1-1-2010

Modeling of Doubly Fed Induction Generators for Distribution System Power Flow Analysis

Amitkumar Dadhania
Ryerson University

MODELING OF DOUBLY FED INDUCTION GENERATORS FOR DISTRIBUTION SYSTEM POWER FLOW ANALYSIS

by

Amitkumar Dadhania,

B.Eng., India, 1996

A thesis presented to

Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2011

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

(Amitkumar Dadhania)

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

(Amitkumar Dadhania)

MODELING OF DOUBLY FED INDUCTION GENERATORS FOR DISTRIBUTION SYSTEM POWER FLOW ANALYSIS

Amitkumar Dadhania
Master of Applied Science
Department of Electrical and Computer Engineering
Ryerson University, Toronto, 2011

ABSTRACT

Large-scale integration of Wind Generators (WGs) with distribution systems is underway right across the globe in a drive to harness green energy. The Doubly Fed Induction Generator (DFIG) is an important type of WG due to its robustness and versatility. Its accurate and efficient modeling is very important in distribution systems planning and analysis studies, as the older approximate representation method (the constant PQ model) is no longer sufficient given the scale of integration of WGs.

This thesis proposes a new three-phase model for the DFIG, compatible with unbalanced three-phase distribution systems, by deriving an analytical representation of its three major components, namely the wind turbine, the voltage source converter, and the wound-rotor induction machine. The proposed model has a set of nonlinear equations that yields the total three-phase active and reactive powers injected into the grid by the DFIG as a function of the grid voltage and wind turbine parameters. This proposed model is integrated with a three-phased unbalanced power flow method and reported in this thesis. The proposed method opens up a new way to conduct power flow studies on unbalanced distribution systems with WGs.

The proposed DFIG model is verified using Matlab-Simulink. IEEE 37-bus test system data from the IEEE Distribution System sub-committee is used to benchmark the results of the power flow method.

ACKNOWLEDGEMENT

During the period of my Masters Study, Ryerson University and all Professors from the Electrical Engineering department provided me enormous academic support. First of all I express my sincere appreciation to Ryerson University and its faculty members. In addition, Ryerson University Scholarship Program provided me financial support for the period of my Masters study. Therefore, I would like to express my thankfulness to Ryerson University for considering me in their scholarship program.

I would like to express deep gratitude to my supervisor Prof. Dr. Bala Venkatesh from Ryerson University and co-supervisor Dr. Vijay Sood from the UOIT, for their guidance, encouragement and valuable instructions throughout the period of this thesis preparation and the Masters degree in Ryerson University.

My special thanks go to Dr. Alexandre Nassif, Post Doctoral Fellow at Ryerson University for providing his valuable suggestions in improving this thesis. I would also like to acknowledge all moral support given by my friends from the Power and Energy Analysis Research Laboratory during the study.

Finally, I would like to thank my family members and friends, whose names are not mentioned above, for their unconditional encouragement and great help.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AC	Alternating current
CanWEA	Canadian Wind Energy Association
DC	Direct Current
DFIG	Doubly Fed Induction Generator
DG	Distributed Generation
d-q	Direct and Quadrature axis
DS	Distribution System
ERR	Error Value
GE	General Electric
IEEE	Institute of Electrical and Electronics Engineers
IGBT	Insulated Gate Bi-polar Junction Transistor
IT	Iteration Number
Imag	Imaginary Component
KVA	Kilo Volt Ampere
LF	Load Flow
MW	Mega Watts
PE	Power Electronics
PWM	Pulse Width Modulation
TOL	Tolerance
TS	Transmission System
VSC	Voltage Source Converter
WECS	Wind Energy Conversion Systems
WEG	Wind Electric Generator
WF	Wind Farm
WG	Wind Generator
WPP	Wind Power Plant
WRIM	Wound Rotor Induction Machine
WT	Wind Turbine
SCIM	Squirrel Cage Induction Machine
SG	Synchronous Generator
KCL	Kirchhoff Current Law
KVL	Kirchhoff Voltage Law

NOMENCLATURE

A	Swept area of the rotor
$\beta.$	Blade pitch angle
C_1 *to* C_6	Constant coefficients
Cp	Power coefficient
$\bar{E}_{012}$	Sequence induced voltages
E_{wind}	Energy available in the wind
$i=0,1,2$	i represents zero(0) , positive(1) and negative(2) sequence
I_{dc}	DC current
$\lvert\bar{I}_{s012}\rvert$and $\lvert\bar{I}_{r012}\rvert$	Magnitude of stator and rotor sequence current
k_r and k_g	Proportionality constants of modulation index
k_t	Speed transformation constant
L_{ml}	Magnetizing inductance
L_{sl} and L_{rl}	Stator and rotor leakage inductances
λ	Tip speed ratio
λ_i	Constant (relates to λ and Bitta)
M_r , M_g	Rotor and grid side VSC's PWM modulation indices.
P	Active Power
P_{abc}	Real power generated on each phase of DFIG bus
P_{gabc} and Q_{gabc}	Actual real and reactive power of grid side VSC
P_{gabc} and Q_{gabc}	Real and reactive power of GS-VSC
P_m	Mechanical power developed by wind turbine
P_{m012}	Sequence mechanical power
P_{ma} , P_{mb} , P_{mc}	Mechanical power for phase a, b and c
P_{r012}	At the rotor frequency, the real power flow through the rotor slip-rings to VSC in form of sequence component.
P_{rabc}	Actual active power of RS-VSC
P_{rcl012}	Rotor copper winding losses in form of sequence component
P_{s012}	At the stator frequency, the real power flow through the stator in form of sequence component.

P_{sabc} and Q_{sabc}	Actual active and reactive power on each phase of stator
P_{scl012}	Stator copper winding losses in form of sequence component
P_{wind}	Instantaneous wind power.
P_{tabc}	Grid-side VSC active power at PCC.
Q	Reactive Power
Q_{abc}	Real power generated on each phase of DFIG bus
Q_{gabc}	Reactive power controlled by GS-VSC
Q_{tabc}	Grid-side VSC reactive power at PCC.
$Q_{max\text{-}min}$	Specified reactive power limit
Q_{s012}	Sequence stator reactive power
Q_{sabc}	Actual Stator reactive power
R_s and R_r	Stator and rotor resistances
ρ	Air density
R_t	Radius to the tip of the rotor
s	Slip of the machine
s_0 , s_1 , s_2	Zero, positive and negative sequence slip
$s_i = s_{012}$	Slip in general form of sequence components.
V_{dc}	Constant dc voltage
$\lvert \bar{V}_{gabc}\rvert\angle\delta_{gabc}$	Grid-side VSC output AC voltage magnitude and angle
$\lvert \bar{V}_{sabc}\rvert\angle\delta_{sabc}$	Stator voltage magnitude and angle
$\lvert \bar{V}_{rabc}\rvert\angle\delta_{rabc}$	Rotor-side VSC output AC voltage magnitude and angle
$\lvert V_{ga}\rvert \angle\delta_{ga}$	Phase-a grid side VSC output voltage magnitude and angle
$\lvert V_{gb}\rvert \angle\delta_{gb}$	Phase-b grid side VSC output voltage magnitude and angle
$\lvert V_{gc}\rvert \angle\delta_{gc}$	Phase-c grid side VSC output voltage magnitude and angle
$\bar{V}_{gabc}$	Grid-side VSC output voltage
$\bar{V}_{rabc}$ and $\bar{I}_{rabc}$	Output voltage and current flow of Rotor-side VSC
$\bar{V}_{sabc}$	Stator output voltages/grid voltages
$[\bar{V}_{LL012}]$	Generalized form of sequence line-line voltage matrices
$[\bar{V}_{LLabc}]$	Generalized form of actual line-line voltage matrices
$[\bar{V}_{LN012}]$	Generalized sequence line-neutral voltage matrices
$\bar{V}_{r012}$ and $\bar{I}_{r012}$	Sequence rotor line to neutral voltages and currents
$\bar{V}_{s012}$and $\bar{I}_{s012}$	Sequence stator line to neutral voltages and currents

V_{sLLabc}	Actual line-to-line stator/grid voltages
ω_s and ω_r	Synchronous and rotor angular speeds
ω_t	Turbine rotor speed
ω_{wind}	Free wind speed
Z_g	Impedance (combine)
Z_{kg}	Impedance of grid-side VSC
Z_{kr}	Impedance of rotor-side VSC
Z_{s012} , Z_{r012} , Z_{m012}	Sequence stator, rotor and magnetizing impedances
Z_{tr}	Transformer impedance

Chapter 1
Introduction

The first section of this chapter provides a brief background of wind electric generators and their scale of integration in power systems. The second section reviews the relevant literature and provides the logic behind the ideas and applications proposed in this thesis. The third section describes the scope and objective of this research and presents the thesis outline.

1.1 Background

The emerging awareness for environmental preservation concurrent with the increasingly power demand have become commonplace to utilities. To satisfy both these conflicting requirements, utilities have focused on the reduction of high-polluting sources of energy. The desire to seek alternative renewable energy resources has led to the widespread development of distributed generations (DGs). In many countries, wind electric generators (WEGs) are becoming the main renewable source of electric energy.

According to the Canadian Wind Energy Association (CanWEA), wind energy will contribute to an amount of 5-10% of Canada's total electricity supply by 2010 [24]. In other words, it is anticipated that the wind power participation will be in excess of 10,000 MW. Such trend is not unique in Canada. Take as an example the scenario in Denmark, which today has a wind participation that accounts for over 20% of its total produced power. CanWEA foresees this figure as an attainable goal for Canada. If Canada is able to generate 20% of its electricity from wind energy, wind power would be the second largest source of electricity behind hydro and ahead of coal, natural gas and nuclear [24].

An individually installed WEG is commonly referred to as a wind turbine (WT) or simply as a wind generator (WG), and a group of such generators is referred to as a wind power plant (WPP) or wind farm (WF). Wind farms of all sizes are continuously being connected directly to the power grids and they have the potential to replace many of the conventional power plants. This means that wind turbines should possess the general characteristics of conventional power plants such as simplicity of use, long and reliable useful life, low maintenance and low initial cost. Moreover, large wind farms should satisfy very demanding technical requirements such as frequency and voltage control, active and reactive power regulation, and fast response during transient and dynamic situations [14].

In an attempt to satisfy the above requirements, many topologies of WGs have been proposed. WGs can either operate at fixed speed or at variable speed. Due to various reasons such as reduced mechanical stress, flexible active-reactive power controllability, good power quality, low converter rating and low losses, nowadays the most popular topology is the variable speed type Double Fed Induction Generator (DFIG) [1]-[2]. The DFIG has been modeled using several techniques. However, the research proposed so far has shown limitations, as explained in the following section.

1.2 Review of Related Research

Even though the DFIG type WG is a very complex structure, it has been traditionally modeled by a constant PQ or PV bus in Power Flow studies. The DFIG was modeled as a PQ load bus when operated in the power factor controlled mode, which means that the specified reactive power is zero. Alternatively, the DFIG was modeled as a PV bus when operated in a voltage controlled mode, which means that the specified reactive power limit is applied [4]-[5]. Whereas such modeling techniques were considered satisfactory at the time DGs when first integrated in Power Flow studies, they are clearly inadequate to accurately represent the generator behavior. It also became apparent that the situation would be even more critical when unbalances were present.

In a further development [5], J. F. M. Padrón and A. E. F. Lorenzo proposed a steady-state model for the DFIG consisting of a fixed PQ and PV bus model. The authors limited their study to a single phase model and solution. In [6], A. E. F. Lorenzo and J. Cidras, as well as U. Eminoglu in [8], used the conventional fixed speed induction motor type WT topology to propose a single phase RX bus model. In the sequence, their models were compared with the (at that time) conventional fixed PQ bus model. In spite of providing some improvement, this RX modeling approach is complex and fairly difficult to implement. In addition, it is difficult to validate this model for three phase DFIGs integrated to distribution systems.

These researches improved addressing the scenario where DGs are connected to transmission systems by representing the control action of the variable-speed generators, but they were still not accurate enough to completely account for the unbalances. Nowadays distribution-connected DGs have become prevalent and have by far exceeded the amount of transmission-connected DGs. Distribution systems are much more

unbalanced than transmission systems, and the fixed, balanced, single phase approaches used for transmission systems are not suitable and precise for distribution systems.

Subsequently, M. Zhao, Z. Chen and F. Blaabjerg presented several models for variable speed WTs, which were developed for power flow integration in distribution systems [9]. They modeled the variable speed WTs as a fixed balanced PQ bus when solving the power flow. Therefore, similarly to previously published papers, this technique still lacked modeling accuracy, as it considered a fixed PQ bus to represent the WG. In addition, due to its complexity, it is difficult to analyze and validate.

During the preparation of his thesis work, this researcher has conducted an extensive literature search and has found several research papers where the DFIG model was developed for dynamic and transient purposes only. However, these models are not suitable for integration with power flow algorithms and are difficult to validate, as they belong to a different context.

1.3 Motivation of this research

Figure 1.1 represents part of a distribution system where a DFIG is connected on the ith bus.

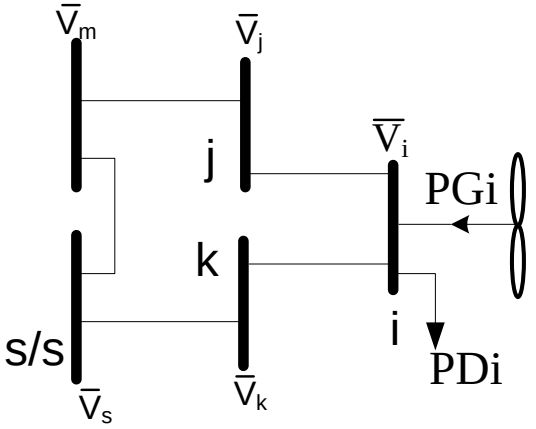

Figure 1.1: DFIG integrated distribution system

If the representation of this DFIG follows the traditional modeling approach (fixed *P* and *Q*, the active power balance equation of the ith bus (DFIG-bus) can be represented by

$$PG_i - PD_i - P_i(V, \delta) = 0,$$

where PG_i is the generated power by the DFIG independently on the ith bus voltage during each iteration of the power flow analysis.

As a common practice distribution system lines are never transposed and inherently unbalanced. In reality the DFIG's stator is directly connected to the ith bus via a step-up transformer. Hence, $PG_i = f(\overline{V}_i, \omega_{wind})$ where $\overline{V}_i$ is the DFIG (ith) bus voltage and ω_{wind} is the wind speed. Therefore, and further to the conclusions presented in the previous section, this type of fixed PQ modeling technique is inadequate to represent the DFIG's behavior and cannot provide an accurate voltage solution. With the increasing number of DFIGs being connected to distribution systems, a more accurate three-phase model of DFIGs is urgently needed to distinguish the complete state of the DFIG and which provides more reliable voltage solutions of unbalanced distribution systems that connects them. Therefore, in order to obtain a precise voltage solution during this research, this researcher proposes a new generic three phase active model of the DFIG, which can be accounted for in the following power balance equation in the distribution system load flow analysis (the reactive power balance equation can be obtained via a similar relationship):

$$PG_i(\overline{V}_i, \omega_{wind}) - PD_i - P_i(V, \delta) = 0.$$

1.4 Objective and Contributions of this research and Thesis Outline

Due to the inherently unbalanced nature of distribution systems, previous approaches using single/balanced phase(s) are inaccurate. For the same reason, fixed PQ models for the DFIG cannot provide an accurate power flow solutions for distribution system or unbalanced transmission system. With the widespread installation of DFIGs and their increasing capacity, the response of DFIGs to grid disturbances has become an important issue. As a result, it is very important for utilities to analyze the unbalanced voltage profiles more precisely during power flow analysis. In this thesis scope of work, this translates into adopting a more accurate technique to model DFIGs to increase the reliability of power flow solutions.

The main objective of this thesis is to propose a new model for the DFIG by deriving an analytical representation of its three major components. Detailed models of the wind turbine, the VSC, and the WRIM were developed. In order to reach a solution for the actual active and reactive powers injected by the generator, an iterative approach was adopted. The resulting model was validated with time-domain simulations carried out with Matlab-Simulink. The model is shown to be very accurate and it can be easily integrated in Power Flow programs. The developed model was then incorporated into a three-phase Power Flow program to solve a typical distribution system. A ladder iterative solution was

used. The IEEE 37-bus unbalanced distribution system was used to benchmark the Power Flow methods. The obtained results clearly indicate that significantly more accurate results are obtained with the proposed modeling of the DFIG.

In order to get more precise voltage solution of the DFIG integrated distribution system power flow analysis, the main contributions of this research are those presented in Chapters 3 and 4. **The structure of this thesis** is as follows:

Chapter 2 presents the wind energy conversion system along with its power controls. A detailed analysis of the DFIG topology is presented.

Chapter 3 presents the modeling of all DFIG elements and the proposed algorithm to obtain its complete model. This model is validated through time-domain simulations in Matlab-Simulink.

Chapter 4 presents the Power Flow solution using the ladder iterative technique with the integration of both (1) the traditional DFIG model, and (2) the proposed DFIG model with the IEEE-37 bus distribution system. Finally the resulting errors from using the traditional DFIG model are quantified by comparing the results from both models.

Chapter 5 presents the conclusions and contributions of this thesis, as well as suggestions for future work.

Chapter 2
Wind Energy Systems

This chapter reviews the concepts of wind energy conversion system and its basic process of wind energy extraction, conversion and power regulation. Conversion systems can apply different topologies of wind generators. Since the Double Fed Induction Generator is the focus of this thesis, this chapter presents a detailed analysis of the topology.

2.1 Wind Energy Conversion Systems

Figure 2.1 represents the complete wind energy conversion systems (WECS), which converts the energy present in the moving air (wind) to electric energy.

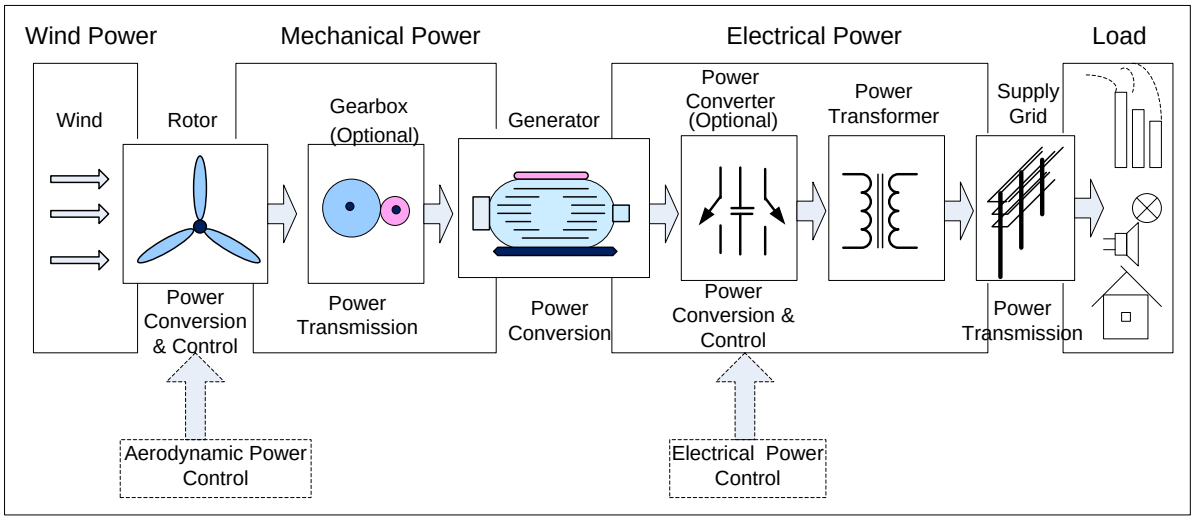

Figure 2.1: Wind Energy Conversion Systems

The wind passing through the blades of the wind turbine generates a force that turns the turbine shaft. The rotational shaft turns the rotor of an electric generator, which converts mechanical power into electric power. The major components of a typical wind energy conversion system include the wind turbine, generator, interconnection apparatus and control systems.

The power developed by the wind turbine mainly depends on the wind speed, swept area of the turbine blade, density of the air, rotational speed of the turbine and the type of connected electric machine.

As shown in figure 2.1, there are primarily two ways to control the WECS. The first is the Aerodynamic power control at either the Wind Turbine blade or nacelle, and the second is the electric power control at an interconnected apparatus, e.g., the power electronics converters. The flexibility achieved by these two control options facilitates extracting maximum power from the wind during low wind speeds and reducing the mechanical stress on the wind turbine during high wind speeds. Both these control methods are presented next.

2.1.1 Aerodynamic Power Control and Power Curve

The key idea behind the aerodynamic control is the utilization of the power curve. The power curve is a piece of information usually provided by the turbine manufacturer that describes the performance of the wind turbine at each wind speed. Maximum mechanical power can be achieved by controlling the wind turbine as constrained by the power curve.

Figure 2.2 shows an example of power curve. The curve displays the turbine mechanical power as a function of turbine speed, for wind speeds ranging from 5 m/s to 16.2 m/s. To achieve maximum power from the wind turbine, the WT is controlled in order to follow the thick (0-A-B-C-D) curve.

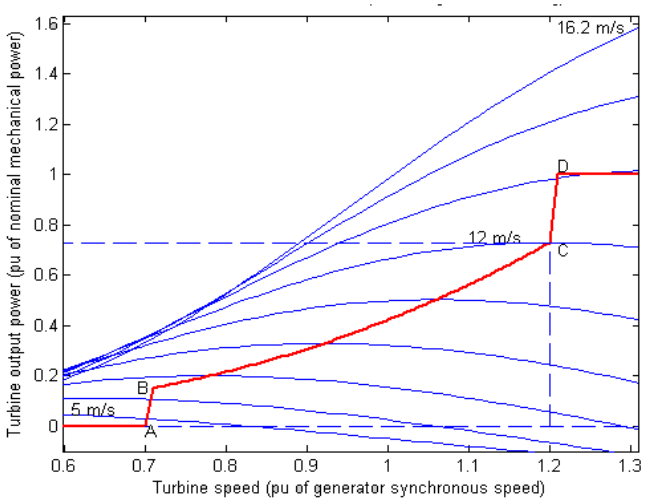

Figure 2.2: Sample power curve

Below the cut-in wind speed (< 5m/s – point A), the power in the wind is too low for useful energy production and so the wind turbine remains shut down. At higher wind speeds but below the rated wind speed (i.e., between B and C), the wind turbine power output increases due to a cubic relationship with wind speed. In this range, the turbine is controlled in order to extract the maximum power from the wind passing across the rotor disc. Between the rated wind speed and the maximum operating wind speed (i.e., between C and D), the

aerodynamic rotor is arranged to control the mechanical power extracted from the wind, i.e., the mechanical power on the rotor shaft is intentionally reduced in order to reduce the mechanical load/stress on the turbine. Finally, at very high wind speeds (beyond point D), the turbine is shut down to avoid damage. Therefore, in this curve point A is referred to as the cut-in speed and point D is referred to as the cut-out speed.

In summary, the aerodynamic wind power control is essentially intended to control the input power of the wind turbine.

There are three ways to perform aerodynamic power control.

1. Pitch Control: The blades are physically rotated around their longitudinal axis.
2. Stall Control: The angle of the blade is fixed, but the aerodynamic performance of the design is such that at high wind speeds the blades stall.
3. Yaw Control: In this technique the entire nacelle is rotated around the tower to yaw (oscillate around a vertical axis) the rotor out of the wind. Due to its complexity and susceptibility to stress, this technique is not commonly used.

Currently, Pitch Control is the most common method for aerodynamic control. Almost all variable speed wind turbine topologies (including the DFIG) use Pitch Control. At wind speeds below the rated speed, it is used to maximize the energy capture. At wind speeds above the rated speed, it is used to reduce the mechanical stress on the system.

2.1.2 Electrical Power Control and Wind Electric Generators

Depending on the type of power electronic apparatus used in the WG topology, on the desired electric output power, and on the control scheme, a WG can be operated at either fixed speed or variable speed:

(1) **Fixed speed:** This category of WG is not controlled by any interconnected power electronics device and is typically composed of small to medium size wind turbines. Permanent magnet synchronous or squirrel-cage induction generators are often used because of their reliability and cost. They are directly connected to the grid and employ stall control of the turbine blades. The speed variation from no load to full load is very small, i.e., almost fixed, so this topology is also referred to as “fixed” speed WG. Because this generator operates at nearly fixed speed (driven by the grid frequency), it yields variations of the output power according to the wind speed. Therefore, large WG power output can cause the grid voltage to experience fluctuations, especially if connected to weak AC systems.

For this reason, with increased generation sizes (MW-level), variable speeds WGs have become prevalent.

(2) **Variable speed:** This type of WG is regulated externally by interconnected power electronics converters or similar apparatus to realize power control, soft start and interconnection functions.

Variable speed high power wind turbines can apply squirrel cage or wound rotor type induction generators, as well as permanent magnet synchronous generators or wound field synchronous generators. They are typically equipped with forced commutated PWM inverters/rectifiers to provide a fixed voltage and fixed frequency and apply Pitch Control of the turbine blades. Nowadays, effective power control can be achieved in some wind turbines by using double PWM (pulse-width modulation) converters, which provides a bidirectional high quality power flow between the WG and the utility grid. These types of wind turbine can generate more energy for a given wind speed. Active and reactive power can be easily controlled by these converters.

Depending on the connection of their power electronics apparatus, these types of WGs can be categorized as single fed or doubly fed types.

The single fed variable speed approach consists of a converter connected in series between the generator and the grid that allows a unidirectional power flow. The converters must withstand the full power rating of the generator, representing an increase in cost and losses.

