

618813409
73
820
483
5708

DIRECT VOLTAGE CONTROL FOR STAND-ALONE WIND ENERGY CONVERSION SYSTEMS WITH ENERGY STORAGE

by

WEI HUANG

BSc, Huazhong University of Science and Technology, China, 2005

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, Canada

© Wei Huang 2008

**PROPERTY OF
RYERSON UNIVERSITY LIBRARY**

ABSTRACT

Direct Voltage Control for Stand-alone Wind Energy Conversion Systems with Energy Storage

Wei Huang

Electrical and Computer Engineering

Ryerson University

Toronto 2008

A control method for the stand-alone wind power generation system with induction generator and energy storage devices is proposed in this thesis. A fixed-speed self-excited induction generator is directly connected to the standalone power system, while battery powered energy storage devices are employed to balance the system power flow. A DC-AC power converter is connected between the energy storage device and the standalone power system, which maintains the voltage and frequency constant. Direct voltage control with current limits is developed for the converter with dynamic fast response. Mathematical models are developed to analysis the system performance as well as to design the lead-lag regulators in the control system. The proposed system is verified in the simulation and experiment.

ACKNOWLEDGMENTS

I wish to express my deep gratitude to my supervisor, Professor Dewei Xu for his support and the knowledge he shared during my graduate studies at Ryerson University.

I am grateful to Professor Bin Wu, Professor Richard Cheung, Dr. Yongqiang Lang, and all fellow students at LEDAR for their useful discussions on my research.

I also wish to share my achievements with my families and my wife Wei Li. I am very grateful for her understanding and support.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION.....	1
1.1 Wind Energy Conversion System	1
1.2 Stand-alone Wind Energy Conversion System.....	4
1.3 Motivation and Objective	7
1.4 Thesis Outline	8
CHAPTER 2 MODEL DEVELOPMENT FOR THE STANDALONE WECS	10
2.1 Introduction	10
2.2 Wind Turbine Characteristic	11
2.3 Induction Generator Model.....	12
2.4 Inductive Load and Self-Excitation Capacitor Model	15
2.5 Energy Storage System Model	15
2.6 Standalone WECS Model	16
2.7 Small Signal Model.....	17
2.8 Conclusions	20
CHAPTER 3 CONTROL OF STANDALONE WECS.....	21
3.1 Introduction	21
3.2 Direct Voltage Control.....	22
3.2.1 Voltage and Frequency Control	22
3.2.2 Direct Voltage Controller Design	23
3.3 Conclusions	28
CHAPTER 4 SIMULATION AND EXPERIMENTAL RESULTS.....	29
4.1 Introduction	29
4.2 Simulation Model	30
4.3 Simulation Results.....	32
4.3.1 Non-ESS Stand Alone Wind Power Generation System.....	32
4.3.2 Simulation Results with ESS	33
4.4 Experimental Setup.....	37
4.5 Experimental Results and Analysis	40

4.5.1	Resistive Load Test.....	40
4.5.2	Inductive Load Test	44
4.6	Conclusions	48
CHAPTER 5 CONCLUSIONS		49
5.1	Conclusions	49
5.2	Major Contributions	50
5.3	Further Research Works	50
REFERENCES.....		52
APPENDICES.....		55
Appendix A Simulation Models.....		55
Appendix B System Parameters		56
Appendix C Code for Bode Plots.....		57
Appendix D DSP Program Code		58

LIST OF TABLES

Table 3-1: Parameters of the Lab Prototype System	25
Table 3-2: Parameters of lead-lag Controller.....	27
Table 4-1: System parameters of prototypes system	31
Table B-1 Parameters of DC Motor.....	56
Table B-2: System Parameters of Prototypes System.....	56

LIST OF FIGURES

Figure 1-1: Typical diagram of wind energy conversion systems	2
Figure 1-2: Constant-speed wind energy conversion systems	2
Figure 1-3: Typical configurations of variable speed wind energy conversion systems	3
Figure 1-4: Schematic of general wind-diesel system	5
Figure 1-5: Proposed wind energy conversion system	8
Figure 2-1: Proposed wind energy conversion system	10
Figure 2-2: Variables in three-phase (<i>abc</i>) stationary frame and two-phase (<i>dq</i>) synchronous frame	13
Figure 2-3: Induction generator equivalent circuits under synchronous reference frame	14
Figure 2-4: Energy storage system circuit	16
Figure 2-5: d-q Equivalent circuits in synchronous reference frame	17
Figure 2-6: Small signal model for d-q equivalent circuits in synchronous reference frame	20
Figure 3-1: Control block diagram of the whole system	21
Figure 3-2: Block diagram using transfer functions	22
Figure 3-3: System blocks model in S- domain	24
Figure 3-4: Bode plots of uncompensated loop	25
Figure 3-5: Bode plots of uncompensated loop with different loads	26
Figure 3-6: Bode plot of compensated loop gain	27
Figure 4-1: Block diagram of experimental set-up	29
Figure 4-2: System models of proposed WECS	30
Figure 4-3: Direct voltage controller model	31
Figure 4-4: Wind power generation system without ESS	32
Figure 4-5: System voltage and frequency without ESS	32
Figure 4-6: Local bus voltage and frequency during load step change (with R Load) in Simulation	33
Figure 4-7: d-q axis inverter voltage during load step change (with R Load) in Simulation	34
Figure 4-8: d-q axis inverter current during load step change (with R Load) in Simulation	34
Figure 4-9: Current waveforms during load step change (with R load) in simulation	35
Figure 4-10: Voltage and frequency during load step change (with RL load) in simulation	36
Figure 4-11: Currents waveforms during load step change (with RL load) in simulation	36
Figure 4-12: d-q axis inverter voltages during load step change (with RL Load) in Simulation	37
Figure 4-13: d-q axis inverter currents during load step change (with RL Load) in Simulation	37
Figure 4-14: Experimental setup	38

Figure 4-15: DSP controller board..... 39

Figure 4-16: Steady-State waveforms with resistive load 41

Figure 4-17: Transient waveforms of local bus voltage and load current during load step change 42

Figure 4-18: Transient waveforms of generator and converter currents during load step change (with R Load)..... 43

Figure 4-19: d-q axis Inverter side current during load step change (with R Load)..... 44

Figure 4-20: Steady-State voltage and current waveforms with RL load..... 45

Figure 4-21: Transient waveforms of local bus voltage and load current during load step change (with RL load)..... 46

Figure 4-22: Transient waveforms of currents of generator and inverter during load step change (with RL load)..... 46

Figure 4-23: d-q axis Inverter side current and voltage during step load change (with RL Load)..... 47

Figure 4-24: d-q axis Inverter current during step load change with compensation (with R-L Load) 47

Figure A-1: Simulink model of WECS..... 55

Figure A-2: Direct voltage controller model 55

LIST OF PRINCIPAL SYMBOLS

ρ	Air density (kg/m ³)
A	Turbine swept area (m ²)
v_{wind}	Wind velocity (m/s)
λ	Tip speed ratio of the blade tip speed to wind speed
β	Blade pitch angle (deg)
P_m	Mechanical output power of the turbine (W)
ω_e	Angular electrical speed
ω_r	Angular rotor speed
T_e	Electrical torque of the induction machine
$i_{ds}(t), i_{qs}(t)$	Stator side current of induction generator in d-q reference frame
$i_{dr}(t), i_{qr}(t)$	Rotor side current of induction generator in d-q reference frame
$v_{ds}(t), v_{qs}(t)$	Local bus terminal voltage in d-q reference frame
$v_{dr}(t), v_{qr}(t)$	Rotor side voltage of induction generator in d-q reference frame
$i_{Cd}(t), i_{Cq}(t)$	Self-excited capacitor current in d-q reference frame
$v_{ld}(t), v_{lq}(t)$	Load voltage in d-q reference frame
$i_{ld}(t), i_{lq}(t)$	Load current in d-q reference frame
$v(t)$	Converter output voltage
$i(t)$	Converter output current
E_s	Ideal voltage source representing the energy storage devices

L	Inductance of converter side filter
C	Capacitance of self-excited capacitor
f_{ref}	Reference frequency for the local bus
v_{ref}	Reference voltage for the local bus
v	Converter-side feedback voltage
T_s, f_s	Sampling period, sampling frequency
G_{c0}	Gain of the lead-lag regulator
T_i, T_z, T_p	Time constant of lead-lag regulator
K_{PWM}	Gain of the PWM converter
m_a	Modulation index of the converter

Chapter 1 Introduction

Currently the world's energy consumption is greatly dependent on exhaustible fossil fuel, which is also main source contributing the green house gas and pollution to the environment. In this perspective, utilization of renewable energy, such as wind and solar energy, has gained considerable momentum since the oil crises of the 1970s. Wind power, as one of the alternative energy options, is gaining increasing significance throughout the world. Wind energy is non-depleting, site-dependent, non-polluting, becoming the most promising energy source for the future. Large and grid tied wind energy conversion system has been widely installed in many countries, in the effort to minimize the dependence on fossil based non-renewable fuels [1]. Wind energy has experienced accelerated expansion in recent years and its global production is predicted to grow from 60,000 MW in 2005 to 300,000 MW in 2015 [2].

1.1 Wind Energy Conversion System

A basic wind energy conversion system (WECS) consists of a wind turbine with tower, gearbox, generator, power converter and control system, as shown in Figure 1-1[3]. There are two types of wind turbines used in practice: vertical wind turbine and horizontal wind turbine according to the rotating of the blades. Most modern wind turbines are horizontal type with two or three blades due to higher efficiency and higher power capability.

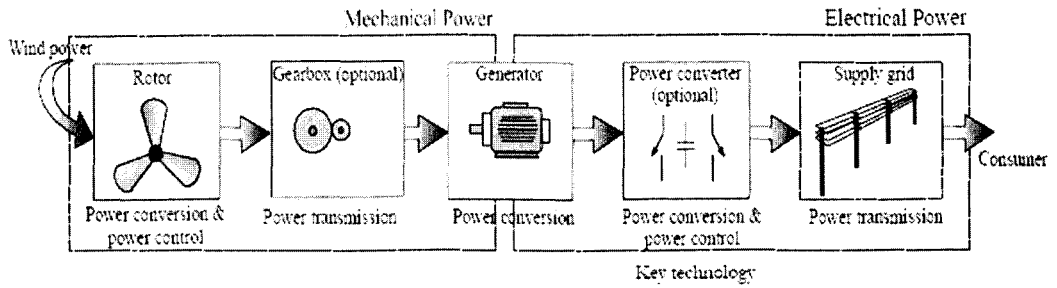


Figure 1-1: Typical diagram of wind energy conversion systems

WECS can also be classified into two types of the systems according to its operation: constant speed or variable speed operation systems. Fix speed WECS operate at a nearly fix speed while variable speed system adjusts its rotor speed according to different wind speed. Generally variable speed system extracts more wind energy (10-15%) than the fixed speed system.

A constant speed WECS usually contains a squirrel cage induction generator, as shown in Figure 1-2. Gearbox is necessary for the coupling of the low speed wind turbine and the high speed induction generator. SCR controlled starter between the generator and the grid aims at reducing the inrush current during connection to the grid. In normal operation, induction generator absorbs reactive power. Capacitor bank provides the required reactive power for generator. If enough capacitor is provided, the generator can be self-excited, which is useful when grid is not available. The speed of the generator varies little when the power changes. Thus the speed is almost constant. Constant speed WECS is usually easier to built up and cost less than the variable speed WECS. But because of the fixed speed, the overall energy conversion efficiency is not high.

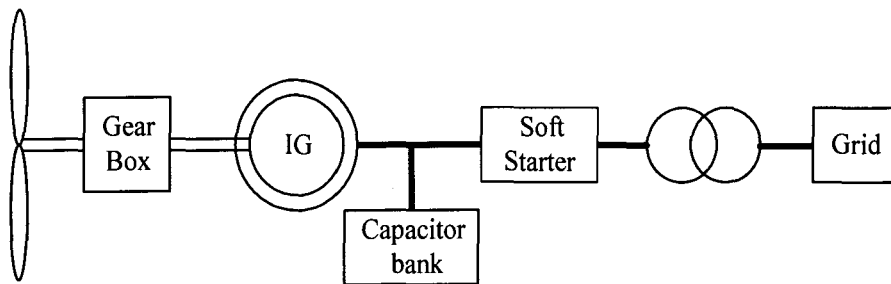
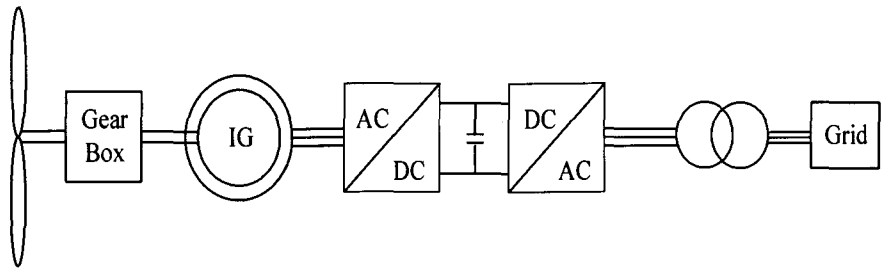
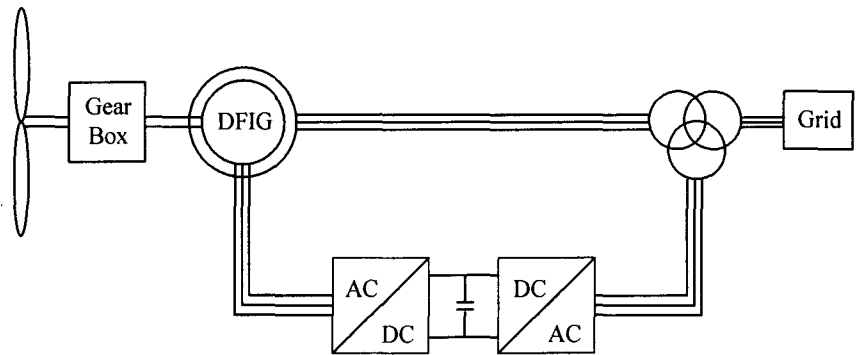


Figure 1-2: Constant-speed wind energy conversion systems

Variable speed configurations allow the rotor to rotate at required speed by introducing power electronic converters between the generator and the grid. Wind turbine system can operate at its optimum tip-speed ratio[4]. Thus more energy can be converted into electricity. Double-fed induction generators (DFIG), wound field synchronous generators and permanent magnet synchronous generators are three typical generators used in high power WECS. For medium and small power wind turbines, permanent magnet generators and squirrel cage induction generators are more often used because of their reliability and cost effectiveness. The typical configurations of variable speed WECS are shown in Figure 1-3[5].



(a) Inductor generator with full power converter



(b) Doubly-fed induction generator with fractional power converter

Figure 1-3: Typical configurations of variable speed wind energy conversion systems

Power converters are used to control the power flow between the wind turbine and grid when the variable frequency and variable voltage energy from generator is fed to the grid with fixed frequency and voltage. Most modern power converters are forced commutated PWM converters to provide a fixed voltage and frequency output with a high power capability. Both voltage source voltage controlled converters and voltage source current controlled converters have been

applied in wind energy applications. For certain high power wind turbines, effective power flow control can be achieved where real power and reactive power are independently controlled.

Most of the large wind energy conversion systems are grid connected. And many turbines are installed in a utility level wind farm with several hundred MW installed capacity. The prediction and power fluctuation are the main problem for the integration of wind farm to the grid. Many research efforts have been to solve these problems.

WECS sometimes can be used in the standalone power system in remote area where grid connection is not available. Hybrid system with battery, diesel generator and wind turbine/generator are connected together to provide a quality power to the load. The systems are often used in the small community; the capacity varies from several kilo-watts to hundreds of kilo-watt.

1.2 Stand-alone Wind Energy Conversion System

Many remote communities around the world cannot be physically or economically connected to an electric power grid. The electricity demand in these areas is conventionally supplied by small isolated diesel plants. These have the advantage of being able to deliver the required power whenever it is necessary. However, they also suffer from a number of drawbacks. Diesel generator engines are inherently noisy and expensive to run, especially for consumers in rural areas where fuel delivery costs may be high. In addition, small consumers always have a low load factor; this in turn reduces the overall efficiency and increases percentage maintenance costs. Considerable fuel savings can be achieved by integrating renewable sources such as wind and/or solar energy with existing diesel plants.

However, many off-grid places are in regions of high wind energy potential. Wind energy is one of the most important and promising forms of renewable energy sources. Its use is becoming more and more popular nowadays. This is because the price of fossil fuels is continuously increasing and the wind energy is a clean and inexhaustible energy source. But due to great varia-

tion in wind speed which occurs from season to season, it can not be used as an autonomous source of generation. Hence, it is necessary to explore possibilities of combining a wind generator with the diesel generator in order to reduce the running cost per kWh and to increase the reliability of the power supply[6].

A schematic diagram of a wind-diesel system is shown in Fig. 1-4. A classic wind-diesel power system consists of the following components: one or more wind turbines, one or more diesel generators, consumer load, an additional controllable dump load, energy storage system and control unit[7]. In order to save the fuels cost, it is desirable to shut off the diesel generator(s) whenever possible. Energy storage system is used to compensate the fluctuation in the system coming from the wind fluctuation and to provide the short period energy during the diesel generator on or off periods. For the excess power from wind turbines, dump load will be used to resolve this problem.

