In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature ___________________________________

Date _______________________________________

Allowing Type-3 Wind Turbines to Participate in Frequency Regulation using a Genetic Algorithm for Controller Parameter Tuning

By

Shat C. Pratoomratana

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Measurement and Control Engineering

Idaho State University

Spring 2019

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of Shat C. Pratoomratana find it satisfactory and recommend that it be accepted.

Dr. Marco P. Schoen,
Major Advisor

Dr. Kenneth Bosworth,
Committee Member

Dr. Gene Stuffle,
Graduate Faculty Representative

Dedication

For:

Lance

Acknowledgments

I would first like to thank my thesis advisor Dr. Marco P. Schoen, who was able to instill a love of control systems in me, and allowed me the freedom and latitude to choose a research project that was of interest to me. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to thank Jake Gentle of the Idaho National Laboratory, who helped to direct the focus of my work by informing me on the latest trends in the industry that are important to researchers.

Finally, I would like to thank all my fellow graduate students at Idaho State University, who I had the pleasure of getting to know while being able to share the ups and downs of completing a graduate degree.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MW	Mega Watt
HAWT	Horizontal Axis Wind Turbine
VAWT	Vertical Axis Wind Turbine
DFIG	Doubly-Fed Induction Generator
KE	Kinetic Energy
WRIG	Wound Rotor Induction Generator
RSC	Rotor Side Converter
GSC	Grid Side Converter
AGC	Automatic Generation Control
RoCoF	Rate of Change of Frequency
PI	Proportional – Integral
PID	Proportional –Integral-Derivative
Hz	Hertz
KRK	Klein-Rogers-Kunder
PSS	Power System Stabilizer
ISE	Integral Squared Error
p.u.	Per-Unit

List of Symbols

P	Power
m	Mass
v	Wind Velocity (m/s)
$\dot{m}$	Mass Flow Rate
ρ	Air Density
A	Area Swept By Rotor Blades
C_p	Power Coefficient
λ	Tip Speed Ratio
ω_r	Rotor Angular Speed (rad/sec)
R	Blade Length
β	Blade Pitch (degrees)
P_{ord}	Electrical Power Output
P_{setlf}	Electrical Power Set-Point
P_{FC}	Supplemental Power Order
©	Copyright

Allowing Type-3 Wind Turbines to Participate in Frequency Regulation using a Genetic Algorithm for Controller Parameter Tuning

Thesis Abstract--Idaho State University (2019)

With wind penetration levels increasing in power systems across the world, different challenges are encountered with respect to the controllability and operation of a power system. The frequency regulation of a power grid is highly impacted when a considerable amount of wind energy is connected to the system. This thesis uses a dynamic, non-linearized system developed in Matlab/Simulink©, to study how Type-3 wind turbines impact the stability and frequency response of a test power system. In this work, a proposed frequency sensitive pitch angle controller is implemented and tuned through the use of a Genetic Algorithm. Time simulations are used to demonstrate the transient and steady-state performance of the proposed controllers in the system with 25% wind penetration. The results show that the addition of the tuned frequency sensitive controllers improved the settling frequency, rate of change of frequency, and frequency nadir compared to the wind turbines without these controllers.

Key Words: Wind Turbine, DFIG, Frequency Regulation, Pitch Angle Control, Genetic Algorithm, Parameter Tuning

CHAPTER 1: INTRODUCTION

1.1 WIND ENERGY GROWTH

At the end of 1980 on Crotched Mountain in Southwest New Hampshire, the first wind farm started producing electricity, thus, signaling the beginning of the modern wind turbine industry. Since then, the installed capacity of wind generation has been steadily increasing, with 52,429 Mega-Watts (MW) installed in 2017 bringing the global installed capacity of wind energy up to 539,123 MW [1]. Particularly in the United States, the wind power capacity is experiencing strong growth. The United States increased its total wind power capacity to 88,973 MW with the addition of 7,017 MW of capability added in 2017 [2]. Wind power amounted to 25% of all capacity additions in 2017. Over the last decade, wind accounted for 30% of all U.S. capacity additions [2]. Fig. 1 shows the capacity additions for different energy sectors, in different regions of the United States.

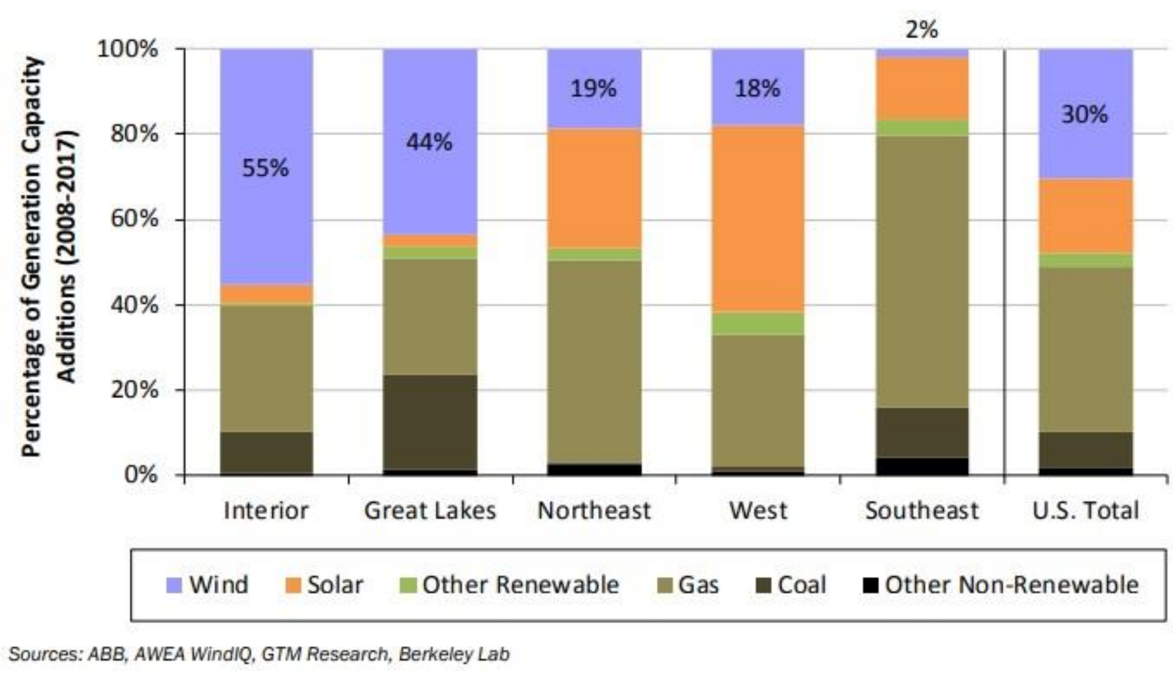


Fig. 1: Generation capacity additions by United States by region and energy type [2]

Wind penetration has also increased in many countries, with wind penetration being described as the fraction of energy produced by wind compared with the total generation. Fig. 2 shows the 23 countries that have the greatest amount of wind penetration in 2016 and 2017. As can be seen from the Fig. 2, Denmark is the world leader in wind penetration at 48%, followed by Ireland and Portugal with roughly 30%. The United States comes in at fifteenth place with about 7% wind penetration. On a global scale the total wind penetration is about 5% [2].

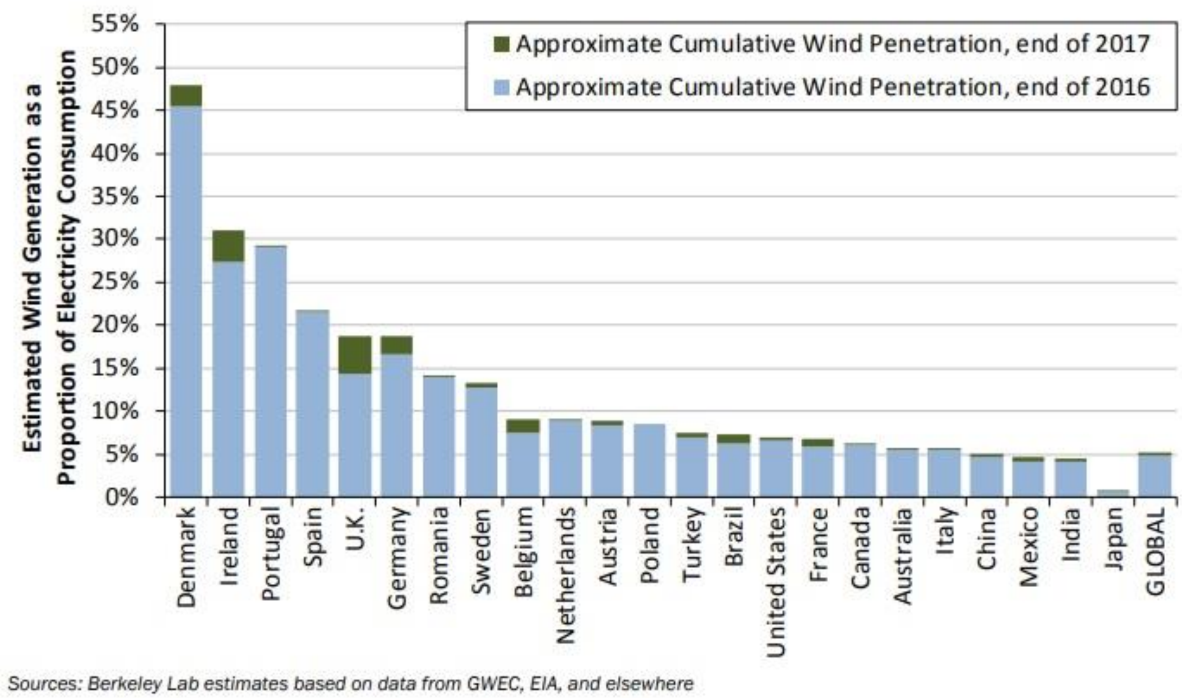


Fig. 2: Approximate wind energy penetration in the 23 countries with the greatest installed wind power capacity [2]

1.2 PROBLEM STATEMENT

While it is undeniable that the increased growth of wind and renewable energy across the world is very important to help curb carbon dioxide emissions and create a more sustainable way of producing energy, this situation poses new challenges for the operation of power systems. One reason for this is due to the fact that wind power exhibits a high degree of uncertainty and variability due to the uncertainty and variability of its "fuel" source. This variability can possibly cause a situation where there is not enough generation to balance the load if there is not enough available wind [3]. The main concern that will be studied here is the fact that the frequency regulation of a power grid is highly impacted when a considerable amount of electronically decoupled systems, such as wind energy, is connected to the system [4].

Grid operators desire to have generation that can provide regulation in order to maintain the necessary balance between generation and load, which then regulates the grid frequency. If the supply of power generated does not meet the required demanded power, the frequency of the system will change depending on the total system inertia [4]. System inertia is the actual amount of inertia that is given to the system by the large rotating masses of conventional generation units. Since electrical machines operate on the principle of an opposing electromagnetic and mechanical torque, changes in rotational speed will cause a change in the frequency of the system. A generator or load can be considered to contribute to system inertia if a change in system frequency causes a change in its rotational speed and, thus, its kinetic energy [5]. Fast acting power electronics essentially decouple the rotating mass of a wind turbine so that the power system can't "see" the inertia of the rotating turbine [4]. With higher wind penetration levels the total inertia of the system is decreased, meaning there is less natural frequency regulation present in the system. In this work solutions for allowing wind generation to participate in the frequency regulation of the power grid will be discussed, implemented, and analyzed.

1.3 LITERATURE REVIEW

There are a number of different approaches that researchers have proposed in literature to allow variable speed wind turbines to participate in frequency regulation. These approaches fall into two main categories. The first is an approach that emulates an inertial response by extracting extra power from the wind turbine rotor in the case of an under frequency event. The second is an approach that actually operates the wind turbine in a de-loaded mode to provide extra headroom, or spinning reserve, for frequency regulation.

The studies done in [6]-[13] provide an inertial response, sometimes called "virtual inertia," by immediately changing the torque set point of the generator when an under frequency event is detected. This allows the release of the rotor kinetic energy, which in turn causes the wind turbine rotor speed to decrease. This solution reduces the rate of change of frequency and the maximum frequency deviation from nominal [7, 8]. One drawback to this method is that the extra power provided is only very temporary, up to 10 seconds, it also does not impact the steady state frequency [14]. Another drawback to this method is due to the increase in torque in the generator, the rotor speed will decrease while power is produced above the power set-point. This may cause a situation where the wind turbine may stall [14]. Besides the possibility of a stall, this method increases power in the short run, but will require a recovery period in which the turbine rotor must spin up back to speed. During this recovery period the wind turbine will produce less power after the inertial response, which could affect the grid frequency [13]. Fig. 3

shows the relationship between the wind turbine rotor speed and the wind turbine power output when an inertial response has been activated.

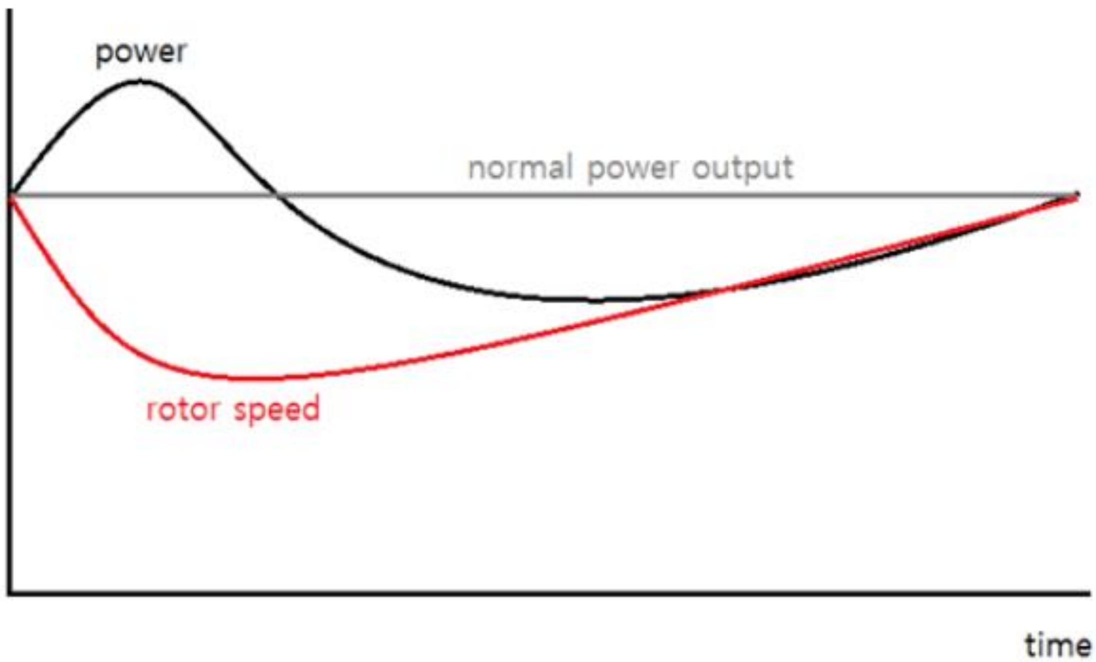


Fig. 3: Relationship between rotor speed and power produced after an inertial response [14]

The other main category of frequency control in wind turbines is achieved by de-loading the wind turbine. De-loading can be achieved in two different ways. The first method of de-loading is achieved by operating the wind turbine generator at a suboptimal generator speed [15]-[22]. The second method of de-loading is achieved by pitching the blades to intentionally spill power so that the reserve energy can be used for frequency regulation [15]-[17], [23]. Both of these solutions avoid the problems associated with the inertial response techniques, however, come with problems of their own. In fact, these problems are not so much associated with technical issues as with inertial control, but have more to do with loss of revenue. Typically de-loading of wind turbines is not practiced due to the potential loss of revenue (less power produced means less money for the operators of the wind plant). However, as power system operators have realized the problems associated with high wind penetration, some interconnect standards are requiring wind power to provide frequency regulation in the same way that conventional generation does [24]-[27]. De-loading is one way operators may act in accordance to abide by these new interconnect standards. On the flip side, the potential loss in revenue can be made up by allowing wind turbines to participate in the frequency regulation trade market [28]. In this work the option of de-loading through the use of pitch control is studied.

CHAPTER 2: OPERATING PRINCIPLES

2.1 WIND TURBINE TYPES

In general there are two classes of wind turbines that are determined by their axis of rotation. A turbine with a rotor axis of rotation that is horizontal to the ground is called a Horizontal Axis Wind Turbine (HAWT), while a turbine with a rotor axis of rotation that is vertical to the ground is called a Vertical Axis Wind Turbine (VAWT). HAWTs dominate as the main type of turbine used in large scale wind energy production [29]. HAWTs can then be divided further into two different categories: fixed speed and variable speed wind turbines. In fixed speed wind turbines the rotational speed of the turbine is set at a constant speed. When a fixed speed wind turbine is connected to a squirrel cage induction generator it is known as a Type-1 wind turbine. Type-2 wind turbines are connected to wound rotor inductor generators in a similar manner to Type-1 generators with the addition of variable resistors to help keep constant power output even during gust conditions [30]-[32]. While the operation and control of fixed speed wind turbines is fairly simple they are unable to efficiently extract the maximum available power from the wind. Due to this reason, the industry has adopted variable speed wind turbines as the turbine of choice due to advantages such as improved efficiency for a large range of wind speeds, improved power quality, and reduced mechanical stress [33].

Variable speed wind turbines are able to adjust their rotor speed based on wind speed and rated power of the generator through means such as pitching the blades or by adjusting the generator torque. Variable speed wind turbines can also be divided into two types based on the generator used. A Type-3 wind turbine is characterized by the use of a Doubly Fed Induction Generator (DFIG) that contains a multistage gearbox and a back to back partially rated power electronic converter in the rotor circuit [30]. Type-4 wind turbines utilize a Direct Drive

Synchronous Generator (DDSG) with a full scale back-to-back power electronics converter due to the elimination of the gearbox [30]. Here, the basic division between different types of wind turbines has been established, but not expanded on. This work will focus on the operation and control of Type-3 wind turbines, thus, leaving further explanation on other wind turbine types beyond the scope of this work.

2.2 WIND TURBINE AERODYNAMICS

The purpose of a wind turbine is to capture the kinetic energy contained in the wind and convert it into electrical energy. To derive the equation for the amount of power the turbine is able to extract from the wind we begin with the equation for the kinetic energy contained in wind [34, 35]. This is shown in equation (1):

$$KE = \frac{1}{2}mv^2 \qquad (1)$$

The kinetic energy equation must be fleshed out to incorporate the volume of wind traveling through the swept area of the rotor blades. The power of moving air is represented in terms of the mass flow rate $\dot{m}$, the velocity of the wind entering the blades V, and the velocity of the wind exiting the blades, V_0 [36].

$$P = \frac{1}{2}\dot{m}(V^2 - V_0^2) \qquad (2)$$

The mass flow rate is given by Equation (3), where ρ is the density of air and A is the area swept by the rotor blades:

$$\dot{m} = \rho A(\frac{V+V_0}{2}) \qquad (3)$$

Combining Equation (2) and (3) will give an expression for the power that is captured by the rotor:

$$P = \frac{1}{2}(\rho A(\frac{V+V_0}{2}))(V^2 - V_0^2) \tag{4}$$

The terms in Equation (4) can be rearranged to describe the efficiency of the rotor given by Equation (5):

$$C_p = \frac{(1+\frac{V_0}{V})+(1-(\frac{V_0}{V^2}^2)}{2} \tag{5}$$

The term C_p is called the power coefficient and describes the fraction of the wind energy extracted by the wind turbine [36]. The reasoning behind defining this parameter is due to the fact that the wind turbine rotor cannot extract all of the energy from the wind stream, as this would require the wind to become stationary on the downwind side of the rotor. Equation (5) may be a good qualitative way to think about the amount of power extracted from the wind but an alternate definition based on blade theory and momentum theory will relate the power captured by the turbine to actual turbine parameters. The power coefficient can be described as a function of the blade pitch angle β, and by the tip speed ratio λ. The tip speed ratio is the ratio of the tangential speed of the tip of the blade to the wind speed, given by:

$$\lambda = \frac{\omega_r R}{V} \tag{6}$$

Where ω_r is the angular speed of the wind turbine rotor and R is the radius of the turbine (blade length) [37]. Now the power coefficient can be described as a function of blade pitch and tip speed ratio:

$$C_p(\lambda, \beta) = c_1(\frac{c_2}{\lambda_i} - c_3\beta - c_4)e^{\frac{-c_5}{\lambda_i}} + c_6\lambda \tag{7}$$

and [37]

$$\frac{1}{\lambda_i} = \frac{1}{\lambda+0.08\beta} - \frac{0.035}{\beta^3+1} \tag{8}$$

The coefficients c_1 thru c_6 are given by 0.5176, 116, 0.4, 5, 21, and 0.0068, respectively [37].

It can be seen that the rotor efficiency is highly non-linear and makes the entire system a non-linear system. Tip speed ratio serves as a parameter for several curves that are dependent on the blade pitch angle seen in Fig. 4 below.

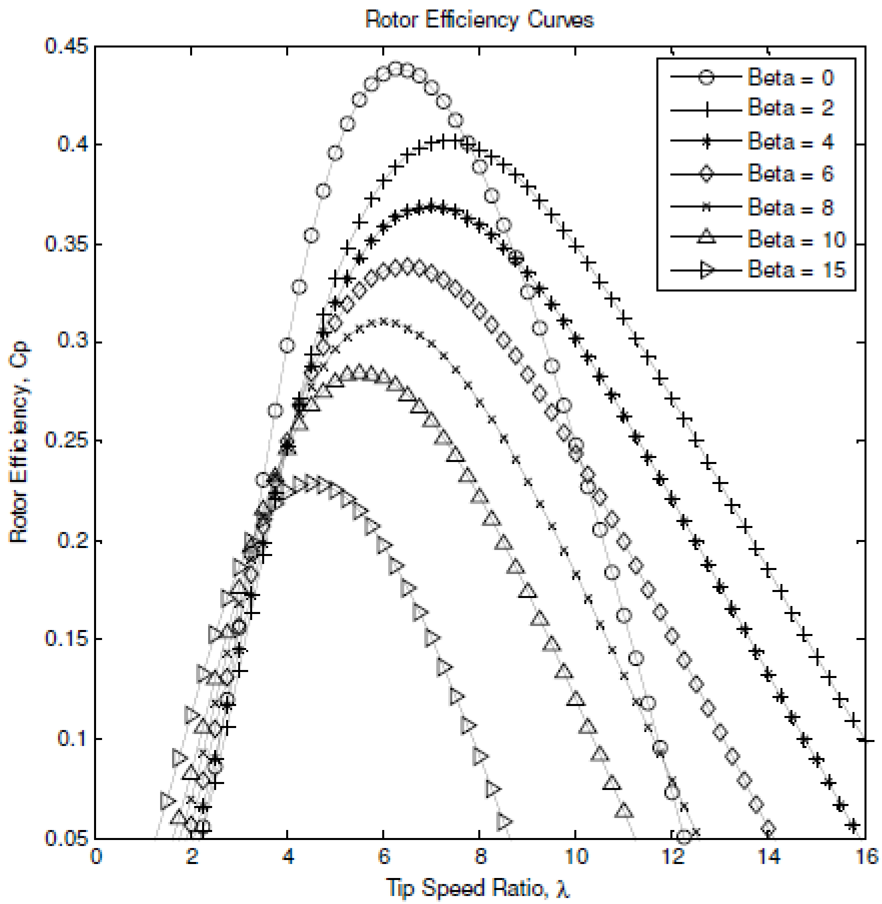


Fig. 4: Rotor efficiency curves for different values of the blade pitch and tip speed ratio [36]

Due to reasons mentioned above, the turbine cannot extract all the energy from the wind and the theoretical upper limit for rotor efficiency is the Betz Limit of $\frac{16}{27}$ [38]. Given all this information, the above equations can be combined to give the total power captured by the turbine as:

$$P = \frac{1}{2}\rho A v^3 C_p(\lambda, \beta) \tag{9}$$

2.3 Regions of Operation

Wind turbine control can be separated into four main categories, as seen in Fig. 5. Region 1 is simply when there is not enough wind power to turn the turbine until the "cut-in" wind speed is achieved, then the generator is turned on and starts producing power. Once the wind speeds are high enough (above the cut-in speed), the turbine is in Region 2. In this below rated region of operation the purpose is to maximize aerodynamic efficiency and capture as much energy as possible from the wind [14]. In Region 3 wind speeds are high enough to allow the generator to produce its maximum rated power. In this case the goal is to regulate the rotor speed through the use of pitch control and power control to keep the wind turbine at safe operating levels. Region 4 is condition when the turbine shuts down due to high wind speeds to prevent damage to the turbine. In Fig. 5 the difference between the available power and the power extracted is due to the Betz limit mentioned previously.

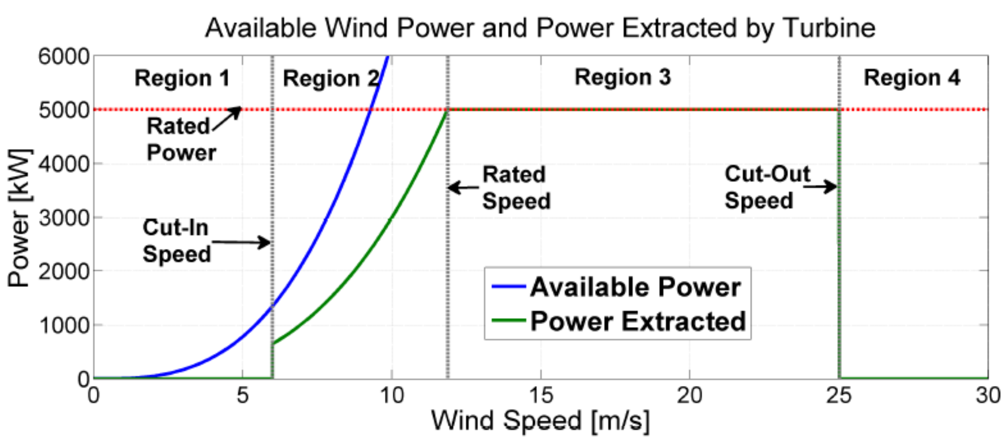


Fig. 5: Wind power, turbine power, and operating regions for an example 5 MW turbine [14]

2.4 DOUBLY FED INDUCTION GENERATOR

The aerodynamic torque captured by the blades is transferred to the hub, which connects the blades to a drivetrain and then a generator. Normally, the drivetrain includes a gearbox to scale rotational speed and torque to levels that are suitable for the generator configuration. However, in this work the drive train has been eliminated in favor of a single lumped mass model, meaning the mass from the wind turbine rotor and the mass from the generator are treated as a single mass [39]. For a full treatment and derivation of the drivetrain with a gearbox please see [40].

A very important property that variable speed wind turbines should possess is to be able to provide a constant frequency output voltage from a variable speed system. A DFIG can have its rotor speed vary while still providing constant voltage and frequency [41]-[45]. This allows more flexibility in power conversion and also allows for better stability in frequency and voltage control in the power system. A DFIG consists of a wound rotor induction generator (WRIG) with the stator windings directly connected to the three-phase grid/load and the rotor windings connected to a back-to-back partially rated (20–30% rating) power converter [46]-[53]. Fig. 6 shows how the power converter is connected to the grid and the generator as well as the power flow of the system as a whole.

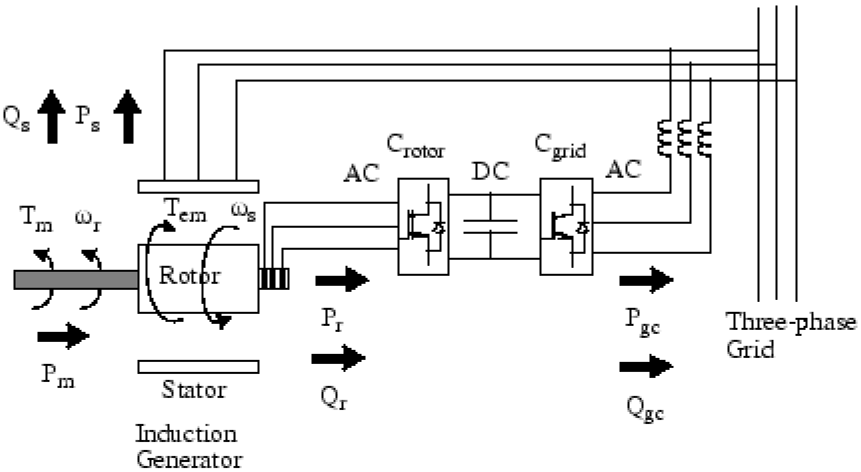


Fig. 6: Schematic of a DFIG showing the power flow [54]

The power converter is essentially an AC/DC/AC converter that consists of two components: a rotor side converter (RSC) and a grid side converter (GSC). The RSC and GSC are voltage converters that use power electronic devices (Insulated Gate Bipolar Transistors) to create an AC voltage from a DC voltage. A coupling inductor is used to connect the GSC to the grid while a capacitor is connected to the DC side of the converter and acts as the DC voltage source. Slip rings and brushes connect the three-phase rotor windings to the RSC while the three-phase stator winding is connected directly to the grid. The mechanical torque generated by the wind turbine is converted into electrical power by the induction generator and is then transmitted to the power grid by the stator and the rotor windings [50, 51].

2.5 POWER SYSTEM FREQUENCY CONTROL

The frequency of a power system is an important parameter that must be kept in a tight range for the system to maintain efficiency and reliability. The power system in North America is divided in four different sectors, or interconnects, that can be thought of as being independent islanded networks that are frequency independent from each other [56]. Fig. 7 shows the domains of each interconnect.