The doubly fed approach is an alternative to the single fed approach. Currently, many manufacturers have adopted this technology and are producing wind turbines which are coupled to doubly fed induction machines, e.g., DeWind, GE Wind Energy, Nordex and Vestas. In this topology, the power captured by the wind turbine is converted into electrical power by the wound rotor induction machine (WRIM). This power is transmitted to the grid by both the stator (directly) and the rotor windings (via power electronics converters). Therefore, due to the feature of double sided power transfer to the grid, this type of wind turbine is referred to as the doubly fed Induction generator (DFIG). The DFIG has been the most popular option for wind power generation applications. Next section analyzes its structure and operating principle.

2.2 Doubly Fed Induction Generator

Figure 2.3 presents the topology of the DFIG, which will be thoroughly analyzed in this section.

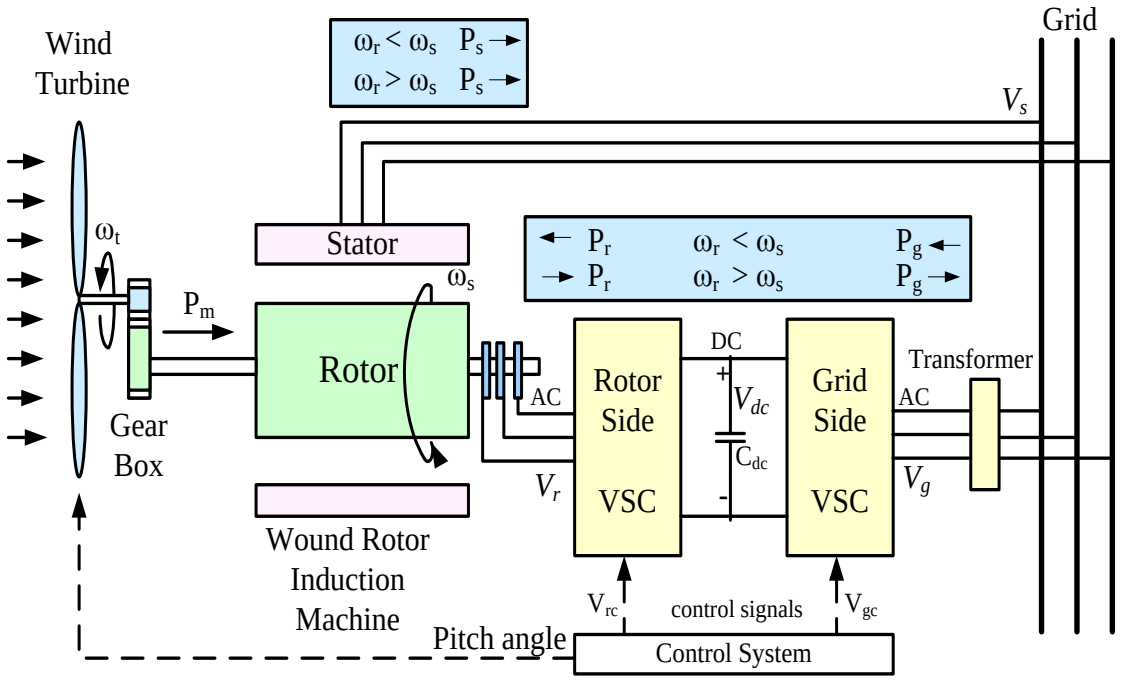

Figure 2.3: Doubly Fed Induction Generator type WT

2.2.1 Structure

As shown in figure 2.3, the DFIG consists of two bi-directional voltage source converters with a back-to-back DC-link, a wound rotor induction machine, and the wind turbine.

Wound Rotor Induction Machine:

The WRIM is a conventional 3-phase wound rotor induction machine. The machine stator winding is directly connected to the grid and the rotor winding is connected to the rotor-side VSC by slip rings and brushes.

Voltage Source Converters:

This type of machine is equipped with two identical VSCs. These converters typically employ IGBTs in their design.

The AC excitation is supplied through both the grid-side VSC and the rotor-side VSC. The grid side VSC is connected the ac network. The rotor side converter is connected to the rotor windings. This grid side VSC and the stator are connected to the ac grid via step up transformer to elevate the voltage to the desired grid high voltage level.

The VSCs allow a wide range of variable speed operation of the WRIM. If the operational speed range is small, then less power has to be handled by the bi-directional power converter connected to the rotor. If the speed variation is controlled between +/- 30 %, then the converter must have a rating of approximately 30 % of the generator rating. Thus the required converter rating is significantly smaller than the total generator power, but it depends on the selected variable speed range and hence the slip power [17]. Therefore, the size and cost of the power converter increases when the allowable speed range around the synchronous speed increases.

DC-link with Capacitor:

The capacitor connected to the DC-link acts as a constant, ripple-free DC voltage source, an energy storage device and a source of reactive power. Moreover, the DC-link provides power transmission and stabilization between both unsynchronized AC systems.

Control System:

The control system generates the following commands: the pitch angle command, which is used by the aerodynamic Pitch Control to control the wind power extracted by turbine blades; the voltage command signal V_{rc}, which is intended to control the rotor side VSC; and the signal V_{gc} , which is intended to control the grid side VSC (to control the electrical power) [3]. In turn, the rotor-side VSC controls the power of the wind turbine, and the grid-side VSC controls the dc-bus voltage and the reactive power at the grid terminals.

By implementing pulse width modulation, it is possible to control the VSCs to generate an output waveform with desired phase angle and voltage magnitude, and at the same time reduce lower order harmonics [9].

2.2.2 Operating Principle

A wide range of variable speed operating mode can be achieved by applying a controllable voltage across the rotor terminals. This is done through the rotor-side VSC. The applied rotor voltage can be varied in both magnitude and phase by the converter controller, which controls the rotor currents. The rotor side VSC changes the magnitude and angle of the applied voltages and hence decoupled control of real and reactive power can be achieved.

The rotor-side VSC controller provides two important functions:

- Variation of generator electromagnetic torque and hence rotor speed.

- Constant stator reactive power output control, stator power factor control or stator terminal voltage control.

The grid-side VSC controller provides:

- Regulation of the voltage of the DC bus capacitor.
- Control of the grid reactive power.

The DFIG exchanges power with the grid when operating in either sub or super synchronous speeds. These operating modes are analyzed as follows.

Power flow/Operating modes:

The DFIG stator is connected to the grid with fixed grid frequency (f_s) at fixed grid voltage (V_s) to generate constant frequency AC Power during all operating conditions and the rotor is connected to the frequency converter/VSC having a variable (slip/rotor) frequency ($f_r = s \, . \, f_s$). At constant frequency f_s, the magnetic field produced in the stator rotates at constant angular velocity/speed ($\omega_s = 2 \, \pi \, f_s$), which is the synchronous speed of the machine. The stator rotating magnetic field will induce a voltage between the terminals of the rotor. This induced rotor voltage produces a rotor current (I_r), which in turn produces a rotor magnetic field that rotates at variable angular velocity/speed ($\omega_r = 2 \, \pi \, f_r$). Usually the stator and rotor have the same number of poles (P) and the convention is that the stator magnetic field rotates clockwise. Therefore, the stator magnetic field rotates clockwise at a fixed constant speed of $\omega_s \, (rpm) = 120 \, f_s \, / \, P$. Since the rotor is connected to the variable frequency VSC, the rotor magnetic field also rotates at a speed of $\omega_r \, (rpm) = 120 \, f_r \, / \, P$.

Sub-synchronous speed mode:

Figure 2.4 illustrates the case where the rotor magnetic field rotates at a slower speed than the stator magnetic field.

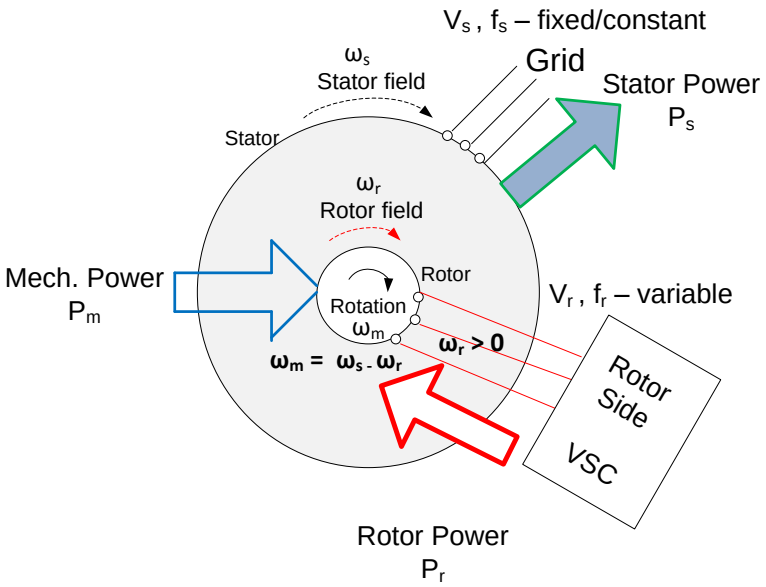

Figure 2.4: Sub-synchronous operating mode of DFIG

The machine is operated in the sub-synchronous mode, i.e., $\omega_m < \omega_s$,

- if and only if its speed is exactly $\omega_m = \omega_s - \omega_r > 0$, and
- both the phase sequences of the rotor and stator mmf's are the same and in the positive direction, as referred to as as positive phase sequence ($\omega_r > 0$) [6].

This condition takes place during slow wind speeds. In order to extract maximum power from the wind turbine, the following conditions should be satisfied:

- The rotor side VSC shall provide low frequency AC current (negative V_r will apply) for the rotor winding.
- The rotor power shall be supplied by the DC bus capacitor via the rotor side VSC, which tends to decrease the DC bus voltage. The grid side VSC increases/controls this DC voltage and tends to keep it constant. Power is absorbed from the grid via the grid side VSC and delivered to the rotor via the rotor side VSC. During this operating mode, the grid side VSC operates as a rectifier and rotor side VSC operates as an inverter. Hence power is delivered to the grid by the stator.
- The rotor power is capacitive [22].

Super-synchronous speed mode:

The super-synchronous speed mode is achieved by having the rotor magnetic field rotate counterclockwise. Figure 2.5 represents this scenario. However, in order to represent the

counterclockwise rotation of the rotor, which is analytically equivalent to inverting the direction of the rotor magnetic field.

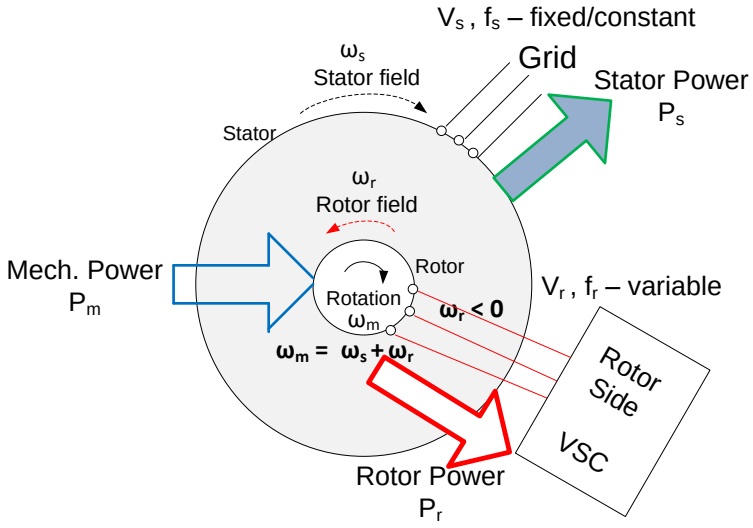

Figure 2.5: Super-synchronous operating mode of DFIG

The machine is operated in the super-synchronous mode, i.e., $\omega_m > \omega_s$,

- if and only if its speed is exactly $\omega_m = \omega_s - (-\omega_r) = \omega_s + \omega_r > 0$, and
- the phase sequence in the rotor rotates in opposite direction to that of the stator, i.e., negative phase sequence ($\omega_r < 0$) [6].

This condition takes place during the condition of high wind speeds. The following conditions need to be satisfied in order to extract maximum power from the wind turbine and to reduce mechanical stress:

- The rotor winding delivers AC power to the power grid through the VSCs.
- The rotor power is transmitted to DC bus capacitor, which tends to raise the DC voltage [22]. The grid side VSC reduces/controls this DC-link voltage and tends to keep it constant. Power is extracted from the rotor side VSC and delivered to the grid. During this operating mode, the rotor side VSC operates as a rectifier and the grid side VSC operates as an inverter. Hence power is delivered to the grid directly by the stator and via the VSCs by the rotor.
- The rotor power is inductive. [22]

Synchronous speed mode:

The synchronous speed mode is represented by figure 2.6.

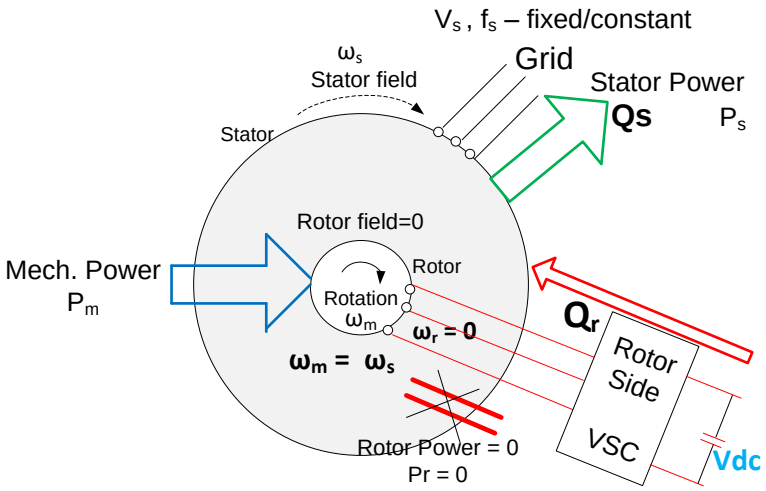

Figure 2.6: Synchronous operating mode of DFIG

The machine is operated in the synchronous speed mode, i.e., $\omega_m = \omega_s$,

- if and only if its speed is exactly $\omega_m = \omega_s - 0 = \omega_s > 0$, and
- the phase sequence in the rotor is the same as that of the stator, but no rotor mmf is produced ($\omega_r = 0$).

The following conditions are necessary in order to extract maximum power from the wind turbine under this condition:

- The rotor side converter shall provide DC excitation for the rotor, so that the generator operates as a synchronous machine.
- The rotor side VSC will not provide any kind of AC current/power for the rotor winding. Hence the rotor power is zero ($P_r = 0$).
- A substantial amount of reactive power can still be provided to the grid by the stator [20].

As per the operating modes described above, at any wind speeds a wide range of variable speed operation can be performed to achieve maximum wind power extraction.

Chapter 3
Proposed Model of DFIG type WG

The complete DFIG structure is modeled and validated in this chapter. Section 3.1 presents the modeling technique and Section 3.2 presents the validation of the model.

3.1 Proposed Algorithm of DFIG Modeling

As shown in figure 3.1, the complete modeling of the three-phase DFIG incorporates three sub-models: (1) a Wind Turbine, (2) two VSCs connected back-to-back via a DC-link, and (3) a three phase WRIM.

The three following sub-sections describe each model and the fourth sub-section presents the development of the proposed algorithm for obtaining the full DFIG model.

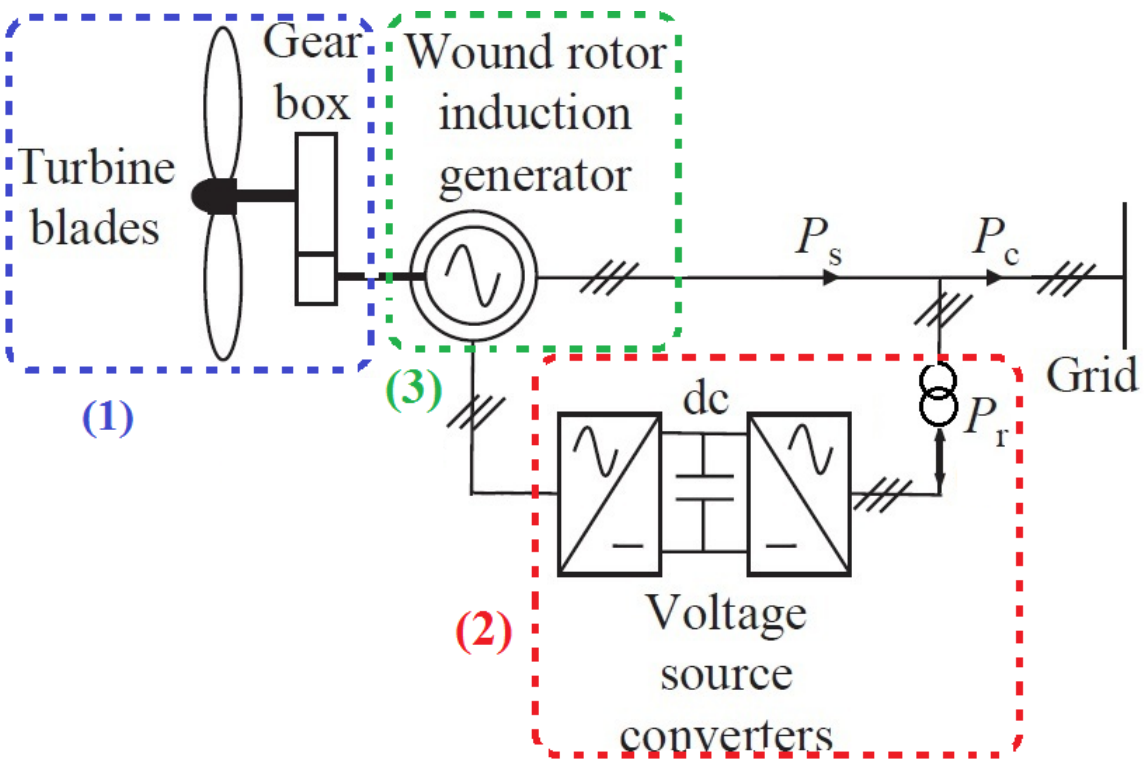

Figure 3.1: DFIG type WG with sub-models

3.1.1 Wind Turbine Model

The concept and modeling approach of this model is well-known. The wind turbine extracts wind energy from the swept area of the rotor disc and converts it into electrical energy. The energy available in the wind is given by

$$E_{wind}(joules) = \frac{1}{2}\rho\, A\, \omega_{wind}^3\, t\ = P_{wind}.t, \tag{1}$$

where ρ is the air density, A is the swept area of the rotor (m^2), and ω_{wind} is the free wind speed (m/s), t is the time (sec) and P_{wind} is the instantaneous wind power.

Not all available wind power (P_{wind}) can be converted to mechanical power (P_m) by the turbine blades. As per Betz theorem [12], the power coefficient (C_P) determines the maximum power that can be extracted from the wind flow and is defined as

$$C_p\ = \frac{P_m}{P_{wind}} < 59.3\%. \tag{2}$$

Therefore, the actual mechanical power captured by the wind turbine can be obtained by

$$P_m = \frac{1}{2}\,\rho\, A\, \omega_{wind}^3\, C_p. \tag{3}$$

Knowing the value of wind speed (ω_{wind}), one may determine the from manufacturer supplied curves (*msc1*), the value of tip speed ratio (λ) that gives the highest value of power coefficient C_p. Further, from manufacturer supplied curves (*msc2*), one may also determine the value of blade pitch angle β for a chosen value of λ such that power output of the turbine is maximum. Therefore, these manufactured supplied curves (msc) give:

$$\lambda = msc1(\omega_{wind}) \tag{4}$$

$$\beta = msc2(\lambda) \tag{5}$$

C_p is a function of the tip speed ratio (λ) and of the blade pitch angle (β), and is defined as

$$C_p(\lambda,\beta) = C_1\left(\frac{C_2}{\lambda_i} - C_3\beta - C_4\right)\mathrm{e}^{\frac{-C_5}{\lambda_i}} + C_6\lambda, \tag{6}$$

where C_1 to C_6 are constant coefficients given by C_1=0.5176, C_2=116, C_3=0.4, C_4=5, C_5=21, C_6 =0.0068. λ_i relates to λ and β through the following relationship:

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1}. \quad (7)$$

Knowing λ, one may determine the value of turbine speed ω_t that extracts maximum wind power as below:

$$\omega_t = \frac{\lambda \,.\, \omega_{wind}}{R_t}, \quad (8)$$

where R_t is the radius to the tip of the rotor.

Summarizing, by knowing the wind speed (ω_{wind}), one may determine the values of λ and β from the manufacturer supplied curves for maximum extraction of wind power. By substituting the values of λ and β in (6) and (7) one may find C_p. In turn when this C_p value is used (3), one can obtain the maximum mechanical power P_m developed by the wind turbine for any particular value of wind speed ω_{wind} and corresponding value of turbine speed ω_t is obtained from (8) [3].

The convention adopted in this thesis assumes a negative value for the generated power and a positive value for the consumed power. Therefore, in wind turbine model the produced mechanical power P_m and the developed mechanical torque T_m are negative.

The complete modeling of the wind turbine can be represented in the flowchart shown in figure 3.2. The wind turbine model and manufacturers supplied curves used during this research work were developed by Hydro Quebec [25]-[26] and provided by Mathworks in the Matlab-Simulink library [3].

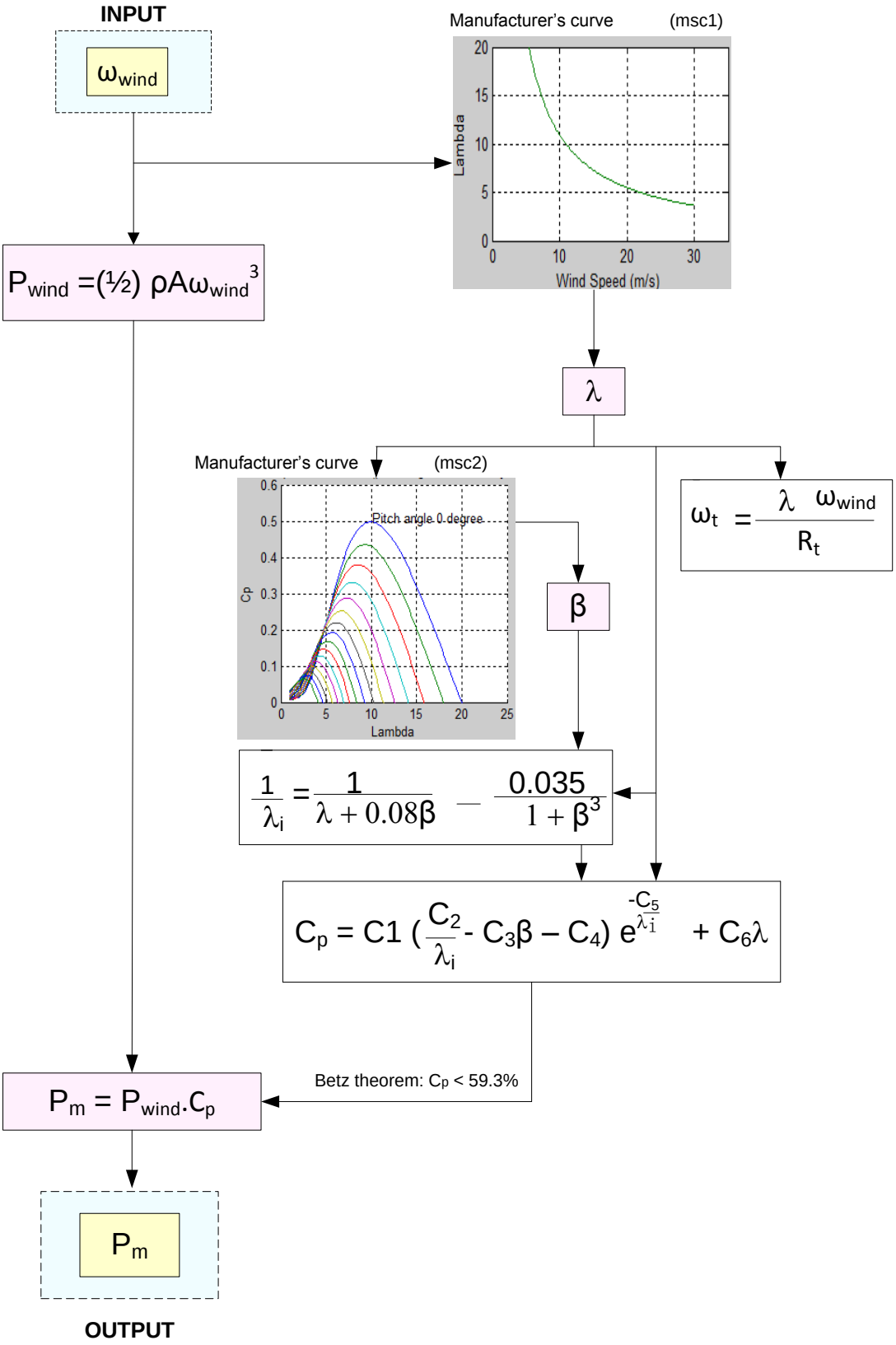

Figure 3.2: Flow chart of proposed wind turbine model

3.1.2 VSC with DC link Model

The VSC model used in this thesis was derived from that used for HVDC converters presented in [13]-[17]. This section presents the simplifications conducted in this thesis to achieve the model and the functionality of the VSC in DFIG. As presented in figure 3.1, The DFIG uses a pair of VSCs connected back-to-back through a DC link. The rotor-side VSC is connected to the WRIM rotor winding and the grid-side VSC is connected through a transformer to the stator terminals.