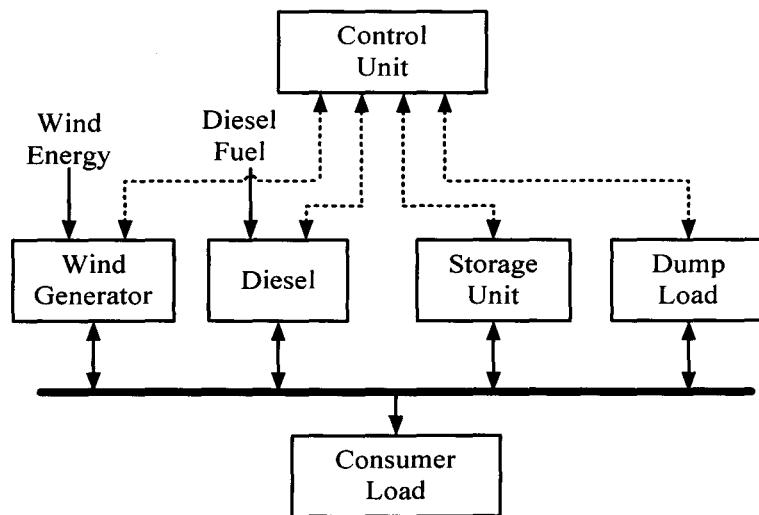


Figure 1-4: Schematic of general wind-diesel system

From the configuration of wind-diesel power system, it obviously shows that wind-diesel hybrid system has its own problems. Power fluctuation issue is one of the most important issues. It is well known that the mechanical shaft power obtained from a wind turbine may be approximated by[8] :

$$P_m = \frac{1}{2} \rho \pi R^2 C_p(\lambda, \beta) V^3 \quad (1-1)$$

where ρ is the air density, $C_p(\lambda, \beta)$ is the power coefficient, R is the blade radius, β is the blade pitch angle, λ is the tip speed ratio and V is the effective wind speed respectively.

Equation (1-1) shows that small variations in the wind speed produce large changes in the captured power. These power fluctuations transferred to the output by the conversion system, including generator, power converter if existed, especially when the system is a fix speed system. For variable speed systems, part of the power fluctuation is absorbed as kinetic energy in the turbine. However, even for variable speed systems, power fluctuations can still be a serious problem if the generator is feeding a weak grid or a stand-alone load. For these applications, the wind turbine is augmented by an additional source, usually a diesel generator. In such hybrid systems, wind speed fluctuations not only produce fluctuations in the generator output voltage, but also an unacceptable number of start/stop cycles of the diesel generator if a temporary energy buffer is not available.

In [11-14], flywheel energy storage device is used as energy storage system to smooth the fluctuations in the wind-diesel system when the energy storage system is needed. Papers [11, 12] suggest that use vector control to control the induction machine to drive the flywheel system. [13] also shows a new sensorless field oriented control method to drive the induction machine. Almost all the control methods are based on the front-end converter. There are other papers [15-17] using battery bank or super-capacitor as the energy storage devices. The different kinds of energy storage device have its own advantages and disadvantages. Battery is a cost effective solution which is widely used in the energy storage system. But it has a high maintenance cost and low cycle life. Super-capacitor is a new energy storage device with much higher life but the capacity is low.

In [9, 10], sensorless control for double-fed induction generator in stand-alone system are presented. For the Doubly-fed induction generator, sensorless operation is desirable because the use of a position encoder has several drawbacks in term of robustness, cost, cabling and maintenance.

nance. Most of the sensorless control methods are all based on the principle of rotor position observers using the rotor current error. For DFIG, itself can regulate the output voltage and the frequency regardless of the load and rotational speed. Batteries, as the energy storage devices, are integrated into the dc link of back-to-back converter for the DFIG, which enhance the system performance by expanding the operation range.

1.3 Motivation and Objective

Based on the configuration of the stand-alone system presented in the previous sections, the typical system consists of wind turbine, diesel engine, energy storage system and control part. In this project, the induction generator is used to be the wind turbine generator. For a small scale wind-diesel wind power system, squirrel cage induction generator is a better choice when cost and reliability are the first two considerations. As long as the energy storage devices are always need, a fixed-speed self-excited induction generator has its own advantages including cost, robustness and reliability can be used in the stand-alone system. The fluctuation in the power from the wind can be smoothed by the energy storage system with an appropriate controller.

As presented in the former section, many researches have been done on doubly-fed induction generator control. It usually uses a fractional size back-to-back converter to control the active power and reactive power to keep the constant voltage and frequency. And the controller usually contains two control loops to regulate the voltage and frequency. For a doubly-fed generator based system, the control of system voltage and frequency are very complicated since it is coupled by the control of induction generator's speed and torque.

Most of the control strategies for power balance controlling in the stand-alone system contain two cascaded loops: voltage loop and frequency loop, which based on the assumption that frequency loop is usually slower than the voltage loop. In these control systems, the dynamic response of grid voltage sometimes is not satisfied when the load is light.

In this thesis, the induction generator with self-excitation capacitor is directly connected to the standalone power system, as shown in Fig.1-5. The energy storage system (ESS) is employed to maintain the system frequency as well as voltage at collect bus. It supplies or absorbs the fractional power of the system, and keeps the system power flow balanced. When the wind energy is not enough for a long period, the back-up generators including diesel generator should be turned on to provide the rest of the energy. The renewable energy converters in the system are maintained at highest efficiency with the employed ESS, which reduces the energy consumption from diesel generators, reduces system operation cost as well as the greenhouse gas emission and pollution to the environment.

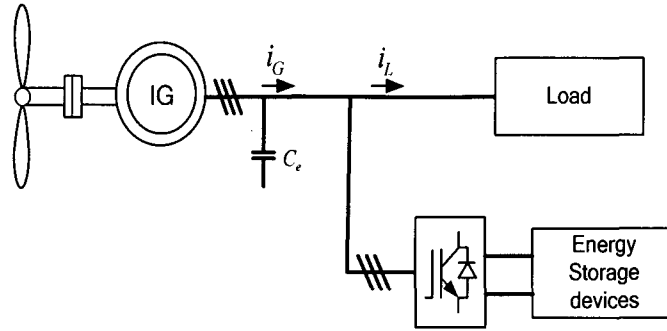


Figure 1-5: Proposed wind energy conversion system

Therefore, the main objectives of the thesis are:

1. Design a suitable controller for converter, which smooth the power fluctuation of wind energy conversion system.
2. Develop mathematical models for the system with induction generator and energy storage converter. This model will be used in the controller design.
3. Conduct variety of experiments to verify the performance of the developed control system.

1.4 Thesis Outline

This thesis is organized to provide the detailed information about the research work on standalone wind energy conversion system with induction generator and energy storage devices in five chapters.

This chapter, chapter 1 presents the background of wind energy conversion systems, including the configurations, operation and grid connection. Standalone power system with integrated wind energy is introduced. Research motivation is discussed in this chapter.

Chapter 2 presents the development of mathematical model for each component in the standalone system with wind turbine and energy storage devices. The model is developed in the d-q reference frame. Finally the whole system model is developed.

Chapter 3 presents the direct voltage controller design for standalone WECS. Small signal model of wind energy power system is developed, followed by theoretical analysis of the control scheme with dump load and current protection. The diagram of controller scheme is discussed.

Chapter 4 shows all the simulation models for different components in Matlab/Simulink. The simulation results are also given in this chapter. Experimental results are provided to compare with simulation.

Chapter 5 presents the conclusions of the thesis, major contributions and future work are discussed later.

All other relevant supporting materials are attached in appendices.

Chapter 2 Model Development for the Standalone WECS

2.1 Introduction

The diagram of the proposed wind energy conversion system is shown in Fig. 2-1. The electrical part consists of the induction generator with self-excitation capacitors, the energy storage devices with power converter, consumer load and dump load.

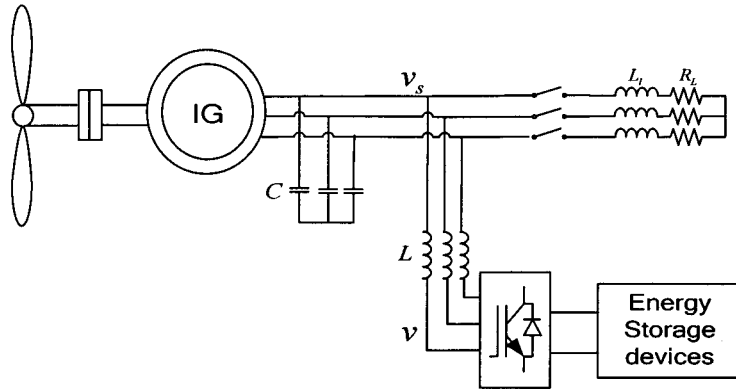


Figure 2-1: Proposed wind energy conversion system

In the system, self-excitation capacitors are needed to provide sufficient reactive power to the induction generator, which reduce the reactive power consumption from the power converter. It is assumed that the wind turbine is properly selected, which can provide most of the energy to the load. Thus the power converter and energy storage devices only provide fractional power to maintain the power balance between generator and load. In the system, the dump load will be connected to absorb these extra powers from induction generator. The dump load can be represented additional resistor. In the practical industry application, most of the energy storage devices usually can only store or supply limited power. So a back-up generator is essentially re-

quired if continuous power flow is required. In this thesis, energy storage devices are treated as an ideal voltage source. Different characteristics of the batteries are not the main research focus. Battery management is also not a consideration in this thesis.

Power fluctuation in the standalone system is one of the most common problems. Before deriving a solution for the controller, it is imperative to produce a comprehensive mathematical model of the whole electrical system that can be used in the controller design. The following sections provide different part mathematical models and the whole system model is presented in the last section.

2.2 Wind Turbine Characteristic

Generally, the wind turbine systems can be classified according to their aerodynamic configuration, transmission design and their operational speed range. Further classification can be performed based on the power output, the generator types, grid connection, etc.

The aerodynamic configuration divides the wind turbines in two main categories: the horizontal axis wind turbine (HAWT) and vertical axis wind turbine (VAWT). Due to reasons of mechanical resonance, efficiency and size more than 95 percent of the currently operating wind turbine designs are HAWT. The choice of this configuration has one major disadvantage that the main components of the energy conversion system such as generator, gearbox, pitch control mechanism, the power converter and the grid-interface device, have to be located in the nacelle of the turbine. This leads to the increased demands on the stability, strength and cost of the support structure.

Regardless of the aerodynamic configuration of the wind turbine, its absolute maximum power extraction efficiency is restricted by the Betz limit of 59.3 percent.

The mechanical power extracted from the wind is mainly governed by three quantities namely, the area swept by rotor blades, the upstream wind velocity and rotor co-efficient.

$$P_m = \frac{1}{2} \rho \pi R^2 C_p(\lambda, \beta) V^3 \quad (2-1)$$

C_p , the power coefficient of rotor, itself is a function of tip speed ratio and pitch angle. A typical coefficient of the wind turbine is given at 0° pitch angle. The optimal C_p is only possible at certain speed ratio. Yet, it is not always possible to maintain zero blade pitch, because it would create unwanted output fluctuations as well as greatly increase the electrical and mechanical load on the whole system when the wind is too strong. In order to partially cure this problem, the pitch control mechanisms (or stall control) are installed to regulate the position of the blades and maintain the highest possible power the system can deliver.

2.3 Induction Generator Model

The use of reference frame theory can simplify the analysis of electric machines and also provides a powerful tool for the digital implementation of sophisticated control schemes. The synchronous reference frame is one of the most commonly used in the research and application. The transformation of the three-phase (abc -axis) variables of an induction motor to the equivalent two-phase (dq -axis) variables can be performed by[19].

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos \theta & \cos(\theta - 2\pi/3) & \cos(\theta - 4\pi/3) \\ -\sin \theta & -\sin(\theta - 2\pi/3) & -\sin(\theta - 4\pi/3) \end{bmatrix} \cdot \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (2-2)$$

where x represents either current, voltage, or flux linkage, and θ is the angular displacement between the a -axis and d -axis of the three-phase and two-phase reference frames as shown in Fig. 2-2. The three-phase variables, x_a, x_b and x_c , are in the stationary reference frame which does not rotate in space whereas the two-phase variables, x_d and x_q , are in the synchronous reference frame whose direct (d) and quadrature (q) axes rotate in space at the synchronous speed ω_e . Note that ω_e is the angular electrical speed of the rotating magnetic field of the motor, given by

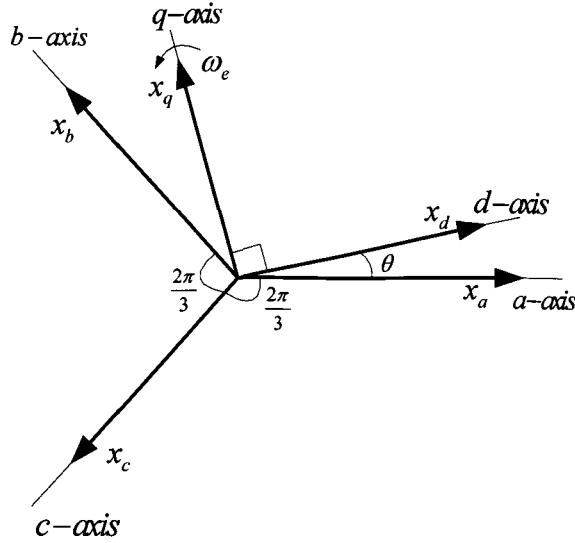


Figure 2-2: Variables in three-phase (*abc*) stationary frame and two-phase (*dq*) synchronous frame

$$\omega_e = 2\pi f_s \quad (2-3)$$

where f_s is the frequency of the stator variables. The angle θ can be found from

$$\theta(t) = \int \omega_e(t) dt + \theta_0 \quad (2-4)$$

In order to model the induction generator, the d-q equivalent circuit of induction machine is obtained. Refer all the variables of rotor side to stator side, following equations can be obtained by using Clark-Park transformation. The corresponding equivalent circuits are shown in Fig. 2-3.

$$v_{ds} = -R_s i_{ds} + \frac{d\psi_{ds}}{dt} - \omega_e \psi_{qs} \quad (2-5)$$

$$v_{qs} = -R_s i_{qs} + \frac{d\psi_{qs}}{dt} + \omega_e \psi_{ds} \quad (2-6)$$

Where:

$$\psi_{qs} = -L_{ls} i_{qs} - L_m (i_{qr} + i_{qs})$$

$$\psi_{ds} = -L_{ls} i_{ds} - L_m (i_{dr} + i_{ds})$$

Since the rotor rotates at speed ω_r , the d-q axes rotate at the speed $\omega_e - \omega_r$ referred to the rotor frame. Therefore, the equations are:

$$v_{dr} = 0 = -R_r i_{dr} + \frac{d\psi_{dr}}{dt} - (\omega_e - \omega_r) \psi_{qr} \quad (2-7)$$

$$v_{qr} = 0 = -R_r i_{qr} + \frac{d\psi_{qr}}{dt} + (\omega_e - \omega_r) \psi_{dr} \quad (2-8)$$

Where:

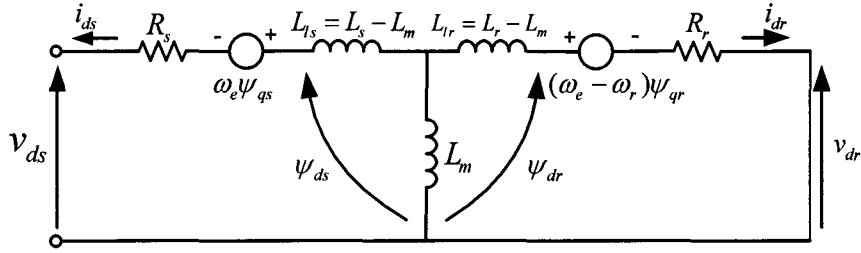
$$\psi_{dr} = -L_{lr} i_{dr} - L_m (i_{dr} + i_{ds})$$

$$\psi_{qr} = -L_{lr} i_{qr} - L_m (i_{qr} + i_{qs})$$

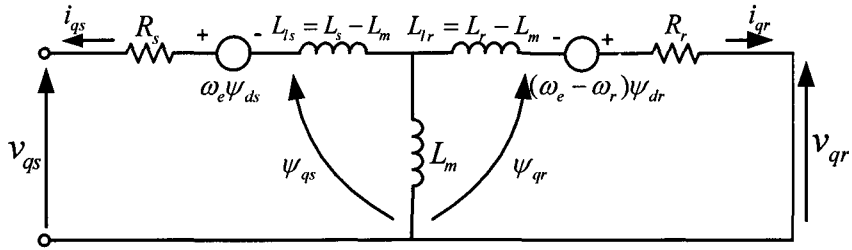
The rotor speed ω_r in equations (2-7) and (2-8) usually is treated as a constant during the electrical transient. The electrical torque of the induction machine is given by

$$T_e = \frac{3}{2} \frac{P}{2} L_m (i_{ds} i_{qr} - i_{qs} i_{dr}) \quad (2-9)$$

Where P is the number of the poles.



