model of a wind turbine there are several transformations for the signals. A special library which contains the main transformations has been implemented as shown in Fig 4- 6.

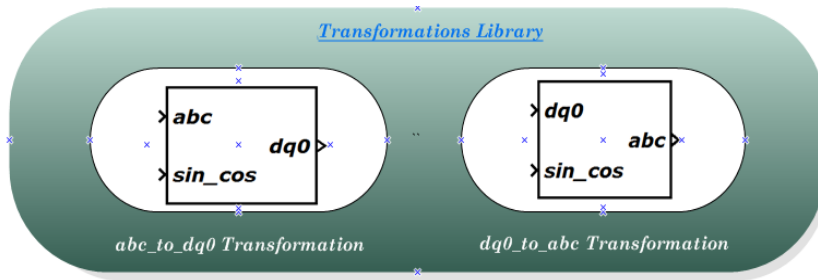


Figure 4- 6: Content of the Transformations Library.

4.1.7 Measurements Library

This library contains some special blocks as calculation of the period for a sinusoidal variable, calculation of the grid angle using a Phase Locked Loop system, calculation of the average wind for a given time interval, different modes of active and reactive power calculation, measurement of voltage current, current and impedance as shown in Figure 4- 7.

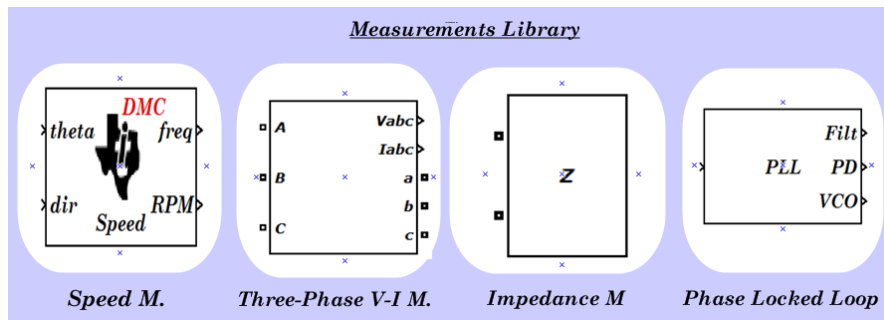


Figure 4- 7: Measurements Library

4.1.1 Control Library

This library contains models related with the control of the wind turbine system. Since the main component of a control scheme is the PI Controller, a model with an anti wind up structure has been implemented in Simulink.

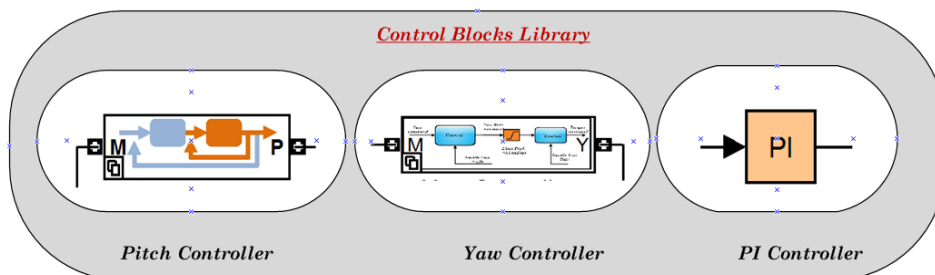


Figure 4- 8: Content of the Control Blocks Library

Chapter Five

5.1 Overview of the Wind Turbine Simulink Model

The power output of wind turbine is regulated by blade pitch control, in which the blades are rotated about their axis to adjust their aerodynamic angle of attack with respect to the relative wind. The rotor is pointed into the wind by rotating the Nacelle about the tower axis, which is referred to as Yaw control. Wind direction sensors on the Nacelle tell the Yaw controller where to point the rotor. Torque and speed sensors on the generator and drive train enable the blade-pitch controller to regulate the power output and rotor speed to prevent overloading structural components. The turbines generally start operating in winds at about 4 m/s and reach their rated power output at about 14.8 m/s. At about 25 m/s the control system will shut down the turbine by feathering the blades (pitching it to stop rotation) and applying brake in the gear train..

The Simulink model of wind turbine is composed of:

1. Wind input
2. Blade load lift and drag
3. Main controller
4. Blade = which is modeled as rigid body
5. Nacelle = which has different subsystem:
 - Pitch system
 - Gear train
 - Generator
 - Yaw system and
 - Different mechanical linkage
6. Tower
7. Grid connection.

The wind speed and direction imported in the Turbine Input block as signal as shown in the figure 5-1, these two signals (represented as “2”) moves in to the Blade Load and Main Controller block.

In the Blade Load block: the signals used to determine deferent aerodynamic load of the system. The aerodynamic load which determined from the Blade Load block will apply to the three blades (BL1, BL2 and BL3) .where BL1: Blade load 1.

In the Blades block the blade loads are integrate with the airfoil profile of the blades. As a result, the structural strength and aerodynamic performance of the blade are determined.

B1, B2 and B3 ports are used for connecting the blade with the pitch actuating cylinder system in the Nacelle block. Where B1: Blade 1

PA1, PA2 and PA3 are Connection Port block transfers both the conserving and the physical signal connections to the outside boundary of a subsystem block. Where PA1: Port A1

In the main controller block: the two signals used to determine the turbine state of the machine: the start up, generating, brake and parking state of the machine. It depends on the magnitude and direction of the wind.

PC: is the pitch command which vary based on the wind speed magnitude. It further optimized in the pitch controller block to send better response command in to the pitch actuating cylinder system in the Nacelle block.

YC: is yaw command which vary based on the wind direction. It further optimized in the yaw controller block to send better response command in to the yaw actuating system in the Nacelle block.

Br: is a brake command which, used to engage and disengage the frictional clutch in the gear train, activated when the wind speed is reach cut out speed or below cut in.

G: is generator command which initiate the generator when wind speed is at nominal speed.

In Figure 5- 1, scope blocks, signal viewers, test points and data logging provide ways to display and capture results from the simulations.

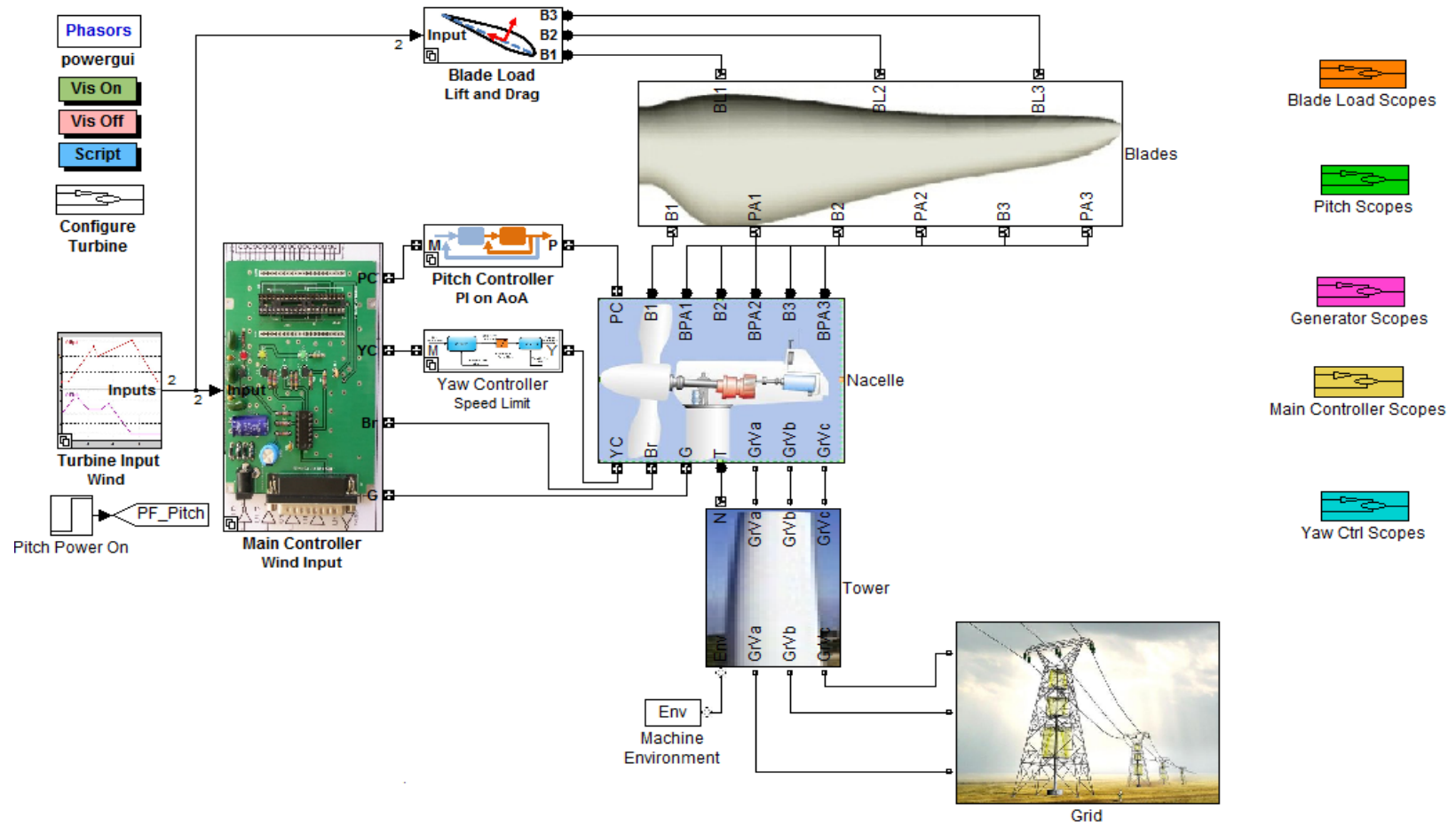
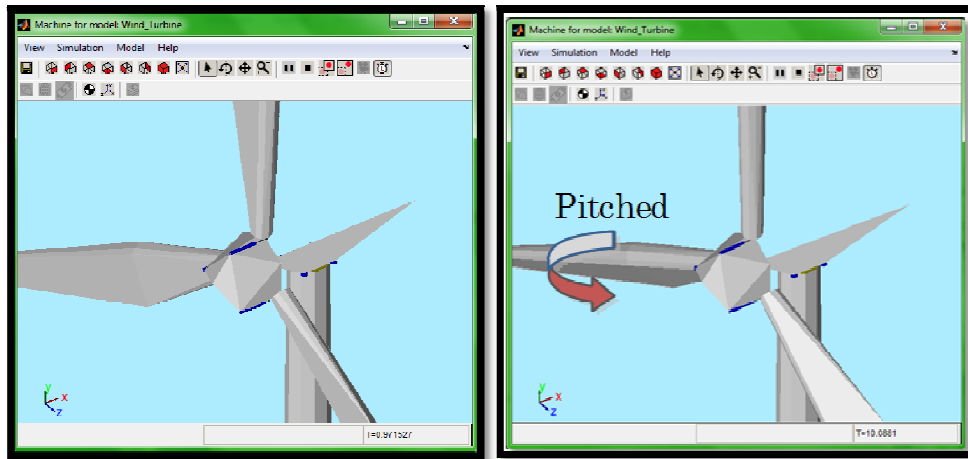


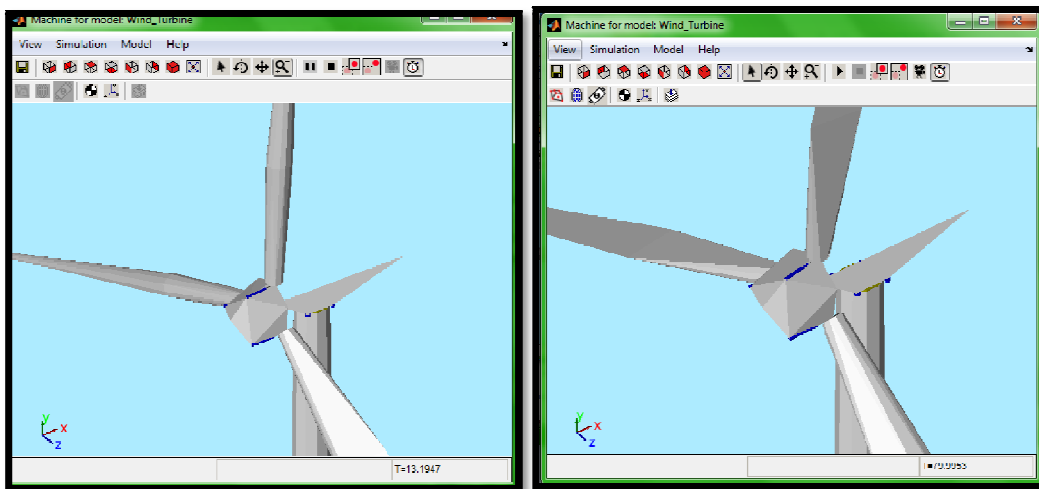
Figure 5- 1: Wind Turbine Overall System

There is also a 3D animation which animates automatically from the model as shown in Figure 5- 2. Consider simulation of the whole system.



(a)

(b)



(c)

(d)

Figure 5- 2: 3D Animation

In Figure 5-3 shows the wind speed and direction simulation of turbine input. The three scopes in Figure 5-4, 5-5 and 5-6 are used to show the operating conditions of the wind turbine model based on the turbine input.

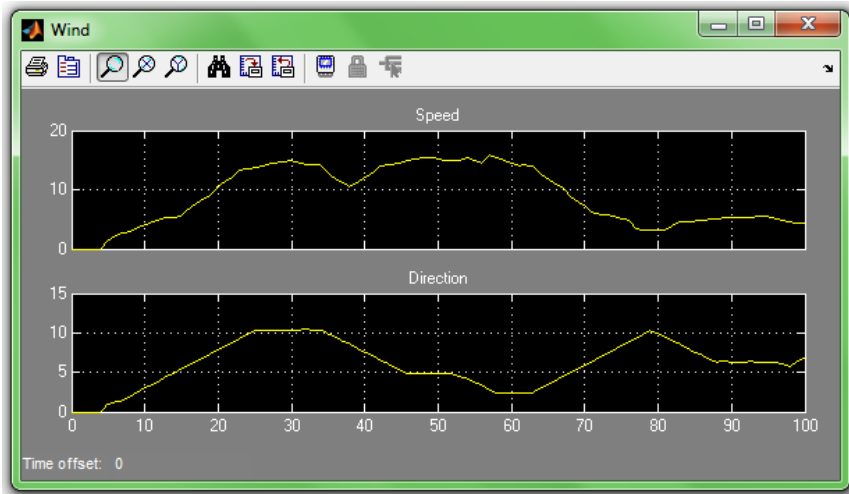


Figure 5- 3: Wind Speed and Direction

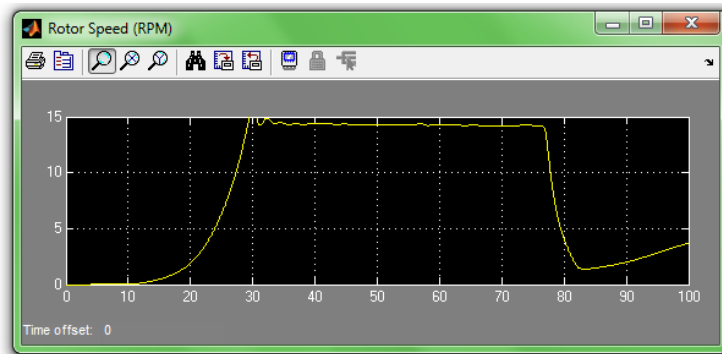


Figure 5- 4: Rotor Speed (RPM)

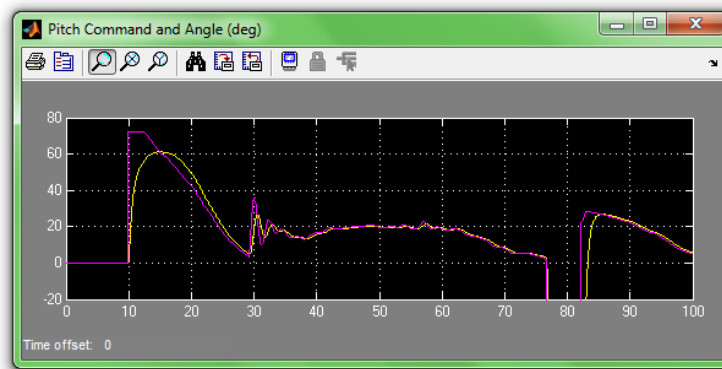


Figure 5- 5: Pitch Angle and Command (Deg)

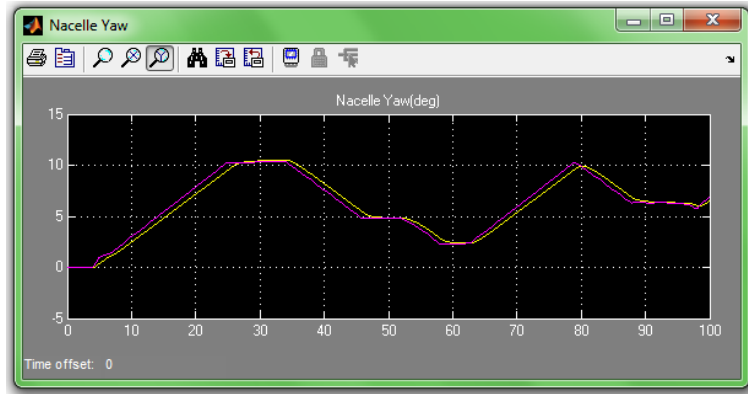


Figure 5- 6: Nacelle Yaw

At the start of simulation the wind speed is too low as shown in Figure 5- 3, so the supervisory control makes the turbine in parking state, Figure 5- 2(A).

Once the wind speed accelerates to a high enough speed, the supervisory control will startup the wind turbine. The blade pitched into the wind direction in order to catch the wind and the blades generate lift force. Therefore the rotor starts to rotate, (Figure 5- 2(B)).

Adopting a large positive pitch angle, as shown in Figure 5- 5, is used to generate a large starting torque which makes the rotor spin (Figure 5- 2(c)). At 90° of positive pitch, the blade is said to be ‘feathered’.

In Figure 5- 4 the rotor speed is accelerated until nominal speed. At nominal speed, the pitch of the blade adjusts itself to spin the wind turbine at the constant speed as shown Figure 5- 5.

Now that the wind turbine is at the normal operation, the wind direction change the Yaw angle of the Nacelle adjusting itself in the wind direction as shown in Figure 5- 6.

At the end of Figure 5- 3, the wind speed is dropping below cut in. Once the wind speed passes this level the supervisory control stops the turbine. As a result, the turbine is pitched away from the wind direction as shown in the Figure 5- 2 (D).

A negative 90° pitch angle is usually used when shutting down the system because this minimizes the rotor idling speed at which the parking brake is applied (Figure 5- 2(D)).

Chapter Six

6.1 Modeling Subsystem of the Wind Turbine

In this chapter the structure of new MATLAB/Simulink subsystem models of wind turbine applications is presented. The content of the subsystem of wind turbine is classified as:

- 1 Wind input to the system
- 2 Blade
- 3 Nacelle
- 4 Tower
- 5 Machine Environment
- 6 Main controller
- 7 Pitch controller
- 8 Yaw controller

6.2 Wind Input to the System (Turbine Input)

The wind system is modeled to import wind speed and direction data from Excel worksheet, MATLAB window or directly from data logger using signal builder as signal to the system. Wind inputs to this model include:

- Steady wind, including vertical and horizontal gradients
- Turbulent wind in time series form
- Wind direction as a function of time.

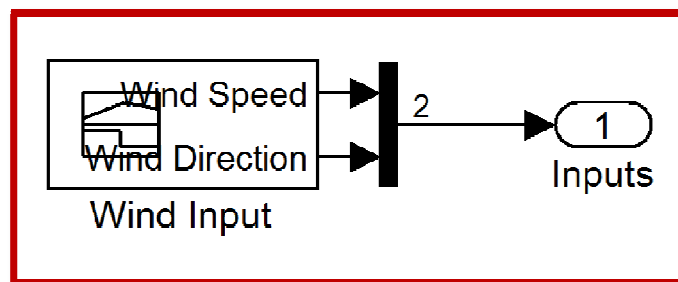


Figure 6- 1: Signal Builder Block

Data type: The system within the signal builder uses 2 types of input data wind speed and direction. Figure 6- 3 is zoomed in plot of Figure 6- 2 to show the detail performance of the wind turbine under various conditions.

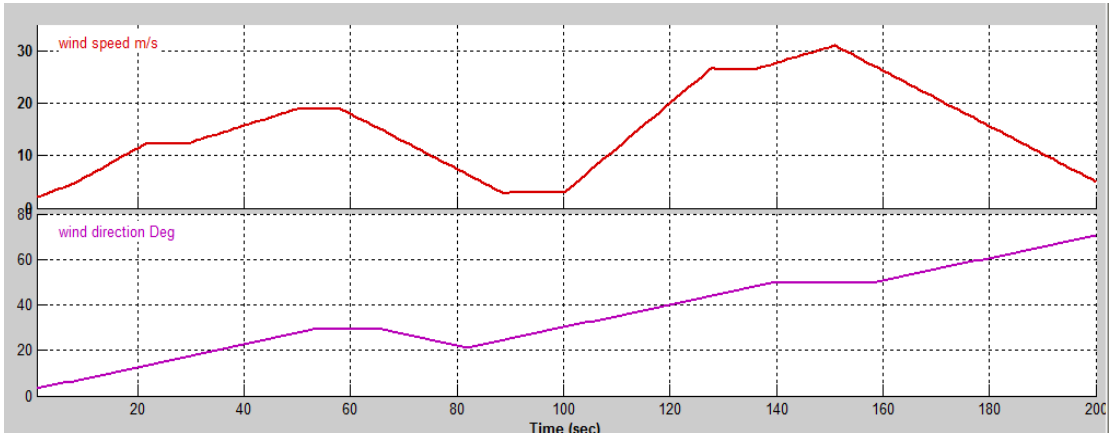


Figure 6- 2: Wind Speed and Direction

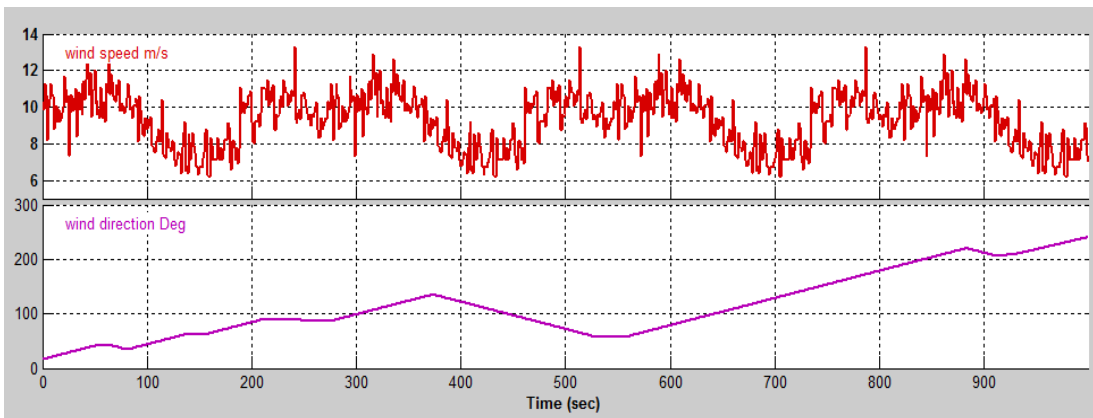


Figure 6- 3: Wind Speed and Direction (Full)

Wind direction is sensed by a wind vane on top of the Nacelle and monitored by the automatic Yaw control system. In this model, wind direction is represented in Table 1.

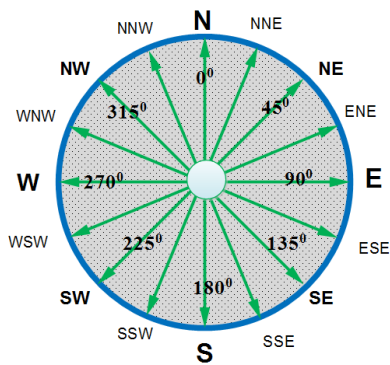


Figure 6- 4: Wind Direction Indicator Diagram

Table 1 Representative Angle of the Wind Direction

Polar Name	Representative Angle
N	0 ⁰
NNE	22.5 ⁰
NE	45 ⁰
ENE	67.5 ⁰
E	90 ⁰
ESE	112.5 ⁰
SE	135 ⁰
SSE	157.5 ⁰
S	180 ⁰
SSW	202.5 ⁰
SW	225 ⁰
WSW	247.5 ⁰
W	270 ⁰
WNW	292.5 ⁰
NW	315 ⁰
NNW	337.5 ⁰
N	360 ⁰

Figure 6-4 represents how the wind direction is interpreted in the model. Example in Figure 6- 2:

- Wind direction 56.25⁰ means the wind is coming into the turbine in the NE–ENE direction as shown in the Figure 6- 4.

Integrating the wind direction into the model is helpful in determining the orientation of the wind turbine towards the wind. The Yaw drive is used to keep the rotor facing into the wind as the wind direction changes.

6.3 Aerodynamic Loads (Blade Lift and Drag)

The aerodynamic design of a wind turbine blade defines its width, thickness, direction and profile. It is a matter of finding the best compromise between air flow and strength (profile and structure). The purpose is to optimize performance while minimizing loads.

When the wind strikes the wind turbine, it produces a moment which makes the wind turbine to spin. Modeling the load on the blade due to the wind load is done by using Simulink and embedded MATLAB to create the model at varying levels of details. There are two ways of modeling force of the wind on blade of the wind turbine.

i. Single element method

It uses the blade as one single element. This is a very simple model for interaction between the wind and the blade.

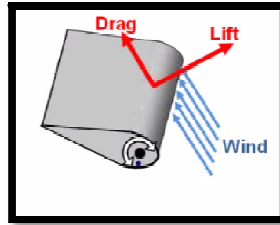


Figure 6- 5: Blade as Single Element

ii. Multiple element method

The blade divided into a series of elements. Then the lift and drag force are calculated at different radius for given wind speed and direction.

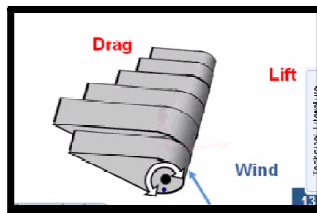


Figure 6- 6: Blade as Multiple Elements

The application of momentum theory to a blade element enables the aerodynamic forces on the blade to be calculated at different radii.

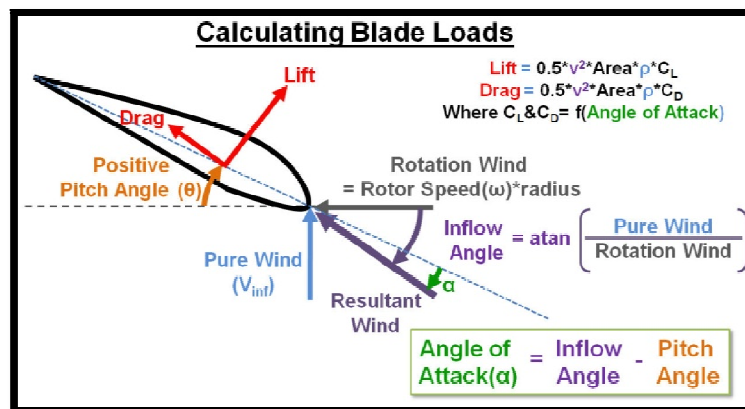


Figure 6- 7: Blade Loads

The forces acting on blade can be resolved into stream wise (drag) and normal (lift) components. The drag on a body immersed in an oncoming flow is defined as the force on the body in a direction parallel to the flow direction.

The Simulink model of the above blade load diagram is shown in the Figure 6- 8.

$$\text{Rotation wind (angular velocity)} = \text{rotor speed } (\omega) * \text{radius} \quad (6.1)$$

$$\text{Inflow angle} = a \tan\left(\frac{\text{pure wind}}{\text{rotational wind}}\right) \quad (6.2)$$

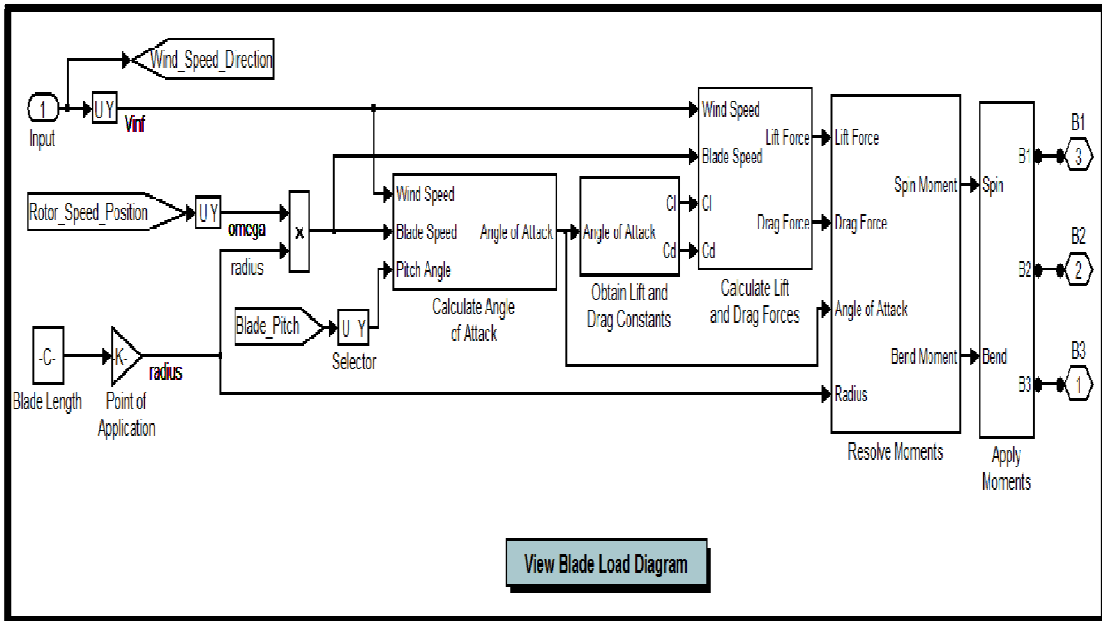


Figure 6- 8: Simulink Representation of Blade Loads

Angle of attack is the angle between air flow incident on the blade and the blade chord line.

$$\text{Angle of attack} = \text{inflow angle} - \text{pitch angle} \quad (6.3)$$

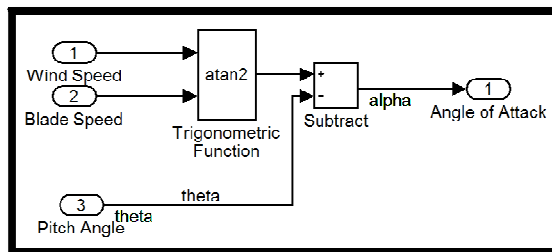


Figure 6- 9: Simulink Representation of the Angle of Attack

Maximum blade loadings are in the out-of-plane direction and occur when the wind direction is either approximately normal to the blade, giving maximum drag, or at an angle of between 12° and 16° to the plane of the blade when the angle of attack is such as to give maximum lift.

where: C_L and $C_D = f(\text{Angle of attack})$ (6.4)

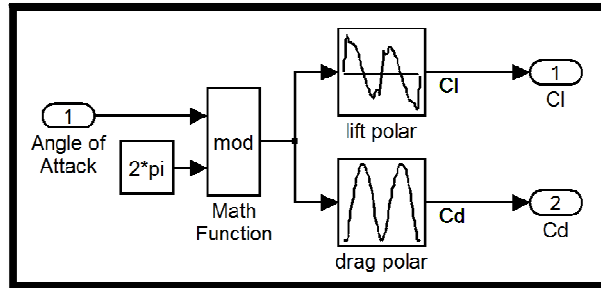


Figure 6- 10: Lift and Drag Coefficient

$$\text{Lift} = 0.5v^2 A \rho C_L = \frac{1}{8} \rho \pi d C_L \rho u d \quad (6.5)$$

$$\text{Drag} = 0.5v^2 A \rho C_D = \frac{1}{8} \rho \pi d d \rho C_D = \frac{1}{8} \rho \pi d C_D \rho u d \quad (6.6)$$

$\rho u d / \mu$ is known as the Reynolds number (Re) and represents the ratio of the inertia force acting on a unit volume of fluid, as it is accelerated by a pressure gradient and the viscous force on the same volume of fluid which is resisting the motion of the fluid. The nature of the flow pattern around an aerofoil is determined by the Reynolds number and this significantly affects the values of the lift and drag coefficients. The turbine blade changes its angle of attack as it makes its orbit about the rotational axis, Reynolds number changes also.

Lift and drag coefficients are obtained from the generated aerofoil data from selected NACA 8412 blade profile.

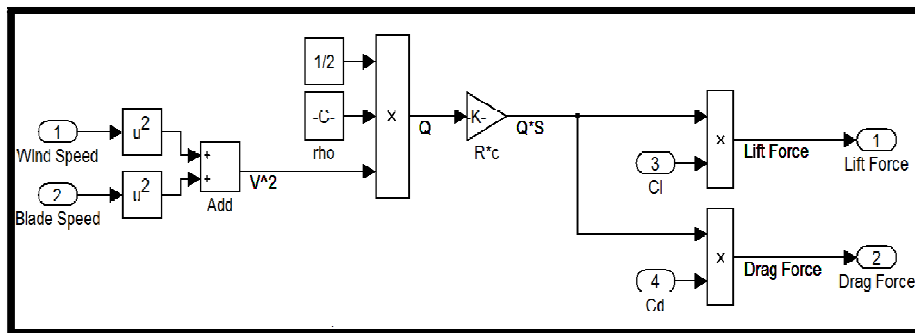


Figure 6- 11: Lift and Drag force

Then using the above values of lift and drag forces, the bend moment and spin moment on the blade modeled in Simulink environment is represented as follows in Figure 6- 12.

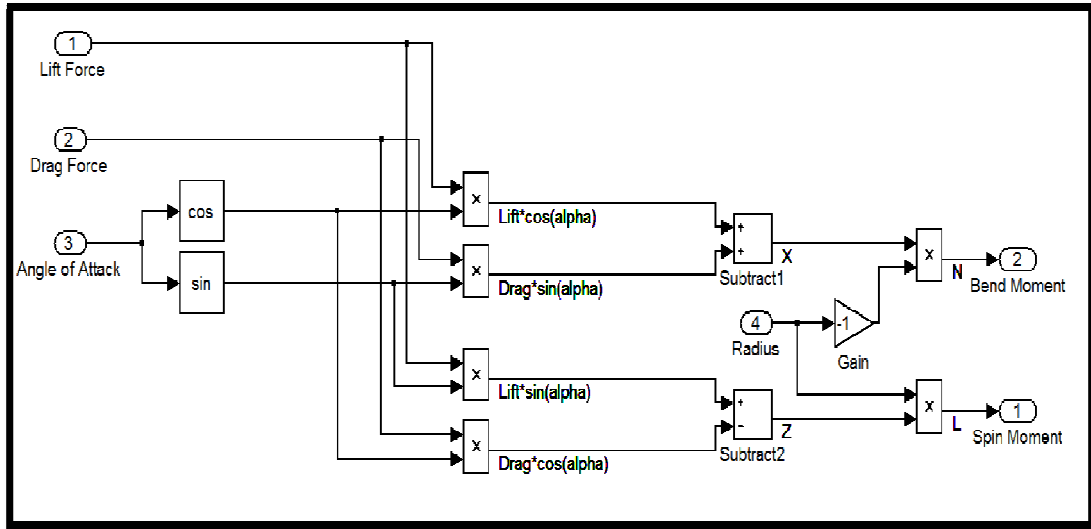


Figure 6- 12: Moments on the Blade

Finally these moments will be applied to each blade (B1, B2 and B3). As shown in the Figure 6- 13.