Considering a simplified version of average model, the internal switching actions and topological structure of the VSC are neglected [13]-[15]. Figure 3.3 represents the complete equivalent circuit model of both VSCs and the DC link in between them. In this model of the rotor-side VSC, when viewed from the AC side, it is represented by a controlled voltage source $\bar{V}_{rabc}$ (9) behind an appropriately valued impedance Z_{kr}.

$$\bar{V}_{rabc} = |\bar{V}_{rabc}| . (cos\, \delta_{rabc} + j\, sin\, \delta_{rabc}), \tag{9}$$

In this model of the grid-side VSC, when viewed from the AC side, it is represented by a controlled voltage source $\bar{V}_{gabc}$ (10) behind an appropriately valued impedance Z_{kg}.

$$\bar{V}_{gabc} = |\bar{V}_{gabc}| . (cos\, \delta_{gabc} + j\, sin\, \delta_{gabc}), \tag{10}$$

The VSC's when seen from the DC side, they are represented by a controlled current source I_{dc}. Therefore, in this representation, each VSC has two separate parts: a three phase controlled voltage source and a controlled current source [13]-[17].

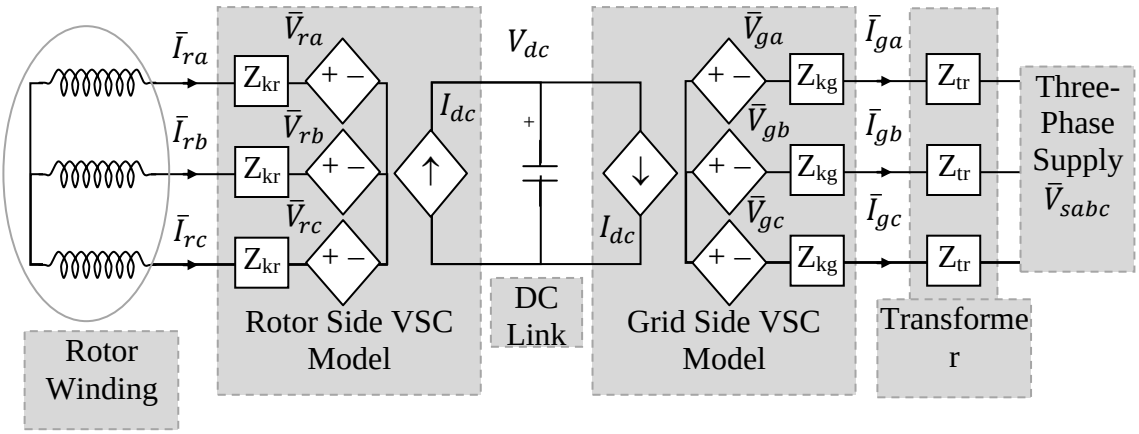

Figure 3.3: Equivalent circuit of VSC with DC link model in DFIG

In figure 3.3, ignoring the shunt elements of the equivalent circuit of a transformer, it is represented by series impedance of Z_{tr}.

In the VSC models, δ_{gabc} and δ_{rabc} are the phase angles of the fundamental components of the pulse width modulation (PWM) waveforms. $|\bar{V}_{gabc}|$ and $|\bar{V}_{rabc}|$ are controlled by the modulation indices (M_g and M_r) of the PWMs. In both VSCs, M_g and M_r are defined as the ratio of AC-side VSC's fundamental component of the output voltages $|\bar{V}_{gabc}|$ and $|\bar{V}_{rabc}|$ respectively to the constant dc input voltage (V_{dc}) of the inverter as defined below:

$$M_g = k_g \frac{|\bar{V}_{gabc}|}{V_{dc}}, \tag{11}$$

$$M_r = k_r \frac{|\bar{V}_{rabc}|}{V_{dc}}, \tag{12}$$

where k_g and k_r are proportionality constants. The controllers are designed to keep the DC link voltage (V_{dc}) constant. Thus, the desired output AC voltage can be accomplished by the PWMs by controlling the modulation indices and phase angles. Both VSCs operate in a complementary manner and therefore, the power flow model of the VSCs with DC link is based on the use of one controllable voltage source for the rectifier (modeled as (9)) and another controllable voltage source for the inverter (modeled as (10)) and vice versa[14]-[15]. The link is assumed lossless.

In this thesis, it is proposed to simplify the model of figure 3.3 further to the model as shown as in figure 3.4 and it is found adequate for power flow studies. The ratings of the VSCs and the dc-link are typically less than 30% of the total rating of the DFIG.

Some losses of converters and dc-link can be assumed negligible without noticeable loss of accuracy. In the simplified circuit shown in figure 3.4, the impedance Z_g comprises the resistance that accounts for losses in the rotor-side VSC, grid-side VSC, DC-link and transformer. The inductance of Z_g comprises the reactance of the grid-side converter and transformer.

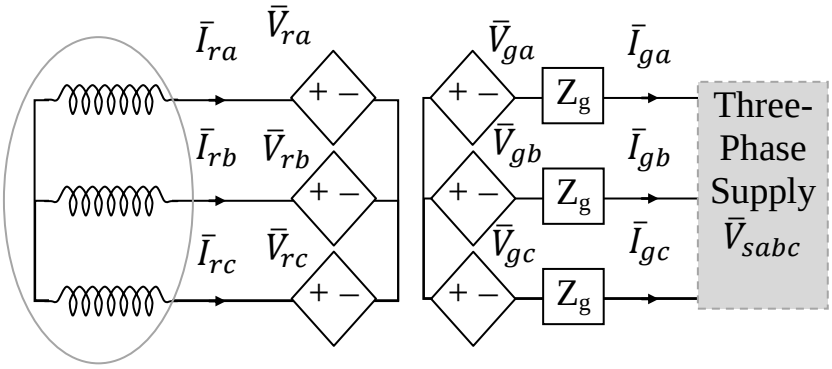

Figure 3.4: Proposed average model of VSCs in DFIG.

The real and reactive power output of the grid side VSC before Z_g is given by

$$P_{gabc} = real\left\{ \bar{V}_{gabc}.\left(\frac{\bar{V}_{gabc} - \bar{V}_{sabc}}{Z_g} \right)^{*} \right\},$$

and

$$Q_{gabc} = imag\left\{ \bar{V}_{gabc}.\left(\frac{\bar{V}_{gabc} - \bar{V}_{sabc}}{Z_g} \right)^{*} \right\}, \tag{13}$$

The DC link is assumed to be lossless [14]-[15]. Therefore,

$$P_{gabc} = P_{rabc}. \tag{14}$$

where the real power output of the rotor-side VSC is P_{rabc}.

The real and reactive power output of the grid side VSC reaching the PCC (point of common coupling) is given by

$$P_{tabc} = real\left\{ \bar{V}_{sabc}.\left(\frac{\bar{V}_{sabc} - \bar{V}_{gabc}}{Z_g} \right)^{*} \right\},$$

and

$$Q_{tabc} = imag\left\{ \bar{V}_{sabc}.\left(\frac{\bar{V}_{sabc} - \bar{V}_{gabc}}{Z_g} \right)^{*} \right\}, \tag{15}$$

3.1.3 Three Phase WRIM Model

It is proposed to model the three phase WRIM operating under unbalanced voltage conditions by using symmetrical components. Since the WRIM is delta or ungrounded star connected, there will not be any zero sequence current flow. Therefore, the zero sequence equivalent circuit networks ($i = 0$) are omitted in the modeling [18], [19].

The proposed positive ($i = 1$) - and negative-sequence ($i = 2$) steady state equivalent circuit of the WRIM is represented in figure 3.5[20]. In this figure, all rotor side parameters are referred to the stator side. The parameters R_s and R_r are stator and rotor resistances, L_{sl} and L_{rl} are the stator and rotor leakage inductances, L_{ml} is the magnetizing inductance, and ω_s and ω_r are the synchronous and rotor angular speeds. These angular speeds are related by

$$\omega_r = \omega_t \,.\, k_t \,. \tag{16}$$

where k_t accounts for the transformation inside the gear box and the number of pairs of poles such that the mechanical turbine speed is translated into the WRIM's electric rotor speed. The sequence machine slip s is the slip of the machine during normal operating condition and $s_i = s_{012}$ is given by is the slip of sequence components.

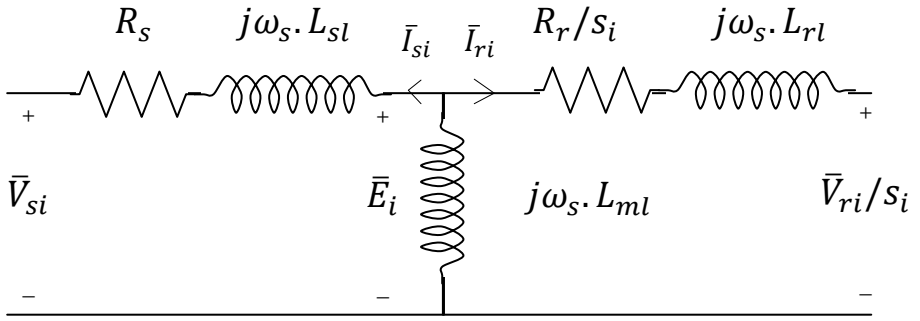

Figure 3.5: Steady state sequence equivalent circuit of WRIM.

The behavior of the positive sequence circuit is the same as for normal balanced conditions and the negative sequence circuit sets up a reverse rotating field, so that if the rotor slip is s with respect to the positive sequence field, it will be (2 – *s*) relative to the negative sequence field. Therefore, the zero-, positive- and negative-sequence slip values of the WRIM are expressed as [18]:

$$s_{012} = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 0 \\ s \\ 2-s \end{bmatrix} = \begin{bmatrix} 0 \\ (\omega_s - \omega_r)/\omega_s \\ (\omega_s + \omega_r)/\omega_s \end{bmatrix}. \tag{17}$$

The resistance and reactance of all passive components remain the same and constant in both positive and negative sequence networks. Only the behavior of the load resistance will reverse due to the effect of slip for the case of negative sequence circuit.

At the stator frequency, for each sequence network, the stator, rotor and magnetizing impedances are given by: $Z_{s012} = R_s + j\,\omega_s L_{sl}$,

$$Z_{r012} = R_r/s_{012} + j\,\omega_s L_{rl}, \text{ and}$$

$$Z_{m012} = j\,\omega_s L_{ml}, \text{ respectively.}$$

Transforming the supply three-phase voltages $\bar{V}_{sabc}$ and rotor voltage $\bar{V}_{rabc}$ to sequence components using generic equations shown in appendix 6.1 [19]:
Writing a node equation sequence induced voltages can be found as:

$$\bar{E}_{012} = \frac{\frac{\bar{V}_{s012}}{Z_{s012}} + \frac{(\bar{V}_{r012}/s_{012})}{Z_{r012}}}{\frac{1}{Z_{s012}} + \frac{1}{Z_{r012}} + \frac{1}{Z_{m012}}}. \tag{18}$$

The sequence stator and rotor currents $\bar{I}_{s012}$ and $\bar{I}_{r012}$ can be calculated as below:

$$\bar{I}_{s012} = \frac{\bar{V}_{s012} - \bar{E}_{012}}{Z_{s012}}, \tag{19}$$

$$\bar{I}_{r012} = \frac{(\frac{\bar{V}_{r012}}{s_{012}}) - \bar{E}_{012}}{Z_{s012}}, \tag{20}$$

The core losses are negligible when compared to the stator and rotor copper winding losses P_{scl012} and P_{rcl012}, which can be calculated as

$$P_{scl012} = \left|\bar{I}_{s012}\right|^2 . R_s \text{ and } P_{rcl012} = \left|\bar{I}_{r012}\right|^2 . R_r \tag{21}$$

At the stator frequency, the real and reactive power flow through the stator P_{s012} and Q_{s012} can be calculated by

$$P_{s012} = real\,\{\bar{V}_{s012}.\bar{I}^*_{s012}\}. \tag{22}$$

$$Q_{s012} = imag\,\{\bar{V}_{s012}.\bar{I}^*_{s012}\}. \tag{23}$$

At the rotor frequency, the real power flow through the rotor slip-rings to VSC is P_{r012} and can be calculated as

$$P_{r012} = real\,\{\bar{V}_{r012}.\bar{I}^*_{r012}\}. \tag{24}$$

3.1.4 Proposed Complete DFIG Model Algorithm

Sub-items (3.1.1)-(3.1.3) were used to develop the important elements of the DFIG. The complete model is given by equations:

1. Wind turbine: (3) – (8);
2. Rotor and Grid side VSCs with a DC link: (13) - (15);
3. WRIM: (14) - (24)
4. The active power balance equation that combines elements of all the three elements of the DFIG is given below [20]:

$$\sum_{i=0,1,2} P_{si} + \sum_{i=0,1,2} P_{ri} = \frac{P_m}{3} + \left(\sum_{i=0,1,2} P_{sci} + \sum_{i=0,1,2} P_{rci} \right) \quad (25)$$

where a third of the mechanical power is supplied by each phase of the symmetrical components.

5. As stated earlier, the DFIG is modeled to have power factor control mode such that the output maintains a unity power factor. At the bus where the DFIG is connected (PCC), the reactive power balance equation is therefore given by

$$Q_{tabc} + Q_{sabc} = 0\,, \quad (26)$$

Therefore, knowing $\bar{V}_{sabc}$ and ω_{wind} and solving equations (3)-(8), (13)-(15) and (14)-(24), to satisfy (25) and (26), the state of the DFIG can be determined when it operates at unity power factor. All sequence component quantities are then transformed into actual phase (abc) quantities using the generic transformations presented in Appendix 6.1.

The power generated on each phase of the DFIG can be finally computed as:

$$P_{abc} = P_{sabc} + P_{tabc}, \text{ and } Q_{abc} = Q_{sabc} + Q_{tabc}. \quad (27)$$

3.2 Model Validation

In this section, the developed model is validated by using time-domain simulation via the Matlab-Simulink package.

Data for Validation:

The Matlab-Simulink and Proposed Matlab-Programming models were compared in the same SI unit data. Per unit system data cannot provide accurate results due to numerical computational errors. Analysis with SI unit data gives more precise results comparisons of waveforms, scalar and vector quantity measurements.

The machine data set and the wind turbine data set were used for both models according to [3], and are presented in Tables 3.1 and 3.2, respectively.

Table 3.1: Machine data set for both DFIG Models

Wound Rotor Induction Machine Model			
Inputs	Stator/grid Voltage (line-line)(rms)	V_{SL-L}	2400 V
	Output of Wind Turbine mechanical torque	T_m	(Nm) from WT
Data	Nominal power	P_n	2250 HP *746 VA
	Nominal Voltage (line-line)(rms)	V_{nL-L}	2400 V
	Nominal frequency	f_n	60 Hz
	Stator resistance	R_s	0.029 Ω
	Stator inductance	L_{sl}	0.226/377 H
	Rotor resistance	R_r'	0.022 Ω
	Rotor inductance	L_{rl}	0.226/377 H
	Mutual inductance	L_{ml}	13.04/377 H
	Inertia coefficient	J	63.87 $kg.m^2$
	Friction coefficient	F	0 N.m.s
	Pole pairs	P	2

Table 3.2: Wind Turbine data set for both DFIG Models

Wind Turbine Model			
Inputs	Wind Speed	ω_{wind}	14 m/s
	Pitch angle	$\beta.$	0 degree
	Generator speed	ω_r	(rad/sec)from IG
Data	Nominal mechanical output power	P_{nom}	1.5 MW
	Base power of the electrical generator	S_b	2250HP
	Base wind speed	ω_{wbase}	12 m/s
	Maximum power at base wind speed (pu of moninal mech. power)		0.73
	Base rotational speed (pu of base generator speed)		1.2
VSC and Transformer			
Data	Total impedance	Z_g	0.345 ohm

To justify the proposed algorithm of DFIG model in steady state condition, with these same inputs data sets developed another Matlab-Simulink model and check the identity of the proposed algorithm as under:

3.2.1 Matlab-Simulink Model

The Simulink model diagram is shown in figure 3.6. The internal modeling of an induction machine is done in the *d-q* reference frame. By using this model at any given wind speed, and bus bar voltage the the power generated by the DFIG and the losses can be computed. The control models of the VSC were omitted from the Matlab-Simulink DFIG models. This decision does not affect the accuracy of the DFIG model under steady-state conditions. During the validation process, the DFIG is operated in the power factor control mode so the simulation study is intended to measure and compare voltages, currents and flow of generated active powers only.

In this time-domain simulation, the *d* and *q* components of the injected rotor voltage with respect to the stator voltage can be controlled. By changing the machine speed-torque characteristic, a wide range of variable speed operation could be achieved. Negative rotor voltage ($-V_r$) indicates an increase in the machine speed, whereas positive rotor voltage ($+V_r$) indicates a decrease in the machine speed.

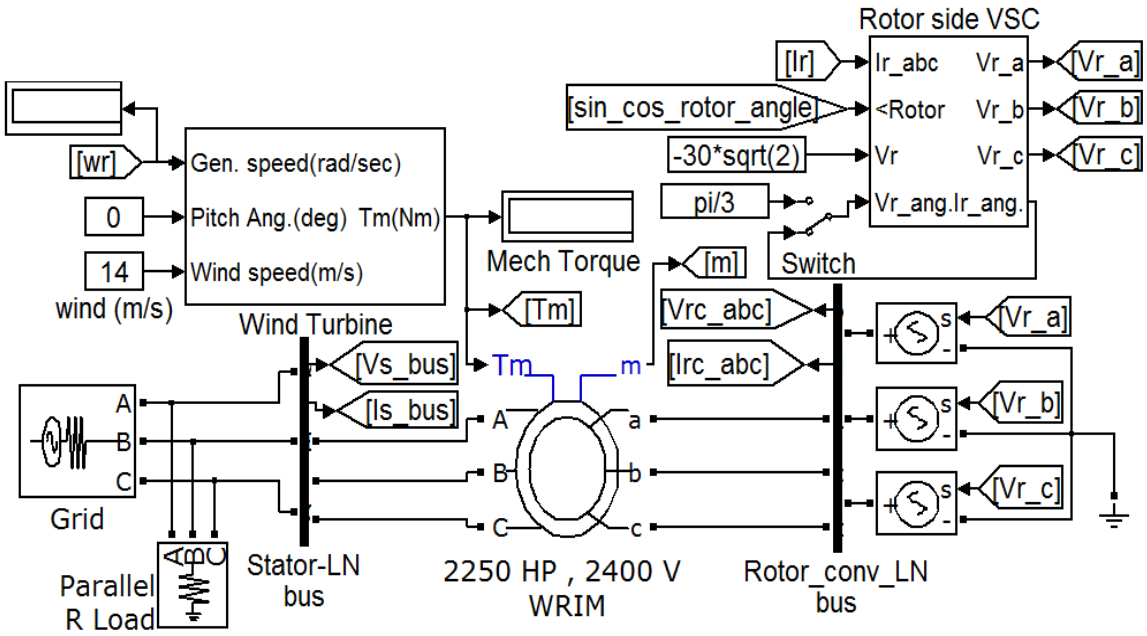

Figure 3.6: DFIG model in Matlab-Simulink

To accurately represent the DFIG, the following blocks were developed and implemented as shown in Figure 3.6:

Wind Turbine: The wind turbine model receives the wind speed, pitch angle and generator speed as inputs and in turn computes and applies mechanical power/torque to the rotor shaft of the WRIM.

Rotor side VSC: The rotor side VSC injects variable frequency AC voltage to the rotor winding to change the WRIG speed torque characteristics in order to achieve maximum power at any wind speed. Rotor voltage is produced at the rotor frequency.

WRIM: The WRIM receives the negative mechanical power/torque from the WT model, the 3-phase stator voltage from the grid at grid frequency, and the 3-phase rotor voltage from rotor side VSC at variable rotor frequency. These inputs are used to generate electric power.

Simulink Model Results:

The wind turbine model is set at wind speed 14m/s higher than the rated wind speed of 12m/s, in order to produce the rated mechanical torque of the WRIG, which is -6924 Nm.

Measurements: This block was developed in the Simulink environment to facilitate the acquisition of simulated data. The result of this block is shown in figure 3.7. The actual SI unit values of per phase voltages and currents were obtained under steady state condition.

A Phase Locked Loop (PLL) subsystem is used to measure the angle with respect to the stator reference voltage. The PLL is a feedback control system which automatically adjusts the phase of a locally generated signal to match the phase of an input signal.

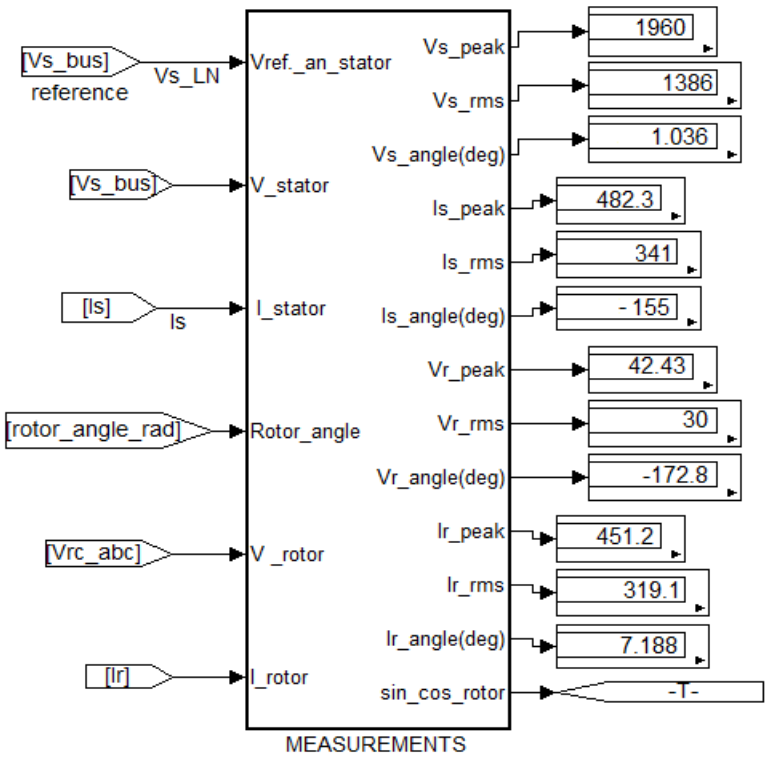

Figure 3.7: Measurements of voltage and currents in Simulink

Scopes: Figures 3.8 and 3.9 show the stator and rotor voltages and currents waveforms as seen by the scopes in the Simulink model.

Figure 3.8 presents the applied 3-phase stator/grid voltage of 1960 V(peak-peak) and the 3-phase current flow of 482 A (peak-peak) across the stator winding.

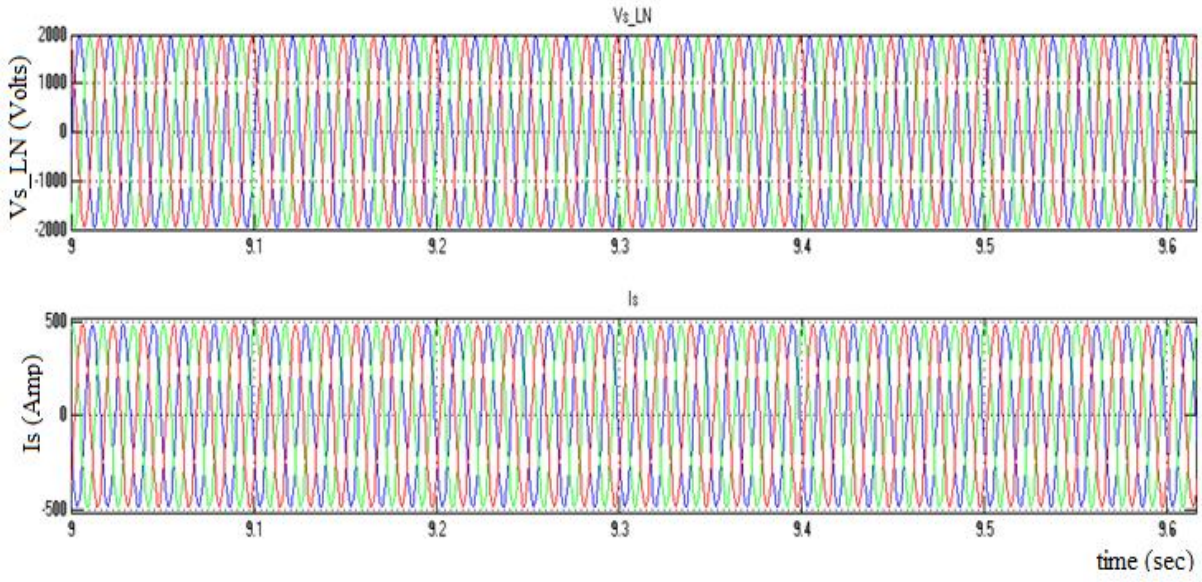

Figure 3.8: Three phase stator voltages and currents waveforms

Figure 3.9 presents the 3-phase rotor voltage of -42.43 V(peak-peak) injected by VSC and 3-phase current flow of 451 A (peak-peak) across the rotor winding.