(a) d-axis equivalent circuit



(b) q-axis equivalent circuit

Figure 2-3: Induction generator equivalent circuits under synchronous reference frame

2.4 Inductive Load and Self-Excitation Capacitor Model

Capacitor is necessary in the standalone WECS. The capacitor can provide reactive power to the induction generator. The equations (2-12) and (2-13) represent the relation between capacitor currents and voltages in synchronous reference frame. For the load current and voltage, it has

$$i_s = i_c + i_l + i \quad (2-10)$$

$$v_l = Ri_l + \omega L_l i_l \quad (2-11)$$

Through the Clark-Park transformation, the d-q synchronous reference frame equations can be obtained:

$$i_{ds} = i_{cd} + i_{ld} + i_d + C\omega_e v_{qs}, \quad i_{qs} = i_{cq} + i_{lq} + i_q - C\omega_e v_{ds} \quad (2-12)$$

$$\frac{d}{dt} v_{ds} = \left(\frac{1}{C} i_{ds} - \frac{1}{C} i_{ld} - \frac{1}{C} i_d \right) + \omega_e v_{qs}, \quad \frac{d}{dt} v_{qs} = \left(\frac{1}{C} i_{qs} - \frac{1}{C} i_{lq} - \frac{1}{C} i_q \right) - \omega_e v_{ds} \quad (2-13)$$

Equations (2-14) shows the load voltages and currents in reference frame, assume of inductive load.

$$v_{ld} = R_L i_{ld} + L_l \frac{d}{dt} i_{ld} - \omega_e i_{lq}, \quad v_{lq} = R_L i_{lq} + L_l \frac{d}{dt} i_{lq} + \omega_e i_{ld} \quad (2-14)$$

2.5 Energy Storage System Model

The storage system consists of a bi-directional PWM voltage source based converter, grid connection inductors and batteries. In this thesis, the battery's characteristic is neglected and ideal voltage source E_s is adopted. Since battery power management is another complex topic, the research here focuses on the converter control to maintain a smooth power flow to the load. The differential equations can be established for the energy storage system.

$$v(t) = dE_s \quad (2-15)$$

$$\frac{di(t)}{dt} = \frac{1}{L} [v(t) - Ri(t) - v_s(t)] \quad (2-16)$$

In Eq. 2-15 and 2-16, d is the duty ratio, $v(t) = [v_a(t) \ v_b(t) \ v_c(t)]^T$ is converter output phase voltage. $v_s(t) = [v_{as}(t) \ v_{bs}(t) \ v_{cs}(t)]^T$ is terminal (or bus) voltage and $i(t) = [i_a(t) \ i_b(t) \ i_c(t)]^T$ is converter side current.

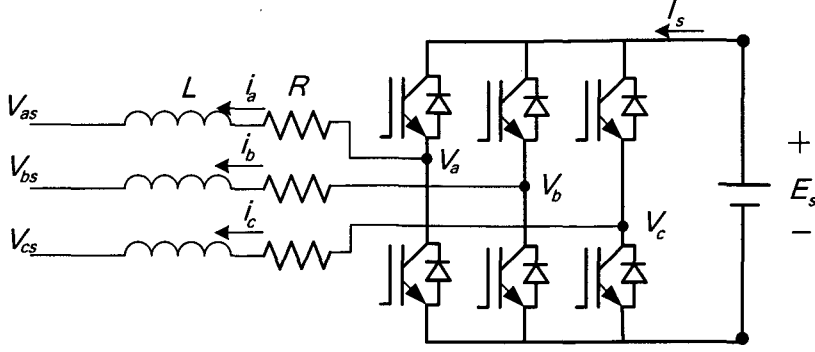


Figure 2-4: Energy storage system circuit

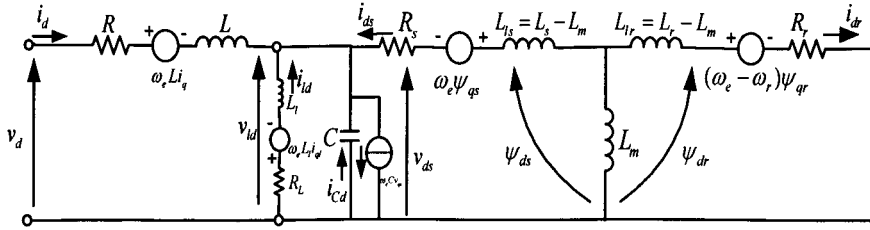
The above equations can be transferred to selected dq reference frame by applying park-clark transformation. Assume that the three-phase system is balanced three-phase three-wire system, in which zero components can be removed. The two differential equations can be written in Eq. (2-17) and (2-18), where ω is the rotating speed of reference frame.

$$\frac{di_d(t)}{dt} = \frac{1}{L} [v_d(t) - Ri_d(t) - v_{ds}(t)] + \omega_e i_q(t) \quad (2-17)$$

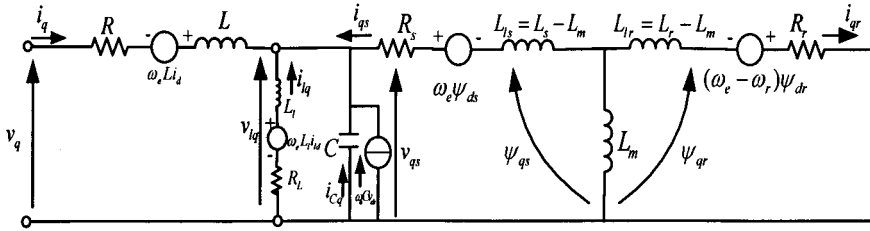
$$\frac{di_q(t)}{dt} = \frac{1}{L} [v_q(t) - Ri_q(t) - v_{qs}(t)] - \omega_e i_d(t) \quad (2-18)$$

2.6 Standalone WECS Model

The completed dynamic model can be presented based on the previous analysis. According to the set of 10 differential equations corresponding to variables $i_{ds}, i_{qs}, i_{dr}, i_{qr}, v_{ld}, v_{lq}, i_{ld}, i_{lq}, i_d, i_q$, the d-q equivalent circuit at synchronously rotating frame can be obtained, as shown in figure 2-5.



(a) d-axis equivalent circuit



(b) q-axis equivalent circuit

Figure 2-5: d-q Equivalent circuits in synchronous reference frame

2.7 Small Signal Model

Eq. (2-5—2-18) present the whole system mathematical model. Base on that, the whole system small signal models can be obtained by introducing the small variation.

To construct a small-signal ac model at a quiescent operation point, assume that system currents and voltages are equal to some given steady state values I and V (quiescent operating point), plus superimposed small ac variations \hat{i} and \hat{v} . Hence, we have

$$v_d = V_d + \hat{v}_d; \quad v_{ds} = V_{ds} + \hat{v}_{ds}; \quad v_{ld} = V_{ld} + \hat{v}_{ld}; \quad i_d = I_d + \hat{i}_d; \quad i_{ds} = I_{ds} + \hat{i}_{ds}; \quad i_{ld} = I_{ld} + \hat{i}_{ld} \quad (2-19)$$

$$v_q = V_q + \hat{v}_q; v_{qs} = V_{qs} + \hat{v}_{qs}; v_{lq} = V_{lq} + \hat{v}_{lq}; i_q = I_q + \hat{i}_q; i_{qs} = I_{qs} + \hat{i}_{qs}; i_{lq} = I_{lq} + \hat{i}_{lq} \quad (2-20)$$

$$\omega_e = \Omega_e + \hat{\omega}_e; \omega_r = \Omega_r + \hat{\omega}_r \quad (2-21)$$

With the assumptions that the ac variations are small in magnitude compared to the steady values, then the equations can be simplified. This is done by substituting Eqs. (2-20) and (2-21) into Eqs. (2-5) and (2-6).

$$V_{ds} + \hat{v}_{ds} = -R_s(I_{ds} + \hat{i}_{ds}) + \frac{d(\psi_{ds} + \hat{\psi}_{ds})}{dt} - (\Omega_e + \hat{\omega}_e)(\psi_{qs} + \hat{\psi}_{qs}) \quad (2-23)$$

$$V_{qs} + \hat{v}_{qs} = -R_s(I_{qs} + \hat{i}_{qs}) + \frac{d(\psi_{qs} + \hat{\psi}_{qs})}{dt} + (\Omega_e + \hat{\omega}_e)(\psi_{ds} + \hat{\psi}_{ds}) \quad (2-24)$$

By cancelling the steady-state terms and neglecting high order ac terms in Eqs. (2-23) and (2-24), the first-order ac terms on both side of the equations are left. Hence,

$$\hat{v}_{ds} = -R_s \hat{i}_{ds} + \frac{d\hat{\psi}_{ds}}{dt} - \Omega_e \hat{\psi}_{qs} - \hat{\omega}_e \psi_{qs} \quad (2-25)$$

$$\hat{v}_{qs} = -R_s \hat{i}_{qs} + \frac{d\hat{\psi}_{qs}}{dt} + \Omega_e \hat{\psi}_{ds} + \hat{\omega}_e \psi_{ds} \quad (2-26)$$

Where:

$$\hat{\psi}_{qs} = -L_{ls} \hat{i}_{qs} - L_m (\hat{i}_{qr} + \hat{i}_{qs})$$

$$\hat{\psi}_{ds} = -L_{ls} \hat{i}_{ds} - L_m (\hat{i}_{dr} + \hat{i}_{ds})$$

Similarly, eqs. (2-25) – (2-26) can be derived.

$$\hat{v}_{dr} = -R_r \hat{i}_{dr} + \frac{d\hat{\psi}_{dr}}{dt} - (\Omega_e - \Omega_r) \hat{\psi}_{qr} - (\hat{\omega}_e - \hat{\omega}_r) \psi_{qr} \quad (2-27)$$

$$\hat{v}_{qr} = -R_r \hat{i}_{qr} + \frac{d\hat{\psi}_{qr}}{dt} + (\Omega_e - \Omega_r) \hat{\psi}_{dr} + (\hat{\omega}_e - \hat{\omega}_r) \psi_{dr} \quad (2-28)$$

Where:

$$\hat{\psi}_{dr} = -L_{lr} \hat{i}_{dr} - L_m (\hat{i}_{dr} + \hat{i}_{ds})$$

$$\hat{\psi}_{qr} = -L_{lr} \hat{i}_{qr} - L_m (\hat{i}_{qr} + \hat{i}_{qs})$$

And similar procedures can be done on Eqs (2-130), (2-14), (2-16) to (2-18).

$$\frac{d}{dt} \hat{v}_{ds} = \frac{1}{C} \hat{i}_{ds} - \frac{1}{C} \hat{i}_{ld} - \frac{1}{C} \hat{i}_d + \Omega_e \hat{v}_{qs} + \hat{\omega}_e V_{qs} \quad (2-29)$$

$$\frac{d}{dt} \hat{v}_{qs} = \frac{1}{C} \hat{i}_{qs} - \frac{1}{C} \hat{i}_{lq} - \frac{1}{C} \hat{i}_q - \Omega_e \hat{v}_{ds} - \hat{\omega}_e V_{ds} \quad (2-30)$$

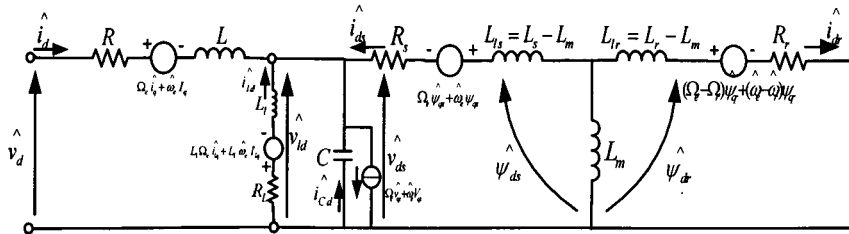
$$\hat{v}_{ld} = R \hat{i}_{ld} + L_l \frac{d}{dt} \hat{i}_{ld} - L_l \Omega_e \hat{i}_{lq} - L_l \hat{\omega}_e I_{lq} \quad (2-31)$$

$$\hat{v}_{lq} = R \hat{i}_{lq} + L_l \frac{d}{dt} \hat{i}_{lq} + L_l \Omega_e \hat{i}_{ld} + L_l \hat{\omega}_e I_{ld} \quad (2-32)$$

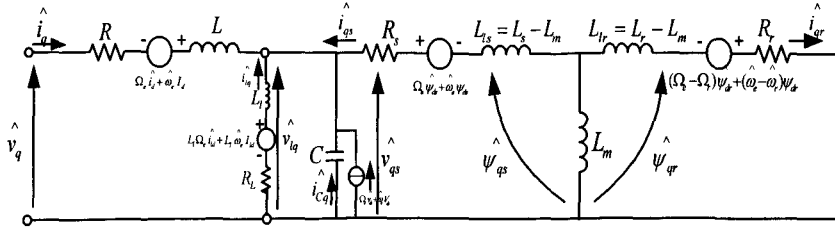
$$\frac{d \hat{i}_d}{dt} = \frac{1}{L} [\hat{v}_d - R \hat{i}_d - \hat{v}_{ds}] + \Omega_e \hat{i}_q + \hat{\omega}_e I_q \quad (2-33)$$

$$\frac{d \hat{i}_q}{dt} = \frac{1}{L} [\hat{v}_q - R \hat{i}_q - \hat{v}_{qs}] - \Omega_e \hat{i}_d - \hat{\omega}_e I_d \quad (2-34)$$

The small signal equivalent circuit for the whole system can be given based on Eqs. (2-25) – (2-34), as shown in Fig. 2-6.



(a) d-axis equivalent circuit



(b) *q*-axis equivalent circuit

Figure 2-6: Small signal model for d-q equivalent circuits in synchronous reference frame

2.8 Conclusions

In this chapter, the models of wind energy conversion system with induction generator and energy storage system are presented based on the model for each component. System model is developed to describe the dynamic behavior. Small signal model is derived based on the quiescent operating point and linearization. The model will be used for the controller design in Chapter 3.

3.2 Direct Voltage Control

3.2.1 Voltage and Frequency Control

In Figure 3-1, the displacement angle for the synchronous reference frame is generated by a given frequency, which is the reference frequency in the standalone WECS. In a standalone system, no synchronization is needed since the induction generator is indirectly controlled by the power converter through adjusting the bus voltage. In the diagram, transformation block receives the synchronous angle and transform the three phase local bus voltage into d-q reference frame. Two lead-lag regulators are employed to directly regulate the d and q component voltages. The target of the control system is to force the bus voltage follow the reference voltage through a fixed frequency transformation. The frequency is indirectly controlled through the transformation block by the given frequency. Reference v_d^* and v_q^* are the desired bus voltage and the fixed frequency is the desired system operating frequency. If the system outputs follow the references, then active and reactive power balances are maintained.

The system can be decoupled on a d-q reference frame. The plant transfer functions can be derived from the small signal model in the Chapter2. Figure 3-2 shows the block diagram using transfer functions for d or q axis, while the cross-coupling terms in the system in the system are treated as the disturbances. In Figure 3-2, the PWM converter is treated as a power amplifier with the gain K_{pwm} and $G(s)$ is the transfer function from $v_{d(q)}$ to $v_{d(q)s}$.

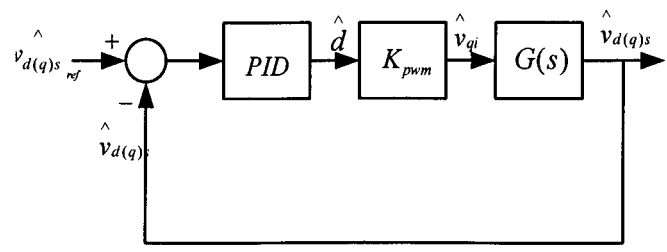


Figure 3-2: Block diagram using transfer functions

The controller for each axis uses a single loop to directly regulate the voltage. In this case, compared with conventional power control with cascaded loops, it has a much faster dynamic performance with even higher system bandwidth. Two lead-lag regulators are employed to regulate the d and q component voltages to obtain a higher phase margin and highest control bandwidth. The reference voltages can be arbitrary selected as long as the magnitude of the voltage is desired voltage. Usually the d-component reference voltage v_d^* is selected as the required local bus voltage, while the q-component reference voltage v_q^* is set to be zero. Space vector modulation (SVM) method is adapted to generate desired PWM gating signals.

It should be noted that the proposed system has no capability to protect the energy storage system (EES) from overload since the output current is not sensed in the control system. In order to protect the device from overload, several protection schemes have been developed. In this paper, dump load and current limit are employed together as the protection scheme. When the EES side current exceeds the maximum operating current of switching devices, the proper gating signal will be turned off to prevent the increase of current. The input power to EES is also detected, and the dump load is switched on when the input power exceeds the maximum power capability of the converter (or an energy storage device is applicable). The protection scheme with combination of fast act current limit and slow act dump load (due to power detection) functions well during overload. Other protections regarding over voltage, over temperature and devices short through are similar to regular converter system.

3.2.2 Direct Voltage Controller Design

Shown in Figure 3-3, the system transfer function $G(s)$ can be obtained from the small signal model. The cross-coupling terms in the system are neglected since they are treated as the disturbances and compensated by either feed-forward or regulators.