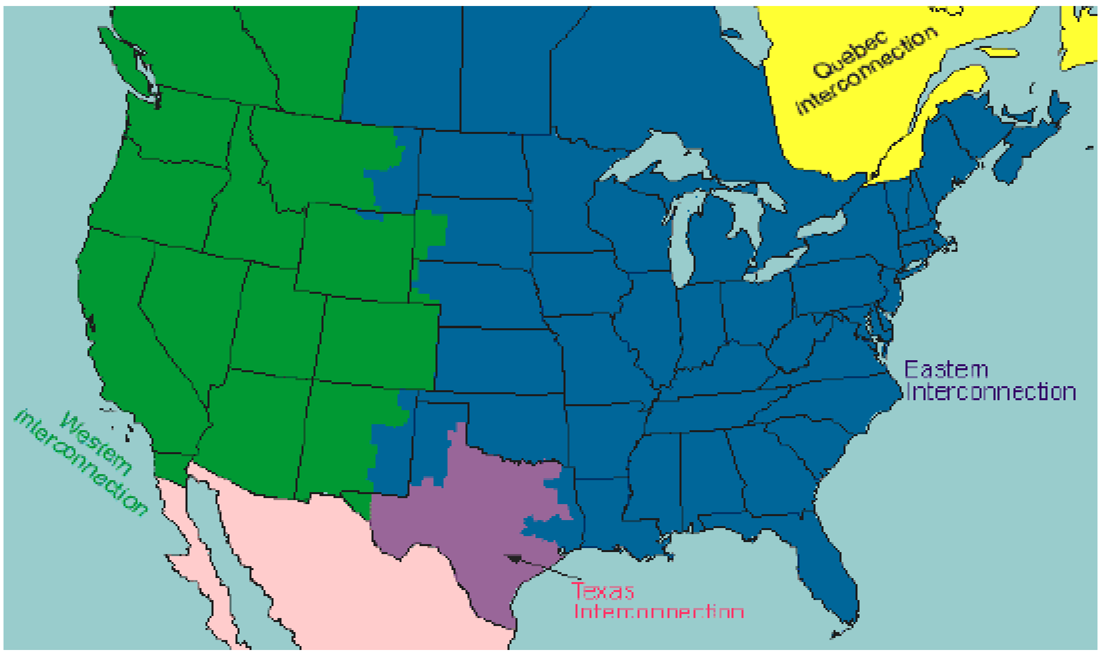


Fig. 7: North American Interconnects [56]

Each interconnect can be seen as a single large machine, with dispersed generation units that work together to meet the demand of supplying electricity to customers. The frequency of each interconnect system is determined by the combined rotational speeds of all the generating units within the interconnect. When the total generation in the interconnect exceeds the customer demand, this will tend to cause the frequency of the system to rise above the nominal value, which is 60 Hz in North America [57]. In a contrary manner, when the total generation is not high enough to meet customer demands the frequency will tend to drop below nominal. Fig. 8 gives a good intuition of this idea by imagining that generation and demand are on opposite sides of a scale that need to be balanced.

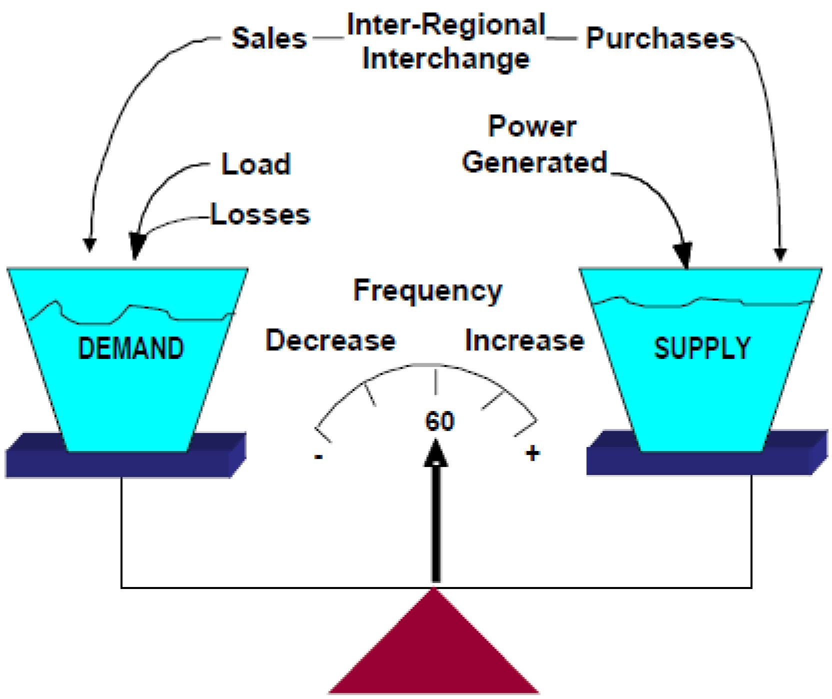


Fig. 8: Balancing supply and demand to maintain system frequency [56]

Due to this need for balance, power system operators need controllable generation that will react to any imbalances in the system. The control actions used to control the system frequency are typically applied in different time frames. Table 1 describes the different control regimes with the rows corresponding to classifications on how the generation and load should respond and the appropriate time fames in which these responses should take place.

Table 1: Control Regimes showing the time frames in which they act [56]

Control	Ancillary Service/IOS	Timeframe
Primary Control	Frequency Response	10-60 Seconds
Secondary Control	Regulation	1-10 Minutes
Tertiary Control	Imbalance/Reserves	10 Minutes - Hours
Time Control	Time Error Correction	Hours

Due to the decoupled nature of wind turbines, their impact on the system frequency is most noticeable in the first few seconds of an under frequency event. In light of this fact, the work presented here will focus on the primary control regime.

Primary control relates to the responses of the generation units, including generator governors, which stabilize the system frequency whenever there is a change in the load/demand balance. Primary Control is provided in the first few seconds following a frequency change and is continued until it is replaced by the secondary control, also commonly known as Automatic Generation Control (AGC). In large part primary frequency control is provided by generators adjusting their power production through the use of generator governors. Governors operate in the timeframe of milliseconds to seconds and they protect from the effects of over frequency events, but their major benefit comes from protecting the system when frequency has dropped too low, especially in cases where loss of generation causes sudden decreases in system frequency [58]. This is done by allowing the governor to constantly regulate the amount of mechanical input energy to the shaft of the electric generator. The amount of regulation provided is called the slope or droop, and is measured in percent of frequency change to cause maximum power from the generator to be applied against the frequency error. It is important to realize the limitations of governors as well. While most generators can easily reduce their output in response to their governor's actions, increasing output is more problematic. This is due to the fact that generating units may already be near the top of their output capability. When low frequencies cause generator governors to request more output, if there is no headroom (spinning reserve) available on a generator's output, the governor will be able to do little to increase its output and help stabilize low frequencies back to nominal [58, 59]. Fig. 9 shows a typical frequency response of the primary control action on a power system after a loss of generation event. Please see [56] for a more detailed description of the standards and norms for frequency regulation.

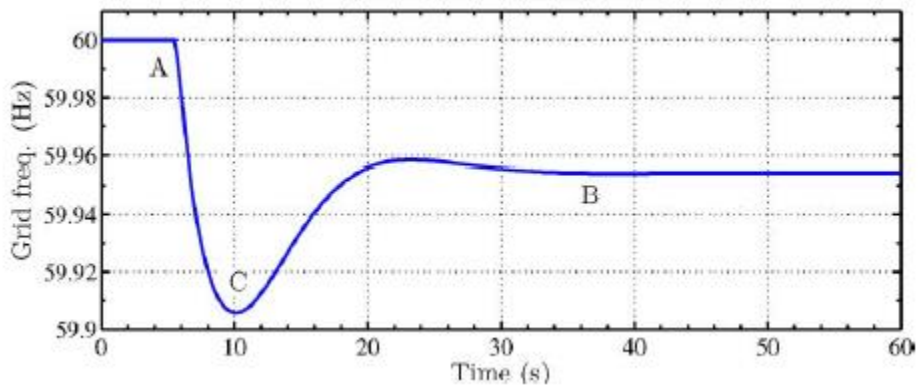


Fig. 9: Primary Frequency response due to loss of generation [56]

There are three important factors to consider when examining the frequency response of a power system. The first is the Rate of Change of Frequency (RoCoF), which describes the rate (how fast) the frequency is declining and should not be above certain limits due to the fact that very rapid changes in the frequency will be highly stressful on the mechanical components of the generating units [60]. Second is the Frequency Nadir, which is the lowest value that the frequency reaches before it begins to recover, and can be seen as point C in Fig. 9. Lastly, the Settling Frequency is the frequency at which the frequency stabilizes due to the primary frequency control, and is seen as point B in Fig. 9. Notice that during primary frequency control the frequency is not brought back to nominal, but tends to settle somewhere below nominal. This is normal for this particular control regime which typically last only 10-60 seconds. After this timeframe the secondary control (AGC) is activated to bring the system frequency back to nominal; this can be seen in Fig. 10. Note that the differences in typical timeframes for each control regime between Fig. 9 and 10 may not be exactly the same, but give a more general feel for the timeframes involved for each control regime.

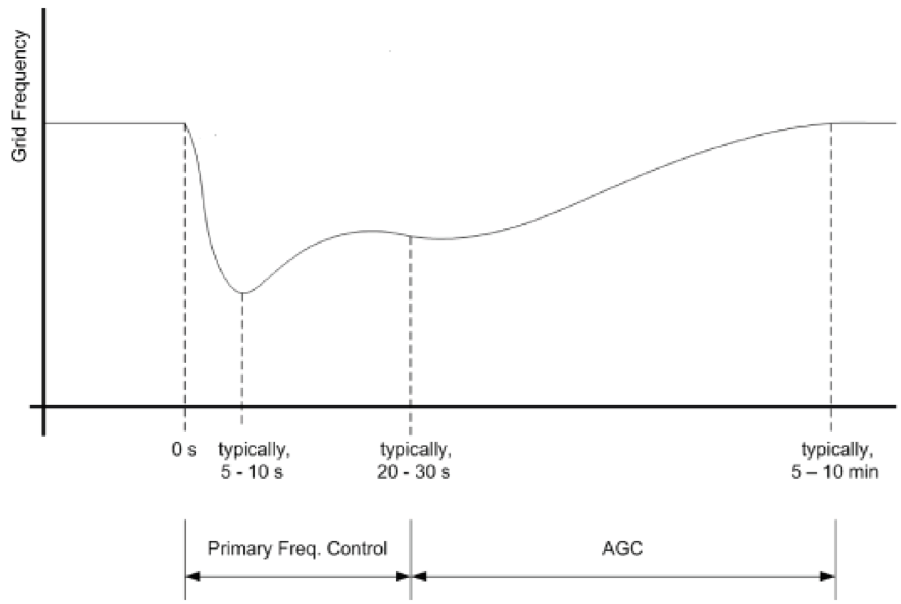


Fig. 10: Frequency response due to primary frequency control and AGC [61]

CHAPTER 3: ENABLING FREQUENCY REGULATION IN WIND TURBINES

3.1 EXISTING PITCH CONTROL

During normal operation of a wind turbine there may be wind speeds that will cause the rotor to spin faster than its rated speed. The solution of being able to limit the speed of the turbine rotor is the addition of pitch control. The practical implications of this is when the available wind power is above the equipment rating, the blades are pitched (in other words the angle of attack is increased) away from zero degrees, this allows less wind to impact the turbine blades, thus, lowering the rotor speed and limiting the mechanical power delivered to the rotor, gearbox, and generator. This is typically achieved using a Proportional-Integral (PI) controller that takes the rotor speed compared to a reference speed as an error signal and outputs a command to pitch the blades should the wind speed become too high [62, 63]. Fig. 11 shows a block diagram that shows the control topology of the active power control portion of the wind turbine/generator system.

The two main blocks to note at this point are the Pitch Angle Controller, and the Pitch Angle Compensator, both of which are PI controllers. As can be seen the pitch angle controller takes a rotor speed error as an input and outputs a pitch command, as explained above. The input to the pitch angle compensator shows that electric power output (P_{ord}) is set to follow the dispatched power set-point (P_{setlf}), and the purpose for this controller is to provide a supplemental pitch command that will cause the blades to pitch when the output power rises above the set-point [62, 63].

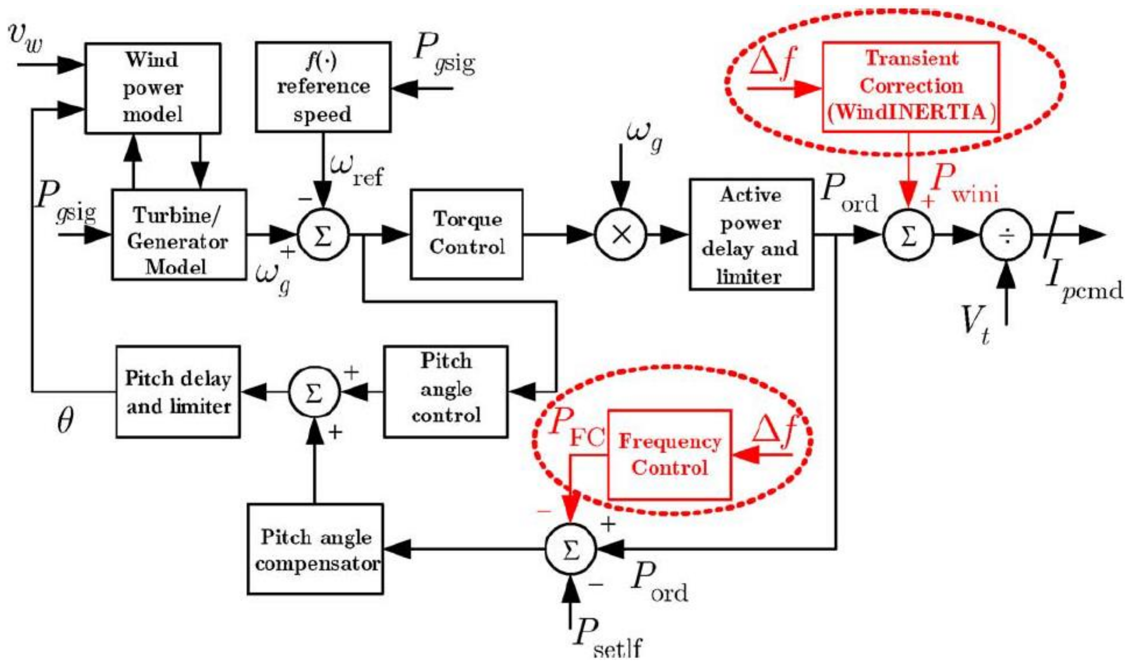


Fig. 11: Block diagram showing active power control portion of the wind turbine [62, 63]

In this configuration of having only a pitch controller and compensator, the pitch angle of the blades depends only on the rotor speed and power output. To make the pitch system responsive to frequency deviations, additional controllers must be incorporated. This type of controller is a two part controller that is shown in red in Fig. 11.

3.2 GOVERNOR/DROOP CONTROL

To allow a frequency response in wind turbines, the solution presented here is to add a governing/droop like control to the pitch control system. In Fig. 11 this controller is labeled in red as “frequency control.” The proportional control structure show in Fig. 12, is designed to generate a supplemental power order P_{FC}. The dead-band serves two purposes: it stops the controller from acting on small fluctuations in the frequency that is noise as well as making the controller unresponsive to over frequency events, which is less problematic when wind power is connected to the system [62, 63].

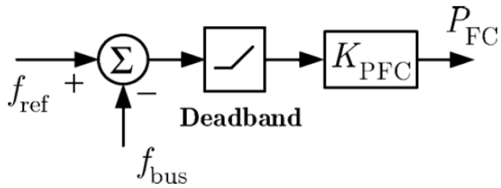


Fig. 12: Proportional governor/droop control to enable a frequency response in wind turbines [62, 63]

The controller itself is quite simple; only consisting of a proportional parameter that takes a frequency error as an input. The frequency error is a comparison between the actual grid frequency and the reference frequency (60 Hz in North America). The value of the proportional coefficient will determine how strongly the controller will participate in frequency regulation. It is important that the output of this controller be summed with the error that is the input to the pitch compensator. Adding the output of this controller to any other point in Fig. 11 will cause the integrators in the other PI controller blocks to cancel out the effects of the proposed governor controller in steady state [62, 63]. Summing the supplemental power order generated by this controller to the input of the pitch compensator will guarantee modifications in the power captured by the wind turbine by supplying a higher power order. The main effect this controller has on the frequency response of the system is improving the settling frequency, and it does offer some small benefit in improving the frequency nadir [63].

3.3 TRANSIENT FREQUENCY RESPONSE CONTROLLER

To improve the frequency nadir (and RoCoF) of the system, it is prudent to add a transient frequency response controller that will not interfere with any of the existing controllers. The idea is to introduce a rate control which will respond quickly to frequency variation. In this vein, the control structure called WindINERTIA control developed by GE for their Type-3 wind turbines is shown in Fig. 13.

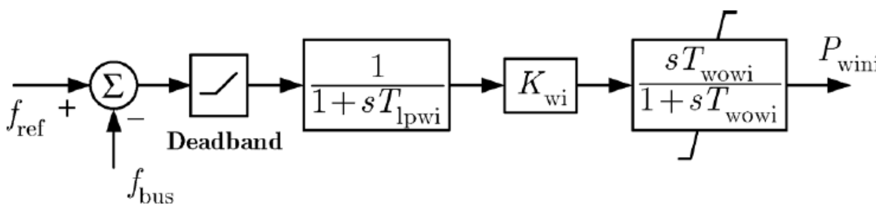


Fig. 13: WindINERTIA Transient Frequency Response Controller [62, 63]

This controller structure also takes a frequency error as an input and the dead-band serves the same purpose as in the governing control. This controller is also a simple proportional controller but also features a low-pass filter and a washout filter; the washout filter is the main ingredient that allows the passage of only transient inputs. The special structure of a washout filter allows it to act as a high-pass filter that will reject steady-state inputs while passing only transient inputs [64], meaning in steady-state this controller will be inactive. The value of the proportional coefficient will determine how strongly the controller will participate in frequency regulation. With this information, it is appropriate to add this control signal to the injection point as shown in Fig. 11, because in this configuration the WTG will boost its power output immediately during an under frequency event.

CHAPTER 4: SIMULATION

4.1 KRK TEST SYSTEM

The Klein-Rogers-Kunder (KRK) is a simple test power system that was developed to study low frequency inter-area oscillations [65, 66]. The system consists of two identical areas connected through a relatively weak tie line. Each area includes two generating units with equal power outputs and the power production in each area equals the load, shown in Fig. 14. The numbers above each line in Fig. 14 indicate the tie-line impedance. Even though the system is small and simple the authors of [66] state that conclusions drawn from their work apply well to large, more complicated systems. The simplicity and accuracy of the model makes it ideal for simulation. The full symmetry of the system also makes it very useful for studying how wind penetration affects the frequency response.

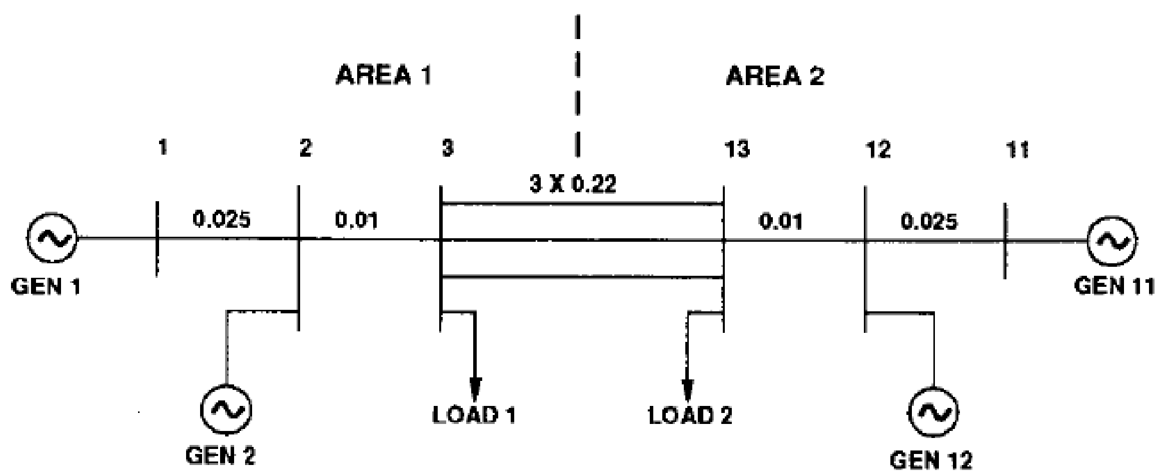


Fig. 14: Original KRK Test System [66]

In order to test how wind penetration affects the frequency response the system is modified slightly by adding a small generator in the middle of the transfer path between the two areas, shown in Fig. 15. This will allow for a loss of generation event, by tripping the generator that will cause the frequency to decline. Due to its location, tripping the generator will not excite any inter area modes [66, 67] and there will be no power transfer between the two areas due to the equal balance of generation and load (demand) in each area.

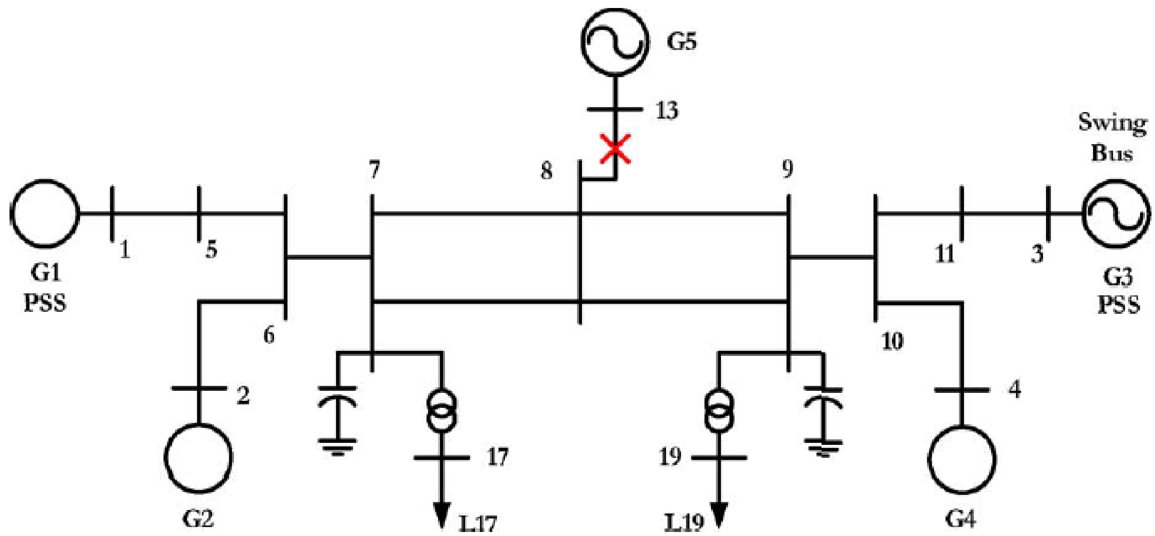


Fig. 15: Modified KRK system with a small generator in the center of the transfer path [67]

4.2 KRK IMPLEMENTATION IN SIMULINK

In this work Matlab/Simulink© was the software of choice used as the platform for simulation of the system. This choice was driven by a number of factors, the main being that Simulink© is a physical modeling environment that includes a large number of highly detailed components found in power systems. Most of the component blocks that were used came from the Simscape Power Systems Toolbox [68]. Matlab/Simulink© also has powerful features that allow for flexible controller design and analysis. This section will describe the actual implementation of the KRK system in Simulink© and the reader is referred to the documentation provided by MathWorks [69] for a full description on how each block is precisely modeled.

Fig. 16 shows the top level view of how the KRK system is implemented in Simulink©. Each colored box is a subsystem that contains the necessary components for each area.

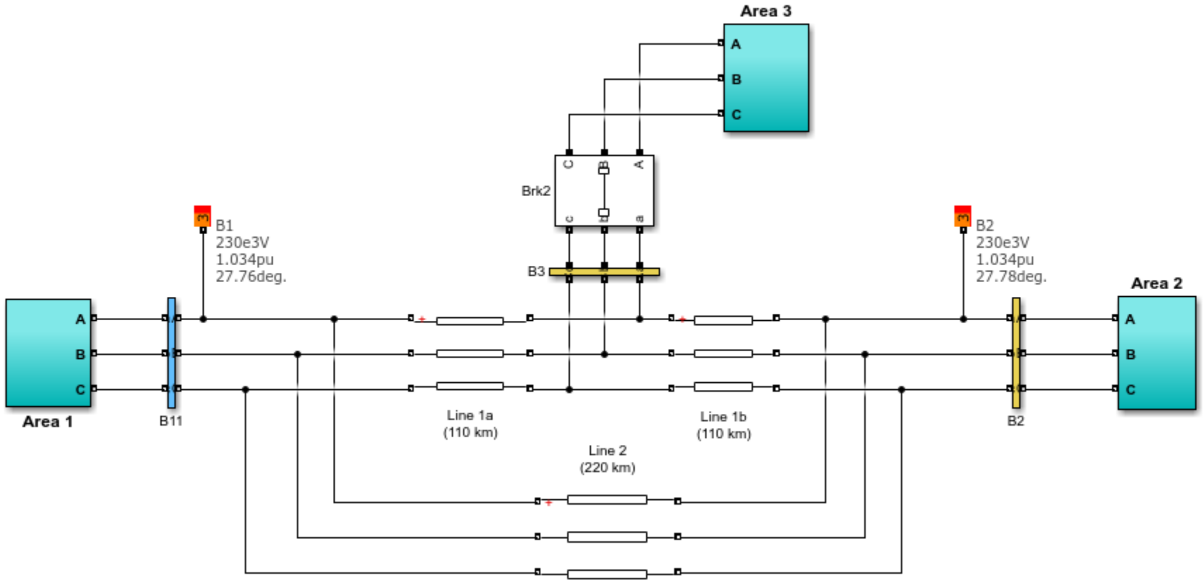


Fig. 16: Top level KRK system in Simulink©

It is important to note that Area 3 is the added small generator that is in the center of the transfer path, and is able to be tripped through the use of a breaker. The complete specifications used for the KRK system such as tie line length/impedance, system voltage etc. were implemented as described in [65], and for brevity the specifications and parameters used for each block in the simulation are detailed in Appendix A.

Fig. 17 shows Area 1 of the simulation. The generators are 3-phase synchronous machines that are modeled with IEEE type 1 synchronous machine voltage regulator combined with an exciter as well as having a tandem-compound steam prime mover system, including speed regulator, steam turbine, shaft, and a power system stabilizer (PSS) [69]. Each generator is connected to a 20kV/230kV step up transformer that is fed into the tie-lines. The load is modeled as a constant impedance load with capacitor banks in parallel for power factor correction that will allow for a better voltage profile [65]. The tie-lines are three-phase and modeled using the pi model.

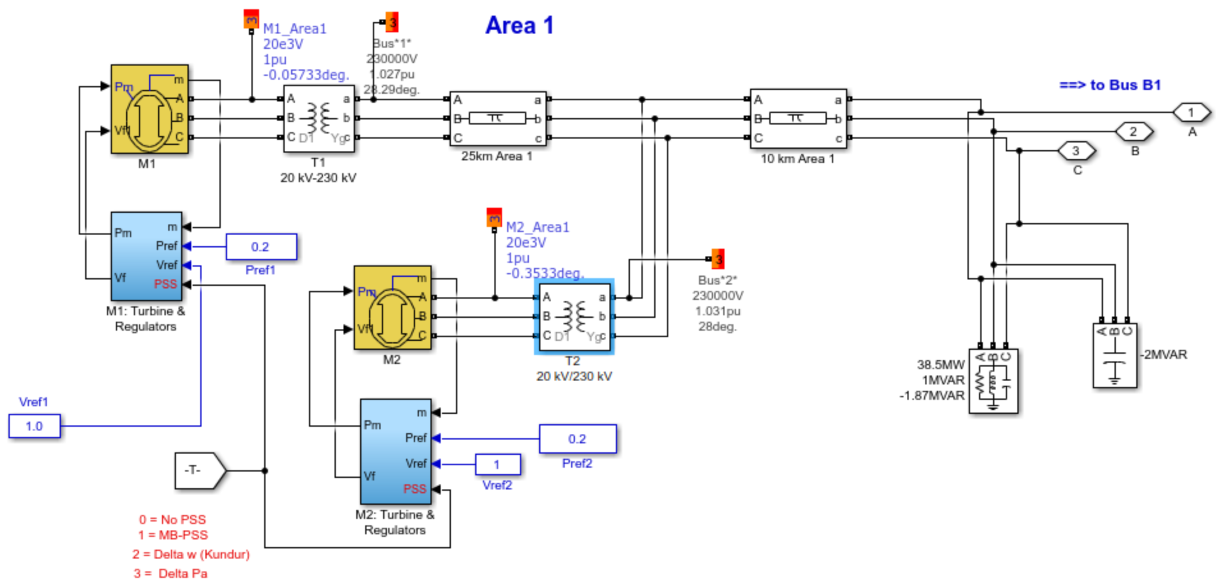


Fig. 17: KRK Area 1 in Simulink©

For the sake of comparison there will be two different versions of the simulation: the original, in which Area 1 and Area 2 are identical, and a modified version in which one of the generators in Area 2 is replaced by a wind turbine of the same rating. Fig. 18 shows how Area 2 is modified to include a wind turbine and Fig. 19 shows how the small generator is connected to the system. Table 2 shows the load and power generation specifications; note that the amount of generation appears slightly higher than the load. This is done to make up for system losses and allows for the balance of generation/load in each area.