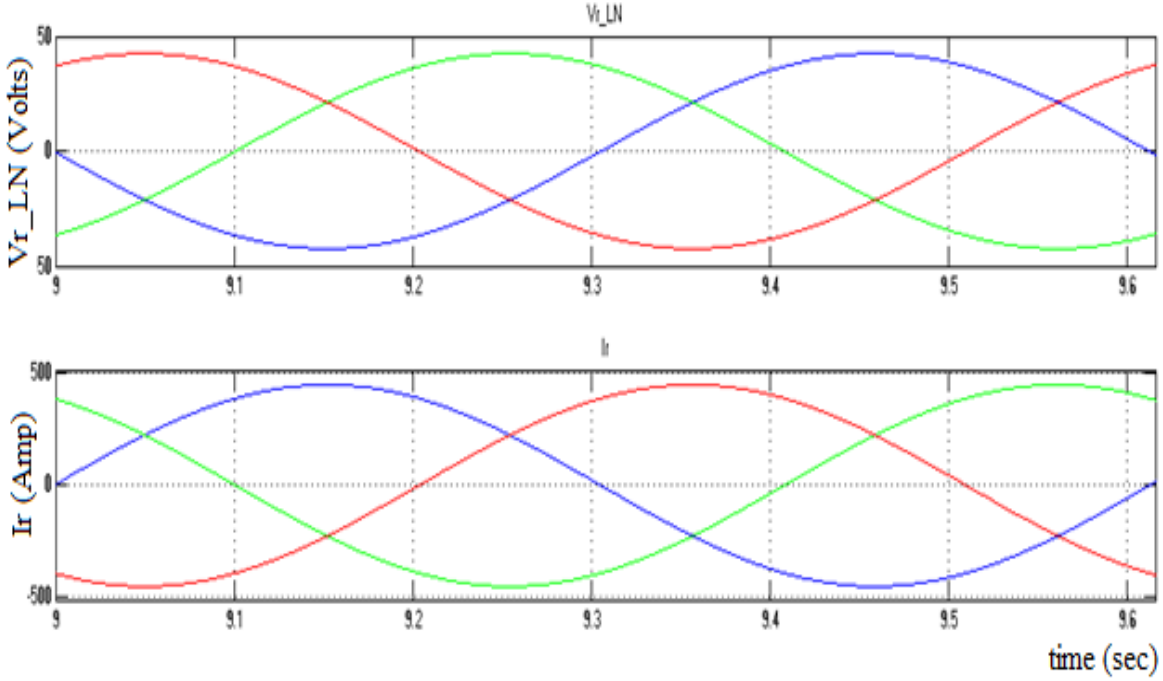

Figure 3.9: Three phase rotor voltages and currents waveforms

As indicated by the MEASUREMENTS block, the angle of rotor voltage is -172.8° and that of the rotor current is 7.188°. This means that the angle difference between the rotor voltage and current is very close to 180°, as also indicated from the above waveforms.

3.2.2 Proposed DFIG Model in Matlab-Programming code

The proposed DFIG model algorithm presented in section 3.1.4 was developed using Matlab programming code in the symmetrical components framework. With the same input data the results of the proposed algorithm are represent in Table 3.3.

Table 3.3: Results of Proposed DFIG Model Algorithm

ωwind = 14 ωr = 193.6268 β = 0	Wind Turbine Model	Tm_3ph = -6924 ; Pm/3 = -4.5161e+005
(Mag.) (Angle-Deg) **Vsabc_SI**=1.0e+003 * 1.3856 ∠ 0.0010 1.3856 ∠-0.1190 1.3856 ∠ 0.1210 **Vs012** = 1.0e+003 * 0.0000 ∠-0.1582 1.3856 ∠ 0.0010 0.0000 ∠ 0.0450 **Vr012 =** 0.0000 ∠177.1176 30.0000 ∠-172.8002 0.0000 ∠-90.0420 **E012** = 1.0e+003 * 0.0000 ∠-0.0884 1.3650 ∠ 0.0042 0.0000 ∠-0.0923 **Ir012 =** 0.0000 ∠-171.2389 322.6126 ∠ 7.4980 0.0000 ∠-175.0851 **Is012 =** 0.0000 ∠ 8.8864 344.8603 ∠-154.8614 0.0000 ∠ 5.0401	**(Mag.) (Angle-Deg)** All values are in per phase **Vrabc =** 30.0000 ∠-172.8001 30.0000 ∠ 67.1998 30.0000 ∠ -52.8002 **Vgabc** = 1.0e+003 * 1.3371 ∠ -0.0889 1.3371 ∠ -0.2089 1.3371 ∠ 0.0311 **Irabc =** 322.6126 ∠7.4980 322.6126 ∠-112.5020 322.6126 ∠127.4980 **Isabc =** 344.8602∠-154.8614 344.8603∠ 85.1386 344.8603∠ -34.8614	s = -0.0272 ; Stator power : Psa=Psb=Psc = -4.3619e+005 Rotor power : Pra=Prb=Prc = -9.6782e+003 Per phase stator copper loss Pscla=Psclb=Psclc=3.4489 e+003 * Per Phase rotor copper loss Prcla=Prclb=Prclc=2.2897 e+003 * Total per phase copper loss Pcla=Pclb=Pclc=5.7387 e+003 * Grid side VSC power Pga=Pgb=Pgc=-9.6782e+003 * DFIG generated per phase power (W) : P_a= P_b = P_c = -4.4587e+005

3.2.3 Comparison of Results for both DFIG Models

With these same data sets as inputs, both DFIG models yielded two sets of results that were virtually identical. These results are presented in Table 3.4, and validate the DFIG model proposed in this thesis.

Table 3.4: Comparison of Results for both DFIG Models

Parameter	Matbal-Simulink Model	Proposed Model*
ω_{wind} (m/s)	14	14
β(deg)	0	0
Tm (Nm)	-6924	-6924
Pm (MW)	1.35	1.35
$\bar{V}_{sa}$ (Volts)	1386 ∠1.036	1385.6 ∠1.036
$\bar{I}_{sa}$ (Amps)	341 ∠-155	344.86 ∠-154.861
$\bar{V}_{ra}$ (Volts)	30 ∠-172.8	30 ∠-172.8001
$\bar{I}_{ra}$ (Amp)	319.1 ∠7.188	322.61 ∠7.49
Ps + Pr (1-ph)	-0.4457 MW	-0.4458 MW

*Note: The proposed model was also verified through hand calculations

The encountered differences between the two models are negligible. The very small differences can be explained mainly by the assumption of losses VSCs or in a lesser extent by the difference in modeling techniques: the Simulink library uses a modeling approach based on the d-q reference frame, whereas the modeling approach used in this thesis uses the symmetrical component method.

This chapter presented the modeling and validation of the DFIG. Chapter 4 will present its integration with the ladder power flow technique and quantify the differences obtained between the proposed and the traditional DFIG models.

Chapter 4
New Power Flow approach with the proposed DFIG Model

Three Phase distribution networks are inherently unbalanced due to unbalanced loads (single, two, and three-phase), non-symmetrical conductor spacing of three phases (un-transposed), and sections with different size, lengths and combination of phases. Compared to meshed transmission network, radial distribution networks have higher level of mutual and self magnetic flux linkages among phases, which increases mutual impedances and the R/X ratios. The power flow method used in this thesis is the three-phase ladder iterative technique due to its simple and robust nature. In this technique matrices inversion is not required hence there is not any convergence issue created. It must be stated that the proposed DFIG model is equally suitable for use with Newton Raphson Technique.

For sake of operation and analysis of distribution systems, electrical power always flows from substations to the end of the feeders. However, nowadays, due to increase in size and number of WGs, the connections of WGs are expected to cause reverse power flow, which creates complicated voltage profiles in the system. As a result, due to the aforementioned limitations of other power flow methods and to this relatively new scenario, the ladder power flow technique is a more suited solution due to this sweeping-based solution algorithm, as will be explained in this chapter.

This chapter proposes a new approach of power flow that incorporates the developed active generic model of DFIG. A comparison with traditionally used fixed PQ type DFIG model is also presented.

4.1 Power Flow method description

The power flow was solved using the ladder iterative technique. The flowchart presented in figure 4.1 presents the steps of the ladder-iterative technique power flow algorithm:

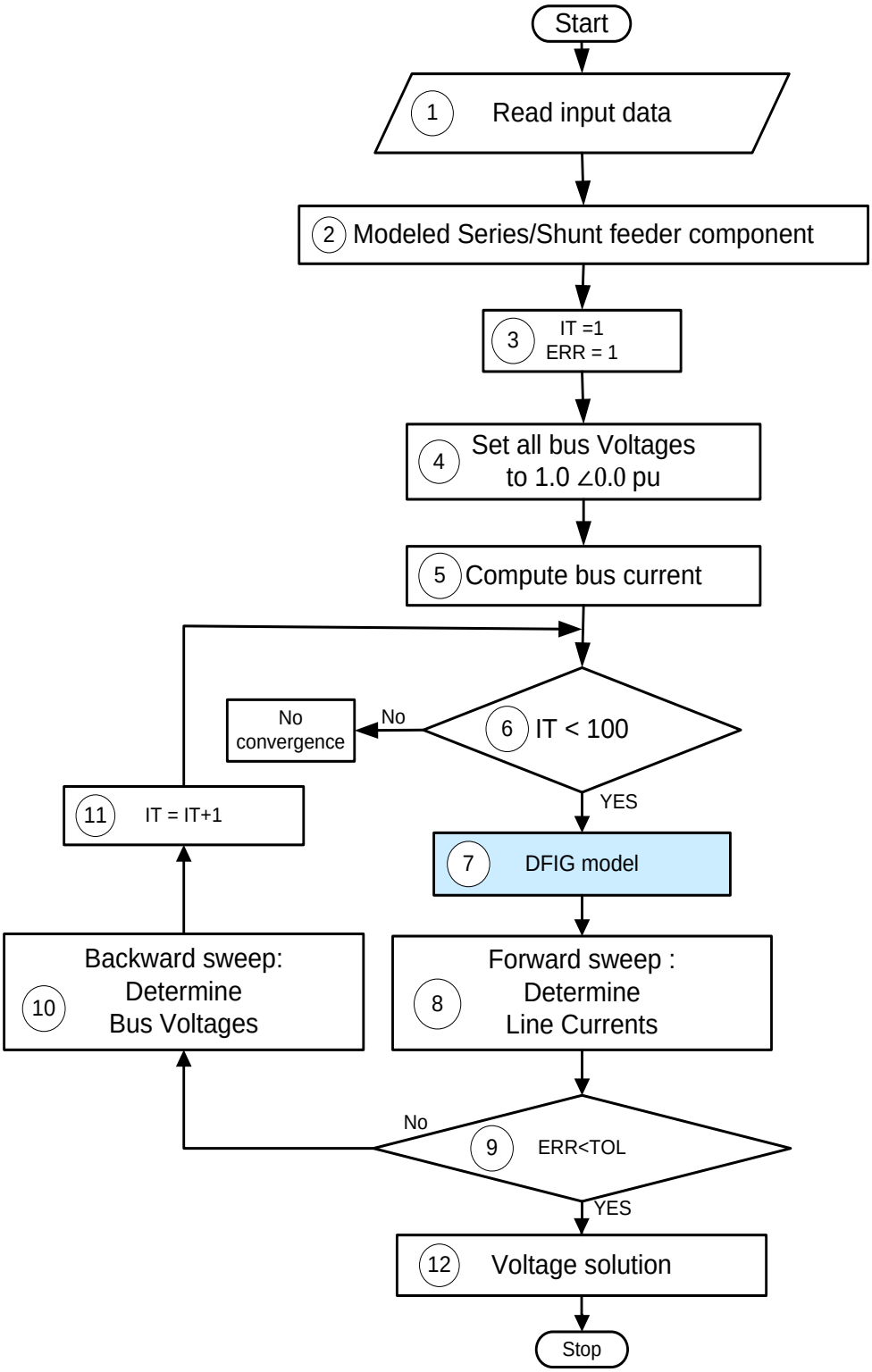

Figure 4.1: Flowchart of power flow using conventional ladder iterative technique

The steps of Ladder iterative algorithm are explained using a simple example of a five bus system with a DFIG installed in bus 5, as shown in figure 4.2:

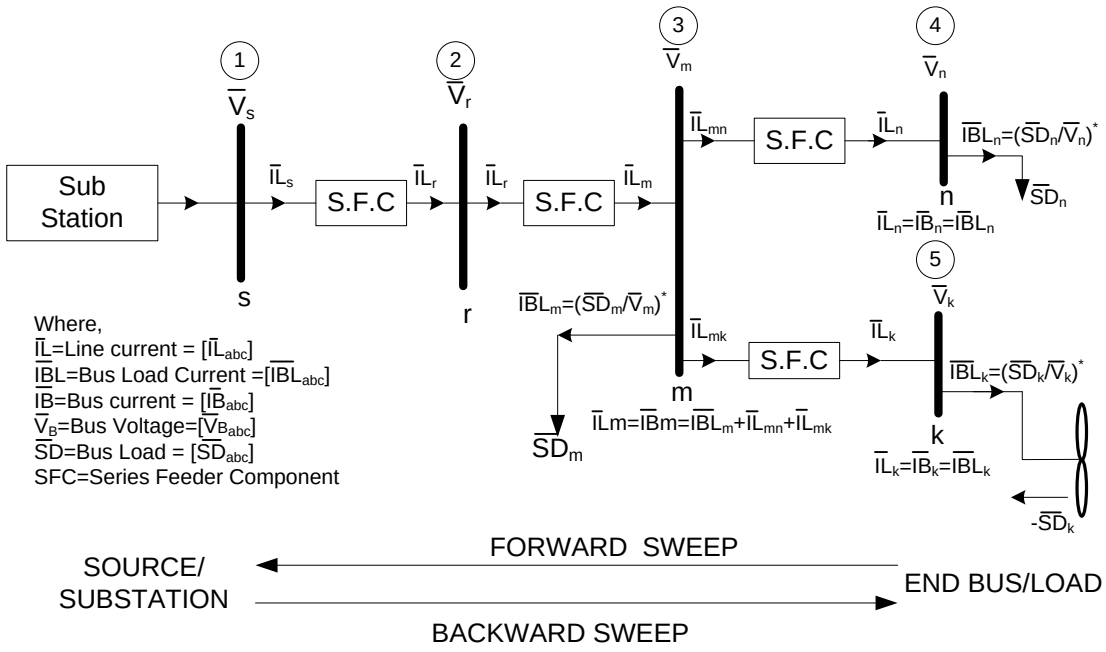

Figure 4.2: General example of feeder integrated with DFIG

Step 1: Read input data (as explained in appendix-6.2.1).

It is assumed that initially the DFIG does not generate/consume any power, i.e., at node-775, P_{abc}=Q_{abc}=0.

Step 2: The series feeder components are modeled.

In the generalized matrix form, the models of each series feeder components are developed to compute a, b, c, d, A and B parameters (to be used in step 8) as per appendix-6.2.5.

Step 3: Initialization: IT = 1 and ERR=1

Step 4: Initialization: assume all $\overline{V}_{bus} = 1\mathbf{p.u}$.

All node (bus) voltages are assumed to be the same as the source (nominal) voltages. As shown in figure 4.3, $\overline{V}_{nominal} = \overline{V}_{node-1} = \overline{V}_{source} = \overline{V}_s = 1\text{p.u.}$ and assume all other node voltages = $\overline{V}_r = \overline{V}_m = \overline{V}_n = \overline{V}_k = 1\text{p.u.}$

Step 5: Compute the end (last) nodes current of the system (Initialize forward sweep).

Compute the bus load current at all nodes (buses) as per appendix-6.2.3 by using either $\overline{IBL_n} = (\frac{\overline{SD_n}}{\overline{V_n}})^*$, for the case of delta connected constant PQ or constant current load, or $\overline{IBL_n} = (\frac{\overline{V_n}}{\overline{Z_n}})$ and $\overline{Z_n} = \frac{|\overline{V_n}|^2}{\overline{SD_n^*}}$, for the case of delta connected constant impedance type load, where n is a generic bus. Further details on the load modeling and bus load current computation are explained in appendix-6.2.3.

The same bus load current flowing from the bus will flow to all end nodes of the system, i.e., $\overline{IBL_n} = \overline{IB_n} = \overline{IL_n}$. Similarly, from DFIG connected bus-k, $\overline{IBL_k} = \overline{IB_k} = \overline{IL_k}$.

Accordingly, for the intermediate and DFIG buses a similar procedure is applied and the bus current ($\overline{\text{IB}}$) will be updated.

Step 6: While IT<100

The maximum number of iterations is specified (refer to appendix-6.2.1).

Step 7: Integrate the DFIG type WG, as per appendix 6.2.6 & Update the System Loads' P and Q.

Step 8: Forward Sweep (FS) Calculation:

The path of current determination is from the last node to the first node/source:

A- Update the bus load currents including the DFIG.
As the DFIG is integrated, the DFIG connected bus load current will be updated by a procedure similar to that of step 5.

B- Find the line current ($\overline{I_L}$) flow from the sending end node to the receiving end node (i.e., from last node to the first node), passing through the series feeder components.

In this step the developed models of each series components as per step-2 have been used to compute the sending end nodal currents and voltages.

Update the sending-end (s)/input node voltage $[\overline{V}_{abc}]_s$ and current (line) $[\overline{IL}_{abc}]_s$ from the computed a, b, c, d parameters (obtained from step 2) and receiving-end (r)/output nodal voltages (similar to step 4) and currents (obtained from step 8-A).

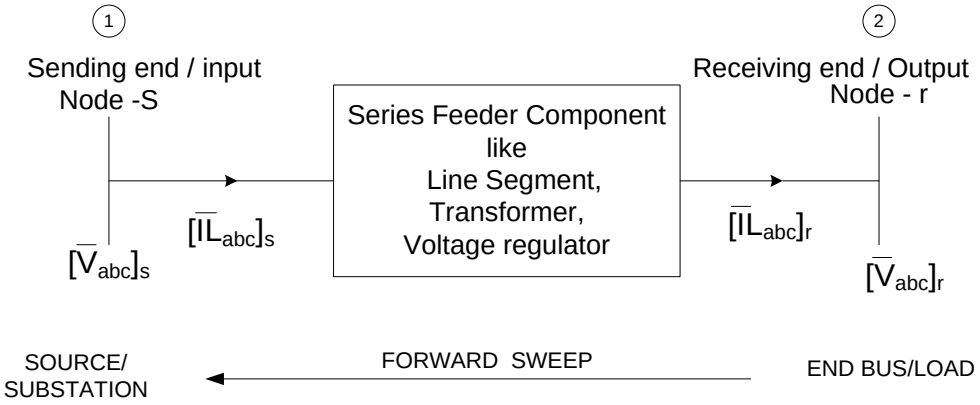

Figure 4.3: Compute voltage and current from series feeder component

Figure 4.3 presents a small common portion of figure 4.2 and its generalized equations of the sending end voltage and currents are described as

$$[\overline{V}_{abc}]_s = [a].[\overline{V}_{abc}]_r + [b].[\overline{I}_{abc}]_r, \tag{28}$$

$$[\overline{I}_{abc}]_s = [c].[\overline{V}_{abc}]_r + [d].[\overline{I}_{abc}]_r. \tag{29}$$

C- Update all bus load currents ($\overline{\text{IBL}}_{\text{node}}$) from all the updated bus nodal voltages.

Via a procedure similar to those of steps 5 and 8-A, the new updated sending end voltage (from step 8-B) and updated power after integrating the DFIG model (from step 7), the new bus load currents ($\overline{\text{IBL}}_{\text{node}}$) at all buses are updated.

For all nodes update either $\overline{IBL}_{nodes} = (\frac{\overline{SD}_{node}}{\overline{V}_{node}})^*$ in case of constant PQ or constant current load or $\overline{IBL}_{node} = (\frac{\overline{V}_{node}}{\overline{Z}_{node}})$ and $\overline{Z}_{node} = \frac{|\overline{V}_{node}|^2}{\overline{SD}^*_{node}}$ in case of constant impedance type load.

D- Update/Compute all bus currents from bus load currents by KCL

As per figure 4.2 (take as an example node-3(m)), Update/Compute the bus current ($\overline{IL}_m$=$\overline{IB}_m$) from the line currents $\overline{\text{IL}}_{\text{mn}}$ and $\overline{\text{IL}}_{\text{mk}}$ that flow through the series feeder component (as computed in step-8-B) and the bus load current ($\overline{\text{IBL}}_{\text{m}}$) (as computed as step-8-C), i.e., $\overline{IL}_m$=$\overline{IB}_m$= $\overline{IBL}_m$ + $\overline{IL}_{mn}$ + $\overline{IL}_{mk}$.

Step 9: Compute ERR and Check ERR<TOL

In step 8-B, during the calculation of all sending end bus voltages, calculate the voltage at Node-1 and compare the calculated voltage at Node-1 with the specified source/nominal voltage (1p.u.). The difference between these two voltages of the same source/substation bus (node-1) is referred to as the error value (ERR) of this algorithm.

In general, $ERR(p.u.) = \left|\overline{V}_{node-1\ calculated} - \overline{V}_{node-1\ specified\ (1pu)}\right|$ (30)

The desired tolerance value of the error (TOL - usually 0.001p.u.) is specified in the data file.

If ERR>TOL then proceed with the backward sweep-step 10; otherwise, go to step 12.

This Forward sweep (FS) and Backward sweep (BS) process is continued until the error value i.e., the difference between the computed and specified voltage at the source is smaller than the given tolerance value.

Step 10: Backward Sweep (BS) Calculation

The path of voltage determination is from the source node to the end node:

A- If the error value is not smaller than the tolerance, use the specified source voltage and the forward sweep bus current ($\overline{IB} = \overline{IL}$) (Step 8-D) flowing from Node-1 to Node-2 and compute the new voltage at Node-2.

In general, as shown in figure 4.3, the output (Node r or 2) and input (Node n or 1) voltages are given by:

$$[\overline{V}_{abc}]_r = [A].[\overline{V}_{abc}]_s - [B].[\overline{IL}_{abc}]_r, \quad (31)$$

where the parameters A and B were computed in step 2 (see appendix 6.2.5).

B- The backward sweep continues using the new updated upstream voltage and series feeder component current from the forward sweep to compute the new downstream voltage.

C- This backward sweep calculation is completed when the updated new voltages at all end nodes have been completed.

Step 11: IT = IT + 1.

This completes the first iteration and the next iteration will start at step 6.

Now the forward sweep calculation will start by using the new updated end voltages received from backward sweep.

Step 12: Solution:

Continue the forward sweep and backward sweep calculation until the calculated voltage at the source/Node-1 is smaller than the specified tolerance value, i.e., until ERR<TOL.

At this point the voltages at all nodes and current flowing in all components / segments are known.

4.2 Test system description

The unbalanced IEEE 37-bus test system [21] is used to test the proposed technique. The power flow solution with ladder iterative technique was used in this thesis. The proposed DFIG model was integrated by installing a WG into one of the buses (Bus # 775), as shown in figure 4.4.

Further changes were made into the original IEEE 37-bus system as follows:

(1) The 1.5MW rated DFIG was connected to bus 775.

(2) According to DFIG's ratings, the rating of the transformer XFM-1 was changed from 500 kVA to 1.5 MVA, and its low voltage side rating was changed from 480V to 2.4kV.

(3) To clearly see the effect of the DFIG model on the system, a 1.0 MW load is added to each of the phases of buses 730 and 731. These additional loads make voltage changes at DFIG more pronounced.

(4) The system base power was changed from 2.5 MVA to 1.5 MVA so that the machine's rated power is used as the system base.

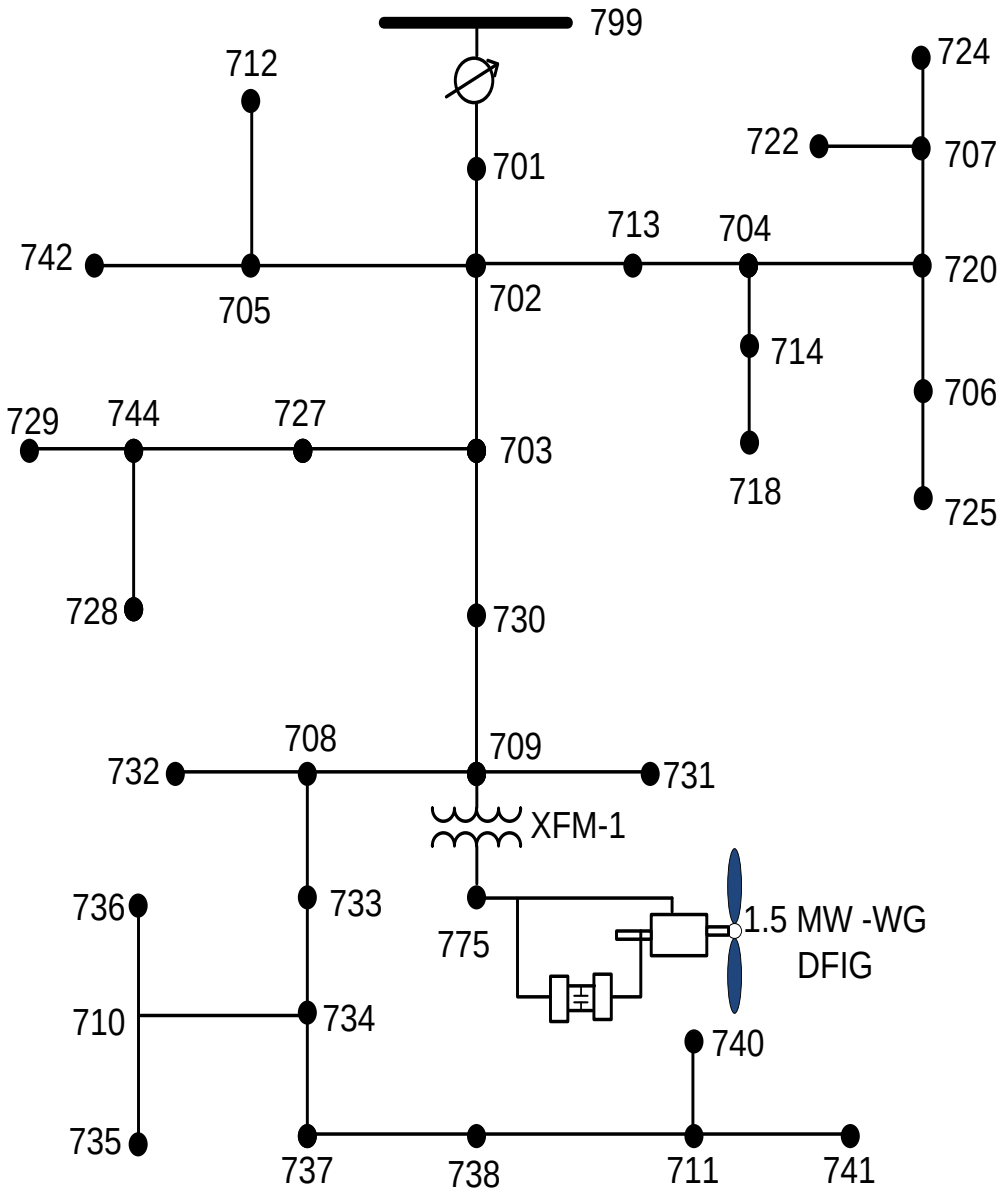

Figure 4.4: DFIG type WG connection to the IEEE 37 bus system

4.3 Integration of DFIG models in Power flow Analysis – Two Approaches

Depending on the DFIG modeling approach, the power flow analysis can be carried out in two different approaches. The first is the traditional approach, where the power-flow is solved using the DFIG modeled as a fixed *PQ* load [4]-[11].This is the traditional approach that is widely used. In the second approach, the power flow is solved using the proposed DFIG model as presented in this thesis. Both these approaches are programmed into a code and are tested.