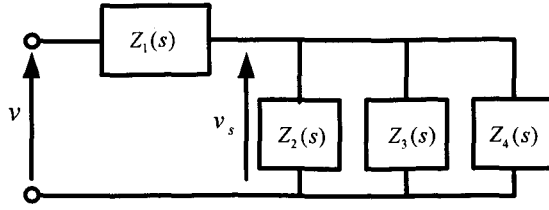


Figure 3-3: System blocks model in S- domain

The local bus voltage is the system output. The converter voltage, which controlled by the PWM signals is the system input. Then converter voltage to local bus voltage transfer function $G(s)$ is shown in Eq. (3-1).

$$G(s) = \frac{v_s(s)}{v(s)} = \frac{Z_2(s)Z_3(s)Z_4(s)}{Z_2(s)Z_3(s)Z_4(s) + Z_1(s)[Z_3(s)Z_4(s) + Z_2(s)Z_3(s) + Z_2(s)Z_4(s)]} \quad (3-1)$$

Where $Z_1(s)$ is the transfer function of converter side filter. $Z_2(s)$ and $Z_3(s)$ are the transfer functions of load and self-excitation capacitors. In the analysis, resistive load is assumed since most of the consumer loads in the home are resistance loads, such as lighting, heating devices and so on.

$$Z_1(s) = sL \quad (3-2)$$

$$Z_2(s) = R_L \quad (3-3)$$

$$Z_3(s) = \frac{1}{sC} \quad (3-4)$$

$Z_4(s)$ represents the induction generator in S domain. The magnetization inductance is much bigger than the stator and rotor leakage inductances. So L_m can be neglected and the transfer function can be simplified in Eq.(3-5).

$$Z_4(s) = (R_s + R_r) + s(L_s + L_r) \quad (3-5)$$

Substitute the Eqs. (3-2) – (3-5) into Eq. (3-1), we have

$$G(s) = \frac{R_L[(R_s + R_r) + s(L_s + L_r)]}{(R_L + sL + s^2 R_L LC_e)[(R_s + R_r) + s(L_s + L_r)] + sLR_L} \quad (3-6)$$

The bode plots of the uncompensated loop gain $Tu(s)$ is shown in Figure 3-3, where

$$T_u(s) = K_{pwm} G(s) \quad (3-7)$$

The parameters are selected based on the 2kW lab prototype system which is shown in Table 3-1.

Table 3-1: Parameters of the Lab Prototype System

Load, R_L	22 Ω
Converter filter inductance, L	10mH/20A
Excitation Capacitance, C	141 μF
Stator Resistance, R_s	0.46 Ω
Rotor Resistance, R_r	0.7 Ω
Stator Leakage Inductance, L_s	3.5mH
Rotor Leakage Inductance, L_r	3.5mH
Converter Gain, K_{pwm}	245

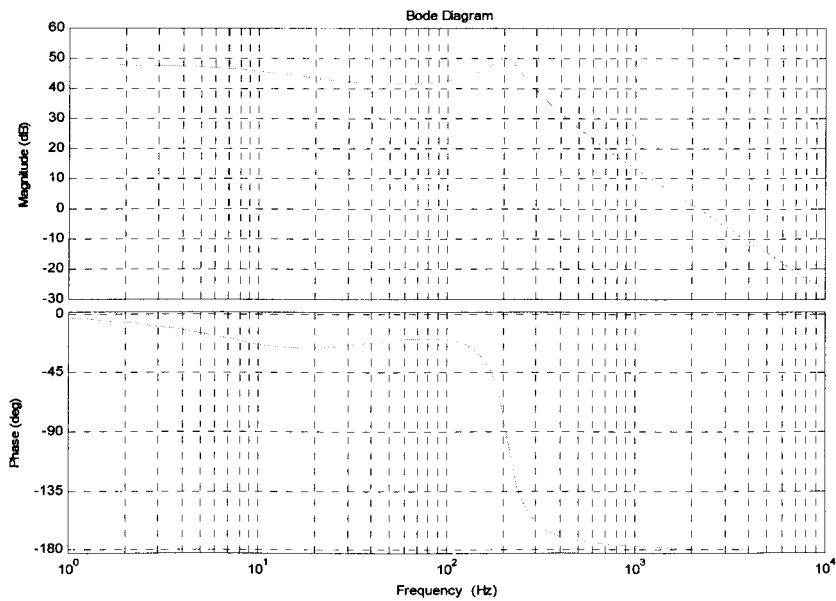


Figure 3-4: Bode plots of uncompensated loop

To analysis the load effect on the Bode plots, Fig. 3-5 shows the plots of the uncompensated loop under different loads. The load changes from 10 ohm to 100ohm in four steps. It can be seen that different loads only affect the overshoot value in magnitude plot and the slope of the curve at in phase plot (at resonant frequency). If the control bandwidth is selected to be much higher than the resonant frequency, the load variation has little effect to the dynamic response. In the prototype system, the bandwidth is selected at 1kHz while the switching frequency of the converter is 10kHz.

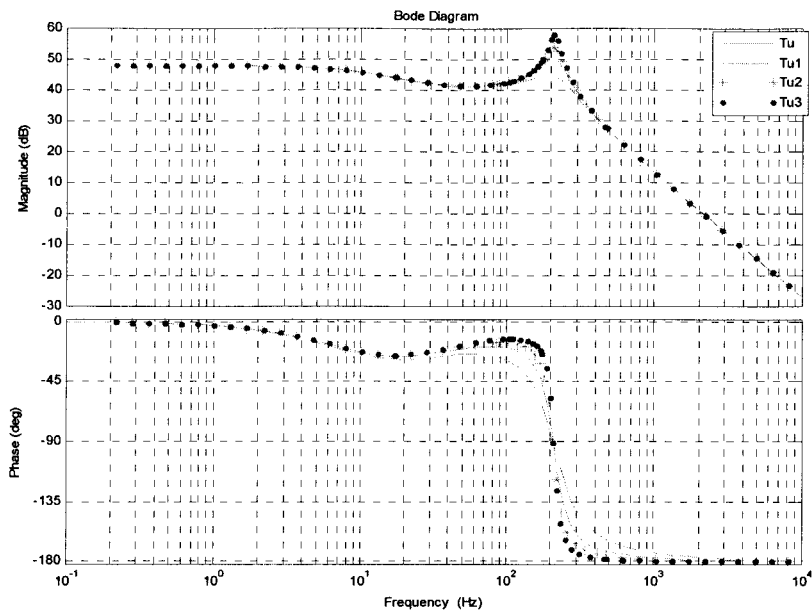


Figure 3-5: Bode plots of uncompensated loop with different loads

For the uncompensated loop, the cross-over frequency is around 2kHz, and the phase margin is no more than 5 degrees. For a stable system, the best phase margin should be around 52 degrees. In order to gain a higher bandwidth and enough phase margin, lead-lag regulators are used in equation 3-8. Thus the dynamic response of the control system will be faster than the cascaded system using PI regulators. In the prototype system the switching frequency is 10kHz. So the cross-over frequency of the compensated loop is selected to be around 1kHz ($1/10^{\text{th}}$ of the switching frequency). By proper selecting the regulator parameters, the compensated loop has

around 52 degrees phase margin at 1kHz. The bode plots of compensated loop gain $T(s)$ (Eq.3-9) are shown in Figure 3-6.

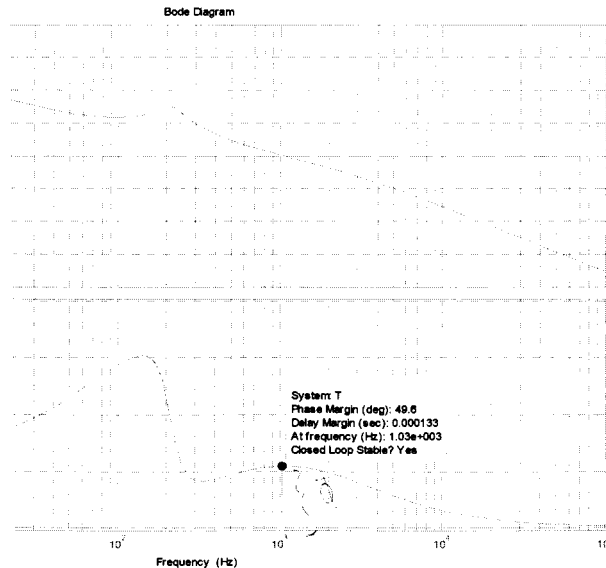
$$G_c(s) = G_{c0} \frac{1+T_z s}{1+T_p s} \frac{1+T_i s}{T_i s} \quad (3-8)$$

$$T(s) = G_c(s) K_{pwm} G(s) \quad (3-9)$$

The parameters value of the lead-lag controller is shown in table 3-2, which are calculated through traditional method.

Table 3-2: Parameters of lead-lag Controller

T_z	T_p	T_i
$\frac{1}{2\pi * 340}$	$\frac{1}{2\pi * 2900}$	$\frac{1}{2\pi * 100}$



5: Bode plot of compensated loop gain

3.3 Conclusions

In this chapter, a novel direct voltage control is proposed. Based on this analysis, the system local bus voltage and frequency can be regulated by the two control loops for d-q reference frame. But due to the non-sensed current, current protection scheme is employed in this control system. Based on the developed system transfer function, the direct voltage controller can be designed based on the traditional lead-lag regulator with increased the system phase margin and extended the system bandwidth. Thus reliability and dynamic performance are improved.

Chapter 4 Simulation and Experimental Results

4.1 Introduction

Both simulation models and experimental setup were built for the verification of the proposed wind energy conversion system with induction generator and energy storage. The whole system model was created using Simulink/Matlab. In the simulation, the system consists of the three major components: the wind power generation system with energy storage devices, the controller for power converter and the load. The wind turbine drives the induction generator to run at the based speed. In order to simulate the wind fluctuation, the wind speed is assumed to have the random changing around the base speed time by time. And the controller is used to regulate the local bus voltage and frequency constant. The details will be presented in the following sections.

As shown in Figure 4-1, the experimental set-up consists of a back to back converter, isolated transformer, a DC motor, an induction generator and a DSP-FPGA based digital controller. The proposed system is based on the 2kW LabVolt wound-rotor induction machine set.

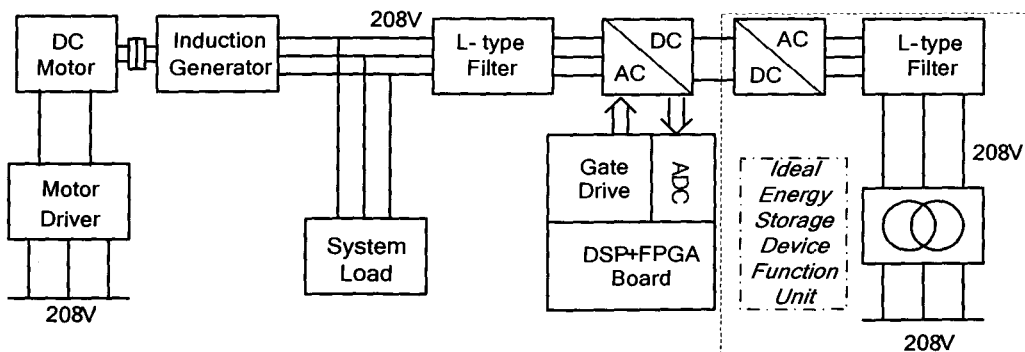


Figure 4-1: Block diagram of experimental set-up

The multipurpose back to back power converter was designed and built for several projects. In this project, one side of the converter works as a PWM rectifier, which maintains the DC link

voltage constant. This part is functional as an ideal energy storage device. And the other side works as an inverter to maintain the local bus voltage and frequency constant with direct voltage control. The DC motor and induction generator are mechanically coupled together. The DC motor is operated as a prime mover to drive the induction generator in the super-synchronous speed mode. The Texas instruments TMS320F2812 DSP and Altera’s Cyclone FPGA were employed as the core of the digital controller. Some experimental variables, such as dq components voltage and current, are captured from DSP internal memory since these variables are not directly available on the oscilloscopes.

4.2 Simulation Model

The whole system consists of the three major components: the wind power generator with energy storage, the controller for power converter and the load, as shown in Fig. 4-2.

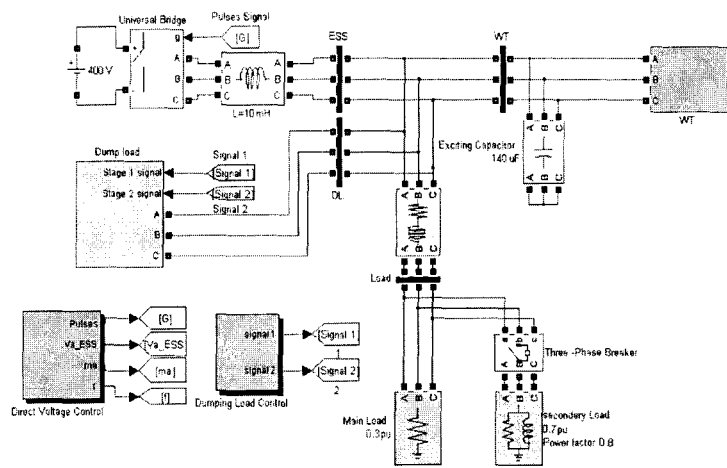


Figure 4-2: System models of proposed WECS

The system parameters of the prototypes system is shown in Table 4-1. Same paramters are used in the simulation.

Table 4-1: System parameters of prototypes system

System Ratings				Generator Parameters	
Power	2kW	Current	5.55A	Stator, Rotor Resistances	0.032pu
Voltage	208V	Frequency	60Hz	Stator, Rotor Leakage Inductances	0.061pu
Excitation Capacitor		141 μ F		Magnetization Inductance	0.875pu
Converter Inductance		0.2pu		Pole Pairs	2

Figure 4-3 shows the implementation of the controller outlined in Chapter 3. The voltage references and frequency reference are given as constant values. The local bus voltages in pu system are sensed and feedback to the controller. The lead-lag regulators are used to regulate the bus voltage. The output of the regulator is used to generate the reference voltage for power converter, from where PWM gating signals are generated using SVM.

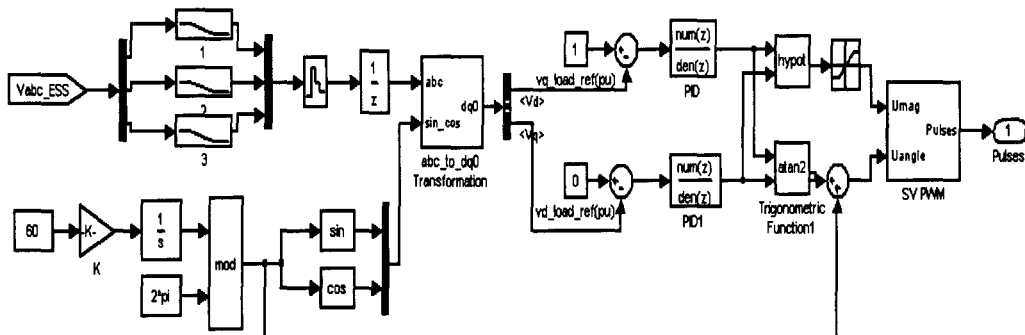


Figure 4-3: Direct voltage controller model

The energy storage devices (here is the ideal voltage source) are connected to the local bus through the inverter. Through the proposed controller, the local bus voltage and frequency should be kept constant in steady-state, and meanwhile, the system power flow should maintain balanced.

4.3 Simulation Results

4.3.1 Non-ESS Stand Alone Wind Power Generation System

When the WECS is connected to the load without the energy storage system, the local bus voltage and frequency will have a large fluctuation due to the wind speed fluctuation. The system configuration in simulation is shown in Figure 4-4. The wind turbine model use the standard block provided with the Simulink's Simpowersystem block set. The based wind speed is set to be 10m/s. A random source, whose range is from -1 to +1, is used to simulate the fluctuation in the wind. As shown in Fig. 4-5, the voltage and frequency have large fluctuations produced by wind generator without ESS. Both are not constant. So the power quality is not acceptable.

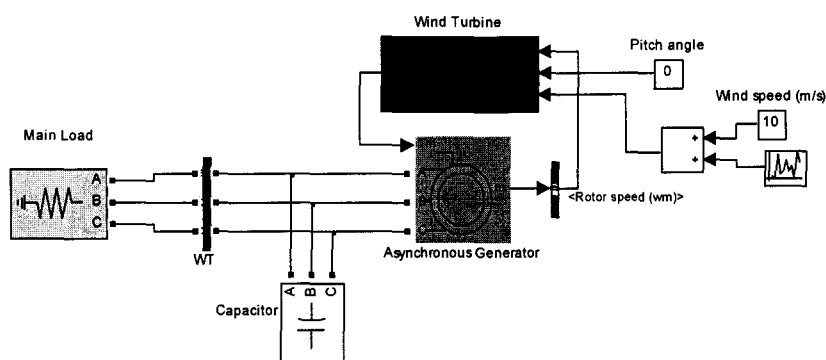


Figure 4-4: Wind power generation system without ESS

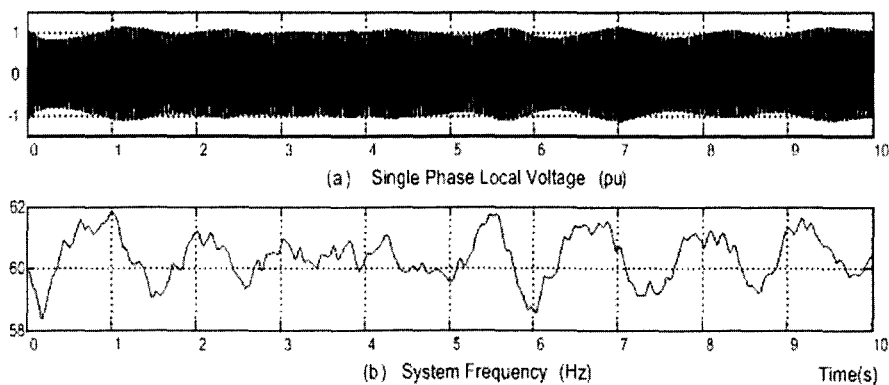


Figure 4-5: System voltage and frequency without ESS

4.3.2 Simulation Results with ESS

A. Resistive load Step Change

In order to test the controller performance both in steady state and transient response, a step load change was held at 1.0s in the simulation. The results are shown in Figure 4-6—4-13. With the EES, the voltage and frequency keep constant whenever before or after the load change. And meanwhile, the system power flow is maintained balanced.