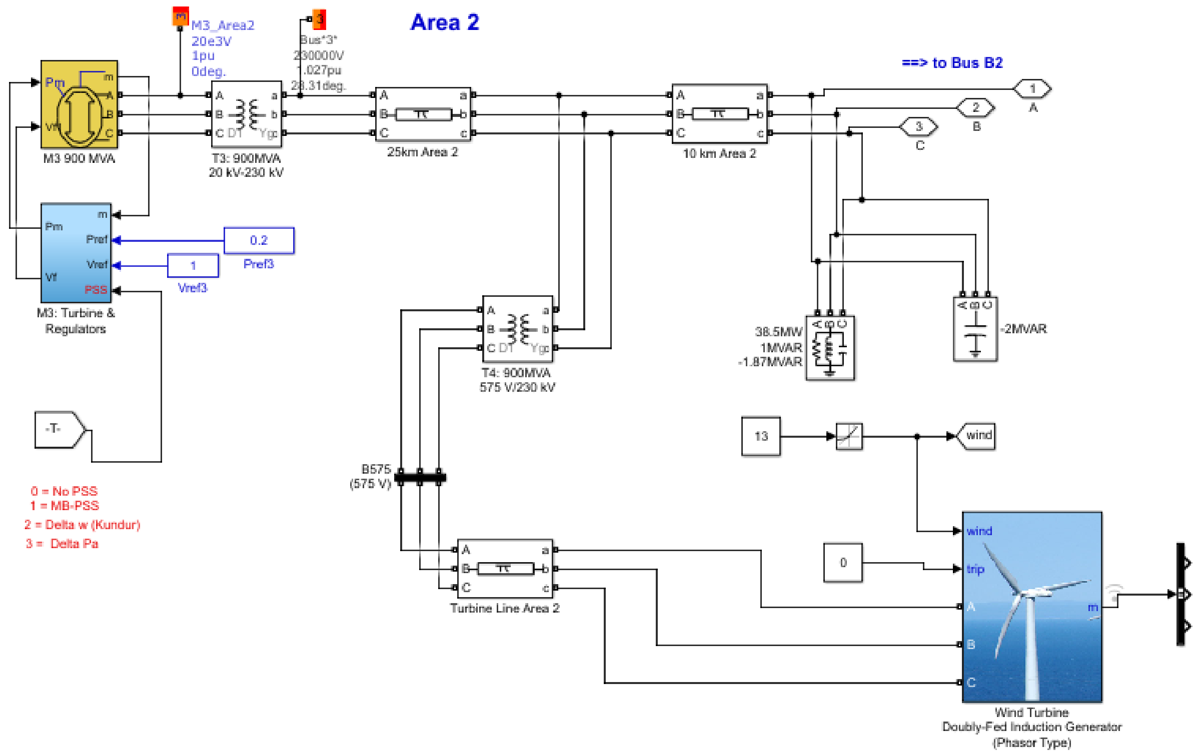


Fig. 18: KRK Area 2 in Simulink© with a Wind Turbine

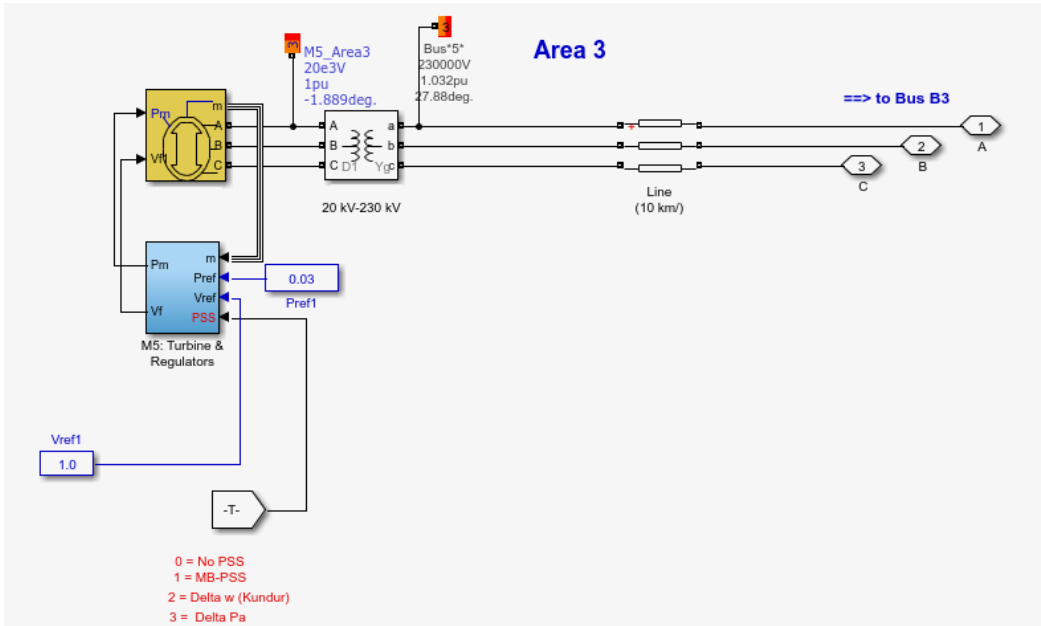


Fig. 19: Addition of a small generator in Simulink©

Table 2: Test System Generation and Load Parameters

	Power (MW)	Reactive Power (MVAR)
Area 1: Gen 1	20	X
Area 1: Gen 2	20	X
Area 2: Gen 1	20	X
Area 2: Gen 2	20	X
Area 3: Gen 1	3	X
Load: Area 1	38.5	-2.87
Load: Area 2	38.5	-2.87

4.3 TYPE-3 WIND TURBINE IMPLEMENTATION IN SIMULINK

The block in Simulink© that is used to simulate the wind turbine/generator system is called "Wind Turbine Doubly-Fed Induction Generator (Phasor Type)." It has three inputs consisting of a wind speed, trip signal, and three-phase grid connection. There are three different modeling options for simulating the wind turbine system: detailed, average, and phasor model. The detailed and average models are suitable for observing harmonics and control system dynamics over relatively short periods of times from hundreds of milliseconds to about one second [69]. This is due to the discreet nature of the simulations with small time steps. The phasor model is a continuous time simulation and is better suited to simulate the low frequency electromechanical oscillations over long periods of time from tens of seconds to minutes [69]. In the phasor simulation method, the sinusoidal voltages and currents are replaced by phasor quantities (complex numbers). The phasor model is used in this study due to the fact that the timeframe of the primary frequency regime is on the order of 60 seconds.

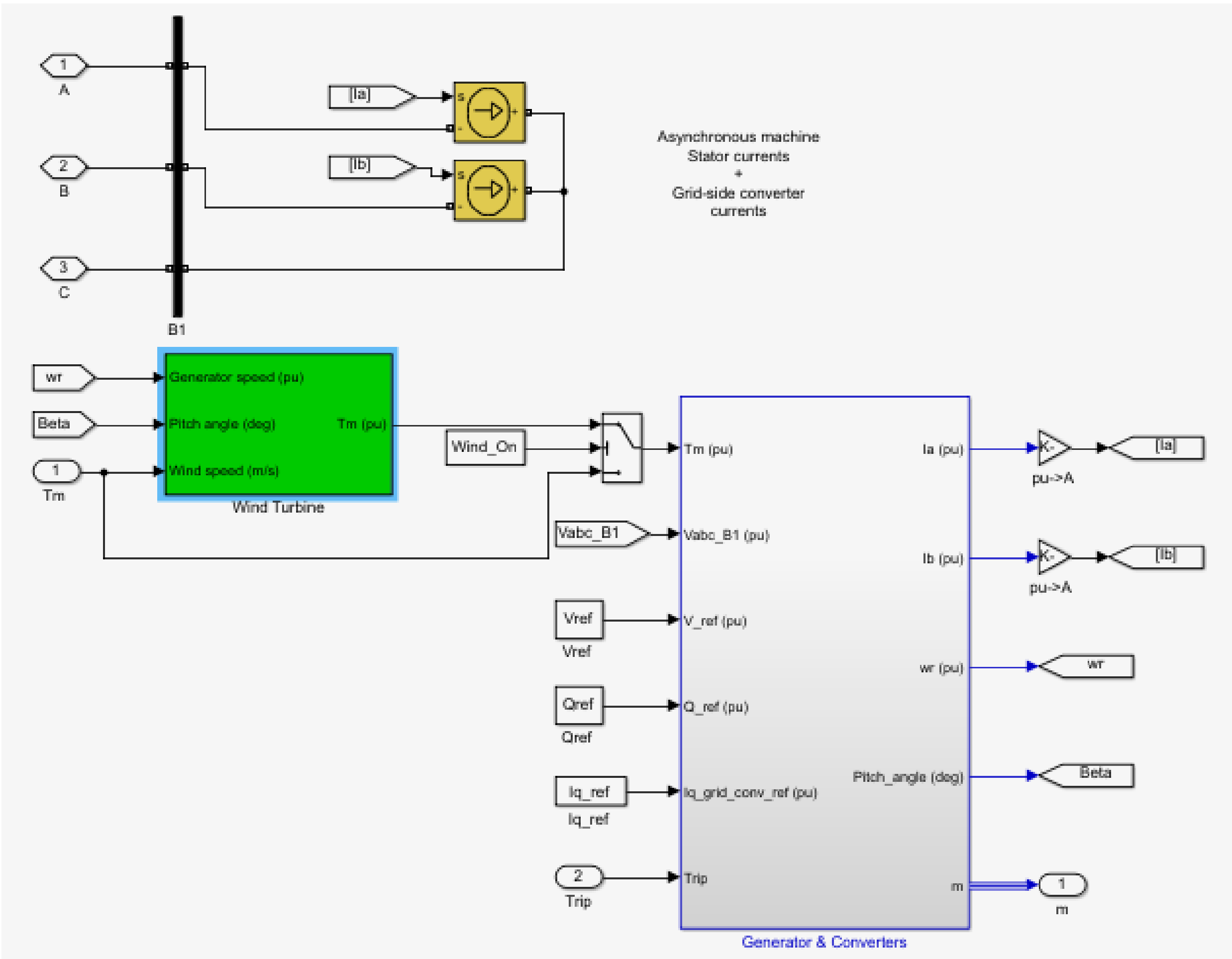


Fig. 20: Subsystems of the wind turbine phasor model

The wind turbine block is made up of a number of subsystems shown in Fig. 20. The top portion of Fig. 20 models the system's connection to the grid. The green block takes the wind speed, pitch angle, and rotor speed as inputs and calculates the power extracted from the wind as described in the Wind Turbine Aerodynamics section. It then converts the power to a torque and inputs this signal into the block labeled "Generator & Converters."

This Generators & Converters block also has a number of subsystems as shown in Fig. 21. The yellow block is a phasor model of the asynchronous machine that takes grid voltages a rotor control signal as an input and calculates the output stator currents and the electrical torque.

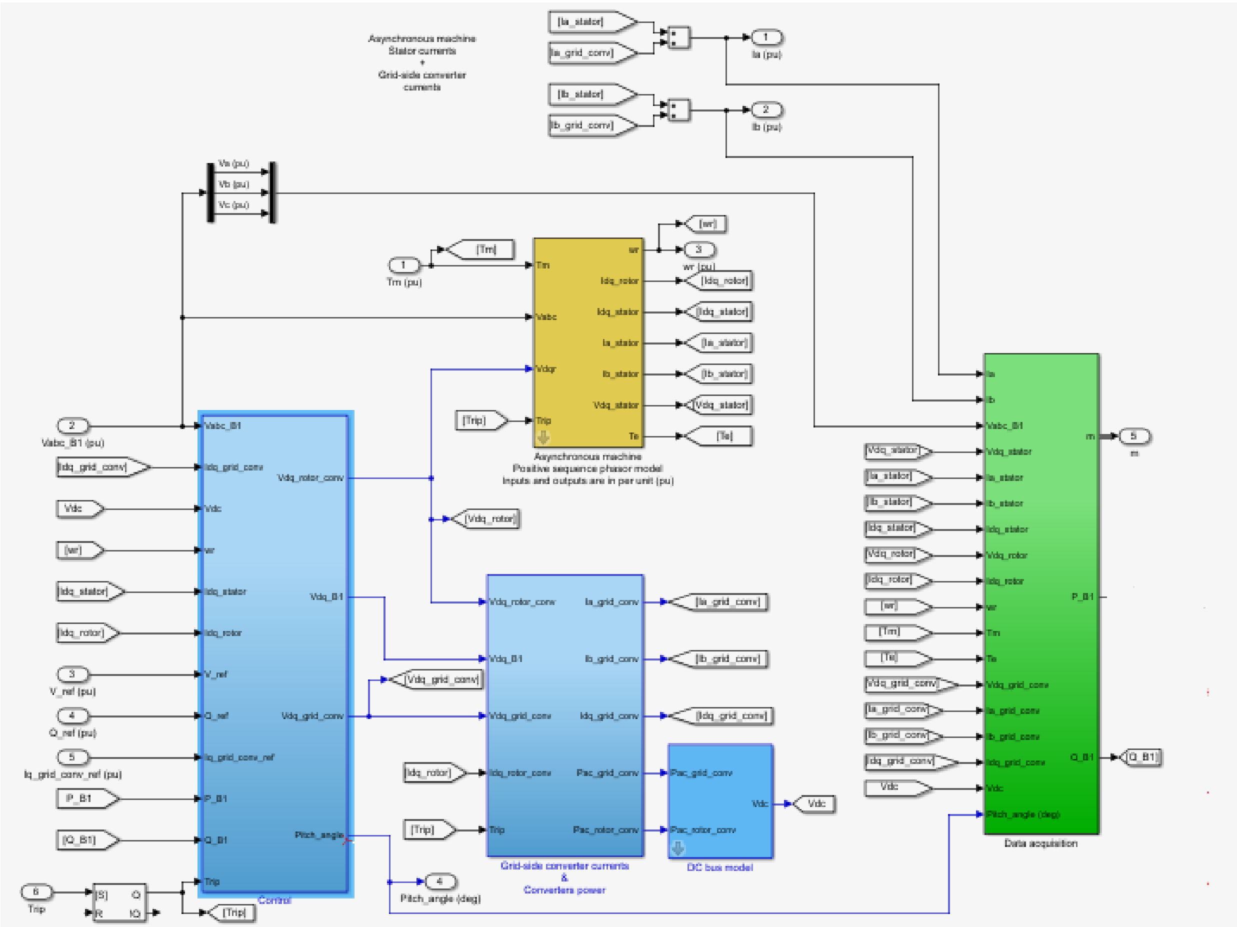


Fig. 21: Sub systems in the “Generator & Converters” block

The “Grid-side converter currents & Converter power” block models the partially rated power converter by taking control voltages as inputs and calculates the grid converter currents. It also calculates the converter power and sends the signal to the “DC Bus model” which models how the capacitor acts as a DC bus for the AC/DC/AC converter. The large green block is simply a data acquisition block for use in seeing the appropriate output signals of the wind turbine. The final large blue block is the “Control” block and this block provides the control signals to the system.

The subsystems of the Control block is shown in Fig. 22. The “wind_dfig_grid” block is the control system for the grid-side converter system. This block contains a current regulator that is controlled by a PI controller.

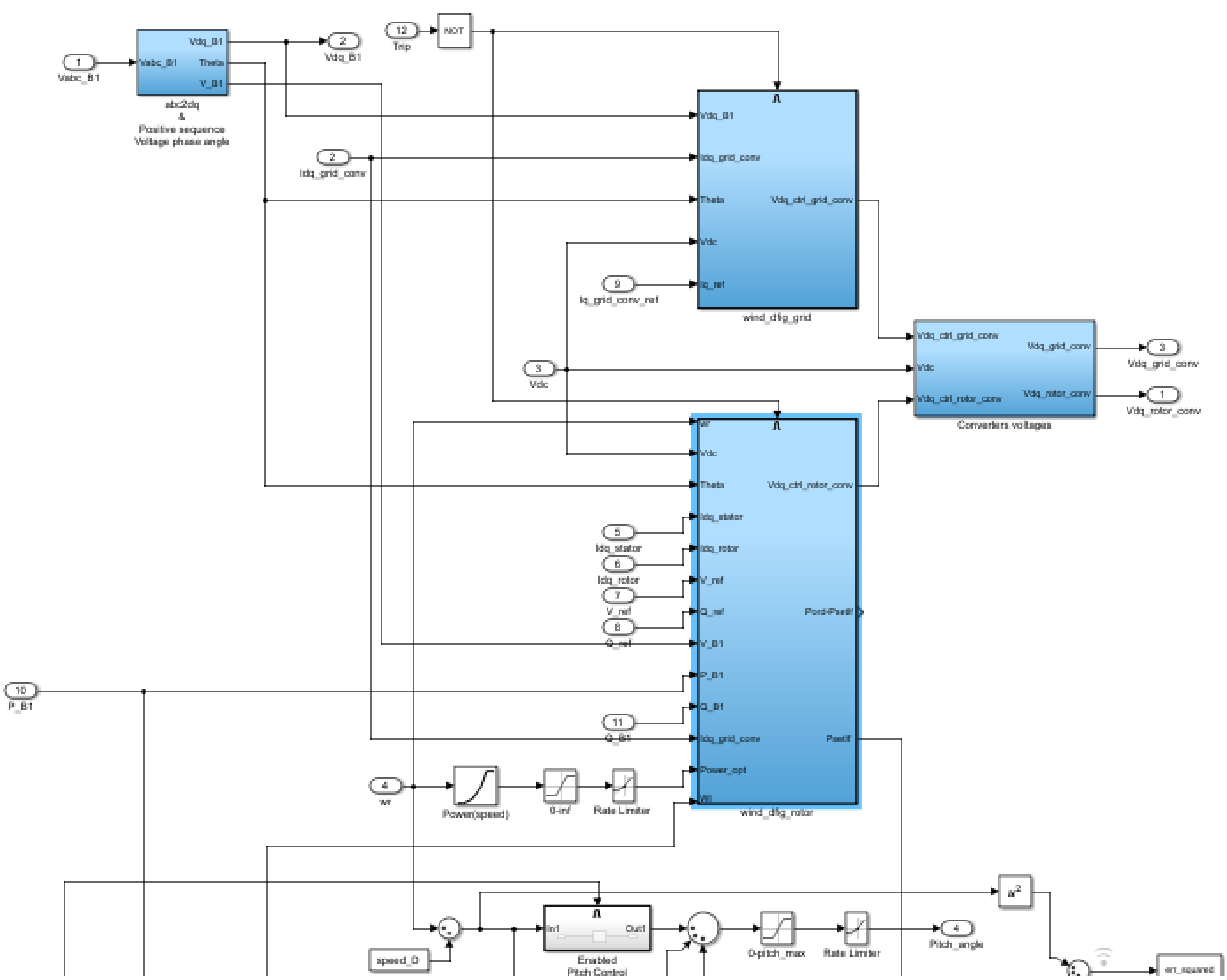


Fig. 22: Subsystem in the “Control” block

The “wind_dfig_rotor” block is the control system for the rotor-side converter system. It consists of a voltage regulator, reactive power regulator, and power regulator all controlled by PI controllers. This block has also been modified to output the dispatched power set-point to be used for the pitch compensator. Fig. 23 shows the implementation of the pitch control system. As can be seen the pitch system contains a pitch controller, pitch compensator, and the two part frequency sensitive controller which takes a frequency error as an input. This implementation matches what is outlined in Fig. 11.

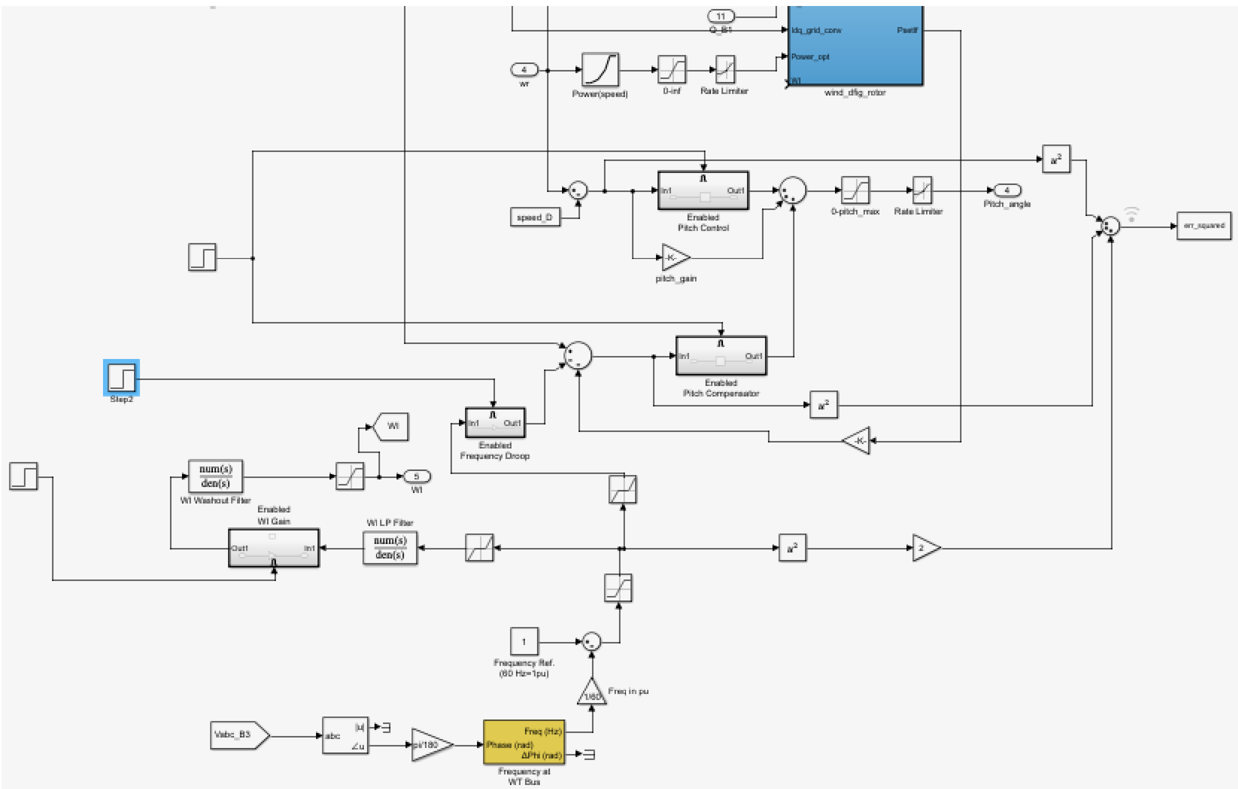


Fig. 23: Implemented Pitch Control System

Initial conditions are generated and loaded into the Matlab© workspace so that the Simulink© model starts in steady state with minimal transients due to the system having to “starting up”. Even when using an initial state vector there is a very brief transient in the first two seconds of the simulation. To prevent the pitch controllers from acting on this transient the controllers were designed as so called "enabled" controllers; which essentially means a signal is sent to the controller after the transients have passed to tell the controller to turn on. Underneath all of the enabled subsystem blocks is simply the P or PI controller. Each of the PI controllers is using anti-wind-up back-calculation methods and saturation limits, and the two filters associated with the WindINERTIA controller are modeled using transfer function blocks (Appendix A.8).

During the loss of generation event, the tripping of the small generator equates to a loss of about 3.6% of total generation. The loss of generation event takes place at 100 seconds into the simulation. The reason for this is because of the pesky transients at the very start of the simulation. Even though the transients are very brief, they are large. Making the pitch controls "enabled" helped to mitigate this problem but this solution would be troublesome to implement on the rest of the controls already designed into the Simulink© wind turbine model. At 100 seconds the system is truly in steady state and can then be disturbed by the loss of generation event.

4.4 NEED FOR CONTROLLER PARAMETER TUNING

Controller tuning refers to the selection of control parameters (PI coefficients) to ensure the best response of the controller. Choosing parameters that are too slow will cause the response of the system to be slow and sluggish, the controller will not handle upsets, and it will take too long to reach the set-point. However, choosing parameters that are too aggressive can cause the system to overshoot the set-point and/or become unstable [70]. The addition of the frequency sensitive controllers will change the dynamics of the system. In effect this means that even if the pitch controller and compensator were tuned to their ideal values prior to the addition of the frequency controllers, they will have to be re-tuned due to the change in system dynamics [70, 71]. The addition of the controllers will tend to push the system closer to instability, so careful selection of the parameter values must be ensured.

Due to the widespread industrial use of PID controllers there are several tuning techniques that are widely used such as Zeigler-Nichols, manual tuning, root locus analysis, etc. These techniques are useful because they are based on linear analysis methods which are very mature mathematically and can be applied easily to simple systems [70]. The drawbacks to the linear analysis method are that the system (plant) must be linearized in order to apply linear analysis techniques. If the equations describing the system are highly non-linear then the act of linearizing the system will cause much of the dynamics associated with the non-linear system to be lost. The Simulink© model that has been developed is a *very* non-linear and high fidelity simulation with many rich, interacting dynamics that will all be lost if linearized. In order to tune the parameters of the pitch control system without first linearizing the system other tuning methods outside the domain of linear analysis are needed.

4.5 GENETIC ALGORITHM

Parameter tuning can be framed in the context of an optimization problem, because this is, in essence, what we are trying to achieve: find an optimal set of control parameters that will minimize some operating criteria. Due to the high non-linearity of the system, heuristic optimization algorithms were considered.

The algorithm that was chosen is the Genetic Algorithm (GA). The reason this particular algorithm was chosen is because of its ability to efficiently approach global minimum in high dimensional search spaces. This algorithm can solve both constrained and unconstrained optimization problems and is based on natural selection, which is the driving force behind biological evolution [72, 73]. The genetic algorithm iteratively modifies a population of initially randomly generated individual solution. At each iteration the genetic algorithm uses a weighted random selection mechanism to choose individuals from the current population to be parents and uses them to produce the children for the next generation. Over successive iterations, the population "evolves" toward an optimal solution. The genetic algorithm can be used to solve many optimization problems that are not well suited for standard optimization algorithms, such as problems in which the objective function is stochastic or highly nonlinear [72-74].

In general the genetic algorithm uses three different rules to determine how the current population will evolve: selection rules, cross-over (mating) rules, and mutation rules. The selection rules choose the “chromosomes” (parents) in the population that will contribute to the population of the next generation. The mating rules determine how two chromosomes form children for the next generation. Finally, the mutation rules apply random mutations (changes) to chromosomes to form children for the next generation [72, 73].

Fig. 24 shows a visual interpretation of how all these terms are organized. In this work each gene in each chromosome is one parameter value that needs to be tuned. Each chromosome has eight parameters (genes): a proportional and integral parameter for each the pitch controller and compensator (four parameters), a proportional parameter for the governing/droop control (one parameter), a proportional parameter for WindINERTIA (one parameter), and a time constant parameter for each the low pass and washout filter in the WindINERTIA controller (two parameters). Fig. 24 also shows two how two chromosomes may perform a cross-over.

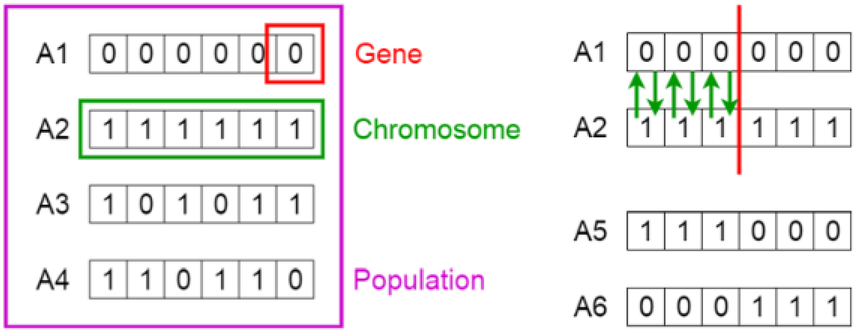


Fig. 24: Visual representation of a gene, chromosome and population, also showing cross-over [75]

4.6 Implementing the Genetic Algorithm for Controller Parameter Tuning

The genetic algorithm code that is used in this work has been tailored to accommodate a range of parameter values to be chosen as well as functionality that allows the genetic algorithm code written in Matlab© to interface and communicate with the Simulink© model. To see the full implementation of the code please see Appendix B.

The main body of the algorithm is contained in the GPid_con.m file. It first asks for user input to determine number of iterations (generations), population size, mutation rate, etc. as well as asking for a high and low value of each parameter. The high and low values for each parameter were determined by using a trial and error process: if the parameter ranges were set too high, instability manifests itself as uncontrolled oscillations in the response of the system. The parameters used for the inputs to the algorithm for each successful run are detailed in Appendix C as well as the solution and the associated optimal parameters.

Next, the algorithm generates a matrix of randomly generated chromosomes with each of the parameters (genes) being within their respective high and low limits designated by the user. After generating the initial population the Simulink© model and its corresponding initial conditions are loaded into memory. The initial condition were generated to ensure that the model starts in steady state, so the controllers will not act on any transients that may result should the system not start in steady state.

Now the code enters the main loop whose function is to modify and create each new generation of chromosomes at each iteration (generation). The first thing that happens is the algorithm calls a function named “costfunction.m.” This function is also a loop that computes the cost associated with each chromosome. It is prudent at this point to define the cost function that is used. In this optimization problem it is desirable to minimize the error that is being input into each controller, and it is by this metric that the cost function will be designed around. The Integral Squared Error (ISE) performance metric is used to build the cost function. In mathematical terms the cost function that needs to be minimized in this work is of the form:

$$min J = \int_0^t e_1^2(t) + e_2^2(t) + 2e_3^2(t)\, dt \qquad (10)$$

Where J is the cost function and $e_1(t), e_2(t)$, and $e_3(t)$ are the input errors into the pitch controller, pitch compensator, and both parts of the frequency controller respectively. The reason for the double weighting of $e_3(t)$ is because this particular error signal is the same signal that is being input into both parts of the proposed frequency controllers, therefore it makes sense to weight this signal doubly in order to make the algorithm more sensitive to minimizing this particular error signal.

During each iteration of the costfuntion.m loop each parameter of a single chromosome is assigned a variable name and using this variable name they are set as the actual parameter values for the controllers in Simulink©. Next the code runs the Simulink© model using the parameters set by the algorithm. The simulation has been designed so that as it runs the errors from the controllers are being summed and squared as in Equation 10. Once the simulation is complete, it outputs a time series vector of values to the Matlab© workspace. The code then integrates the ISE signal and sets this value equal to a cost that is associated with this particular chromosome (set of parameters). This loop will find the cost associated with each chromosome in the population and stores these values in a vector.

Now, the algorithm sorts the chromosomes by putting the ones with the lowest associated cost (best performing) at the top, the best performing chromosomes have a higher probability of surviving and passing on their genes. The algorithm calls the "pairing.m" function which is an implementation of selection rules that uses a top-down probability option to select the chromosomes that will contribute to the population at the next generation. Now that the appropriate chromosomes have been chosen the "matecon.m" function is called. This function takes the selected chromosomes and applies crossover (mating) rules to produce "children" that will replace some of the underperforming chromosomes in the initial population. Finally, the main loop calls the "mutatecon.m" function. This function is a loop that randomly selects parameters and randomly changes or "mutates" them to different values that are within their associated range. This is the end of one iteration (generation) of the main loop. The total number of times that the algorithm actually runs the Simulink© simulation is the number of iterations (100 in this work) multiplied by the population size for the initial generation (96 in this work). So for one run of the Genetic Algorithm in this work, the Simulink© simulation runs 9600 times, resulting in a simulation time of about 12 – 16 hours. The number of times the Simulink© model is run is the cause for such a long simulation time. As the main loop iterates and “evolves”, the cost associated with the best chromosome plotted against the generation number should be a monotonically decreasing function if the algorithm is functioning correctly. To see the genetic algorithm codes in their entirety please see Appendix B.

4.7 Case Studies

In order to observe how wind penetration affects the system and how well the controllers perform, they need to be compared to different cases. The base case is simply the KRK system with no wind generation. In the next three cases, a wind turbine of the same rating will replace one of the generators in area 2; this represents 25% wind penetration in the system. The three cases with wind penetration to be compared are: the wind turbine with pitch controller + pitch compensator, the wind turbine with pitch controller + pitch compensator + governing/droop control, and finally the wind turbine with pitch controller + pitch compensator + governing/droop control + WindINERTIA. The Genetic Algorithm is then run for each case that contains the wind turbine, meaning that the first, second, and third cases that contain wind penetration will have four, five, and eight, parameters, respectively, that need to be tuned. The solution (minimum) that the algorithm finds along with the associated optimal parameters can be found in Appendix C.

CHAPTER 5: DISCUSSION AND ANALYSIS

5.1 FREQUENCY RESPONSE & BLADE PITCH

When the Genetic Algorithm has run its course, optimal parameter values are generated. Using these generated parameters, various aspects of system performance are evaluated. First and foremost, is to inspect how the system frequency responds to a loss of generation event as shown in Fig. 25, while Table 3 documents the important frequency response characteristics. It can be immediately seen that the addition of wind does indeed negatively impact all three frequency response characteristics; most noticeably when there are no frequency sensitive controllers present. The addition of the droop (governing) controller shows a large gain in the settling frequency, so much so that it is only slightly less than the case with no wind generation. This controller can also be seen to improve the frequency nadir and RoCoF.