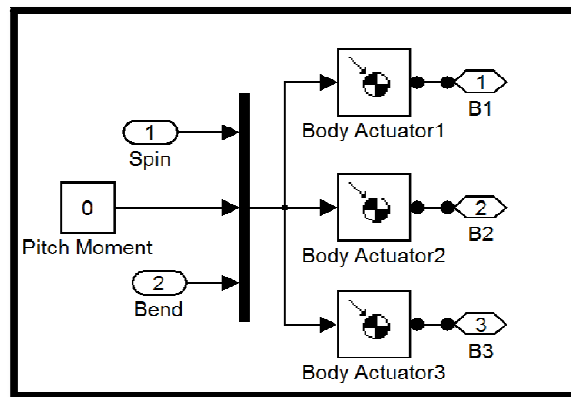


Figure 6-13: Blade Actuator

All the above aerodynamics load subsystems are shown as a single embedded system in Figure 6-14.

The embedded MATLAB parameters inside the rotor disk aerodynamics load in Figure 6- 14 are expressed in Appendix C.

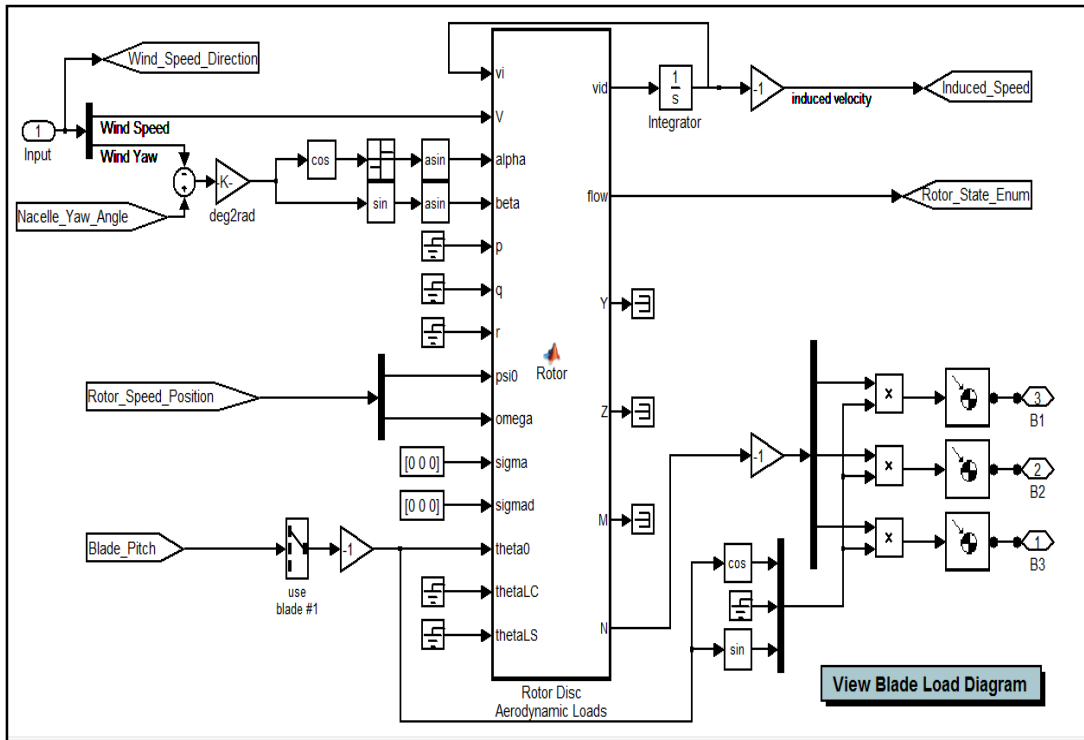


Figure 6-14: Embedded Systems of Aerodynamic Loads

6.4 Blade

The blade design process can be divided into two stages: the aerodynamic design and the structural design. The aerodynamic design addresses the selection of the optimum geometry of the blade. The blade geometry is defined by the aerofoil family and the chord, twist and thickness distributions. The structural design consists of blade material selection.

6.4.1 Aerodynamics Design

The shape of the aerodynamic profile is decisive for blade performance. Blade profiles are previously chosen from a widely used catalogue of airfoil profiles developed in wind tunnel research by NACA. The selected NACA four-digit aerodynamics profile of the blade airfoil: [NACA 8412](#) is developed in the model.

The NACA four-digit airfoil profile is developed using analytical equations that describe the camber (curvature) of the mean-line (geometric centerline) of the airfoil section as well as the section's thickness distribution along the length of the airfoil as illustrated below.

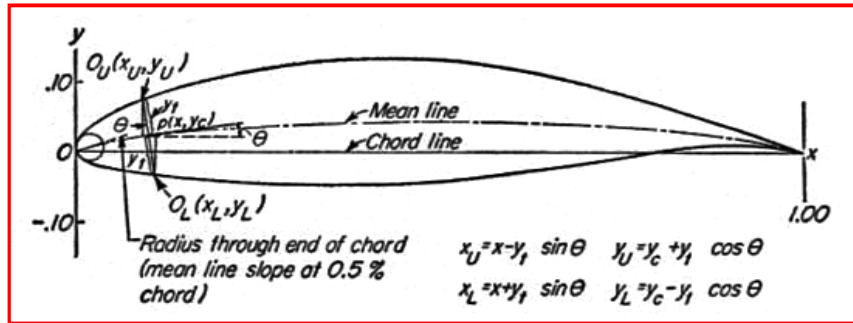


Figure 6- 15: NACA four-digit airfoil profile

In this model based design, the selected airfoil profile is NACA8412.

Table 2 NACA 8412

NACA8412	
M	8%
P	4/10
T	12%

where:

- m**: Maximum camber as a percentage of chord
- p**: Chord wise Position of the maximum camber as
- t**: Maximum thickness as a percentage of chord

The first digit, when multiplied by 0.15, gives the designed coefficient of lift (c_L) as shown in the Appendix G.

The following equations are used by NACA wind turbine blade airfoil developers. The project used these equations from NACA and developed in its own Simulink environment.

Utilizing these m, p and t values, it computed the coordinates for an entire airfoil using the following relationships:

- Pick values of x from 0 to the maximum chord c.
- Compute the mean camber line coordinates by plugging the values of m and p into the following equations for each of the x coordinates.

The equations are:

$$y_c = \left(\frac{m}{p^2} \right) (2px - x^2); \text{ for } x \leq p \quad (6.7)$$

$$y_c = \left(\frac{m}{(1-p)^2} \right) ((1-2p) + 2px - x^2); \text{ for } x > p \quad (6.8)$$

where: $c =$ Chord length,

$x =$ Coordinates along the length of the airfoil, from 0 to c (which stands for chord, or length)

$y =$ Coordinates above and below the line extending along the length of the airfoil, these are either y_t for thickness coordinates or y_c for camber coordinates.

- Calculate the thickness distribution above (+) and below (-) the mean line by plugging the value of t into the following equation for each of the x coordinates.

$$y_t = (t/0.2)(0.2969\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \quad (6.9)$$

The final coordinate for the upper surface (x_U, y_U) and lower surface (x_L, y_L) are given by:

$$x_U = x - y_t(\sin\theta) \quad (6.10)$$

$$y_U = y_c + y_t(\sin\theta) \quad (6.11)$$

$$x_L = x + y_t(\sin\theta) \quad (6.12)$$

$$y_L = y_c - y_t(\sin\theta) \quad (6.13)$$

where

$$\theta = \arctan(\Delta y_c / \Delta x)$$

All the above equations are embedded in Simulink environment as shown in Appendix D. The embedded Simulink program generates the NACA 4 digit coordinates with desired number of panels (line elements) on it as shown in Figure 6- 16:

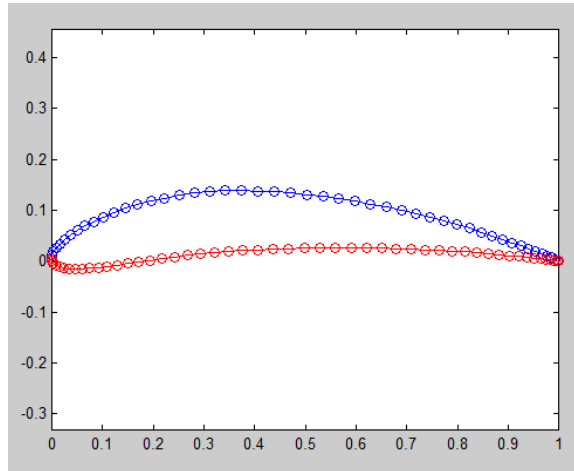


Figure 6-16: Generated NACA 8412 Airfoil Profile

Appendix E shows how the blade profiles have been integrated in Simulink environment from analytical equations. Data points for each surface (upper & lower) are shown in Appendix F.

This profile has good all-round properties, giving a good power curve and a good pitch. The blade is tolerant of minor surface imperfections, such as dirt on the blade profile surface.

6.4.2 Structural Design

The structural design consists of blade material selection and the determination of a structural linkage. The actuation force signals and blade linkages connected in to the blade using the port CS1 CS2 and CS6 as shown in the Figure 6- 17.

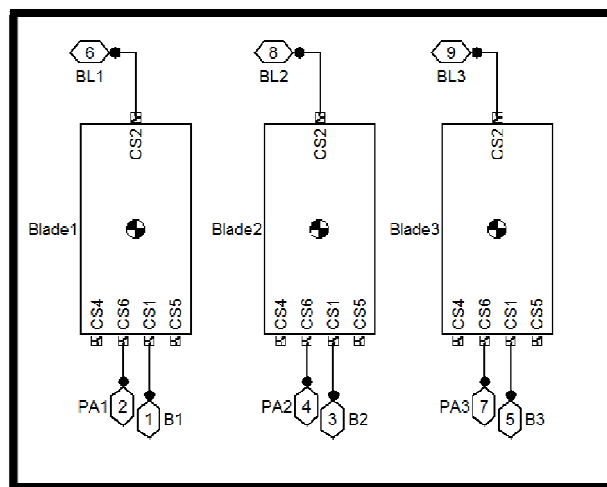


Figure 6- 17: Blade

The blade in the model represents as user-defined rigid body (blade defined by mass m , inertia tensor I and coordinates origins and axes for center of gravity (CG) and other user-specified body coordinate systems. This dialog sets body initial position and orientation, unless body and/or connected joints are actuated separately. This dialog box also provides optional settings for customized body geometry and color.

The body block represents a rigid body with properties that are customized. The representations that specify include:

- The body's mass and moment of inertia tensor
- The coordinates for the body's center of gravity (CG)
- One or more body coordinate systems (CSs).

A rigid body is defined in space by the position of its CG and its orientation in some defined coordinate system.

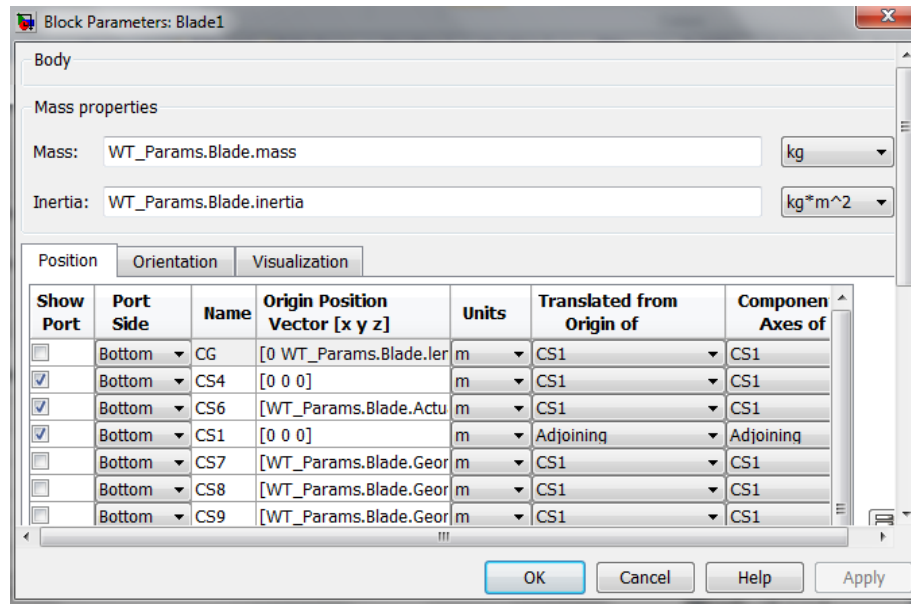


Figure 6- 18: Blade body block parameter

6.4.3 Defining a Body with Mass, Coordinate System and Visualization Properties Mass Properties

The mass properties are defined by the body's mass and inertia tensor as shown in Figure 6- 18. The mass is the body's inertia and controls the translational acceleration of the CG in response to an applied force. The inertia tensor measures the distribution of mass density in the body and controls the rotational acceleration of the body about the CG in response to an applied torque.

The components of the inertia tensor control the initial orientation of the body and are always interpreted as being in the CG-CS axes. The orientation of the CG-CS axes with respect to another CS external to the body (the World CS, a CS on a ground, or a CS on another body) then determines the orientation of the body with respect to other bodies or with respect to world.

Coordinate System Properties

The coordinate system properties are defined by the body's body CSs. The CS with its origin at the CG is required. The CG point specifies both the initial position of the whole body and the origin of the CG-CS and place one or more additional Body CSs on a body. The body dialog requires at least one. Then define each body CS by the position of its origin and the orientation of its CS axes.

Position

The Position fields for each body CS specify the position of that CS's origin as a translation vector:

- The numerical components of the vector carry units.
- The translation vector's components are oriented with respect to another set of CS axes.
- The origin is displaced from the origin of another.

Show Port	Port Side	Name	Origin Position Vector [x y z]	Units	Translated from Origin of	Components in Axes of
<input type="checkbox"/>	Bottom	CG	[0 WT_Params.Bla	m	CS1	CS1
<input checked="" type="checkbox"/>	Bottom	CS4	[0 0 0]	m	CS1	CS1
<input checked="" type="checkbox"/>	Bottom	CS6	[WT_Params.Blade	m	CS1	CS1
<input checked="" type="checkbox"/>	Bottom	CS1	[0 0 0]	m	Adjoining	Adjoining
<input type="checkbox"/>	Bottom	CS7	[WT_Params.Blade	m	CS1	CS1
<input type="checkbox"/>	Bottom	CS8	[WT_Params.Blade	m	CS1	CS1
<input type="checkbox"/>	Bottom	CS9	[WT_Params.Blade	m	CS1	CS1

Figure 6- 19: Position of the Blade

Orientation

The orientation fields for each body CS specify the orientation of that CS's triad of axes as a rotation:

- The orientation vector specifying the rotation vector has three components;
- The numerical components of the vector carry units;
- The rotation is oriented with respect to some other, pre-existing set of CS coordinate axes; and
- The orientation vector's components are interpreted in the convention of a rotation representation.

Show Port	Port Side	Name	Orientation Vector	Units	Relative CS	Specified Using Convention
<input type="checkbox"/>	Bottom	CG	[0 0 0]	deg	CS1	Euler X-Y-Z
<input checked="" type="checkbox"/>	Bottom	CS4	[0 0 0]	deg	CS1	Euler X-Y-Z
<input checked="" type="checkbox"/>	Bottom	CS6	[0 0 0]	deg	CS1	Euler X-Y-Z
<input checked="" type="checkbox"/>	Bottom	CS1	[0 0 0]	deg	Adjoining	Euler X-Y-Z
<input type="checkbox"/>	Bottom	CS7	[0 0 0]	deg	CS1	Euler X-Y-Z
<input type="checkbox"/>	Bottom	CS8	[0 0 0]	deg	CS1	Euler X-Y-Z

Figure 6- 20: Orientation of the Blade

Rotation Conventions

Euler angles mathematically represent a three-dimensional spherical rotation as a product of three successive independent rotations about three independent axes by three independent (Euler) angles.

The Euler angle convention specifies the rotation of the body CS axes by rotating about three axes in a sequence. The components of the 1-by-3 row vector are the angles of rotation about those three axes, respectively in sequence, in degrees.

For example, Euler X-Y-Z means rotate about the original X axis, then about the first intermediate Y axis and then about the second intermediate Z axis

Visualization Properties

The visualization properties of a body include its color and geometry (surface size and shape) of the model.

The above blade position, orientation and visualization representations are applied for each body of this model

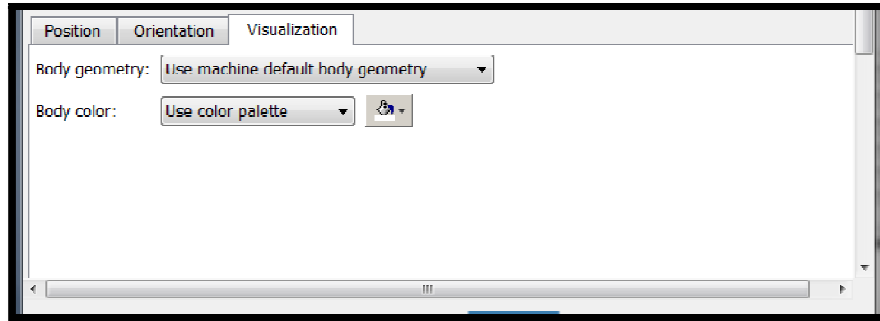


Figure 6- 21: Visualization of the blade

6.4.4 Modeling the Blade Pitch Linkage

The purpose of most wind turbines is to extract as much energy from the wind as possible and each component of the turbine has to be optimized for that goal. Optimal blade design is influenced by the mode of operation of the turbine, that is, fixed rotational speed or variable rotational speed and, ideally, the wind distribution at the intended site.

A turbine operating at variable speed can maintain the constant tip speed ratio required for the maximum power coefficient to be developed regardless of wind speed. To develop the maximum possible power coefficient a suitable blade geometry and better linkage of the blade with the remaining turbine parts are necessary is required.

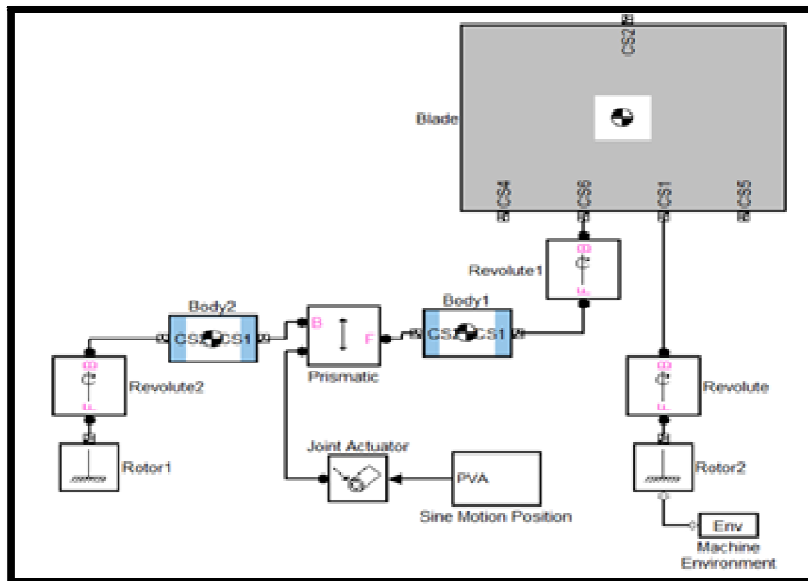


Figure 6- 22: Pitch Actuation Linkage

At the start up of the turbine state, the blade rotates at its own axis. To rotate the blade,

there are mechanical linkages which contract or extend. This will be done using SimMechanics under the Simulink environments as shown Figure 6-22.

The blade is represented as the rigid body and has one degree of freedom. Running the model results a 3D animation that is automatically generated from the model. The animation shows how the system is moving.

The revolute represents one rotational degree of freedom. The follower body (F) rotates relative to the base body (B) about a single rotational axis going through collocated body coordinate system origins. Sensor and actuator ports are added. Base-follower sequence and axis direction determine sign of forward motion by the right-hand rule.

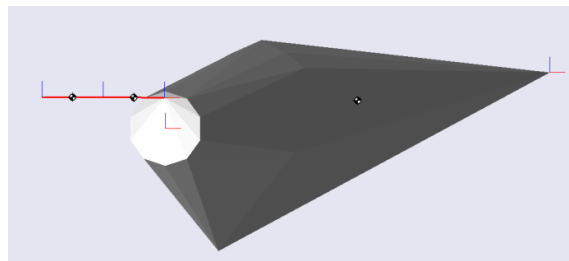


Figure 6- 23: 3D Animation of Blade

6.4.5 Modeling the Blade as Flexible Body

When simulating a mechanical system, it is supportive to include flexible bodies. To model this blade as a flexible body in SimMechanics, the project applies the most common flexible body approximation methods: the lumped-parameter approximation and the state space/frequency response method using finite element analysis (FEA) results.

I. Lumped Parameter Approach

The lumped parameter approach approximates a flexible body as a set of rigid bodies coupled with springs and dampers. It can be implemented by a chain of alternating bodies and joints. The springs and dampers act on the bodies or the joints. The spring stiffness and damping coefficients are functions of the material properties and the geometry of the flexible elements.

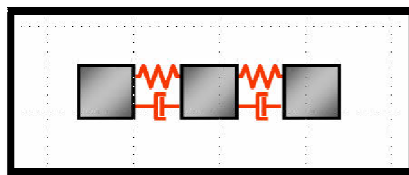


Figure 6- 24: Flexible Cantilever Lumped Parameter Approach

The simplifying assumptions of lumped element model in mechanical systems with its domain are:

- All objects are rigid bodies and
- All interactions between rigid bodies take place via kinematics pairs (joints) springs and dampers.

In this section, lumped parameter method is first applied to an abstract beam and then to a flexible cantilever.

There are three blades connected to the rotor. The force and torque are measured at the connection points of the blade and rotor using the joint sensor as shown in Figure 6- 25.

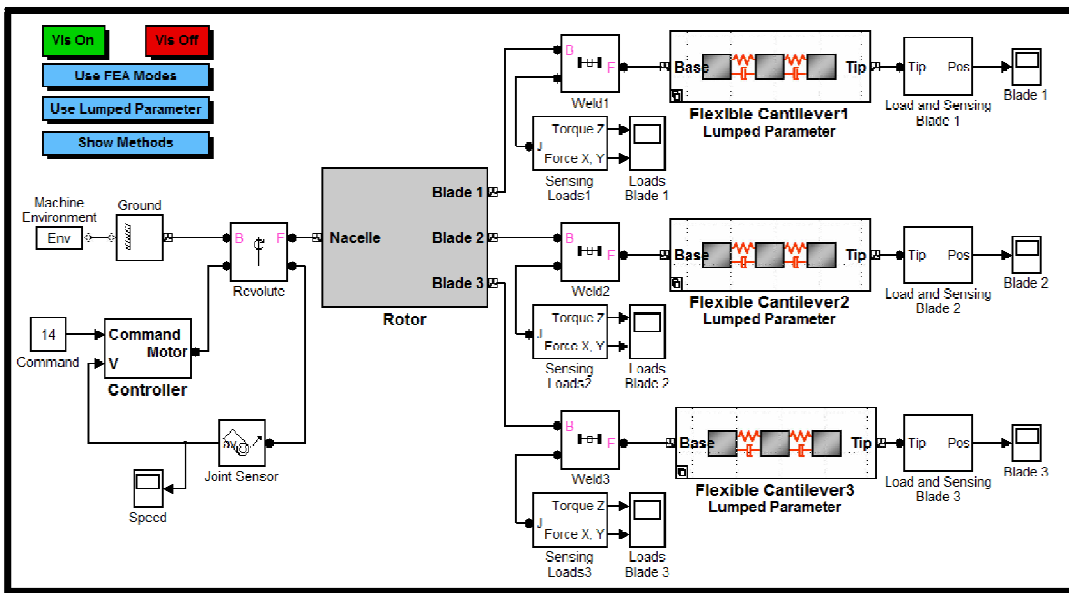


Figure 6- 25: Simulink Model of Lumped Parameter Approach

In the flexible cantilever subsystem, the beam is represented by a series of generalized beam elements (GBEs), each of which is a body-joint-body combination. This formulation actuates and senses the joints, taking advantage of the relative degrees of freedom inherent in SimMechanics.

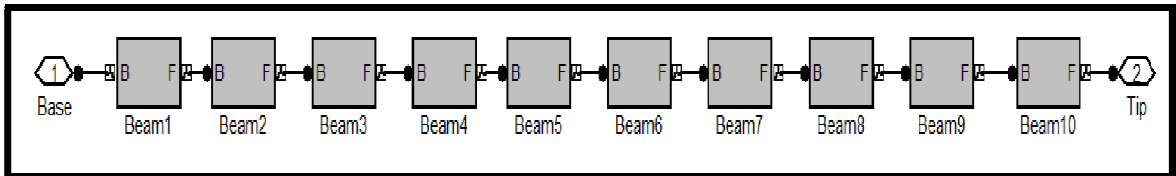


Figure 6- 26: Discretization of a Beam into Generalized Beam Elements

The subsystem of each beam element of the blade is connected by spring and damper as shown in the Figure 6- 27.

The lumped parameter method outlined here uses purely local forces as functions of local deflections. This approximation is useful for estimating the gross behavior and endpoint deflection of the beam but not for modeling local dynamics along the beam or the beam's vibration modes

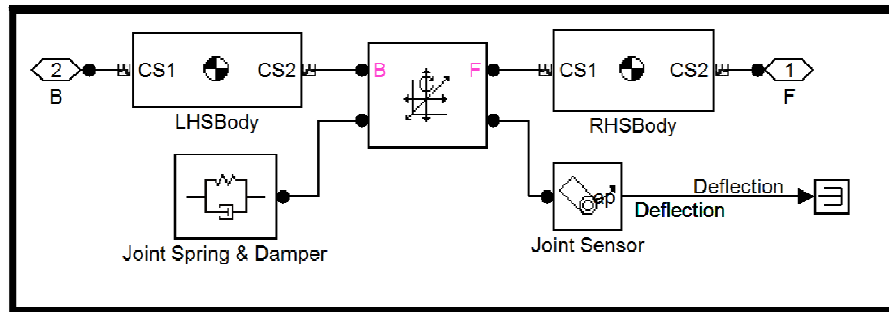


Figure 6- 27: Joint Spring and Damper

The restoring moment generated by bending the cantilever is proportional to its curvature. This method approximates the curvature by the local difference in slopes rather than by the correct nonlocal difference in slopes between neighboring GBEs, which cannot be represented in a series of independent elements.

While lumped parameter methods can be extended to more complicated systems, they are best suited for systems with chain like geometries. It is easier to model bodies with multidimensional geometries using other approaches, such as FEA.

II. Using Finite Element Analysis Results

A more powerful approach to modeling flexible bodies in SimMechanics is to first use an FEA program to obtain the frequency or modal response of the deformed bodies and then to incorporate this response into a SimMechanics model by superimposing flexible body deflection on rigid body motion. This approach involves three steps:

1. Model the flexible parts as rigid bodies using body blocks, making sure to insert a CS at each point to measure or actuate a flexible-body deflection.
2. At each of these CSs, connect another body through a joint that includes primitives corresponding to the deflections specified in the FEA model.
3. Use the appropriate loads on the body as input to this black box— the load force from one or more connected bodies or from external conditions.

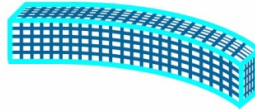


Figure 6- 28: Flexible Cantilever (FEA)

Let the blade be replaced by a cantilever beam. Then it will be done with both methods with SimMechanics to compare the two methods.

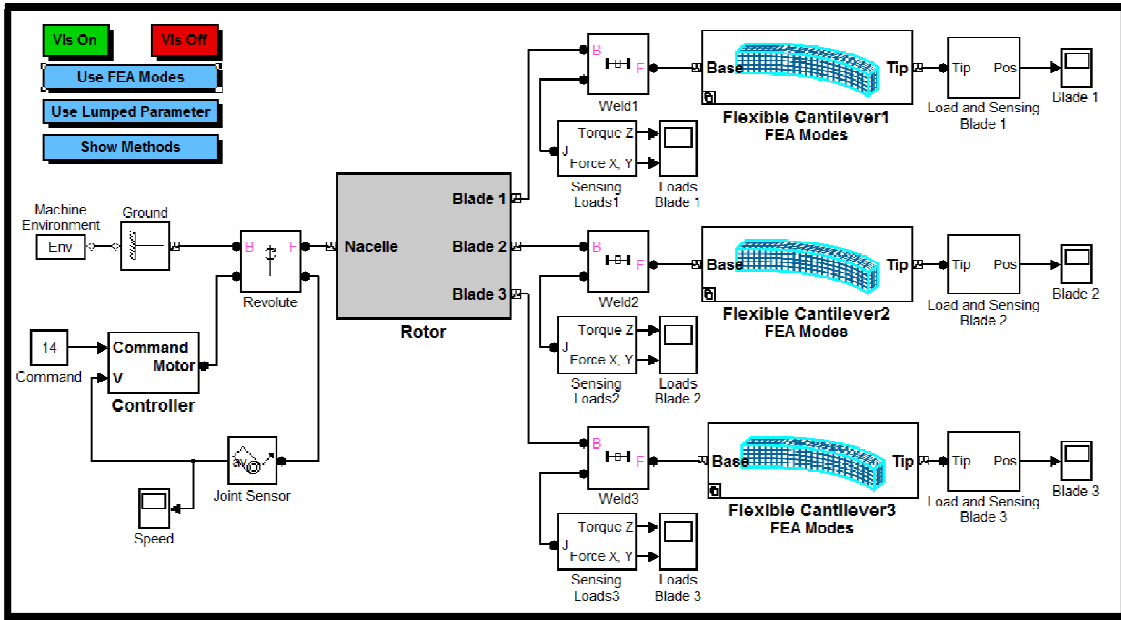


Figure 6- 29: Using Finite Element Analysis

The most direct way to implement the black box in Simulink is to use the State Space or LTI System block. Either block implements the flexible body dynamics in a state-space model and uses the results to actuate the motion of mass less bodies to which other SimMechanics blocks can be connected.

For the body modeled in finite element program it can export to eignmodes from finite element program import into Simulink environment.

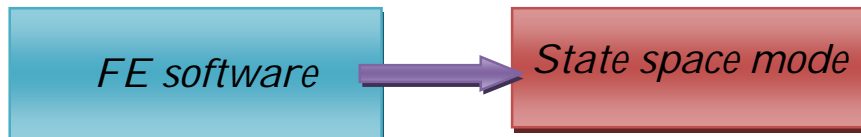


Figure 6- 30 FiniteElement to State Space Modal

To superimposing deflection due to flexible onto rigid body motion, the Figure 6-31 shows how it is done. The blade is assumed as a rigid body at the end of the rigid body put deflection joint and mass less body.

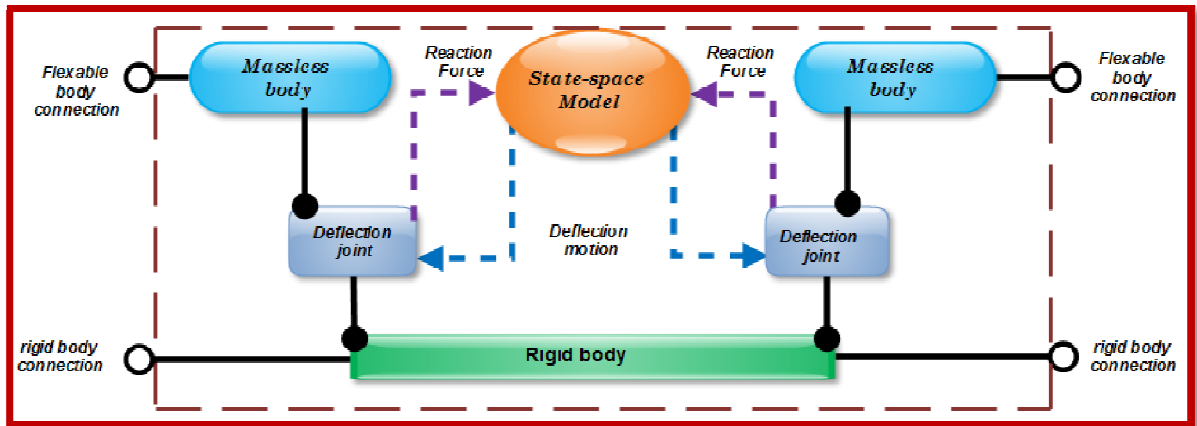


Figure 6- 31: Embedding State-Space Flexible Body Model

Deflection joints measure the reaction force into the steady space model. The state-space model will calculate the steady state motion and use the deflection motion to drive the deflection joint. In this way, the model superimposes the deflection due to flexibility on the rigid body motion.

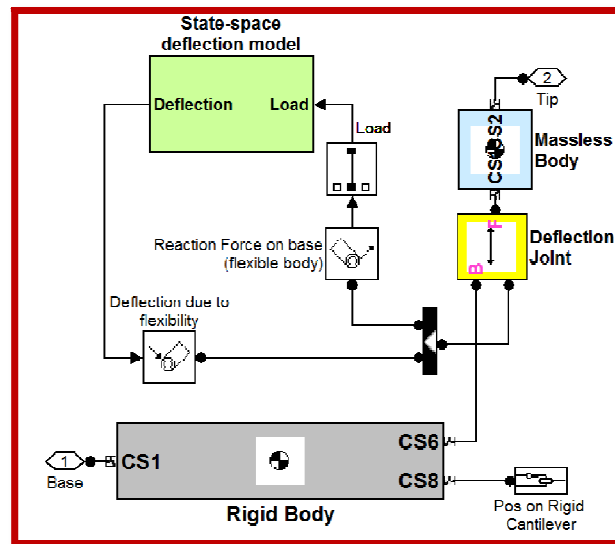


Figure 6- 32: Embedded state-space flexible body model in SimMechanics

. Figure 6- 32 shows how embedding of state-space flexible body model is done into a SimMechanics model in Simulink.

6.4.6 Comparing Methods and Results

The blade torque is shown in Figure 6- 33 when applying lumped parameters to generalized beam elements and a bending beam.

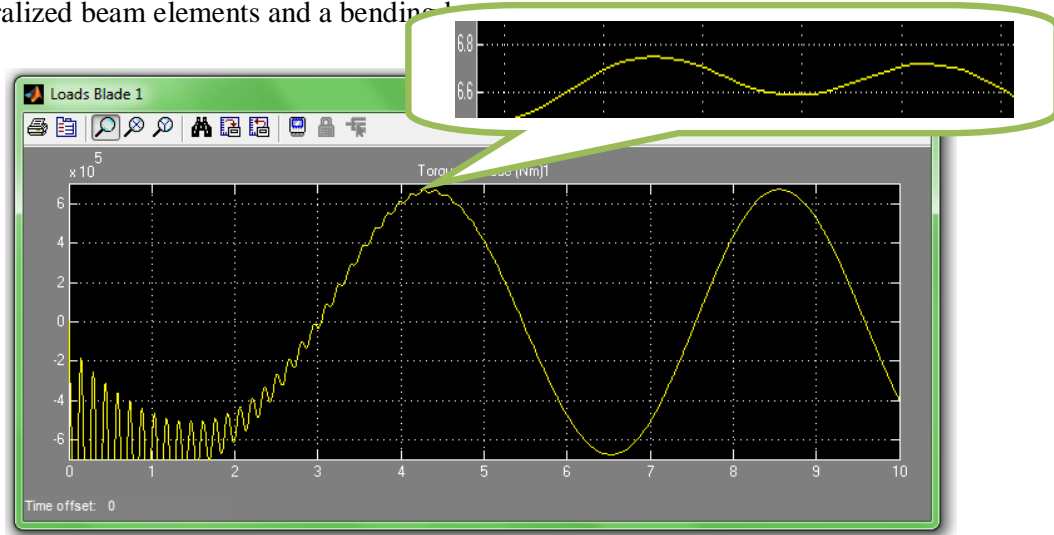


Figure 6- 33: Motion of Body-loaded Beam, Modeled by SimMechanics with 10 GBEs

The blade base torque is shown in Figure 6- 34 when applying Finite Element analysis results to generalized beam elements and a bending beam.