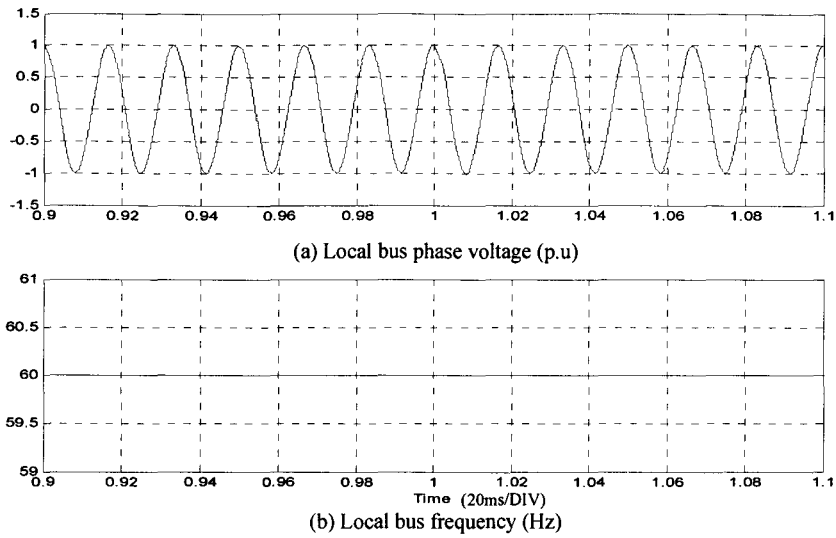


Figure 4-6: Local bus voltage and frequency during load step change (with R Load) in Simulation

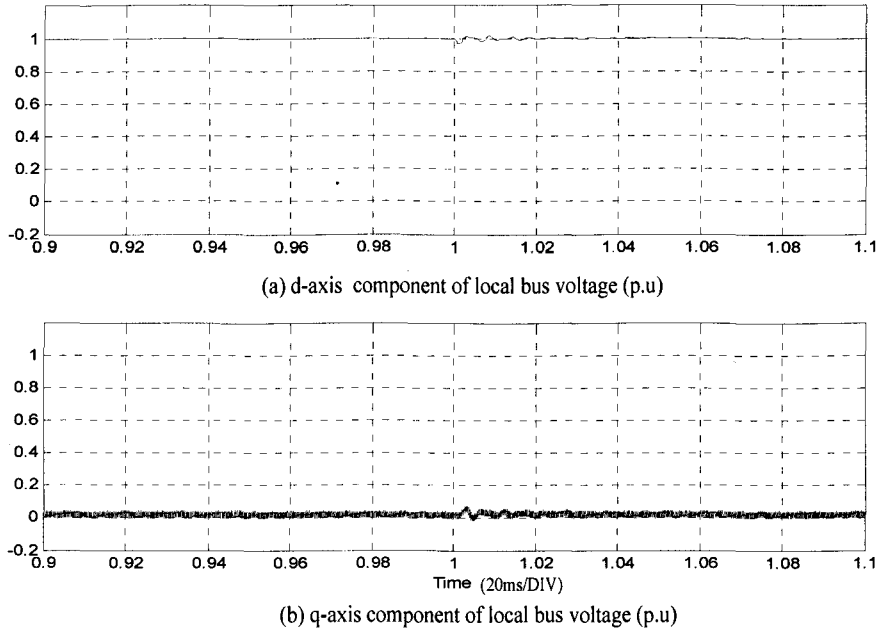


Figure 4-7: d-q axis inverter voltage during load step change (with R Load) in Simulation

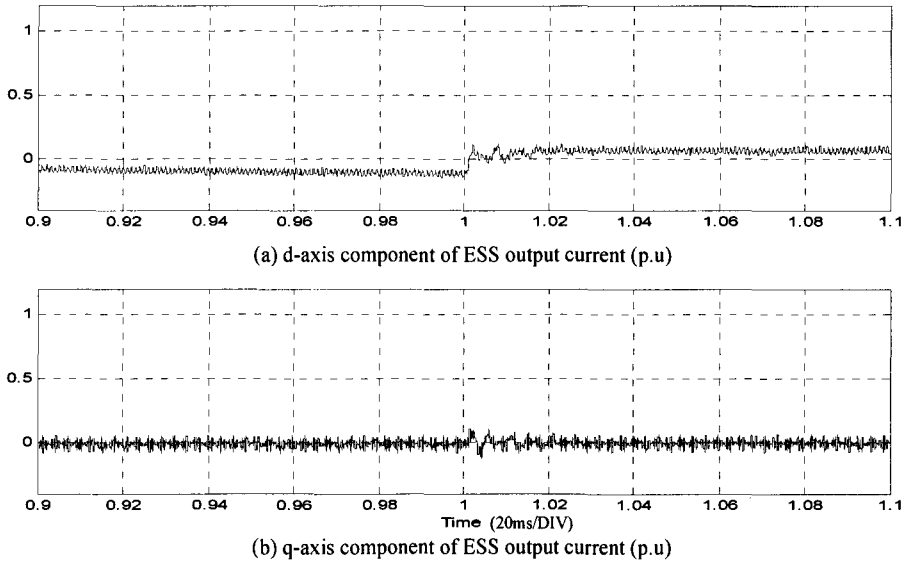


Figure 4-8: d-q axis inverter current during load step change (with R Load) in Simulation

In Fig.4-5, the voltage and frequency have large fluctuations by wind generator without ESS. When the EES is employed with proposed control system, the voltage and frequency are maintained constant as shown in Fig.4-6. Figure 4-7 and Figure 4-8 show the waveforms of d,q axis voltages and currents. During the load change, the voltages track the reference voltages in

very well with a small transient. It also can be observed from the waveforms that the frequency transient is less than $\pm 0.1\text{Hz}$ and voltage dip is less than 1%. There is no steady-state error in the voltage and frequency due to the merits of synchronous frame and lead-lag regulators.

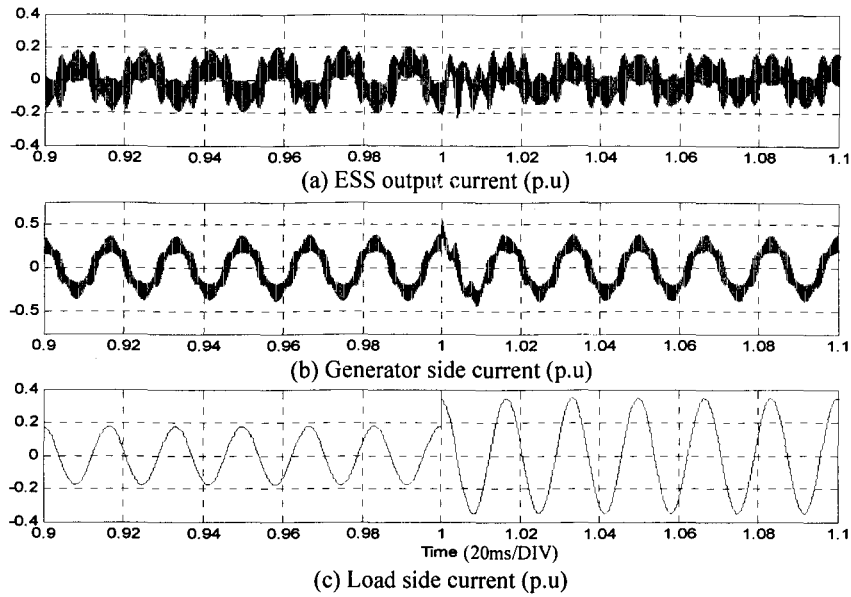


Figure 4-9: Current waveforms during load step change (with R load) in simulation

Fig.4-9 shows the current waveforms from the generator, ESS and load. The ESS absorbs the real power before 1.0s and provides real power after 1.0s. EES also compensates the fluctuation in the real power due to wind fluctuation at the same time. In the whole process, the power flow in the system maintained balance all the time.

B. Inductive Load Step Change

Fig.4-10 to Fig.4-13 show the simulation results with RL-load change at 1.0s. Fig.4-10 shows the local bus voltage and frequency. It has a similar response with the R load change. There's no significant change both in voltage and frequency. Fig. 4-11 shows the current waveforms with RL-load in simulation. The generator and EES together provide the real power. The reactive power is compensated from ESS. Fig. 4-12 and Fig.4-13 shows the d-q axis voltage and current simulation waveforms during load change. The d/q component voltages have very small dip. The d/q current components reflected the real power and reactive power from the EES respectively.

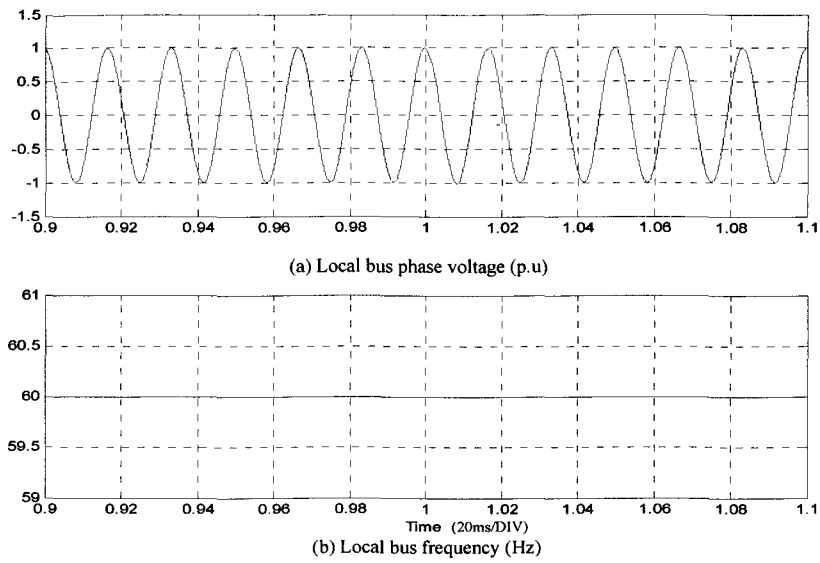


Figure 4-10: Voltage and frequency during load step change (with RL load) in simulation

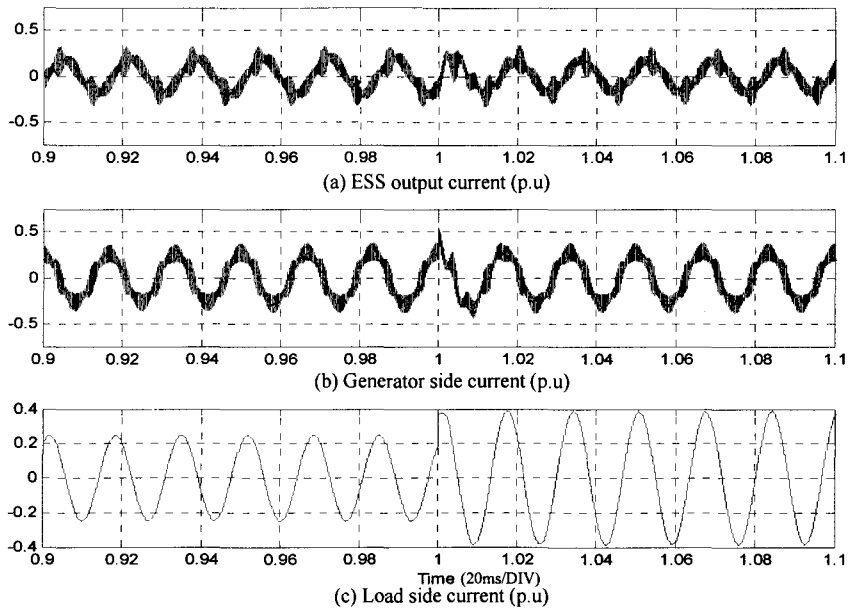


Figure 4-11: Currents waveforms during load step change (with RL load) in simulation

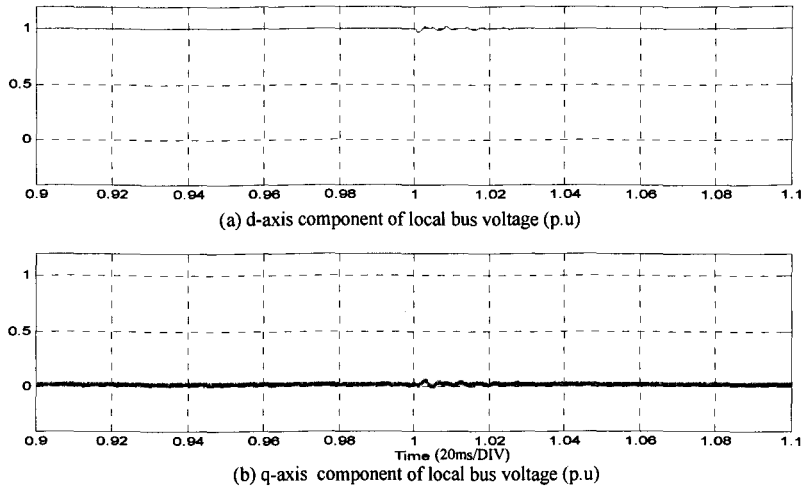


Figure 4-12: d-q axis inverter voltages during load step change (with RL Load) in Simulation

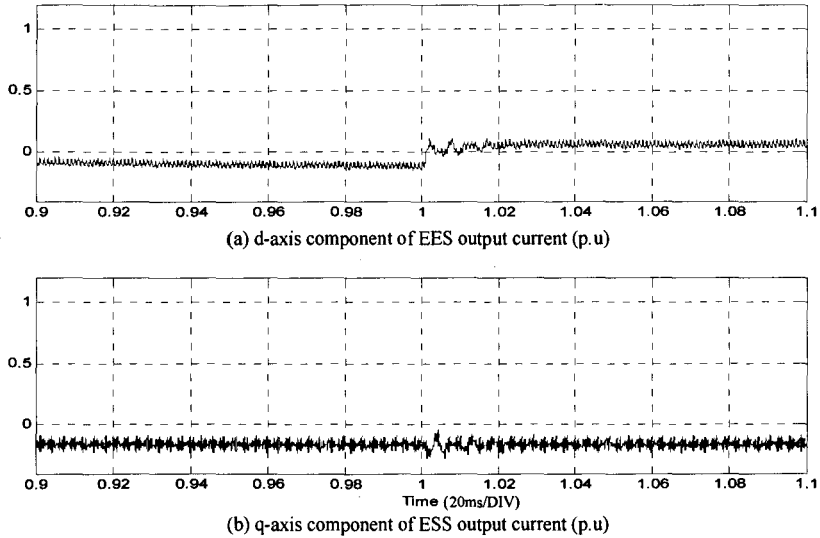


Figure 4-13: d-q axis inverter currents during load step change (with RL Load) in Simulation

Simulation results clearly show that the direct voltage control has good steady-state performance and fast transient response. The experimental results will be presented in the following section. Companied with the experimental results, the controller performance can be better tested.

4.4 Experimental Setup

The whole system experimental setup is shown in Fig.4-14, where part “A” is the DC motor, and part “B” is the induction generator. The DC motor and induction generator are mechanically coupled together. The DC motor is operated as a prime mover to drive the induction generator in the super-synchronous speed mode. Part “C” is the self-excitation capacitor. The output of the induction generator is connected to the local bus, and only provides the real power to the load. The energy storage system is simulated by a back-to-back converter, which is part “D” in Figure 4-14. One side of the converter works as a PWM rectifier, which maintains the DC link voltage constant. This part is functional as an ideal energy storage device. The power fluctuation is buffed through the grid. And the other side works as an inverter to maintain the local bus voltage and frequency constant with direct voltage control, and meanwhile, keep the system power balanced. The DC link voltage is rated at 400V, and the Local bus voltage was rated at 208V/5.55A. Part “F” is the DSP-FPGA controller board. And part “E” is the filter inductor for PWM converter.

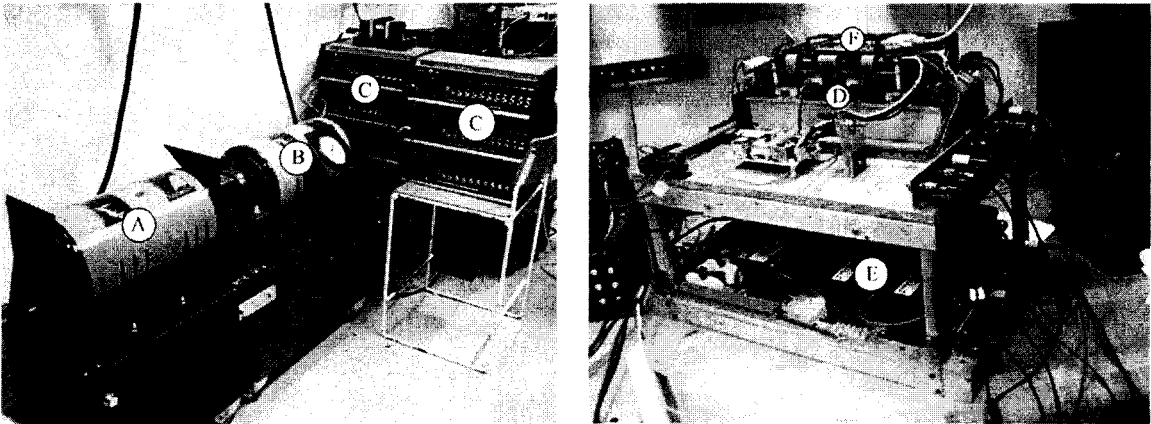


Figure 4-14: Experimental setup

(A: DC motor; B: Induction generator; C: Self-excited capacitor; D: Converter; E: Line inductors; F: DSP-FPGA controller board)

The controller design developed in Chapter3 was implemented using the DSP-FPGA platform as shown in Fig.4-15. The controller hardware used in this experiment consisted of the analogue signal conditioning board, DSP (“A”), FPGA, interface circuit (“B”) and a LED display (“C”).