The addition of the WindINERTIA control makes a very small (almost negligible) impact on the settling frequency, but its main purpose is achieved by making further gains on improving the frequency nadir and RoCoF. It should be noted, however, that while the addition of the frequency sensitive controllers is better in all three aspects compared to not having these controllers in the wind turbine, the frequency nadir and RoCoF is still considerably poor compared to the case with no wind generation.

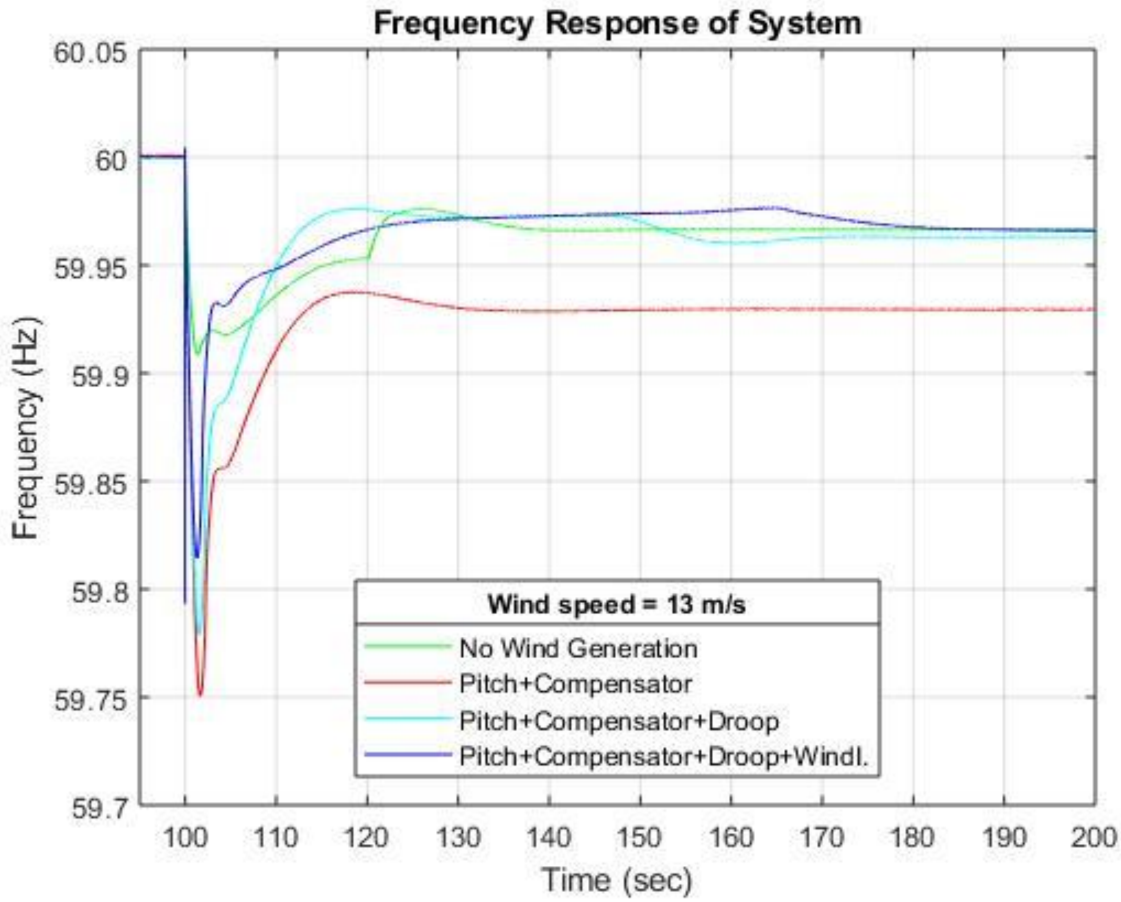


Fig. 25: Frequency Response of the system with 13 m/s wind speed

Table 3: Frequency Response Characteristics at wind speed 13m/s

Case	Freq. Nadir (Hz)	Settling Freq. (Hz)	RoCoF (Hz/s)
No Wind	59.9086	59.9666	-0.0687
Pitch+Comp	59.7505	59.9287	-0.1511
Pitch+Comp+Droop	59.7789	59.9634	-0.1430
Pitch+Comp+Droop+WindI.	59.8209	59.9655	-0.1344

Since these additional controllers modify the existing pitch control system, it is important we inspect how the pitch angle of the blades changes during an under frequency event, as shown in Fig. 26. When the frequency sensitive controllers are not present, it can be seen that the pitch of the blades remains relatively undisturbed, except for some small fluctuations that come from the pitch compensator detecting small fluctuations in the power set-point due to the other generating units having to pick up the extra generation. With the addition of the frequency sensitive controllers, two important aspects of the controller can be seen: de-loading prior to a frequency deviation, and pitch modification during a frequency deviation. In steady state, before the loss of generation event happens, the pitch angle of the blade is kept a few degrees higher when the frequency sensitive controllers are implemented.

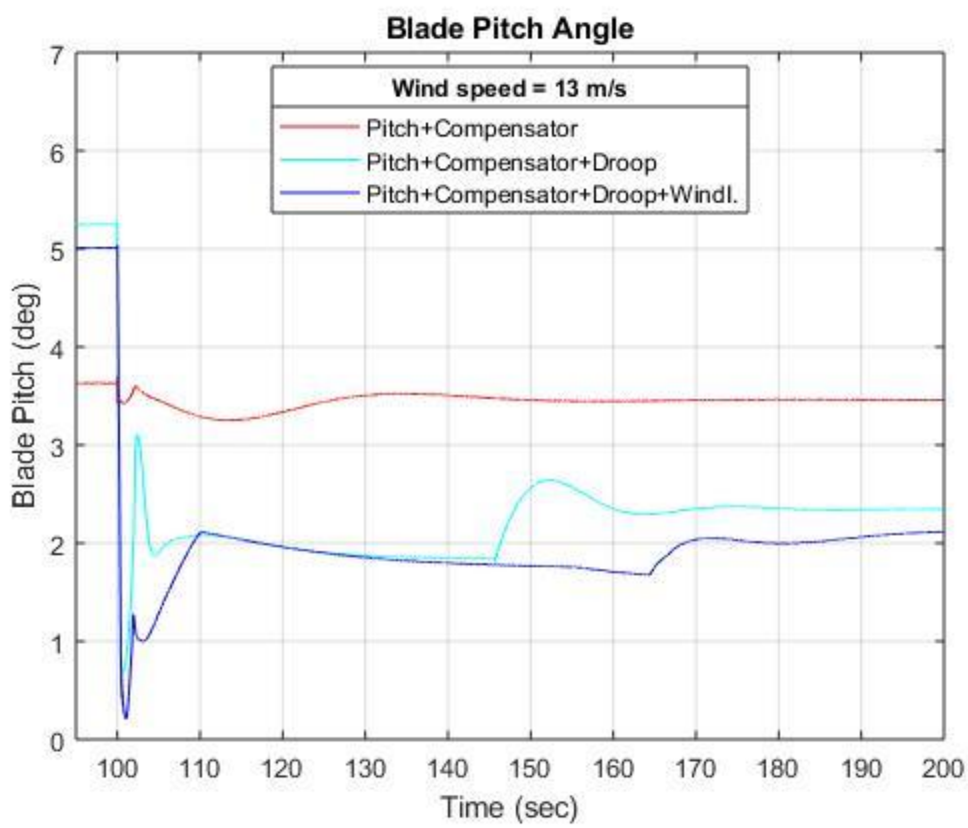


Fig. 26: Blade Pitch Angle after Loss of Generation Event with 13 m/s wind speed

The implications of this is that the wind turbine is producing less power before the loss of generation event but has a larger amount of power headroom (spinning reserve) to contribute to the system should there be a loss of generation event. In this manner, it is seen that the proposed controllers do indeed de-load the wind turbine's power output by modifying the pitch of the blade. When the loss of generation event does take place, the frequency of the system begins to decline and the frequency sensitive controllers respond by pitching the blades toward the wind which will increase the rotor speed and power output. During the loss of generation event the blades quickly pitch (10 degrees/sec slew rate [39]) to produce more power and the system frequency reaches a steady state value relatively quickly with few oscillations that are not large in magnitude. It can be seen that when introducing the WindINERTIA controller the overshoot of the pitch before reaching steady state is reduced. This is due to the fact that genetic algorithm chose parameter values that were slightly more aggressive in the case without WindINERTIA in order to minimize the input errors into the controllers. An interesting observation to note is that at around 145 seconds for the case without WindINERITA and at about 165 seconds for the case with WindINERTIA, the pitch seems to spike upwards and settle at a value slightly higher than what the pitch seemed to be asymptotically approaching in the time frame of about 110 – 140 seconds. This anomaly will be explained shortly (see Fig. 29).

It is also of interest to see what the frequency response of the system looks like when the wind speed is increased from 13 m/s to 15 m/s. The idea is that when there is more available wind power the wind turbine will be able to provide more power head room thus increasing the turbine's capability for providing frequency regulation. The frequency response of the system to a loss of generation event with a wind speed of 15 m/s is shown in Fig. 27. It was expected that the frequency nadir and RoCoF would be improved by increasing the wind speed, due to reasons mentioned above, but by comparing Fig. 25 and 27 it can be seen that the frequency responses are exactly the same! After a bit of thought, this seems to make sense, because even though the power head room is increased by having more available wind energy, the controller parameters are still fixed and respond at the same rate and still have the same dynamical response. While increasing the head room does not appear to affect how the wind turbine improves frequency stability, there is a benefit of increased headroom that can be deduced by looking at the pitch of the blades. This is shown in Fig. 28.

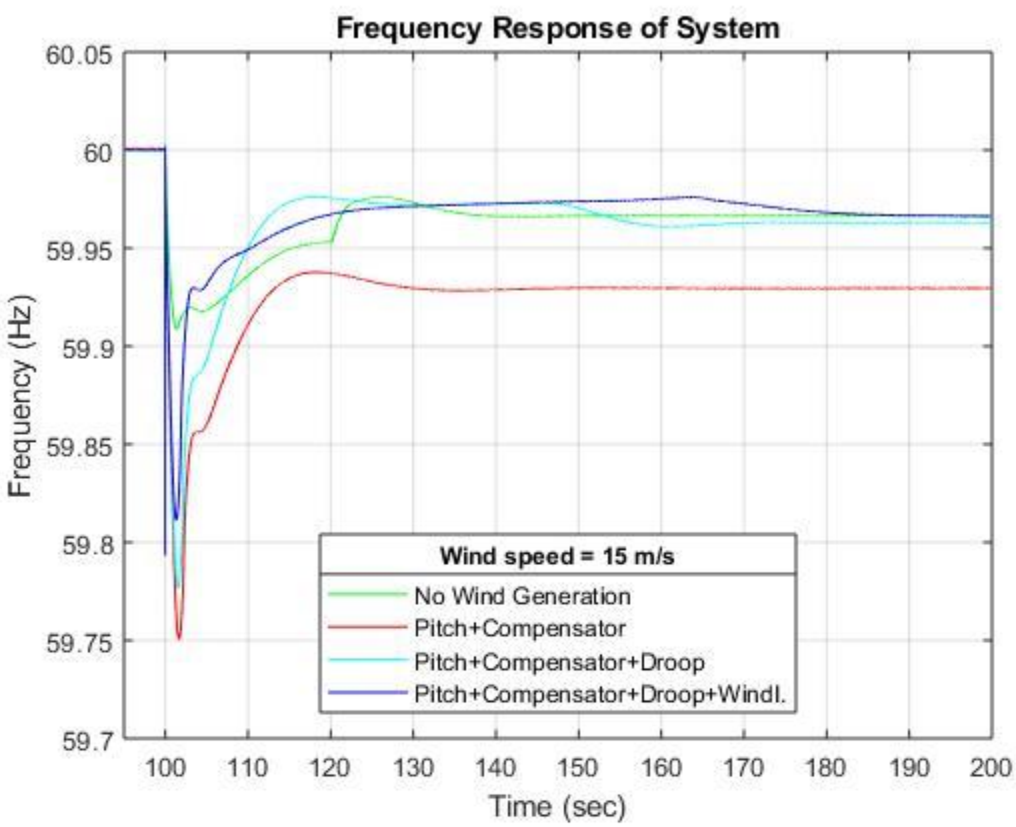


Fig. 27: Frequency Response of the system with 15 m/s wind speed

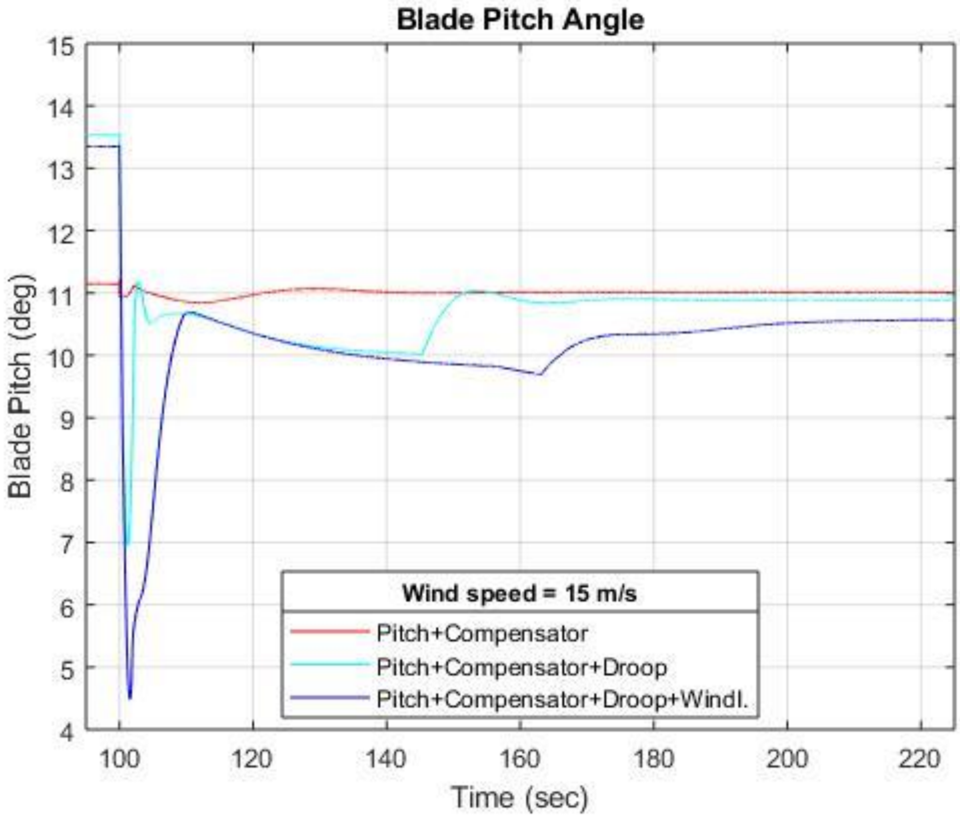


Fig. 28: Blade Pitch Angle after Loss of Generation Event with 15 m/s wind speed

The benefit in having this increased head room after the under frequency event comes from the fact that after a frequency deviation, once the pitch angle has settled to a steady-state value, that value will determine the amount of power headroom available. By comparing Fig. 26 and 28 it can be see that when there is more available wind power the pitch of the blades settles at a higher steady state-value after the under frequency event has occurred. This higher blade pitch angle, in essence, indicates that there is more available power head room for any further under frequency events that may occur in the future.

5.2 Back to De-Loaded Mode

As mentioned earlier, a strange anomaly occurs: at around 145 seconds for the case without WindINERITA, and at about 160 seconds for the case with WindINERTIA, the pitch seems to spike upwards and settle at a value slightly higher than what the pitch seemed to be asymptotically approaching in the time frame of about 110 – 140 seconds. By inspecting Fig. 29 some conclusion can be drawn. In Fig. 29 the rotor speed is being compared to the blade pitch angle in the case where all controllers are present and at a wind speed of 15 m/s. Recall that the input to the pitch controller in an error signal that is generated by comparing the actual rotor speed to the speed reference value, which is 1.2 p.u. in this work (Appendix A.5). In Fig. 29 it can be seen that when the rotor speed reaches a speed of 1.2 p.u. the pitch control system will take action and not allow the rotor speed to exceed 1.2 p.u. In the case where the frequency controllers are not present, the rotor speed will stay at a constant 1.2 p.u. in order to produce the maximum amount of power. In the case where the frequency sensitive controllers are present, once the rotor speed reaches the reference value of 1.2 p.u. the frequency sensitive controllers have the effect of pitching the blades back to set the wind turbine back into its de-loaded mode. It is also interesting to note that the rate at which the rotor speed approaches its reference value is on the order of the time frame that the primary frequency regime acts in, which is roughly 10 – 60 seconds after an under frequency event. This type of behavior is actually very advantageous due to the fact that once the primary frequency regime is over the AGC (secondary control) takes over to restore the frequency to nominal, while the wind turbine is setting itself back to a de-loaded mode. The effect of the wind turbine coming back into the de-loaded mode can also be seen in the frequency responses of Fig. 25 and 27. At the instance of time that the turbine starts to enter its de-loaded mode, the frequency of the system can be seen to dip slightly. This is due to the fact that the wind turbine is producing less power (generation) thus lowering the settling frequency slightly. In Table 3 the settling frequency values for the case with and without WindINERTIA are actually the values for when after the turbine starts to go into its de-loaded mode again. So, if the settling frequency was determined before this happened it would actually show improvement over the case with no wind generation.

Given that the pitching of the blades due to a loss of generation event changes the amount of power generated by the wind turbine, it is of interest to inspect how the wind turbine output power compares to the output power of conventional generating units.

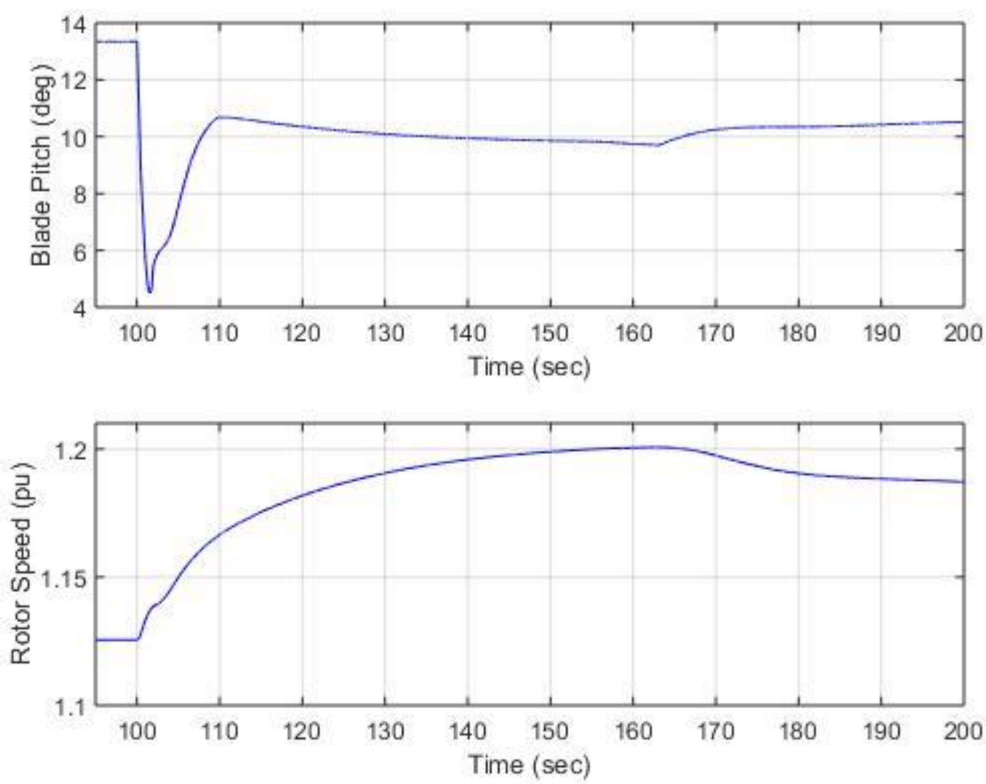


Fig. 29: Rotor Speed vs. Blade pitch with all controls at 15 m/s wind speed

5.3 POWER OUTPUT

Fig. 30 and 31 show the power output of the conventional generating units and the wind turbine without and with the addition of both parts of the frequency sensitive controller, respectively. By observing Fig. 30, it can be seen that before the loss of generation event all the generators, including the wind turbine, are all producing the proper (and equal) amount of power to satisfy the load demands. Once the loss of generation event occurs, it can be seen that the wind turbine does not produce any additional power but instead simply keeps producing the maximum amount of power it can gather from the wind. Since the wind turbine does not pick up any of the extra generation, the other three conventional generation units pick up approximately 2 MW extra each.

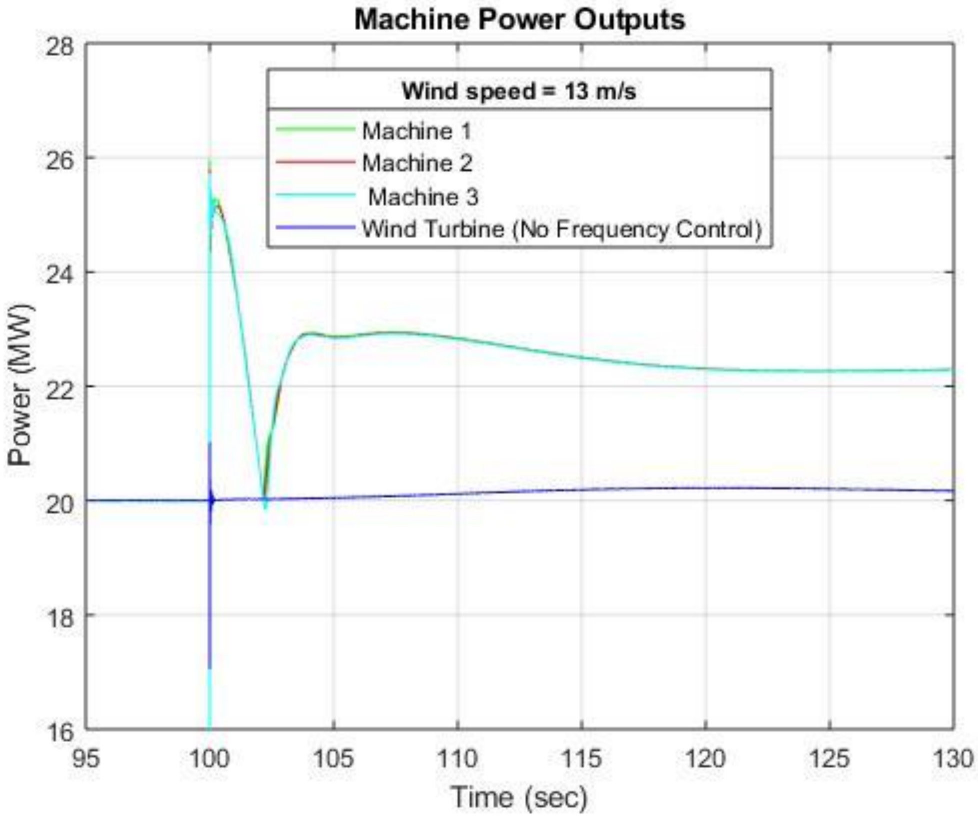


Fig. 30: Power Outputs of the Wind Turbine and Conventional Generators: No Frequency Control at Wind Speed of 13 m/s

Now looking at Fig. 31 -- the case where all the frequency sensitive controllers are implemented -- shows a different picture. The first thing to notice is that the wind turbine is only producing 18 MW of power compared to the 20 MW in the previous case. This is due to the fact that the wind turbine is acting in a de-loaded mode, so its power output is less than the power available in the wind. Due to this it can be see that the other generating units are picking up some of this slack. Once the loss of generation event occurs, the conventional generating units adjust their power as expected but the wind turbine also adjusts its power output to pick up some of the extra demand (power) needed. This is more evidence that the addition of the frequency sensitive controllers is performing as intended: in the 60 seconds that the primary frequency control is taking place the outputs of the wind turbine and the conventional generating units approach the same power output, that is, until the wind turbine re-enters its de-loaded mode about 60 seconds after the loss of generation event.

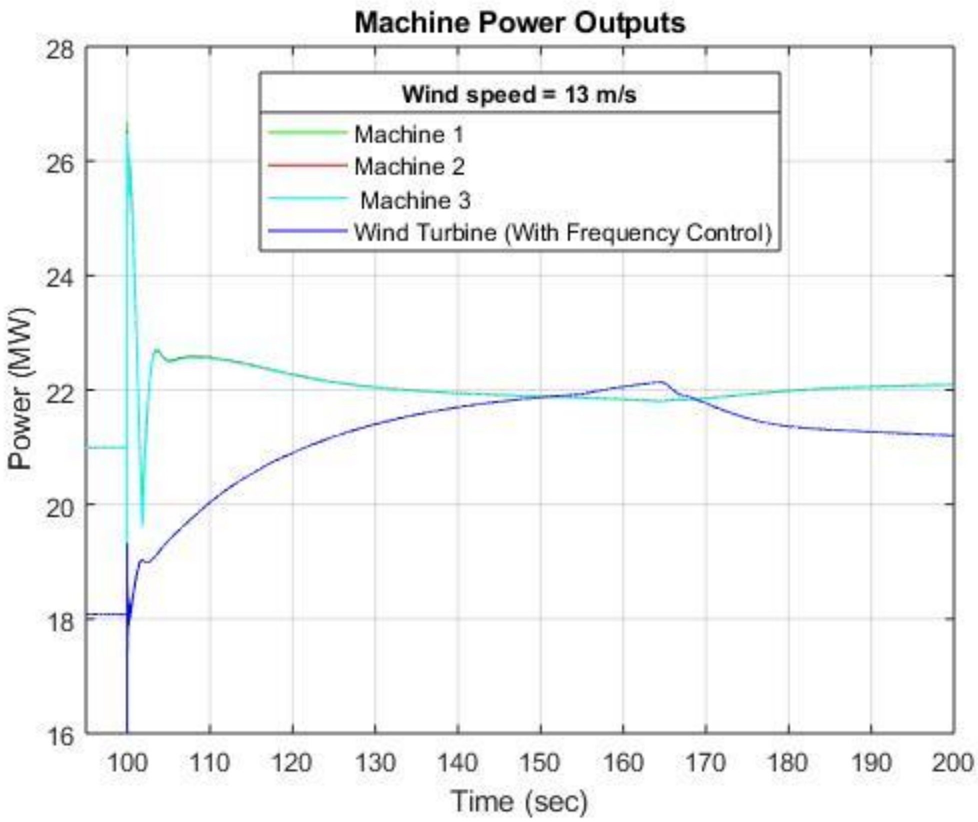


Fig. 31: Power Outputs of the Wind Turbine and Conventional Generators: All Controls at Wind Speed of 13 m/s

By inspecting Fig. 31 it can also be seen that in about the first few seconds after the loss of generation event the rate of power output is quite a bit higher than the rate of output power in the next 60 seconds or so. This is due to the action of the WindINERTIA controller. The output of the WindINERTIA controller is shown in Fig. 32. Examining this figure shows that once the loss of generation event occurs the controller very briefly injects a considerable amount of power directly into the power output of the wind turbine. This brief injection of extra power is what causes the power output to increase rapidly for a brief period of time; this is the idea behind this controller being a rate control. The faster the wind turbine can initially inject power into the grid helps to improve the frequency nadir and RoCoF. Fig. 32 also shows that the washout filter that is present in the WindINERTIA controller is effective at passing only transient inputs.

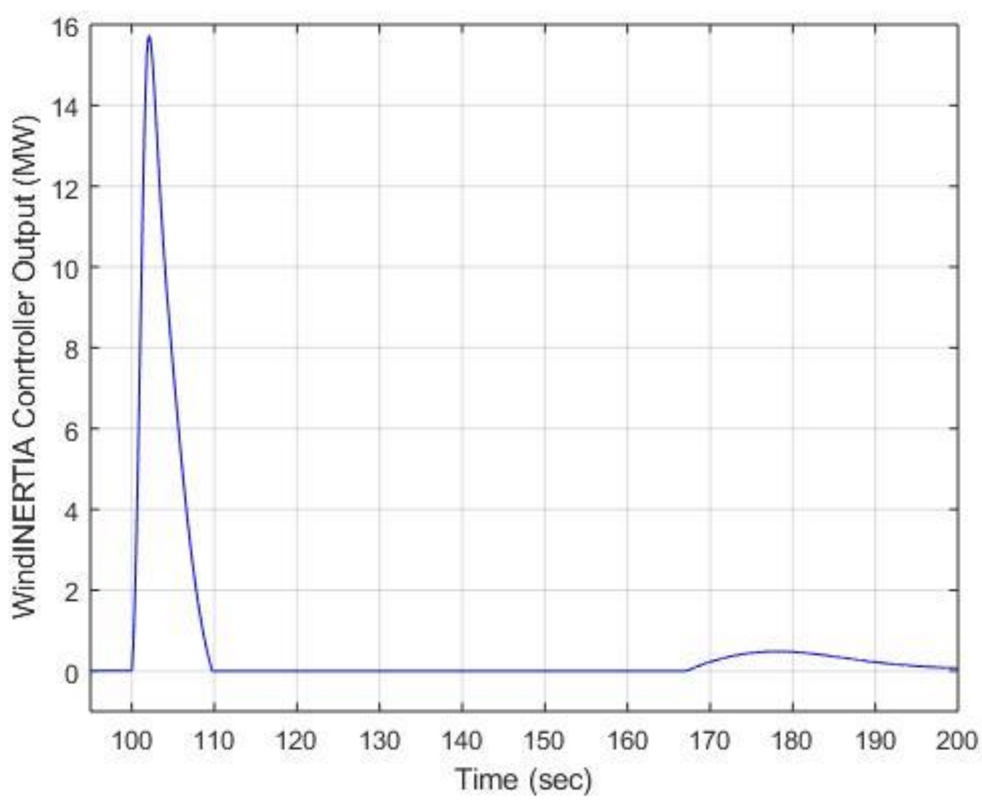


Fig. 32: Output of the WindINERTIA Controller

5.4 GENETIC ALGORITHM BEHAVIOR

Now that it has been shown that the implementation of the frequency controllers perform as expected, how can it be known that the genetic algorithm actually found the optimal parameters? By inspecting Fig. 33 some conclusions can be drawn. What Fig. 33 shows is a plot of the cost associated with the best chromosome (set of controller parameter values) of each generation (iteration) vs. that generation. The main clue that the algorithm is performing as intended is that the resulting plot is monotonically decreasing, this is important because this shows that the algorithm is always approaching a minimum value. The population may "stagnate" for several generations indicated by the straight horizontal lines, until after some time the algorithm "evolves" towards better solutions.