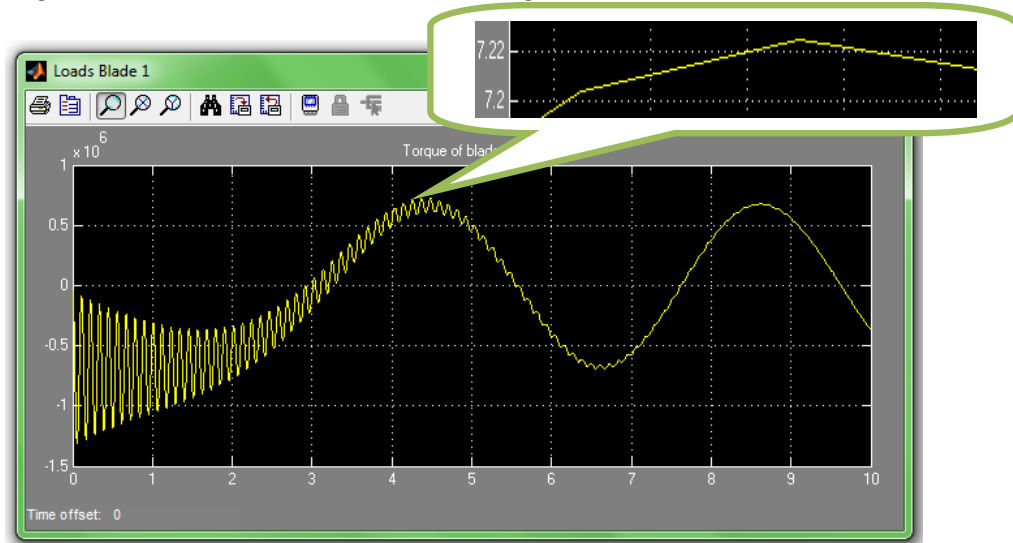


Figure 6- 34: Motion of Body-loaded Beam, Modeled by FEA and State Space Dynamics

With a reasonable number of discrete elements, the lumped parameter method gives accurate results for static bending, but the FEA method gives more accurate results for

vibration modes and frequencies,

Each method has advantages and disadvantages. The lumped parameter method is easy and quick to implement for flexible bodies with chain-like geometries but is more difficult and ambiguous and hence less useful, for modeling bodies with higher-dimensional geometries. The simplification to independent GBEs also incorrectly represents the curvature bending moment.

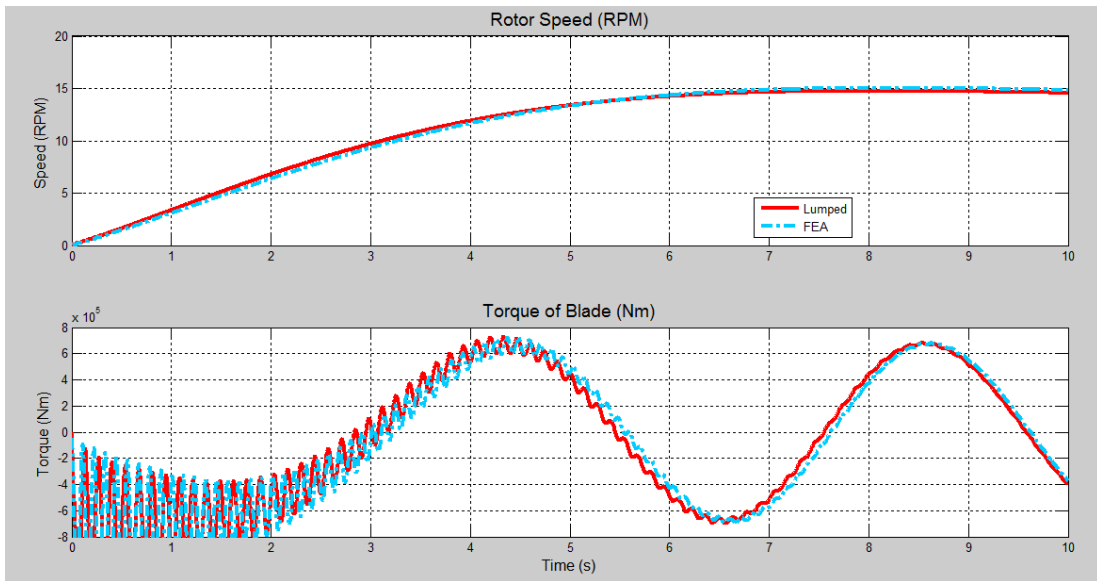


Figure 6- 35 Comparison of the Two Methods

To correctly approximate the curvature moment it must apply the moment on each joint as a function of its own and neighboring joint deflections.

The FEA/state-space method is straightforward to implement for anybody geometry and any number of degrees of freedoms. It is also more naturally suited to control design and analysis problems. Once you embed your FEA state space into a SimMechanics model, this method introduces Transfer Function blocks as which is a solution for the acceleration of the Mass less Bodies. The acceleration depends on the reaction force, which is itself a function of the acceleration and have effect on fidelity.

When compare it with gear train in put torque and the result of the generated blade torque of the two methods.

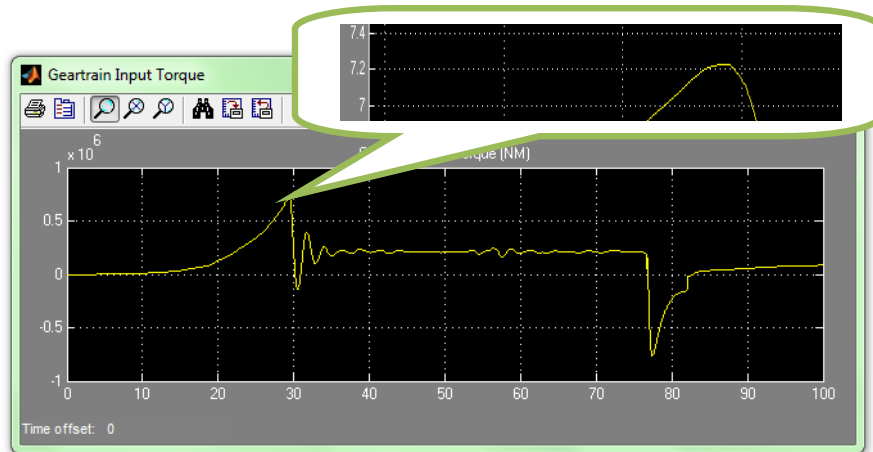


Figure 6- 36: Gear train input torque

Table 3 Comparison of FEA and Lumped Parameter

Method	Lumped parameter method	FEA/state-space method
Blade output torque	667 kNm	722.5 kNm
Required gear train input torque	722.25 kNm	722.25 kNm

As shown in the above table, the FEA/state-space method approximates the gear train starting torque better.

6.5 Nacelle

Nacelle structure is the primary load path from the turbine shaft to the tower. The Nacelle bed plate assembly provides a stiff floor on which to mount the turbine shaft bearings, the power train components and the Yaw drive mechanism.

One fairly common arrangement is a casting in the form of an inverted frustum which supports the low-speed shaft main bearing at the front and the port and starboard gearbox supports towards the rear, with the generator mounted on a fabricated platform projecting to the rear and attached to the main casting by bolts.

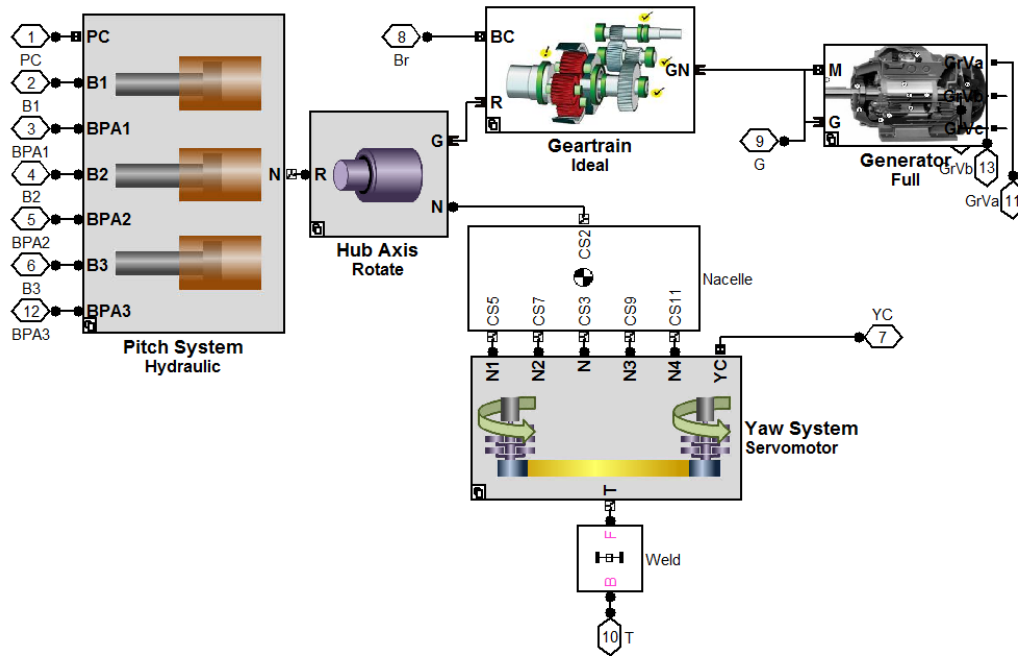


Figure 6- 37: Subsystem of Nacelle

Inside the Nacelle there are the following component Gearbox Variable-Speed Electronics, Generator ,Main Structure ,Electrical Connections ,Control, Safety , Condition Monitoring ,Low Speed Shaft ,Nacelle Cover ,Yaw Drive and Bearing Hydraulic and Cooling Systems ,Bearings Mechanical Brakes, High-Speed Shaft, etc. Service personnel may enter the Nacelle from the tower of the turbine.

To minimize hub and gear case deflections that can cause premature failure of bearings and gears, Nacelle structural supports must be designed to provide adequate stiffness.

6.6 Gear Train

The power from the rotation of the wind turbine rotor is transferred to the generator through the gear train, i.e. through the main shaft, the gearbox and the high speed shaft. The gear train besides connecting the rotor with the generator the purpose of the gear train allows the generator to rotate much faster than the rotor.

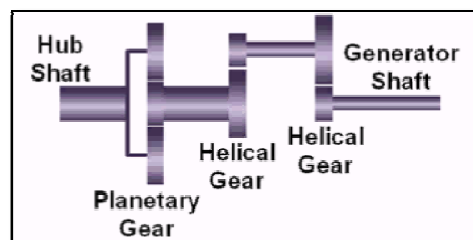


Figure 6- 38: Gear train

The shaft from hub come to the gear train connect the planetary gear the sun gear of the planetary gear rotate much faster than hub. The sun gear attached to the next steps of another gear finally with the generator shaft.

The Simulink representative of the gear train is shown in Figure 6- 39.

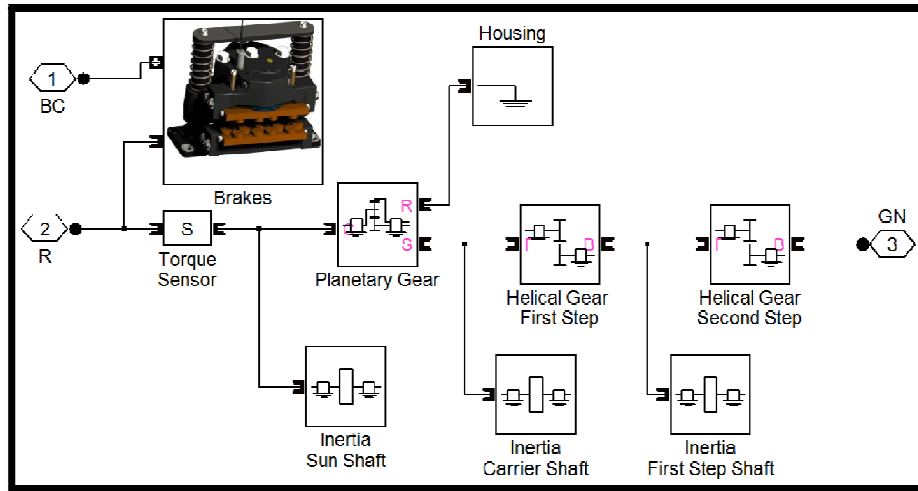


Figure 6- 39: Simulink Gear Train

6.6.1 Components in the Subsystem of the Gear Train

Brake: This wind turbine model brake typically consists of a steel brake disc acted by one brake calipers. The disc can be mounted on rotor shaft (known as the low-speed shaft) or on the shaft between the gearbox and the generator (known as the high-speed shaft).

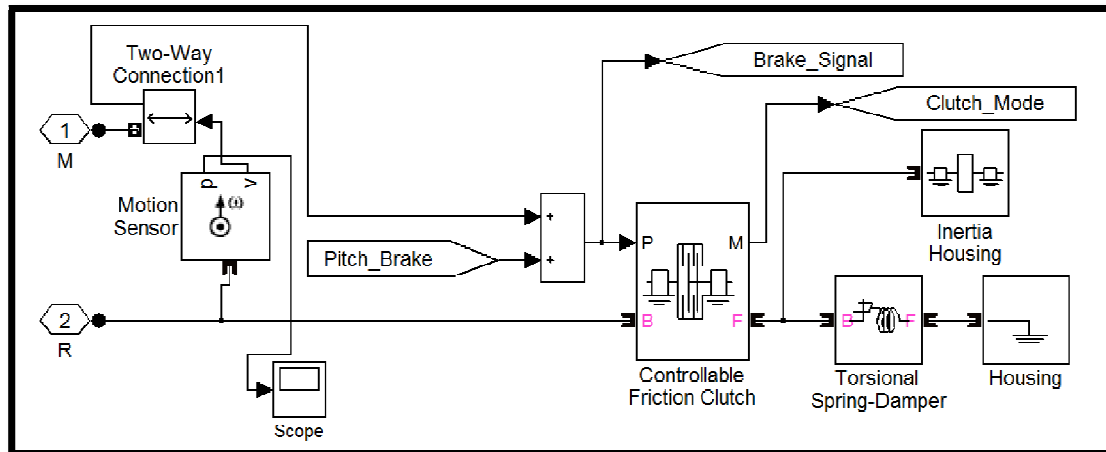


Figure 6- 40: Subsystem of Brake

Friction Clutch: A friction clutch inside the brake transfers torque between two driveline axes by coupling them with friction. The Controllable Friction Clutch block models a standard friction clutch with kinetic friction and static (locking) friction acting on the two axes. The clutch requires a dimensionless input pressure signal P (brake_signal) that modulates the applied kinetic friction. This signal should be positive or zero. Signal P less than zero are interpreted as zero. A signal P of unity corresponds to the full normalized kinetic friction as shown in the Figure 6- 40.

The clutch can be bidirectional, with the follower axis moving relative to the base axis with either sign; or unidirectional, with the follower moving relative to the base with positive sign only. The unidirectional clutch is equivalent to a friction clutch connected in parallel to a one-way clutch that disengages when the relative motion reverses.

For the clutch to lock, the relative follower-base speed must fall below the velocity tolerance and the normalized pressure must be greater than the engagement threshold. A locked clutch remains locked unless the friction constraint torque across the clutch exceeds the static friction limit = (Static friction peak factor)*(Kinetic friction torque).

Torque Sensor: The Torque Sensor block measures the torque transferred along a driveline axis at the point where the Torque Sensor is inserted. A positive torque is transferred from the base (B) axis to the follower (F) axis at that point if the follower axis accelerates positively with respect to the base. The torque is output as a Simulink signal in newton-meter.

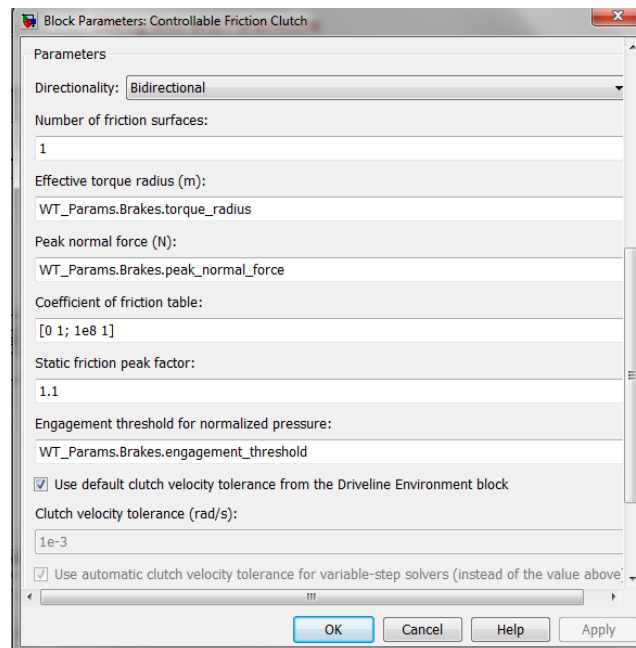


Figure 6- 41: Clutch Block Parameter

Planetary Gear: The Planetary Gear block represents a set of carrier, ring, planet and sun gear wheels. A planetary gear set can be constructed from planet-planet and ring-planet gears. The ring and sun co-rotate with a fixed gear ratio and in opposite directions with respect to the carrier.

The Housing block prevents any driveline component connected to it from rotating about its driveline axis by locking its motion to zero angular velocity.

The nominal rotor speed is 14.3004 rev/min. The gear train ratios should satisfy the generator requirement shown in table.

Table 4: Generator Requirement

Type description	1 speed generator, water cooled
Rated power	2650 kW
Apparent power	2808 kVA
Rated current I_N	2740 A
Max power at Class F PF _{ma}	2815 kW
Max current at Class F IF _{max}	2914 A
No load current I_0	430 A
Number of poles P	6
Synchronous rotation speed n_0	1200 rpm
Rotation speed at rated power	1214 rpm

Therefore in order to get the rated rotational speed of the generator (1214rpm) the gear train gear ratio model as to be 84.3.

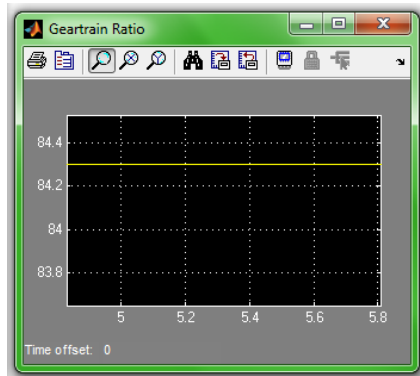


Figure 6- 42: Gear train ratio

When multiply the rotor speed with the gear train ratio rated speed of the generator becomes

Rotor speed=14.55 rev/min

Gear train ratio =84.3

Rated speed of the generator = $14.55 \text{ rev/min} \times 84.3 = 1212.5 \text{ rev/min} \sim \mathbf{1214 \text{ rev/min}}$.

Figure 6- 43 shows the comparison of the above value in generator power versus generator speed graph. The rated power of the generator at the 1214 rpm is 2.5 MW and this power output will feed in to the grid.

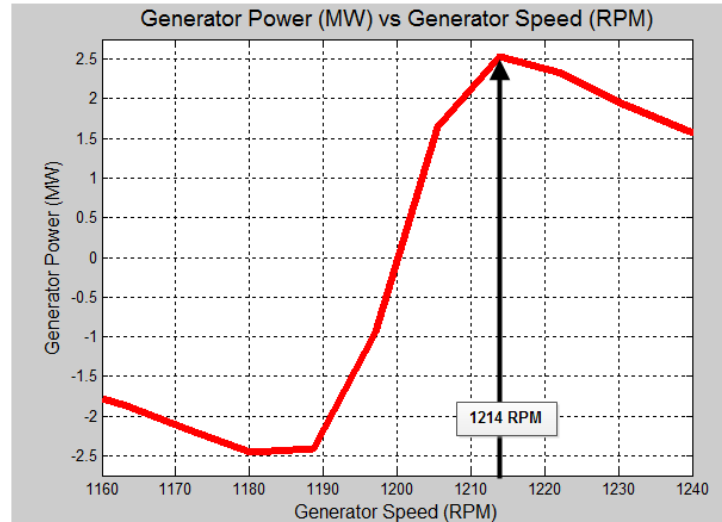


Figure 6- 43: Generator power versus generator speed

6.7 Generator

The wind turbine generator converts mechanical energy to electrical energy. Wind turbine generators are a bit unusual, compared to other generating units ordinarily find attached to the electrical grid. One reason is that the generator has to work with a power source (the wind turbine rotor) which supplies very fluctuating mechanical power (torque).

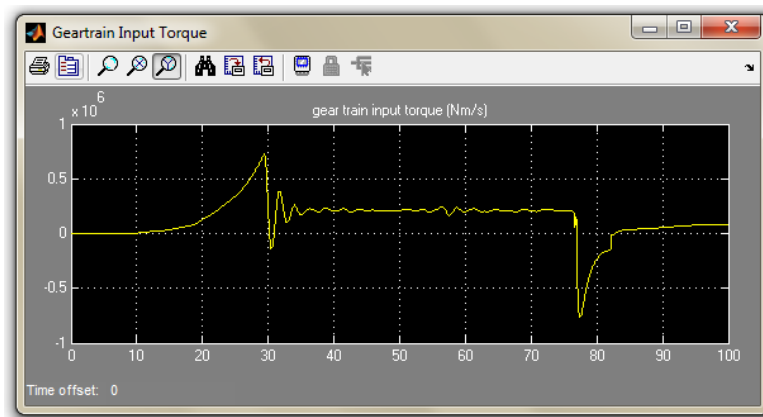


Figure 6- 44: Gear train input torque

The Asynchronous Machine block implements a three-phase asynchronous machine (single squirrel-cage). It operates in either generator or motor mode it depends on the gear train input torque

The mode of operation is dictated by the sign of the mechanical torque:

- If T_m is positive, the machine acts as a generator.
- If T_m is negative, the machine acts as a motor.

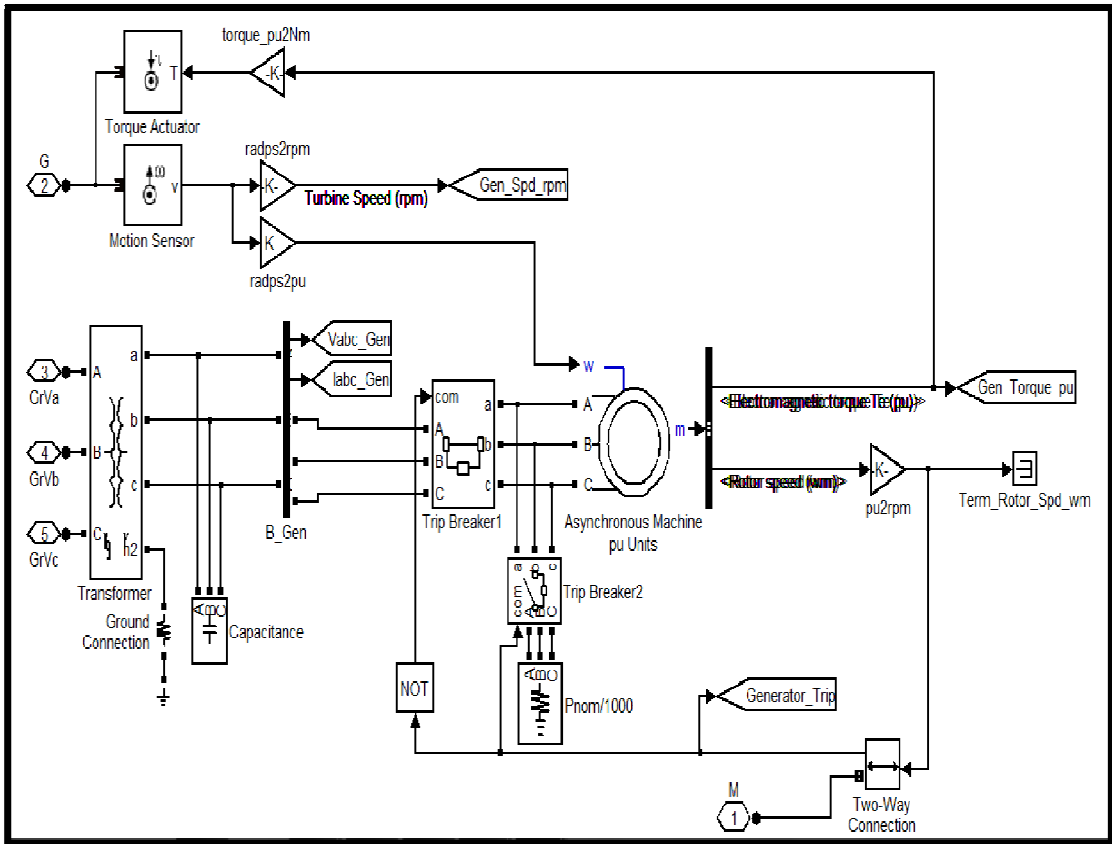


Figure 6- 45: Subsystem of the Generator

The electrical part of the machine is represented by a fourth-order state-space model and the mechanical part by a second-order system. All electrical variables and parameters are referred to the stator. All stator and rotor quantities are in the arbitrary two-axis reference frame. The subscripts used are defined as follows:

The Three-Phase Breaker block implements a three-phase circuit breaker where the opening and closing times can be controlled either from an external Simulink signal (external control mode), or from an internal control timer (internal control mode).

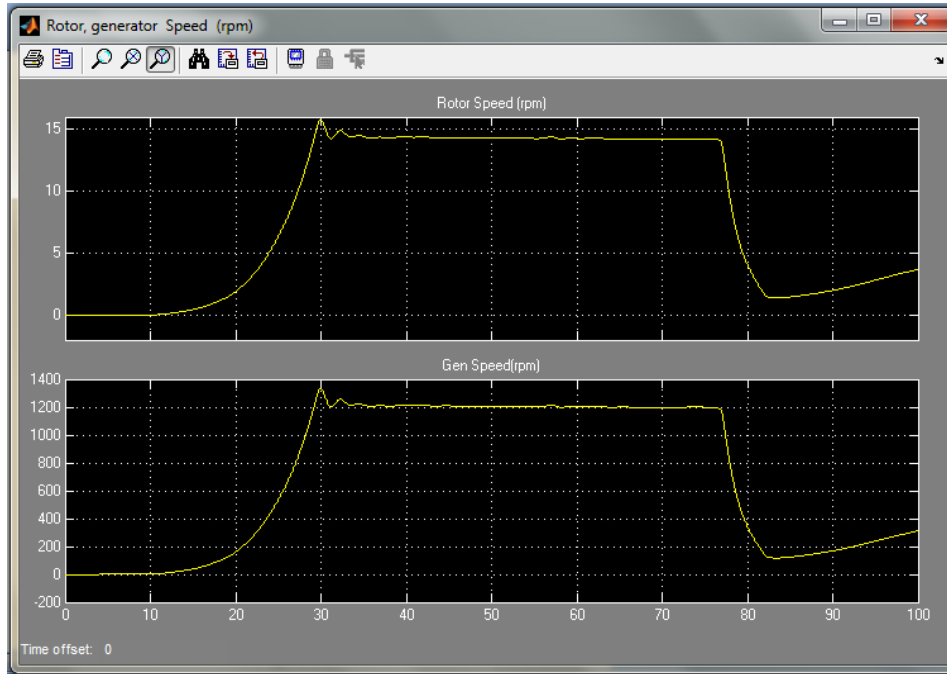


Figure 6- 47: Rotor and generator speed

Its power to the grid is shown in Figure 6- 48, which is 2.5MW.

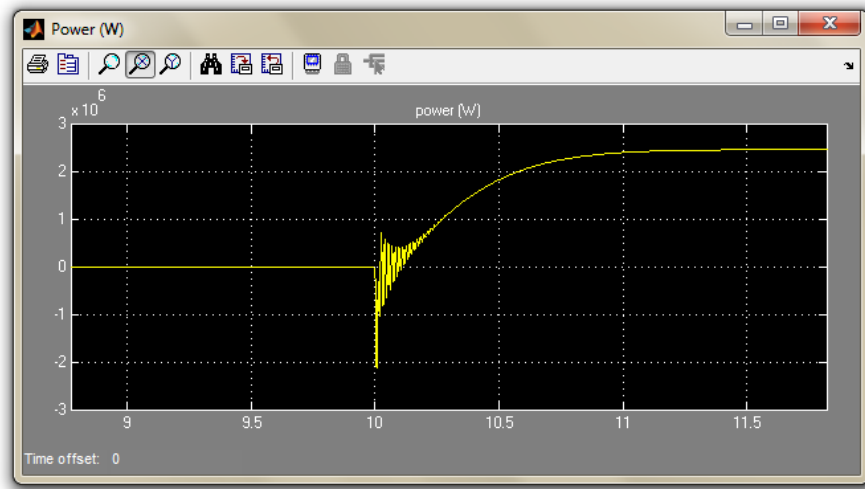


Figure 6- 48: Power to the grid

6.8 Tower

A HAWT tower raises the rotor and Nacelle (N) to the specified hub elevation, the distance from the ground to the center of the swept area. Minimum tower height is determined by the required ground clearance (distance between the lowest point of the swept area and the ground). An increase in tower height above this minimum depends on

the trade-off between the marginal increases in energy capture (because average wind speeds generally increase with increasing elevation) and the marginal increase in system cost, including construction and maintenance costs.

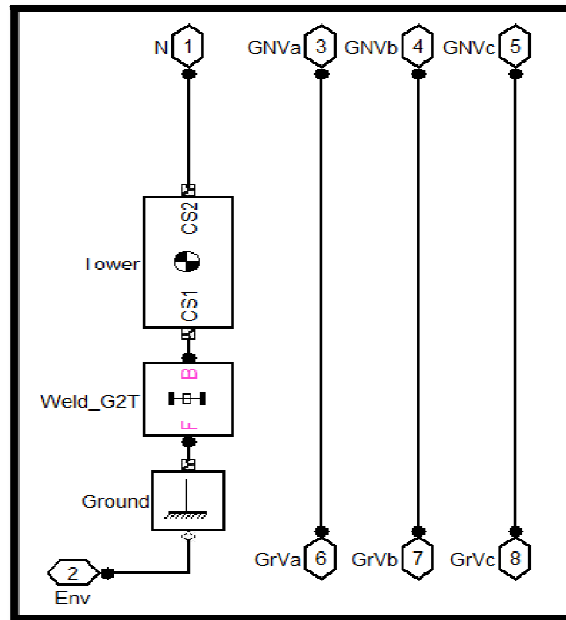


Figure 6- 49: Subsystem of tower

The Tower represents a user-defined rigid body. Body defined by mass m , inertia tensor I and coordinates origins and axes for center of gravity (CG). This dialog sets body initial position and orientation, unless Body and/or connected Joints are actuated separately. This dialog also provides optional settings for customized body geometry and color

Wind turbine grid-tied power system is primarily composed of a wind turbine, a rectifier, electronic (dump) load and a grid-tied inverter, which eventually convert and feed the electrical energy generated by the wind turbine into the utility grid.

The basic requirement of the tower is specified in Appendix A.

6.9 Transmission to Grid

These systems basically involve 3 connected components:

1. Energy source
2. Grid-interactive (GI) inverter - converts direct current (DC) electricity into Conventional 220V alternating current (AC) electricity
3. The Three-Phase V-I Measurement block is used to measure instantaneous three-phase voltages and currents in a circuit.

The power cables from the Nacelle (N)(GNVa, GNVb and GNVc) to (GrVa, GrVb and GrVc) from the wind turbine generator pass through the tower into the grid.

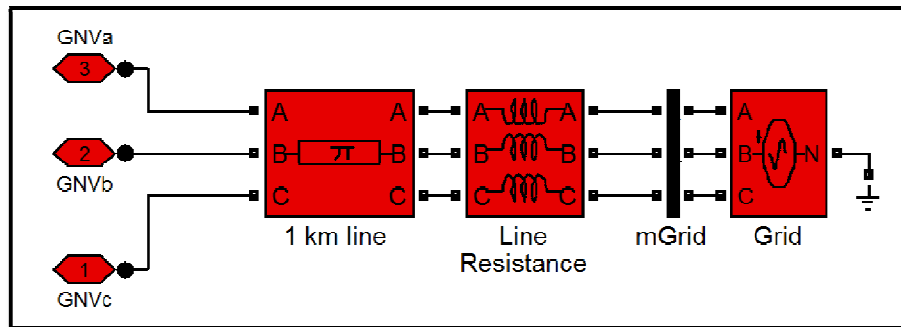


Figure 6- 50: Transmission Line to Grid

The three-phase PI Section Line blocks implements a balanced three-phase transmission line model with parameters lumped in a PI section.

Then three-phase mutual inductance Z1-Z0 block implements a three-phase balanced inductive and resistive impedance with mutual coupling between phases.

The Three-Phase V-I Measurement block is used to measure instantaneous three-phase voltages and currents in a circuit. The block can output the voltages and currents in volts and amperes.

Finally three-phase programmable voltage source (GRID) implement three-phase voltage source with programmable time variation of amplitude, phase, frequency and harmonics depend on the end user need.

6.10 Machine Environment

Defines the mechanical simulation environment for the machine to which the block is connected: gravity, dimensionality, analysis mode, constraint solver type, tolerances, linearization and visualization.

Table 5: Environment Requirements.

Temperature interval for operation	-30 to +30°C
Temperature interval for structure	-40 to +50°C

The following block parameter show mechanical simulation environment of the wind turbine

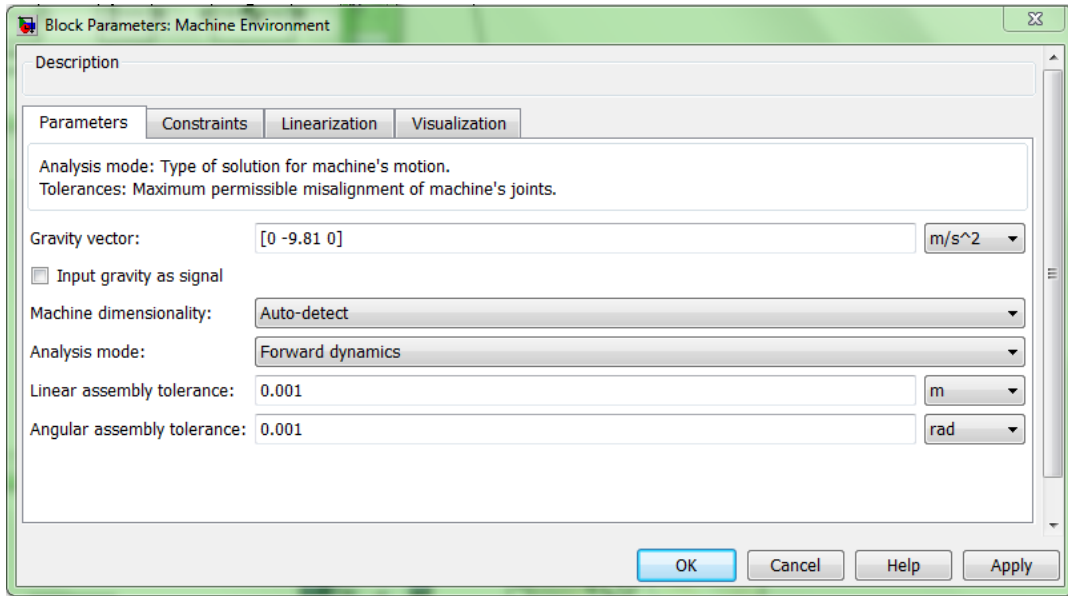


Figure 6- 51 Environment Block Parameter

Chapter Seven

7.1 Control System Design of the Wind Turbine

A primary function of the control system is to maintain the machine operating Parameters within their normal limits. The wind turbine control system consists of a number of sensors, a number of actuators and a system consisting of hardware and software which processes the input signals from the sensors and generates output signals for the actuators. The sensors might include:

- An anemometer,
- A wind vane,
- At least one rotor speed sensor,
- An electrical power sensor,
- A pitch position sensor,
- Various limit switches,
- Vibration sensors,
- Temperature and oil level indicators,
- Hydraulic pressure sensors,
- Operator switches push buttons, etc.

The actuators include a hydraulic or electric pitch actuator, sometimes a generator torque controller, generator contactors, switches for activating shaft brakes, Yaw motors, etc.

The system that processes the inputs to generate outputs usually consists of a computer or microprocessor-based controller which carries out the normal control functions needed to operate the turbine. The safety system must be capable of overriding the normal controller in order to bring the turbine to a safe state if a serious problem occurs.

Wind turbines are optimized to produce maximum power output at the nominal wind speed. Beside, its necessary to limit the power output at high wind speed condition in order to protect the structural integrity of the wind turbine.