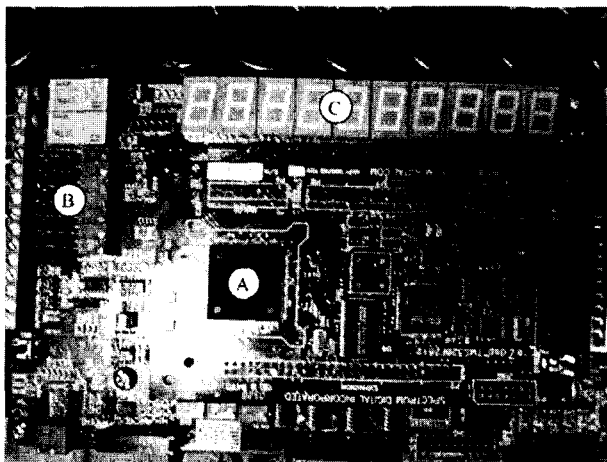


Figure 4-15: DSP controller board

(A: DSP chip; B: Interface circuit; C: LED display)

The controller algorithm from chapter3 was transferred into the DSP using C programming language. The complete code for the controller is given in Appendix D.

The system voltages and currents are sensed and fed to the conditioning circuit, which fed the signals to the A/D converter. The results of A/D converter are fetched by the DSP and used as the feedback signals. The final output signals of the DSP board are the switch signals for the PWM converter. The role of the FPGA in this experiment is to provide the phase information for the grid-side rectifier and to provide protection for the power converter in case of abnormal conditions. For this purposes the over-voltage, over-current and over-temperature protection circuits are interfaced directly with the FPGA to provide instantaneous protection of the power devices. In this system, the switching frequency is selected at 4.5 kHz. Higher switching frequency obviously would be better in the performance, but switching loss should be considered in the practical design.

The operational sequence of the controller is developed as following. First the grid side converter is energized. Once it is activated, the converter boosts the DC voltage to the required 400V. At this time the local bus side converter is turned on and works with constant V/f control strategy to drive the induction generator speeding up until it goes into steady state. The large starting current in the induction machine is avoided in the V/F control. And then, adjust the DC

motor input voltage to increase the input power until the rotating speed beyonds the synchronous speed. Now, the induction machine is operating in generator mode. The load then is connected to the local bus. And the controller starts to maintain the local bus voltage and frequency constant. The turn-off sequence for the converter is the opposite procedure of the start-up. In practical, the wind turbine speeds up the generator. The generator builds up the voltage with self-excitation capacitor. When the voltage reaches the desired level, the generator is brought to the local bus and ESS is started to maintain the local bus voltage and current. During the connection, synchronization is needed between ESS and generator. But in the experimental setup the synchronization is not necessary since the machine is started through the converter directly.

4.5 Experimental Results and Analysis

The performance of the direct voltage control was evaluated on the prototype system at several load conditions. The load step change is necessary. Through the load step change, the transient procedures are captured using the oscilloscopes through the voltage and current sensors. For different kind of loads, the local bus voltage and frequency are kept constant with direct voltage control. And the system power flow is maintained balanced with the ESS.

4.5.1 Resistive Load Test

Resistive load is connected on the local bus. When the generator starts up and goes into steady state, a three phase resistive load is connected to the system local bus. The resistor value is 120Ω . The steady state waveforms of the voltage and current at different sides are shown in Fig. 4-16.

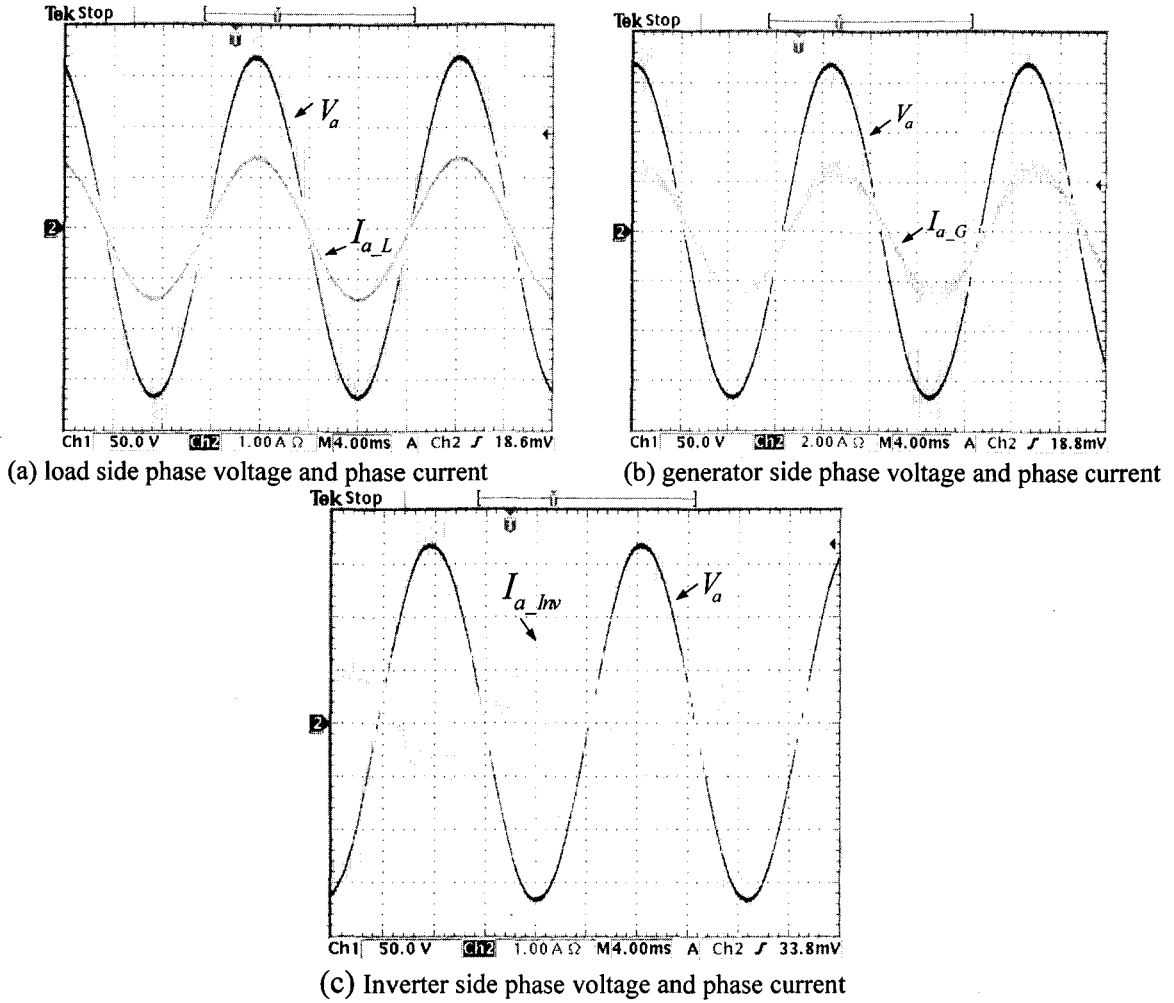


Figure 4-16: Steady-State waveforms with resistive load

Fig. 4-16(a) shows the waveforms of local bus voltage and the load current. As we can see, the local bus voltage is 120V in steady state. The load current is around 1A. Fig 4-16(b) and (c) shows the local bus voltage, generator side current and inverter side current. The generator provides the real power to load and EES. From Fig.4-16(b), the generator side current is around 1.7A. The load current is 1A. Fig.4-16(c) shows the additional 0.7A current entering the converter. From the current waveforms, it can be concluded that the system real power is balanced by ESS. Under same local bus voltage, the current reflects the system power flow.

When the wind power generation system runs in the steady state, another resistive load is switched on and off from the local bus. The transient waveforms of local bus voltage and currents were captured, as shown in Fig. 4-17.

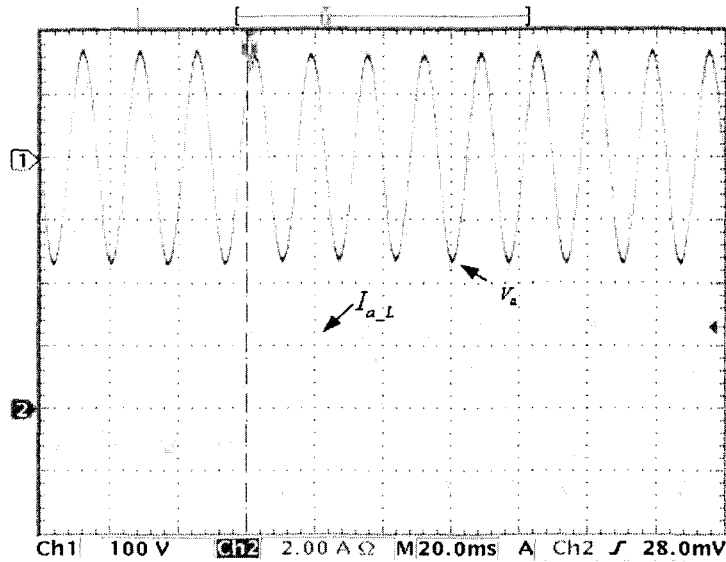
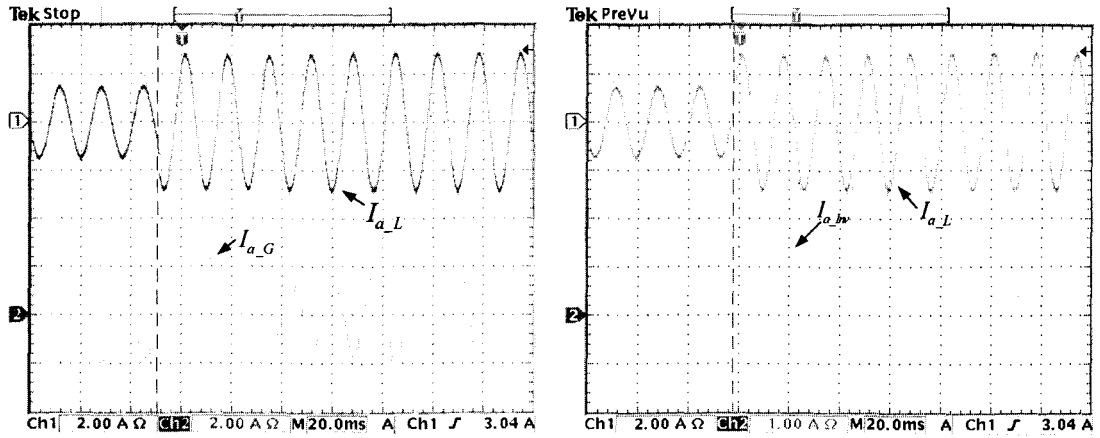


Figure 4-17: Transient waveforms of local bus voltage and load current during load step change

Fig. 4-17 shows the waveforms of local bus voltage and load current during the load change. After the additional resistor is switched on, the load current increases twice. From Fig.4-17, it can be seen that the local bus voltage has very small transient and thus a very small frequency change.

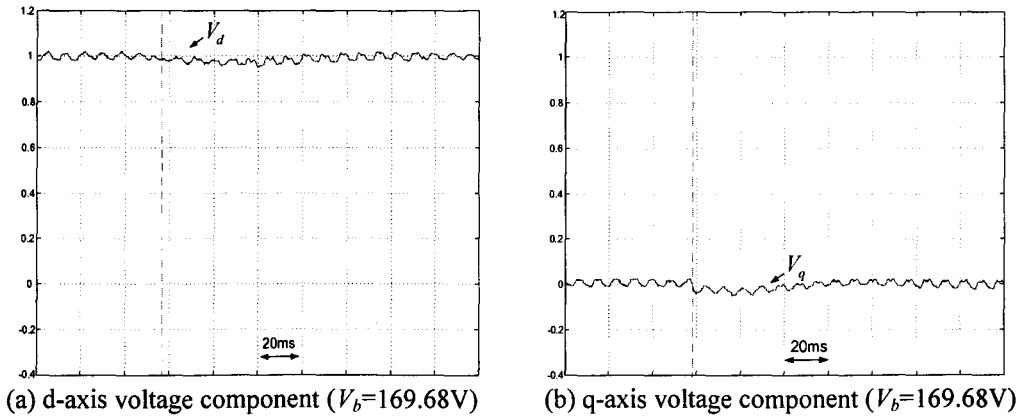
Fig 4-18 shows the waveforms of generator side current, inverter side current and load side current in the transient. It should be noted that the fluctuation in the generator current came from the oscillation of motor/generator set caused by load change. EES quickly compensates this oscillation and maintain the bus voltage/frequency constant. It also shows in Fig.4-18(b) that the ESS changes the power directional from absorbing excess real power to supplying the additional real power to the load.



(a) Load phase current and generator side phase current (b) Load phase current and inverter side phase current

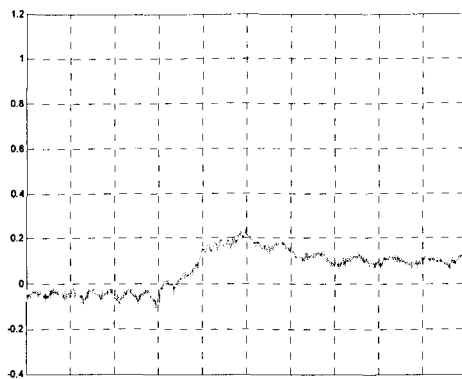
Figure 4-18: Transient waveforms of generator and converter currents during load step change (with R Load)

Voltage and current dq components waveforms of converter side are also captured from the DSP memory during the load change, as shown in Fig. 4-19. The reference voltage for v_d and v_q are 1.0 pu and 0 pu respectively. The voltage dq components trace the references very well. The current dq components are often referred to the real power and reactive power respectively. As it can be seen, i_d changes from negative to positive, indicating the power flow change for ESS. From the waveforms, it can be concluded that the control system has fast dynamic response during transient. And the steady-state error is almost zero because of the lead-lag regulators.

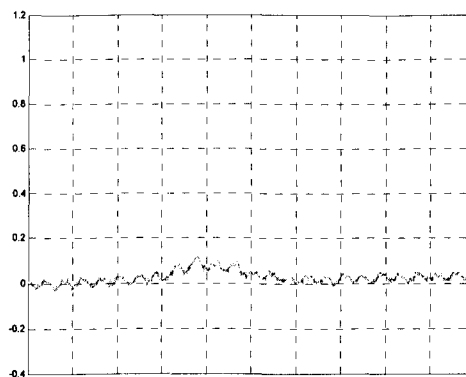


(a) d-axis voltage component ($V_b=169.68V$)

(b) q-axis voltage component ($V_b=169.68V$)



(a) d-axis current component ($I_d=7.85A$)

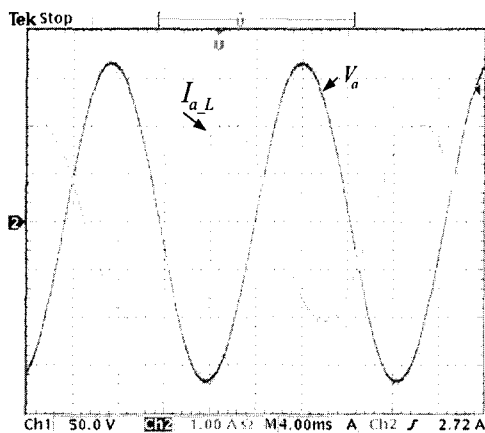


(b) q-axis current component ($I_q=7.85A$)

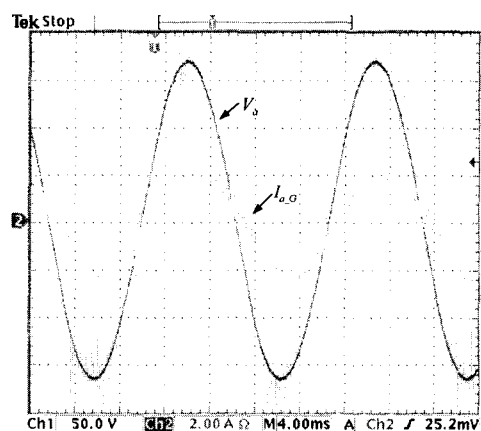
Figure 4-19: d-q axis Inverter side current during load step change (with R Load)

4.5.2 Inductive Load Test

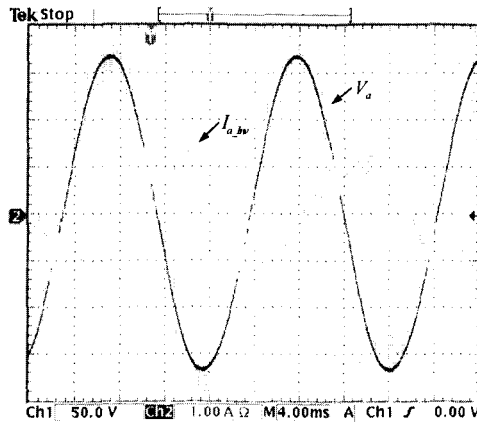
The system performance is also verified by connecting the inductive load. A three-phase 120Ω resistor bank and a $0.31H$ inductor bank are connected to the bus in parallel. With RL load, the controller still maintains the local bus voltage and frequency constant by providing the reactive power to the load. The steady state waveforms are shown in Fig. 4-20.