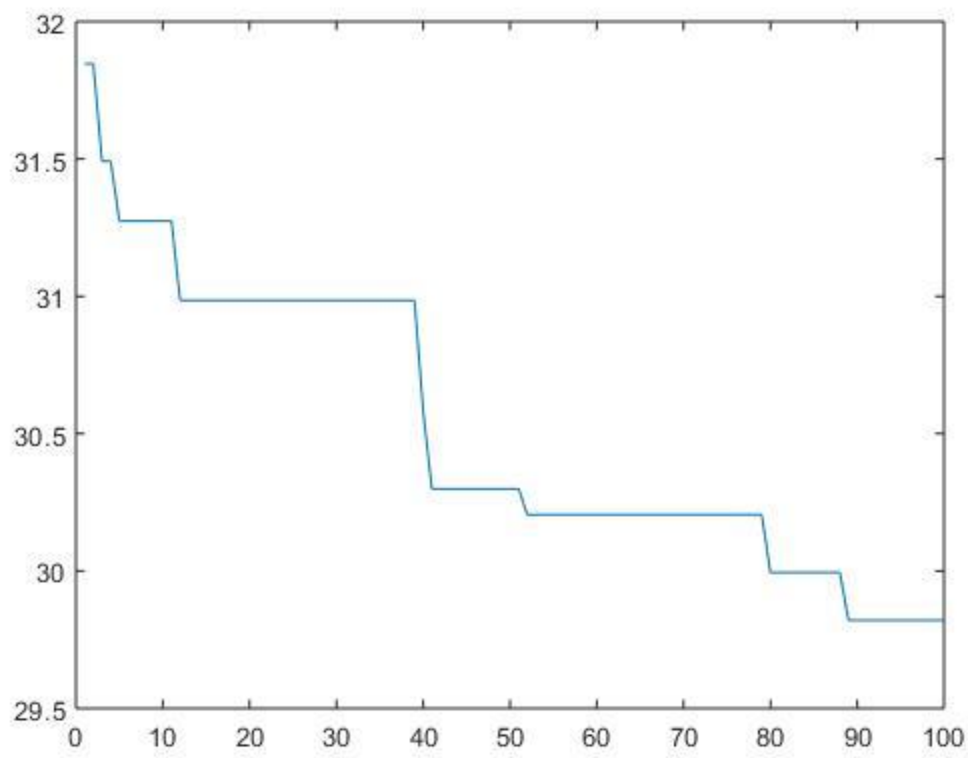


Fig. 33: Output of a Genetic Algorithm run showing iterations (generations) vs. cost

In attempting to make sense of the value of the cost associated with each iteration, it can be thought of as follows: the algorithm is essentially searching through a nine-dimensional search space, with the first eight dimensions relating to the eight parameters needing to be optimized; these eight parameters values form a surface in a nine-dimensional space where the minimum cost is the value on the surface in the 9^{th} dimension. In a sense, the value of the cost is almost arbitrary due to the fact that this nine-dimensional surface is impossible to visualize and can have a minimum that is truly around the 30 value shown in Fig. 33. The important factor, however, is the fact that the cost is a monotonically decreasing function.

There are some drawbacks to the use of a genetic algorithm in this particular application. To know when the algorithm has reached a true optimum, the cost must stay at a steady-state value for a significant number of iterations. This was difficult to achieve in this work due to the fact that in order to run only 100 iterations, the algorithm the run time was on the order of 12 to 16 hours. By looking at the sample Genetic Algorithm run in Fig. 33, it shows that the cost stagnates for about 10 generations before ending. In order to feel comfortable that the algorithm has found a true optimum it would be ideal for the simulation to run for several hundred more generations to truly see when the cost reaches its minimum value. Due to the extremely long simulation time for only 100 iterations, several hundred iterations were not attempted due to time and computer memory constraints.

6 CONCLUSIONS

6.1 NOVELTIES & CONTRIBUTIONS

A majority of the material presented is not new; the proposed frequency controllers have been studied in [67, 76-77]. In all of the studies involving enabling frequency control of wind turbines, the systems have been studied through the use of linear analysis methods. A novel contribution of this work is the use of a high fidelity simulation that is studied using methods outside the realm of linear analysis by employing a Genetic Algorithm to tune the controller parameters. These particular controllers have never been attempted to be tuned through the use of a Genetic Algorithm, either in a linearized or non-linear system. While the Genetic algorithm has been used to tune controller parameters in wind turbine before [78-79] they are typically concerned on how the tuned parameters affect the wind turbine as an isolated system that is not connected to the grid. In other words, they are attempting to optimize the wind turbine as an isolated system rather that optimizing its response to certain events that occur in the power grid such as frequency deviations and faults. It was shown that it is indeed possible to study the frequency response of the power grid with high wind penetration levels, while not linearizing the system, and tuning the proposed pitch controllers with a Genetic Algorithm.

6.2 FUTURE WORK & DRAWBACKS

Due to the flexible nature of Matlab/Simulink© development environment, this model can be easily modified and added on to in order to perform further investigation. For instance in this work only 25% wind penetration was tested and it would be interesting to see how the system responds with a higher level of wind penetration, up to 50%. With the addition of higher wind penetration levels more realistic wind profiles could be used. In this work the wind speeds were of a constant value and by using realistic wind profiles modeled by the Weibull distribution [80] or by using real wind data, the properties of the controllers could be further studied.

Using the KRK system as a test system is advantageous for testing and tuning the controllers due to its simple nature and ability to accurately simulate frequency responses for large systems. Once the controllers are tuned in the KRK system the parameters can be applied to wind turbines that are replacing conventional generation units in more realistic systems such as the 39-bus, 10-machine New-England reduced system [81].

As mentioned previously, only the parameters associated with the pitch control system were tuned while the other control system parameters (various power, voltage, and current regulators) in the wind turbine were left at a set value. The code would be simple enough to modify those in order to tune all of the control parameters associated with the wind turbine control. This would produce a truly optimal wind turbine control set up. In order to do this however, a more systematic way of choosing the appropriate high and low ranges for all the parameter values would be essential in order for the system to remain stable for all combination of parameter values. This implementation would work particularly well for the genetic algorithm due to its ability to search through high dimensional spaces of cost functions that can be highly non-linear and stochastic. Although, there is the drawback: in order to search such a large space many, many iterations of the algorithm would need to be run. With the simulation in its current state this could take on the time order of several days to a week to complete a single run. More computing power and perhaps fine tuning of the model to make it quicker and more efficient would help with the simulation time.

While the controllers studied in this work were simple PI controllers (only Proportional control in the case of the frequency sensitive controllers) it would be interesting to see the effects of adding different types controllers such as neuro-fuzzy, state-space, neural network, etc. Other types of intelligent optimization algorithms can also be considered such as a particle swarm algorithm, simulated annealing, or ant colony optimization. These algorithms work in a similar manner to the Genetic Algorithm in the fact that they all emulate certain physical systems that naturally find their way to optimum values through some element of random chance.

In most work that focuses on controller design, the derivation and implementation of a working model is usually the most time consuming aspect. With the highly flexible design of the Simulink© model presented in this work there will be little work and time that needs to be expended on creating a dynamic model from scratch. Since the model has been built and tested, the ability to expand and extend this work should be relatively simple.

REFERENCES

[1] *Global Wind Report 2017*. [Online]. Available: http://files.gwec.net/register?file=/files/GWR2017.pdf. [Accessed: 24-Mar-2019].

[2] "2017 Wind Technologies Market Report," *Energy.gov*. [Online]. Available: https://www.energy.gov/eere/wind/downloads/2017-wind-technologies-market-report. [Accessed: 24-Mar-2019].

[3] "NERC IVGTF Task 2.4 Report- Operating Practices, Procedures, and Tools," NERC, Tech. Rep., NERC, 2011.

[4] E. Muljadi, V. Gevorgian, M. Singh, and S. Santoso, "Understanding inertial and frequency response of wind power plants," in *2012 IEEE Power Electronics and Machines in Wind Applications*, 2012.

[5] E. Welfonder, "Least-cost dynamic interaction of power plants and power systems," in *Control Engineering Practice*, vol. 5, no. 9, pp. 1203–1216, 1997.

[6] J. Ekanayake and N. Jenkins, "Comparison of the response of doubly fed and fixed-speed induction generator wind turbines to changes in network frequency," in *IEEE Trans. Energy Convers.*, vol. 19, no. 4, pp. 800–802, Dec. 2004.

[7] J. Morren, S. W. De Haan, W. L. Kling, and J. Ferreira, "Wind turbines emulating inertia and supporting primary frequency control," in *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 433–434, Feb. 2006.

[8] J. M. Mauricio, A. Marano, A. Gómez-Expósito, and J. L. Martinez-Ramos, "Frequency regulation contribution through variable-speed wind energy conversion systems," in *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 173–180, Feb. 2009.

[9] G. Lalor, A. Mullane, and M. O'Malley, "Frequency control and wind turbine technologies," in *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp.1905–1913, Nov. 2005.

[10] M. Kayikçi and J. V. Milanović, "Dynamic contribution of DFIG-based wind plants to system frequency disturbances," in *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 859–867, May 2009.

[11] D. Gautam, L. Goel, R. Ayyanar, V. Vittal, and T. Harbour, "Control strategy to mitigate the impact of reduced inertia due to doubly fed induction generators on large power systems," in *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 214–224, Feb. 2011.

[12] A. Mullane and M. O'Malley, "Modifying the inertial response of power-converter based wind turbine generators," in *Proc. 3rd IET Int. Conf. Power Electronics, Machines and Drives,* 2006.

[13] L. N. Papangelis and C. D. Vournas, "Design principles of wind turbine inertia emulators for autonomous systems," in *Preprints 19th World Congr.,* 2014, pp. 5844–5851.

[14] J. Aho, A. Buckspan, J. Laks, P. Fleming, Y. Jeong, F. Dunne, M. Churchfield, L. Pao, and K. Johnson, "A tutorial of wind turbine control for supporting grid frequency through active power control," in *Proc. Amer. Control Conf. (ACC),* 2012, 2012, pp. 3120–3131.

[15] R. G. De Almeida and J. A. Peças Lopes, "Participation of doubly fed induction wind generators in system frequency regulation," in *IEEE Trans. Power Syst.,* vol. 22, no. 3, pp. 944–950, Aug. 2007.

[16] E. Loukarakis, I. Margaris, and P. Moutis, "Frequency control support and participation methods provided by wind generation," in *Proc. 2009 IEEE Electrical Power & Energy Conf. (EPEC),* 2009, pp. 1–6.

[17] E. Muljadi, V. Gevorgian, M. Singh, and S. Santoso, "Understanding inertial and frequency response of wind power plants," in *Proc. IEEE Symp. Power Electronics and Electric Machines in Wind Applications,* 2012, pp. 1–8.

[18] B. H. Chowdhury and H. T. Ma, "Frequency regulation with wind power plants," in *Proc. 2008 IEEE Power and Energy Soc. General Meeting—Conversion and Delivery of Electrical Energy in the 21st Century,* 2008, pp. 1–5.

[19] P.-K. Keung, P. Li, H. Banakar, and B. T. Ooi, "Kinetic energy of wind-turbine generators for system frequency support," in *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 279–287, Feb. 2009.

[20] G. Ramtharan, N. Jenkins, and J. Ekanayake, "Frequency support from doubly fed induction generator wind turbines," in *IET Renew. Power Gener.*, vol. 1, no. 1, pp. 3–9, 2007.

[21] A. Teninge, C. Jecu, D. Roye, S. Bacha, J. Duval, and R. Belhomme, "Contribution to frequency control through wind turbine inertial energy storage," in *IET Renew. Power Gener.*, vol. 3, no. 3, pp. 358–370, 2009.

[22] I. Erlich and M. Wilch, "Primary frequency control by wind turbines," in *Proc. 2010 IEEE Power and Energy Soc. General Meeting*, 2010, pp. 1–8.

[23] L. Holdsworth, J. Ekanayake, and N. Jenkins, "Power system frequency response from fixed speed and doubly fed induction generator- based wind turbines," *Wind Energy*, vol. 7, no. 1, pp. 21–35, 2004.

[24] ERCOT Nodal Protocols April 2, 2013, ERCOT.

[25] Wind Turbines Connected to Grids with Voltages Above 100 KV-Technical Regulation for the Properties and the Regulation of Wind Turbines, Elkraft System and Eltra Regulation, 2004.

[26] E. Fagan, S. Grimes, J. McArdle, P. Smith, and M. Stronge, "Grid code provisions for wind generators in ireland," in *Proc. 2005 IEEE Power Eng. Soc. General Meeting*, 2005, pp. 1241–1247.

[27] M. Altın, O. Goksu, R. Teodorescu, P. Rodriguez, B.-B. Jensen, and L. Helle, "Overview of recent grid codes for wind power integration," in *Proc. 2010 12th Int. Conf. Optimization of Electrical and Electron. Equipment (OPTIM)*, 2010, pp. 1152–1160.

[28] B. Kirby, M. Milligan, and E. Ela, "Providing minute-to-minute regulation from wind plants," in *Proc. 9th Annu. Large-Scale Integration of Wind Power into Power Systems and Transmission Networks for Offshore Wind Power Plant*, 2010.

[29] Eriksson, Sandra, Hans Bernhoff, and Mats Leijon, "Evaluation of different turbine concepts for wind power," in *Renewable and Sustainable Energy Reviews* 12.5 (2008): 1419- 1434.

[30] N. Goudarzi, and W. Zhu, “A review on the development of wind turbine generators across the world,” in *International Journal of Dynamics and Control*, vol. 1, no. 2, pp. 192-202, 2013.

[31] B. Wu, *Power conversion and control of wind energy systems*. Piscataway, NJ: IEEE Press, 2011.

[32] M. R. Islam, Y. Guo, and J. Zhu, “A review of offshore wind turbine nacelle: technical challenges and research and developmental trends,” in *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 161-176, 2014.

[33] Bhukya, Jawaharlal, and Vasundhara Mahajan, "The controlling of the DFIG based on variable speed wind turbine modeling and simulation," in *2016 IEEE 6th International Conference on Power Systems (ICPS)*. IEEE, 2016.

[34] H. Li, F. Li, Y. Xu, D. T. Rizy, and J. D. Kueck, "Interaction of multiple distributed energy resources in voltage regulation," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century,* 2008 IEEE, 2008, pp. 1-6.

[35] G. M. Masters, *Renewable and efficient electric power systems*. Nueva Jersey (Estados Unidos): John Wiley &son, 2013.

[36] G. Ofualagba and E. U. Ubeku, "Wind energy conversion system- wind turbine modeling," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, 2008 IEEE, 2008, pp. 1-8.

[37] Rashad M. Kamel, A. Chaouachi and Ken Nagasaka, “Enhancement of micro-grid performance during islanding mode using storage batteries and new fuzzy logic pitch angle controller,” in *Energy Conversion and Management.* 52 (2011) 2204-2216.

[38] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: theory, design and application*. Chichester: John Wiley & Sons, 2011.

[39] M. Shao, N. Miller, J. Sanchez-Gasca, and J. MacDowell, “Modeling of GE Wind Turbine-Generator for Grid Studies, Version 4.6,” in *Tech. Rep., General Electric International, Inc.,* 2013.

[40] Boukhezzar, Boubekeur, and Houria Siguerdidjane, "Nonlinear control of variable speed wind turbines for power regulation," in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*. IEEE, 2005.

[41] Pena R., Clare J.C., Asher G.M., “A doubly fed induction generator using back-to-back PWM converters supplying an isolated load from a variable speed wind turbine,” in *IEEE Proc. Electr. Power Appl.*, 1996, 143, (5), pp. 380–387

[42] I. Boldea, *The electric generators handbook*. Boca Raton: CRC/Taylor & Francis, 2006.

[43] Liao Y., Putrus G.A., Smith K.S., “Evaluation of the effects of rotor harmonics in a doubly-fed induction generator with harmonic induced speed ripple,” in *IEEE Trans. Energy Convers.*, 2003, 18, (4), pp. 508–515

[44] Cadirci I., Ermism., “Commutation angle analysis of a double output induction generator operating in sub- and supersynchronous modes,” in *Proc. Seventh IEEE Mediterranean Electrotechnical Conf.*, 1994, vol. 2, pp. 793–796

[45] Seman S., Niiranen J., Kanerva S., Arkkio A., “Analysis of a 1.7 MVA doubly fed wind-power induction generator during power system disturbances,” in *Proc. Nordic Workshop on Power and Industrial Electronics*, June 2004

[46] Boldea I., “Control of electric generators: a review,” in *IEEE 26th Annual Industrial Electronics Society Conf.,* 2003, vol. 1, pp. 972–980

[47] Hoffman W., Okafor F., “Optimal power utilization with doubly fed full controlled induction generator,” in *Proc. Sixth IEEE Africon Conf.*, October 2002, vol. 2, pp. 693–698

[48] Ramos C.J., Martins A.P., Araujo A.S., Carvalho A.S., “Current control in the grid connection of the double-output induction generator linked to a variable speed wind turbine,” in *IEEE 28th Annual Conf.*, 2002, vol. 2, pp. 979–984

[49] Peterson A., Lundberg S., Thiringer T., "A DFIG wind turbine ride through system influence on the energy production," in *Nordic Wind Power Conf.*, March 2004, pp. 1–7

[50] Zhi D., Xu L., "Direct power control of DFIG with constant switching frequency and improved transient performance," in *IEEE Trans. Energy Convers.*, 2007, 22, (1), pp. 110–118

[51] Xu L., Cartwright P., "Direct active and reactive power control of DFIG for wind energy generation," in *IEEE Trans. Energy Convers.*, 2006, 21, (3), pp. 750–758

[52] Bozhko S.V., Gimnez R.B., Li R., Clare J.C., Asher G.M., "Control of offshore DFIG-based wind farm grid with linecommutated HVDC connection," in *IEEE Trans. Energy Convers.*, 2007, 22, (1), pp. 71–78

[53] Pena R., Crdenas R., Escobar E., Clare J., Wheeler P., "Control system for unbalanced operation of stand-alone doubly fed induction generators," in *IEEE Trans. Energy Convers.*, 2007, 22, (2), pp. 544–545

[54] "Wind Turbine Doubly-Fed Induction Generator (Phasor Type)," *Implement phasor model of variable speed doubly-fed induction generator driven by wind turbine - Simulink.* [Online]. Available: http://www.mathworks.com/help/physmod/sps/powersys/ref/windturbinedoublyfedinductiongeneratorphasortype.html. [Accessed: 30-Mar-2019].

[56] Prepared by NERC RS Committee. *Balancing and Frequency Control: A Technical Document Prepared by the NERC Resources Subcommittee*. NERC Technical Report, 2011.

[57] P. Kundur, *Power System Stability and Control*. New York, NY, USA: McGraw-Hill, 1994, vol. 7.

[58] J. W. Ingleson and E. Allen, "Tracking the Eastern interconnection frequency governing characteristic," in *Proc. 2010 IEEE Power and Energy Soc. General Meeting*, 2010, pp. 1–6.

[59] S. Sharma, S.H. Huang, and N. D. R. Sarma, “System inertial frequency response estimation and impact of renewable resources in ERCOT interconnection,” in *Proc. 2011 IEEE Power and Energy Soc. General Meeting*, 2011, pp. 1–6.

[60] Rate of Change of Frequency DS3 Advisory Council Discussion Paper, EIRGRID and SoNi, 2013.

[61] Pourbeik, Pouyan, et al. "Providing primary frequency response from photovoltaic power plants," in *Proc. CIGRE Symp.*. 2017.

[62] F. Wilches-Bernal, J. J. Sanchez-Gasca, and J. H. Chow, “Implementation of wind turbine generator models in the power system Toolbox,” in *Proc. 2014 IEEE Power and Energy Conf. Illinois (PECI)*, 2014, pp. 1–5.

[63] M. Shao, N. Miller, J. Sanchez-Gasca, and J. MacDowell, “Modeling of GE Wind Turbine-Generator for Grid Studies, Version 4.6,” in *Tech. Rep., General Electric International, Inc.*, 2013.

[64] Hassouneh, Munther A., Hsien-Chiarn Lee, and Eyad H. Abed, "Washout filters in feedback control: Benefits, limitations and extensions," in *Proceedings of the 2004 American control conference*, Vol. 5. IEEE, 2004.

[65] P. Kundur, *Power System Stability and Control*. New York, NY, USA: McGraw-Hill, 1994, vol. 7.

[66] M. Klein, G. J. Rogers, and P. Kundur, “A fundamental study of interarea oscillations in power systems,” in *IEEE Trans. Power Syst.*, vol. 6, no. 3, pp. 914–921, Aug. 1991.

[67] Wilches-Bernal, Felipe, Joe H. Chow, and Juan J. Sanchez-Gasca, "A fundamental study of applying wind turbines for power system frequency control," in *IEEE Transactions on Power Systems* 31.2 (2016): 1496-1505.

[68] “Simscape Power Systems,” *MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/products/simpower.html.html. [Accessed 30-Mar-2019].

[69] Mathworks.com. (2019). *Simscape ElectricalDocumentation*. [Online] Available: https://www.mathworks.com/help/physmod/sps/ [Accessed 24-Mar-2019].

[70] A. O'Dwyer, Handbook of PI and PID controller tuning rules, World Scientific, 2009.

[71] J. C. Basilio, S. R. Matos, “Design of PI and PID controllers with transient performance,” *IEEE Transactions on Education,* Vol. November (2002), pp. 364-370.

[72] Poli, Riccardo, et al. *A field guide to genetic programming*. Lulu. com, 2008.

[73] *How the Genetic Algorithm Works.* [Online]. Available: https://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html. [Accessed: 10-Mar-2019].

[74]Pei-Jin Wang, “PI-tuning methods based on GA,” *in International Conference on Machine Learning and Cybernetics*, IEEE, 2002, pp. 544-547.

[75] Vijini Mallawaarachchi, “Introduction to Genetic Algorithms - Including Example Code,” *Towards Data Science*, 08-Jul-2017. [Online]. Available: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3. [Accessed: 24-Mar-2019].

[76] Y. Wang, G. Delille, H. Bayem, X. Guillaud, and B. Francois, “High wind power penetration in isolated power systems–Assessment of wind inertial and primary frequency responses,” in *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2412–2420, Aug. 2013.

[77] Wilches-Bernal, Felipe, Joe H. Chow, and Juan J. Sanchez-Gasca, "Power system frequency control using Type-4 wind turbine generators," in *2017 North American Power Symposium (NAPS)*. IEEE, 2017.

[78] Zhao, M., Zhe Chen, and Frede Blaabjerg, "Optimisation of electrical system for offshore wind farms via genetic algorithm." in *IET Renewable Power Generation* 3.2 (2009): 205-216.

[79] Eidiani, Mostafa, Natan Asghari Shahdehi, and Hossein Zeynal, "Improving dynamic response of wind turbine driven DFIG with novel approach," in *2011 IEEE Student Conference on Research and Development*. IEEE, 2011.

[80] R. Kollu, S. R. Rayapudi, S. Narasimham, and K. M. Pakkurthi, “Mixture probability distribution functions to model wind speed distributions,” in *Int. J. Energy Environ. Eng.*, vol. 3, no. 1, p. 27, 2012.

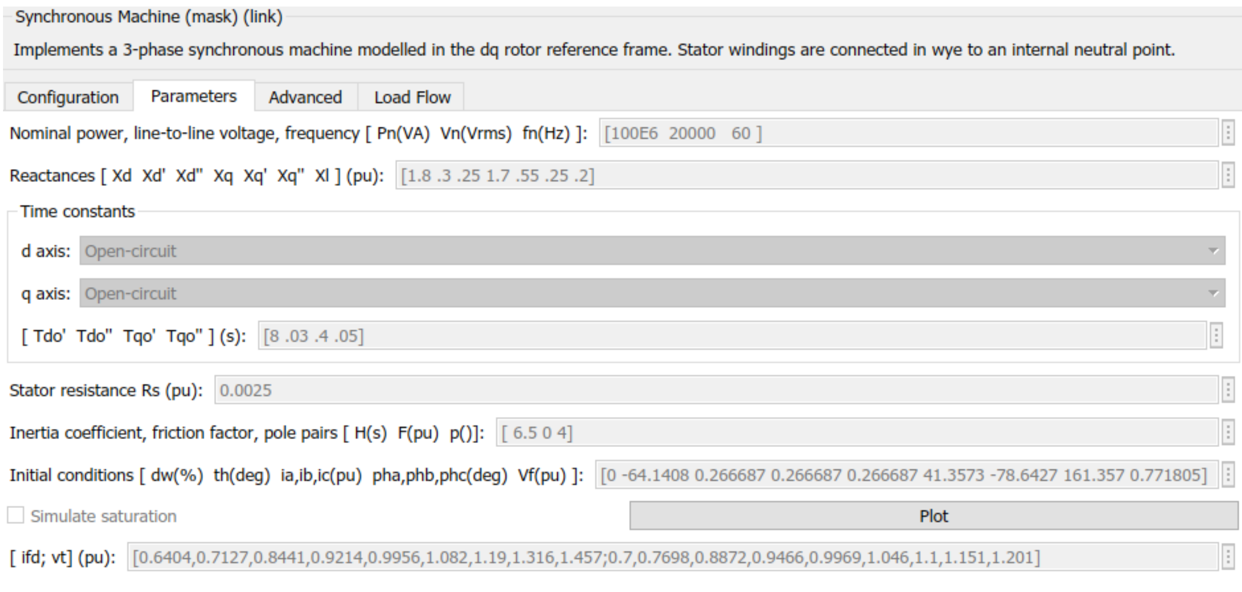
[81] Pai, M. A. *Energy function analysis for power system stability*. Springer Science & Business Media, 2012.

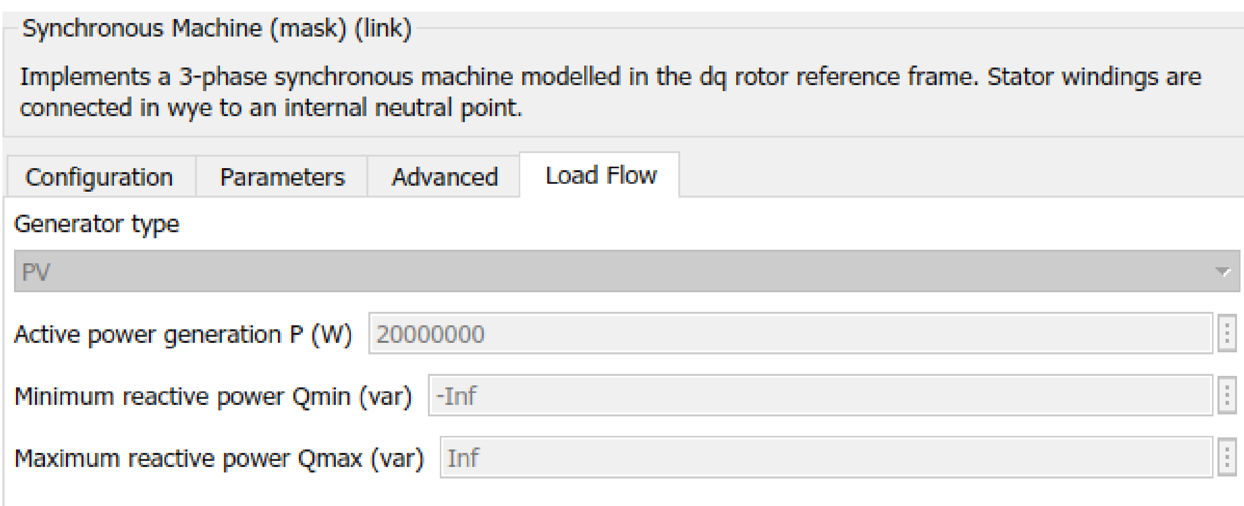
APPENDIX A: SIMULINK© MODEL PARAMETERS

This appendix details the parameters used for all of the blocks in the Simulink© model.