4.3.1 Conventional Power Flow approach with traditional DFIG Model (LF-1)

In this common approach, the DFIG is modeled as a fixed *PQ* load. A diagram illustrating this approach is shown in figure 4.5.

As computed in section 3.2, at a fixed wind speed 14m/s, the grid voltages is set to 1p.u. (1386V) and the reactive power is set to zero. The DFIG generates active power equal to -0.4458 MW = -0.891 p.u. per phase.

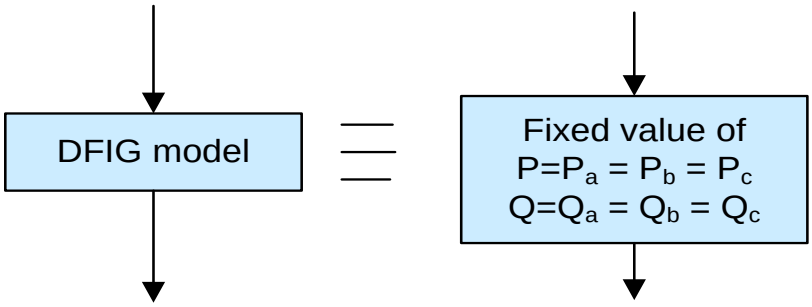

Figure 4.5: DFIG modeled as a fixed *PQ* load

Due to the unbalanced nature of the distribution system, in this type of power-flow method, the power produced by the DFIG is assumed constant on each of the three phases, even though the voltage profile on each phase became more unbalanced in each iteration.

The results of this power flow (LF-1) solution are illustrated in tables 4.1-4.2 and figures: 4.7-4.9.

4.3.2 New Power Flow approach with the proposed active DFIG Model (LF-2)

In this proposed approach, the power flow is solved using the proposed DFIG model derived in Chapter 3.1. The diagram of the proposed algorithm (3.1.4) for the DFIG model is shown in figure 4.6. The DFIG model is solved using a standard non linear equation solver in each iteration of the power flow solution.

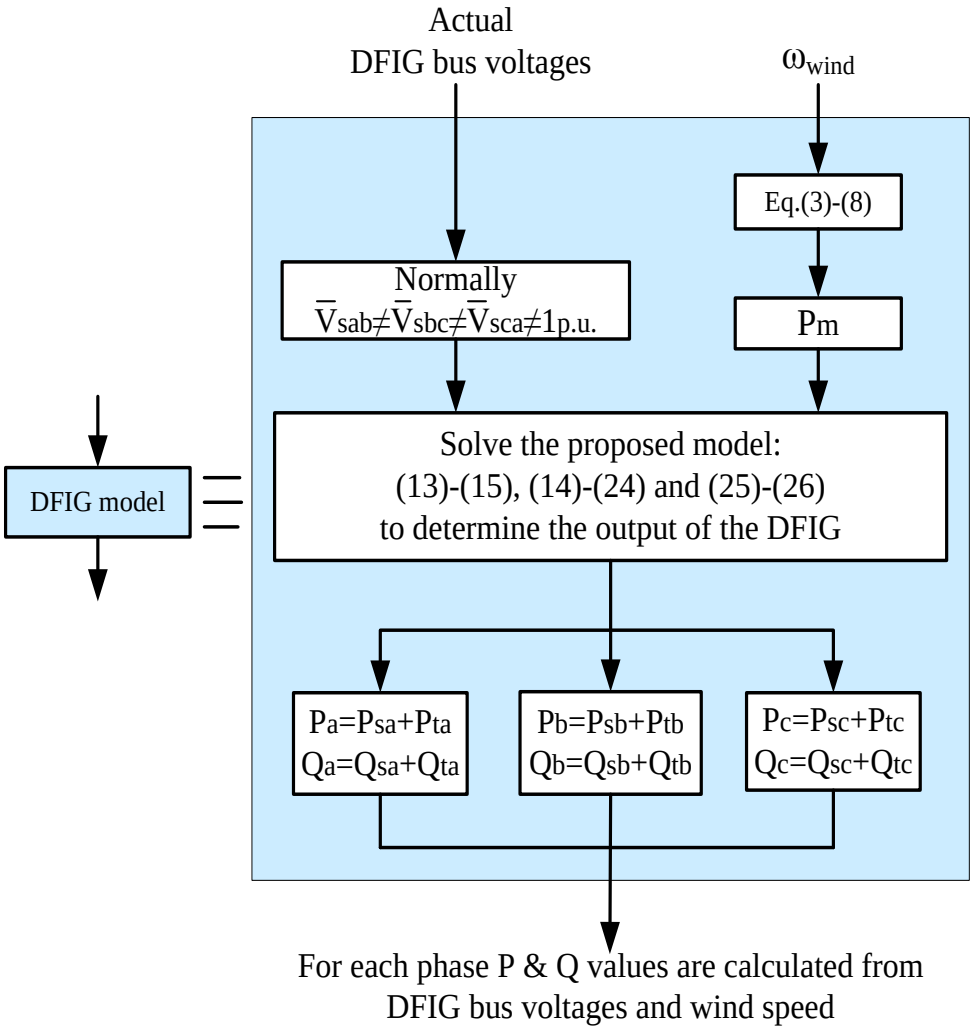

Figure 4.6: Proposed active DFIG model

In the first iteration of the ladder solution, all bus voltages are assumed to be 1 p.u. Therefore, the initial value of the DFIG stator voltage is also 1 p.u., which results in a computed injected active power P_{abc} = -0.4458 MW = -0.891 p.u. and in a controlled reactive power Q_{abc} = 0 p.u. on each phase. The resulting P_{abc} and Q_{abc} act as a negative load bus model for the next power flow iteration.

In the following iterations, due to the presence of system unbalance, the DFIG stator voltage becomes unbalanced and the DFIG model yields unbalanced P_{abc} and Q_{abc} values. These P_{abc} and Q_{abc} act as a variable unbalanced *PQ* load bus. The same process will repeat until convergence.

In each iteration, the DFIG model gives the actual value of currents, voltages, powers and losses on each phase.

The results of this proposed method of power flow (LF-2) solutions are illustrated in tables 4.1-4.2 and figures: 4.7-4.9.

4.3.3 Results and Comparisons of both Power flow Approaches

The IEEE 37-bus unbalanced test system that was integrated with the DFIG modeled in section 3.1 was used in the power flow studies. In this section the voltage solutions of both power flow approaches are presented in Table 4.1.

Figures 4.7-4.9 further quantifies the difference between the results from the two power flow models.

Table 4.1: Results from both power flow methods for the IEEE-37 bus test system

Bus	Phase-a-b Vab (pu)		Phase-b-c Vbc (pu)		Phase-c-a Vca (pu)	
Name	LF-1	LF-2	LF-1	LF-2	LF-1	LF-2
799	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
RG7	1.0437	1.0437	1.0250	1.0250	1.0345	1.0345
701	1.0145	1.0141	0.9970	0.9971	0.9978	0.9981
702	0.9945	0.9937	0.9795	0.9797	0.9750	0.9757
703	0.9698	0.9684	0.9595	0.9599	0.9487	0.9498
727	0.9686	0.9673	0.9589	0.9592	0.9477	0.9488
744	0.9679	0.9666	0.9585	0.9588	0.9473	0.9484
728	0.9675	0.9662	0.9581	0.9584	0.9469	0.9480
729	0.9675	0.9662	0.9584	0.9588	0.9472	0.9483
730	0.9446	0.9427	0.9383	0.9390	0.9240	0.9254
709	0.9403	0.9381	0.9349	0.9356	0.9199	0.9214
708	0.9378	0.9356	0.9338	0.9346	0.9175	0.9191
732	0.9377	0.9355	0.9337	0.9345	0.9171	0.9186
733	0.9337	0.9315	0.9320	0.9328	0.9137	0.9152
734	0.9302	0.9280	0.9305	0.9313	0.9103	0.9118
710	0.9297	0.9275	0.9295	0.9303	0.9087	0.9102
735	0.9296	0.9274	0.9293	0.9301	0.9081	0.9096
736	0.9292	0.9270	0.9280	0.9288	0.9084	0.9099
737	0.9268	0.9246	0.9295	0.9303	0.9081	0.9096
738	0.9256	0.9234	0.9291	0.9299	0.9069	0.9085
711	0.9253	0.9231	0.9289	0.9297	0.9059	0.9075
740	0.9252	0.9230	0.9287	0.9295	0.9054	0.9069
741	0.9252	0.9230	0.9288	0.9296	0.9056	0.9072
731	0.9284	0.9262	0.9235	0.9242	0.9084	0.9100
XF7	0.9403	0.9381	0.9349	0.9356	0.9199	0.9214
DFIG	0.9377	0.9354	0.9323	0.9333	0.9172	0.9193
705	0.9939	0.9931	0.9781	0.9784	0.9737	0.9744
712	0.9938	0.9930	0.9780	0.9782	0.9730	0.9737
742	0.9936	0.9928	0.9773	0.9775	0.9735	0.9742
713	0.9932	0.9924	0.9776	0.9778	0.9732	0.9738
704	0.9915	0.9907	0.9750	0.9752	0.9713	0.9720
714	0.9912	0.9904	0.9749	0.9751	0.9712	0.9719
718	0.9899	0.9891	0.9747	0.9749	0.9708	0.9715
720	0.9902	0.9894	0.9716	0.9719	0.9689	0.9696
706	0.9901	0.9893	0.9712	0.9714	0.9688	0.9694
725	0.9900	0.9892	0.9708	0.9711	0.9687	0.9694
707	0.9885	0.9876	0.9665	0.9667	0.9674	0.9680
722	0.9883	0.9875	0.9659	0.9662	0.9672	0.9679
724	0.9882	0.9873	0.9656	0.9658	0.9672	0.9679

LF-1: DFIG modeled as a Fixed PQ load LF-2: Proposed DFIG model

Figure 4.7 shows that the difference in the line to line voltage (ab) solutions for both power flow methods is about 0.0023 p.u. at the DFIG and nearby buses. Similar results can be seen for the other line to line voltages BC and CA as shown in figures 4.8 and 4.9.

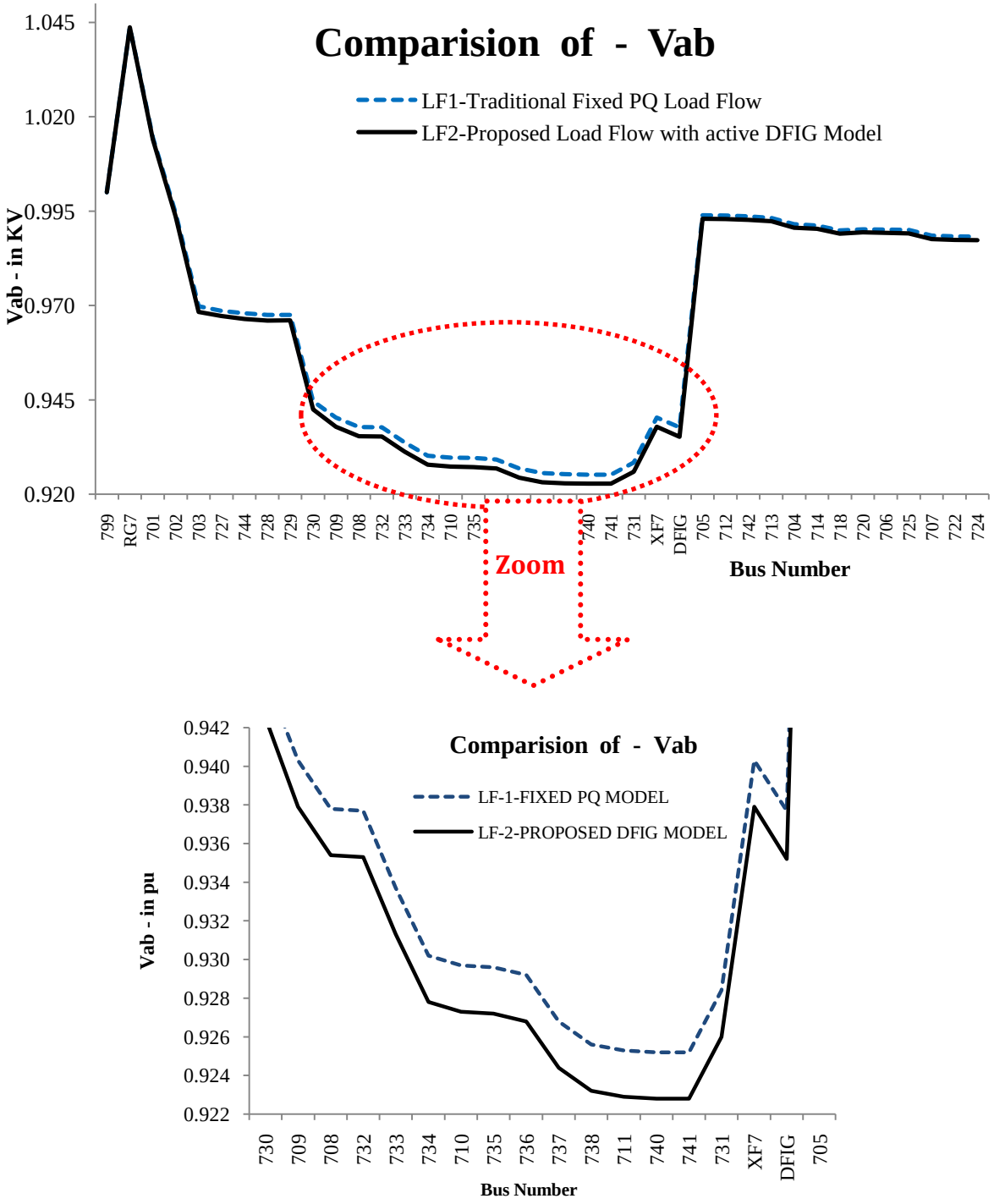

Figure 4.7: Comparision of phase a-b line to line voltage for both power-flow methods

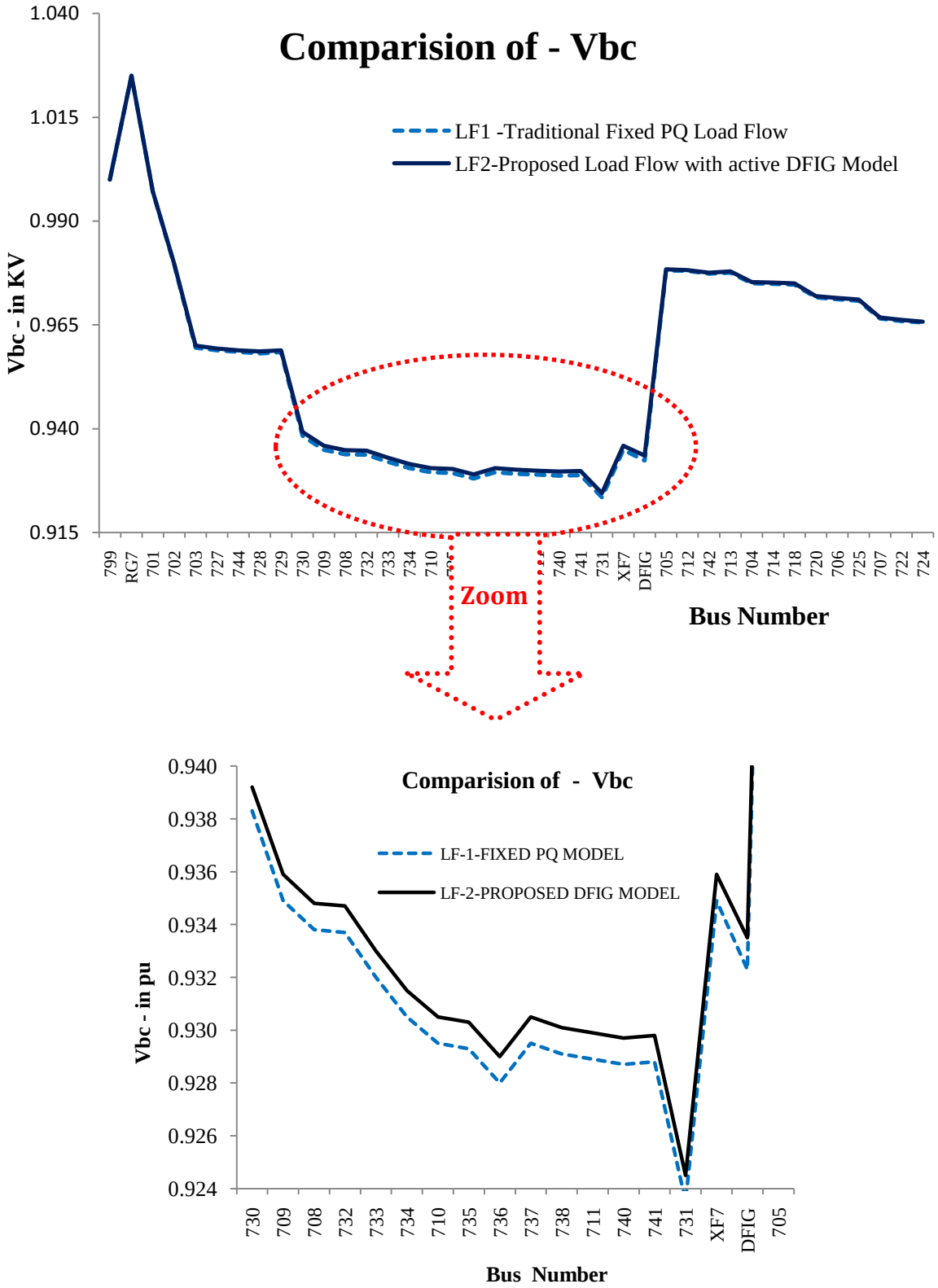

Figure 4.8: Comparision of phase b-c line to line voltage for both power-flow methods

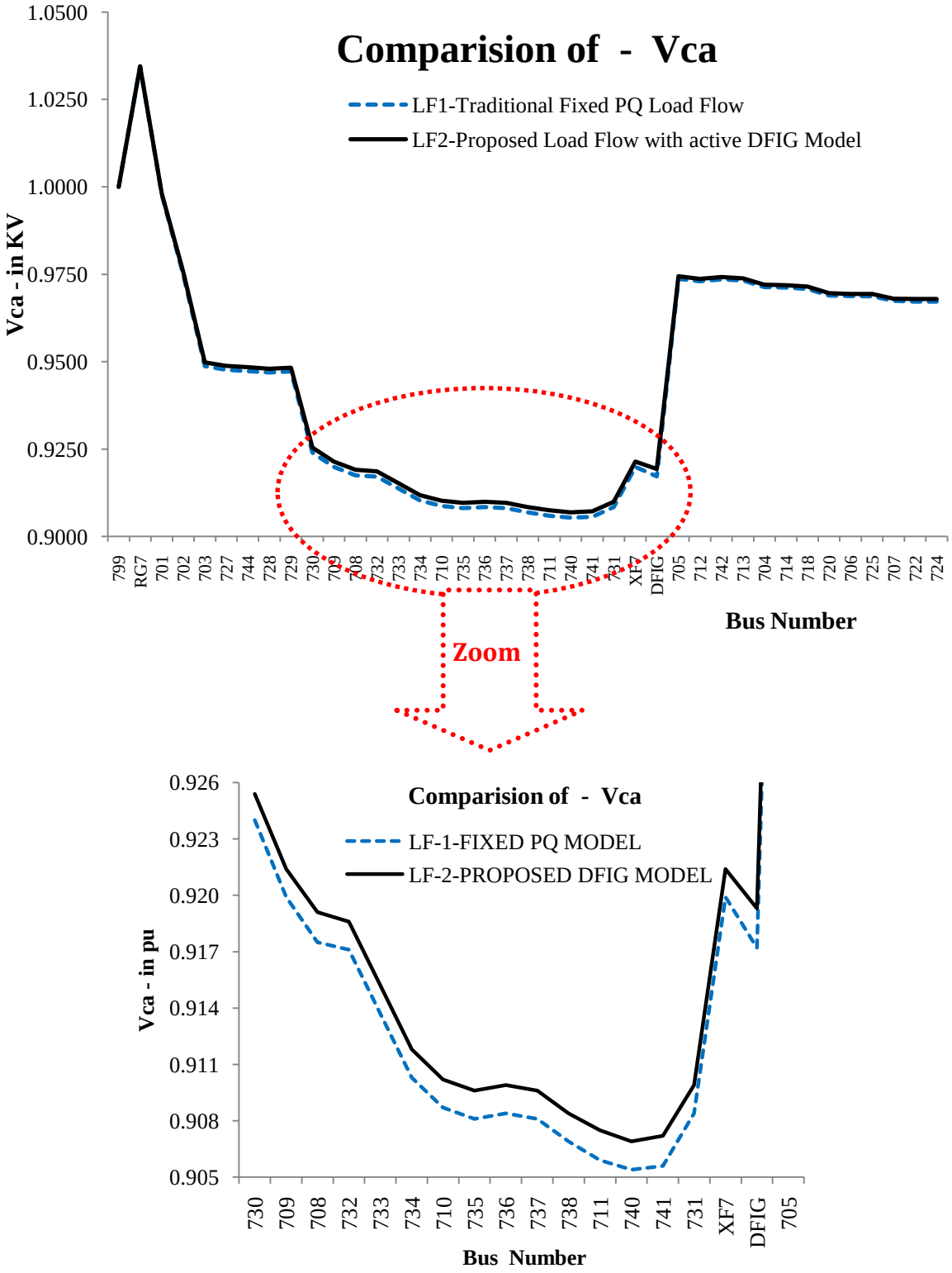

Figure 4.9: Comparision of phase c-a line to line voltage for both power-flow methods

This comparison between the voltage solutions obtained by the two types of power flow methods (LF1 and LF-2) highlights the significant impact of proposed active DFIG model on each phase of the system. Significant differences in voltage, power and losses on the DFIG bus for both power flow methods can also be noticed and are summarized in Table 4.2.

Table 4.2: Comparison of Voltages, Powers and Losses in DFIG in both LF-methods

	L-L bus vol.(pu)		DFIG bus power(MW)		DFIG's loss(W)	
	LF-1 Fixed PQ	LF-2 Proposed	LF-1 Fixed PQ	LF-2 Proposed	LF-1 Fixed PQ	LF-2 Proposed
Phase-a	0.9377	0.9354	-0.4458	-0.4082	5738	6116
Phase-b	0.9323	0.9333	-0.4458	-0.4520	5738	7973
Phase-c	0.9172	0.9193	-0.4458	-0.4727	5738	7829

The results presented in this chapter reveal that the detailed modeling of the DFIG can overcome the inaccuracies of the traditional model used by many utilities to solve the power flow. In addition, the proposed modeling is practical and is ideal for applications in unbalanced transmission systems and distribution systems.

4.3.4 Validation of Proposed New Approach of Power Flow

In order to further verify the correctness of the power flow results obtained using the proposed DFIG model, the following procedure was adopted.

1. The power output of the DFIG, as determined using the proposed model reported in Table 4.2, was taken as fixed PQ load (P_a = -0.4082MW, P_b = -0.4520 MW and P_c = -0.4727 MW) and
   The voltage solution was determined using ladder iterative technique (Figure 4.1). This voltage solution was identical to the results obtained using the proposed model (V_{sab} = 0.935 p.u., V_{sbc} = 0.933 p.u., V_{sca} = 0.9193 p.u.).
2. Using this DFIG model on MATLAB-Simulink, bus voltages were taken as (V_{sab} = 0.935 p.u., V_{sbc} = 0.933 p.u., V_{sca} = 0.9193 p.u.) with a fixed wind speed 14 m/s.

The same power output was obtained: (P_a = -0.4082MW, P_b = -0.4520 MW and P_c = -0.4727 MW).

It confirmed the proposed DFIG model and power flow method LF-2.

Chapter 5
Conclusions and Suggestions for Future Research

5.1 Conclusions

The DFIG (Doubly-Fed Induction Generator) is popularly used in wind generators. Traditionally, a DFIG is modeled as a fixed negative PQ load in power flow studies. This fixed PQ model of a DFIG leads to inaccurate voltage solutions in power flow studies. With the widespread use of DFIG based wind generators in distribution systems, their accurate modeling is imperative, and is the focus of this thesis. The main contributions of this research work can be summarized as follows:

1. It develops and reports an accurate three-phase model, using a set of nonlinear equations. The proposed model accounts for a wind turbine, two VSCs connected back-to-back through a DC link and a WRIM (Wound Rotor Induction Machine).
2. The proposed model validated by comparing its results with those obtained from MATLAB-Simulink and hand calculations.
3. The proposed model can be easily integrated into power flow algorithms such ladder iterative technique and the revised power flow algorithm is presented.
4. Power Flow analysis results of an unbalanced three-phase IEEE 37-bus test system are reported. Results obtained using the fixed PQ model and the proposed model are reported and compared. The proposed model is shown to be accurate.