7.1.1 Closed-loop Control: Issues and Objectives

The closed-loop controller is usually a software-based system that automatically adjusts the operational state of the turbine in order to keep it on some pre-defined operating curve or characteristic. Some examples of such control loops are:

- Control of blade pitch in order to regulate the power output of the turbine to the rated level in above-rated wind speeds;
- Control of blade pitch in order to follow a predetermined speed ramp during start-up or shut-down of the turbine;
- Control of generator torque in order to regulate the rotational speed of a variable speed turbine;
- Control of Yaw motors in order to minimize the Yaw tracking error.

Some of these control loops may require very fast responses in order to prevent the turbine wandering far from its correct operating curve. Such controller's may need to be designed very carefully if good performance is to be achieved without detrimental effects on other aspects of the turbine's operation. Others, such as Yaw control, are typically rather slow acting and careful design is then much less critical.

This chapter examines the main issues behind closed-loop controller design and presents some of the techniques that can be used to a successful design.

7.2 Main Controller

The magnitude and direction of wind from wind data sheet, MATLAB file or directly from the data logger comes as an input to the main controller. The Supervisory control within the main controller use this input in order to control the operating condition (turbine state) of the wind turbine.

Supervisory control is the means whereby the turbine is brought from one operational state to another. The operational states include:

- stand-by, when the turbine is available to run if external conditions permit,
- start-up,
- power production,
- shutdown, and
- Stopped with fault.

It is possible to envisage other states, or it may be useful to further subdivide some of these states. As well as deciding when to initiate a switch from one state to another, the supervisory controller will carry out the sequence control required. The control sequence for start-up of a variable-speed pitch-regulated wind turbine consists of the following steps:

- Power-up the pitch actuator;
- Release the shaft brake;

- Ramp the pitch position demand at a fixed rate to some starting pitch;
- Wait until the rotor speed exceeds a certain small value;
- Engage the closed loop pitch control of speed;
- Ramp the speed demand up to synchronous speed;
- Wait until the speed has been close to the target speed for a specified time;
- Close the generator contactors;
- Engage the closed loop pitch control of power; and
- Ramp the power demand up to the rated level.

The supervisory controller must check that each stage is successfully completed before moving on to the next. If any stage is not completed within a certain time, or if any faults are detected, the supervisory controller should change to shut-down the system.

A turbine state machine is a representation of an event-driven (reactive) system. In an event-driven system, the system responds by making a transition from one state (mode) to another. This action occurs in response to an event, as long as the condition defining the change is true

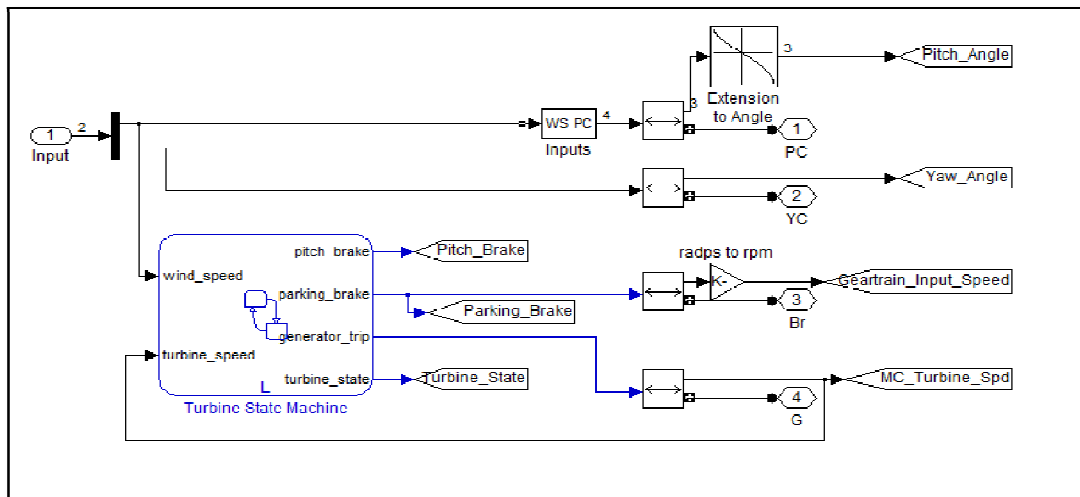


Figure 7- 1: Subsystem in the main controller (Supervisory control)

A state flow chart is a graphical representation of a finite state machine, where states and transitions form the basic building blocks of the system as shown Figure 7- 2. The transmission has these operating states: park, start up, generating and turbine state. As the wind speed increase or decrease, the system makes a transition from one state to another, for example, it translate from park to start up when the wind speed increase from cut in.

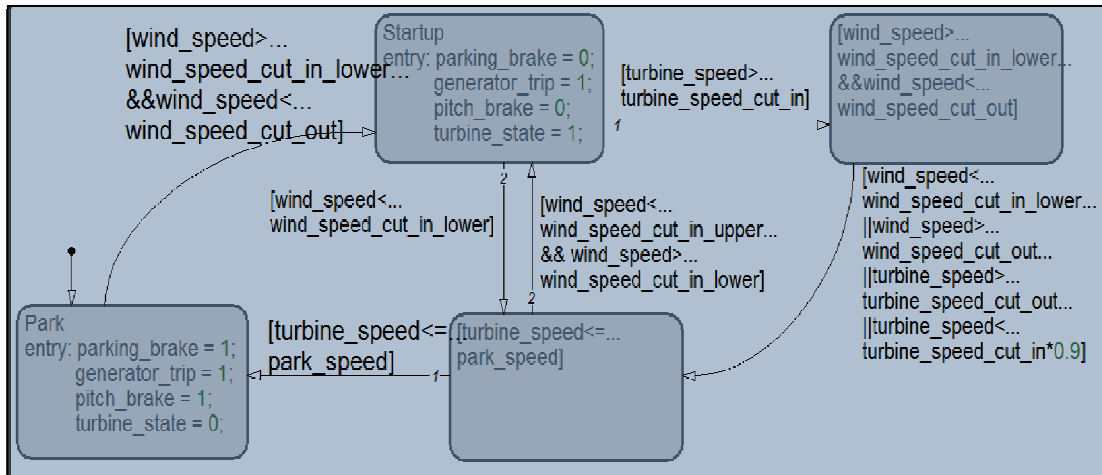


Figure 7- 2: Subsystem in turbine state machine

Supervisory control used to analyze the operating condition of the turbine and determine which system should be turned on or turned off.

The turbine state machine divided in to four state flow supervisory control state

1. Parking state

The wind turbine doesn't rotate so the park brake should be turned on which means:

- The wind turbine is not in the direction of the wind. That show the turbine is down.
- The generator not generating power or connects to the grid

Then when wind speed gets equal and greater than the cut in speed the supervisory control state change to start up state.

2. Start up state

In this state the wind speed is between the cut in and cut out.

- The wind turbine starts to spin and attempting to produce power;
- Still the generator not connected to the grid.

When the turbine speed get higher above the certain required speed (nominal speed) it will change to the generating state.

3. Generating state

Now the wind speed at nominal speed. Therefore

- It is enough to generating power;
- The generator connected to the grid.

4. Brake state

When the wind speed is above the cut out or below the cut in, the supervisory control sends a signal into the break system to

- Engage the frictional clutch in the gear train as shown in the Figure 7- 3.

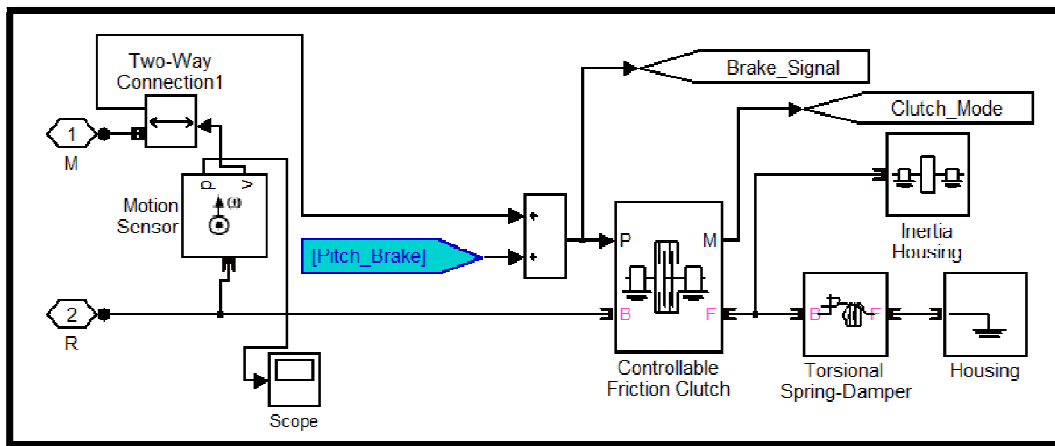


Figure 7- 3: Controllable frictional clutch in brake sub system

Therefore, the proper gear train input speed from the various condition of the wind speed will determine.

The pitch brake, parking brake turbine and generating states send signals to blocks that have the specified tag to display the operating condition in to main controller scope.

Running the supervisory control used to display the turbine state to the user. the scope in Figure 7- 4.shows the transition of wind turbine moves from state to state

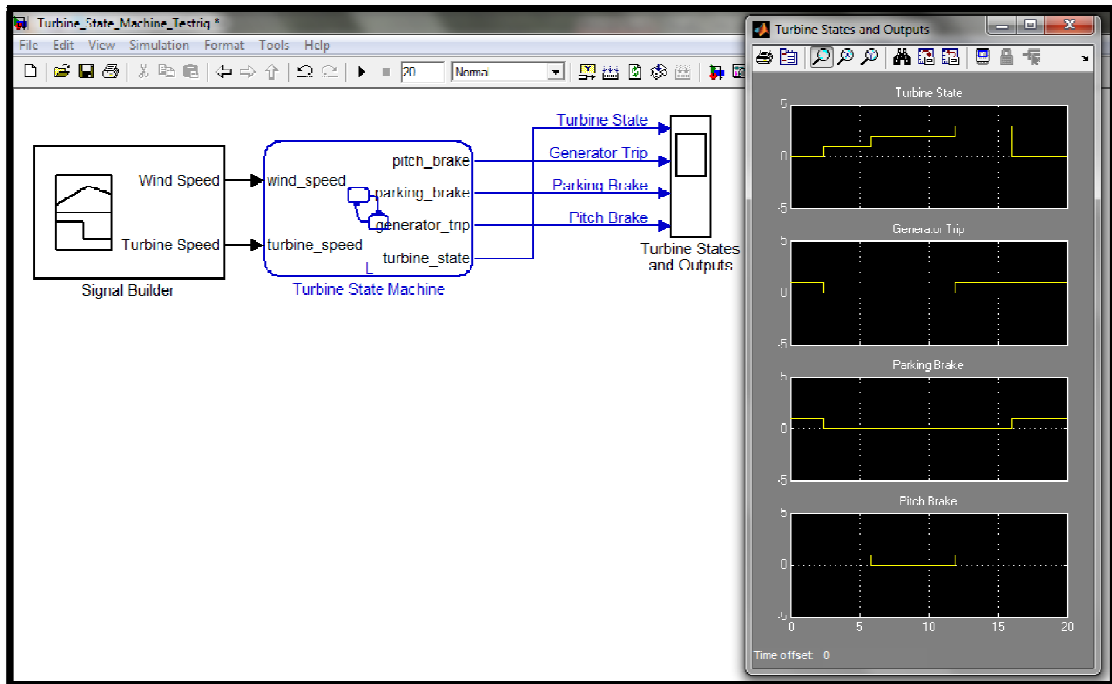


Figure 7- 4 Display of turbine state machine

7.3 Pitch Control

Pitch control is the most common means of controlling the aerodynamic power generated by the turbine rotor. Pitch control also has a major effect on all the aerodynamic loads generated by the rotor.

Below rated wind speed, the turbine should simply be trying to produce as much power as possible, so there is generally no need to vary the pitch angle. The aerodynamic loads below rated wind speed are generally lower than above rated, so again there is no need to modulate this using pitch control.

For above rated wind speed, pitch control provides a very effective means of regulating the aerodynamic power and loads produced by the rotor so that design limits are not exceeded. In order to achieve good regulation, however, the pitch control needs to respond very rapidly to changing conditions. This highly active control action needs very careful design as it interacts strongly with the turbine dynamics.

Below rated wind speed, the pitch setting should be at its optimum value to give maximum power. It follows that when the wind speed rises above rated, either an increase or a decrease in pitch angle will result in a reduction in torque. An increase in pitch angle, defined as turning the leading edge into wind, reduces the torque by decreasing the angle of attack and hence the lift. This is known as pitching towards feather. A decrease in

pitch, i.e., turning the leading edge downwind, reduces the torque by increasing the angle of attack towards stall, where the lift starts to decrease and the drag increases. This is known as pitching towards stall.

7.4 Pitch controller

The primary objective of the pitch controller may be to limit power or rotor speed in high winds and used to optimize energy capture in low winds. However, since the controller can also have a major effect on structural loads and vibrations, it is vital to consider these when designing the control algorithm. Thus a fuller description of the pitch controller objectives might be:

- To regulate aerodynamic torque in above-rated wind speeds;
- To minimize peaks in gearbox torque;
- To avoid excessive pitch activity;
- To minimize tower base loads as far as possible by controlling tower vibration, and
- To avoid exacerbating hub and blade root loads.

The main objective controlling rotor speed using pitch angle is:

- To rotate the rotor at constant speed and
- To control the lift produced by the blade by controlling the pitch angle

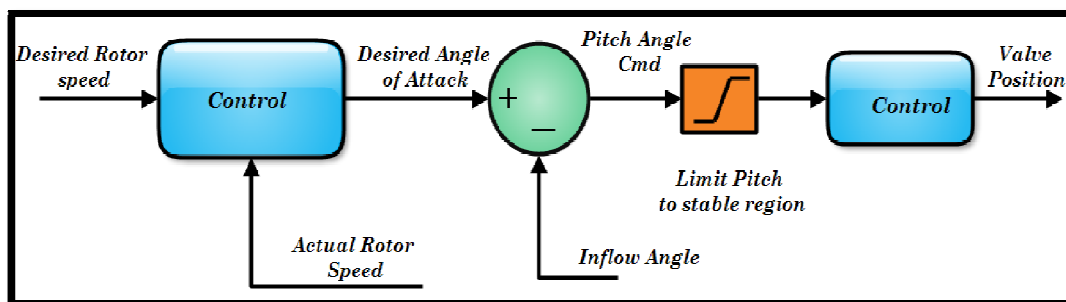


Figure 7- 5: Control stricture for pitch control

There is a desired rotor speed and it needs to determine the pitch angle command. The control system will compare the actual rotor speed with desired rotor speed to determine the desired angle of attack. Then it compares the inflow angle and desired angle of attack to determine the pitch angle command, which positioned the 4 way directional valves in order to actuate the actuator cylinder in the required speed and time as shown in the Figure 7-5.

The Simulink representative of the above control structure for pitch control are shown in Figure 7- 6, 7-7 and 7-8.

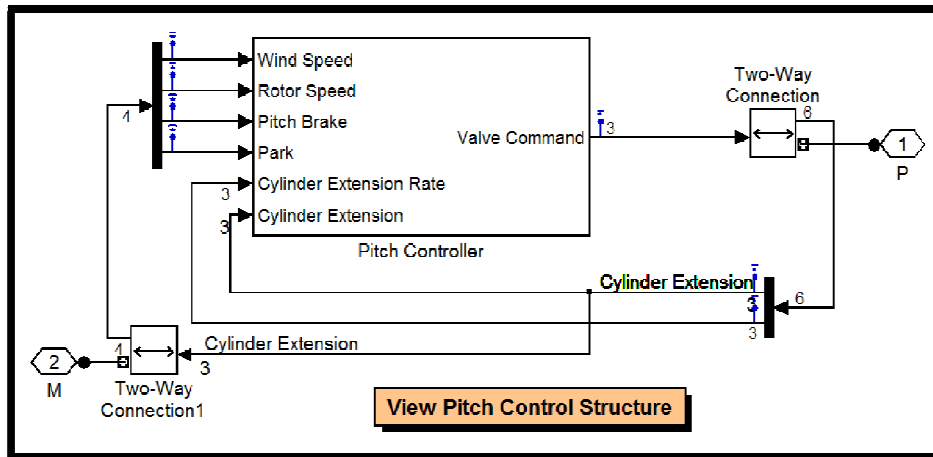


Figure 7- 6: Over all model of the pitch control of the wind turbine model

From wind input to the to the system the supervisory control system check for pitch brake and park condition in order to get the desired angle of attack using such switch system as shown in the figure 7-7.

The Switch block passes through the first input or the third input based on the value of the second input. The first and third inputs are called data inputs. The second input is called the control input. Specify the condition under which the block passes the first input by using the Criteria for passing first input.

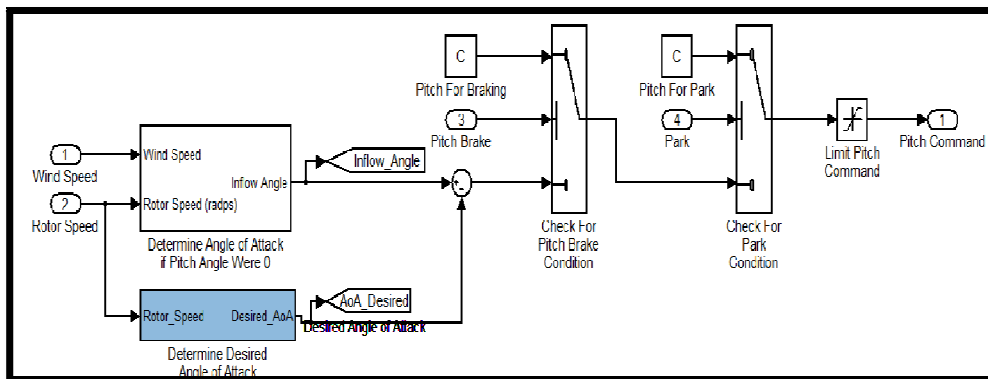


Figure 7- 7: Simulink representation for determining the pitch command

The pitch command that generated from the pitch controller then use for positioned the 4 way directional valves in order to actuate the actuator cylinder in the required speed and time

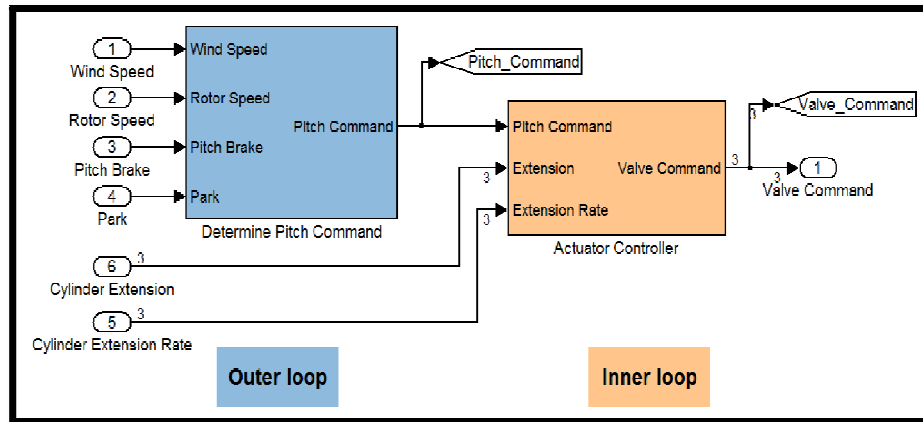


Figure 7- 8: Subsystem of the pitch controller

7.3.1 PI and PID controllers

In the subsystem of the actuator controller there is PI controller. A brief general description is given here of PI and PID controllers, since they will be referred to a number of times in the subsequent sections. The proportional and integral (PI) controller is an algorithm which is very widely used for controlling all kinds of equipment and processes. The control action is calculated as the sum of two terms, one proportional to the control error, which is the difference between the desired and actual values of the quantity to be controlled and one proportional to the integral of the control error. The integral term ensures that in the steady state the control error tends to zero. If it did not, the integral term would make the control action continue to increase. The proportional term makes the algorithm more responsive to rapid changes in the quantity being controlled.

A differential term is often added, which gives a contribution to the control action proportional to the rate of change of the control error. This is then known as a PID controller. In terms of the Laplace operator S , which can usefully be thought of as a differentiation operator, the PID controller from measured signal x to control signal y can be written as follows:

$$y = \left(k_p + \frac{k_i}{s} + \frac{k_d s}{1+sT_d} \right) x \quad (7.1)$$

Where K_p , K_i and K_d are the proportional, integral and derivative gain respectively. The denominator of the differential term is essentially a low-pass filter and is needed to ensure

that the gain of the algorithm does not increase indefinitely with frequency, which would make the algorithm very sensitive to signal noise. Setting $K_d = 0$ results in a PI controller.

It is often the case that the control action is subject to limits. Clearly the choice of controller gains is crucial to the performance of the controller. With too little overall gain, the turbine will wander around the set point, while too much gain can make the system completely unstable.

Inappropriate combinations of gains can cause structural responses to become excited. This section outlines some of the techniques which have been found to be useful in designing closed loop control algorithms for wind turbines, such as the gains of a PI controller and will compare the ideal pitch actuator system with optimized system.

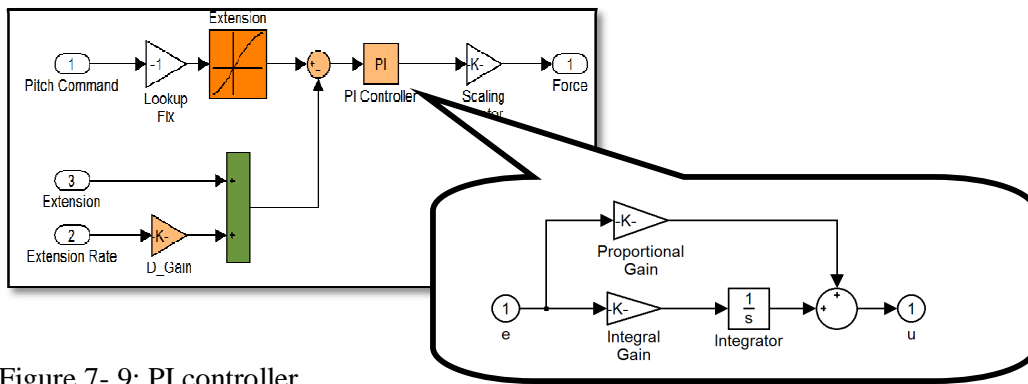


Figure 7- 9: PI controller

7.3.2 Pitch Actuation System Simulink Model

An important part of the control system of a pitch-controlled turbine is the pitch actuation system.

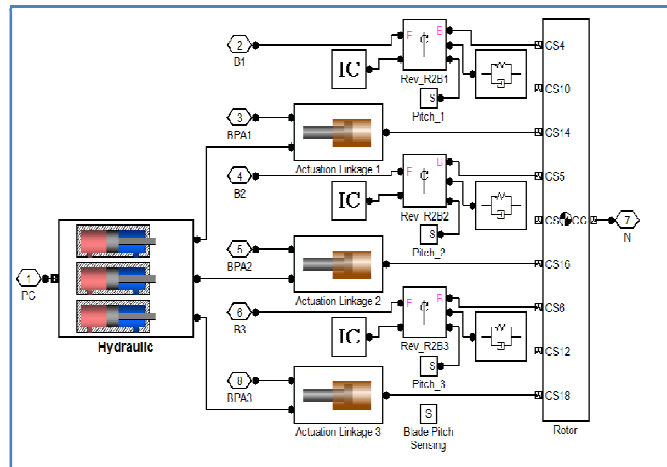


Figure 7- 10: Subsystem of pitch actuation system

When the pitch actuation system get command from the pitch controller (PC), the hydraulic cylinder inside the system will extend or contract to actuate the blade. Therefore the blade pitched in to or away from the wind direction to catch wind.

The hydraulic system connected into the rotor using such actuation linkage and revolute. The revolute block represents single rotational degrees of freedom about a specified axis between two bodies the rotor and actuator.

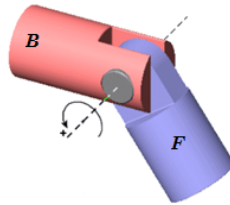


Figure 7- 11: Revolute Motion of Follower (blue) Relative to Base (red)

The follower (F) Body rotates relative to the base (B) Body about a single rotational axis going through collocated Body coordinate system origins. Sensor and actuator ports can be added.

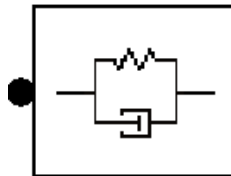


Figure 7- 12: Spring & Damper block

The Joint Spring & Damper block models a damped linear oscillator force acting along a prismatic primitive or a damped linear oscillator torque acting about a revolute primitive. The joint primitives are connected between those two bodies and the force or torque acts between these bodies. The sign of the force or torque is set by the base (B)-to-follower (F) sequence of the bodies. These models represent damped linear translational and tensional springs in the prismatic and revolute cases, respectively.

In Figure 7- 10 there are also blade pitch sensors which measures linear/angular position, velocity, acceleration, computed force/torque and/or reaction force/torque of a Joint primitive. Outputs are Simulink signals.

Within the hydraulic system there are three actuators which actuate each blade. When they get pitch command PF_Pitch from the pitch controller system as shown in the Figure 7- 13. A hydraulic actuator would usually be controlled by means of a proportional valve

controlling the flow of oil to the actuator cylinder. The valve opening and hence the oil flow rate, would be set in proportion to the required pitch rate. The demanded pitch rate come directly from the turbine controller, or it come from a pitch-position feedback loop. In this case the turbine controller generates a pitch-position demand. This is compared to the measured pitch position and the pitch-position error is turned into a pitch-rate demand through a fast PI or PID control loop, implemented either digitally or by means of a simple analogue circuit.

Inside each actuators the 4-Way Directional Valve D block simulates a configuration of hydraulic continuous 4-way directional valve (Figure 7- 15). The block has four hydraulic connections, corresponding to inlet port (P), actuator ports (A and B) and return port (T) and one physical signal port connection (S), which controls the spool position.

The Fixed Orifice block models a sharp-edged constant-area orifice, flow rate through which is proportional to the pressure differential across the orifice. The model accounts for the laminar and turbulent flow regimes by monitoring the Reynolds number (Re) and comparing its value with the critical Reynolds number (Re_{cr}).

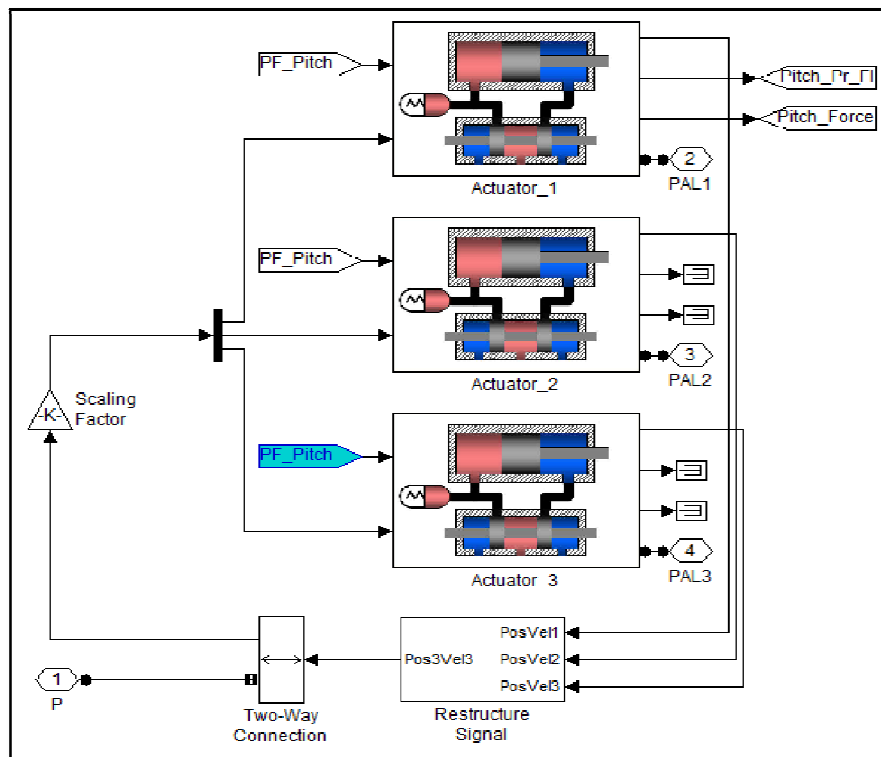


Figure 7- 13: Hydraulic subsystem

Spring-Loaded Accumulator in this block modeled as when fluid entering the accumulator compresses the spring, thus storing hydraulic energy. Since the spring compression increases as fluid enters the chamber and decreases as the accumulator is discharged, the pressure is not constant. The spring is preloaded. Therefore, fluid starts entering the chamber only after the inlet pressure crosses over this threshold.

There is Double-Acting Hydraulic Cylinder system in order to actuate the blade. This Cylinder block models a device that converts hydraulic energy into mechanical energy in the form of translational motion. Hydraulic fluid pumped under pressure into one of the two cylinder chambers forces the piston to move and exert force on the cylinder rod.

The model of the cylinder is built of the following building blocks: Translational Hydro-Mechanical Converter, Piston Chamber, Translational Hard Stop and Ideal Translational Motion Sensor. The rod motion is limited with the mechanical Translational Hard Stop block.

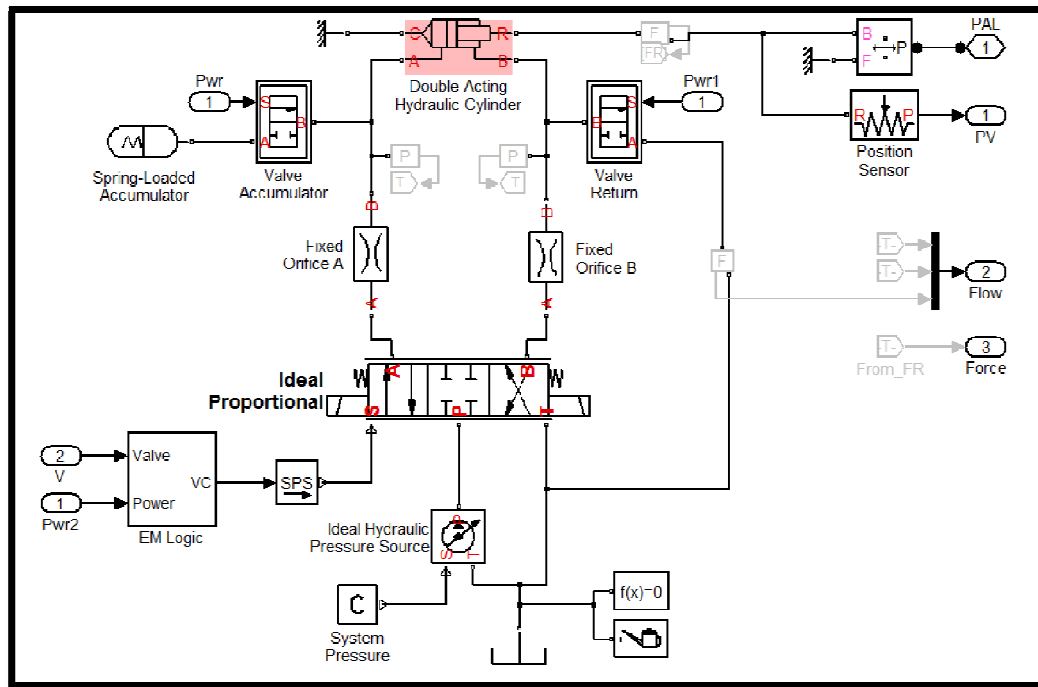


Figure 7- 14: Subsystem of hydraulic actuator system

Connections R and C in fig 7-14 are mechanical translational conserving ports corresponding to the cylinder rod and cylinder clamping structure, respectively. Connections A and B are hydraulic conserving ports. Port A is connected to chamber A

and port B is connected to chamber B. The block directionality is adjustable and can be controlled with the Cylinder Orientation parameter.

The energy through hydraulic port A or B is directed to the appropriate Translational Hydro-Mechanical Converter block and Hydraulic Piston Chamber block. The converter transforms hydraulic energy into mechanical energy, while the chamber accounts for the fluid compressibility in the cylinder chamber. The rod motion is limited with the mechanical Translational Hard Stop block in such a way that the rod can travel only between cylinder caps. The Ideal Translational Motion Sensor block in the schematic is introduced to determine an instantaneous piston position, which is necessary for the Hydraulic Piston Chamber blocks.

7.3.3 Modeling the pitch control system based on the ideal system

It's very common when engineers designing a system, they don't have the entire requirement they need to produce the final design. The requirement they have will change over the course development process as other portion of the design change.

It use Simulink model based design in order to determine a detail requirement for pitch actuation system shown in the Figure 7- 15. The wind turbine rotates about an axis attached the blade that has a mechanical linkage to extend or contract.

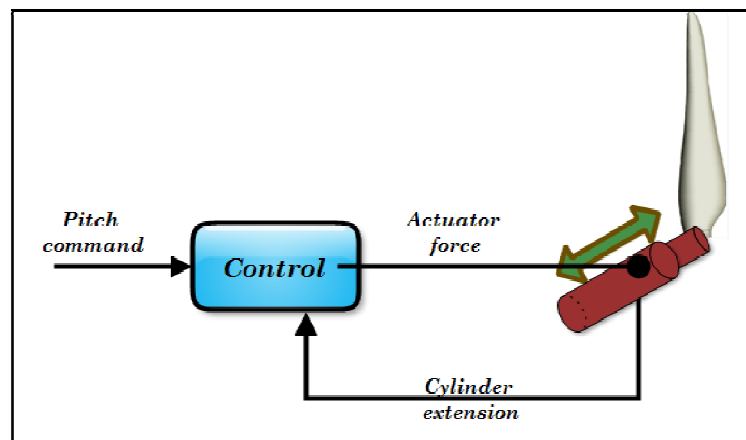


Figure 7- 15: Pitch actuation system

In this system, the input to those systems is a pitch command. Figure 7- 16 shows the measured pitch angle used to identify how the system responds to the system.

In order to determine the system performance requirement for the pitch actuator (force and speed), first the system can be modeled as ideal pitch actuator and controller under Simulink environment as requirement. Then optimizing the system to approach the ideal

system makes the pitch control system to give better response with the required time and speed. In ideal model, the ideal joint sensor and joint actuator modeled as to produce much force as fast as needed.

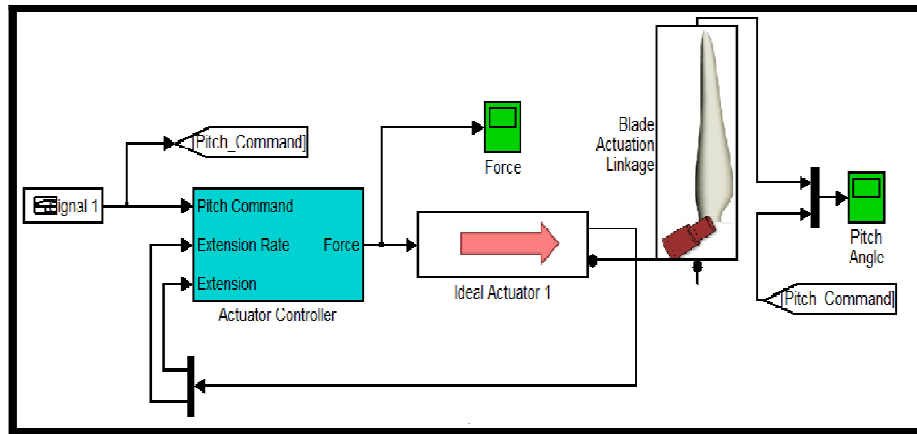


Figure 7- 16: Ideal pitch control system

The simulation result of ideal pitch control system is shown in Figure 7- 17.