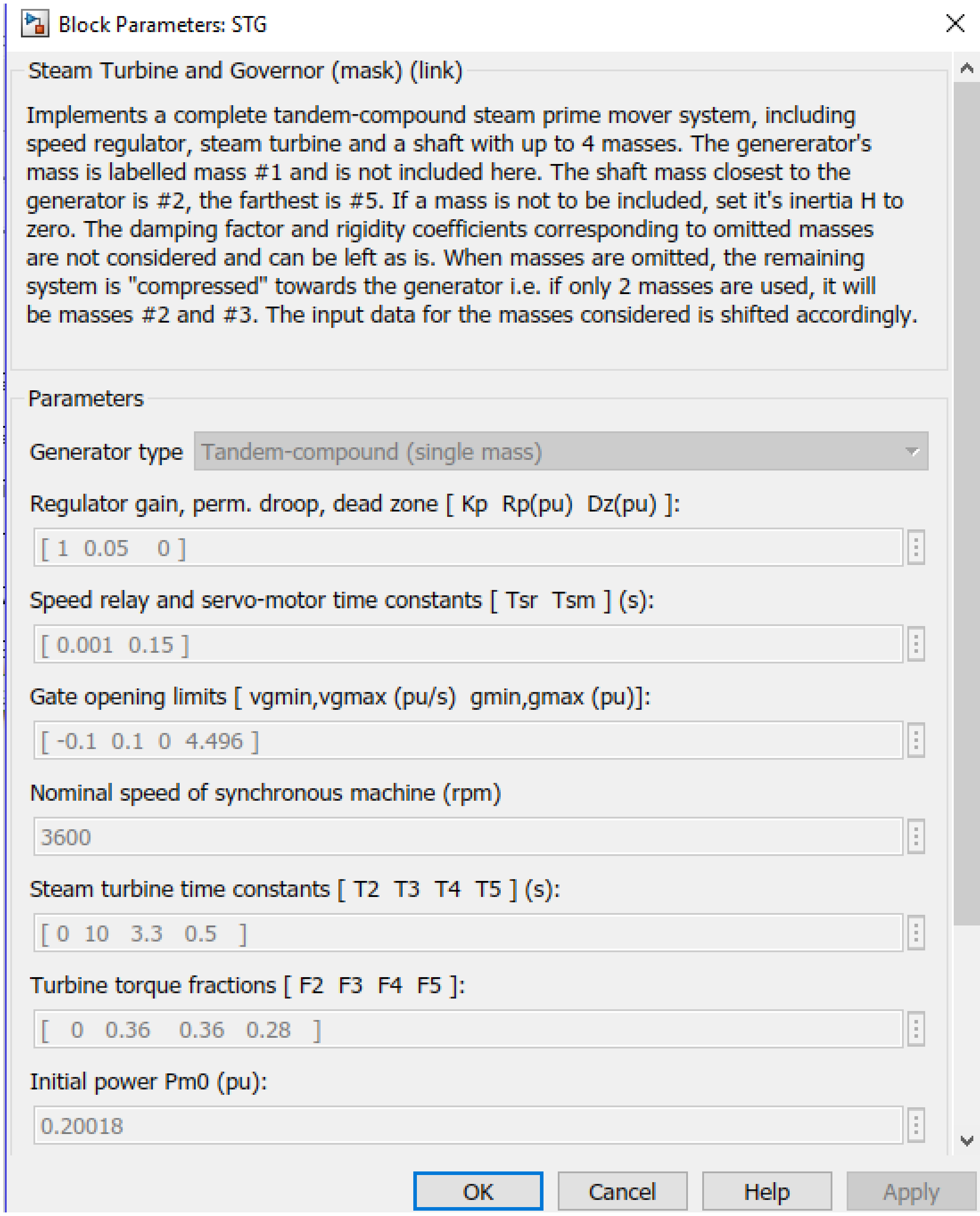
A.1 TRADITION GENERATING UNITS

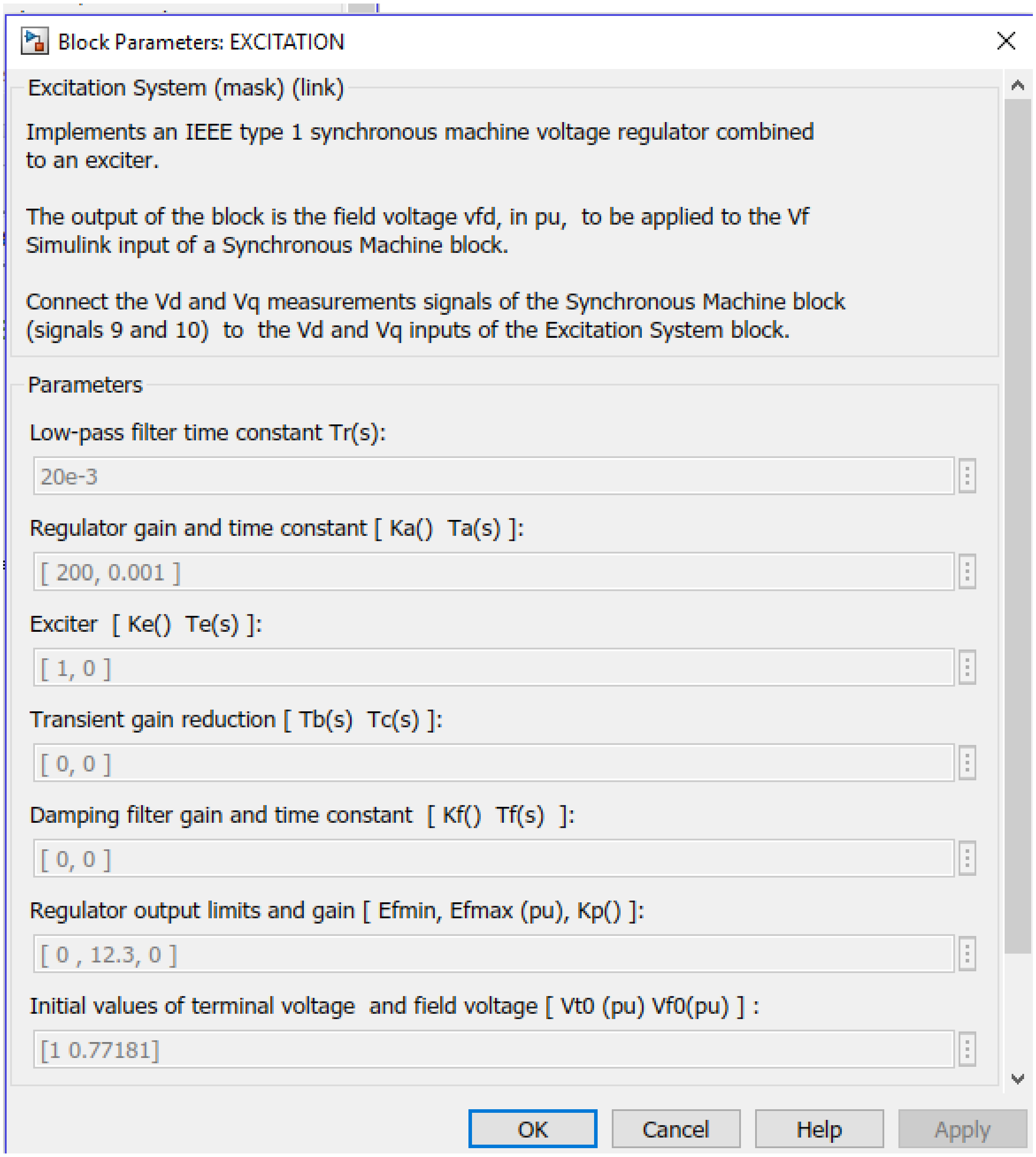
All machines use the same parameters except for the generator in area 3 where the only difference is that the Active Power Generation is 3 MW.

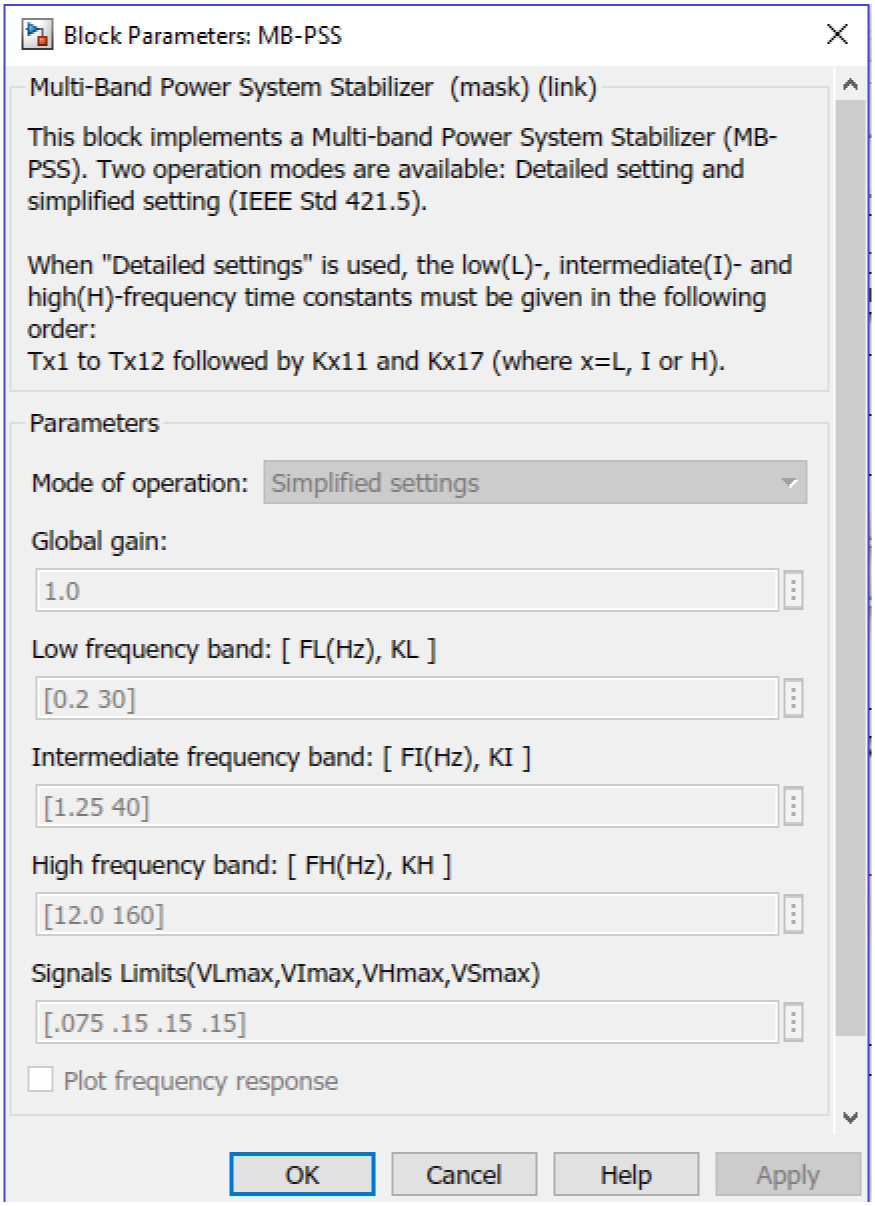




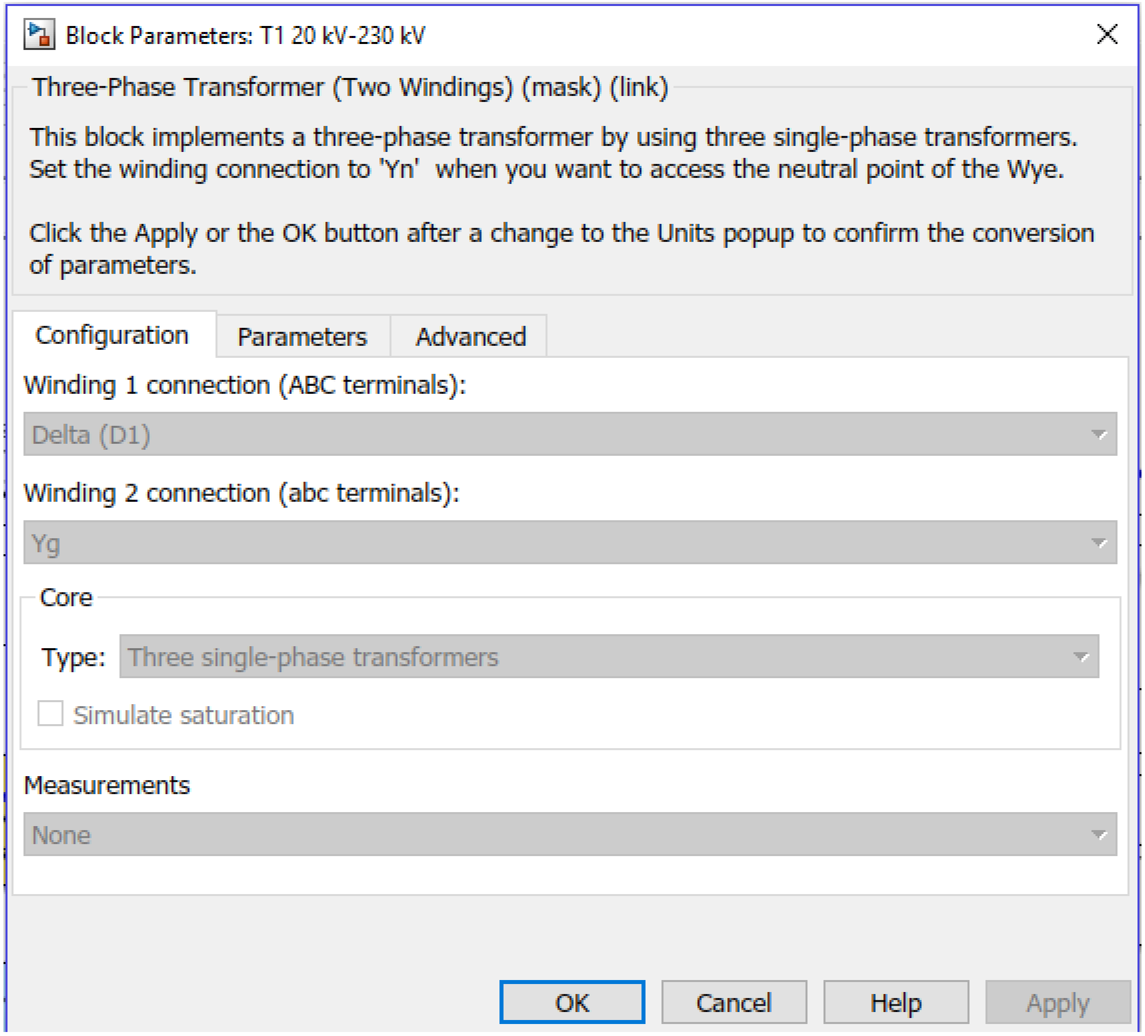
A.2 Traditional Generation: Turbine & Regulators

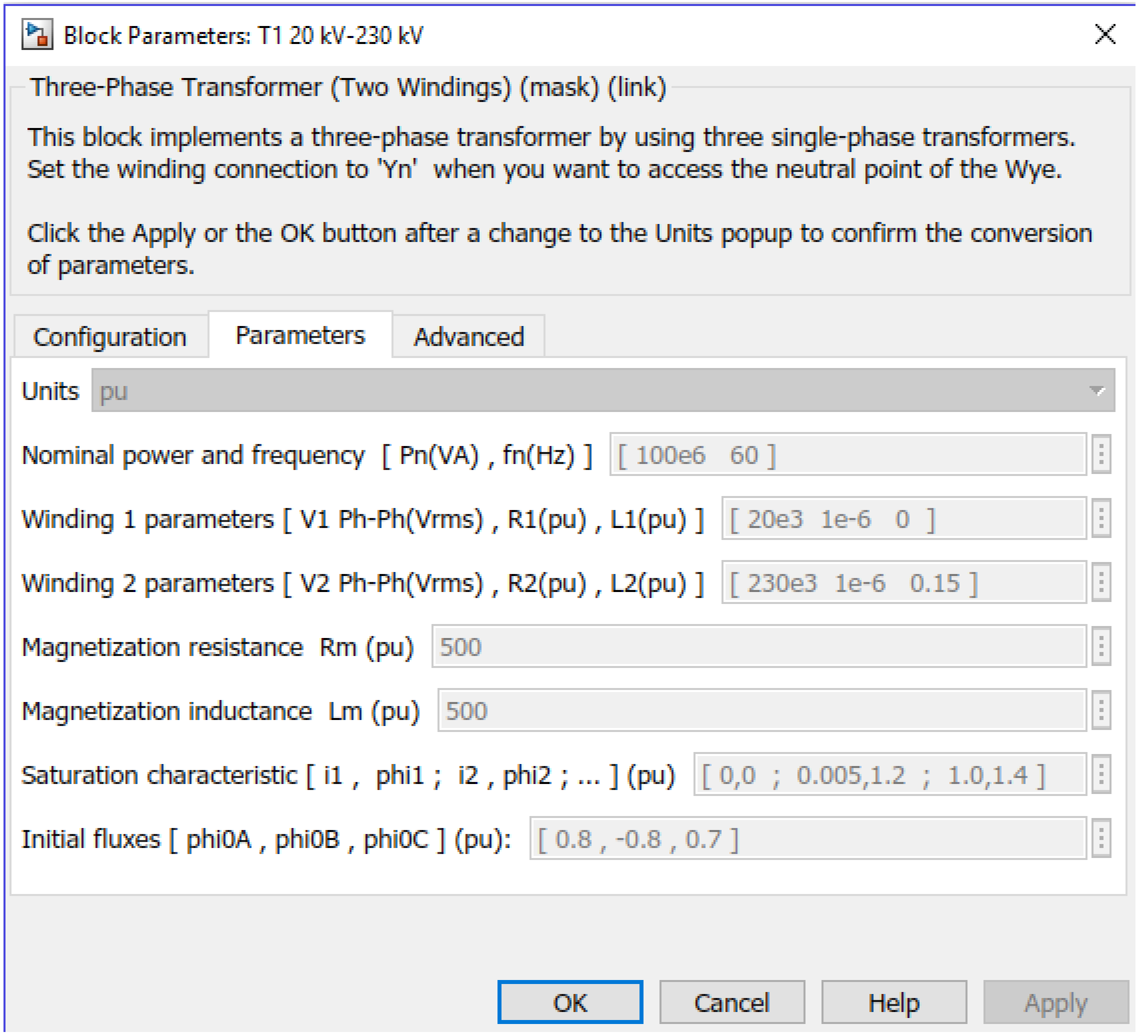


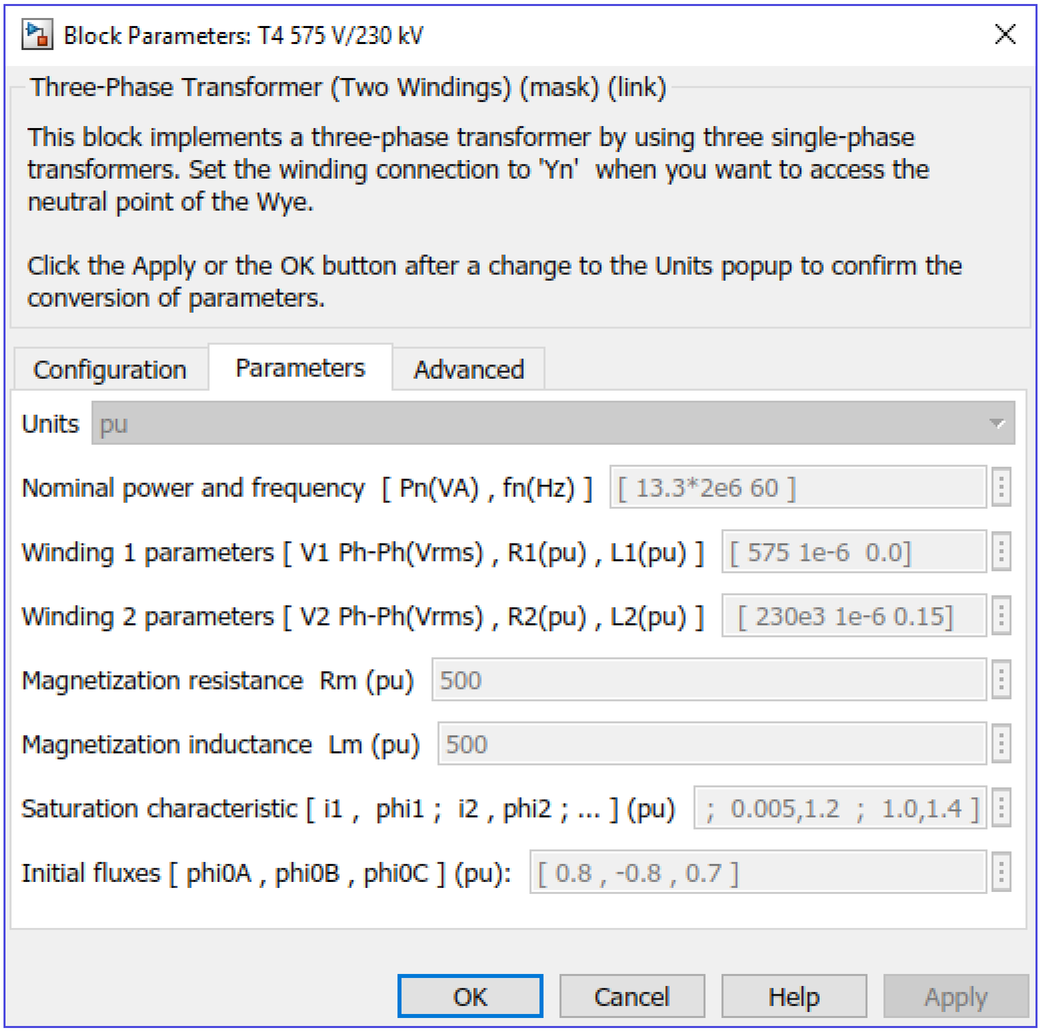

Block Parameters: EXCITATION
Excitation System (mask) (link)
Implements an IEEE type 1 synchronous machine voltage regulator combined to an exciter.
The output of the block is the field voltage vfd, in pu, to be applied to the Vf Simulink input of a Synchronous Machine block.
Connect the Vd and Vq measurements signals of the Synchronous Machine block (signals 9 and 10) to the Vd and Vq inputs of the Excitation System block.
Parameters
Low-pass filter time constant Tr(s):
20e-3
Regulator gain and time constant [Ka() Ta(s)]:
[200, 0.001]
Exciter [Ke() Te(s)]:
[1, 0]
Transient gain reduction [Tb(s) Tc(s)]:
[0, 0]
Damping filter gain and time constant [Kf() Tf(s)]:
[0, 0]
Regulator output limits and gain [Efmin, Efmax (pu), Kp()]:
[0 , 12.3, 0]
Initial values of terminal voltage and field voltage [Vt0 (pu) Vf0(pu)] :
[1 0.77181]
OK
Cancel
Help
Apply


Block Parameters: MB-PSS
Multi-Band Power System Stabilizer (mask) (link)
This block implements a Multi-band Power System Stabilizer (MB-PSS). Two operation modes are available: Detailed setting and simplified setting (IEEE Std 421.5).
When "Detailed settings" is used, the low(L)-, intermediate(I)- and high(H)-frequency time constants must be given in the following order:
Tx1 to Tx12 followed by Kx11 and Kx17 (where x=L, I or H).
Parameters
Mode of operation:
Simplified settings
Global gain:
1.0
Low frequency band: [FL(Hz), KL]
[0.2 30]
Intermediate frequency band: [FI(Hz), KI]
[1.25 40]
High frequency band: [FH(Hz), KH]
[12.0 160]
Signals Limits(VLmax,VImax,VHmax,VSmax)
[.075 .15 .15 .15]
Plot frequency response
OK
Cancel
Help
Apply

A.3 THREE-PHASE TRANSFORMERS

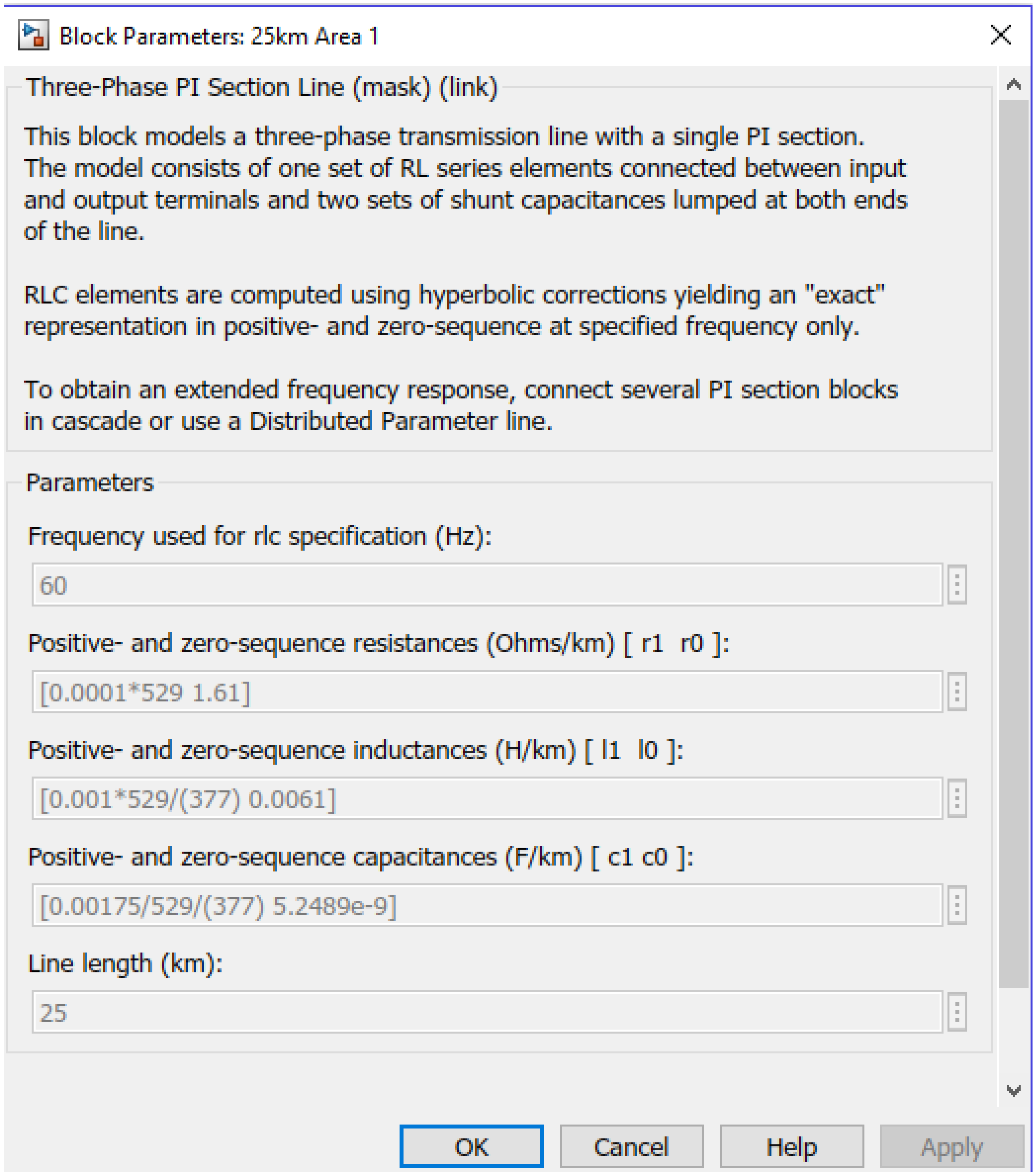



Block Parameters: T1 20 kV-230 kV
Three-Phase Transformer (Two Windings) (mask) (link)
This block implements a three-phase transformer by using three single-phase transformers. Set the winding connection to 'Yn' when you want to access the neutral point of the Wye.
Click the Apply or the OK button after a change to the Units popup to confirm the conversion of parameters.
Configuration
Parameters
Advanced
Units
pu
Nominal power and frequency [Pn(VA) , fn(Hz)]
[100e6 60]
Winding 1 parameters [V1 Ph-Ph(Vrms) , R1(pu) , L1(pu)]
[20e3 1e-6 0]
Winding 2 parameters [V2 Ph-Ph(Vrms) , R2(pu) , L2(pu)]
[230e3 1e-6 0.15]
Magnetization resistance Rm (pu)
500
Magnetization inductance Lm (pu)
500
Saturation characteristic [i1 , phi1 ; i2 , phi2 ; ...] (pu)
[0,0 ; 0.005,1.2 ; 1.0,1.4]
Initial fluxes [phi0A , phi0B , phi0C] (pu):
[0.8 , -0.8 , 0.7]
OK
Cancel
Help
Apply


Block Parameters: T4 575 V/230 kV
Three-Phase Transformer (Two Windings) (mask) (link)
This block implements a three-phase transformer by using three single-phase transformers. Set the winding connection to 'Yn' when you want to access the neutral point of the Wye.
Click the Apply or the OK button after a change to the Units popup to confirm the conversion of parameters.
Configuration
Parameters
Advanced
Units
pu
Nominal power and frequency [Pn(VA) , fn(Hz)]
[13.3*2e6 60]
Winding 1 parameters [V1 Ph-Ph(Vrms) , R1(pu) , L1(pu)]
[575 1e-6 0.0]
Winding 2 parameters [V2 Ph-Ph(Vrms) , R2(pu) , L2(pu)]
[230e3 1e-6 0.15]
Magnetization resistance Rm (pu)
500
Magnetization inductance Lm (pu)
500
Saturation characteristic [i1 , phi1 ; i2 , phi2 ; ...] (pu)
; 0.005,1.2 ; 1.0,1.4]
Initial fluxes [phi0A , phi0B , phi0C] (pu):
[0.8 , -0.8 , 0.7]
OK
Cancel
Help
Apply

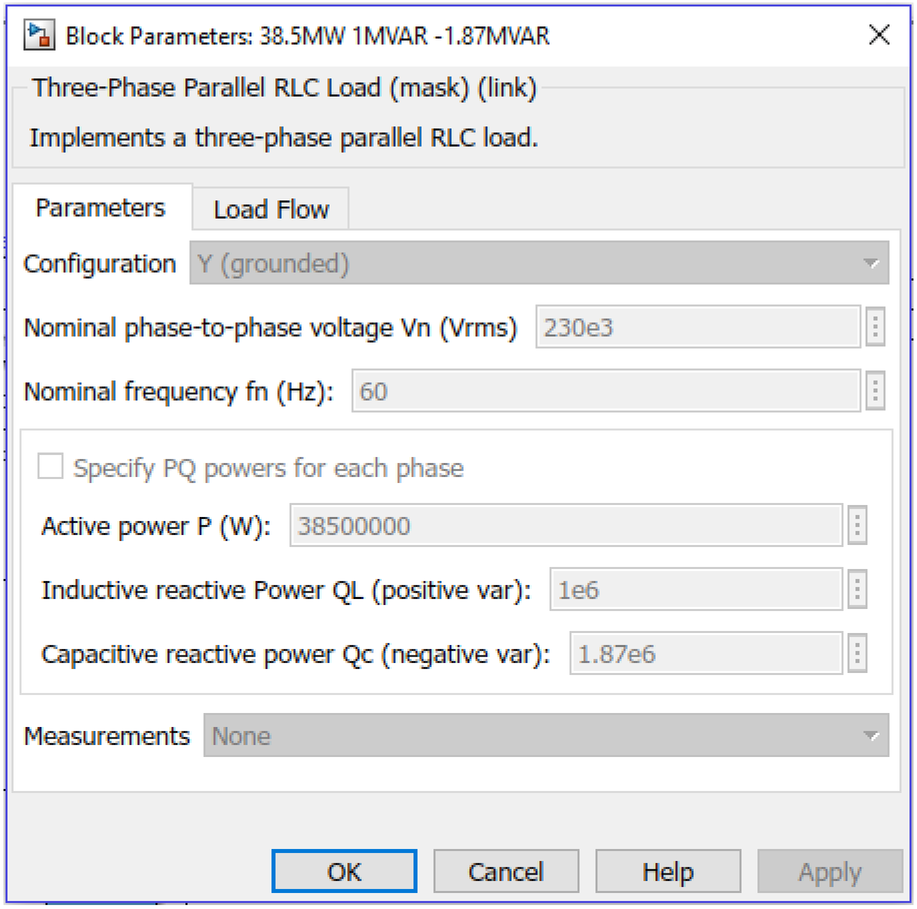
A.4 Three-Phase Transmission Lines

The parameters for each transmission line are equal except for the line length parameter, and the line impedance, which is different for different portions of the system, detailed in [65].