The proposed model creates an accurate three-phase representation of a DFIG and is suitable for power flow studies. It is amenable for unbalanced conditions. It is suitable for the ladder iterative technique (as shown in this thesis) and equally suitable for Newton-Raphson Technique as well.

5.2 Suggestions for Future Research

As any research work, this thesis can be improved in several ways. In this section, four suggestions for future research development are provided.

- During this research work, the VSC and dc-link losses are considered to be zero. One way to circumvent this limitation is to include the full losses model for these components in the system. In addition, this thesis also neglected the VSC losses associated with the PWM scheme, which were addressed in [17]. These losses result

from including complex models that include, but are not limited to, equivalent impedances. Therefore, it is possible to further expand the model proposed in this thesis by incorporating these detailed losses to get an even more accurate picture of the DFIG.

- The stator of the DFIG is directly connected to the grid and the rotor is connected to the grid through a back-to-back VSC. Hence, the normal frequency deviation could affect the DFIG performance and output power. As a future research development, it could be proposed to analyze in more detail this effect in the power flow approach.
- A more detailed power flow could also be developed so that other types of machines can be integrated as well. For example, DFIGs can be connected to a bus and an inverter-based WG can be connected to another. This improved power flow technique could be designed in a way that computationally efficient calculations are performed to provide fast system solutions.
- In reality, the wind speed is not constant over the turbines of all WGs in a wind farm. This can be explained by generators installed in a queue being hit by the wind. In this scenario, the wind that reaches the second generator will have a slower speed than that of the first generator. As the wind passes through several generators, the wind speed will change. This effect was not taken into account in this thesis, but could be developed in future research using a probabilistic approach.

Appendix

6.1 General equations used in WRIM model and DFIG algorithm

The following generic equations were used for the symmetrical components transformations:

- the actual three-phase line to line voltages to sequence components

$$[\bar{V}_{LL012}] = [A]^{-1}.[\bar{V}_{LLabc}], \quad (32)$$

- the sequence components line to line voltages to the sequence components line to neutral voltages

$$[\bar{V}_{LN012}] = [T].[\bar{V}_{LL012}], \quad (33)$$

- the sequence components line to neutral voltages to the sequence components line to line voltages

$$[\bar{V}_{LNabc}] = [A].[\bar{V}_{LN012}], \quad (34)$$

where $a = 1\angle 120^\circ$, $t = \frac{1}{\sqrt{3}}\angle -30^\circ$, $t^* = \frac{1}{\sqrt{3}}\angle 30^\circ$,

$$[A]^{-1} = \frac{1}{3}\begin{bmatrix}1 & 1 & 1\\ 1 & a & a^2\\ 1 & a^2 & a\end{bmatrix}, [A] = \begin{bmatrix}1 & 1 & 1\\ 1 & a^2 & a\\ 1 & a & a^2\end{bmatrix} \text{and } [T] = \begin{bmatrix}1 & 0 & 0\\ 0 & t^* & 0\\ 0 & 0 & t\end{bmatrix}.$$

Similar equations can be used for the currents.

6.2 Equations and Matlab program codes of Power-Flow method

6.2.1. Input Data file for Matlab Power flow program.

```
4.8-kvolts-radial-37-node-system
2001
PEAK_LOAD
001
38      38      0001    1       4.8       4   26    1    35    0    0
0    1
1.0     10e-8   1.5     1.05    0.95    100
MAIN

no  bn  bnam    p1-pd   p2-qd   p2-pd   p2-qd   p3-pd   p3-qd  type
1   1   B799    0.000   0.000   0.000   0.000   0.000   0.000   001

no  bn  bnam    p1-pd   p1-qd   p2-pd   p2-qd   p3-pd   p3-qd  type
1   3   B701    0.140   0.070   0.140   0.070   0.350   0.175   001
2   14  B712    0.000   0.000   0.000   0.000   0.085   0.040   001
3   15  B713    0.000   0.000   0.000   0.000   0.085   0.040   001
4   16  B714    0.017   0.008   0.021   0.010   0.000   0.000   002
5   17  B718    0.085   0.040   0.000   0.000   0.000   0.000   003
6   18  B720    0.000   0.000   0.000   0.000   0.085   0.040   001
7   19  B722    0.000   0.000   0.140   0.070   0.021   0.010   002
8   20  B724    0.000   0.000   0.042   0.021   0.000   0.000   003
9   21  B725    0.000   0.000   0.042   0.021   0.000   0.000   001
10  22  B727    0.000   0.000   0.000   0.000   0.042   0.021   001
11  23  B728    0.042   0.021   0.042   0.021   0.042   0.021   001
12  24  B729    0.042   0.021   0.000   0.000   0.000   0.000   002
13  25  B730    1.000   0.000   1.000   0.000   1.085   0.040   003
14  26  B731    1.000   0.000   1.085   0.040   1.000   0.000   003
15  27  B732    0.000   0.000   0.000   0.000   0.042   0.021   001
16  28  B733    0.085   0.040   0.000   0.000   0.000   0.000   002
17  29  B734    0.000   0.000   0.000   0.000   0.042   0.021   001
18  30  B735    0.000   0.000   0.000   0.000   0.085   0.040   001
19  31  B736    0.000   0.000   0.042   0.021   0.000   0.000   003
20  32  B737    0.140   0.070   0.000   0.000   0.000   0.000   002
21  33  B738    0.126   0.062   0.000   0.000   0.000   0.000   001
22  34  B740    0.000   0.000   0.000   0.000   0.085   0.040   001
23  35  B741    0.000   0.000   0.000   0.000   0.042   0.021   002
24  36  B742    0.008   0.004   0.085   0.040   0.000   0.000   003
25  37  B744    0.042   0.021   0.000   0.000   0.000   0.000   001
26  38  B775    0       0       0       0       0       0       011

con dis diam nn c-dia  con-GMR c-res s-diam str-GMR s-resis
721 0.5 1.98 20 1.150  0.03680 0.105 0.1019 0.00330 05.9026
722 0.5 1.56 16 0.813  0.02600 0.206 0.0808 0.00262 09.3747
723 0.5 1.10 07 0.414  0.01250 0.769 0.0641 0.00208 14.8722
724 0.5 0.98 06 0.292  0.00883 1.541 0.0641 0.00208 14.8722

no  fb  tb  tpa         tpb             tpc    con
1   1   2   6.699843    3.902439024     0      303

no  fb  tb  vllh    vlll    r           x       rat    con
2   11  38  4.8     2.4    0.0009    0.0181    1.5     404

no  fb  tb  lengt zer   pos   con
3   2   3   1.850 0.0   0.0   721
4   3   4   0.960 0.0   0.0   722
5   4   5   1.320 0.0   0.0   722
6   5   25  0.600 0.0   0.0   723
```

7	25	11	0.200	0.0	0.0	723
8	11	10	0.320	0.0	0.0	723
9	10	28	0.320	0.0	0.0	724
10	28	29	0.560	0.0	0.0	723
11	29	32	0.640	0.0	0.0	723
12	32	33	0.400	0.0	0.0	723
13	33	13	0.400	0.0	0.0	723
14	13	35	0.400	0.0	0.0	723
15	4	7	0.400	0.0	0.0	724
16	7	36	0.320	0.0	0.0	724
17	4	15	0.360	0.0	0.0	723
18	15	6	0.520	0.0	0.0	723
19	6	18	0.800	0.0	0.0	723
20	18	8	0.600	0.0	0.0	723
21	8	21	0.280	0.0	0.0	724
22	5	22	0.240	0.0	0.0	724
23	22	37	0.280	0.0	0.0	723
24	37	24	0.280	0.0	0.0	724
25	6	16	0.080	0.0	0.0	724
26	16	17	0.520	0.0	0.0	724
27	18	9	0.920	0.0	0.0	724
28	9	20	0.760	0.0	0.0	724
29	9	19	0.120	0.0	0.0	724
30	7	14	0.240	0.0	0.0	724
31	10	27	0.320	0.0	0.0	724
32	11	26	0.600	0.0	0.0	723
33	29	12	0.520	0.0	0.0	724
34	12	30	0.200	0.0	0.0	724
35	12	31	1.280	0.0	0.0	724
36	13	34	0.200	0.0	0.0	724
37	37	23	0.200	0.0	0.0	724

6.2.2. Main Program file reads the data file and all other function files.

```
% Main File reads all data and all function files
function DATA_VLN_DFIG

global Z721 Z722 Z723 Z724 Y721 Y722 Y723 Y724
global LENGTH ZZERO ZPOSITIVE CONFIG
global NB NBB NS NG NLB NTR NTRL NT NSHC NSVS NSHR NSH NREG
global VSLACK TOLER PBASE VLMAX VLMIN ITMAX
global BIND BSN BNAM PD1 QD1 PD2 QD2 PD3 QD3 LTP PDD1 QDD1 PDD2 QDD2 PDD3
QDD3 LTPD
global TRNO FB TB  VLLH VLLL R X PRAT  PD1G QD1G PD2G QD2G PD3G QD3G W SB
SD SG W SB SD SG err_SE  SE err_E E
global TPA TPB TPC VLN
global VBASE
global ifp                          % input file pointers

temp1 = input('Input  File Name:' , 's' );
ifp = fopen(temp1,'r');

temp = fscanf(ifp,'%s',[1]);
temp = fscanf(ifp,'%s',[1]);
temp = fscanf(ifp,'%s',[1]);

temp = fscanf(ifp,'%s',[1]);       % 001
ttt  = fscanf(ifp,'%d %d %d %d %f %d %d %d %d %d %d %d %d',[1,13]);    %
read - following data
        NB    = ttt(1);       % NB   = Number of Buses ;
        NBB   = ttt(2);
        NS    = ttt(3);       % NS   = Slack Bus Number
        NG    = ttt(4);       % NG   = Number of Generators ;
        VBASE = ttt(5);       % VBASE= Base Voltage in KV ;
        NCON = ttt(6);        % NCON = No. of Lines Configuration
        NLB   = ttt(7);       % NLB  = No. of Load Buses ;
        NTR   = ttt(8);       % NTR  = No. of Transformers ;
        NTRL = ttt(9);        % NTRL = No. of Transmission Lines;
        NSHC = ttt(10);       % NSHC = No. of shunt capacitors ;
        NSVS = ttt(11);       % NSVS = No. of switchable capacitors;
        NSHR = ttt(12);       % NSHR = No. of Shunt reactors ;
        NREG = ttt(13);       % NREG = No. of Regulators ;
%% Add components/lines together :
NT  = NTR + NTRL+ NREG;       % NT  = No. of total lines
NSH = NSHC + NSVS + NSHR;     % NSH = No. of shunts.
%%
ttt    = fscanf(ifp,'%f %f %f %f %f %d',[1,6]);                    %
read - following data
        VSLACK = ttt(1);      % VSLACK = Slack Bus Voltage ;
        TOLER  = ttt(2);      % TOLER  = Tolerance ;
        PBASE  = ttt(3);      % PBASE  = Base MVA ;
        VLMAX  = ttt(4);      % VLMAX  = Maximum Load Bus Voltage ;
        VLMIN  = ttt(5);      % VLMIN  = Minimum Load Bus Voltage ;
        ITMAX  = ttt(6);      % ITMAX  = Maximum Number of Iterations ;
%%
ttt    = fscanf(ifp,'%s',[1]);        % read - "MAIN" - region
%% Initialise matrix/columns for data
BSN   = zeros(NG+NLB,1);      % BSN = Bus Sr. Number
BNAM  = cell (NG+NLB);        % BNAM = Bus Name

PDD1  = zeros(NG+NLB,1);      QDD1    = zeros(NG+NLB,1);
PDD2  = zeros(NG+NLB,1);      QDD2    = zeros(NG+NLB,1);
```

```
PDD3  = zeros(NG+NLB,1);     QDD3     = zeros(NG+NLB,1);
LTPD  = zeros(NG+NLB,1);

% Line Segment data
PD1   = zeros(NB,1);         QD1      = zeros(NB,1);
PD2   = zeros(NB,1);         QD2      = zeros(NB,1);
PD3   = zeros(NB,1);         QD3      = zeros(NB,1);
LTP   = zeros(NB,1);
% BSNMAX=0;
ttt       = fscanf(ifp,'%s %s %s %s %s %s %s %s %s %s',[1,10]);
for k = 1:NG

    %ref.sr.no and bus Nos. - GSP/G-bus No.
    ttt       = fscanf(ifp,'%d %d',[1,2]);   % decimal - nos.
              % ref. no. = ttt(1) - so not req.
                BSN(k)    = ttt(2);

    %BusName - GSP / generator -bus name
    ttt       = fscanf(ifp,'%s',[1]);
                BNAM(k)= cellstr(ttt);

    %Loads on each phases of GSP - bus
    ttt       = fscanf(ifp,'%f %f %f %f %f %f %d',[1,7]);
    PDD1(k)     = ttt(1);    QDD1(k)     = ttt(2);
    PDD2(k)     = ttt(3);    QDD2(k)     = ttt(4);
    PDD3(k)     = ttt(5);    QDD3(k)     = ttt(6);

    LTPD(k)     = ttt(7);

    PDD1(k) = PDD1(k)/(PBASE/3);          QDD1(k) = QDD1(k)/(PBASE/3);
    PDD2(k) = PDD2(k)/(PBASE/3);          QDD2(k) = QDD2(k)/(PBASE/3);
    PDD3(k) = PDD3(k)/(PBASE/3);          QDD3(k) = QDD3(k)/(PBASE/3);
end
%% Reading LOAD BUS data (SPOT LOADS)
    ttt       = fscanf(ifp,'%s %s %s %s %s %s %s %s %s %s',[1,10]);
for k = NG+1:NG+NLB
    % ALL SAME AS ABOVE GSP-
    ttt       = fscanf(ifp,'%d %d',[1,2]);
                BSN(k)    = ttt(2);

    ttt       = fscanf(ifp,'%s',[1]);
                BNAM(k)= cellstr(ttt);

    ttt         = fscanf(ifp,'%f %f %f %f %f %f %d',[1,7]);
    PDD1(k)     = ttt(1);                 QDD1(k)     = ttt(2);
    PDD2(k)     = ttt(3);                 QDD2(k)     = ttt(4);
    PDD3(k)     = ttt(5);                 QDD3(k)     = ttt(6);
    LTPD(k)     = ttt(7);

    PDD1(k) = PDD1(k)/(PBASE/3);          QDD1(k) = QDD1(k)/(PBASE/3);
    PDD2(k) = PDD2(k)/(PBASE/3);          QDD2(k) = QDD2(k)/(PBASE/3);
    PDD3(k) = PDD3(k)/(PBASE/3);          QDD3(k) = QDD3(k)/(PBASE/3);
end
%% Now arrange in bus sequence no.
for i = 1:NG+NLB
    PD1(BSN(i),1)     = PDD1(i,1);  QD1(BSN(i),1)     = QDD1(i,1);
    PD2(BSN(i),1)     = PDD2(i,1);  QD2(BSN(i),1)     = QDD2(i,1);
    PD3(BSN(i),1)     = PDD3(i,1);  QD3(BSN(i),1)     = QDD3(i,1);
    PD3(BSN(i),1)     = PDD3(i,1);
    LTP(BSN(i),1)     = LTPD(i,1);
```

```
end
%% not use to read data- but use in program.
    BIND = zeros(NB,1);
for k = 1:NB
    BIND(k) = k;
end
%% Reading  LINE CONFIGURATION

      ttt    = fscanf(ifp,'%s %s %s %s %s %s %s %s %s %s',[1,10]);
for k=1:NCON
      ttt    = fscanf(ifp,'%d %f %f %d %f %f %f %f %f %f',[1,10]);

      CONF(k)  = ttt(1);

      switch CONF(k)

          case 721
              [Z721 Y721]= CABLEZY(ttt(2), ttt(3), ttt(4), ttt(5),
ttt(6), ttt(7), ttt(8), ttt(9), ttt(10));

          case 722
              [Z722 Y722]= CABLEZY(ttt(2), ttt(3), ttt(4), ttt(5),
ttt(6), ttt(7), ttt(8), ttt(9), ttt(10));
          case 723
              [Z723 Y723]= CABLEZY(ttt(2), ttt(3), ttt(4), ttt(5),
ttt(6), ttt(7), ttt(8), ttt(9), ttt(10));
          case 724
              [Z724 Y724]= CABLEZY(ttt(2), ttt(3), ttt(4), ttt(5),
ttt(6), ttt(7), ttt(8), ttt(9), ttt(10));
       end
end

%% VOLTAGE REGULATOR - TRANSFORMER - TRANS.LINE :
% Initialize the matrix for VR , TR , TRL .
VRNO   = zeros(NT,1);   TRNO = zeros(NT,1);
FB     = zeros(NT,1);   TB = zeros(NT,1)  ;
LENGTH = zeros(NT,1);
ZZERO  = zeros(NT,1);   ZPOSITIVE = zeros(NT,1);
CONFIG = zeros(NT,1);

%% Reading voltage regulator data
      ttt    = fscanf(ifp,'%s %s %s %s %s %s %s',[1,7]);   % read - name
like : no.,fb,tb,vllh...
for k=1:NREG
      ttt    = fscanf(ifp,'%d %d %d %f %f %f %d',[1,7]);   % read - data
                VRNO(k)  = ttt(1);      % VRNO = Voltage Regulator Number
                FB= ttt(2);             % FB  = From bus
                TB= ttt(3);             % TB  = To bus
                TPA = ttt(4);           % TPA = Tapping of Phase A
                TPB = ttt(5);           % TPB =                   B
                TPC = ttt(6);           % TPC =                   C
                CONFIG(k)= ttt(7);      % CONFIG = Voltage Regulator
Configuration type Number
end
%% Reading Transformer data
      ttt    = fscanf(ifp,'%s %s %s %s %s %s %s %s %s',[1,9]); % read -
name
for k=NREG+1:NREG+NTR
      ttt     = fscanf(ifp,'%d %d %d %f %f %f %f %f %d',[1,9]);% read -
data
```

```
                TRNO(k) = ttt(1);                  % TRNO = Transformer
Number
                FB(k)   = ttt(2);
                TB(k)   = ttt(3);
                VLLH(k) = ttt(4);                  % VLLH = VLL at HV side
                VLLL(k) = ttt(5);                  % VLLL = VLL at LV side
                R(k)    = ttt(6);
                X(k)    = ttt(7);
                PRAT(k) = ttt(8);
                CONFIG(k)= ttt(9);
                R(k) = R(k)*(PBASE/PRAT(k));      % find P.U.- R(k)
                X(k) = X(k)*(PBASE/PRAT(k));
end
%% Reading transmission line data
      ttt    = fscanf(ifp,'%s %s %s %s %s %s %s',[1,7]);
for k=NREG+NTR+1:NT
      ttt    = fscanf(ifp,'%d %d %d %f %f %f %d',[1,7]);
                TRNO(k) = ttt(1);
                 FB(k)  = ttt(2);
                 TB(k)  = ttt(3);
              LENGTH(k)= ttt(4);
                ZZERO(k)= ttt(5);
           ZPOSITIVE(k)= ttt(6);
              CONFIG(k)= ttt(7);
end
DATASTR;   % read DATA STRUCTURE FUNCTION FILE
PFLOW_VLN_DFIG;
```

6.2.3. Load models description with general equations

In distribution systems, loads can be modeled as wye (star) connected or delta connected. Moreover the loads can be three-phase, two-phase, or single-phase with any unbalanced nature and can be modeled as:

1. Constant real and reactive power (Constant PQ)
2. Constant current
3. Constant Impedance
4. Any combination of above 1, 2 or 3.

Figure 6.1 shows a generic delta connected load. In the IEEE-37 test feeder system, all loads are delta connected. Therefore, only delta connected loads are modeled in this appendix.

These load models are used in the iterative process of power-flow algorithm, where load voltages are initially assumed.

For all loads, the line currents entering the load are required in order to perform the power-flow analysis.

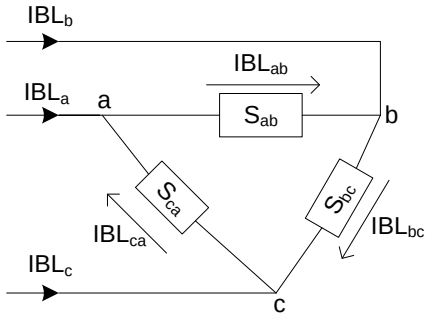

Figure 6.1: Delta connected load

The general equations of current computations for each model are:

Model	Computation of current
Constant PQ (D-PQ) (type-001)	$IBL_{LL} = (\frac{S_{LL}}{V_{LL}})^* = \frac{\lvert S_{LL}\rvert}{\lvert V_{LL}\rvert} \angle\delta_{LL} - \theta_{LL} = \lvert IBL_{LL}\rvert\angle\alpha_{LL}$ The line to line voltages will change after each iteration and as a result the new current magnitude and angles.
Constant I (D-I) (type-002)	Magnitudes of currents are computed similarly to those in the constant PQ model and held constant. Meanwhile, the angle of the voltage δ_{LL} changes after each iteration. This keeps the power factor of the load constant.
Constant Z (D-Z) (type-003)	Constant load impedance, $Z_{LL} = \frac{\lvert V_{LL}\rvert^2}{S_{LL}^*} = \frac{\lvert V_{LL}\rvert^2}{\lvert S_{LL}\rvert}\angle\theta_{LL} = \lvert Z_{LL}\rvert\angle\theta_{LL}$ The delta load IBL is a function of the constant load impedances and are given by $IBL_{LL} = \frac{V_{LL}}{Z_{LL}} = \frac{\lvert V_{LL}\rvert}{\lvert Z_{LL}\rvert} \angle\delta_{LL} - \theta_{LL} = \lvert IBL_{LL}\rvert\angle\alpha_{LL}$ V_{LL} changes after each iteration until convergence is achieved.

6.2.4. Line segments impedance and admittance matrices

```
% from this function file GET - Z and Y matrix.

function [ZABC, YABC] = CABLEZY(D, DOD, NN, DC, GMRC, RC, DS, GMRS, RS)
R = (DOD-DS)/24;
GMRN = (GMRS*NN*(R)^(NN-1))^(1/NN);
RN = RS/NN;
D1 = (D^(NN)-R^(NN))^(1/NN);
D2 = ((2*D)^(NN)-R^(NN))^(1/NN);
D3 = R;

GMRD = [GMRC D1 D2 D3 D1 D2; D1 GMRC D1 D1 D3 D1;...
        D2 D1 GMRC D2 D1 D3; D3 D1 D2 GMRN D1 D2;...
        D1 D3 D1 D1 GMRN D1; D2 D1 D3 D2 D1 GMRN;];

ZABC = zeros(3,3);
YABC = zeros(3,3);
Z1 = zeros(3,3);
Z2 = zeros(3,3);
Z3 = zeros(3,3);
Z4 = zeros(3,3);
 for i = 1:3
    for j=1:3
        if i == j     %  Gives - diagonal elements
            Z1(i,i) =
complex((RC+0.09530),(0.12134*(log(1/GMRD(i,i))+7.93402)));
            Z2(i,i) =
complex((0.09530),(0.12134*(log(1/GMRD(i+3,i))+7.93402)));
            Z3(i,i) =
complex((0.09530),(0.12134*(log(1/GMRD(i,i+3))+7.93402)));
            Z4(i,i) =
complex((RN+0.09530),(0.12134*(log(1/GMRD(i+3,i+3))+7.93402)));

        else          %  Gives - off-diagonal elements
            Z1(i,j) =
complex((0.09530),(0.12134*(log(1/GMRD(i,j))+7.93402)));
            Z2(i,j) =
complex((0.09530),(0.12134*(log(1/GMRD(i+3,j))+7.93402)));
            Z3(i,j) =
complex((0.09530),(0.12134*(log(1/GMRD(i,j+3))+7.93402)));
            Z4(i,j) =
complex((0.09530),(0.12134*(log(1/GMRD(i+3,j+3))+7.93402)));
        end
% %         YABC(i,j) = 0 ;
        YABC(i,i) = 0.000001*complex(0,77.3619/(log((R*12)/(DC/2))-
(1/NN)*log((NN*DS/2)/(R*12))));
    end
 end
 ZABC = Z1-Z2*(Z4\eye(3))*Z3;
end
```

6.2.5. Computation of a, b, c, d, A, B parameters of Series feeder components

The description of the test system includes the voltage regulator, transformer and line segments. Therefore, the parameters a, b, c, d, A, B can be obtained by using the following equations and matlab programming codes.

Equations:

	Voltage regulator	Transformer	Line Segment
	Open delta type (303)	Delta-Delta (404)	
a=	$\begin{bmatrix} aR_a & 0 & 0 \\ 0 & aR_b & 0 \\ -aR_a & -aR_b & 0 \end{bmatrix}$	$\frac{n_t}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + (\frac{1}{2})[Z_L][Y_G]$
b=	$[0]$	$Z_t * n_t * \frac{1}{9} * \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 0 \\ 1 & 2 & 0 \\ -2 & -1 & 0 \end{bmatrix}$	$[Z_L]$
c=	$[0]$	$[0]$	$[Y_G](1 + (\frac{1}{4})[Z_L][Y_G])$
d=	$\begin{bmatrix} \frac{1}{aR_a} & 0 & 0 \\ -\frac{1}{aR_a} & 0 & -\frac{1}{aR_b} \\ 0 & 0 & \frac{1}{aR_b} \end{bmatrix}$	$\frac{1}{n_t}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$[a]$
A=	$[a]^{-1}$	$\frac{1}{3 * n_t}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$	$[a] \Big/ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
B=	$[0]$	$Z_t * \frac{1}{9} * \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 0 & 2 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 0 \\ 1 & 2 & 0 \\ -2 & -1 & 0 \end{bmatrix}$	$[A] * [Z_L]$
where $aR_a = 1 - 0.00625 * tpa$, $aR_b = 1 - 0.00625 * tpb$, $aR_c = 1 - 0.00625 * tpc$.		Where $n_t = 1$	Where $[Z_L]$ and $[Y_G]$ are as per described in line- segment section.