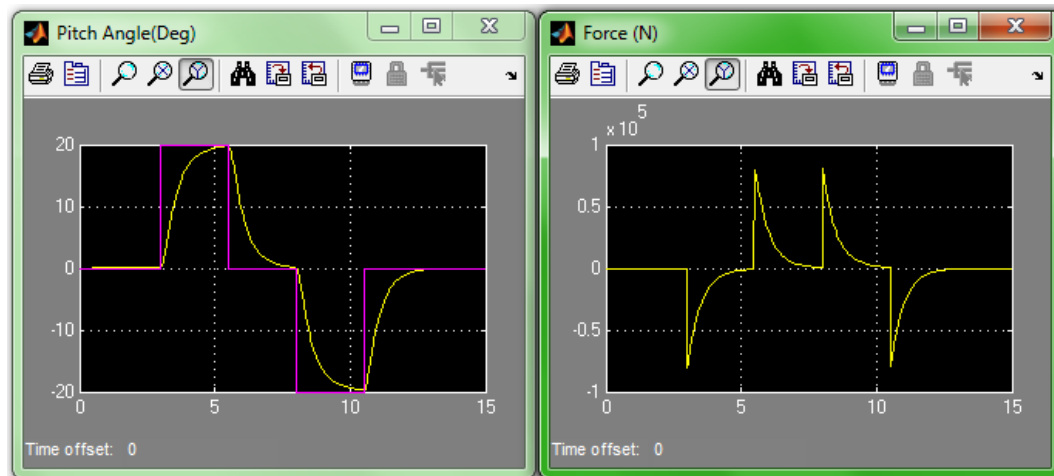


Figure 7- 17: Ideal pitch angle and force

The above ideals pitch angle and force result uses for comparing and contrasting how the output of the analytical design and optimization design close to the ideal.

7.3.4 Initial design of the actual pitch actuator system

Before optimization running the initial designed model (fig 7-18) results a 3D animation of blade and a plot that show the pitch angle and pitch actuation force (fig 7-19).

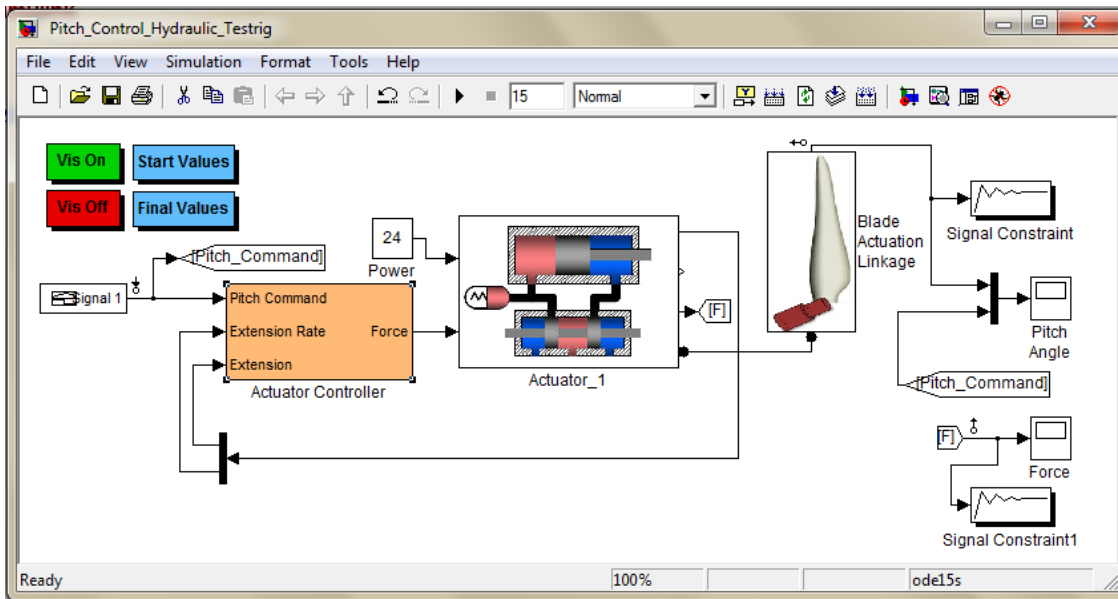


Figure 7- 18 Initial design pitch angle and force

Running the above model which gives the following pitch angle and force

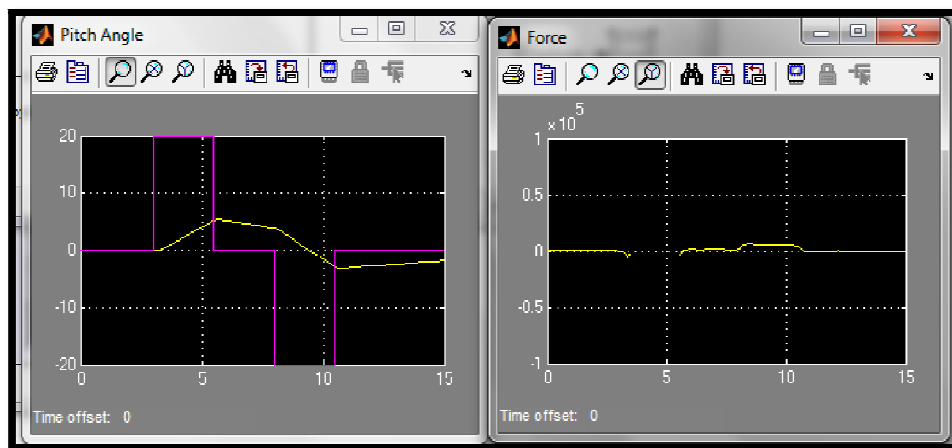


Figure 7- 19: Pitch angle and pitch actuation force

After running the simulation

- The pitch system moves the blade very slowly.
- The pitch system (yellow line) is not the same as the command angle.(pink line)

As a result, the pitch actuation force is too low actuates the system or it react very slowly relative to the commanded angle.

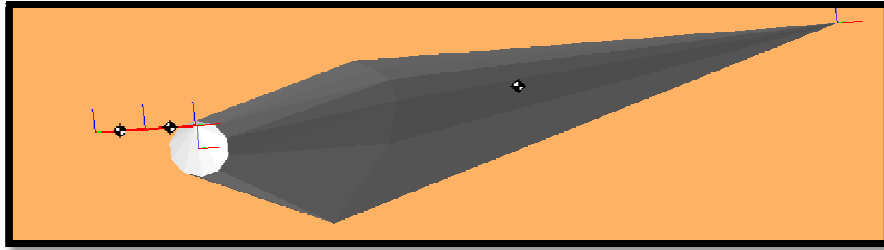


Figure 7- 20: 3D animation of blade

7.3.5 Compensator Design

To get better response of pitch control system compensator design analysis is done in the PI controller. Linear control theory used to determine the dynamics of the closed-loop system from those of the open-loop system. Linearize system using Simulink control design perform linear control design with control system tool box and test and retest controller using non linear system can be made.

Start with non linearized model of the plant which linearizes the model using linear control theory. Then use root locus plot, body diagrammed and other tools to specify the compensator.

The following points show how the compensator designed. To compensate lower response of the system it needs to follow those procedures.

- i In the model shown in Figure 7- 18 open the compensator design



Then control and estimation tool manager tool box will be displayed as shown in Figure 7-21.

- ii select the Blocks to Tune

To select the blocks in model to tune:

- In the left pane, select the location in the model hierarchy that corresponds to the block wanted to tune. In this case in the actuator controller select PI controller.
- In the center pane, select the block to tune. Select proportional gain and integral gain block.

Click **OK**.

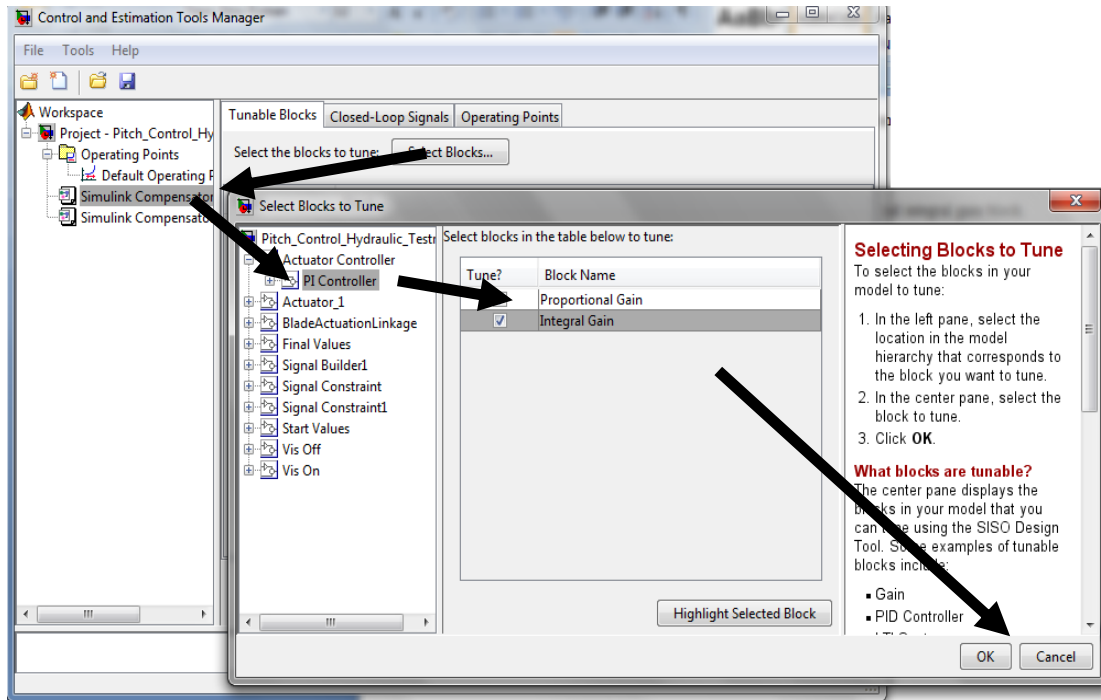


Figure 7- 21: Control and estimation tool manager

- iii Specify closed-loop signals for the design: right click on signals in Figure 7- 19 Simulink model using the Linearization Points menu to select the input and output of the control system.
- iv In the operating point dialog box select simulation snapshot and give some time value in second between the model operating time range.
- v In design configuration wizard select the type of plot which is root locus and open loop bode. In this wizard the SISO Design Tool is a graphical-user interface (GUI) to design compensators. The Graphical Tuning Window, a graphical user interfaces (GUI) for displaying and manipulating bode and root locus for the controller currently being designed. The two are dynamically linked change the gain in the root locus; it immediately affects the Bode diagrams as well. The SISO Design Task-associated LTI Viewer.

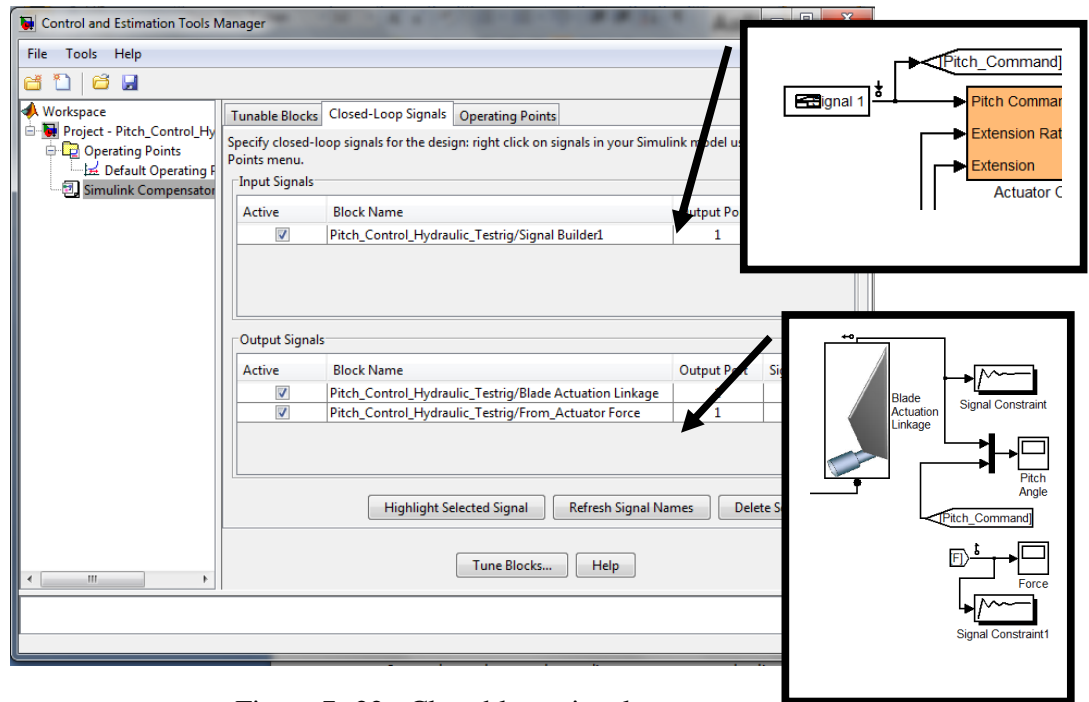


Figure 7- 22 : Closed loop signals

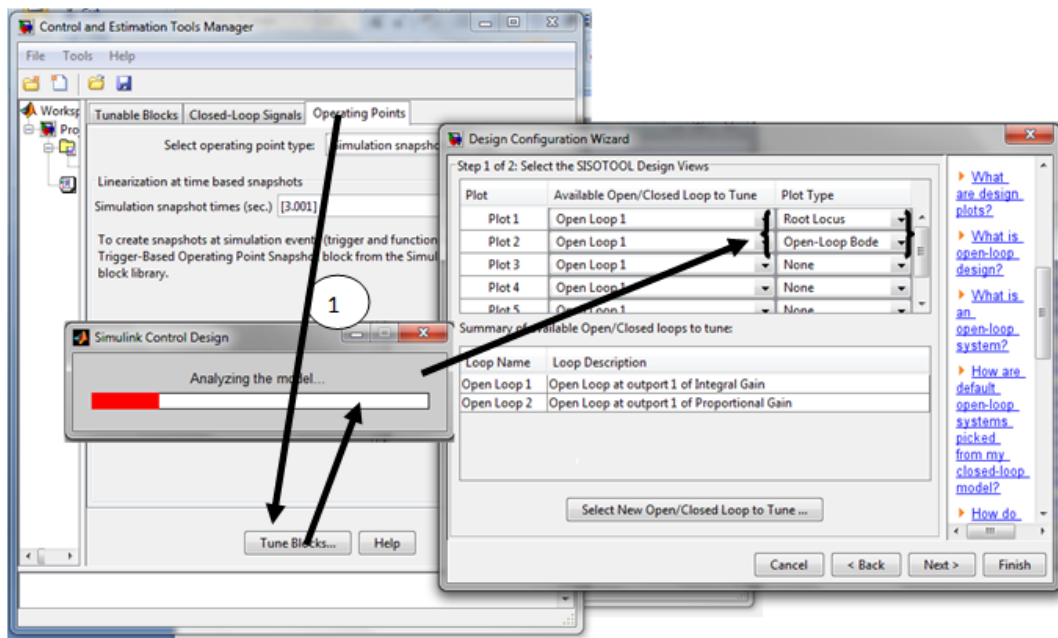


Figure 7- 23: Operating point

vi Analysis plots are plots that show the responses or dynamics of an open- or closed-loop system or tunable block in the model.

In analysis plot select the two step response Figure 7- 24.

- 1 To show the step responses from the input to the angle of the blade reaches.
- 2 Force the actuator is producing.

Click **finish**.

The plot is generated automatically the root locus diagram that show the linearize system Figure 7- 25 and step response diagram Figure 7- 26.

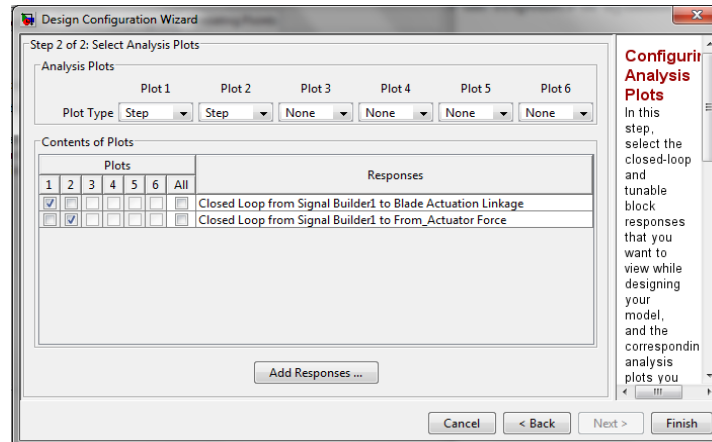


Figure 7- 24: Design configuration wizard

In the root locus plot as the gain increases, the closed-loop poles (p) move from the open-loop poles (x), corresponding to zero feedback gain, to the open-loop zeros (O). (Actually

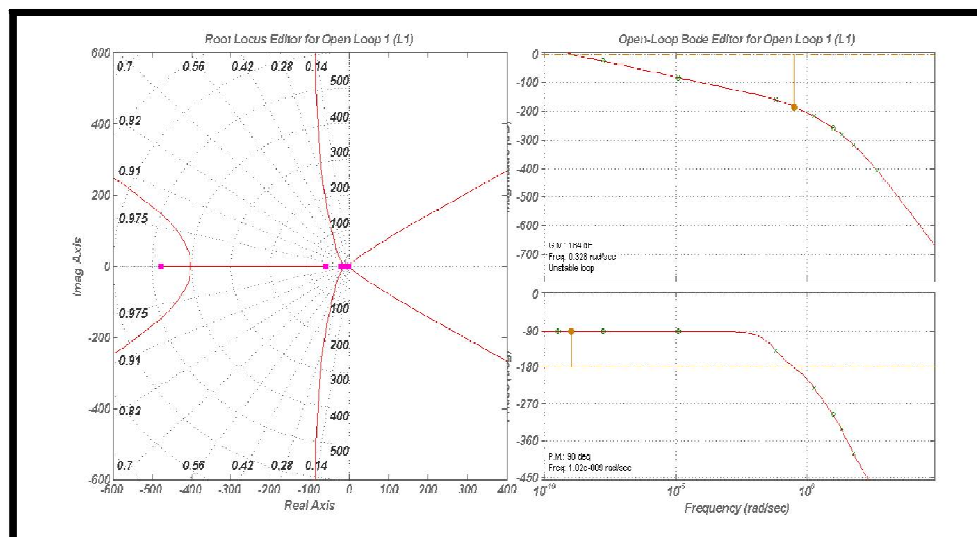


Figure 7- 25: Root Locus

there are usually more poles than zeros; the ‘missing’ zeros can be considered to be equally spaced around a circle of infinite radius.)

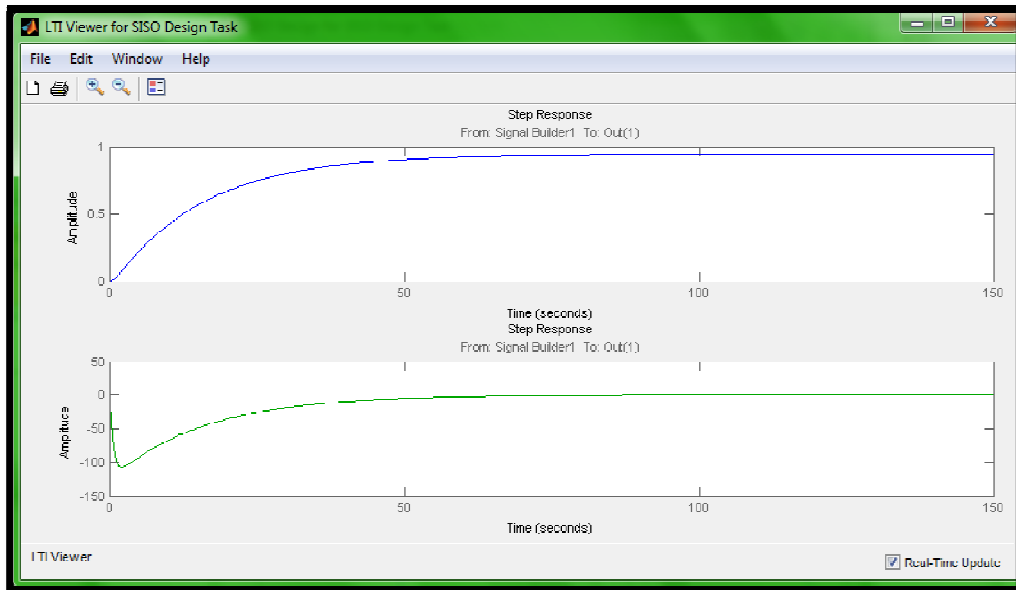


Figure 7- 26: Step response plot

The root locus diagram represent the linearize system and step response plot that show the angle and force of the blade. As shown in the Figure 7- 26 it takes a very long time to the system to react or to reach the command angle (it’s around 100 sec).

So it can be use this plot to adjust the compensator in two ways as follows.

I. Manual compensation (better design)

Dragging the plot using the mouse can increase the proportional gain in the root locus diagram which makes step response plot to update automatically. Therefore it gives better reacting time than before, but it is uncertain.

II. Automatic compensation (optimized design)

It use Automatic optimization based tuning to create an initial compensator design or to refine the current compensator design. Tune elements or parameters such as poles, zeros and gains within any controller in the system and optimize the open and closed loop responses.

The response time is around 100 second as shown in the Figure 7- 26.first let reduce the time axis in LTI viewer preferences to 5 second in order to make the response time on this range.

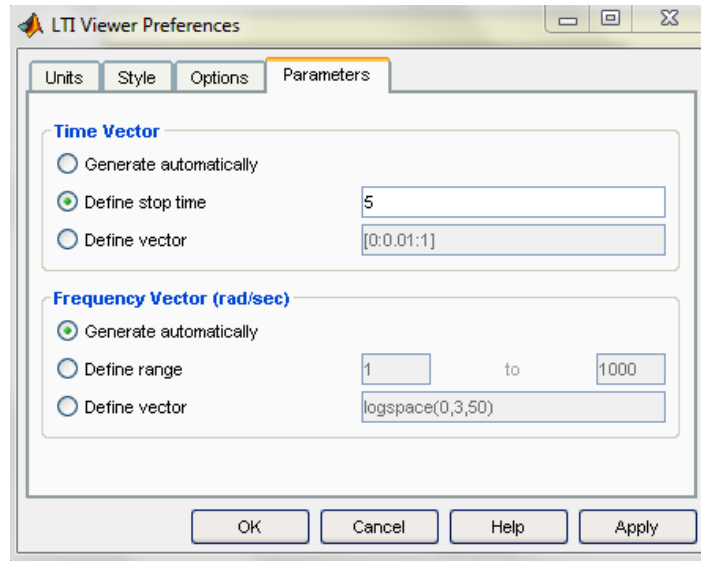


Figure 7- 27: LIT viewer

The following automatic optimization takes place to made the response in the range of point A and B as shown in the Figure 7- 28.

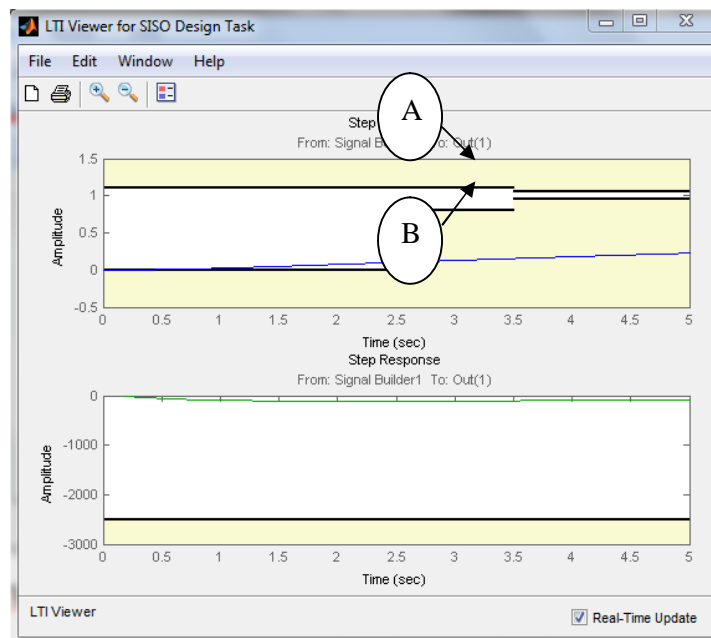


Figure 7- 28: LIT viewer

Figure 7- 29 and 7-30 show the pitch command and force of actuator gradually approaching the step response. This happen when the value of proportional and integral gain of the PI controller becomes within the range of the required gain of the control system.

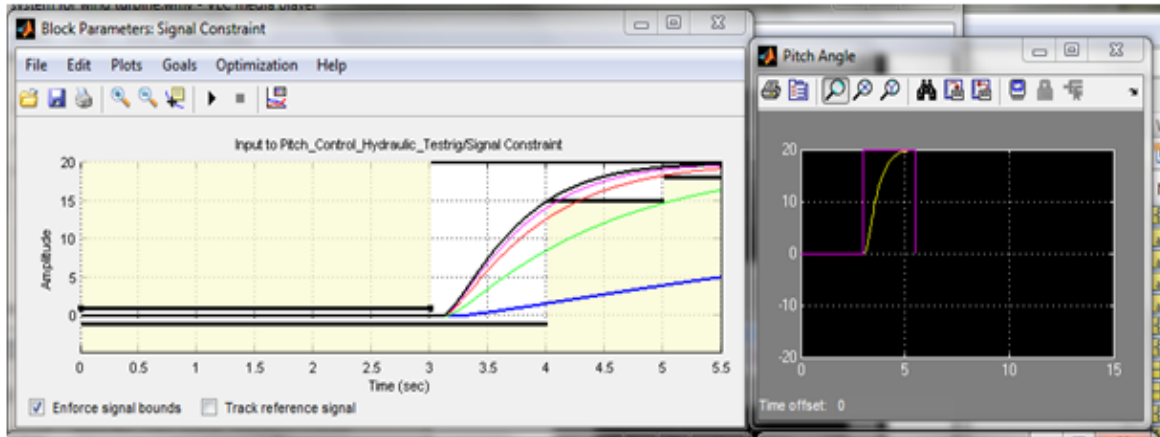


Figure 7- 29: Pitch command

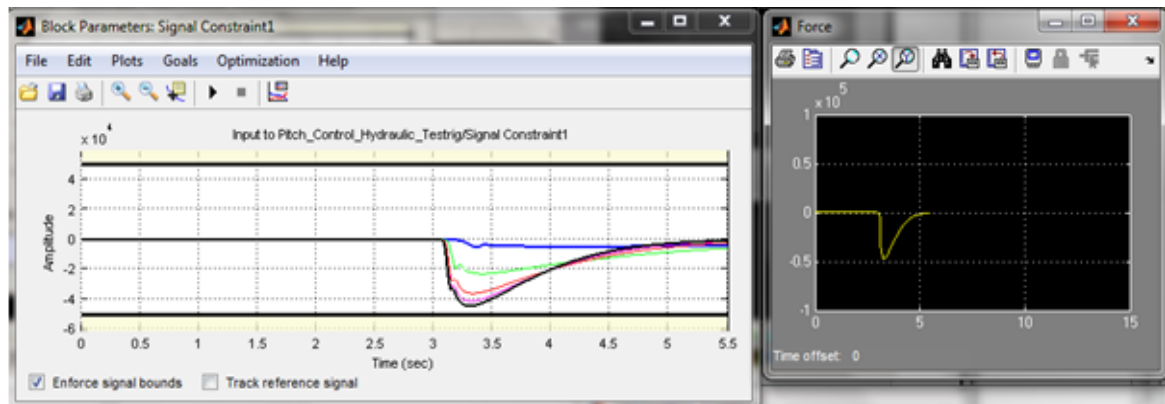


Figure 7- 30: Force command

After automatically compensating, the proportional gain and integral gain value of the system increase which makes pitch actuator response faster than the previous. The control and estimation tools manager summarized as:

Initial Design Value of PI Controller before Optimization (Initial Design)

The proportional and integral gain value which indicated in the Figure 7- 31 are initially designed values before optimization.

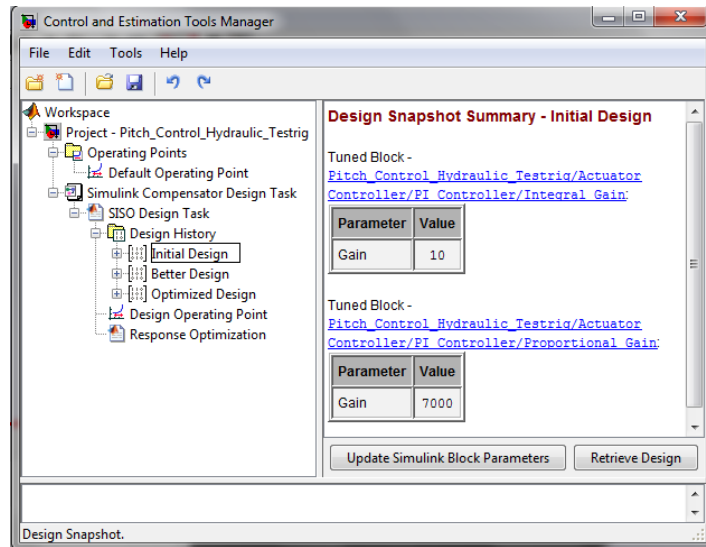


Figure 7- 31: Initial design value of proportional and integral gain

1. Better Design Value Of PI Controller

The proportional and integral gain value which indicated in the fig is better value that comes as a result of manually adjustment of the gain of pi controller.

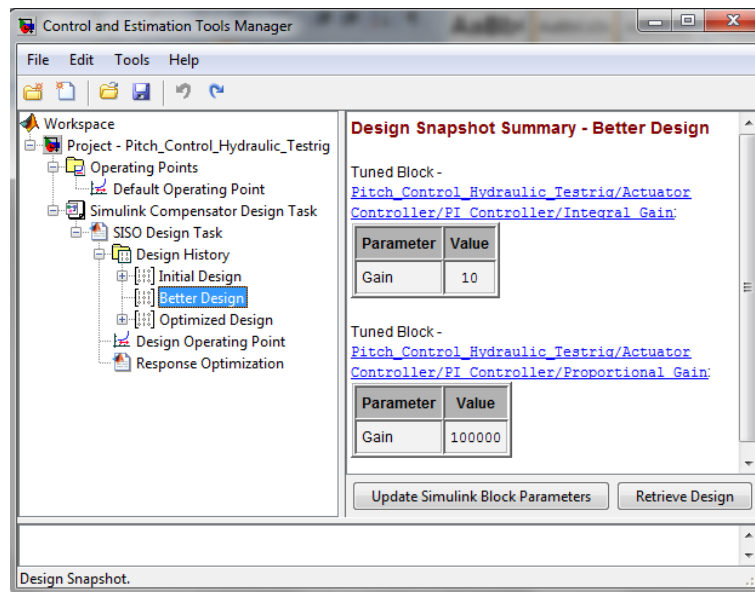


Figure 7- 32: Better design value of proportional and integral gain

The process of manually modifying the system to work is not efficient or use large resources. The design should be optimized to make best use of the available resources.

2. Optimized design value of PI controller

The proportional and integral gain value which indicated fig 7-33 are the values which comes as a result of Simulink optimization of the PI controller.

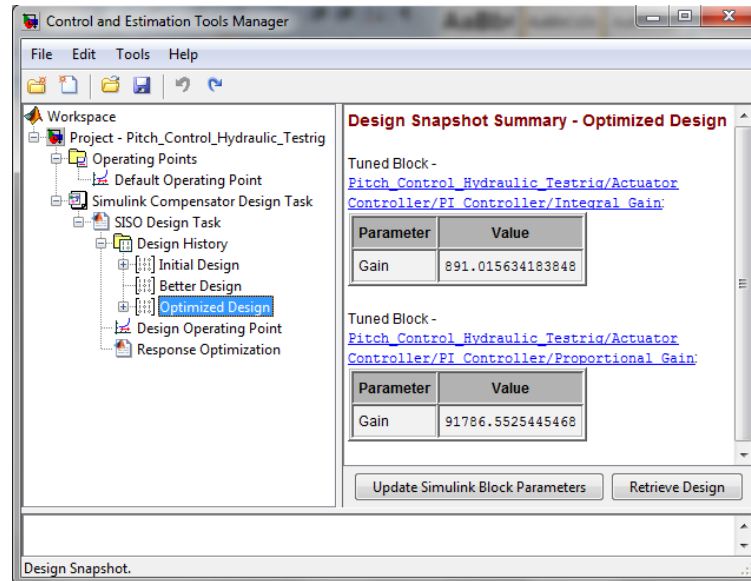


Figure 7- 33: Optimized design value of proportional and integral gain

Therefore this value use for optimized design of the model in order to actuate the blade in required speed and force.