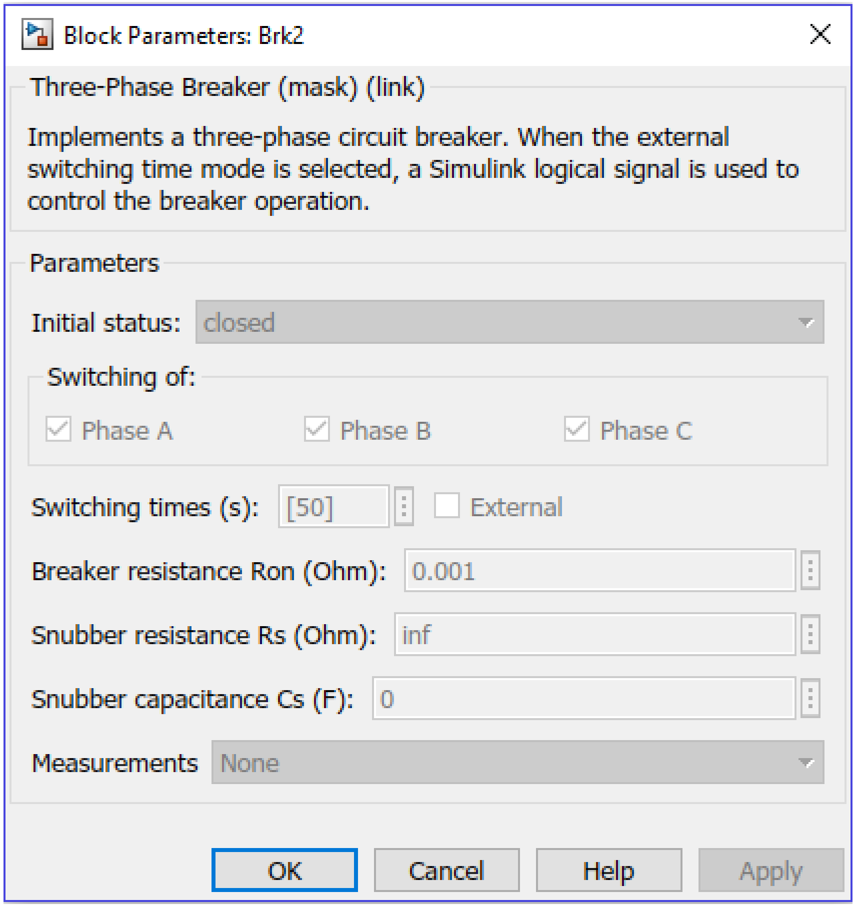


A.5 Electrical Load

The loads in area 1 and area 2 are identical and they are modeled as a constant impedance load.

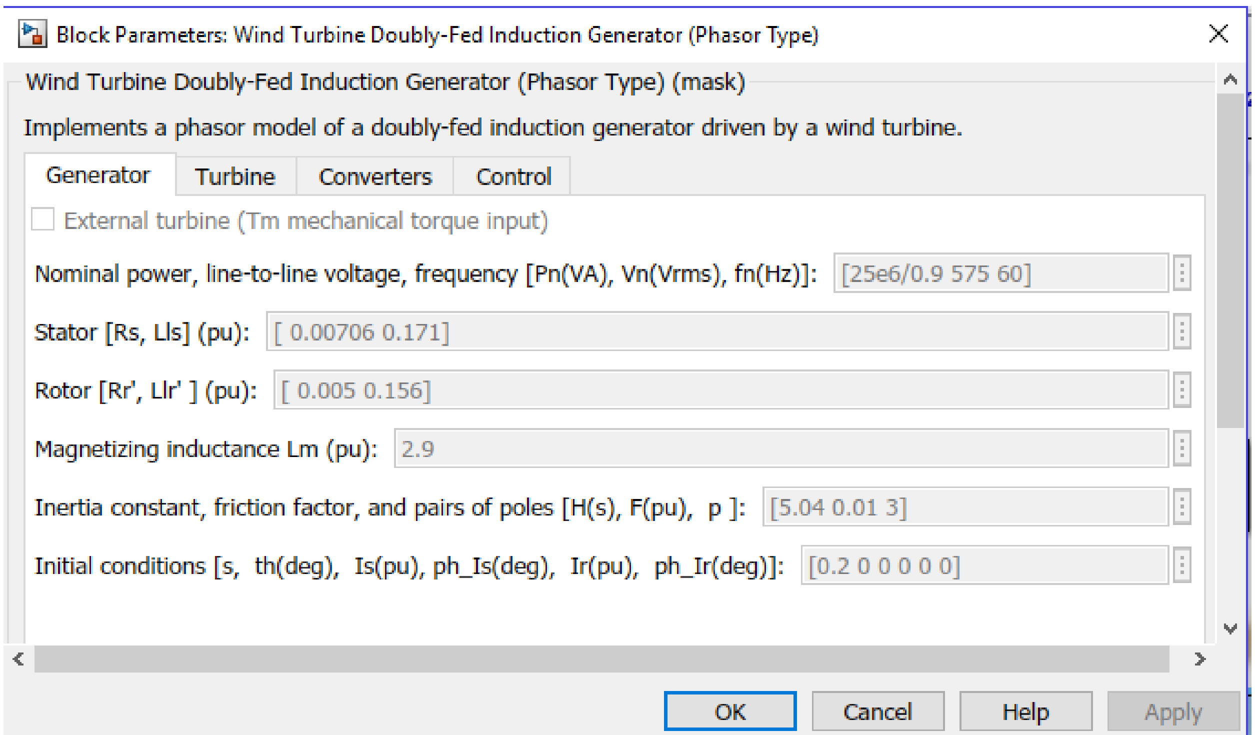


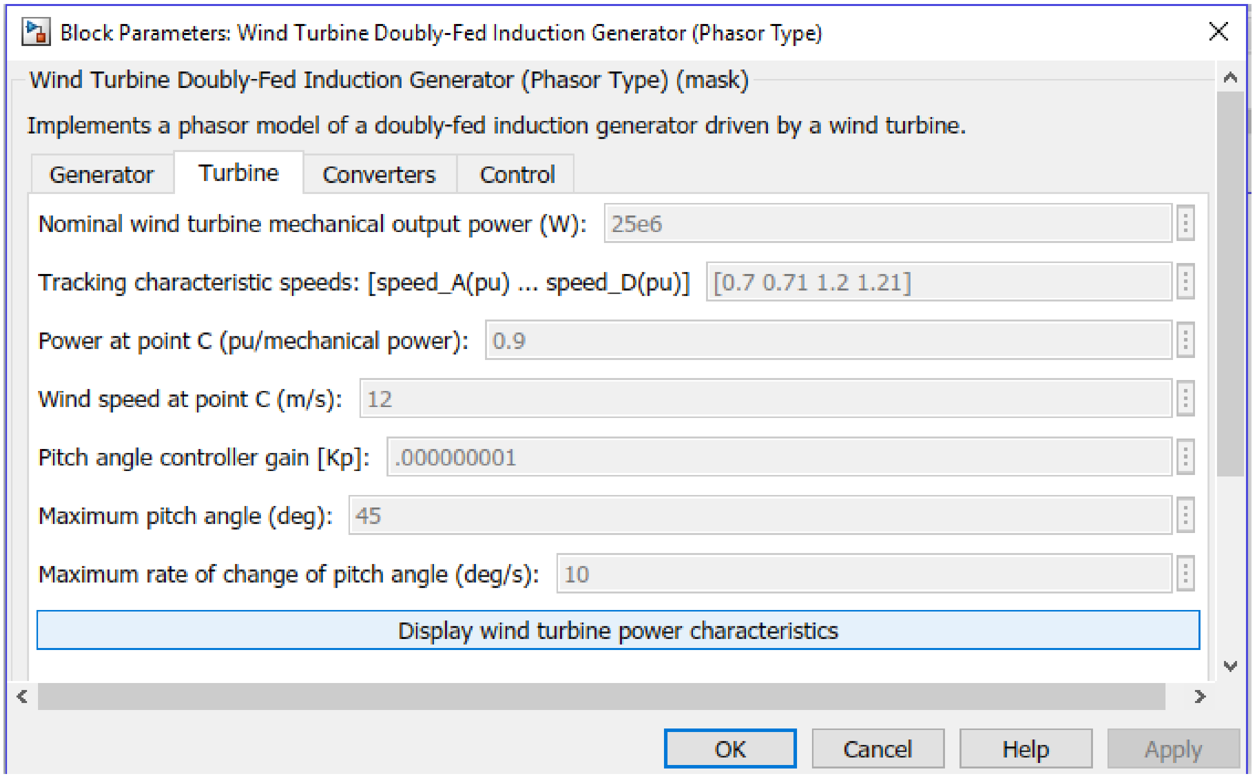
A.6 BREAKER

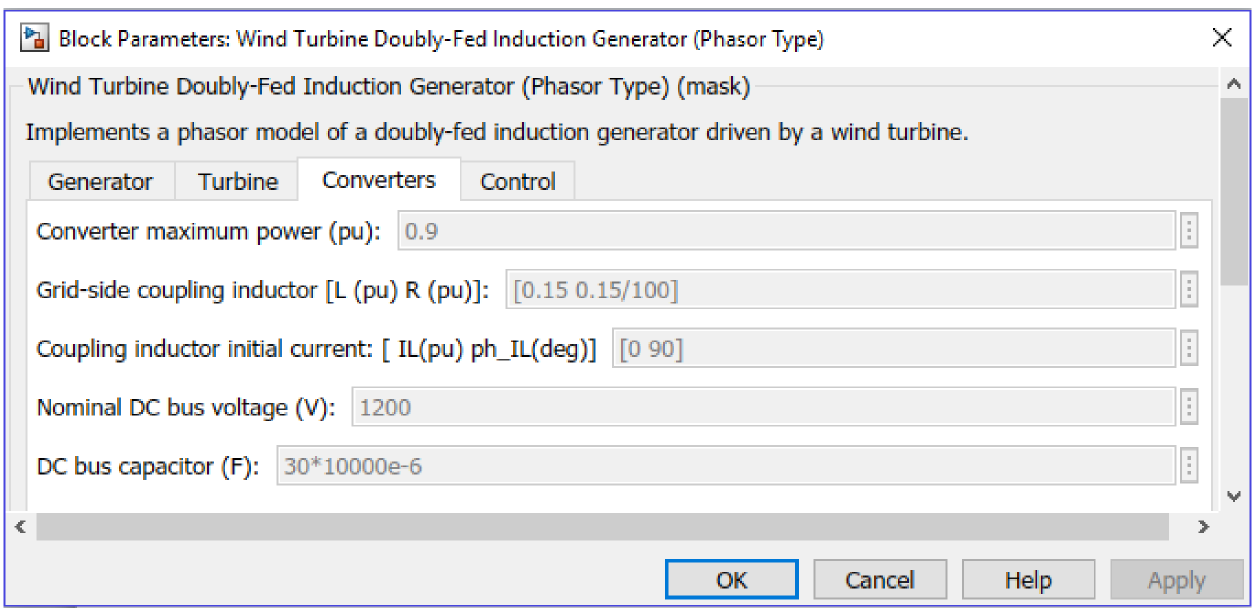

Block Parameters: Brk2
Three-Phase Breaker (mask) (link)
Implements a three-phase circuit breaker. When the external switching time mode is selected, a Simulink logical signal is used to control the breaker operation.
Parameters
Initial status:
closed
Switching of:
Phase A
Phase B
Phase C
Switching times (s):
[50]
External
Breaker resistance Ron (Ohm):
0.001
Snubber resistance Rs (Ohm):
inf
Snubber capacitance Cs (F):
0
Measurements
None
OK
Cancel
Help
Apply

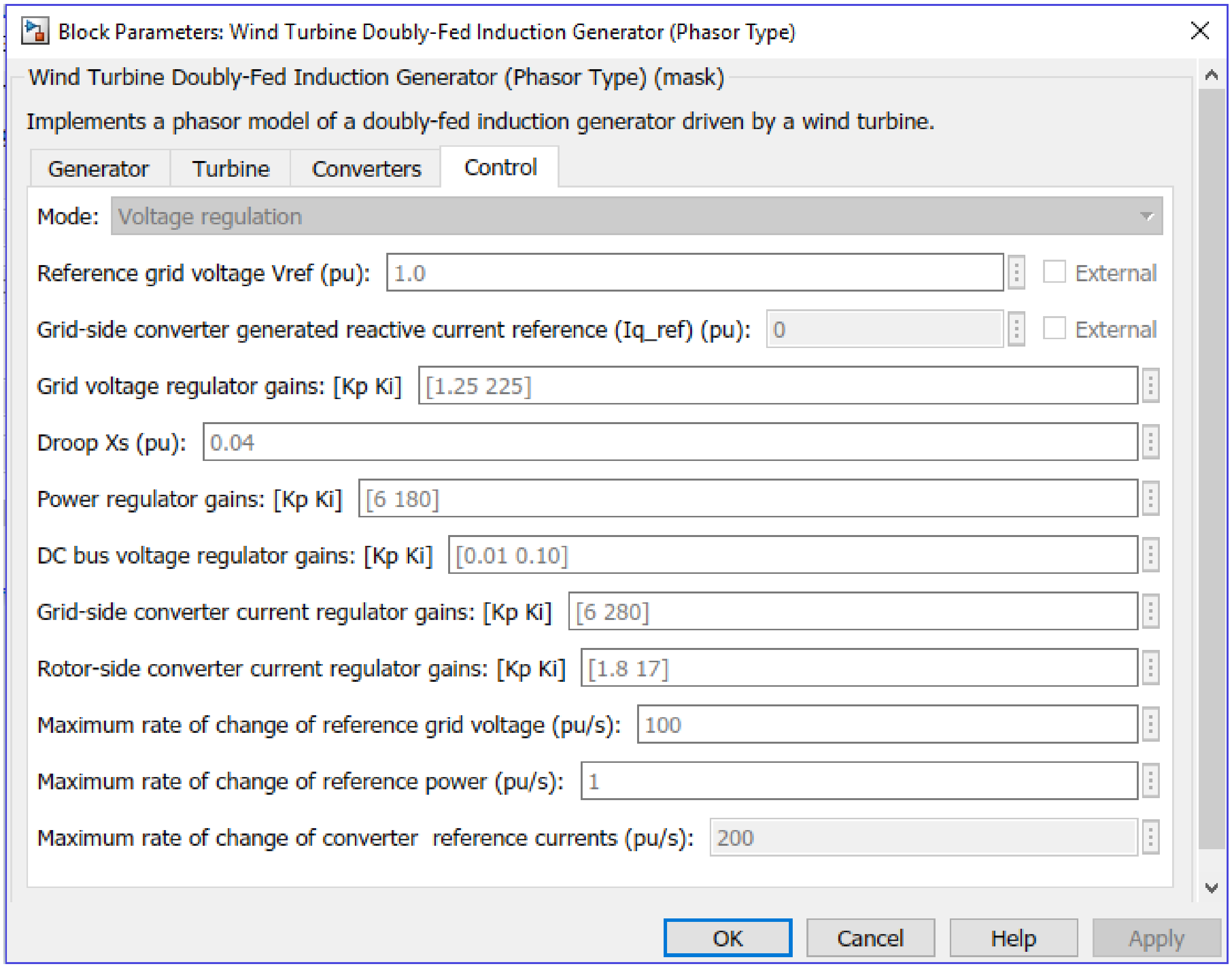
A.7 WIND TURBINE

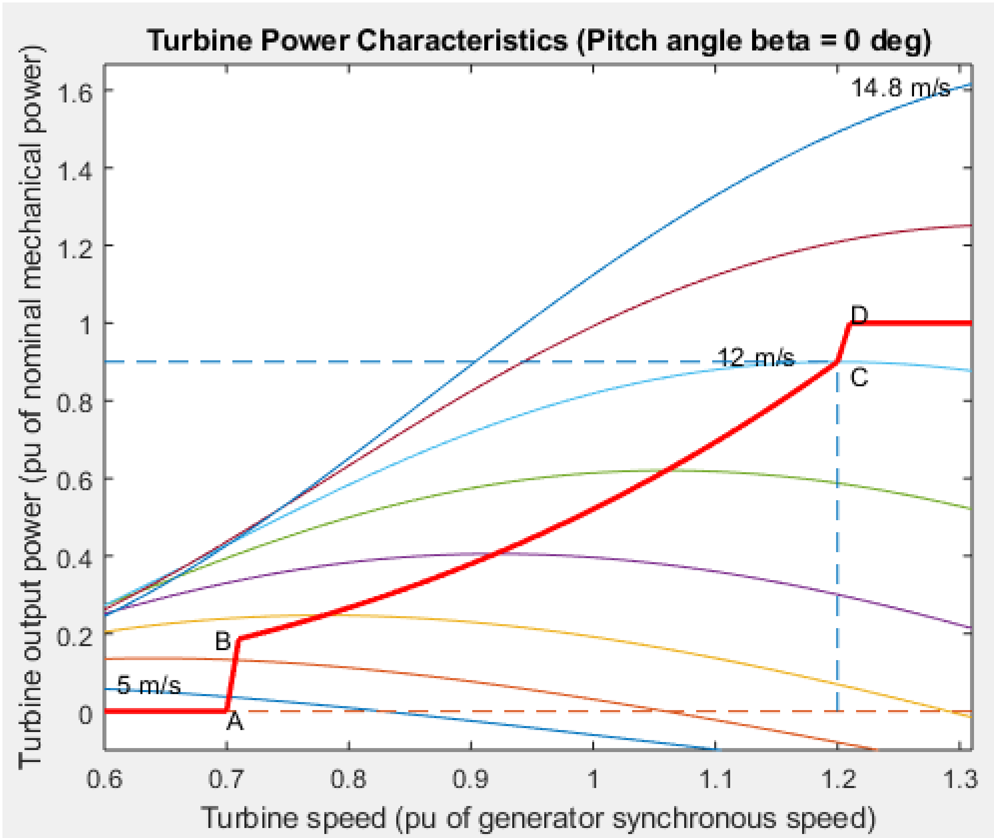
The Pitch Angle Controller Gain parameter is set to a very small value; this is because removing the block from the wind turbine model would cause an error in the simulation. This was bypassed by setting the gain to a very small value and building a custom pitch controller to bypass the one designed into the model.



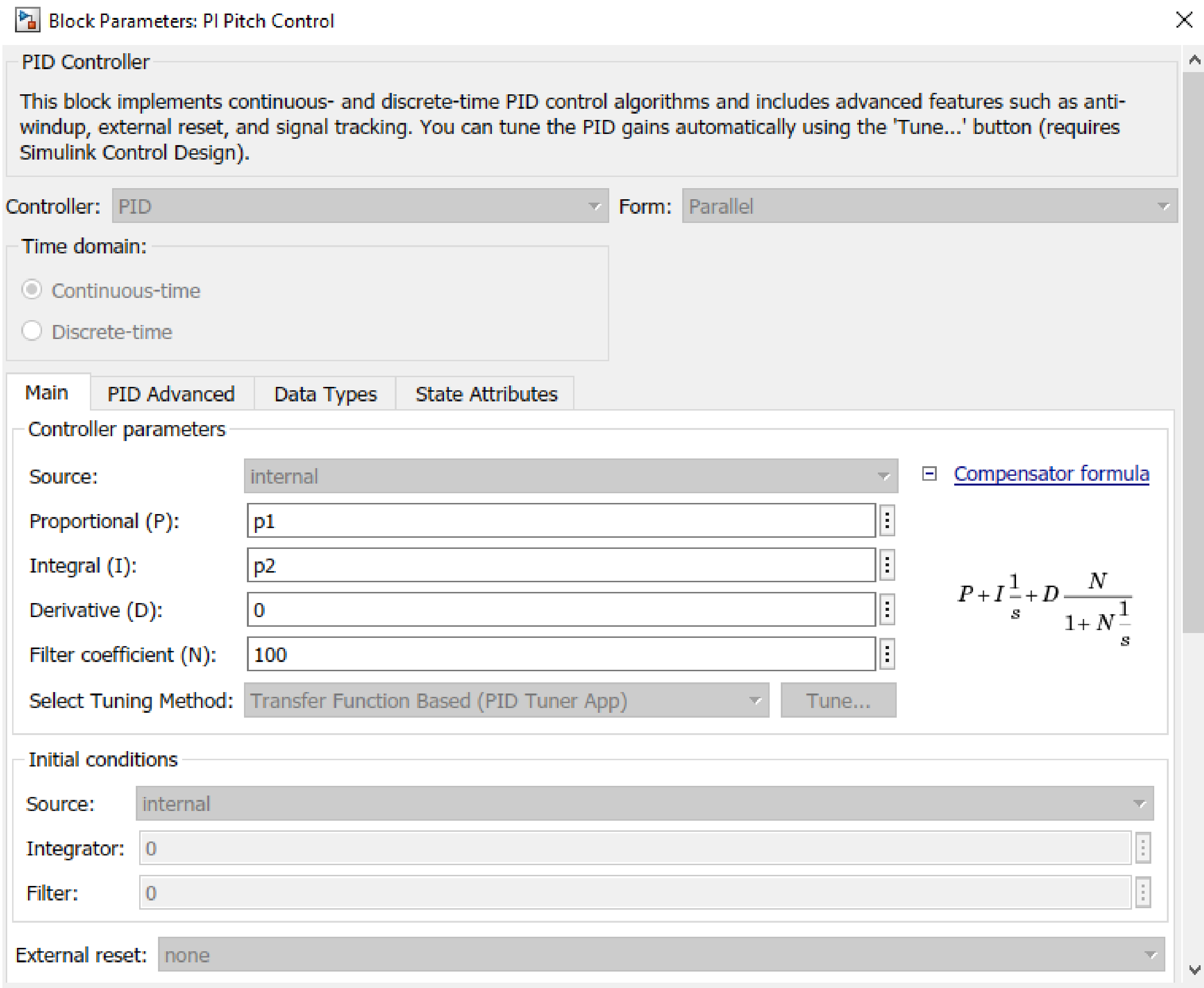

Block Parameters: Wind Turbine Doubly-Fed Induction Generator (Phasor Type)
Wind Turbine Doubly-Fed Induction Generator (Phasor Type) (mask)
Implements a phasor model of a doubly-fed induction generator driven by a wind turbine.
Generator
Turbine
Converters
Control
Nominal wind turbine mechanical output power (W):
25e6
Tracking characteristic speeds: [speed_A(pu) ... speed_D(pu)]
[0.7 0.71 1.2 1.21]
Power at point C (pu/mechanical power):
0.9
Wind speed at point C (m/s):
12
Pitch angle controller gain [Kp]:
.000000001
Maximum pitch angle (deg):
45
Maximum rate of change of pitch angle (deg/s):
10
Display wind turbine power characteristics
OK
Cancel
Help
Apply

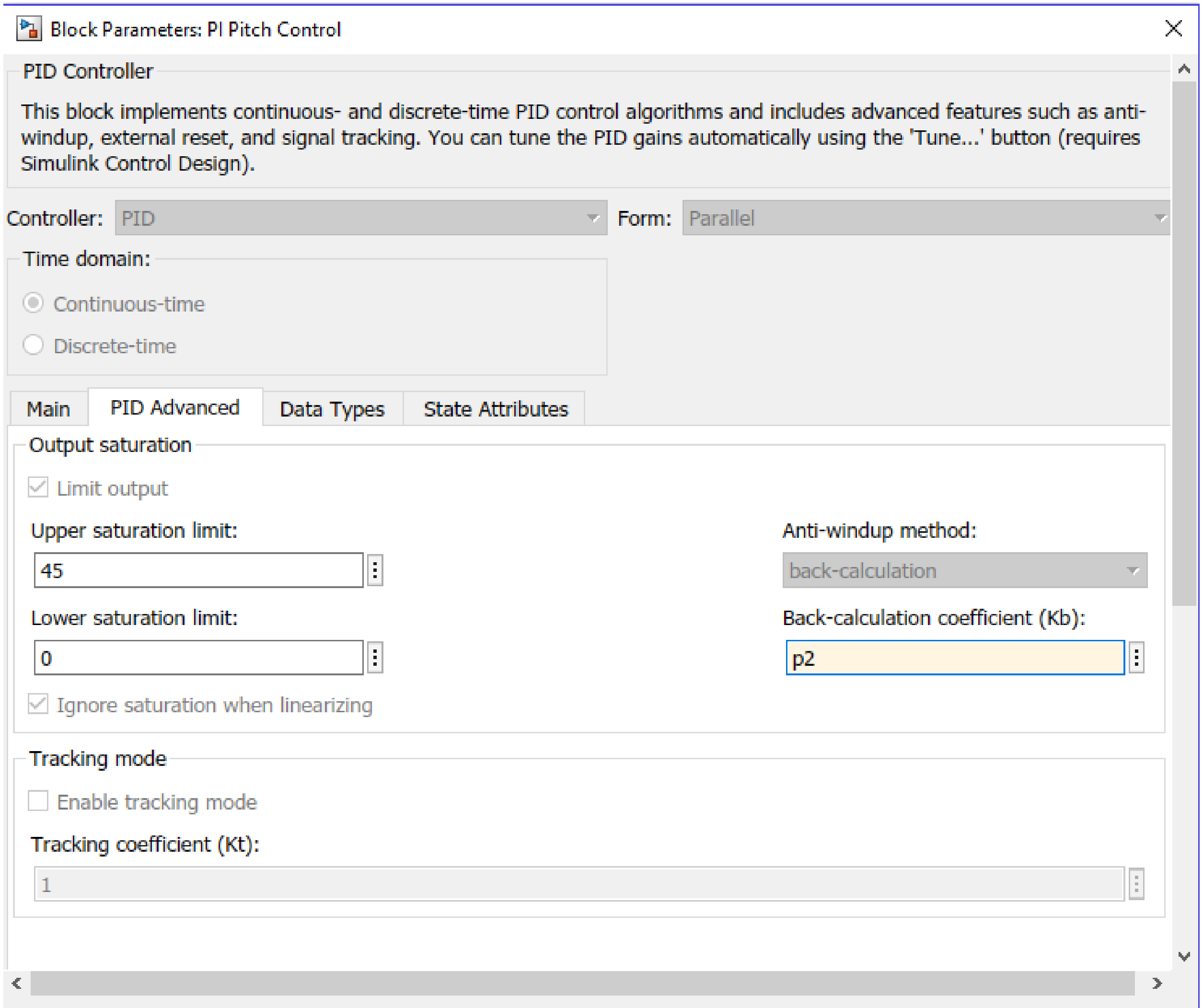

Block Parameters: Wind Turbine Doubly-Fed Induction Generator (Phasor Type)
Wind Turbine Doubly-Fed Induction Generator (Phasor Type) (mask)
Implements a phasor model of a doubly-fed induction generator driven by a wind turbine.
Generator
Turbine
Converters
Control
Converter maximum power (pu):
0.9
Grid-side coupling inductor [L (pu) R (pu)]:
[0.15 0.15/100]
Coupling inductor initial current: [IL(pu) ph_IL(deg)]
[0 90]
Nominal DC bus voltage (V):
1200
DC bus capacitor (F):
30*10000e-6
OK
Cancel
Help
Apply

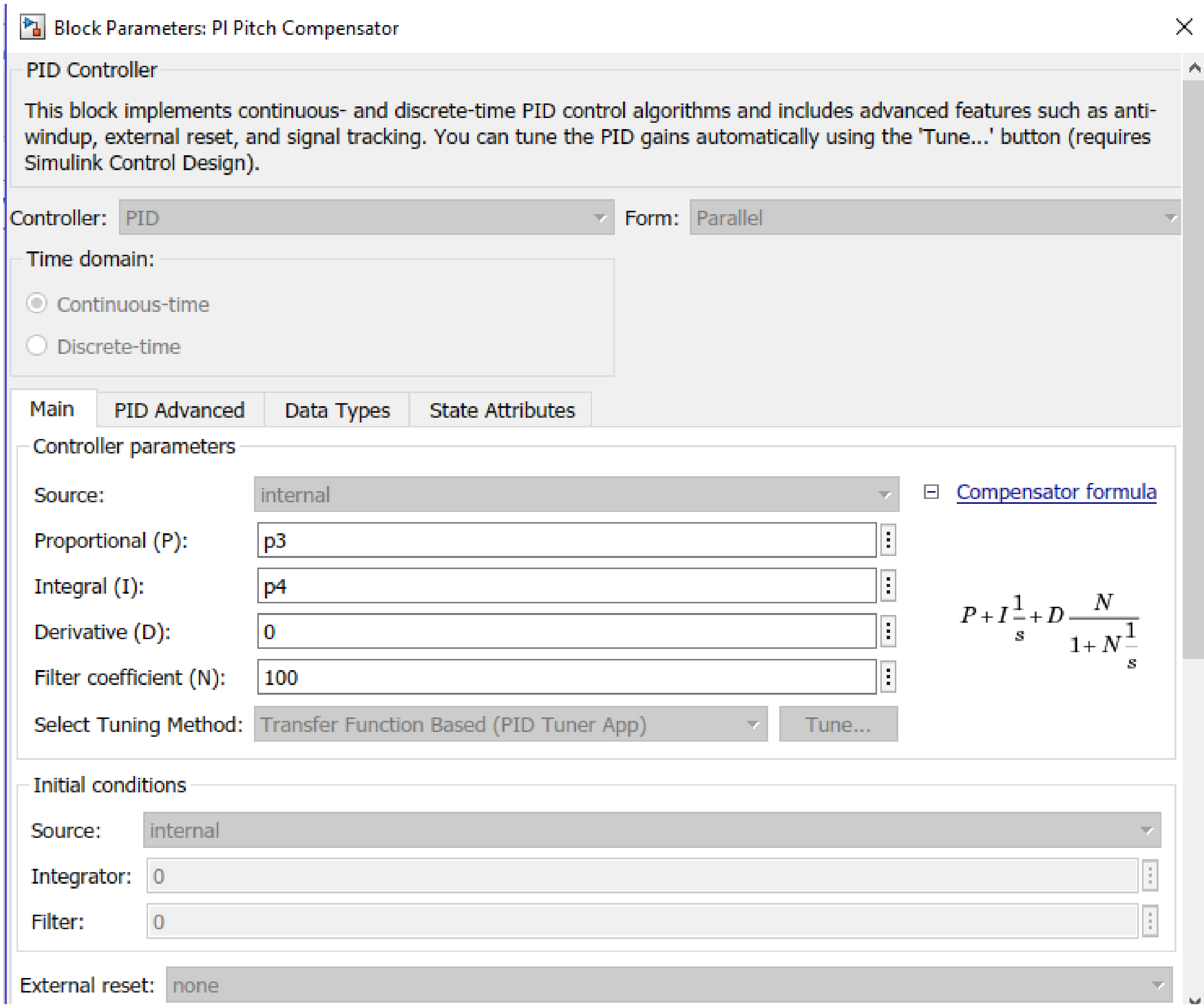

Block Parameters: Wind Turbine Doubly-Fed Induction Generator (Phasor Type)
Wind Turbine Doubly-Fed Induction Generator (Phasor Type) (mask)
Implements a phasor model of a doubly-fed induction generator driven by a wind turbine.
Generator
Turbine
Converters
Control
Mode: Voltage regulation
Reference grid voltage Vref (pu): 1.0
External
Grid-side converter generated reactive current reference (Iq_ref) (pu): 0
External
Grid voltage regulator gains: [Kp Ki] [1.25 225]
Droop Xs (pu): 0.04
Power regulator gains: [Kp Ki] [6 180]
DC bus voltage regulator gains: [Kp Ki] [0.01 0.10]
Grid-side converter current regulator gains: [Kp Ki] [6 280]
Rotor-side converter current regulator gains: [Kp Ki] [1.8 17]
Maximum rate of change of reference grid voltage (pu/s): 100
Maximum rate of change of reference power (pu/s): 1
Maximum rate of change of converter reference currents (pu/s): 200
OK
Cancel
Help
Apply