Matlab Codes:

```
% For Voltage Regulator : File Nmae : VRabcdAB
function [a b c d A B] = VRabcdAB(tpa, tpb, tpc, con)
W   = (1/3)*[2 1 0;0 2 1; 1 0 2];
aRa = 1-0.00625*tpa;
aRb = 1-0.00625*tpb;
aRc = 1-0.00625*tpc;

%Wye
if con == 301
    a = [aRa 0 0; 0 aRb 0; 0 0 aRc];
    b = zeros(3,3);
    c = zeros(3,3);
    d = [1/aRa 0 0; 0 1/aRb 0; 0 0 1/aRc];
    A = inv(a);
    B = zeros(3,3);
end
%Delta
if con == 302
    a = [aRa 1-aRb 0; 0 aRb 1-aRc; 1-aRa 0 aRc];
    b = zeros(3,3);
    c = zeros(3,3);
    d = [aRa 0 1-aRc; 1-aRa aRb 0; 0 1-aRb aRc];
    A = inv(a);
    B = zeros(3,3);
end
%Open delta
if con == 303
    a = [aRa 0 0; 0 aRb 0; -aRa -aRb 0];
    a = W*a*(W^(-1));
    b = zeros(3,3);
    c = zeros(3,3);
    d = [1/aRa 0 0; -1/aRa 0 -1/aRb; 0 0 1/aRb];
    d = W*d*(W^(-1));
    A = [1/aRa 0 0; 0 1/aRb 0; -1/aRa -1/aRb 0];
    A = W*A*(W^(-1));
    B = zeros(3,3);
end
-----------------------------------------------------------------------
% For Transformer , File Name : TRabcdAB
function [a, b, c, d, A, B] = TRabcdAB(PBASE, VLLH, VLLL, R, X, con)
zt= complex(R,X);
W = (1/3)*[2 1 0;0 2 1; 1 0 2];
G = (1/3)*[1 (-1) 0 ; 1  2  0 ; (-2)  (-1)  0 ];
%Delta grounded wye
if con == 401
    nt = sqrt(3);
    a = (-nt/3)*[0 2 1; 1 0 2; 2 1 0];
    b = a*zt*eye(3);
    c = zeros(3,3);
    d = (1/nt)*[1 -1 0; 0 1 -1; -1 0 1];
    A = (1/nt)*[1 0 -1;-1 1 0;0 -1 1];
    B = zt*eye(3);
end
%ungrounded wye delta
if con == 402
    nt = (1/sqrt(3));
    a = nt*[0 -1 0; 0 1 -1; -1 0 1];
    b = (nt/3)*zt*[1 -1 0; 1 2 0; -2 -1 0];
    c = zeros(3,3);
    d = (1/(3*nt))*[1 -1 0; 1 2 0; -2 -1 0];
```

```
    A = (1/(3*nt))*[2 1 0;0 2 1;1 0 2];
    B = (1/3)*zt*[1 0 0; 0 1 0; -1 -1 0];
end
%grounded wye grounded wye
if con == 403
    nt= 1;
    a = nt*eye(3);
    b = zt*a;
    c = zeros(3,3);
    d = (1/nt)*eye(3);
    A = d;
    B = zt*eye(3);
end
%delta delta
if con == 404
    nt = 1; % VLLH/VLLL=1pu ;
    a = (W^(-1))*(nt)*W ;
    b = W* nt * G * zt ;
    c = zeros(3,3);
    d = (1/nt)*eye(3);
    A = (W^(-1))*(nt^(-1))*W ;
    B = W*G*zt ;

end
%Open wye open delta
if con == 405
    nt =1/sqrt(3);
    a = nt*[1 -1 0; 0 1 -1;0 0 0];
    b = nt*zt*[1 0 0; 0 0 -1;0 0 0];
    c = zeros(3,3);
    d = (1/nt)*[1 0 0; 0 0 -1;0 0 0];
    A = (1/(3*nt))*[2 -1 0;-1 1 0;-1 -2 0];
    B = (zt/3)*[2 0 -1; -1 0 -1; -1 0 -2];
end
-----------------------------------------------------------------------
% For Distribution Feeder/line, File Name:TLabcdAB
function [a, b, c, d, A, B] = TLabcdAB(ZL, YG)
a= eye(3)+0.5*ZL*YG;
b= ZL;
c= YG+0.25*YG*ZL*YG;
d= eye(3)+0.5*YG*ZL;
A= a\eye(3);
B= A*ZL;
```

6.2.6. Proposed DFIG-Model algorithm in Matlab programming code

6.2.6.1. DFIG Model Algorithm Main Function file.

```
function FUN_DFIG_PROG_VLN_W12_012

global Vsa Vsb Vsc Vs0 Vs1 Vs2 Zero Pos Neg Zero0 Pos0 Neg0 Vr0 Vr1 Vr2
global p P Fs Rs Rr Lms Xm Zs Zs0 Zs1 Zs2 Ws Xr Xs s s0 s1 s2 Zr0 Zr1 Zr2
Zm Zn Zm0 Zm1 Zm2 VLN vga vgb vgc dvga dvgb dvgc Lg Xg
global Is0 Is1 Is2 Ir0 Ir1 Ir2 E0 E1 E2 seq Pm0 Pm1 Pm2 aseq Pm012  Wwind
Wr  bitta  Pm_3ph
global PD1_dfig  QD1_dfig  PD2_dfig  QD2_dfig  PD3_dfig  QD3_dfig
VLNstator1 VLNstator2 VLNstator3 PD1 PD2 PD3 QD1 QD2 QD3

 %% INPUT  PARAMETERS :

Wwind = 14
Wr = 1849*(2*pi/60)
bitta =0

fun_windturbine_model;   % Function file for Wind Turbine Model

Pm012=Pm_3ph/3

% Pm012 =(-4.5161e+005)  ;
Sbase = 1.5e6 ; Vbase = 2400 ;
p = 2 ;P = 2*p ;Fs= 60 ;
Rs = 0.029 ; Rr = 0.022  ;
Lls = 0.226/377 ;  Llr = 0.226/377 ;  Lms = 13.04/377 ; Lg = 0.345/377 ;
Xs=2*pi*60*Lls  ;  Xr = 2*pi*60*Llr ; Xm = 2*pi*60*Lms ;Xg = 2*pi*60*Lg ;

Zn = 0 ; Zm = 0 ; % assume Neutral and Mutual Impedance = 0
%% Slip Components for sequence network
Ws = 2*pi*Fs/p    ;
s  = (Ws-Wr)/Ws   ;

s0 = s ;   s1 = s ;   s2 = 2-s;
%% Phase Operator
aseq = 1*(cos(2*pi/3)+ 1j*sin(2*pi/3)) ;  % Phase Operator - a - that
cause a counter clockwise rotation of 120 degree of all phases

%% Transfer Grid/stator Voltage (p.u) to Stator Voltage (si)

VLNstator1 = 1 *(cos(1.036*(2*pi/360))  + 1j*sin(1.036*(2*pi/360))) ;
VLNstator2 = 1 *(cos((1.036-120)*(2*pi/360))+ 1j*sin((1.036-
120)*(2*pi/360))) ;
VLNstator3 = 1 *(cos((1.036+120)*(2*pi/360))+
1j*sin((1.036+120)*(2*pi/360))) ;

%     VLNstator1 = VLN(38,1) ;
%     VLNstator2 = VLN(38,2) ;
%     VLNstator3 = VLN(38,3) ;

Display_VLNstator_Vbus_pu=[abs(VLNstator1) , angle(VLNstator1)*360/(2*pi)
; abs(VLNstator2) , angle(VLNstator2)*360/(2*pi) ; abs(VLNstator3) ,
angle(VLNstator3)*360/(2*pi) ]

Vsa =
((abs(VLNstator1))*(Vbase/sqrt(3)))*complex(cos(angle(VLNstator1)),sin(an
gle(VLNstator1))) ;
```

```
Vsb =
((abs(VLNstator2))*(Vbase/sqrt(3)))*complex(cos(angle(VLNstator2)),sin(an
gle(VLNstator2))) ;
Vsc =
((abs(VLNstator3))*(Vbase/sqrt(3)))*complex(cos(angle(VLNstator3)),sin(an
gle(VLNstator3))) ;

Display_Vsabc_SI = [abs(Vsa) , angle(Vsa)*360/(2*pi); abs(Vsb) ,
angle(Vsb)*360/(2*pi) ;abs(Vsc) , angle(Vsc)*360/(2*pi) ]
%%
% V012 = inv(A) * Vabc
Vs0 = (1/3)* ( Vsa +       Vsb  +       Vsc  ) ;
Vs1 = (1/3)* ( Vsa + (aseq*Vsb) + ((aseq^2)*Vsc)) ;
Vs2 = (1/3)* ( Vsa +((aseq^2)*Vsb) +      aseq*Vsc)  ;
Display_Vs012_SI =
[abs(Vs0),angle(Vs0)*(360/(2*pi));abs(Vs1),angle(Vs1)*(360/(2*pi));abs(Vs
2),angle(Vs2)*(360/(2*pi))] ;%%%%%%%%%%%

%% SYMMETRICAL SEQUENCE IMPEDANCE COMPONENTS :
% Zs abc >>>   to >>>> Zs 012
Zs    = Rs + 1i * Xs  ;
Zs0 = Zs + 3*Zn + 3*Zm  ; Zs1 = Zs - Zm  ;Zs2 = Zs - Zm   ;

% Zr abc >>> to >>>> Zr 012
Zr0 = complex(Rr/s0,Xr) + 3*Zn + 3*Zm ; Zr1 = complex(Rr/s1,Xr)- Zm ; Zr2
= complex(Rr/s2,Xr) -  Zm   ;

% Zm abc >>> to >>>> Zm 012
Zm0 = j*Xm + 3*Zn + 3*Zm ; Zm1 = j*Xm - Zm ; Zm2 = j*Xm - Zm ;

%% Find Vr 012 from Pm012 :
options = optimset ('Display','iter');   % Option to display output

Zero0 = [-1e-5 , -1e-6 ];Pos0 = [37.032 -40.229]; Neg0 = [-1e-4 , -0.1] ;
X0 = [-1e-5 , -1e-6 , 37.032 -40.229, -1e-4 , -0.1] ;

[X,fval] = fsolve( @myfunfr012, X0 , options);  % Call solver
Zero(1) = X(1);Zero(2) = X(2);
Pos(1) = X(3);Pos(2) = X(4);
Neg(1) = X(5);Neg(2) = X(6);

Vr0 =  Zero(1)+ i * Zero(2);
Vr1 =  Pos(1) + i * Pos(2);
Vr2 =  Neg(1) + i * Neg(2);

Vs0 ;Is0 ; Ir0 ; E0 ;  Vr0 ;
Vs1 ;Is1 ; Ir1 ; E1  ; Vr1 ;
Vs2 ;Is2 ; Ir2 ; E2  ; Vr2 ;

Display_Vr_012 = [abs(Vr0), angle(Vr0)*(360/(2*pi));abs(Vr1),
angle(Vr1)*(360/(2*pi));abs(Vr2), angle(Vr2)*(360/(2*pi))]
Display_Vs_012 =
[abs(Vs0),angle(Vs0)*(360/(2*pi));abs(Vs1),angle(Vs1)*(360/(2*pi));abs(Vs
2),angle(Vs2)*(360/(2*pi))]
Display_E_012  =
[abs(E0),angle(E0)*(360/(2*pi));abs(E1),angle(E1)*(360/(2*pi));abs(E2),an
gle(E2)*(360/(2*pi))]
Display_Ir_012 =
[abs(Ir0),angle(Ir0)*(360/(2*pi));abs(Ir1),angle(Ir1)*(360/(2*pi));abs(Ir
2),angle(Ir2)*(360/(2*pi))]
```

```
Display_Is_012 = [abs(Is0),angle(Is0)*(360/(2*pi));
abs(Is1),angle(Is1)*(360/(2*pi)) ; abs(Is2) , angle(Is2)*(360/(2*pi))]

%% Stator & Rotor Voltages for phase "a" ; "b" and phase "c"
%% Stator Voltages : from Vso12 >>>to>>>  Vsa , Vsb , Vsc
Vsa = Vs0 +  Vs1        + Vs2        ;
Vsb = Vs0 +  Vs1 *(aseq^2) + Vs2 * aseq    ;
Vsc = Vs0 +  Vs1 * aseq    + Vs2 *(aseq^2) ;
Display_Vsabc = [abs(Vsa) , angle(Vsa)*(360/(2*pi)); abs(Vsb) ,
angle(Vsb)*(360/(2*pi)) ; abs(Vsc) , angle(Vsc)*(360/(2*pi))] ;%%%%%%%%%%

% Rotor Voltages : from Vro12 >>>to>>>  Vra , Vrb , Vrc
Vra = Vr0   +  Vr1          +  Vr2        ;
Vrb = Vr0   +  Vr1 * (aseq^2)  +  Vr2 * aseq    ;
Vrc = Vr0   +  Vr1 * aseq      +  Vr2 *(aseq^2) ;
Display_Vr_abc = [abs(Vra) , angle(Vra)*(360/(2*pi)); abs(Vrb) ,
angle(Vrb)*(360/(2*pi)) ; abs(Vrc) , angle(Vrc)*(360/(2*pi))] %%%%%%%%%%

%%  Stator & Rotor Currents for phase "a" ; "b" and phase "c"
%% Ir 012   >>>>   to   >>>>>  Ir  abc :
Ira =  Ir0 +       Ir1  +        Ir2   ;
Irb =  Ir0 +  (aseq^2*Ir1) +      (aseq*Ir2) ;
Irc =  Ir0 +    (aseq*Ir1) +   ((aseq^2)*Ir2) ;
Display_Ir_abc = [abs(Ira) , angle(Ira)*(360/(2*pi)); abs(Irb) ,
angle(Irb)*(360/(2*pi)) ; abs(Irc) , angle(Irc)*(360/(2*pi))]

% Is 012   >>>>   to   >>>>>  Is  abc :
Isa =  Is0 +       Is1  +        Is2   ;
Isb =  Is0 +  (aseq^2*Is1) +      (aseq*Is2) ;
Isc =  Is0 +    (aseq*Is1) +  ((aseq^2)*Is2) ;
Display_Is_abc = [abs(Isa) , angle(Isa)*(360/(2*pi)); abs(Isb) ,
angle(Isb)*(360/(2*pi)) ; abs(Isc) , angle(Isc)*(360/(2*pi))]

%% Find Pm0 , Pm1 and Pm2

Pm0 = ((abs(Ir0)^2)*Rr)*((1-s0)/(s0))-(real(Vr0*conj(Ir0))*((1-
s0)/(s0)));
Pm1 = ((abs(Ir1)^2)*Rr)*((1-s1)/(s1))-(real(Vr1*conj(Ir1))*((1-
s1)/(s1)));
Pm2 = ((abs(Ir2)^2)*Rr)*((1-s2)/(s2))-(real(Vr2*conj(Ir2))*((1-
s2)/(s2)));

Display_Pm0_Pm1_Pm2 = [ Pm0 , Pm1 , Pm2 ]
%% Stator & Rotor Power  -
Psa = real (Vsa *conj(Isa)) ; Psb = real (Vsb *conj(Isb)) ; Psc = real
(Vsc *conj(Isc));
Qsa = imag (Vsa *conj(Isa)) ; Qsb = imag (Vsb *conj(Isb)) ; Qsc = imag
(Vsc *conj(Isc));
Pra = real (Vra *conj(Ira)) ; Prb = real (Vrb *conj(Irb)) ; Prc = real
(Vrc *conj(Irc));
Qra = imag (Vra *conj(Ira)) ; Qrb = imag (Vrb *conj(Irb)) ; Qrc = imag
(Vrc *conj(Irc));

%% Tolal Resistive / copper losses
Pscl  = ((abs(Isa)^2)*Rs)+ ((abs(Isb)^2)*Rs) + ((abs(Isc)^2)*Rs);
Prcl  = ((abs(Ira)^2)*Rr)+ ((abs(Irb)^2)*Rr) + ((abs(Irc)^2)*Rr);
Ploss =  Pscl + Prcl ;

Pscla=((abs(Isa)^2)*Rs) ;
Psclb=((abs(Isb)^2)*Rs) ;
Psclc=((abs(Isc)^2)*Rs) ;
```

```
Prcla=((abs(Ira)^2)*Rr);
Prclb=((abs(Irb)^2)*Rr);
Prclc=((abs(Irc)^2)*Rr);

Pcla= Pscla + Prcla ;
Pclb= Psclb + Prclb ;
Pclc= Psclc + Prclc ;

Display_Pcl_abc = [ Pcla , Pclb , Pclc]

Pga= Pra ; Pgb=Prb ; Pgc = Prc ;
Qga=-Qsa ; Qgb=-Qsb ; Qgc=-Qsc ;

vga= sqrt((( Xg * Pga / abs(Vsa))^2 ) + (((Xg* Qga +
(abs(Vsa))^2)/abs(Vsa))^2));
vgb= sqrt((( Xg * Pgb / abs(Vsb))^2 ) + (((Xg* Qgb +
(abs(Vsb))^2)/abs(Vsb))^2));
vgc= sqrt((( Xg * Pgc / abs(Vsc))^2 ) + (((Xg* Qgc +
(abs(Vsc))^2)/abs(Vsc))^2));

dvga= angle(Vsa) - atan(vga);
dvgb= angle(Vsb) - atan(vgb);
dvgc= angle(Vsc) - atan(vgc);

Display_Vg_abc = [vga, dvga*360/(2*pi); vgb, dvgb*360/(2*pi)
;vgc,dvgc*360/(2*pi) ]

%% Bus real and reactive Power in abc frame:
P_a = Psa + Pra ;
P_b = Psb + Prb ;
P_c = Psc + Prc ;
Q_a = Qsa + Qra ;
Q_b = Qsb + Qrb ;
Q_c = Qsc + Qrc ;

P3ph_abc = P_a + P_b + P_c   ;
Q3ph_abc = Q_a + Q_b + Q_c   ;
S3ph_abc = P3ph_abc + j*Q3ph_abc  ;

Display_P_Q_abc_si = [P_a , Q_a , P_b , Q_b , P_c , Q_c ]

%% Transfer DFIG power to Power Flow Programme :
PD1_dfig = P_a / (Sbase/3) ;
PD2_dfig = P_b / (Sbase/3) ;
PD3_dfig = P_c / (Sbase/3) ;
QD1_dfig = 0 ;
QD2_dfig = 0 ;
QD3_dfig = 0 ;

PD1(38,1)= PD1_dfig  ;  PD2(38,1)= PD2_dfig  ;
PD3(38,1)= PD3_dfig  ;
QD1(38,1)= QD1_dfig  ;
QD2(38,1)= QD2_dfig  ;
QD3(38,1)= QD3_dfig  ;

Display_PD_QD_pu = [PD1_dfig , QD1_dfig , PD2_dfig , QD2_dfig , PD3_dfig
, QD3_dfig ]
end
```

6.2.6.2. Power Balance Equation Solve

```
function F = myfunfr012(X)

global Vs0 Rs Rr  Zs0 s0  Zr0  Zm0  Is0 Ir0 E0    Vr0
global  Vs1 Rs Rr  Zs1 s1 Zr1 Zm1 Is1 Ir1 E1
global Vs2 Rs Rr  Zs2  s2 Zr2 Zm2 Is2 Ir2 E2 seq  Pm012

Zero(1) = X(1);Zero(2) = X(2);Pos(1) = X(3);
Pos(2) = X(4);Neg(1) = X(5);Neg(2) = X(6);

Vr0 = Zero(1) + i * Zero(2);
E0    = ((Vs0/Zs0)+((Vr0/s0)/Zr0)) / ((1/Zs0) + (1/Zr0) + (1/(Zm0))) ;
Ir0   = ((Vr0/s0)-E0) / Zr0 ;
Is0   = (Vs0-E0) / Zs0 ;

Vr1 = Pos(1) + i * Pos(2);
E1    = ((Vs1/Zs1)+((Vr1/s1)/Zr1)) / ((1/Zs1) + (1/Zr1) + (1/(Zm1))) ;
Ir1   = ((Vr1/s1)-E1) / Zr1 ;
Is1   = (Vs1-E1) / Zs1 ;

Neg(1)
Neg(2)

Vr2 = Neg(1) + i * Neg(2);
E2 = ((Vs2/Zs2)+((Vr2/s2)/Zr2)) / ((1/Zs2) + (1/Zr2) + (1/(Zm2)));
Ir2   = ((Vr2/s2)-E2) / Zr2 ;
Is2   = (Vs2-E2) / Zs2 ;

F =    (real(Vs0*conj(Is0))+  real(Vs1*conj(Is1)) + real(Vs2*conj(Is2)))
;
F = F - (abs (Is0)^2 * Rs + abs (Is1)^2 * Rs + abs (Is2)^2 * Rs) ;
F = F - ( abs (Ir0)^2 * Rr + abs (Ir1)^2 * Rr + abs (Ir2)^2 * Rr ) ;
F = F + (real((Vr0)*conj(Ir0)) + real((Vr1)*conj(Ir1))+ real((Vr2)*
conj(Ir2)));
F = F - ((Pm012)) ;
```

6.2.6.3. Wind Turbine Model

```
function [Pm_3ph] = fun_windturbine_model(Wwind,Wr,bitta )
global Wwind Wr  bitta  Pm_3ph
%% for simplicity in analysis
v_actual     = Wwind          ;
Wr_actual    = Wr             ;
bitta        = bitta          ;
% Nominal / Base Values :
Pmec       = 1500e3           ;% Wind Turbine nominal mech output power
: Pmec = watt
Pelect     = 1500e3/0.9       ;% Nominal Electrical Generator's base
power : Pelec = Watt
v_base     = 12               ;% Wind velocity : v_base = m/s.         (
Base )
Wr_base    = (1800) * (2*pi/60);% Generator speed : Wr_base = rad/sec   (
Base )
Wt_base    = 1.2 * Wr_base    ;% Wind Turbine's nominal speed = p.u. of
Generator's nominal speed (at x-axis of curve)
                               % means , generator run's (1.2 times )
faster than  Wind Turbine
```

```
Pmec_max_base_wind=0.73*Pmec ; % Maximum mech Power at v_base  = p.u. of
Turbine's nominal power ( at y-axis of curve)
Pem = 1.01054 ; % power transformation constant related with matlab
propritory code
%% other base values :
lambda_base = 8.10   ;                    % 4956268 ; % tip speed ratio
nominal constant value
Cp_base     = 0.48                    ;   % Cp max
%% Constant coefficients  :
c1 = 0.5176 ; c2 = 116  ; c3 = 0.4  ; c4 = 5   ;  c5 = 21   ; c6 = 0.0068
;
%% To find Lambda
Wr_pu = Wr_actual / Wr_base               ;
Wt_pu = Wr_pu *Wr_base / Wt_base          ;
v_pu = v_actual / v_base                  ;
lambda_pu = Wt_pu / v_pu                  ;
lambda_actual = lambda_pu * lambda_base   ;
%% To find Cp
lambda_i = 1/(1/(lambda_actual+0.08*bitta)-0.035/(bitta^3+1));
Cp_actual = c1*(c2/lambda_i-c3*bitta-c4)*exp(-
c5/lambda_i)+c6*lambda_actual ;
Cp_pu = Cp_actual / Cp_base  ;
%% To find Pwind  ;  Pturbine  ; Pgenerator
Pwind_pu   = (v_pu)^3  ;
Pmec_pu    = Cp_pu * Pwind_pu  ;
Pelect_pu  = Pmec_pu  * ( Pmec_max_base_wind / Pelect ) ;  % get max
electrical power in pu.
%% OUTPUTs : Pmech (in form of elect) , Wr ( elect) , Tmech (on machine
shaft in form of elect)
Pelect_actual = (-1) * Pelect_pu  *  Pelect  ;
Telect = Pelect_actual /  Wr_actual  ;
Pm_3ph = Pem*Pelect_actual;
Pm_1ph = Pm_3ph/3;
Wr = Wr_actual ;
Tm_3ph = Telect ;
end
```