Comparing the Results

After updating the Simulink block parameter into the current optimized value of the compensator. The optimization result becomes

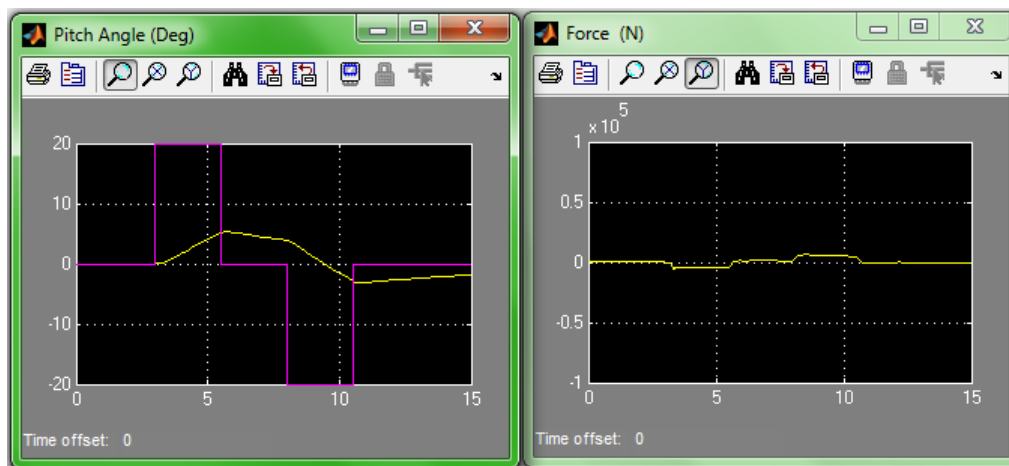


Fig A) Before optimization (Initial design)

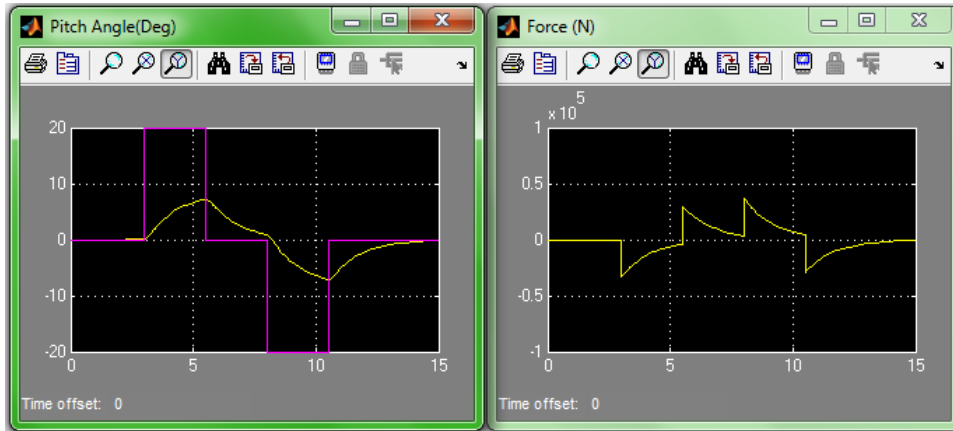


Fig B) Manual compensation (better design)

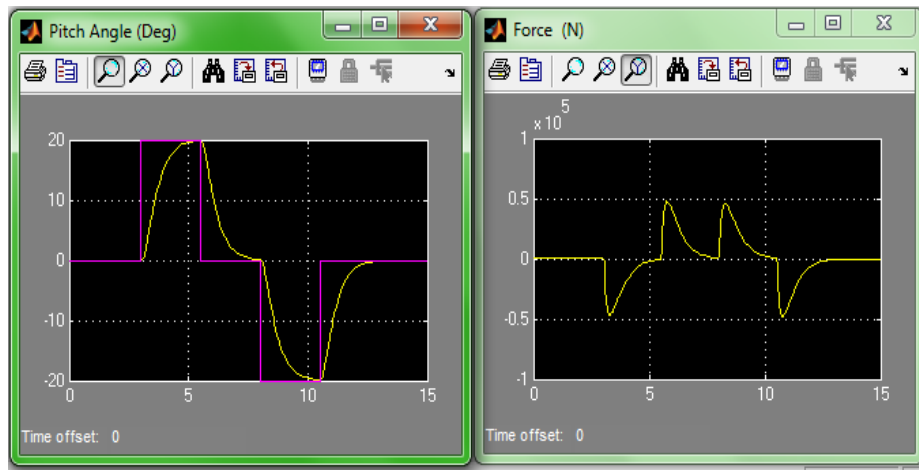


Fig C) After optimization (optimized design)

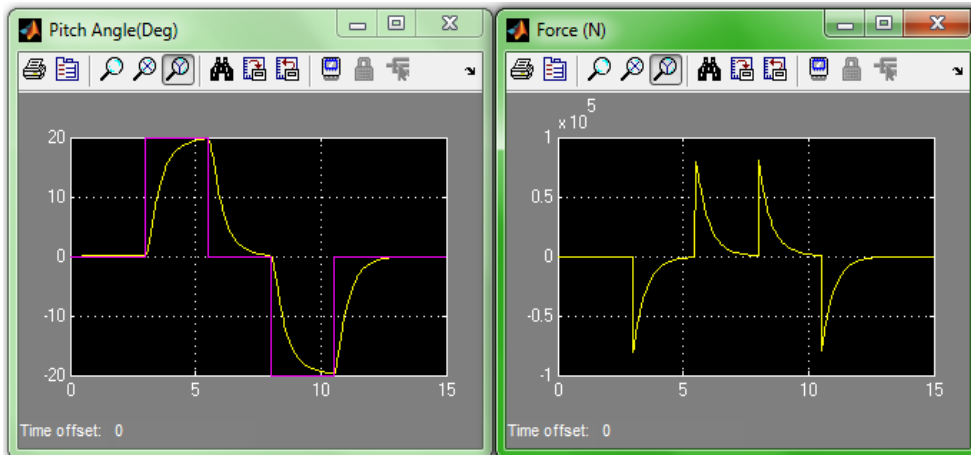


Fig D) Ideal design

Figure 7- 34 Compensation design results

The following Figure 7- 35 show the comparison of the ideal and optimized design

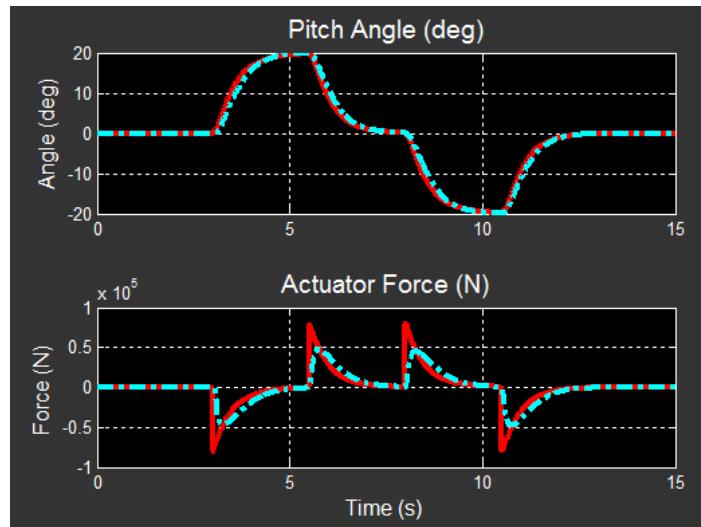


Figure 7- 35 Comparison of the ideal and optimized design

After optimization, the result is

- The pitch angle approaches the command angle Figure 7- 34 (C)
- The system react better than initial and manual compensating design Figure 7- 34 (C).
- The optimized design approaches the ideal pitch actuation system of Figure 7- 35.

Therefore, the force of the pitch actuator is high enough to actuate the blade to catch the wind or it reacts on the time and speed that the design requirement need relative to the step response.

Now the blade can be pitched with in 5 sec to catch the wind when the wind speed is between cutting and cutout or to be pitched away from the wind when the wind speed is above rated.

The optimized pitch angle and force will give information:

- To size the actuator
- To picking from the catalogue or
- To design ourselves

The control action represents the blade pitch used to control power above rated, then when the power drops below rated the pitch will be limited to the fine pitch setting and will not be allowed to drop further. In this situation the integral term of the PI will grow

more and more negative as the power remains below rated. Then when the wind speed rises again and the power rises above rated, the integral term will start to grow again towards zero, but until it gets close to zero it will more than compensate for the proportional and derivative terms. Therefore the pitch may remain ‘stuck’ at fine pitch for a considerable time, depending on how long the power has been below rated, until the integral term has come back closet zero.

7.5 Yaw System

The Yaw drive is the name given to the mechanism used to rotate the Nacelle with respect to the tower on its slewing bearing, in order to keep the turbine facing into the wind and to unwind the power and other cables when they become excessively twisted. It usually consists of an electric servomotor mounted on the Nacelle, which drives a pinion mounted on a vertical shaft via a reducing gearbox. The pinion engages with gear teeth on the fixed slewing ring bolted to the tower, as shown in Figure 7- 36. These gear teeth can either be on the inside or the outside of the tower, depending on the bearing arrangement, but they are generally located on the outside on smaller machines so that the gear does not present a safety hazard in the restricted space available for personnel access.

The wind turbine is said to have a Yaw error, if the rotor is not perpendicular to the wind. A Yaw error implies that a lower share of the energy in the wind will be running through the rotor area.

Wind turbines which are running with a Yaw error are therefore subject to larger fatigue loads than wind turbines which are Yawed in a perpendicular direction against the wind.

The servo motor connected to the Yaw gear using mechanical linkage revolute then with the Yaw ring using gear constraints as shown in Figure 7- 36.

The two bodies (Yaw gear and Yaw ring) connected by a gear constraint block are each restricted to turn relative to another along pitch circles centered at each body. The pitch circle centers are the origins of the body coordinate systems (CSs) at which the gear constraint block is connected on either side. The pitch circles are tangent at one contact point.

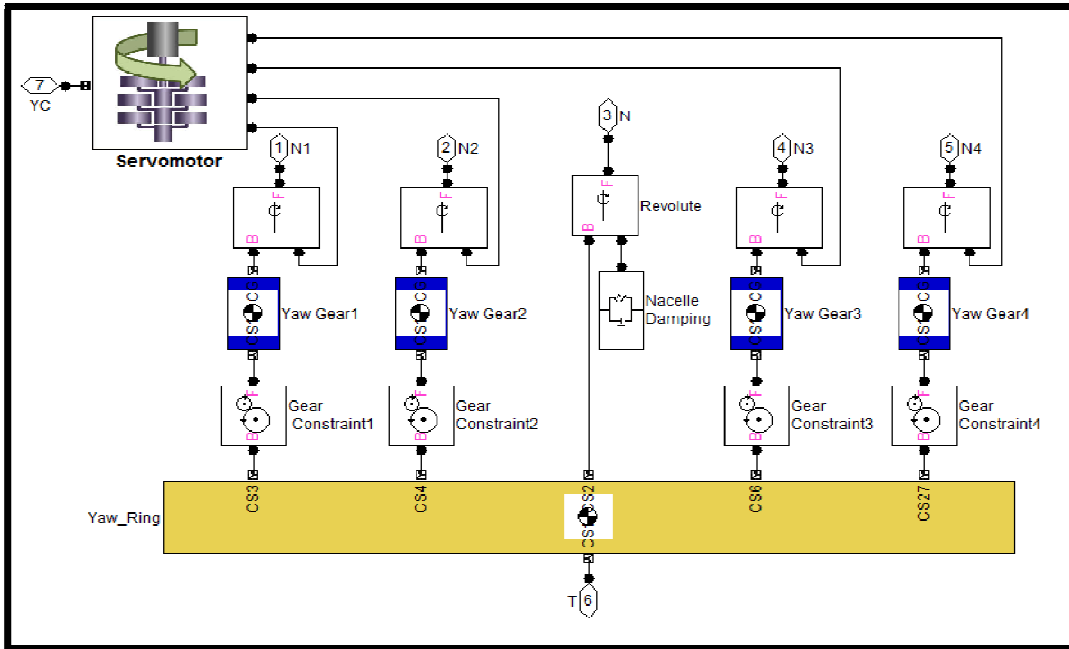


Figure 7- 36: Yaw control system

The two bodies connected by a gear constraint to a third, carrier body (Nacelle N) by revolute or cylindrical joints.

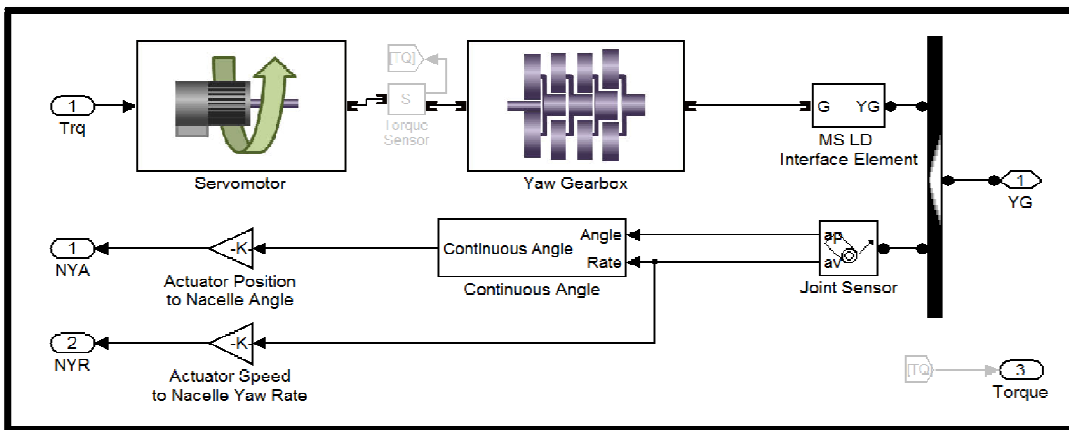


Figure 7- 37: Subsystem of yaw motor

In the Subsystem of the servo motor, there are four Yaw motor inside the Yaw system in order to give high torque to point the wind turbine into the wind direction as shown in the Figure 7- 38.

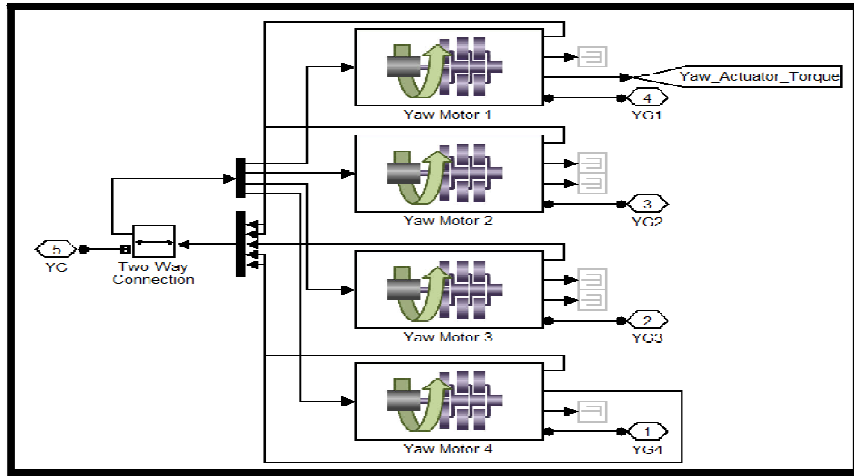


Figure 7- 38: Subsystem of the servo motor

Each Yaw motor connected to each Yaw gear (YG). They actuated when the command signal comes from the Yaw controller (YC). Then the torque from the servo motor (YM) transmitted to the Yaw gear box

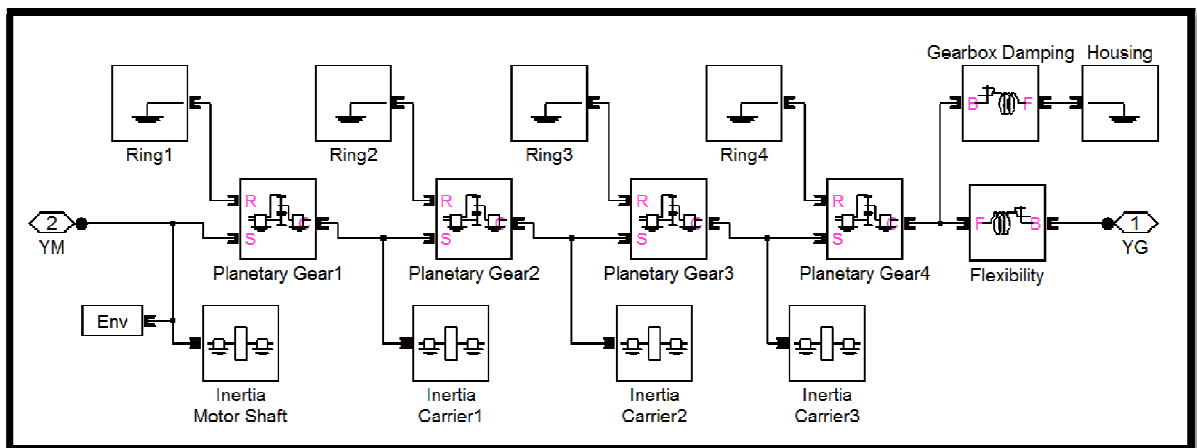


Figure 7- 39: Subsystem of the Yaw gearbox

The Planetary Gear block represents a set of carrier, ring, planet and sun gear wheels and its ring transmute the torque into the Nacelle.

7.4.1 Yaw controller

The Yaw mechanism turns the Nacelle with the hub and the blades so that they face directly into the wind. It is situated between the Nacelle and the tower. The control of the Yaw system is influenced by the wind direction, measured on top of the Nacelle.

As most horizontal-axis wind turbines employ a Yaw drive mechanism to keep the turbine headed into the wind, the use of the same mechanism to Yaw the turbine out of wind to limit power output is obviously an attractive one.

Let's look into the Nacelle from the top view the Yaw actuator system rotates the Nacelle about an axis as shown in Figure 7- 40.

Tower attached with Yaw ring at different side of Yaw ring. There are Yaw gears and the Yaw gears driven by Yaw motor. As the Yaw motor rotates the Yaw gear, the Yaw gears rotate the Yaw ring which rotates the Nacelle.

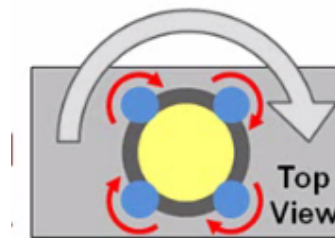


Figure 7- 40: Top view of Nacelle

Its representative Simulink model of Yaw controller shown below In figure 7-41.

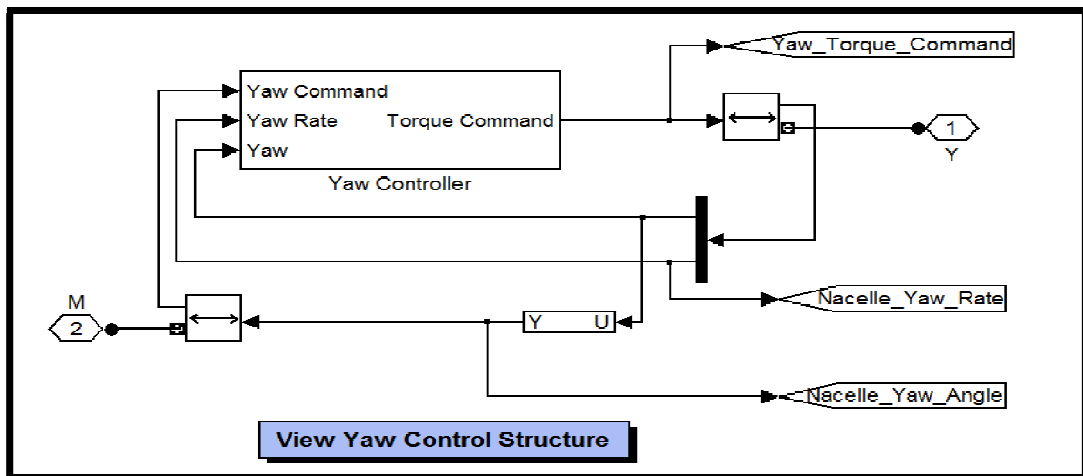


Figure 7- 41: Yaw controller

The Servomotor block represents a brushless motor with closed-loop torque control. This block abstracts the torque-speed behavior of the combined motor and motor driver in order to support system-level simulation where simulation speed is important.

The Yaw command which is the angle the Nacelle should point. One of the necessary things the project need to figure out how much torque is the motor should produce.

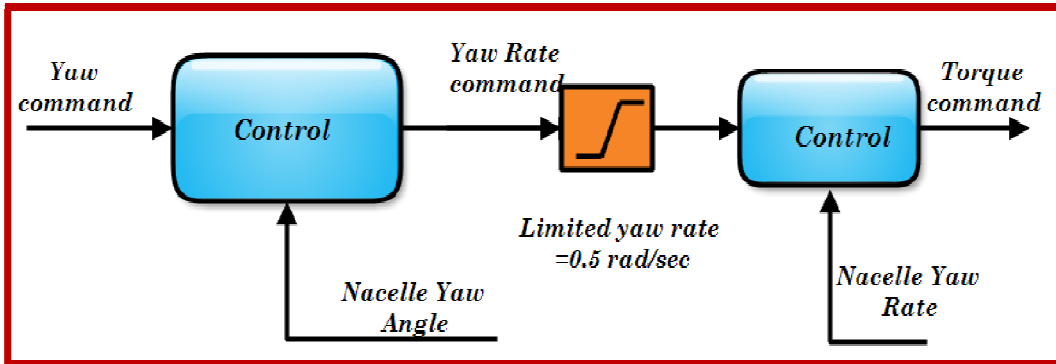


Figure 7- 42: Control structure for Yaw control

The Yaw control system look at the angle the Nacelle has compared to the commanded angle and producing the Yaw rate command. The Yaw rate command will then be limited the requirement to the system the Yaw actuation system cannot rotate faster than 0.5 deg/sec.

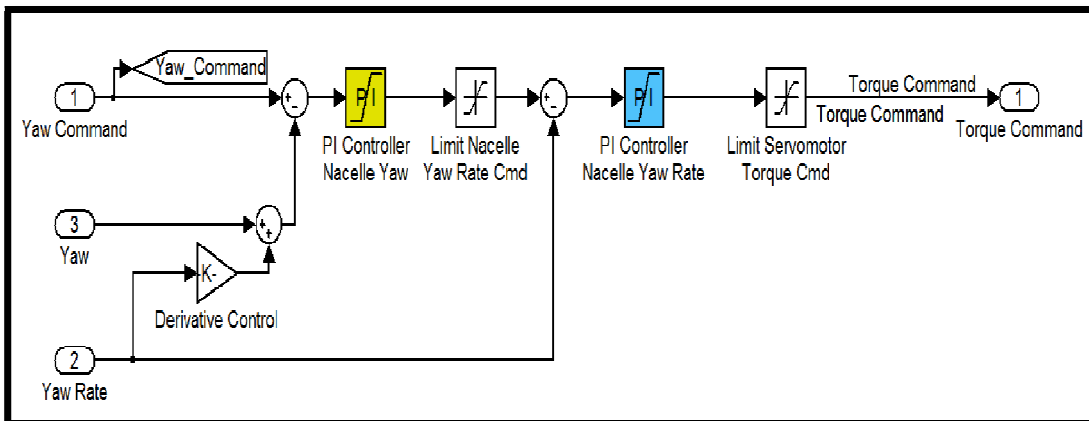


Figure 7- 43: Simulink control structure for Yaw control.

Then it will compare the Yaw rate command with Nacelle Yaw rate in order to get the torque command. Based on the torque command the servomotor actuates the system.

Result: After compensation design analysis like pitch control system, the Yaw actuator system response based on the required with the required torque, time and speed. There are couples of scope from Figure 7-44 to 7-47 and they show how the system performs,

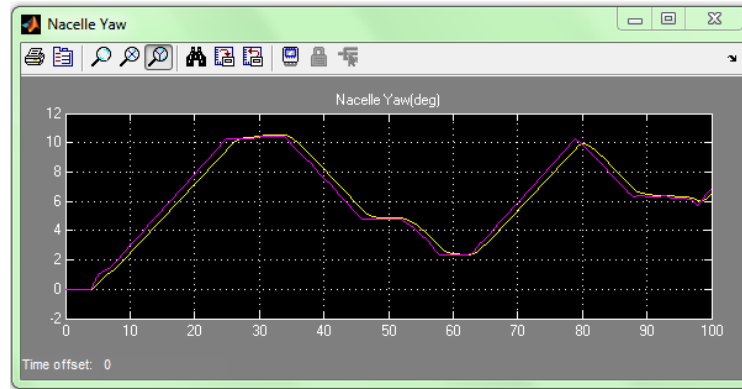


Figure 7- 44: Nacelle Yaw

In Figure 7- 44 the Nacelle rotate or pointed in the direction the same as the wind direction. This means the angle of Nacelle pointed in to the wind and yellow line shows how the Yaw system attempted to track that Yaw command angle.

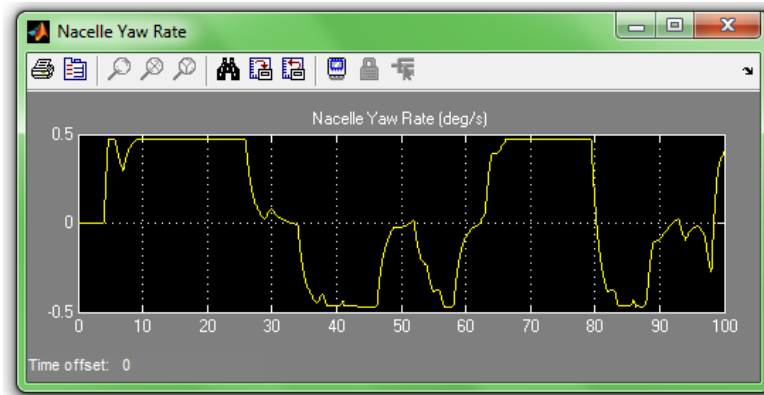


Figure 7- 45: Nacelle Yaw rate

Figure 7- 45 show that the Nacelle Yaw rate is below the limited value of **0.5 deg/sec**.

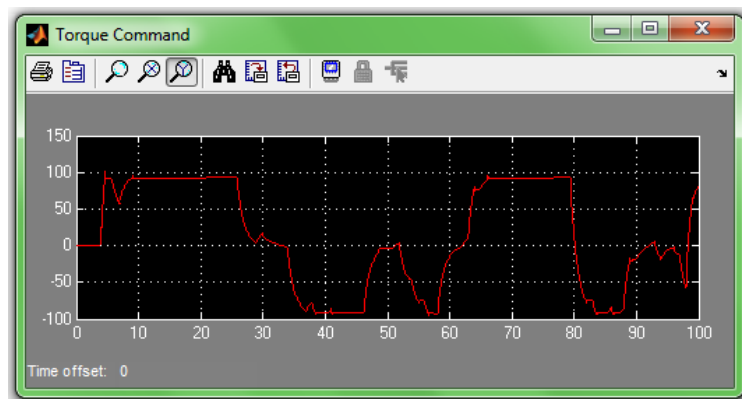


Figure 7- 46: Torque Command

The torque command is the command which comes from the main controller to limit the amount of torque needed to rotate the Nacelle.

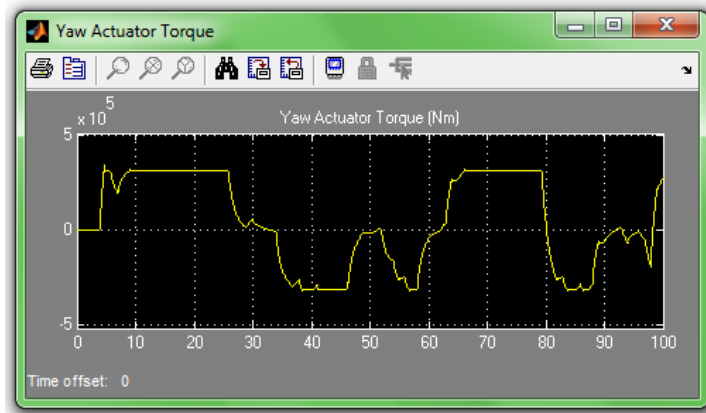


Figure 7- 47: Yaw actuator torque

Figure 7- 47 show how much the Yaw actuator torque has to produce in order to keep the Nacelle pointed into the wind. It use the command (Figure 7-46)from the main controller.

All the above figures show the Yaw system actuator actuates the system based on the torque command. Therefore the Yaw system performs based on the requirement. Look into the Appendix A to see the Yaw requirement.

Chapter Eight

8.1 Conclusions and Recommendations for Future Work

8.1.1 Conclusion

1. In traditional development process, the design of the wind turbine was organized in different environment, tools and department then tested after prototype. As a result, delivery or quality problem arise this makes ROI to decrease. But in modal based design the test and integration will done before prototype in single environment which reduce the expensive testing and prototype costs.
2. In this wind turbine model any wind data of specific site can be analyzed and optimized to get fine response of all system component.
3. For any given wind data, the turbine characteristics report can be auto generated in the model with in two minute interval.
4. Supervisory control determine the operating condition of the turbine depend on the input value of wind speed and direction. The turbine generally starts operating in winds at hub height of about 4 m/s and reaches its rated power output at about 13 m/s. At about 25 m/s the control system will shut down the turbine by pitching the blades.
5. The power output of wind turbine model is regulated by blade pitch control, in which the blades are rotated about their long axis to adjust its aerodynamic angle of attack with respect to the relative wind. Torque and speed sensors on the generator and drive train enable the blade-pitch controller to regulate the power output and rotor speed to prevent overloading structural components.
6. Wind direction sensors on the Nacelle send command to Yaw controller to actuate the system toward the wind direction.
7. Compensation design analysis used by optimizing the proportional and the integral gain for all control systems to track and to be close to the command angle based on the requirement specified.

8.1.2 Recommendations

The following areas of interest can be looked to extend the research work on Model Based Design of HAWT and Its Control System under Simulink Environment.

- Additional practical studies can be done using data logger to see the practical applicability of the project.
- The control system of this project further can be made in to **Embedded Systems/PIC Microcontroller**, which is, a family of Harvard architecture microcontrollers(a computer architecture with physically separate storage and signal pathways for instructions and data) made by Microchip Technology.

Reference

1. Carlin, P.W., A.S. Laxson and E.B. Muljadi, ***The History and State of the Art of Variable-Speed Wind Turbine Technology***. *Wind Energy*, 2003. 6: p. 129-151.
2. Tomas Petru, ***Modeling of Wind Turbines for Power System Studies***, Doctoral Thesis, *Applied Energy*, Volume 87, Issue 2, February 2010, Pages 120-143
3. Slootweg JG, de Haan SWH, Polinder H, Kling WL, ***General model for representing variable speed wind turbines in power system dynamics simulations***. *IEEE Transactions on Power Systems* 2003; 28:144–151.
4. Nicholas W. Miller, Juan J. Sanchez-Gasca, William W. Price, ***Dynamic Modeling of GE 1.5 and 3.6 MW wind turbine-generators for stability simulations***, *GE Power Systems Energy Consulting, IEEE WTG Modeling Panel, Session July 2003*
5. Sheldahl, R.E. and Klimas, P.C., ***Aerodynamic Characteristics of Seven Airfoil Sections Through 180 Degrees Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines***, SAND80-2114, March 1981, Sandia National Laboratories, Albuquerque, New Mexico.
6. David James Burnham, ***Control of Wind Turbine Output Power***, Msc Thesis Work, University of Texas at Austin, Mechanical Eng'g Department, 2009
7. Richard Gagnon, Gilbert Sybille, Serge Bernard, Daniel Paré, ***Modeling and Real-Time Simulation of a Doubly-Fed Induction Generator Driven by a Wind Turbine***, *IEE Proc.-Electr. Power Appl.*, Vol. 143, No. 3, May 1996
8. W.E. Leithead, M.C.M. Rogers, ***Drive-train Characteristics of Constant Speed HAWT's: Part I - Representation by Simple Dynamics Models***, *Wind Engineering* Vol. 20 No. 3 (1996)
9. W.E. Leithead, M.C.M. Rogers, ***Drive-train Characteristics of Constant Speed HAWT's: Part II - Simple Characterization of Dynamics***, *Wind Engineering* Vol. 20 No. 3 (1996)
10. J. Wilkie, W.E. Leithead, C. Anderson, ***Modeling of Wind turbines by Simple Models***, *Wind Engineering* Vol. 13 No. 4 (1990)
11. David A. Spera, ***Fundamental Concepts of Wind Turbine Engineering***, second edition 2003. : p. 48-72.

12. *T. Thiringer, Measurements and Modeling of Low-Frequency Disturbances in Induction Machines, Ph.D. thesis, Chalmers University of Technology 1996,ISBN 91-7197-384-2*
13. *The MathWorks, Sim Power Systems For Use with Simulink, User's Guide Version*
14. *K. Ogata, Modern Control Engineering, Upper Saddle River, NJ: Prentice-Hall, Inc., 1996.*
15. *S. J. Chapman, Electric Machinery Fundamentals, Boston: McGraw-Hill, 3rd ed., 1999.*
16. *J. Campbell, A Scaled Electro-Mechanical Model of a Wind Turbine Power Train, Master's thesis, The University of Texas at Austin, Austin, TX,May 2008.*
17. *Hansen AD, Michalke G., Modeling And Control Of Variable-Speed Multi-Pole Permanent Magnet Synchronous Generator Wind Turbine. Wind Energy2008; 11: 537–554.*
18. *Jauch C. Transient And Dynamic Control Of A Variable Speed Wind Turbine With Synchronous Generator. Wind Energy 2007; 10: 247–269.*
19. <http://www.vistadatavision.com/environmental-monitoring/>
20. <http://www.ekopower.nl/wind-energy-monitoring-data-logger-data-processing.htm>
21. <http://www.ppart.de/aerodynamics/profiles/NACA4.html>
22. <http://www.cyberiad.net/foildata.htm>

Appendix A: Wind Turbine System Requirements

Table A1: Blade Requirements 

FBlade length	40 m
Material	Carbon/wood/glass/epoxy
Standard color	RAL 7035
Gloss	Class 2: (30-70%) to be measured acc. to DS/ISO2813
Blade profiles	NACA 8412
Twist	20°
Largest chord	3.08 m

H

Table A2: Brakes Requirements 

Mechanical	
Type description	Active Brake
Brake disc	Steel, mounted on high speed shaft
Number of calipers	2 piece
Brake Hydraulics	
Voltage	3 x 480 V
Working pressure range	140-150 bar
Oil capacity	11 L

Table A3: Environment Requirements 

Temperature interval for operation	-30 to +30°C
Temperature interval for structure	-40 to +50°C

Table A4: Gear Train Requirements 

Type description	1. step planet, 2. step helical
Gear house material	Cast
Ratio	1:84.3
Mechanical power	1800 kW
Bending strength acc. to ISO 6336	SF > 1.6
Surface durability acc. to ISO 6336	SH > 1.25
Scuffing safety acc. to DNV 41.	SS > 1.3
Shaft seals	Labyrinth
Oil sump	App. 250 l

Table A5: Generator Requirements 

Type description	1 speed generator, water cooled
Rated power	2650 kW
Apparent power	2808 kVA
Rated current I_N	2740 A
Max power at Class F PF _{ma}	2815 kW
Max current at Class F IF _{max}	2914 A
No load current I_0	430 A
Number of poles P	6
Synchronous rotation speed n_0	1200 rpm
Rotation speed at rated power N_n	1214 rpm
Slip at rated power N_n	0.0117
Voltage U_N	3 x 600 V
Frequency F	60 Hz
Coupling	Δ
Enclosure	IP54
Insulation class/ Temperature increase	F/B

Table A6: Main Controller Requirements 

Annual average wind speed 8.5 m/s	8.5 m/s
Wind shear 0.20	0.2
Extreme wind speed	42.5 m/s (10 min. average)
Survival wind speed 59.5 m/s (3 sec. average)	59.5 m/s (3 sec. average)
Automatic stop limit 20 m/s (10 min. average)	20 m/s (10 min. average)
Re-cut in 18 m/s (10 min. average)	18 m/s (10 min. average)
Characteristic turbulence intensity	16% (including wind farm turbulence)
Maximum in-flow angle	8°

Table A7: Tower Requirements 

Type Description Conical, tubular	Conical, tubular
Material Welded steel plate	Welded steel plate
Corrosion class, outside Acc. to DS EN ISO 12944: C5 I	Acc. to DS EN ISO 12944: C5 I
Color RAL 7035	RAL 7035
Access conditions	Internal, safety harness, ladder cage
Tower height	80m

Table A8: Nacelle Requirements 

Material EN-GJS-400-18U-LT	EN-GJS-400-18U-LT
Standard color RAL 7035	RAL 7035
Corrosion class, outside Acc. to DS EN ISO 12944:C5 I	Acc. to DS EN ISO 12944:C5 I
Rotor	
Number of blades 3 pieces	3 pieces
Tip speed (synchronous) 61.8 m/s	61.8 m/s
Rotor shaft tilt 5°	5°
Eccentricity (tower center to hub center)	3447 mm
Solidity (Total blade area/rotor area)	0.05
Rotor orientation	Upwind

Table A9: Pitch Actuation Requirements 

Hydraulic pressure	2e7 Pa
Accumulator Capacity	0.1 L
Accumulator Preload Pressure	1.5e7 Pa
Accumulator Maximum Pressure	2.5e7 Pa

Table A10: Pitch Controller Requirements

Track angle within	1 degree
Rise Time	3 seconds
Settling Time	5 seconds

Table A11: Yaw Actuation Requirements 

Type description	Planetary gear motor
Gear ratio of yaw gear unit	app. 1:1687
Voltage	3 x 480 V
Rotational speed at full load	1140 rpm
Number of yaw gears	4 pieces
Yaw Brake	Hydraulic disc brake
Number of Yaw Friction Units	6 pieces
Voltage	3 x 480 V
Working pressure range	140-150 bar
Oil capacity	App. 10 l.