Turbine Power Characteristics (Pitch angle beta = 0 deg)
14.8 m/s
12 m/s
5 m/s
A
B
C
D
Turbine output power (pu of nominal mechanical power)
Turbine speed (pu of generator synchronous speed)
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3

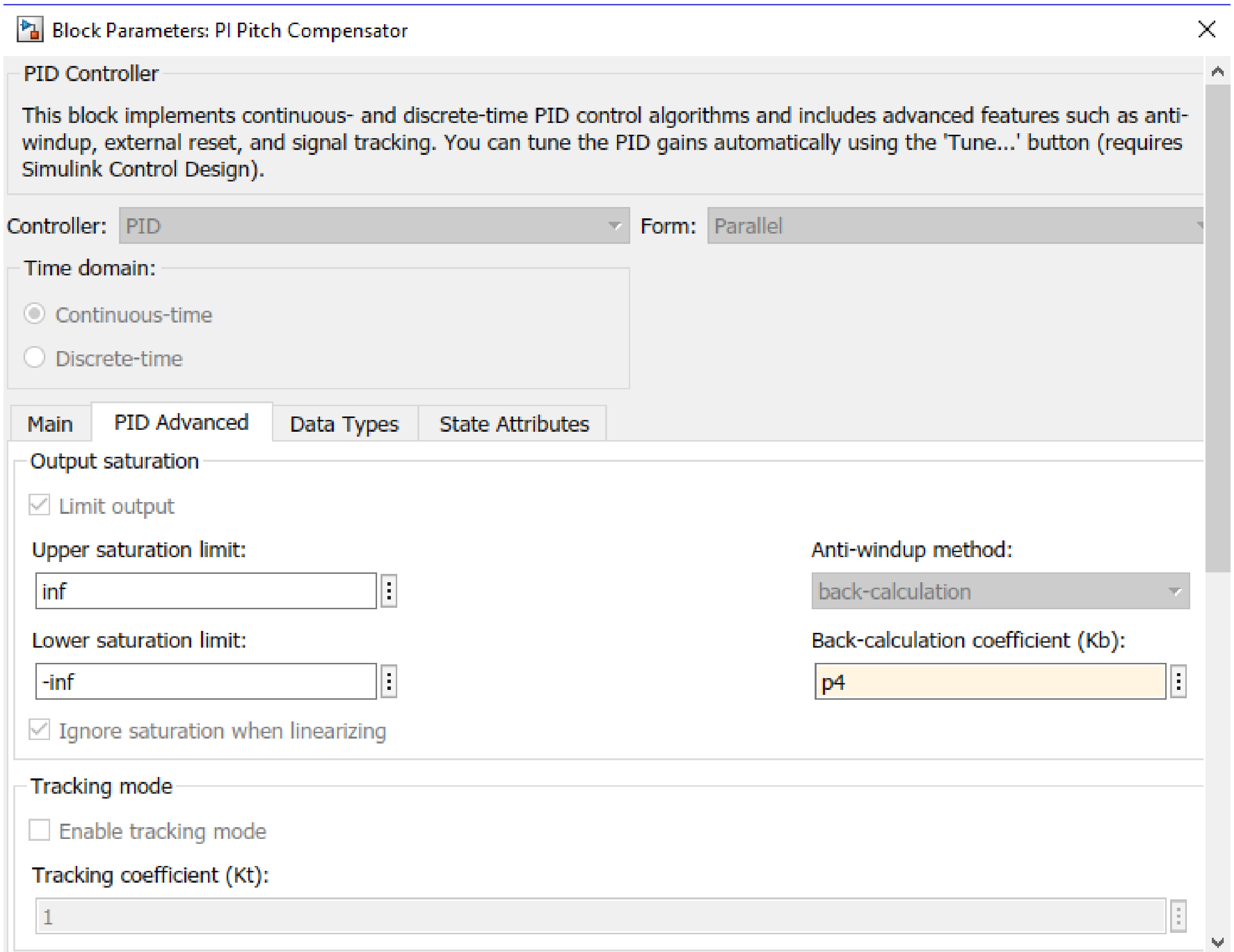
A.8 Pitch Controllers

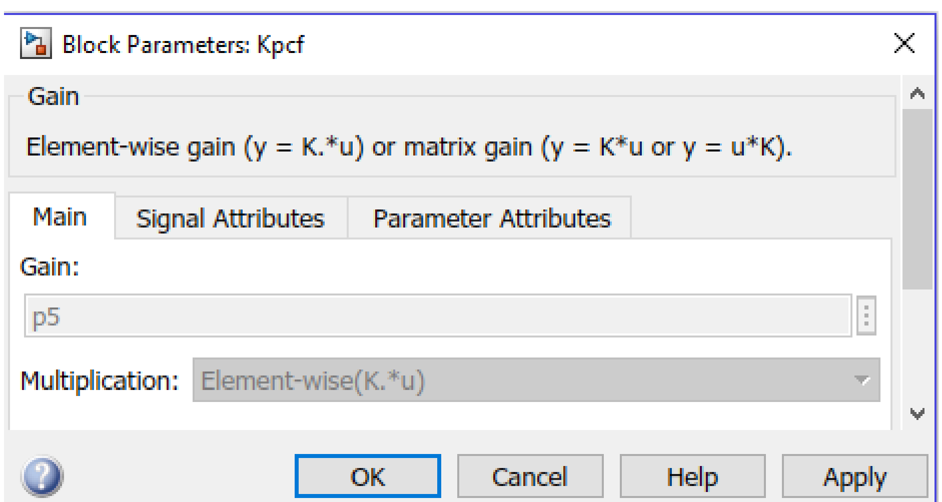


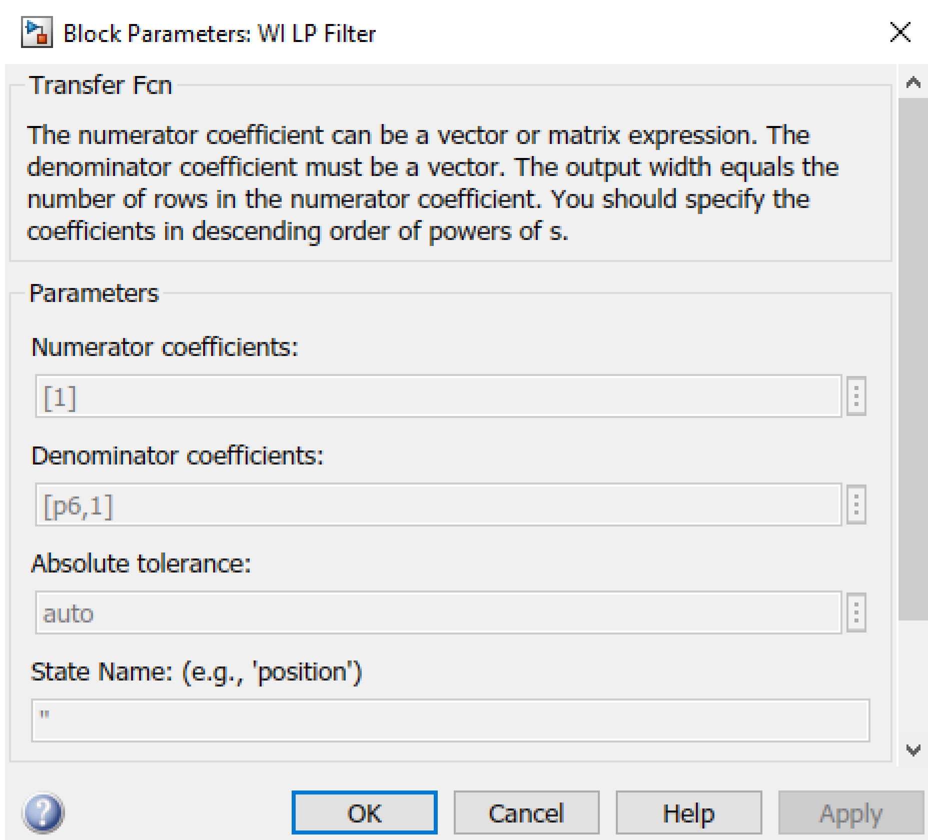

Block Parameters: PI Pitch Control
PID Controller
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).
Controller:
PID
Form:
Parallel
Time domain:
Continuous-time
Discrete-time
Main
PID Advanced
Data Types
State Attributes
Output saturation
Limit output
Upper saturation limit:
45
Lower saturation limit:
0
Anti-windup method:
back-calculation
Back-calculation coefficient (Kb):
p2
Ignore saturation when linearizing
Tracking mode
Enable tracking mode
Tracking coefficient (Kt):
1

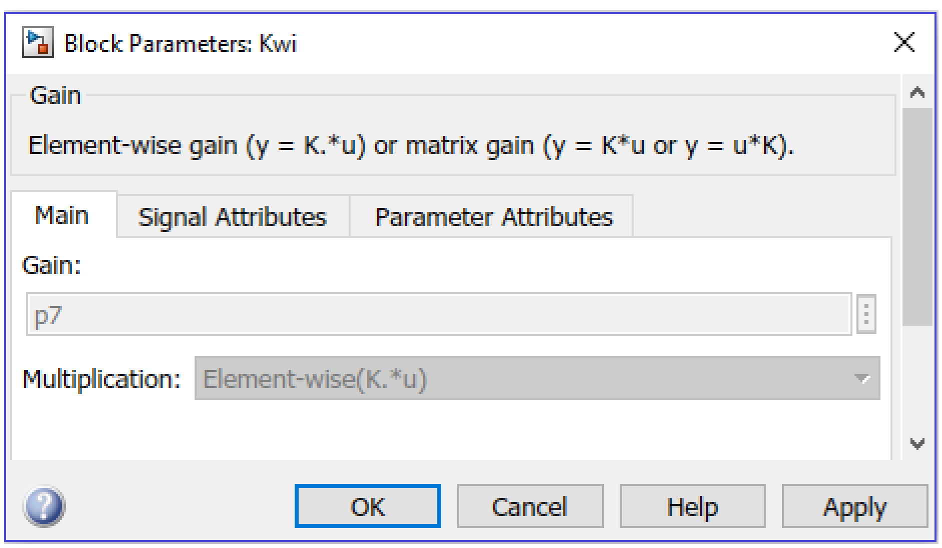


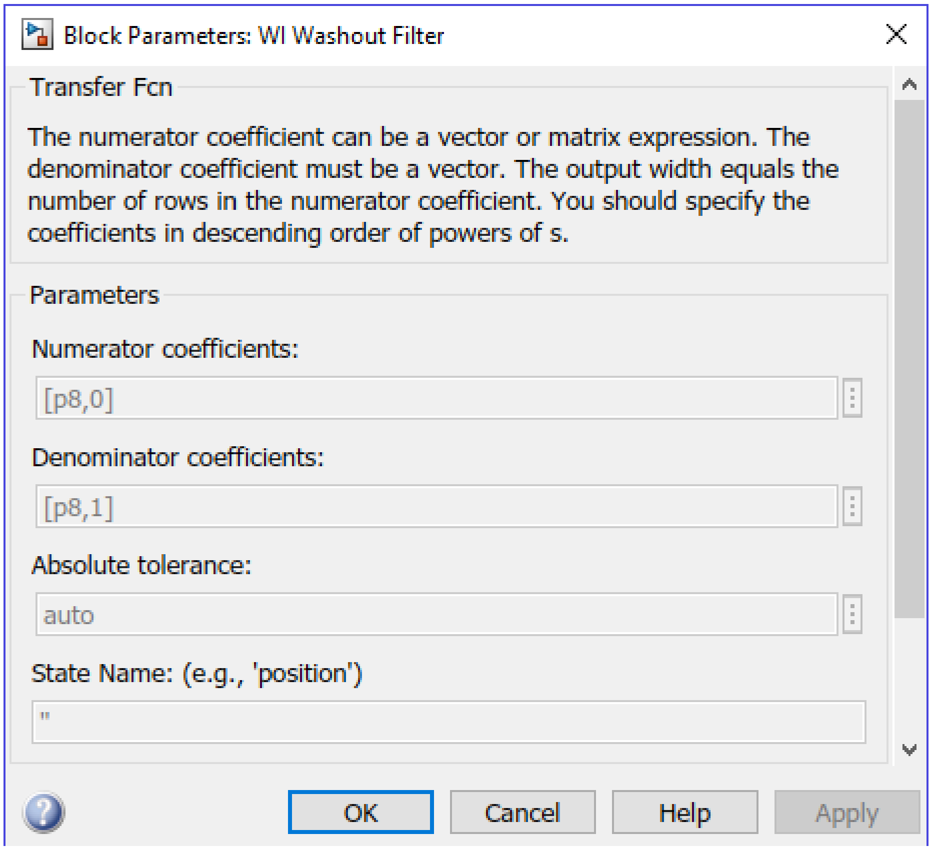
Block Parameters: PI Pitch Compensator
PID Controller
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).
Controller: PID
Form: Parallel
Time domain:
Continuous-time
Discrete-time
Main
PID Advanced
Data Types
State Attributes
Controller parameters
Source: internal
Compensator formula
Proportional (P): p3
Integral (I): p4
Derivative (D): 0
Filter coefficient (N): 100
$P+I\frac{1}{s}+D\frac{N}{1+N\frac{1}{s}}$
Select Tuning Method: Transfer Function Based (PID Tuner App)
Tune...
Initial conditions
Source: internal
Integrator: 0
Filter: 0
External reset: none


Block Parameters: PI Pitch Compensator
PID Controller
This block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. You can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design).
Controller: PID
Form: Parallel
Time domain:
Continuous-time
Discrete-time
Main
PID Advanced
Data Types
State Attributes
Output saturation
Limit output
Upper saturation limit:
inf
Lower saturation limit:
-inf
Anti-windup method:
back-calculation
Back-calculation coefficient (Kb):
p4
Ignore saturation when linearizing
Tracking mode
Enable tracking mode
Tracking coefficient (Kt):
1


Block Parameters: Kpcf
Gain
Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).
Main
Signal Attributes
Parameter Attributes
Gain:
p5
Multiplication:
Element-wise(K.*u)
OK
Cancel
Help
Apply


Block Parameters: WI LP Filter
Transfer Fcn
The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.
Parameters
Numerator coefficients:
[1]
Denominator coefficients:
[p6,1]
Absolute tolerance:
auto
State Name: (e.g., 'position')
''
OK
Cancel
Help
Apply


Block Parameters: Kwi
Gain
Element-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).
Main
Signal Attributes
Parameter Attributes
Gain:
p7
Multiplication:
Element-wise(K.*u)
OK
Cancel
Help
Apply


Block Parameters: WI Washout Filter
Transfer Fcn
The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.
Parameters
Numerator coefficients:
[p8,0]
Denominator coefficients:
[p8,1]
Absolute tolerance:
auto
State Name: (e.g., 'position')
''
OK
Cancel
Help
Apply

APPENDIX B: GENETIC ALGORITHM CODE

This appendix details all of the code (.m files) used for implementing the Genetic Algorithm in Matlab/Simulink©.

B.1 GPID_CON.M

```
% ****************** GPid_con.m *******************
%
%                     Version 1.0
%                    June 25, 2000
%                 Dr. Marco P. Schoen
% ______________________________________________
%                     Version 1.1
%                    March 24, 2019
%                 Shat C. Pratoomratana
%
%Version 1.1 Updates:
%
%--Added LO inputs for 8 parameters.
%--Load Simulink Model and Initial Conditions
%--Modified mutatecon.m to include all LO values
%--Custom costfunction.m file that sets parameters and runs the Simulink
%  model which outputs a time vector (cost function output) that is then
%  integrated by the costfuntion.m file and then set as the cost for that
%  particular set of parameters

%
% Toolbox for genetic programming. Code for continuous
% genetic algorithm.

tic %Start timer
% Define Variables:
maxiterations=input('Maximum Number of iterations: ');
ipopsize=input('Population Size of Generation 0: ');
popsize=input('Population Size for Generations 1 - end: ');
%popsize=popsize*ipopsize;
keep=input('Number of Chromosomes kept for mating: ');
%keep=keep*popsize;
pars=input('Total Number of parameters in a chromosome: ');
mutaterate=input('Mutation rate: ');
% Each Parameter can have and individual high value
% in this case the low end can always be zero
hi1=input('High end of parameter 1 value: ');
hi2=input('High end of parameter 2 value: ');
hi3=input('High end of parameter 3 value: ');
hi4=input('High end of parameter 4 value: ');
hi5=input('High end of parameter 5 value: ');
hi6=input('High end of parameter 6 value: ');
hi7=input('High end of parameter 7 value: ');
```

```
hi8=input('High end of parameter 8 value: ');
lo1=input('Low end of parameter 1 value: ');
lo2=input('Low end of parameter 2 value: ');
lo3=input('Low end of parameter 3 value: ');
lo4=input('Low end of parameter 4 value: ');
lo5=input('Low end of parameter 5 value: ');
lo6=input('Low end of parameter 6 value: ');
lo7=input('Low end of parameter 7 value: ');
lo8=input('Low end of parameter 8 value: ');


op=1;%input('Probability options: 1. Top-Bottom 2.Random 3. Weigh-Rand. : ');

% Create the initial population, evaluate costs, and sort
CHROMOSOMES(:,1)=((hi1-lo1)*(rand(ipopsize,1)))+lo1;
CHROMOSOMES(:,2)=((hi2-lo2)*(rand(ipopsize,1)))+lo2;
CHROMOSOMES(:,3)=((hi3-lo3)*(rand(ipopsize,1)))+lo3;
CHROMOSOMES(:,4)=((hi4-lo4)*(rand(ipopsize,1)))+lo4;
CHROMOSOMES(:,5)=((hi5-lo5)*(rand(ipopsize,1)))+lo5;
CHROMOSOMES(:,6)=((hi6-lo6)*(rand(ipopsize,1)))+lo6;
CHROMOSOMES(:,7)=((hi7-lo7)*(rand(ipopsize,1)))+lo7;
CHROMOSOMES(:,8)=((hi8-lo8)*(rand(ipopsize,1)))+lo8;
% CHROMOSOMES will be a matrix of random numbers within hi - lo

% Load Initial conditions into workspace so simulink model starts in Steady
% state
    load PCKWI_15ms_IC.mat
    disp('Initial conditions loaded into workspace.')

%  The name of the model is:
    model = 'wt25small_PCKpfcWI15ms.slx';

%  Make sure the model is loaded in memory
    open_system(model);
    disp([model,' is loaded into memory.'])

% Loop:
gen=0;quit=0;w=1;
h = waitbar(0,'Please wait...');
while (gen<maxiterations && (~quit))
   disp(['Generation...',num2str(w)]);
   gen=gen+1;
   cost=costfunction(CHROMOSOMES);
   New=[cost,CHROMOSOMES];
   New2=sortrows(New,[1]); cost=New2(:,1);CHROMOSOMES=New2(:,2:pars+1);
   mincost(gen)=min(cost);
   meancost(gen)=mean(cost);
   stdcost(gen)=std(cost);
   % Pairing,Mating, and Mutation
   %CHROMOSOMES=New2(1:popsize,2:3);cost=New2(1:popsize,1);
   [Mom,Dad]=pairing(CHROMOSOMES,cost,keep,popsize,op);
   CHROMOSOMES=matecon(Mom,Dad,CHROMOSOMES,keep,popsize,pars);
```

```
CHROMOSOMES=mutatecon(CHROMOSOMES,mutaterate,popsize,pars,hi1,hi2,hi3,hi4,hi5,hi6,hi7,hi8,lo1,lo2,lo
3,lo4,lo5,lo6,lo7,lo8);
  % Check for Conversions
  % if mincost(gen)< ... and/or meancost(gen) < ... and or stdcost(gen)< ...         quit=1
  waitbar(gen/maxiterations);
  w=w+1;
end
close(h);
TopChrom=CHROMOSOMES(1,:)
min(cost)
plot(mincost)
toc %Stop timer
elapsedTime = toc;
```

B.2 COSTFUNCTION.M

```
function cost=costfunction(CHROMOSOMES)
[row,col]=size(CHROMOSOMES);
cost=zeros(row,1);
k=1;
model = 'wt25small_PCKpfcWI15ms';
for i=1:row
  p1=CHROMOSOMES(i,1);%Kp for pitch controller
  p2=CHROMOSOMES(i,2);%Ki for pitch controller
  p3=CHROMOSOMES(i,3);%Kp for pitch compensator
  p4=CHROMOSOMES(i,4);%Ki for pitch compensator
  p5=CHROMOSOMES(i,5);%Kp frequency droop
  p6=CHROMOSOMES(i,6);%Tlp time constant for WI low pass filter
  p7=CHROMOSOMES(i,7);%Kp WindINERTIA
  p8=CHROMOSOMES(i,8);%Two time constant for WI washout filter

  % Save parameter values to workspace because simulink can only
  % read them from there using the set_param function
  assignin('base','p1',CHROMOSOMES(i,1));
  assignin('base','p2',CHROMOSOMES(i,2));
  assignin('base','p3',CHROMOSOMES(i,3));
  assignin('base','p4',CHROMOSOMES(i,4));
  assignin('base','p5',CHROMOSOMES(i,5));
  assignin('base','p6',CHROMOSOMES(i,6));
  assignin('base','p7',CHROMOSOMES(i,7));
  assignin('base','p8',CHROMOSOMES(i,8));


   % Set parameters in each block
   disp('Setting Control Parameters.')
   pitch_controller_1 = ['wt25small_PCKpfcWI15ms/Area 2/Wind Turbine Doubly-        Fed Induction Generator
(Phasor Type)/Generator &            Converters/Control/Enabled Pitch Control/PI Pitch Control'];
   pitch_compensator_1 = ['wt25small_PCKpfcWI15ms/Area 2/Wind Turbine    Doubly-Fed Induction Generator
(Phasor Type)/Generator &            Converters/Control/Enabled Pitch Compensator/PI Pitch Compensator'];
   freq_droop_1 = ['wt25small_PCKpfcWI15ms/Area 2/Wind Turbine Doubly-Fed          Induction Generator
(Phasor Type)/Generator &            Converters/Control/Enabled Frequency Droop/Kpcf'];
```

```
    lp_tc_1 = ['wt25small_PCKpfcWI15ms/Area 2/Wind Turbine Doubly-Fed      Induction Generator (Phasor
Type)/Generator & Converters/Control/WI LP            Filter'];
    Kwi_1 = ['wt25small_PCKpfcWI15ms/Area 2/Wind Turbine Doubly-Fed Induction      Generator (Phasor
Type)/Generator & Converters/Control/Enabled WI    Gain/Kwi'];
    wo_tc_1 = ['wt25small_PCKpfcWI15ms/Area 2/Wind Turbine Doubly-Fed     Induction Generator (Phasor
Type)/Generator & Converters/Control/WI    Washout Filter'];
    set_param(pitch_controller_1,'P','p1','I','p2');
    set_param(pitch_compensator_1,'P','p3','I','p4');
    set_param(freq_droop_1,'Gain','p5');
    set_param(lp_tc_1,'Denominator','[p6,1]','Numerator','[1]');
    set_param(Kwi_1,'Gain','p7');
    set_param(wo_tc_1,'Denominator','[p8,1]','Numerator','[p8,0]');
    % So the model does not have to recompile on every run
    set_param(model,'FastRestart','on');

    % Simulate
    disp(['Simulating...',num2str(k)]);
    simout = sim(model);
    % Cumulative integration of the error obtained from simulink
    int_err = cumtrapz(simout.err_squared.Data);
    % Since it is a cumulative integration only the last entry of the
    % int_err vector is the actual error
    error = int_err(length(int_err),1);

    cost(i,1)= error;
    k=k+1;
end
```

B.3 MUTATECON.M

```
function
CHROMOSOMES=mutatecon(CHROMOSOMES,mutaterate,popsize,pars,hi1,hi2,hi3,hi4,hi5,hi6,hi7,hi8,lo1,lo2,lo
3,lo4,lo5,lo6,lo7,lo8)
% Inside a loop iterating over the number of mutations, a random
% parameter in the population is selected and replaced by a new
% random parameter

nmu=ceil(popsize*pars*mutaterate);
for i=1:nmu
   hi = 0;
   lo = 0;
   row=ceil(popsize*rand)+1;
   col=ceil(pars*rand);

   if col == 1
      hi = hi1;
      lo = lo1;
   elseif col == 2
      hi = hi2;
      lo = lo2;
   elseif col == 3
      hi = hi3;
```

```
      lo = lo3;
   elseif col == 4
      hi = hi4;
      lo =lo4;
   elseif col == 5
      hi = hi5;
      lo = lo5;
   elseif col == 6
      hi = hi6;
      lo = lo6;
   elseif col == 7
      hi = hi7;
      lo = lo7;
   else
      hi = hi8;
      lo = lo8;
   end

   CHROMOSOMES(row,col)=(hi-lo)*rand+lo;
end
```

B.4 MATECON.M

```
function CHROMOSOMES=matecon(Mom,Dad,CHROMOSOMES,keep,popsize,pars)
% Code for continuous GP mating. Selects a crossover point
% ceil rounds to next higher integer. Row index contains
% first offspring, row intex +1 contains second offspring
% Mom-vector containing row numbers of first parent
% Dad-vector containing row numbers of second parent

CHROMOSOMES;%to test chromosomes on the screen during development phase
replace=(popsize-keep)/2;
for ic=1:replace
   alpha=ceil(rand*pars);i=2*(ic-1)+1;
   beta=rand(1);
   CHROMOSOMES(keep+i,alpha)=CHROMOSOMES(Mom(ic),alpha)-beta*(CHROMOSOMES(Mom(ic),alpha)-
CHROMOSOMES(Dad(ic),alpha));

CHROMOSOMES(keep+i+1,alpha)=CHROMOSOMES(Dad(ic),alpha)+beta*(CHROMOSOMES(Mom(ic),alpha)-
CHROMOSOMES(Dad(ic),alpha));

end;
```

B.5 PAIRING.M

```
function[Mom,Dad]=pairing(CHROMOSOMES,cost,keep,popsize,op)
% Based on the probability option op, one of the following
% three pairing criterias is used:
% 1. Top-Down, 2. Random, 3. Weighted Random
```

```
% for the selection of the parents of the next generation

replacements=(popsize-keep)/2;denum=0;
if op==1
  for r=1:replacements
    denum=denum+r;
  end;
  for n=1:replacements
    probn(n)=n/denum;
  end;
end;
if op==2
  %need to write code
end;
if op==3
  %need to write code
end;

%Cummulative Probabilities
cum=0;odds=zeros(1,replacements);
for i=1:replacements
  cum=probn(i)+cum;
  odds(1,i)=cum;
end;
%Roll dice for Parents
pick1=rand(1,replacements); %vector of random # for Mom
pick2=rand(1,replacements); %vector of random # for Dad

Mom=zeros(1,replacements);Dad=Mom;
for i=1:replacements
  for j=2:replacements
    if (pick1(i)<odds(j) & pick1(i)>odds(j-1))
      Mom(i)=j;
    end;
    if Mom(i)==0
      Mom(i)=1;
    end;
    if (pick2(i)<odds(j) & pick2(i)>odds(j-1))
      Dad(i)=j;
    end;
    if Dad(i)==0
      Dad(i)=1;
    end;
  end;
end;
```

Appendix C: Genetic Algorithm Run Parameters and Solutions

Parameter 1: P coefficient for the Pitch Controller

Parameter 2: I coefficient for the Pitch Controller

Parameter 3: P coefficient for the Pitch Compensator

Parameter 4: I coefficient for the Pitch Compensator

Parameter 5: P coefficient for the Droop/Governing Frequency Control

Parameter 6: Time Constant for WindINERTIA Low-Pass Filter

Parameter 7: P coefficient for WindINERTIA

Parameter 8: Time Constant for WindINERTIA Wash-Out Filter

The “TopChrom” variable is a vector of the optimal parameter found with each parameter indexed as the parameter number, for example, the second number in the TopChrom vector is Parameter 2. The “Ans” variable is the minimum value found on the cost function surface.

C.1 Pitch + Compensator

Maximum Number of iterations: 100
Population Size of Generation 0: 96
Population Size for Generations 1 - end: 48
Number of Chromosomes kept for mating: 24
Total Number of parameters in a chromosome: 4
Mutation rate: 0.04
High end of parameter 1 value: 800
High end of parameter 2 value: 100
High end of parameter 3 value: 50
High end of parameter 4 value: 25
Low end of parameter 1 value: 200
Low end of parameter 2 value: 25
Low end of parameter 3 value: 10
Low end of parameter 4 value: 5

TopChrom =

628.9598   71.3560   28.5594   19.4000

ans =

32.6321

C.2 PITCH + COMPENSATOR + DROOP

Maximum Number of iterations: 100
Population Size of Generation 0: 96
Population Size for Generations 1 - end: 48
Number of Chromosomes kept for mating: 24
Total Number of parameters in a chromosome: 5
Mutation rate: 0.04
High end of parameter 1 value: 800
High end of parameter 2 value: 100
High end of parameter 3 value: 75
High end of parameter 4 value: 50
High end of parameter 5 value: 100
Low end of parameter 1 value: 400
Low end of parameter 2 value: 25
Low end of parameter 3 value: 25
Low end of parameter 4 value: 10
Low end of parameter 5 value: 25

TopChrom =

480.6668   26.2039   35.2232   29.8399   87.2476

ans =

38.2842

C.3 PITCH + COMPENSATOR + DROOP + WIND INERTIA

Maximum Number of iterations: 100
Population Size of Generation 0: 96
Population Size for Generations 1 - end: 48
Number of Chromosomes kept for mating: 24

Total Number of parameters in a chromosome: 8
Mutation rate: 0.04
High end of parameter 1 value: 550
High end of parameter 2 value: 20
High end of parameter 3 value: 35
High end of parameter 4 value: 10
High end of parameter 5 value: 90
High end of parameter 6 value: 10
High end of parameter 7 value: 220
High end of parameter 8 value: 10
Low end of parameter 1 value: 200
Low end of parameter 2 value: 5
Low end of parameter 3 value: 5
Low end of parameter 4 value: 0
Low end of parameter 5 value: 20
Low end of parameter 6 value: 2
Low end of parameter 7 value: 150
Low end of parameter 8 value: 2

```
TopChrom =

  351.6254   16.6484   34.6131    9.2408   84.6891    7.2390   204.2616    9.3884


ans =

   29.7678
```