6.2.7. Ladder algorithm for Power flow analysis in Matlab

```
function PFLOW_VLN_DFIG

global Z721 Z722 Z723 Z724 Y721 Y722 Y723 Y724
global NB NS NT
global ZL
global LENGTH CONFIG PBASE
global a b c d A B V1 V2 I2 VLL VLN IB IBL AIBL ZB m n index enode enode2
enode3 enode4 line IE VBASE IL
global PD1 QD1 PD2 QD2 PD3 QD3 LTP VLLH VLLL R X TOLER W D err_E E IP PG
VP
global CON TPA TPB TPC VF IF  IT

%% Initialise I and V:
 IBL= zeros(NB,3);
AIBL= zeros(NB,3);
  IB= zeros(NB,3);
 ZB = zeros(NB,3);
 VLN = zeros(NB,3);
 VLL = zeros(NB,3);
 IP = zeros(NB,3);
 VP = zeros(NB,3);
[m,n] = size(CON);
D = [1 -1 0; 0 1 -1; -1 0 1];
W = (1/3)*[2 1 0;0 2 1; 1 0 2];

%% Step-3 %%%333333333333333333333333333333333333333333333333
% ERROR
E=1;
IT=1
%%
for i = 1:NB
%% Step-4   %%%%4444444444444444444444444444444444444444444
VLN(i,1:3) = [1, 1*complex(-0.5,-0.866025), 1*complex(-0.5,0.866025)] ;
VLL(i,1:3) = (W^(-1))*(VLN(i,:).');

%% Step-5  %%%%555555555555555555555555555555555555555555555
    switch LTP(i)
        case 001    % D-PQ Type Load
            IBL(i,1:3)=[conj(complex(PD1(i),QD1(i))/VLL(i,1))-
conj(complex(PD3(i),QD3(i))/VLL(i,3))...
                       conj(complex(PD2(i),QD2(i))/VLL(i,2))-
conj(complex(PD1(i),QD1(i))/VLL(i,1))....
                       conj(complex(PD3(i),QD3(i))/VLL(i,3))-
conj(complex(PD2(i),QD2(i))/VLL(i,2))];
        case 002    % D-I Type Load
            IBL(i,1:3)=[conj(complex(PD1(i),QD1(i))/VLL(i,1))-
conj(complex(PD3(i),QD3(i))/VLL(i,3))...
                       conj(complex(PD2(i),QD2(i))/VLL(i,2))-
conj(complex(PD1(i),QD1(i))/VLL(i,1))....
                       conj(complex(PD3(i),QD3(i))/VLL(i,3))-
conj(complex(PD2(i),QD2(i))/VLL(i,2))];
          AIBL(i,1:3)= [abs(IBL(i,1)) abs(IBL(i,2)) abs(IBL(i,3))];
        case 003    % D-Z Type Load
            ZB(i,1:3)=[abs((VLL(i,1)))^2/conj(complex(PD1(i),QD1(i)))...
                      abs((VLL(i,2)))^2/conj(complex(PD2(i),QD2(i)))....
                      abs((VLL(i,3)))^2/conj(complex(PD3(i),QD3(i)))];

            IBL(i,1:3)=[VLL(i,1)/ZB(i,1)-VLL(i,3)/ZB(i,3)...
                        VLL(i,2)/ZB(i,2)-VLL(i,1)/ZB(i,1)....
```

```
                        VLL(i,3)/ZB(i,3)-VLL(i,2)/ZB(i,2)];

        case 011      % Y - PQ Load
            IBL(i,1:3)=[conj(sqrt(3)*complex(PD1(i),QD1(i))/VLL(i,1))...
                        conj(sqrt(3)*complex(PD2(i),QD2(i))/VLL(i,2))....
                        conj(sqrt(3)*complex(PD3(i),QD3(i))/VLL(i,3))];

        case 012      % Y - I Type Load
            IBL(i,1:3)=[conj(sqrt(3)*complex(PD1(i),QD1(i))/VLL(i,1))...
                        conj(sqrt(3)*complex(PD2(i),QD2(i))/VLL(i,2))....
                        conj(sqrt(3)*complex(PD3(i),QD3(i))/VLL(i,3))];

            AIBL(i,1:3) = [abs(IBL(i,1)) abs(IBL(i,2)) abs(IBL(i,3))];

        case 013      % Y - Z Type Load

ZB(i,1:3)=[abs(VLL(i,1)/sqrt(3))^2/conj(complex(PD1(i),QD1(i)))...

abs(VLL(i,2)/sqrt(3))^2/conj(complex(PD2(i),QD2(i)))....

abs(VLL(i,3)/sqrt(3))^2/conj(complex(PD3(i),QD3(i)))];

            IBL(i,1:3)=[VLL(i,1)/(sqrt(3)*ZB(i,1))...
                        VLL(i,2)/(sqrt(3)*ZB(i,2))....
                        VLL(i,3)/(sqrt(3)*ZB(i,3))];
    end
end
IB=IBL ;
%% Step-6   %%%%%6666666666666666666666666666666666666666666666
 while IT<100 && abs(E) > TOLER*1

% % FORWARD SWEEP :
IL= zeros(NT,3);  IE= zeros(NT,1); V2= zeros(3,3); I2= zeros(3,3); PG =
zeros(NB,3);
enode = 0;

%% Step-7  %%%%777777777777777777777777777777777777777777777

% For LF-1
% PD1(38)= -0.8917 ; PD2(38)= -0.8917 ; PD3(38)= -0.8917 ;
% QD1(38)= 0  ; QD2(38)= 0  ; QD3(38)= 0  ;
% For LF-2
FUN_DFIG_PROG_VLN_W12_012 ;

%% Step-8    %%%%888888888888888888888888888888888888888888888888
 for i = 1:NB
%% Step-8A    %%%%8A8A8A8A8A8A8A8A8A8A8A8A8A8A8A8A
               if CON(i,1)== 0   % branch end = 0
                  enode = i;     % enode = receiving end specific bus
number.
                        switch LTP(enode)    % Update loads

                            case 001

IBL(enode,1:3)=[conj(complex(PD1(enode),QD1(enode))/VLL(enode,1))-
conj(complex(PD3(enode),QD3(enode))/VLL(enode,3))...

conj(complex(PD2(enode),QD2(enode))/VLL(enode,2))-
conj(complex(PD1(enode),QD1(enode))/VLL(enode,1))....
```

```
conj(complex(PD3(enode),QD3(enode))/VLL(enode,3))-
conj(complex(PD2(enode),QD2(enode))/VLL(enode,2))];
                              case 002

IBL(enode,1:3)=[conj(complex(PD1(enode),QD1(enode))/VLL(enode,1))-
conj(complex(PD3(enode),QD3(enode))/VLL(enode,3))...

conj(complex(PD2(enode),QD2(enode))/VLL(enode,2))-
conj(complex(PD1(enode),QD1(enode))/VLL(enode,1))....

conj(complex(PD3(enode),QD3(enode))/VLL(enode,3))-
conj(complex(PD2(enode),QD2(enode))/VLL(enode,2))];


IBL(enode,1:3)=[AIBL(enode,1)*complex(cos(angle(IBL(enode,1))),sin(angle(
IBL(enode,1))))...

AIBL(enode,2)*complex(cos(angle(IBL(enode,2))),sin(angle(IBL(enode,2)))).
...

AIBL(enode,3)*complex(cos(angle(IBL(enode,3))),sin(angle(IBL(enode,3))))]
;
                              case 003
                                  IBL(enode,1:3)=[VLL(enode,1)/ZB(enode,1)-
VLL(enode,3)/ZB(enode,3)...

VLL(enode,2)/ZB(enode,2)-VLL(enode,1)/ZB(enode,1)....

VLL(enode,3)/ZB(enode,3)-VLL(enode,2)/ZB(enode,2)] ;
                              case 011

IBL(enode,1:3)=[conj(sqrt(3)*complex(PD1(enode),QD1(enode))/VLL(enode,1))
...

conj(sqrt(3)*complex(PD2(enode),QD2(enode))/VLL(enode,2))....

conj(sqrt(3)*complex(PD3(enode),QD3(enode))/VLL(enode,3))]  ;
                              case 012

IBL(enode,1:3)=[conj(sqrt(3)*complex(PD1(enode),QD1(enode))/VLL(enode,1))
...

conj(sqrt(3)*complex(PD2(enode),QD2(enode))/VLL(enode,2))....

conj(sqrt(3)*complex(PD3(enode),QD3(enode))/VLL(enode,3))];


IBL(enode,1:3)=[AIBL(enode,1)*complex(cos(angle(IBL(enode,1))),sin(angle(
IBL(enode,1))))...

AIBL(enode,2)*complex(cos(angle(IBL(enode,2))),sin(angle(IBL(enode,2)))).
...

AIBL(enode,3)*complex(cos(angle(IBL(enode,3))),sin(angle(IBL(enode,3))))]
;
                              case 013

IBL(enode,1:3)=[VLL(enode,1)/(sqrt(3)*ZB(enode,1))...

VLL(enode,2)/(sqrt(3)*ZB(enode,2))....
```

```
VLL(enode,3)/(sqrt(3)*ZB(enode,3))];
                        end
                    IB(enode,1:3)= IBL(enode,1:3);
                end

%% Step-8B      %%%%%8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B8B
                while enode > 0
                        for k=1:(n-1)/2     % k = 1:3  , n=7
                            for j=1:NB
                                if CON(j,2*k+1)== enode
                                     enode2 = j ;
                                   line = CON(j,2*k);
                                end
                            end
                        end
                        switch CONFIG(line)
                                case 721
                                                    ZL =
LENGTH(line)*(1/5.28)*Z721/(VBASE^2/PBASE);
                                                    YG =
LENGTH(line)*(1/5.28)*Y721*(VBASE^2/PBASE);
                                                [a b c d A B]= TLabcdAB(ZL,YG);
                                case 722
                                                    ZL =
LENGTH(line)*(1/5.28)*Z722/(VBASE^2/PBASE);
                                                    YG =
LENGTH(line)*(1/5.28)*Y722*(VBASE^2/PBASE);
                                                [a b c d A B]= TLabcdAB(ZL,YG);
                                case 723
                                                    ZL =
LENGTH(line)*(1/5.28)*Z723/(VBASE^2/PBASE);
                                                    YG =
LENGTH(line)*(1/5.28)*Y723*(VBASE^2/PBASE);
                                                [a b c d A B]= TLabcdAB(ZL,YG);
                                case 724
                                                    ZL =
LENGTH(line)*(1/5.28)*Z724/(VBASE^2/PBASE);
                                                    YG =
LENGTH(line)*(1/5.28)*Y724*(VBASE^2/PBASE);
                                                 [a b c d A B]= TLabcdAB(ZL,YG);
                                case 303
                                                 [a b c d A B]= VRabcdAB(TPA(line),
TPB(line), TPC(line), CONFIG(line));
                                case 404
                                                 [a b c d A B]= TRabcdAB(PBASE,
VLLH(line), VLLL(line), R(line), X(line), CONFIG(line));
                                 otherwise
                                                    ZL = LENGTH(line)*(1/5.28)*
Z(CONFIG(line))/(VBASE^2/PBASE);
                                                    YG = LENGTH(line)*(1/5.28)*
Y(CONFIG(line))*(VBASE^2/PBASE);
                                                [a b c d A B]= TLabcdAB(ZL,YG) ;
                        end
                        V2 = a*[VLN(enode,1); VLN(enode,2);
VLN(enode,3);] + b*[IB(enode,1); IB(enode,2); IB(enode,3);] ;
                        I2 = c*[VLN(enode,1); VLN(enode,2);
VLN(enode,3);] + d*[IB(enode,1); IB(enode,2); IB(enode,3);] ;
                                                % enode = receiving end bus
no.
                        IL(line,1:3)= I2 ;
```

```
                        IE(line,1) = 1    ;
                        VLN(enode2,1:3)= V2 ;

                         VLL(enode2,1:3)= ((W^(-1))*[VLN(enode2,1);
VLN(enode2,2); VLN(enode2,3)]).'  ;  % VLN to VLL  convert

%% Step - 8C    %%%%%%%8C8C8C8C8C8C8C8C8C8C8C
                        switch LTP(enode2)    % Update loads

                           case 001

IBL(enode2,1:3)=[conj(complex(PD1(enode2),QD1(enode2))/VLL(enode2,1))-
conj(complex(PD3(enode2),QD3(enode2))/VLL(enode2,3))...

conj(complex(PD2(enode2),QD2(enode2))/VLL(enode2,2))-
conj(complex(PD1(enode2),QD1(enode2))/VLL(enode2,1))....

conj(complex(PD3(enode2),QD3(enode2))/VLL(enode2,3))-
conj(complex(PD2(enode2),QD2(enode2))/VLL(enode2,2))];
                           case 002

IBL(enode2,1:3)=[conj(complex(PD1(enode2),QD1(enode2))/VLL(enode2,1))-
conj(complex(PD3(enode2),QD3(enode2))/VLL(enode2,3))...

conj(complex(PD2(enode2),QD2(enode2))/VLL(enode2,2))-
conj(complex(PD1(enode2),QD1(enode2))/VLL(enode2,1))....

conj(complex(PD3(enode2),QD3(enode2))/VLL(enode2,3))-
conj(complex(PD2(enode2),QD2(enode2))/VLL(enode2,2))];


IBL(enode2,1:3)=[AIBL(enode2,1)*complex(cos(angle(IBL(enode2,1))),sin(ang
le(IBL(enode2,1))))...

AIBL(enode2,2)*complex(cos(angle(IBL(enode2,2))),sin(angle(IBL(enode2,2))
))....

AIBL(enode2,3)*complex(cos(angle(IBL(enode2,3))),sin(angle(IBL(enode2,3))
))];
                           case 003

IBL(enode2,1:3)=[VLL(enode2,1)/ZB(enode2,1)-VLL(enode2,3)/ZB(enode2,3)...

VLL(enode2,2)/ZB(enode2,2)-VLL(enode2,1)/ZB(enode2,1)....

VLL(enode2,3)/ZB(enode2,3)-VLL(enode2,2)/ZB(enode2,2)] ;
                           case 011

IBL(enode2,1:3)=[conj(sqrt(3)*complex(PD1(enode2),QD1(enode2))/VLL(enode2
,1))...

conj(sqrt(3)*complex(PD2(enode2),QD2(enode2))/VLL(enode2,2))....

conj(sqrt(3)*complex(PD3(enode2),QD3(enode2))/VLL(enode2,3))]  ;
                           case 012

IBL(enode2,1:3)=[conj(sqrt(3)*complex(PD1(enode2),QD1(enode2))/VLL(enode2
,1))...

conj(sqrt(3)*complex(PD2(enode2),QD2(enode2))/VLL(enode2,2))....
```

```
conj(sqrt(3)*complex(PD3(enode2),QD3(enode2))/VLL(enode2,3))];


IBL(enode2,1:3)=[AIBL(enode2,1)*complex(cos(angle(IBL(enode2,1))),sin(ang
le(IBL(enode2,1))))...

AIBL(enode2,2)*complex(cos(angle(IBL(enode2,2))),sin(angle(IBL(enode2,2))
))....

AIBL(enode2,3)*complex(cos(angle(IBL(enode2,3))),sin(angle(IBL(enode2,3))
))] ;
                              case 013

IBL(enode2,1:3)=[VLL(enode2,1)/(sqrt(3)*ZB(enode2,1))...

VLL(enode2,2)/(sqrt(3)*ZB(enode2,2))....

VLL(enode2,3)/(sqrt(3)*ZB(enode2,3))];
                            end
%% Step - 8D  %%%%%%%%%%8D8D8D8D8D8D8D8D
                              if enode2==NS
                                 enode=0;
                                 IB(enode2,1:3)= IL(CON(enode2,2),1:3) +
IBL(enode2,1:3) ;

                              elseif CON(enode2,1)== 1
                                  enode=enode2;
                                  IB(enode2,1:3)=
IL(CON(enode2,2),1:3)+IBL(enode2,1:3) ;

                              elseif (CON(enode2,1)== 2) &&
(IE(CON(enode2,2),1)+IE(CON(enode2,4),1)==2)
                                  enode=enode2;
                                  IB(enode2,1:3)=
IL(CON(enode2,2),1:3)+IL(CON(enode2,4),1:3)+IBL(enode2,1:3);

                              elseif (CON(enode2,1)== 3) &&
(IE(CON(enode2,2),1)+IE(CON(enode2,4),1)+IE(CON(enode2,6),1)==3)
                                  enode=enode2;
                                  IB(enode2,1:3)=
IL(CON(enode2,2),1:3)+IL(CON(enode2,4),1:3)+IL(CON(enode2,6),1:3)+IBL(eno
de2,1:3);

                              else
                                  enode=0;
                              end

                 end

      end

%% Step-9  Check ERROR  %%%%%%%%99999999999999999999999999999999
V1=[1 , 1*complex(-0.5,-0.866025) , 1*complex(-0.5,0.866025)] ;     % s/s
voltage 1 p.u.
E = max([abs((V1(1,1)-VLN(NS,1)))+ abs((V1(1,2)-VLN(NS,2)))+
abs((V1(1,3)-VLN(NS,3)))]);   % Error
err_E=abs(E)

%% Step-10 : for backward sweep  %%%10101010101010101010101010101010
```

```
        VLN  = zeros(NB,3);
        VLN(NS,1:3)= V1 ;

        VE=zeros(1,NB);
        VE(1,NS)=1   ;
        line = CON(NS,2) ;
        index=0;
%% Step-10 A                  %%%%10A10A10A10A10A10A10A
         while index < NB
                for k=1:(n-1)/2     %   k = 1:3
                        for j=1:NB
                                if CON(j,2*k)== line
                                    enode3 = j;
                                    enode4 = CON(j,2*k+1);
                                end
                        end
                end

                    switch CONFIG(line)    % Line Configuration
                            case 721
                                ZL =
LENGTH(line)*(1/5.28)*Z721/(VBASE^2/PBASE);
                                YG =
LENGTH(line)*(1/5.28)*Y721*(VBASE^2/PBASE);
                                [a b c d A B]= TLabcdAB(ZL,YG);
                            case 722
                                ZL =
LENGTH(line)*(1/5.28)*Z722/(VBASE^2/PBASE);
                                YG =
LENGTH(line)*(1/5.28)*Y722*(VBASE^2/PBASE);
                                [a b c d A B]= TLabcdAB(ZL,YG);
                            case 723
                                ZL =
LENGTH(line)*(1/5.28)*Z723/(VBASE^2/PBASE);
                                YG =
LENGTH(line)*(1/5.28)*Y723*(VBASE^2/PBASE);
                                [a b c d A B]= TLabcdAB(ZL,YG);
                            case 724
                                ZL =
LENGTH(line)*(1/5.28)*Z724/(VBASE^2/PBASE);
                                YG =
LENGTH(line)*(1/5.28)*Y724*(VBASE^2/PBASE);
                                [a b c d A B]= TLabcdAB(ZL,YG);
                            case 303
                                [a b c d A B]= VRabcdAB(TPA(line),
TPB(line), TPC(line), CONFIG(line));
                            case 404
                                [a b c d A B]=
TRabcdAB(PBASE,VLLH(line),VLLL(line),R(line),X(line),CONFIG(line));
                    end

               V2 = A*[VLN(enode3,1); VLN(enode3,2); VLN(enode3,3)]-
B*[IB(enode4,1); IB(enode4,2); IB(enode4,3)];

               VLN(enode4,1:3)= V2 ;
%%
VLL(enode4,1:3) =(W^(-1))*(VLN(enode4,1:3).');

%%  Step 10-B  %%%%%%%%%%10B10B10B10B10B10B10B10B
               VE(1,enode4)=1 ;
               index = sum(VE);
```

```
                    for m = 1:NB
                          for r=1:(n-1)/2
                                if CON(m,2*r+1)~=0 && VE(CON(m,2*r+1))==0
&& VE(m)==1
                                     line = CON(m,2*r);
                                break
                                end
                          end
                    end
%% Step 10-C  %%%%%%%%%10C10C10C10c10c10c01C01C
          end
 %%  results :
            VF = zeros(NB,7);
            IF = zeros(NB,7);
            VF_VLN=zeros(NB,7);
            VF_VLL=zeros(NB,7);
            PD_QD =zeros(NB,7);

            for i = 1:NB
            VLL(i,1:3) = (W^(-1))*(VLN(i,:).');

            VF_VLN(i,1:7)= [i ...
                              abs(VLN(i,1)) 180*angle(VLN(i,1))/pi....
                              abs(VLN(i,2)) 180*angle(VLN(i,2))/pi....
                              abs(VLN(i,3)) 180*angle(VLN(i,3))/pi] ;

            VF_VLL(i,1:7)= [i ...
                              abs(VLL(i,1)) 180*angle(VLL(i,1))/pi....
                              abs(VLL(i,2)) 180*angle(VLL(i,2))/pi....
                              abs(VLL(i,3)) 180*angle(VLL(i,3))/pi] ;

            VF (i,1:7) = [i ...
                             abs(VLL(i,1))/sqrt(3)
(180*angle(VLL(i,1))/pi)-30....
                             abs(VLL(i,2))/sqrt(3)
(180*angle(VLL(i,2))/pi)-30....
                             abs(VLL(i,3))/sqrt(3)
(180*angle(VLL(i,3))/pi)-30] ;

            IF(i,1:7) = [i ...
                            (2500/(3*4.8))*abs(IB(i,1))
180*angle(IB(i,1))/pi ....
                            (2500/(3*4.8))*abs(IB(i,2))
180*angle(IB(i,2))/pi ....
                            (2500/(3*4.8))*abs(IB(i,3))
180*angle(IB(i,3))/pi];

            PD_QD(i,1:7)= [i PD1(i) QD1(i) PD2(i) QD2(i) PD3(i) QD3(i)]
;
            end
PD_QD
VF_VLN

%% Step 11 UPGRADE ITERATION   %%%%1111111111111111111111111111111
IT=IT+1

end
%% Step-12  Solution %%%%%%12121212121212121212121212121212121212121
PD_QD;
VLN_FINAL = VF_VLN
VF
```

References

[1] J. A. Baroudi, V. Dinavahi, A. M. Knight, "A review of power converter topologies for wind generators", *ELSEVIER-Renewable Energy*, vol. 32, no.14, Nov.2007, pp. 2369-2385.

[2] T. Ackermann, *"Wind Power in Power Systems"*, John Wiley & Sons Ltd, 2005 pp. 64-69.

[3] The Mathworks Inc., "*Simpower system^{TM} 5 Reference*", 2009, pp.2-17-2- 30, 2-881-2-292.

[4] S. Khushalani, J. M. Solanki, N. N. Schulz, " Development of three-phase unbalanced power flow using PV and *PQ* models for distributed generation and study of the impact of DG Models", *IEEE Transactions on Power Systems*, vol.22, No.3, August 2007, pp.1019- 1025.

[5] J. F. M. Padrón and A. E. Feijóo Lorenzo, "Calculating Steady-State Operating Conditions for Doubly-Fed Induction Generator Wind Turbines", *IEEE Transactions on Power Systems*, vol. 25, no. 2, May 2010, pp. 922-928.

[6] U. Eminoglu, "A New Model for Wind Turbine Systems", *Electric Power Components and Systems*, Taylor & Francis, vol. 37, pp. 1180- 1193, 2009.

[7] K. C. Divya, P. S. Nagendra Rao, "Models for wind turbine generating systems and their application in load flow studies", *Electric Power System Research*, vol. 76, no. 9-10, June 2006, pp.844-856.

[8] A. E. Feijóo and J. Cidras, "Modeling of Wind Farms in the Load flow Analysis", *IEEE Transactions on Power Systems*, vol. 15, no. 1, Feb. 2000, pp. 110-115.

[9] M. Zhao, Z. Chen, F. Blaabjerg, "Load flow analysis for variable speed offshore wind farms", *IET Renewable Power Generation*, vol. 3, Iss.2, 2009, pp. 120-132.

[10] J. H. Teng, "Modelling distributed generations in three-phase distribution load flow", *IET Gener. Transm. Distrib.*, vol. 2, No. 3, pp 330-340, 2008.

[11] A. Feijóo, "On *PQ* Models for Asynchronous Wind Turbines", IEEE Transactions on Power Systems, vol. 24, No. 4, Nov. 2009, pp.1890-1891.

[12] T. Burton, D. Sharpe, N. Jenkins, E. Bossanyi, *"Wind Energy Hand Book"*, John Wiley & Sons, 2001, pp. 6-45.

[13] I. Papic, "Mathematical analysis of FACTS devices based on a voltage source converter Part 1: mathematical models", *Electric Power System Research,* vol.56, May 2000, pp.139-148.

[14] E. Acha, V. G. Agelidis, O. Anaya-Lara, T. J. E. Miller, *"Power Electronics control in Electrical System*", Newnes Power Engineering series, 2002, pp. 1-75, 113-116.

[15] E. Acha, C. Angeles, O. L. Tortelli, C. R. Fuerte-Esquivel, "Inclusion of a high voltage DC-voltage source converter model in a Newton-Raphson power flow algorithm", *IEE Proc.- Gener. Transm. Distrib.*, vol. 150, No. 6, Nov. 2003, pp. 691-696.

[16] G. Zhang, Z. Xu, Y. Cai, "An Equivalent Model for Simulating VSC based HVDC", *IEEE-PES Transmission & Distribution Conf. & Exposition*, Oct/Nov. 2001, pp. 20-24.

[17] A. Yazdani, R. Iravani, "Voltage-Source Converters in Power Systems", Wiley, *IEEE press* 2010 pp 25-150.

[18] W. H. Kersting, W. H. Phillips, "Phase Frame Analysis of the effects of Voltage Unbalance on Induction Machines", *IEEE Transactions on Industry Applications*., vol. 33, No.2, March/April 1997, pp. 415-420.

[19] W. H. Kersting, "Distribution System Modeling and Analysis", second edition, CRC Press, 2007, pp. 302-350.

[20] I. Boldea, "*Variable Speed Generators*", CRC Press, 2006, pp.1-10–13.

[21] IEEE PES, "Distribution Test feeders",-37 bus feeder, http://ewh.ieee.org/soc/pes/dsacom/testfeeders/index.html

[22] Balasubramaniam Babypriya, Rajapalan Anita, "Modelling, Simulation and Analysis of Doubly Fed Induction Generator for Wind Turbines", *journal of Electrical Engineering*, vol. 60, No. 2, 2009, pp. 79-85

[23] Nasser Tleis, "Power System Modelling and Fault Analysis, Theory and Practice", Elsevier, Newnes, First edition, 2008, pp. 301-396.

[24] Canadian Wind Energy Association, http://www.canwea.ca/media/index_e.php

[25] Richard Gagnon, "Hydro Quebec Models Wind Power Plant Performance". http://www.mathworks.cn/controlsystems/userstories.html?file=51076&title=Hydro-Qu%E9bec%20Models%20Wind%20Power%20Plant%20Performance

[26] Matlab Simulink demo of Wind Farm- DFIG Average Model, power_wind_dfig_avg.mdl, Richard Gagnon, Hydro-Quebec.