(a) load phase voltage and phase current



(b) generator phase voltage and phase current



(c) Inverter side phase voltage and phase current

Figure 4-20: Steady-State voltage and current waveforms with RL load

Fig.4-20(a) shows the waveforms of local bus voltage and load current. Because of the RL load, the load current and voltage have about 45 degree phase displacement. Fig.4-20(b) and (c) show the waveforms of local bus voltage, generator side current and inverter side current. The generator only provides the real power to the load. So the current and voltage are almost in phase. The inverter absorbs the extra real in the system and provides the reactive power to the load. So the inverter side current lags the voltage at about 45 degree. From the Fig.4-20, it reflects the system power flow always maintains balance with the EES in the steady state.

Load step change is also carried in the RL load condition. When the whole system runs into steady state, an additional load (120ohm) is switched on and off. The transient waveforms are shown in Fig. 4-21 and Fig.4-22.

The local bus voltage and load current transient waveforms are given in Fig.4-21. It is similar to the waveforms with pure resistive load, and the local bus voltage has a very small transient. Fig. 4-22 shows the transient waveforms of the load current, generator side current and inverter side current.

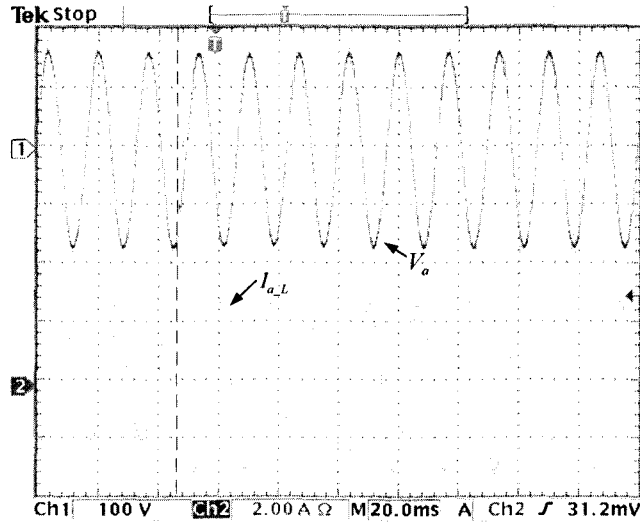
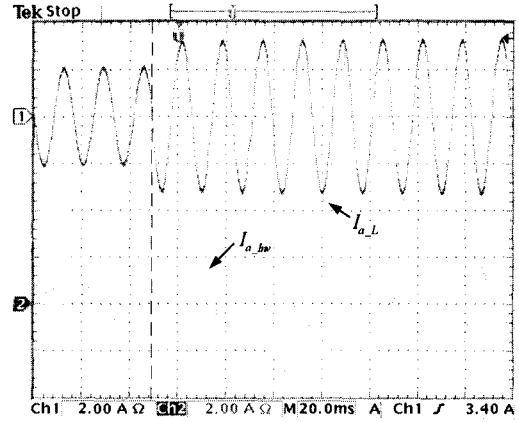
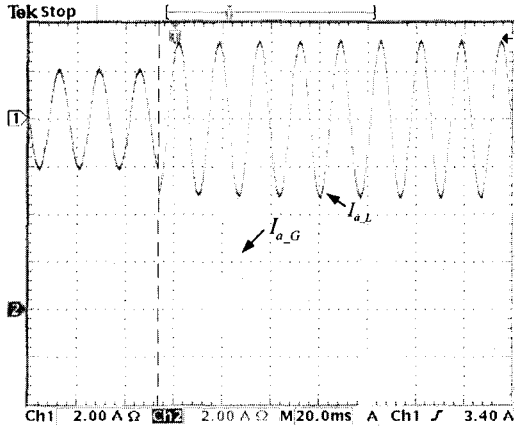


Figure 4-21: Transient waveforms of local bus voltage and load current during load step change (with RL load)



(a) Load phase current and generator side phase current (b) Load phase current and inverter side phase current

Figure 4-22: Transient waveforms of currents of generator and inverter during load step change (with RL load)

Inverter side voltage and current dq components waveforms are also given in Fig. 4-23. The voltage dq components are very similar to the pure resistive load. During the load change, v_d has a very small dip and then went back to the reference value quickly. Current waveforms are also similar to the resistive load condition. Since the load change is only on resistance, i_d has the same waveform as resistive load and i_q maintains at -0.2 pu.

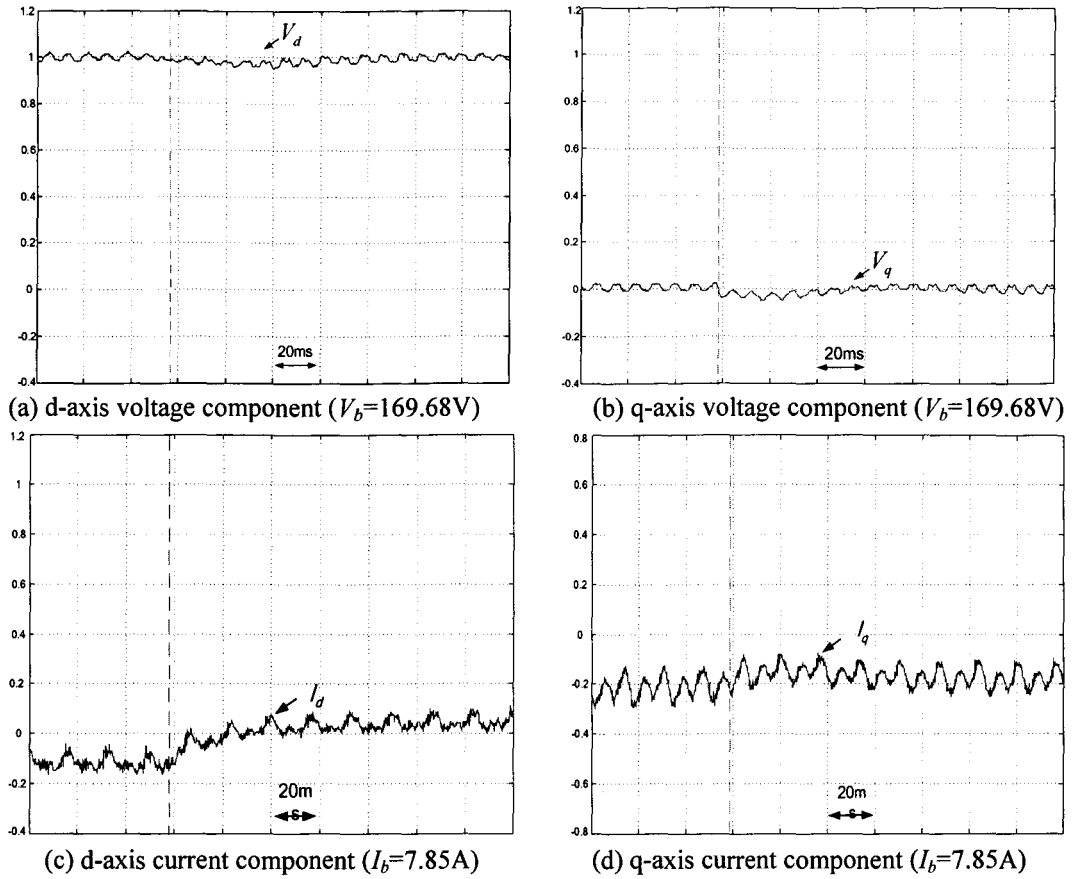


Figure 4-23: d-q axis Inverter side current and voltage during step load change (with RL Load)

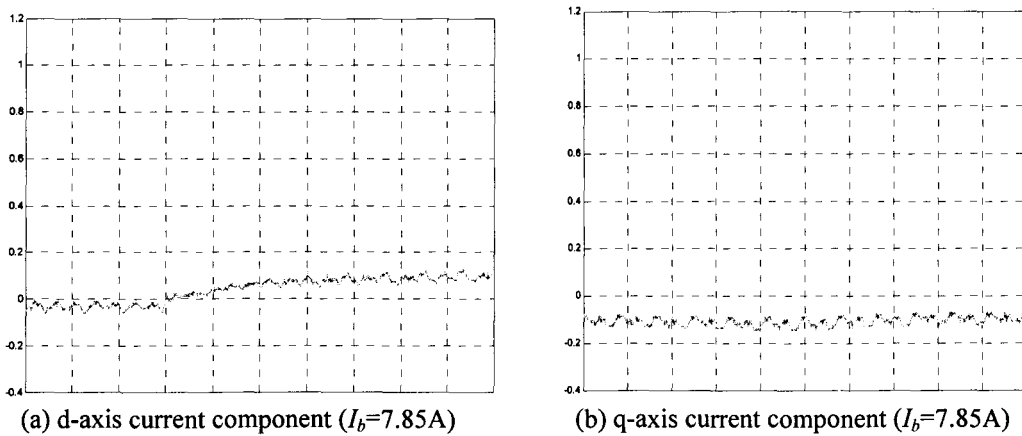


Figure 4-24: d-q axis Inverter current during step load change with compensation (with R-L Load)

It should be noted that current dq components waveforms contain an ac component at fundamental frequency both in R load and R-L load situations. It is verified that the inverter side current has a very small DC component. There are several reasons including non-symmetrical voltage output, sensor dc calibration and temperature drifting of the analog conditioning circuits. The last

two problems can be compensated through DSP using carefully calibrated circuit and dc bias compensation. But the unsymmetrical devices can not be avoided. The small DC voltage will cause large DC current in the system due to small resistance. This is the drawback of the proposed control system, which has no capability to directly regulate the current. So a simple DC compensation scheme is necessary in order to avoid the large DC current. The dq component current waveforms after compensation are shown in Figure 4-24.

4.6 Conclusions

The model for standalone WECS with induction generator and energy storage system is constructed in the Simlink/Matlab software. The whole system with direct voltage control is verified in simulation and experiment. Different load conditions and load changes are carried in this project to verify the effectiveness of the controller. From the simulation and experimental results, it can be seen that the directly voltage control can achieve desired steady state performance with fast dynamic response.

Chapter 5 Conclusions

5.1 Conclusions

From the simulation and experimental results obtained in this thesis, several conclusions can be drawn regarding the design of the directly voltage controller for the standalone wind energy conversion system with induction generator and energy storage system.

The dynamic mathematical model for the whole system is developed. Small signal model is derived based on the dynamic model. The small signal model provides the basis for the controller design. A lead-lag regulator based controller is developed to regulated the local bus voltage constant in synchronous reference frame. The reference frequency and voltages are selected at the desired values. The transformer functions for d-q axis are developed. The regulator parameters is calculated through the traditional design using phase margin and system bandwidth. lead-lag regulators successfully improve system bandwidth while maintain high phase margin. Thus fast dynamic response and small overshoot/undershoot can be expected. The system stability is also improved.

The control scheme developed for the direct control of the system local bus voltage and frequency in the WECS has been tested using the simulation model and experimental setup. The control system successfully maintains the system voltage and frequency constant under different load condition and power fluctuation. The power flow in the system maintains balanced by the power converter and ESS. The transient waveforms shown in the experimental results, corresponding with the simulation results, verify the expected response from controller design.

Overall, this research work has shown that the energy storage system with direct voltage control can be effectively used in the stand-alone WECS. Though this kind of control scheme has

its own drawbacks such as no capability of current protection, simple scheme can be used to overcome the problems of overload protection as well as the dc current. In this thesis, the employment of squirrel cage induction generator gives benefit for reducing the manufacturing and maintenance cost. The system reliability can also be improved compared to DFIG or permanent magnetic machines. Only fractional capacity power converter is required in this configuration, which further reduce the system cost.

5.2 Major Contributions

One of the main contributions of this work is the design of the direct voltage control for the stand-alone WECS with induction generator and ESS. The control scheme shown in Figure 3-1 provides a novel control method compared with conventional cascaded control method. It directly regulated the local bus voltage and frequency, and maintained the power flow balance in the system. It has a faster dynamic performance and higher stability.

The configuration using induction generator is another contribution of this research work. Because its own advantages, such as reliable, low cost, low maintenance, this configuration has better cost per kW. The development of the small signal model is very useful for the controller design. The model accurately represents of the system behaviour around the quiescent operating point.

5.3 Further Research Works

In order to accommodate the system variations and improve the control system performance, the following research work should be done in the further.

1) Overload test should be done in the experimental conditions. The drawback of the direct voltage control is capability of current control. Current protection scheme should be improved if necessary.

2) Multiple generators operation should be investigated in the future research. Sometimes there are more than one wind turbine generators are connected to the local bus. The system model will become complicated and the control of ESS will be difficult.

3) The characteristics of the different energy storage devices should be considered. In this thesis, the energy storage device is assumed as an ideal source. In practice, the characteristic of the energy storage device needs to be considered and optimized power management should be implemented to increase the battery life.

References

- [1] M. A. Elhadidy and S. M. Shaahid, "Role of hybrid (wind+diesel) power systems in meeting commercial loads ", IEEE Transaction on Renewable Energy, vol. 29 pp. 109-118, 2004.
- [2] L. Wei, G. Joos, and C. Abbey, "Wind Power Impact on System Frequency Deviation and an ESS based Power Filtering Algorithm Solution," Conference record of Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES, 2006.
- [3] J. M. Carrasco, E. Galvan, R. Portillo, L.G. Franquelo and J. T. Bialasiewicz. "Power Electronics Systems for the Grid integration of Wind Turbine," Conference record of IECON 2006- 32nd Annal Conference, 2006.
- [4] J. Marques, H. Pinheiro, H. A. Gründling, J. R. Pinheiro, and H. L. Hey, "A Survey on Variable-Speed Wind Turbine System," conference record of CE. 7^o Congresso Brasileiro Eletrônica Potência - COBEP'03, Fortaleza, Brazil, 2003.
- [5] Chen, Z and Spooner, E., "Grid interface options for variable-speed, permanent-magnet generators," Electric Power Applications, IEE Proceedings, Vol 145, pp 273-283, July 1998.
- [6] A. S. Kini, A. S. Kini, and R. Y. Udaykumar, "Modelling And Performance Analysis of A Wind/Diesel Hybrid Power System," conference record of Industry Applications, 2006. 41st IAS Annual Meeting.
- [7] R. Sebastian, M. Castro, E. Sancristobal, F. A. Yeves, J. A. Peire, and J. A. Quesada, "Approaching hybrid wind-diesel systems and controller area network," Conference record of IECON 02, 2002.
- [8] W. E. Leithead, "Dependence of performance of variable speed wind turbines on the turbulence, dynamics and control," Generation, Transmission and Distribution, vol. 137, pp. 403-413, 1990.

- [9] R. Cardenas, R. Pena, J. Proboste, G. A. Asher, and J. A. Clare, "MRAS observer for sensorless control of standalone doubly fed induction generators," *IEEE Transaction on Energy Conversion*, vol. 20, pp. 710-718, 2005.
- [10] R. Cardenas, R. Pena, J. Proboste, G. A. Asher, and J. A. Clare, "Sensorless control of a doubly- fed induction generator for stand alone operation," *Conference record of Power Electronics Specialists Conference*, 2004. PESC 04. 2004 IEEE 35th Annual, 2004.
- [11] R. Cardenas, R. Pena, G. Asher, and J. A. Clare, "Control strategies for energy recovery from a flywheel using a vector controlled induction machine," *Conference record of Power Electronics Specialists Conference*, 2000. PESC 00. 2000 IEEE 31st Annual, 2000.
- [12] R. Cardenas, R. Pena, G. Asher, and J. A. Clare, "Control strategies for enhanced power smoothing in wind energy systems using a flywheel driven by a vector-controlled induction machine," *IEEE Transactions on Industrial Electronics*, vol. 48, pp. 625-635, 2001.
- [13] R. Cardenas, R. Pena, G. M. Asher, J. A. Clare, and R. A. Blasco-Gimenez, "Control strategies for power smoothing using a flywheel driven by a sensorless vector-controlled induction machine operating in a wide speed range," *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 603-614, 2004.
- [14] R. Cardenas, R. Pena, M. Perez, J. A. Clare, G. A. Asher, and P. A. Wheeler, "Power Smoothing Using a Flywheel Driven by a Switched Reluctance Machine," *IEEE Transactions on Industrial Electronics*, vol. 53, pp. 1086-1093, 2006.
- [15] J. Yan, K. Oti, N. Yamamura, and M. A. I. M. Ishida, "A Study on Electric Power Smoothing System for Lead-acid Battery of Stand-alone Natural Energy Power System Using EDLC," *Conference record of Power Conversion Conference - Nagoya*, 2007. PCC '07, 2007.
- [16] A. Rufer and P. Barrade, "A supercapacitor-based energy-storage system for elevators with soft commutated interface," *IEEE Transactions on Industry Applications*, vol. 38, pp. 1151-1159, 2002.
- [17] Z. Jie, Z. Buhan, M. Chengxiong, and A. Y. W. Yunling Wang, "Use of Battery Energy Storage System to Improve the Power Quality and Stability of Wind Farms," *Conference record of Power System Technology*, 2006. PowerCon 2006. International Conference on, 2006.