Table A12: Yaw Controller Requirements 

Max Yaw Rate	0.5 deg/sec
--------------	-------------

Appendix B: Wind Turbine Parameters

%%% STANDARD CONSTANTS

```
radps2rpm = (1/(2*pi))*60;  
degps2radps = pi/180;  
  
torque_pu2Nm = (1/(2*pi))*60;
```

% PHYSICAL PARAMETERS

```
steel_density = 7.85*(100)^3/1000; % kg/m^3  
hydr_fluid_density = 961.873; % kg/m^3
```

```
%HYDR_BONUS_FACTOR = 0.25*1e-3;
```

%%% SIMPLE LIFT AND DRAG PARAMETERS

```
WT_Params.Lift_Drag_Coeff. angle_of_attack = [-12.5 -7.5 -5 -1 0 2 3  
6 8 10 12.5 15 16 18 20 25 30 40 50 60 70 80 90  
100];
```

```
WT_Params.Lift_Drag_Coeff. coeff_lift = [-1.1 -0.7 -0.5 0 0.2 0.42  
0.65 0.9 1.1 1.3 1.4 1.5 1.4 1.2 1.05 1.05 1.0 0.95 0.8 0.6 0.4 0.2 0  
0];
```

```
WT_Params.Lift_Drag_Coeff. coeff_drag = [0 0 0 0 0 0 0  
0 0 0 0.01 0.02 0.05 0.07 0.15 0.3 0.4 0.7 0.9 1.05 1.15 1.2 1.25  
1.3];
```

```
WT_Params.Lift_Drag_Coeff. air_density = 1.22;  
WT_Params.Lift_Drag_Coeff. lift_factor = 200;  
WT_Params.Lift_Drag_Coeff. drag_factor = 10;
```

```
WT_Params.Air.Density = 1.22; % [kg/m^3] (std. day sea-level)  
WT_Params.Air.SpeedOfSound = 340; % [m/s] (std. day sea-level)
```

%%% ROTOR DISC AIRFOIL LIFT & DRAG POLAR (based on NACA8412)

```
WT_Params.Rotor.MaxLiftCoefficient = 1.4;  
WT_Params.Rotor.ParasiteDragCoefficient = 0.0070;  
WT_Params.Rotor.InducedDragFactor = 0.0040;  
WT_Params.Rotor.MaxDragCoefficient = 1.8;
```

%%% NACA8412 airfoil lift and drag coefficients

```
txt = importdata('naca8412cl.txt', ' ', 30);  
WT_Params.NACA8412.Re = txt.data;
```

```

txt = importdata( 'naca8412cl.txt', ' ', 32 );
WT_Params.NACA8412.AoA = txt.data(:,1) /180*pi;
WT_Params.NACA8412.Cl = txt.data(:,end);

txt = importdata( 'naca8412cd.txt', ' ', 32 );
WT_Params.NACA8412.Cd = txt.data(:,end);

%%% TOWER PARAMETERS

WT_Params.Tower.mass = 1000;
WT_Params.Tower.inertia = 1000*[100 0 0;0 1 0;0 0 100];
WT_Params.Tower.height = 100;
WT_Params.Tower.base_radius = 2; % m
WT_Params.Tower.top_radius = 1.15; % m

%%% BLADE PARAMETERS

WT_Params.Blade.mass = 6600;
WT_Params.Blade.inertia= 100*[100 0 0;0 1 0;0 0 100];

Rectangular_Block_Inertia(1.5, 20, 3.4, 0.001*steel_density);

WT_Params.Blade.inertia = 1.0e+004*[2.7461 0 0;0 0.0921 0;0 0 2.6840];
WT_Params.Blade.length = 40;
WT_Params.Blade.width = 2;
WT_Params.Blade.damping = 1000;

WT_Params.Blade.Actuator_Offset.x = 0;
WT_Params.Blade.Actuator_Offset.y = -0.1;
WT_Params.Blade.Actuator_Offset.z = 0.5;

WT_Params.Blade.Geometry.root_radius = 0.6;
WT_Params.Blade.Geometry.ctr_to_leading_edge.z = 1.2;
WT_Params.Blade.Geometry.ctr_to_leading_edge.x = 0;
WT_Params.Blade.Geometry.ctr_to_a1.z = 1;
WT_Params.Blade.Geometry.ctr_to_a1.x = 0.4;
WT_Params.Blade.Geometry.ctr_to_a2.z = 1;
WT_Params.Blade.Geometry.ctr_to_a2.x = -0.45;
WT_Params.Blade.Geometry.ctr_to_b1.z = 0.75;
WT_Params.Blade.Geometry.ctr_to_b1.x = 0.7;
WT_Params.Blade.Geometry.ctr_to_b2.z = 0.75;
WT_Params.Blade.Geometry.ctr_to_b2.x = -0.7;
WT_Params.Blade.Geometry.ctr_to_c1.z = 0.3;
WT_Params.Blade.Geometry.ctr_to_c1.x = 0.8;
WT_Params.Blade.Geometry.ctr_to_d1.z = -0.6;
WT_Params.Blade.Geometry.ctr_to_d1.x = 0.55;
WT_Params.Blade.Geometry.ctr_to_e1.z = 0.2;
WT_Params.Blade.Geometry.ctr_to_e1.x = 0;
WT_Params.Blade.Geometry.ctr_to_trailing_edge.z = -2.2;
WT_Params.Blade.Geometry.ctr_to_trailing_edge.x = -0.7;

```

```
WT_Params. Blade.Geometry.scaling_tip = 0.5;
```

%%% ROTOR PARAMETERS

```
WT_Params.Rotor.mass = 8500;  
WT_Params.Rotor.inertia= 10*[1 0 0;0 10 0;0 0 1];  
WT_Params.Rotor.radius = 2;  
WT_Params.Rotor.Geometry.Length = 1.6*WT_Params.Rotor.radius;  
WT_Params.Rotor.Pitch_Actuator_Offset.x = 1.2*WT_Params.Rotor.radius;  
WT_Params.Rotor.Pitch_Actuator_Offset.y =  
WT_Params.Rotor.radius+WT_Params.Blade.Actuator_Offset.y;  
WT_Params.Rotor.Pitch_Actuator_Offset.z = 0.5;  
WT_Params.Rotor.Pitch_Actuator_Offset.radius = 1.8;  
WT_Params.Rotor.Pitch_Actuator_Offset.angle = 15*0;  
WT_Params.Rotor.damping = 100;  
  
WT_Params.Rotor.nominal_rpm = 14.3004; %RPM  
WT_Params.Rotor.min_rpm = WT_Params.Rotor.nominal_rpm*0.95; %RPM  
WT_Params.Rotor.max_rpm = WT_Params.Rotor.nominal_rpm*1.05; %RPM
```

%%% GEARTRAIN PARAMETERS

```
WT_Params.Geartrain.mass = 23000;  
WT_Params.Geartrain.Gear_Ratio.first_step = 4;  
WT_Params.Geartrain.Gear_Ratio.planetary = 5;  
WT_Params.Geartrain.Gear_Ratio.second_step_primary_generator = 3.5125;  
WT_Params.Geartrain.inertia_carrier_shaft = 10;  
WT_Params.Geartrain.inertia_sun_shaft = 10;  
WT_Params.Geartrain.inertia_first_step_shaft = 10;  
WT_Params.Geartrain.MSLD_Interface_TF_Coeff = 1e-4;
```

%%% PITCH CONTROLLER PARAMETERS, INNER LOOP

```
WT_Params.Pitch_Controller.P_Gain = 100000;  
WT_Params.Pitch_Controller.I_Gain = 10;  
WT_Params.Pitch_Controller.D_Gain = 0.1;  
WT_Params.Pitch_Actuator.hydraulic_actuator_scale_factor = 6.2500e-010;  
WT_Params.Pitch_Actuator.ideal_actuator_scale_factor = 4.7;
```

% OUTER LOOP -- BLOCKS IN LIGHT GRAY BENEATH MAGENTA BLOCK

```
WT_Params.Pitch_Controller.Angle_of_Attack.P_Gain = 20;  
WT_Params.Pitch_Controller.Angle_of_Attack.I_Gain = 5;  
WT_Params.Pitch_Controller.Angle_of_Attack.D_Gain = 200;  
WT_Params.Pitch_Controller.Angle_of_Attack.max_angle = 10.26;  
WT_Params.Pitch_Controller.emergency_brake_pitch_angle = -95;  
WT_Params.Pitch_Controller.park_pitch_angle = 0;
```

```
WT_Params.Pitch_Controller.Angle_Of_Attack.max_lift = 15;
WT_Params.Pitch_Controller.Angle_Of_Attack.enough_lift = 10;
WT_Params.Pitch_Controller.Angle_Of_Attack.stall = 25;
```

%%% PITCH ACTUATOR PARAMETERS

```
WT_Params.Pitch_Actuator.hydraulic_pressure = 2e7;
WT_Params.Pitch_Actuator.hydraulic_tf_coeff = 0.0001;
WT_Params.Pitch_Actuator.Servo.gain = 20;
WT_Params.Pitch_Actuator.Servo.time_constant = 0.002;
WT_Params.Pitch_Actuator.Servo.saturation = 0.3;
WT_Params.Pitch_Actuator.servo2valve_conversion = 0.25*1e-3;
```

```
WT_Params.Pitch_Actuator.Valve.max_area = 5e-03;
WT_Params.Pitch_Actuator.Valve.max_opening = 0.003;
WT_Params.Pitch_Actuator.Valve.leakage_area = 1e-12;
WT_Params.Pitch_Actuator.Valve.initial_opening_pa = 0;
WT_Params.Pitch_Actuator.Valve.initial_opening_at = 0;
WT_Params.Pitch_Actuator.Valve.initial_opening_pb =
WT_Params.Pitch_Actuator.Valve.initial_opening_pa =
WT_Params.Pitch_Actuator.Valve.initial_opening_bt =
WT_Params.Pitch_Actuator.Valve.initial_opening_at;
```

```
WT_Params.Pitch_Actuator.Cylinder.damping = 500;
WT_Params.Pitch_Actuator.Cylinder.piston_area = 0.005;
WT_Params.Pitch_Actuator.Cylinder.stroke = 1.4967;
WT_Params.Pitch_Actuator.Cylinder.initial_disp_from_a = 0.4450;
WT_Params.Pitch_Actuator.Cylinder.contact_stiffness = 1e7;
WT_Params.Pitch_Actuator.Cylinder.contact_damping = 850;
WT_Params.Pitch_Actuation.Rotor_Connection.damping = 10000;
```

```
WT_Params.Pitch_Actuator.Geometry.outer_radius = 0.1;
WT_Params.Pitch_Actuator.Geometry.inner_radius = 0.03;
```

```
WT_Params.Pitch_Actuator.ori_fice_area = 1e-03;
```

```
WT_Params.Pitch_Actuator.Sensor.initial_position = 0;
```

```
WT_Params.Pitch_Actuator.Accumulator.capacity = 0.1;
WT_Params.Pitch_Actuator.Accumulator.preload_pressure = 1.5e7; % Pa
WT_Params.Pitch_Actuator.Accumulator.maximum_pressure = 2.5e7; % Pa
WT_Params.Pitch_Actuator.Accumulator.initial_volume = 0.1;
WT_Params.Pitch_Actuator.Emergency_Pitch_Valve.max_area = 5e-04;
WT_Params.Pitch_Actuator.constraint_tol = 1e-9;
```

```
WT_Params.Pitch_Actuator.cylinder_length = 0.8*WT_Params.Rotor.radius;
WT_Params.Pitch_Actuator.cylinder_inertia =[1.0358 0 0;0 39.1912 0;0 0
39.1912];
WT_Params.Pitch_Actuator.cylinder_mass = 207.1681;
```

```

WT_Params.Pitch_Actuator.rod_inertia = [0.0757 0 0;0 13.9908 0;0 0 13.9908];
WT_Params.Pitch_Actuator.rod_mass = 74.7445;
WT_Params.Pitch_Actuator.ideal_actuator_time_constant = 1e-1;
WT_Params.Pitch_Actuator.hydraulic_actuator_time_constant = 1e-1;

```

%%% GENERATOR PARAMETERS

```

WT_Params.Generator.mass = 8500;
WT_Params.Generator.nominal_power = 1.65e6/0.9;
WT_Params.Generator.voltage = 600;
WT_Params.Generator.frequency = 60;
WT_Params.Generator.Stator.resistance = 0.004843; % PER UNIT
WT_Params.Generator.Stator.inductance = 0.1248; % PER UNIT
WT_Params.Generator.Rotor.resistance = 0.004377; % PER UNIT
WT_Params.Generator.Rotor.inductance = 0.1791; % PER UNIT
WT_Params.Generator.mutual_inductance = 6.77; % PER UNIT
WT_Params.Generator.poles = 6;
WT_Params.Generator.Simple.damping = 20;
WT_Params.Generator.Simple.inertia = 250;

WT_Params.Transformer.mass = 8000;
WT_Params.Transformer.rated_power = 3.16e6;

WT_Params.Generator.Breakers.ron = 0.001;
WT_Params.Generator.Breakers.rp = 1e6; %OHMS
WT_Params.Generator.Breakers.cp = inf; %FARADS

WT_Params.Generator.radps2pu =
(1/(2*pi*WT_Params.Generator.frequency/(WT_Params.Generator.poles/2))); =
WT_Params.Generator.torque_pu2Nm =
WT_Params.Generator.nominal_power/(WT_Params.Generator.frequency*2*pi/(WT_
Params.Generator.poles/2)); =

```

%%% GRID PARAMETERS

```

WT_Params.Grid.voltage = 25e3;
WT_Params.Grid.frequency = WT_Params.Generator.frequency;

```

%%% BRAKE PARAMETERS

```

WT_Params.Brakes.stiffness = 0;
WT_Params.Brakes.damping = 1e6*10;
WT_Params.Brakes.torque_radius = 1;
WT_Params.Brakes.torque_radius = 1;
WT_Params.Brakes.peak_normal_force = 1e5*10;
WT_Params.Brakes.engagement_threshold = 1e-1;
WT_Params.Brakes.inertia_housing = 1e5*10;

```

%%% YAW ACTUATOR PARAMETERS

```
WT_Params.Yaw_Actuator.inertia_motor_shaft = 0.01;
WT_Params.Yaw_Actuator.inertia_carrier = 0.01;
WT_Params.Yaw_Actuator.p_gear_ratio_step_1 = 5;
WT_Params.Yaw_Actuator.p_gear_ratio_step_2 = 4.5;
WT_Params.Yaw_Actuator.p_gear_ratio_step_3 = 4;
WT_Params.Yaw_Actuator.p_gear_ratio_step_4 = 3;
WT_Params.Yaw_Actuator.yaw_gear_diameter = 0.368;
WT_Params.Yaw_Actuator.yaw_gear_thickness = 0.1;
WT_Params.Yaw_Actuator.Gearbox_Flexibility.stiffness = 1e5;
WT_Params.Yaw_Actuator.Gearbox_Flexibility.damping = 1e7;
WT_Params.Yaw_Actuator.damping = 100000;
WT_Params.Yaw_Actuator.MSLD_Interface_TF_Coeff = 1e-1;

WT_Params.Yaw_Actuator.yaw_ring_diameter = 2.3;
WT_Params.Yaw_Actuator.yaw_ring_thickness = 0.1;

WT_Params.Yaw_Actuator.Motor.max_torque_vector=[900 800 700 0]; %RPM
WT_Params.Yaw_Actuator.Motor.max_speed_vector=[0 600 1200 1410]; %Nm

WT_Params.Yaw_Actuator.Motor.control_time_constant = 0.02*0.01; %s
WT_Params.Yaw_Actuator.Motor.resistance = 3.5; %Ohm
WT_Params.Yaw_Actuator.Motor.damping = 10; %Nm/(rad/s)
WT_Params.Yaw_Actuator.Motor.voltage = 480; %V

WT_Params.Yaw_Actuator.Ideal.scaling_factor = 2;
WT_Params.Yaw_Actuator.max_yaw_rate = 0.5; %deg/sec
WT_Params.Yaw_Actuator.ideal_actuator_time_constant = 1e-1;
```

%%% YAW CONTROLLER PARAMETERS

```
WT_Params.Yaw_Controller.P_Gain = 1;
WT_Params.Yaw_Controller.I_Gain = 0.01;
WT_Params.Yaw_Controller.D_Gain = 0.001;% almost zero
WT_Params.Yaw_Controller.Yaw_Rate.P_Gain = 3000;
WT_Params.Yaw_Controller.Yaw_Rate.I_Gain = 10;
```

%%% NACELLE PARAMETERS

```
WT_Params.Nacelle.mass = 68000-WT_Params.Rotor.mass +
WT_Params.Geartrain.mass + WT_Params.Transformer.mass;
WT_Params.Nacelle.inertia = 100*[10 0 0;0 1 0;0 0 10];
WT_Params.Nacelle.length = 10;
WT_Params.Nacelle.CG_Offset.x = 5;
WT_Params.Nacelle.CG_Offset.y = 0;
WT_Params.Nacelle.CG_Offset.z = 0;
```

```
WT_Params.Nacelle.Yaw_Ctr_Offset.x = 7;  
WT_Params.Nacelle.Yaw_Ctr_Offset.y = 1;  
WT_Params.Nacelle.Yaw_Ctr_Offset.z = 0;  
WT_Params.Nacelle.damping = 1e6;
```

%%% MAIN CONTROLLER PARAMETERS

```
WT_Params.Main_Controller.wind_speed_cut_in_lower = 4;           %m/s  
WT_Params.Main_Controller.wind_speed_cut_out = 20;              %m/s,  
WT_Params.Main_Controller.wind_speed_cut_in_upper = 18;        %m/s,  
WT_Params.Main_Controller.wind_speed_average_period = 600;     %WIND SPEED  
AVERAGE PERIOD, SECONDS  
WT_Params.Main_Controller.turbine_speed_cut_in = 1200;         %RPM  
WT_Params.Main_Controller.turbine_speed_cut_out = 2200;        %RPM  
WT_Params.Main_Controller.turbine_speed_cut_out = 2200;        %RPM  
WT_Params.Main_Controller.turbine_speed_park = 5;              %RPM
```

%%% ENVIRONMENT PARAMETERS

```
WT_Params.Environment.Simple_Lift_And_Drag.AL_Dynamics = 1e-1;
```

Appendix C: Embedded System of Aerodynamic Load

```

function [...
    vid, ... induced velocity rate [m/s^2]
    flow, ... rotor state: 1:heli 2:young 3:windmill
    Y, ... blade force y components [N] (in disc plane)
    Z, ... blade force z components [N] (perpendicular disc plane)
    M, ... blade moment y component [Nm] (flapping)
    N ... blade moment z component [Nm] (shaft torque)
] ...
= Rotor( ...
... states
    vi, ... induced velocity [m/s]
    V, ... airspeed freestream [m/s] [0..inf[
    alpha, ... disc angle-of-attack [rad]
    beta, ... disc angle-of-sideslip [rad]
    p, ... disc rate-of-rotation [rad/s] (about principal axis)
    q, ...
    r, ...
    psi0, ... blade #0 azimuth [rad]
    omega, ... shaft angular rate [rad/s]
    sigma, ... blade flap angles [rad]
    sigmad, ... blade flap angle rates [rad/s]
... controls
    theta0, ... collective pitch command [rad]
    thetaLC, ... longitudinal cyclic pitch command [rad]
    thetaLS, ... lateral cyclic pitch command [rad]
... parameters
    WT_Params ... )

persistent psi I s J
persistent R c AR CLmax CLa CD0 k CDmax
persistent rho a
persistent ds

if isempty(I)

    % blades
    psi = zeros(1,length(sigma));
    I = length(psi); % number of blades (evenly spaced)

    % spanwise segmentation
    s = zeros(10,1); % CHANGE HERE
    J = length(s); % number of spanwise segments

    % rotor
    R = WT_Params.Blade.Length; % rotor radius [m]

```

```

c      = WT_Params. Blade. width;    % blade chord [m]
AR     = R^2/(c*R);                  % blade aspect ratio (approximation)
CLmax  = WT_Params. Rotor. MaxLiftCoefficient; % max(min) lift
      coefficient
CLa    = 2*pi / ( 1 + 2/AR );        % lift coefficient
      gradient
CDO    = WT_Params. Rotor. ParasiteDragCoefficient; % parasite drag coefficient
k      = WT_Params. Rotor. InducedDragFactor;    % induced drag factor
CDmax  = WT_Params. Rotor. MaxDragCoefficient; % max drag coefficient atmosphere
rho    = WT_Params. Air. Density;      % air density [kg/m^3]
a      = WT_Params. Air. SpeedOfSound;  % speed of sound [m/s]

% blade relative angles

for i = 1 : I, psi(i) = 2*pi *(i-1)/I; end

% spanwise segment length and position

ds = R /J;

for j = 1 : J, s(j) = ds/2 + (j-1) *ds; end

end

% airspeed components and angle-of-incidence

v = V *sin( beta );
u = (V^2-v^2)^.5 *cos( alpha );
w = (V^2-v^2)^.5 *sin( alpha );
if u==0 && v==0,
    if w >= 0, gamma = -pi /2;
    else      gamma = +pi /2;
    end
else
    gamma = atan( -w/(u*u+v*v)^.5 );
end

% Agenda

% 1 calc. Thrust at current Induced Velocity using Blade Element
Theory(BET)
% 2 calc. Thrust at current Induced Velocity using Linear Momentum
Theory(LMT)
% 3 calc. Induced Velocity rate of change

```

```

% Thrust at current Induced Velocity using Blade Element Theory (BET)

Y = zeros(length(psi), 1);
Z = zeros(length(psi), 1);
M = zeros(length(psi), 1);
N = zeros(length(psi), 1);

for i = 1 : I % parse all blades

    cos_psi = cos( psi0 + psi(i) );
    sin_psi = sin( psi0 + psi(i) );

    % blade pitch angle

    theta = theta0 + thetaLC *cos_psi + thetaLS *sin_psi;

    for j = 1 : J % parse all spanwise segments

        % blade spanwise total airspeed

        UT = u *sin_psi + v *cos_psi + s(j) *( omega + sigma(i) *( p
* cos_psi - q *sin_psi ) - r );
        UP = w - sigma(i) *( u *cos_psi - v *sin_psi ) - vi + s(j) *( -
sigmad(i) + p *sin_psi + q *cos_psi );
        U2 = ( UT^2 + UP^2 );
        U = U2^.5;

        % blade spanwise inflow angle, [-180..+180] deg

        phi = atan2( UP, UT );
        cos_phi = cos( phi );
        sin_phi = sin( phi );

        % blade spanwise apparent angle-of-attack, [0..180] deg

        AoA = mod( theta + phi , pi );

        % blade spanwise mach number (clamped) and wave drag factor

        mach = U /a;
        if 0.9 < mach
            mach = 0.9;
        end
        WD = 1 / ( 1 - mach^2 );

        % blade span wise angle-of-attack @ stall

        AoAstall = CLmax / ( WD *CLa );
        if AoAstall <= 0 || AoAstall > pi/2

```

```

        AoAstall = pi / 2;
end

% blade spanwise aerodynamic coefficients

if AoA <= AoAstall, % interval [0..AoAstall]
    CL = CLa * AoA;
    CD = WD * CD0 + k * CL^2;
    CL = WD * CL;
elseif AoA >= pi - AoAstall, % interval [pi - AoAstall..pi]
    CL = CLa * ( AoA - pi );
    CD = WD * CD0 + k * CL^2;
    CL = WD * CL;
else % interval [AoAstall..pi - AoAstall]
    delta = ( AoA - pi / 2 ) / ( AoAstall - pi / 2 );
    CL = CLmax * delta;
    CD = CDmax + ( WD * CD0 + k * ( CLmax / WD )^2 - CDmax )
*delta^2;
end

% blade span wise dynamic pressure, lift and drag forces

Q = .5 * rho * U2 * c * ds;
dLift = Q * CL;
dDrag = Q * CD;

% blade segment forces and moments

dY = - dLift * sin_phi + dDrag * cos_phi;
dZ = - dLift * cos_phi - dDrag * sin_phi;
dM = - s(j) * dZ;
dN = s(j) * dY;

% total blade force and moment

Y(i) = Y(i) + dY;
Z(i) = Z(i) + dZ;
M(i) = M(i) + dM;
N(i) = N(i) + dN;

end

end

TBET = - sum( Z );

% Thrust at current Induced Velocity using Linear Momentum Theory (LMT)

if gamma >= 0, % climb and hover

```

```

% flow is reversed when vi < -V/2

% when flow is not reversed, use derived formula

if -V/2 <= vi
    TLMT = 2*rho*pi *R^2*vi *( (V*cos(gamma))^2+(V*sin(gamma)+vi )^2
)^.5;
    flow = 1;
else
    % when flow is reversed, use approximation by Young

    A = ( (5-4*sin(gamma))^5 - 5^5 ) / ( 1-5^5 );
    if -5/3*V < vi
        TLMT = 2*rho*pi *R^2*( -A*((vi -3*V)/7)^2 + (1-
A)*vi *(V^2+vi ^2)^.5 );
        flow = 2;
    else
        TLMT = 2*rho*pi *R^2*( -A*(vi +V)^2 + (1-
A)*vi *(V^2+vi ^2)^.5 );
        flow = 3;
    end
end

else % descent, (gamma < 0)

% flow is reversed when vi > V/2

% when flow is not reversed, use derived formula

if V/2 >= vi
    TLMT = 2*rho*pi *R^2*vi *( (V*cos(gamma))^2+(-V*sin(gamma)-vi )^2
)^.5;
    flow = 1;
else
    % when flow is reversed, use approximation by Young

    A = ( (5+4*sin(gamma))^5 - 5^5 ) / ( 1-5^5 );
    if 5/3*V > vi
        TLMT = 2*rho*pi *R^2*( A*((vi +3*V)/7)^2 + (1-
A)*vi *(V^2+vi ^2)^.5 );
        flow = 2;
    else
        TLMT = 2*rho*pi *R^2*( A*( vi -V )^2 + (1-
A)*vi *(V^2+vi ^2)^.5 );
        flow = 3;
    end
end
end

```

```
% Induced Velocity rate of change

h = R;
vid = ( TBET - TLMT ) /rho /pi /R^2 /h;

% to-do
%eml .extrinsic('myread');

% -- eof
```

Appendix D: The NACA 4 Digit Airfoil

```

function af = naca4gen(iaf)
%
% "naca4gen" Generates the NACA 4 digit airfoil coordinates with desired
% no.
% of panels (line elements) on it.
%
% INPUTS-----
-
%       iaf.designation = NACA 4 digit iaf.designation (eg. 'xxxx') -
STRING !
%       iaf.n = no of panels (line elements) PER SIDE
(upper/lower)
% iaf.HalfCosineSpacing = 1 for "half cosine x-spacing"
%                       = 0 to give "uniform x-spacing"
%       iaf.wantFile = 1 for creating airfoil data file (eg.
' naca2412. dat' )
%                       = 0 to suppress writing into a file
%       iaf.datFilePath = Path where the data file has to be created
%                       (eg. 'af_data_folder/naca4digitAF/')
%                       use only forward slash '/' (Just for OS
portability)
%
% OUTPUTS-----
-
%
Data::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\

%       af.x = x coordinate (nx1 array)
%       af.z = z coordinate (nx1 array)
%       af.xU = x coordinate of upper surface (nx1 array)
%       af.zU = z coordinate of upper surface (nx1 array)
%       af.xL = x coordinate of lower surface (nx1 array)
%       af.zL = z coordinate of lower surface (nx1 array)
%       af.xC = x coordinate of camber line (nx1 array)
%       af.zC = z coordinate of camber line (nx1 array)
%       af.name = Name of the airfoil

% iaf.datFilePath='./'; % Current folder
% iaf.is_finiteTE=0;

% [[Calculating key parameters-----]]

t=str2num(iaf.designation(3:4))/100;
m=str2num(iaf.designation(1))/100;
p=str2num(iaf.designation(2))/10;

```

```

a0= 0.2969;
a1=-0.1260;
a2=-0.3516;
a3= 0.2843;

if iaf.is_finiteTE ==1
    a4=-0.1015; % For finite thick TE
else
    a4=-0.1036; % For zero thick TE
end

% % [[Giving x-spacing-----
]]

if iaf.HalfCosineSpacing==1
    beta=linspace(0,pi, iaf.n+1)';
    x=(0.5*(1-cos(beta))); % Half cosine based spacing
    iaf.header=['NACA' iaf.designation ' : [' num2str(2*iaf.n)
' panels, Half cosine x-spacing']'];
else

    x=linspace(0,1, iaf.n+1)';
    iaf.header=['NACA' iaf.designation ' : [' num2str(2*iaf.n)
' panels, Uniform x-spacing']'];
end

yt=(t/0.2)*(a0*sqrt(x)+a1*x+a2*x.^2+a3*x.^3+a4*x.^4);

xc1=x(find(x<=p));
xc2=x(find(x>p));
xc=[xc1 ; xc2];

if p==0
    xu=x;
    yu=yt;

    xl=x;
    yl=-yt;

    zc=zeros(size(xc));
else
    yc1=(m/p^2)*(2*p*xc1-xc1.^2);
    yc2=(m/(1-p)^2)*((1-2*p)+2*p*xc2-xc2.^2);
    zc=[yc1 ; yc2];

    dyc1_dx=(m/p^2)*(2*p-2*xc1);

```

```

dyc2_dx=(m/(1-p)^2)*(2*p-2*xc2);
dyc_dx=[dyc1_dx ; dyc2_dx];
theta=atan(dyc_dx);

xu=x-yt.*sin(theta);
yu=zc+yt.*cos(theta);

xl=x+yt.*sin(theta);
yl=zc-yt.*cos(theta);

end

af.name=['NACA ' iaf.designation];

indx1=1:min(find(af.x==min(af.x)) ); % Upper surface indices
indx2=min(find(af.x==min(af.x)) ):length(af.x); % Lower surface indices
af.xU=af.x(indx1); % Upper Surface x
af.zU=af.z(indx1); % Upper Surface z
af.xL=af.x(indx2); % Lower Surface x
af.zL=af.z(indx2); % Lower Surface z

af.xC=xc;
af.zC=zc;

leci rFactor=0.8;
af.rLE=0.5*(a0*t/0.2)^2;

le_offs=0.5/100;
dyc_dx_le=(m/p^2)*( 2*p-2*le_offs );
theta_le=atan(dyc_dx_le);
af.xLEcenter=af.rLE*cos(theta_le);
af.yLEcenter=af.rLE*sin(theta_le);

%% [[Writing iaf data into dat file-----
]]

if iaf.wantFile==1
    F1=iaf.header;
    F2=num2str([af.x af.z]);
    F=strvcat(F1,F2);
    fileName=[iaf.datFilePath 'naca' iaf.designation '.dat'];
    dlmwri te(fileName,F,'deli mi ter','')
end

iaf.designation='8412';

```

```
iaf.n=50;% number of panels
iaf.HalfCosineSpacing=1;
iaf.wantFile=1;
iaf.datFilePath='./'; % Current folder
iaf.is_finiteTE=0;

af = naca4gen(iaf); % plot(af.x, af.z, 'bo-')

plot(af.xU, af.zU, 'bo-')

    hold on

plot(af.xL, af.zL, 'ro-')

    axis equal
```

Appendix E: Integrating Airfoil of Blade with the Model

```
function importfile(fileToRead1)

%IMPORTFILE(FILETOREAD1)
% Imports data from the specified file
% FILETOREAD1: file to read

% Auto-generated by MATLAB on 23-Aug-2011 18:29:07

% Import the file

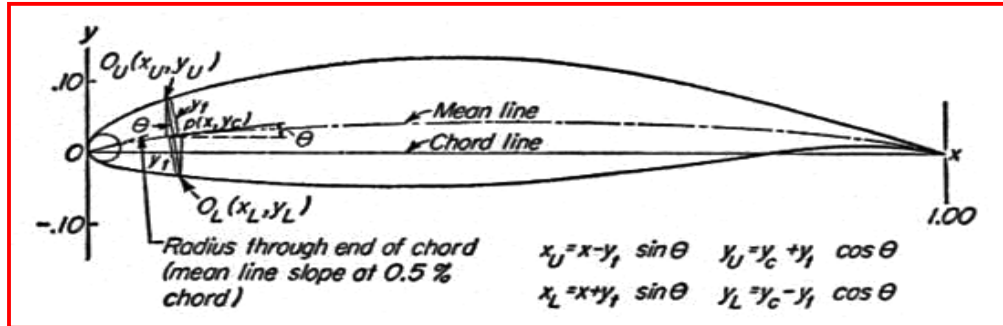
newData1 = importdata(fileToRead1);

% Create new variables in the base workspace from those fields.

vars = fieldnames(newData1);

for
    i = 1:length(vars)
        assignin('base', vars{i}, newData1.(vars{i}));
end
```

Appendix F: NACA 8412 Airfoil



	x	y(t)	y(c)	x(U)	y(U)	x(L)	y(L)
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00214	0.00808	0.00085	-0.00085	0.00836	0.00513	-0.00665
3	0.00856	0.01581	0.00339	0.00275	0.01810	0.01436	-0.01133
4	0.01921	0.02316	0.00750	0.01087	0.02911	0.02756	-0.01411
5	0.03407	0.03007	0.01305	0.02356	0.04122	0.04459	-0.01512
6	0.05307	0.03646	0.01982	0.04083	0.05416	0.06531	-0.01452
7	0.07612	0.04224	0.02755	0.06269	0.06760	0.08955	-0.01250
8	0.10313	0.04735	0.03593	0.08909	0.08115	0.11716	-0.00929
9	0.13397	0.05168	0.04462	0.11997	0.09436	0.14798	-0.00513
10	0.16853	0.05517	0.05321	0.15521	0.10675	0.18185	-0.00032
11	0.20665	0.05774	0.06131	0.19465	0.11779	0.21864	0.00482
12	0.24816	0.05937	0.06847	0.23806	0.12697	0.25826	0.00997
13	0.29289	0.06001	0.07426	0.28519	0.13377	0.30060	0.01475
14	0.34065	0.05966	0.07824	0.33571	0.13769	0.34560	0.01878
15	0.39124	0.05834	0.07996	0.38925	0.13827	0.39322	0.02165
16	0.44443	0.05609	0.07956	0.44485	0.13565	0.44401	0.02347
17	0.50000	0.05294	0.07778	0.50170	0.13069	0.49830	0.02486
18	0.55771	0.04895	0.07447	0.56051	0.12335	0.55491	0.02560
19	0.61732	0.04419	0.06951	0.62099	0.11354	0.61365	0.02547
20	0.67856	0.03869	0.06276	0.68280	0.10122	0.67432	0.02430
21	0.74118	0.03252	0.05413	0.74562	0.08634	0.73674	0.02192
22	0.80491	0.02569	0.04357	0.80911	0.06891	0.80071	0.01823
23	0.86947	0.01821	0.03102	0.87295	0.04890	0.86600	0.01314
24	0.93460	0.01008	0.01649	0.93679	0.02633	0.93240	0.00665
25	1.00000	0.00126	0.00000	1.00031	0.00122	0.99969	-0.00122

Appendix G: Lift and Drag Coefficient

Table G1: Lift Coefficient

	----- REYNOLDS NUMBER -----							
α	80000	160000	360000	700000	1000000	2000000	5000000	10000000
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100	0.1100
2	0.2200	0.2200	0.2200	0.2200	0.2200	0.2200	0.2200	0.2200
3	0.3300	0.3300	0.3300	0.3300	0.3300	0.3300	0.3300	0.3300
4	0.4186	0.4400	0.4400	0.4400	0.4400	0.4400	0.4400	0.4400
5	0.5180	0.5500	0.5500	0.5500	0.5500	0.5500	0.5500	0.5500
6	0.6048	0.6299	0.6600	0.6600	0.6600	0.6600	0.6600	0.6600
7	0.6760	0.7150	0.7390	0.7483	0.7700	0.7700	0.7700	0.7700
8	0.7189	0.7851	0.8240	0.8442	0.8504	0.8800	0.8800	0.8800
9	0.6969	0.8311	0.8946	0.9260	0.9387	0.9574	0.9900	0.9900
10	0.6122	0.8322	0.9440	0.9937	1.0141	1.0433	1.0685	1.1000
11	0.1642	0.7623	0.9572	1.0363	1.0686	1.1138	1.1553	1.1749
12	0.0749	0.5936	0.9285	1.0508	1.0971	1.1667	1.2290	1.2591
13	0.0967	0.3548	0.8562	1.0302	1.0957	1.1948	1.2847	1.3300
14	0.1382	0.2371	0.7483	0.9801	1.0656	1.1962	1.3187	1.3825
15	0.1861	0.2376	0.6350	0.9119	1.0145	1.1744	1.3298	1.4136
16	0.2364	0.2665	0.5384	0.8401	0.9567	1.1356	1.3186	1.4233
17	0.2873	0.3098	0.4851	0.7799	0.8996	1.0921	1.2917	1.4136

18	0.3393	0.3567	0.4782	0.7305	0.8566	1.0510	1.2576	1.3897
19	0.3927	0.4066	0.4908	0.7041	0.8226	1.0173	1.2242	1.3608
20	0.4463	0.4575	0.5247	0.6990	0.8089	0.9954	1.1965	1.3325
21	0.5001	0.5087	0.5616	0.7097	0.8063	0.9837	1.1771	1.3077
22	0.5539	0.5611	0.6045	0.7298	0.8189	0.9827	1.1647	1.2767
23	0.6078	0.6148	0.6528	0.7593	0.8408	0.9910	1.1611	1.1981
24	0.6617	0.6685	0.7015	0.7961	0.8668	1.0078	1.1563	1.1538
25	0.7156	0.7224	0.7511	0.8353	0.9023	1.0317	1.1322	1.1380
26	0.7700	0.7771	0.8055	0.8838	0.9406	1.0591	1.1268	1.1374
27	0.8277	0.8382	0.8788	0.9473	0.9912	1.0810	1.1397	1.1519
30	0.8550	0.8550	0.9300	0.9600	1.0100	1.0700	1.1000	1.1200
35	0.9800	0.9800	0.9820	0.9800	1.0200	1.0620	1.0800	1.1000
40	1.0350	1.0350	1.0350	1.0350	1.0350	1.0550	1.0600	1.0700
45	1.0500	1.0500	1.0500	1.0500	1.0500	1.0500	1.0500	1.0500
50	1.0200	1.0200	1.0200	1.0200	1.0200	1.0200	1.0200	1.0200
55	0.9550	0.9550	0.9550	0.9550	0.9550	0.9550	0.9550	0.9550
60	0.8750	0.8750	0.8750	0.8750	0.8750	0.8750	0.8750	0.8750
65	0.7600	0.7600	0.7600	0.7600	0.7600	0.7600	0.7600	0.7600
70	0.6300	0.6300	0.6300	0.6300	0.6300	0.6300	0.6300	0.6300
75	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
80	0.3650	0.3650	0.3650	0.3650	0.3650	0.3650	0.3650	0.3650
85	0.2300	0.2300	0.2300	0.2300	0.2300	0.2300	0.2300	0.2300
90	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900	0.0900
95	-0.0500	-0.0500	-0.0500	-0.0500	-0.0500	-0.0500	-0.0500	-0.0500

100 -0.1850 -0.1850 -0.1850 -0.1850 -0.1850 -0.1850 -0.1850 -0.1850
105 -0.3200 -0.3200 -0.3200 -0.3200 -0.3200 -0.3200 -0.3200 -0.3200
110 -0.4500 -0.4500 -0.4500 -0.4500 -0.4500 -0.4500 -0.4500 -0.4500
115 -0.5750 -0.5750 -0.5750 -0.5750 -0.5750 -0.5750 -0.5750 -0.5750
120 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700
125 -0.7600 -0.7600 -0.7600 -0.7600 -0.7600 -0.7600 -0.7600 -0.7600
130 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500
135 -0.9300 -0.9300 -0.9300 -0.9300 -0.9300 -0.9300 -0.9300 -0.9300
140 -0.9800 -0.9800 -0.9800 -0.9800 -0.9800 -0.9800 -0.9800 -0.9800
145 -0.9000 -0.9000 -0.9000 -0.9000 -0.9000 -0.9000 -0.9000 -0.9000
150 -0.7700 -0.7700 -0.7700 -0.7700 -0.7700 -0.7700 -0.7700 -0.7700
155 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700 -0.6700
160 -0.6350 -0.6350 -0.6350 -0.6350 -0.6350 -0.6350 -0.6350 -0.6350
165 -0.6800 -0.6800 -0.6800 -0.6800 -0.6800 -0.6800 -0.6800 -0.6800
170 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500 -0.8500
175 -0.6600 -0.6600 -0.6600 -0.6600 -0.6600 -0.6600 -0.6600 -0.6600
180 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

Table G2: Drag Coefficient

----- REYNOLDS NUMBER -----								
α	8×10^3	16×10^4	36×10^4	70×10^4	100×10^4	200×10^4	500×10^4	100×10^5

0	0.0147	0.0116	0.0091	0.0077	0.0074	0.0070	0.0068	0.0068
1	0.0148	0.0117	0.0092	0.0078	0.0075	0.0071	0.0069	0.0068
2	0.0151	0.0120	0.0094	0.0080	0.0076	0.0072	0.0070	0.0069
3	0.0156	0.0124	0.0098	0.0083	0.0079	0.0075	0.0073	0.0071
4	0.0168	0.0132	0.0105	0.0089	0.0083	0.0078	0.0075	0.0074
5	0.0181	0.0142	0.0114	0.0098	0.0091	0.0083	0.0080	0.0077
6	0.0197	0.0160	0.0126	0.0108	0.0101	0.0090	0.0084	0.0081
7	0.0214	0.0176	0.0143	0.0122	0.0111	0.0098	0.0089	0.0086
8	0.0234	0.0193	0.0157	0.0135	0.0126	0.0108	0.0095	0.0090
9	0.0255	0.0212	0.0173	0.0149	0.0138	0.0121	0.0102	0.0096
10	0.0277	0.0233	0.0191	0.0164	0.0152	0.0133	0.0113	0.0103
11	0.0760	0.0256	0.0211	0.0182	0.0168	0.0146	0.0124	0.0114
12	0.1230	0.0281	0.0233	0.0200	0.0186	0.0161	0.0136	0.0123
13	0.1400	0.0302	0.0257	0.0221	0.0205	0.0177	0.0149	0.0134
14	0.1580	0.1040	0.0283	0.0244	0.0225	0.0195	0.0164	0.0147
15	0.1770	0.1770	0.0312	0.0269	0.0249	0.0215	0.0180	0.0161
16	0.1960	0.1970	0.1240	0.0297	0.0275	0.0237	0.0198	0.0176
17	0.2170	0.2170	0.2170	0.1340	0.0303	0.0261	0.0218	0.0194
18	0.2380	0.2380	0.2380	0.2380	0.1450	0.0288	0.0240	0.0213
19	0.2600	0.2600	0.2600	0.2600	0.2600	0.1550	0.0265	0.0234
20	0.2820	0.2820	0.2820	0.2820	0.2820	0.2820	0.1660	0.0257
21	0.3050	0.3050	0.3050	0.3050	0.3050	0.3050	0.3050	0.1770

22	0.3290	0.3290	0.3290	0.3290	0.3290	0.3290	0.3290	0.3290
23	0.3540	0.3540	0.3540	0.3540	0.3540	0.3540	0.3540	0.3540
24	0.3790	0.3790	0.3790	0.3790	0.3790	0.3790	0.3790	0.3790
25	0.4050	0.4050	0.4050	0.4050	0.4050	0.4050	0.4050	0.4050
26	0.4320	0.4320	0.4320	0.4320	0.4320	0.4320	0.4320	0.4320
27	0.4600	0.4600	0.4600	0.4600	0.4600	0.4600	0.4600	0.4600
30	0.5700	0.5700	0.5700	0.5700	0.5700	0.5700	0.5700	0.5700
35	0.7450	0.7450	0.7450	0.7450	0.7450	0.7450	0.7450	0.7450
40	0.9200	0.9200	0.9200	0.9200	0.9200	0.9200	0.9200	0.9200
45	1.0750	1.0750	1.0750	1.0750	1.0750	1.0750	1.0750	1.0750
50	1.2150	1.2150	1.2150	1.2150	1.2150	1.2150	1.2150	1.2150
55	1.3450	1.3450	1.3450	1.3450	1.3450	1.3450	1.3450	1.3450
60	1.4700	1.4700	1.4700	1.4700	1.4700	1.4700	1.4700	1.4700
65	1.5750	1.5750	1.5750	1.5750	1.5750	1.5750	1.5750	1.5750
70	1.6650	1.6650	1.6650	1.6650	1.6650	1.6650	1.6650	1.6650
75	1.7350	1.7350	1.7350	1.7350	1.7350	1.7350	1.7350	1.7350
80	1.7800	1.7800	1.7800	1.7800	1.7800	1.7800	1.7800	1.7800
85	1.8000	1.8000	1.8000	1.8000	1.8000	1.8000	1.8000	1.8000
90	1.8000	1.8000	1.8000	1.8000	1.8000	1.8000	1.8000	1.8000
95	1.7800	1.7800	1.7800	1.7800	1.7800	1.7800	1.7800	1.7800
100	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500	1.7500
105	1.7000	1.7000	1.7000	1.7000	1.7000	1.7000	1.7000	1.7000
110	1.6350	1.6350	1.6350	1.6350	1.6350	1.6350	1.6350	1.6350
115	1.5550	1.5550	1.5550	1.5550	1.5550	1.5550	1.5550	1.5550
120	1.4650	1.4650	1.4650	1.4650	1.4650	1.4650	1.4650	1.4650
125	1.3500	1.3500	1.3500	1.3500	1.3500	1.3500	1.3500	1.3500

130	1.2250	1.2250	1.2250	1.2250	1.2250	1.2250	1.2250	1.2250
135	1.0850	1.0850	1.0850	1.0850	1.0850	1.0850	1.0850	1.0850
140	0.9250	0.9250	0.9250	0.9250	0.9250	0.9250	0.9250	0.9250
145	0.7550	0.7550	0.7550	0.7550	0.7550	0.7550	0.7550	0.7550
150	0.5750	0.5750	0.5750	0.5750	0.5750	0.5750	0.5750	0.5750
155	0.4200	0.4200	0.4200	0.4200	0.4200	0.4200	0.4200	0.4200
160	0.3200	0.3200	0.3200	0.3200	0.3200	0.3200	0.3200	0.3200
165	0.2300	0.2300	0.2300	0.2300	0.2300	0.2300	0.2300	0.2300
170	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400	0.1400
175	0.0550	0.0550	0.0550	0.0550	0.0550	0.0550	0.0550	0.0550
180	0.0250	0.0250	0.0250	0.0250	0.0250	0.0250	0.0250	0.0250

Appendix H: Automatic Report Generation

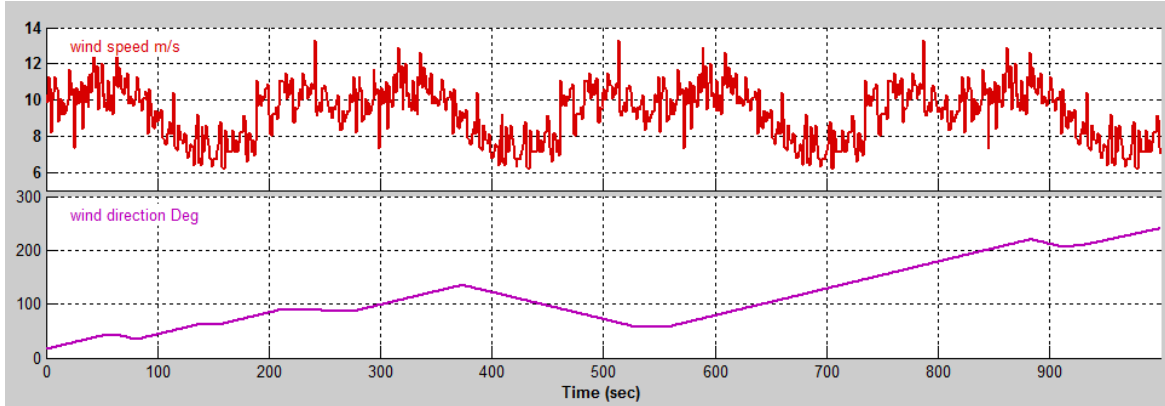


Figure H1: Wind Data

For the wind data shown above the Automatic report generated is shown below.

Wind Turbine Simulation Test Report

This report is generated from the currently running data

Table of Contents

1.	Model Overview Error! Bookmark not defined.	
	Wind Turbine Model, Top Level.....	130
	Wind Turbine Model, Hydraulic Pitch System	130
	Wind Turbine Model, Yaw System	131
	Wind Turbine Model, Yaw Actuator	132
	Wind Turbine Model, Geartrain	133
	Wind Turbine Model, Generator and Grid.....	133
	Wind Turbine Model, Supervisory Control (basic)	134
2.	Simulation: Wind Test Start Stop	135

List of Figures

2-1.	Geartrain Input Torque	136
2-2.	Rotor Speed (RPM)	136
2-3.	Wind	137
2-4.	Pitch Command and Angle (deg)	138
2-5.	Nacelle Yaw	138

1. Model Overview

Wind Turbine Model, Top Level

```
Select_Turbine_Systems('H_Pitch S_Yaw Ge Gn Lift_Drag', WT_Configs)
open_system('Wind_Turbine');
```

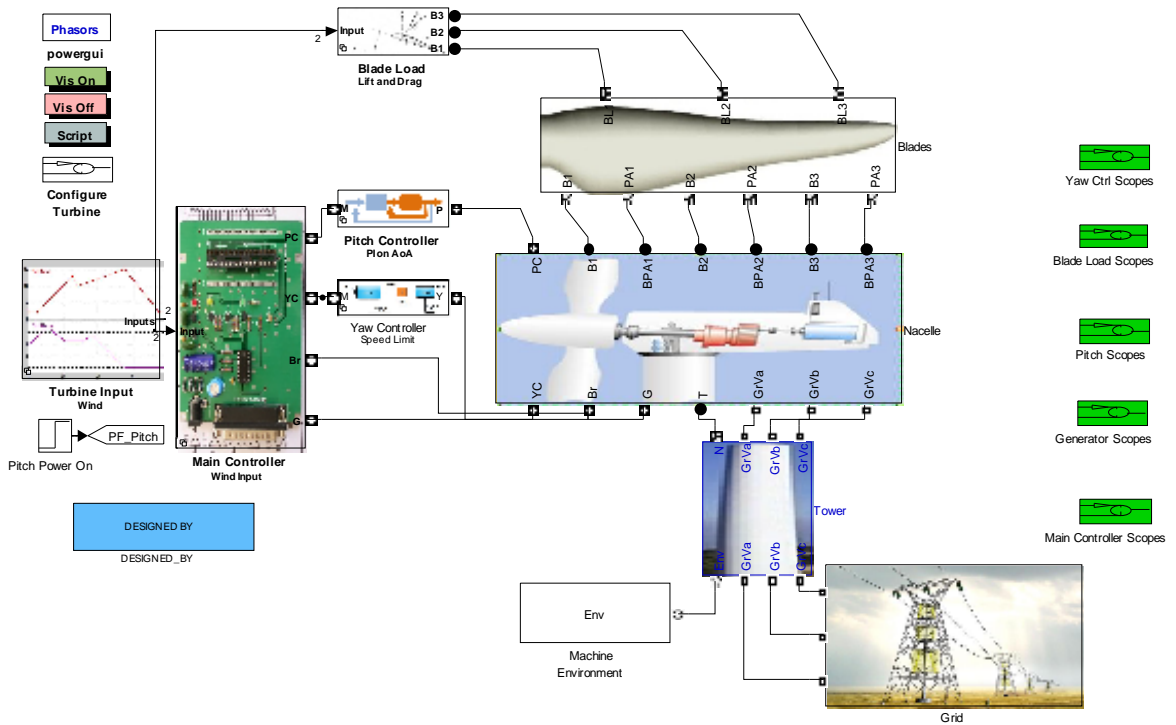


Figure H2: Wind Turbine Model, Top Level

Wind Turbine Model, Hydraulic Pitch System

```
open_system('Wind_Turbine/Nacelle/Pitch
System/Hydraulic/Hydraulic/Actuator_1');
```

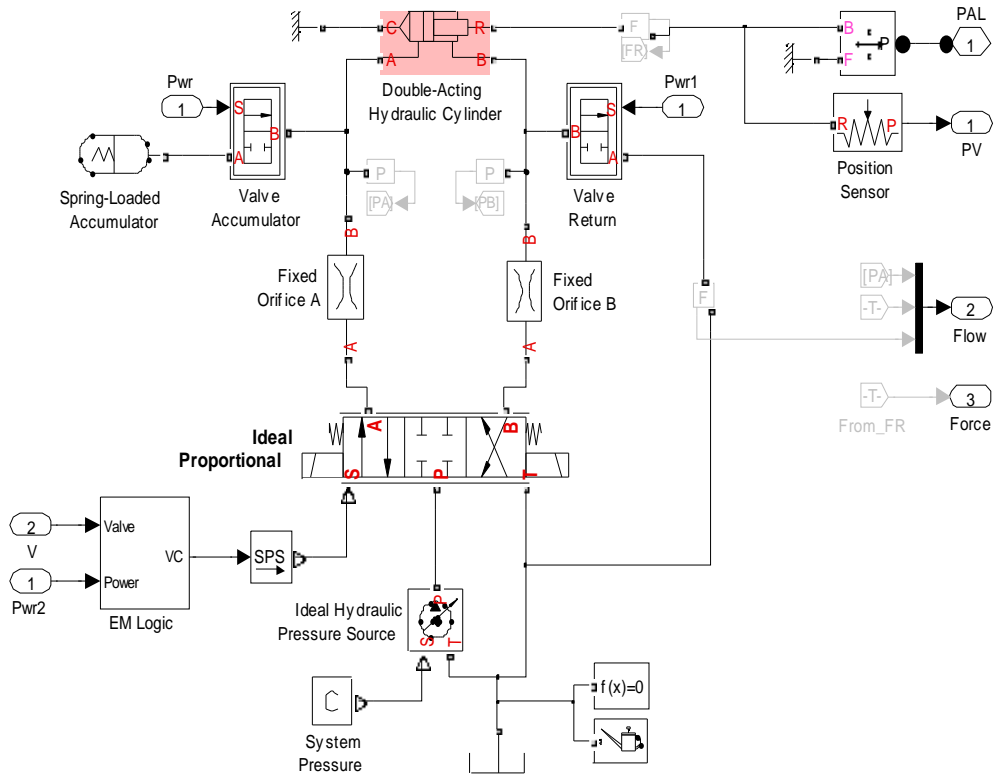


Figure H3: Wind Turbine Model, Hydraulic Pitch System

```
open_system('Wind_Turbine/Nacelle/Yaw System/Servomotor');
```

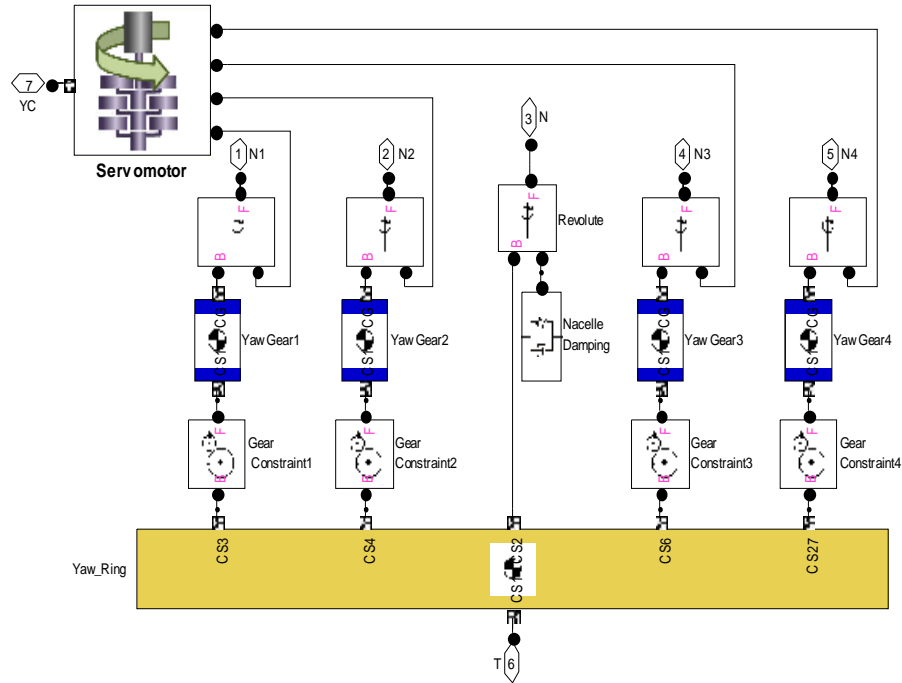


Figure H4: Subsystem of yaw system

Wind Turbine Model, Yaw Actuator

```
open_system('Wind_Turbine/Nacelle/Yaw System/Servomotor/Servomotor/Yaw Motor 1/Servomotor');
```

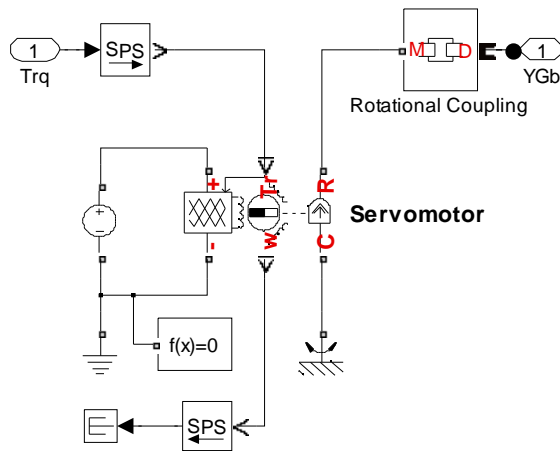


Figure H5: subsystem of servomotor

Wind Turbine Model, Geartrain

```
open_system('Wind_Turbine/Nacelle/Geartrain/Ideal');
```

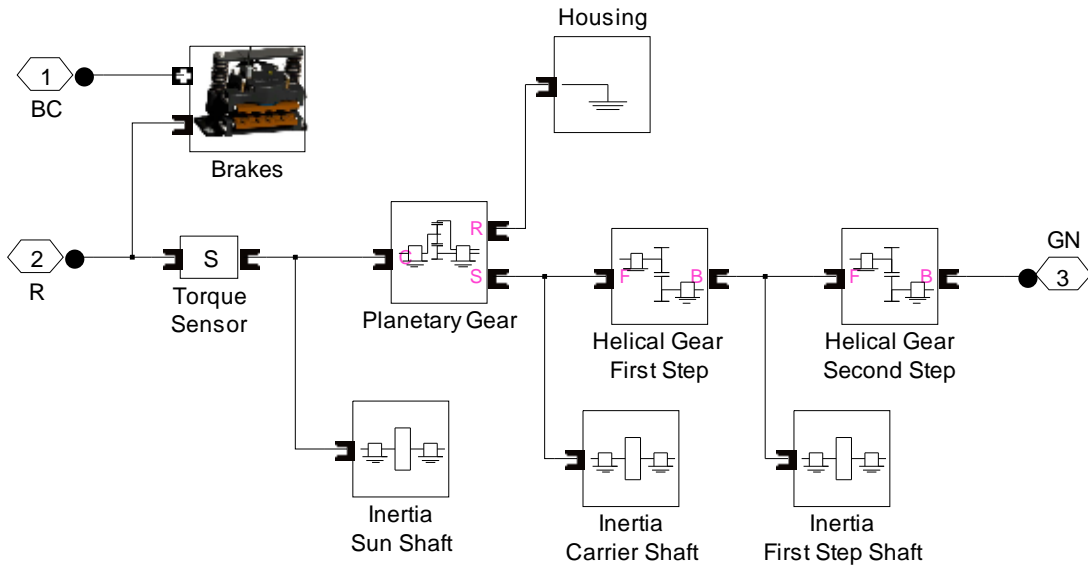


Figure H6: Wind Turbine Model, Gear train

Wind Turbine Model, Generator and Grid

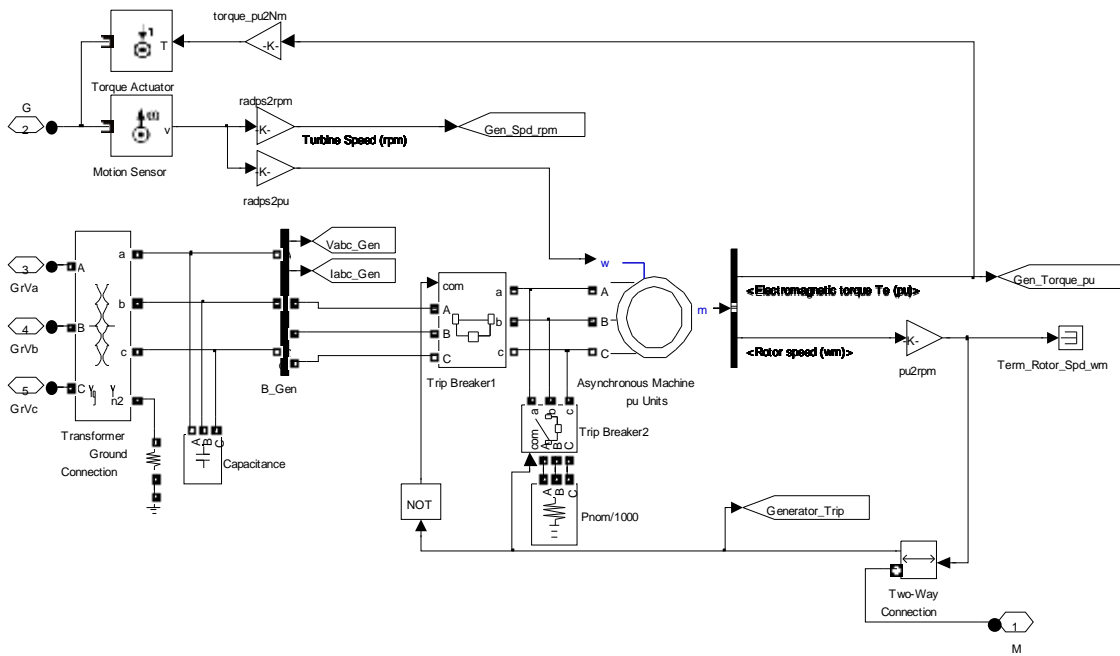


Figure H7: Wind Turbine Model, Generator

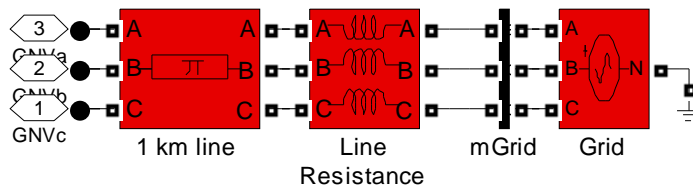


Figure H8: Wind Turbine Model, Grid

Wind Turbine Model, Supervisory Control (basic)

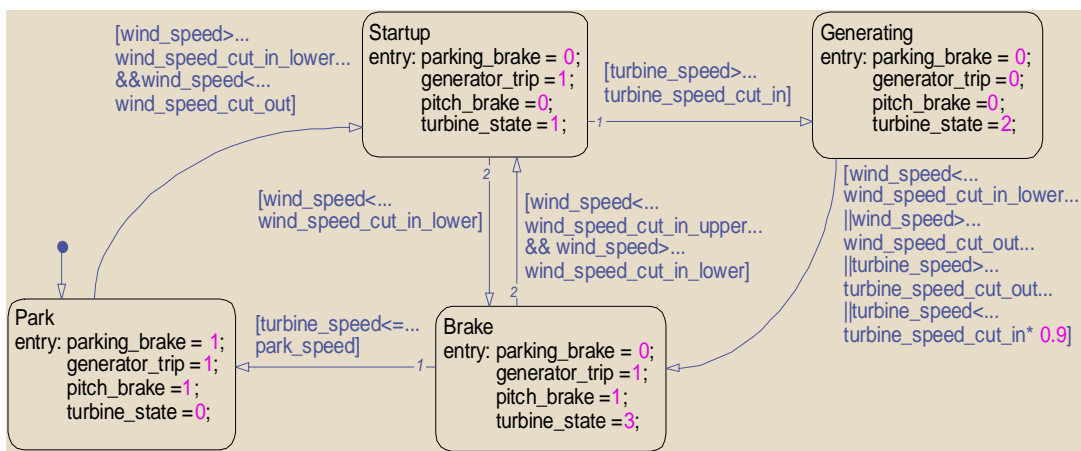


Figure H9: Wind Turbine Model, Supervisory Control (basic)

2. Simulation: Wind Test Start/Stop

```

% CONFIGURE TURBINE
WT_Configuration = 'I_Pitch IA_Yaw Ge Lift_Drag';
Select_Turbine_Systems(WT_Configuration, WT_Configs);

```

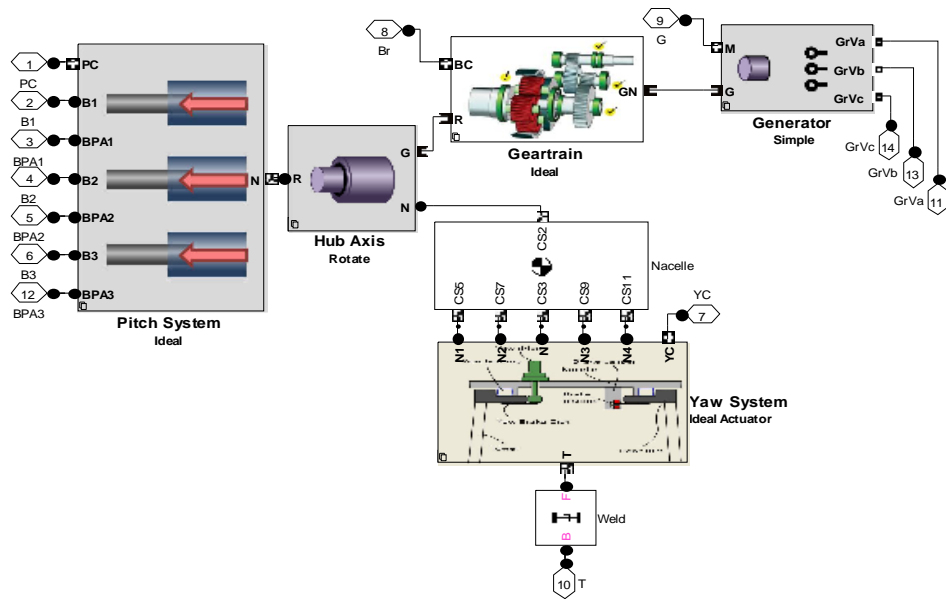


Figure H10: Wind Turbine Model, Nacelle

```

% SAVE ELAPSED TIME
WT_Results_RPT{end+1,1} = WT_Configuration;
WT_Results_RPT{end,2} = Elapsed_Sim_Time;
Simulation_Time = get_param(bdroot, 'StopTime');
WT_Results_RPT{end,3} = 80;
WT_Results_RPT{end,4} = str2num(Simulation_Time)/Elapsed_Sim_Time;

```

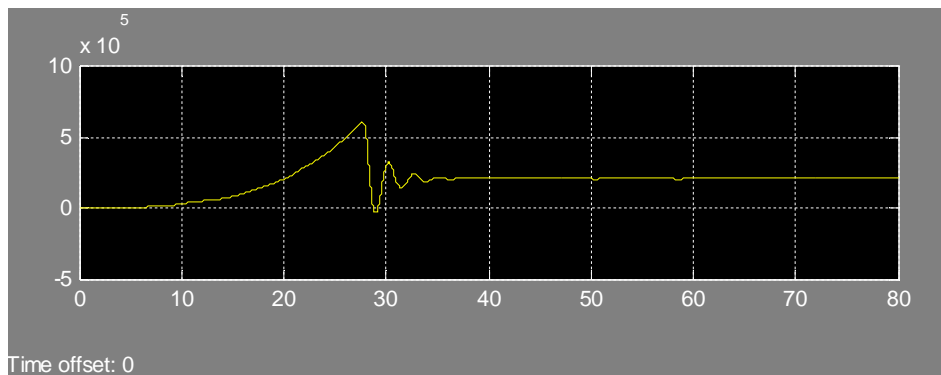


Figure H11: Gear train Input Torque

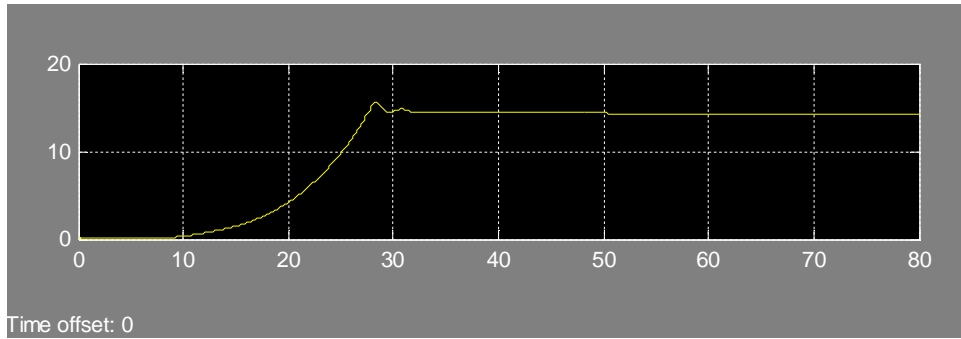


Figure H12: Rotor Speed (RPM)

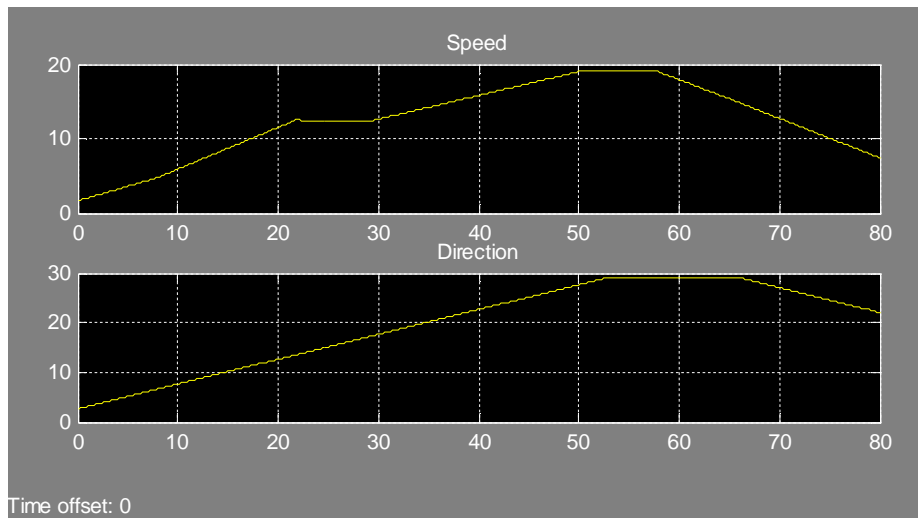


Figure H13: Wind speed and direction

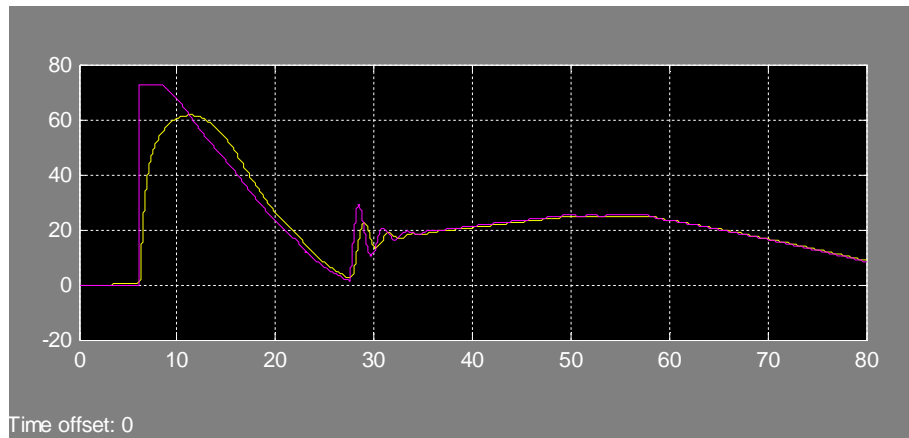


Figure H14: Pitch Command and Angle (deg)

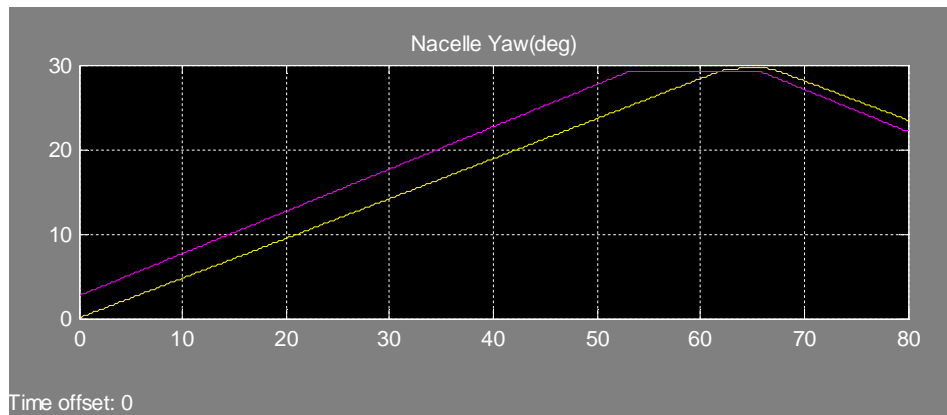


Figure H15:Nacelle Yaw

Designed by Tewodros Walle

Advised by.-Ing. Ababayehu Assefa

Published 27-Aug-2011 09:41:27

END OF THE AUTOGENERATED REPORT