- [18] T. Tudorache, L. Melcescu, and S. V. Paturca, "Finite Element Analysis of Self-Excited Induction Generator for Isolated Small Power Wind Turbines," Conference record of Clean Electrical Power, 2007. ICCEP '07. International Conference on, 2007.
- [19] Bin Wu, High-Power Converters and AC Drives, John Wiley & IEEE Press, 2006.

Appendix B System Parameters

Table B-1 Parameters of DC Motor

Operating Condition	Motor
Power	2 kW
Armature Voltage	120 V
Shunt Voltage	120V
Armature Current	23 A
Shunt Current	0.81A
Speed	1800 rpm
Armature Inductance	5mH
Armature Resistance	2 Ω

Table B-2: System Parameters of Prototypes System

System		Ratings		Generator Parameters	
Power	2kW	Current	5.55A	Stator, Rotor Resistances	0.032pu
Voltage	208V	Frequency	60Hz	Stator, Rotor Leakage Inductances	0.061pu
Excitation Capacitor		141 μ F		Magnetization Inductance	0.875pu
Converter Inductance		0.2pu		Pole Pairs	2

Appendix C Code for Bode Plots

M-file for bode plots of compensated and uncompensated loops

```
clear all;
RL=22;Rs=0.46;Rr=0.70;Lls=0.0035;Llr=0.0035;Li=10e-3;
RL1=10;RL2=40;RL3=100;
C=1.41e-4;
Kpwm=245;
Rl=Rs+Rr;Ll=Lls+Llr;
s=tf('s');
G=(RL*(Rl+s*Ll)/(RL+Rl+s*Ll+s*C*(RL*(Rl+s*Ll)))/(s*Li+(RL*(Rl+s*Ll)/(RL+Rl+s*Ll+s*C*(RL*(Rl+s*Ll)))));
Tu=G*Kpwm;
Gc0=0.077;
Tz=1/2/pi/340; Tp=1/2/pi/2900; Ti=1/(2*pi*100); %Time constant of LEAD-LAG
Gc=Gc0*(((1+Tz*s)/(1+Tp*s))*(1+1/Ti/s)); %LEAD-LAG controller
T=Gc0*Kpwm*Gc; %open loop gain with compensator
bodeplot(Tu,T);
```

Appendix D DSP Program Code

DSP Program Code

```

/*****
/*File Name: wecs.c
*****/

#include "DSP281x_Device.h" // DSP281x
Headerfile Include File
#include "DSP281x_Examples.h" // DSP281x
Examples Include File
#include "FPGA_csr.h"
#include "Display.h"
#include <stdio.h>
#include "IQmathLib.h"

// Useful Definitions
#define ADC_usDELAY 8000
#define ADC_usDELAY2 40

#define PI 3.1415926
#define sqrt3 1.732051
#define sqrt13 0.577350269
#define Sw_freq 8997.1//sampling frequency

#define AVE_SAMPLE 8
//AVE_SAMPLE=2^AVE_SHFT
#define AVE_SHFT 3
#define AVE_SAMPLE_DC 16
//AVE_SAMPLE_DC=2^AVE_SHFT_DC
#define AVE_SHFT_DC 4

#define w_DC 1000.0//cut-off frequency(rad/s) of
low-pass filter for DC voltage

#define XL 0.016//0.016//Line to Converter
inductance

#define sen_posi 0//current sensor position,
0:converter side, 1:grid side
#define Kp_Udc 0.5
#define Ki_Udc 40.0//for DC voltage control loop
#define Kp_I 0.04
#define Ki_I 30.0//for current control loop
#define Kp_E 0.0796
#define Ki_E 0.005//for loacal voltage control loop
#define Kpp_E 4.74
#define Kii_E 3.735
#define Kdd_E 0.0058

#define K_PLL 778700.0//gain for motor angle
close loop

#pragma DATA_SECTION(graph_data,"graph_data");
#pragma DATA_SECTION(graph_data1,"graph_data1");
#pragma DATA_SECTION(graph_data2,"graph_data2");
#pragma DATA_SECTION(graph_data3,"graph_data3");
#pragma DATA_SECTION(graph_data4,"graph_data4");
#pragma DATA_SECTION(graph_data5,"graph_data5");
#pragma DATA_SECTION(graph_data6,"graph_data6");

//pragma
DATA_SECTION(wind_profile,"wind_profile");

*****/
/* VARIABLE DECLARATION
*/
/*****

/* ADC */
Uint16 Sample_ch0[AVE_SAMPLE],
Sample_ch1[AVE_SAMPLE],
Sample_ch2[AVE_SAMPLE],
Sample_ch3[AVE_SAMPLE],
Sample_ch4[AVE_SAMPLE],
Sample_ch5[AVE_SAMPLE_DC],
Sample_ch6[AVE_SAMPLE_DC],
Sample_ch7[AVE_SAMPLE],
Sample_ch8[AVE_SAMPLE_DC],
Sample_ch9[AVE_SAMPLE],
Sample_ch10[AVE_SAMPLE],
Sample_ch11[AVE_SAMPLE],
Sample_ch12[AVE_SAMPLE],
Sample_ch13[AVE_SAMPLE],
Sample_ch14[AVE_SAMPLE],
Sample_ch15[AVE_SAMPLE];

Uint16 ConversionCount=0;
Uint16 ConversionCount1=0;
int32 AD_BIAS0, AD_BIAS1, AD_BIAS2, AD_BIAS3,
AD_BIAS4, AD_BIAS5, AD_BIAS6, AD_BIAS7,
AD_BIAS8, AD_BIAS9, AD_BIAS10, AD_BIAS11,
AD_BIAS12, AD_BIAS13, AD_BIAS14,
AD_BIAS15;

int32 Ia1, Ic1, Ia2, Ic2; //( _iq18 format)

```

```

/* for testing*/
Uint32 watch=0;
Uint16 watch1=0;
Uint16 watch2=0;
Uint16 watch3=0;
Uint16 Testswitch=0;

/*PLL */
Uint16 Theta_con, Theta_err; //phase and error info in
counter format
Uint16 nmax,f;
Uint32 Theta_pu; //line voltage angle in per unit value
 iq28 Theta_line; //line voltage angle in radian value
(0-2pi)
 iq28 Theta_phase,angle_offset; //phase voltage angle in
radian value
 iq28 cos_theta, sin_theta;

/*=====VSR variables===== */
float Udc_ref=400; // DC voltage reference
 iq18 Vab, Vbc, Vca,
    Ia, Ib, Ic,
    Vdc_P,Vdc_N;

 iq18 Va, Vb, Vc;
 iq18 V_alpha, V_beta;
 iq18 V_d, V_q;

/*Inner loop current control*/
 iq18 I_alpha, I_beta;
 iq18 I_d,I_d_ref, I_d_err,I_d_err1,
upperlimit_Id,lowerlimit_Id;
 iq18 I_q,I_q_ref, I_q_err,I_q_err1;

 iq18 V_d_O, V_q_O;
 iq18 I_d_in1, I_d_in2, I_d_out, I_d_out1, I_d_out2;
 iq18 I_q_in1, I_q_in2, I_q_out, I_q_out1, I_q_out2;
 iq28 kp_I, ki_I; //PI controller

/*Outer loop DC voltage control*/
 iq18 Vdc_ref,Vdc,Vdc0,Vdc_L,Vdc_L0,
Vdc_err,Vdc_err1;
 iq28 kp_dc, ki_dc; //PI controller

/*=====VSI variables===== */
float Eq_ref=0;
 iq18 Eab, Ebc, Eca,Ed_ref,Eab0,Vab0,
    Iu, Iv, Iw;
 iq18 Tu, Tw;

 iq18 Ea, Eb, Ec;
 iq18 E_alpha, E_beta;
 iq18 Iuvw_alpha, Iuvw_beta;
 iq18 E_d, E_q,Ed_0, Eq_0;

 iq18 I_E_d, I_E_q;
 iq18 V_d_L,V_q_L;

 iq18 Ed_err, Eq_err, erro_d, erro_q,Ed_err0, Eq_err0,
    Ed_err1, Eq_err1, Eq_D, Ed_D,Eq_D1,
    Ed_D1,
    Ed_0, Eq_0,Ed_1, Eq_1;
 iq28 kp_E, ki_E, kpp_E,kii_E,kdd_E; // LEAD-LAG
controller

/*Low-pass filter*/
 iq28 LPFa_DC,LPFb_DC;

/* SVPWM*/
Uint16 Ts=16672; //PWM period in counter value
Uint16 T1,T2,T0,Tc=0;
int16 section_number;
 iq28 Theta,Theta1,Theta_sec;
 iq18 Mmag,Mmag1;

/*Generator position and speed up*/
 iq18 Speed_n;
 iq28 Theta_INV;
Uint16 Angle_CNT,Angle_CNT0,Speed_CNT;
Uint16 Angle_index=0;
Uint32 f_counter;
Uint16 count_f, fL, F;

/*Key input*/
int16 key_value,key_value1, key_monitor,
key_monitor1,key1;

/* Display*/
Uint16 display_counter,display_counter1;

/*Control flags*/
Uint16 MODE, //1:Rec 2:Inverter
    CONTACTOR, /*For VSR:0-open, 1-close*/
    CONTACTOR1, /*For VSI:0-open, 1-close*/
    FAULT, /*0-no fault 1- fault detected*/
    PLL, /*0-PLL not locked 1-locked*/
    Fault_info_A, /*Result of fault detection from
left (A) side*/
    Fault_info_B, /*Result of fault detection from left (B)
side*/
    Initial_over, /*"0-not over, 1- over"/
    SCREEN; /*0-system status screen0 1-status
screen 1 */
Uint16 Res_flag=0;
Uint16 test_number=0;

Uint16 test_flag1=0;

```

```

Uint16 Ready=0;

Uint16 T_CNT0=0;
Uint16 T_CNT1=0;
Uint16 T_CNT2=0;

/*Graph of waveform*/
int16 graph_data[9000];//Vab
int16 graph_data1[9000];//I_d_ref
int16 graph_data2[9000];//V_d
int16 graph_data3[9000];//V_q
int16 graph_data4[9000];//I_q
int16 graph_data5[9000];//Vdc
int16 graph_data6[9000];//Theta_phase

Uint16 index0=0;
Uint16 graph_index=0;
Uint16 graph_index1=0;
Uint16 graph_index2=0;
Uint16 graph_index_flag=0;
Uint16 data_shift_times=0;
int16 data_shift_times1=0;

/*****
*/
/*FUNCTION DECLARATION      */
/*****

interrupt void adc_isr(void);
interrupt void pdpinta_isr(void);
interrupt void t3cint_isr(void);
//interrupt void EVB_CAPINT4_ISR(void);
//interrupt void EVB_CAPINT5_ISR(void);
//interrupt void t1ufint_isr(void);

void Protection_PLL (void);
void get_theta(void);
void get_frequency (void);
void get_adc_V(void);
void abc_dq_transformation(void);
void speed_up(void);
void VSR_Regulation(void);
void VSI_Regulation(void);
void SVPWM_VSR(void);
void SVPWM_VSI(void);
void Enable_Rec(void);
void Disable_Rec(void);
void Enable_Inv(void);
void Disable_Inv(void);
void Key_input(void);
void Close_Relay(void);
void Close_Relay1(void);
void Open_Relay(void);
void Open_Relay1(void);
void AD_bias16(void);
void Delay_mili_second (int delay);
void get_speed(void);

void Init_variable(void);
void InitGpio(void);
void InitXintfClocks(void);
void InitPeripherals(void);
void InitEVA(void);
void InitEVB(void);
void InitAdc(void);
void init_fpga(void);
void Buffer_rotating(int16 buffer[9000],Uint16
shift_times);

/*****
*/
/* NAME:      main()      */
/* RETURNS:   void        */
/*****

void main(void)
{
// Step 1. Initialize System Control:
// PLL(PLLCR=0xA), WatchDog(disable), enable
Peripheral Clocks
InitSysCtrl();

// Step 2. Initialize the Xintf clocks
InitXintfClocks();

// Step 3. Initialize GPIO:
InitGpio();

// Step 4. Clear all interrupts and initialize PIE vector
table:
DINT; // Disable CPU interrupts
InitPieCtrl(); // Initialize the PIE control registers to
their default state.
// The default state is all PIE interrupts
disabled and flags
// are cleared.

IER = 0x0000; // Disable CPU interrupts
IFR = 0x0000; // Clear all CPU interrupt flags

InitPieVectTable(); // Initialize the PIE vector table

// Interrupts used are re-mapped to ISR functions
EALLOW;
PieVectTable.PDPINTA = &pdpinta_isr;
PieVectTable.ADCINT = &adc_isr;
PieVectTable.T3CINT = &t3cint_isr;
PieVectTable.CAPINT4=&EVB_CAPINT4_ISR;
PieVectTable.CAPINT5=&EVB_CAPINT5_ISR;
//PieVectTable.T1UFINT = &t1ufint_isr;
EDIS;

// Step 5. Initialize all the Device Peripherals:
InitPeripherals();

```

```
// Step 6. User specific code, enable interrupts:
PieCtrlRegs.PIEIER1.bit.INTx1 = 1; //Enable
PDPINTA
PieCtrlRegs.PIEIER1.bit.INTx6 = 1; // Enable
ADCINT in the PIE: Group 1 interrupt 6
PieCtrlRegs.PIEIER4.bit.INTx5 = 1; // Enable t3cint in
the PIE: Group 4 interrupt 5
PieCtrlRegs.PIEIER5.bit.INTx5 = 1; // Enable
CAPINT4 in the PIE: Group 5 interrupt 5
PieCtrlRegs.PIEIER5.bit.INTx6 = 1; // Enable
CAPINT5 in the PIE: Group 5 interrupt 6
//PieCtrlRegs.PIEIER2.bit.INTx6 = 1; // Enable t1ufint
in the PIE: Group 2 interrupt 6
```

```
IER |= M_INT1; // Enable CPU INT1:
IER |= M_INT4; // Enable CPU INT4:
IER |= M_INT5; // Enable CPU INT5:
```

```
EINT; // Enable Global interrupt INTM
ERTM; // Enable Global realtime interrupt DBGM
```

```
Clear_LED();
AD_bias16();
Manu_display();
AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 1; //
enable SEQ1 interrupt (every EOS)
```

```
// Step 7. Infinite loop
while(1)
{
    if (FAULT == 0)Key_input ();
    display_counter++;
    if(display_counter==20000)
    {
        display_counter1++;display_counter=0;
    }
    if (display_counter1==30)
    {
        //Text_LCD_display();
        //key1=1;//for test
        LED_display();
        display_counter1=0;
        display_counter=0;
    }
}
} /*End Main*/
```

```
/* *****
/* NAME: adc_isr()
RETURNS: void
DESCRIPTION: Interrupt for ADC results reviewal
*/
/* *****
```

```
interrupt void adc_isr(void)
```

```
{
Sample_ch0[ConversionCount]=((AdcRegs.ADCRESU
LT0>>4) );
Sample_ch1[ConversionCount]=((AdcRegs.ADCRESU
LT1>>4) );
Sample_ch2[ConversionCount]=((AdcRegs.ADCRESU
LT2>>4) );
Sample_ch3[ConversionCount]=((AdcRegs.ADCRESU
LT3>>4) );
Sample_ch4[ConversionCount]=((AdcRegs.ADCRESU
LT4>>4) );
Sample_ch7[ConversionCount]=((AdcRegs.ADCRESU
LT7>>4) );
Sample_ch9[ConversionCount]=((AdcRegs.ADCRESU
LT9>>4) );
Sample_ch10[ConversionCount]=((AdcRegs.ADCRESU
LT10>>4) );
Sample_ch11[ConversionCount]=((AdcRegs.ADCRESU
LT11>>4) );
//
Sample_ch12[ConversionCount]=((AdcRegs.ADCRESU
LT12>>4) );
//
Sample_ch13[ConversionCount]=((AdcRegs.ADCRESU
LT13>>4) );
//
Sample_ch14[ConversionCount]=((AdcRegs.ADCRESU
LT14>>4) );
//
Sample_ch15[ConversionCount]=((AdcRegs.ADCRESU
LT15>>4) );
if (ConversionCount==AVE_SAMPLE-1)
ConversionCount=0;
else ConversionCount++;
Sample_ch5[ConversionCount1]=((AdcRegs.ADCRESU
LT5>>4) );//Udc+
Sample_ch6[ConversionCount1]=((AdcRegs.ADCRESU
LT6>>4) );//Udc-
```

```
Sample_ch8[ConversionCount1]=((AdcRegs.ADCRESU
LT8>>4));/If
```

```
if (ConversionCount1==AVE_SAMPLE_DC-1)
ConversionCount1=0;
else ConversionCount1++;
```

```
AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;    // Reset
SEQ1
AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;    // Clear
INT_SEQ1 bit
PieCtrlRegs.PIEACK.all = PIEACK_GROUP1; //
Acknowledge interrupt to PIE
} /*End of Adc_isr*/
```

```
/******
```

```
/* NAME:      t3cint_isr()
/* DESCRIPTION: Interrupt for computing
*/
```

```
/******
```

```
interrupt void t3cint_isr(void)
{
```

```
EINT;
watch1= EvbRegs.T3CNT;
```

```
get_theta();
```

```
get_speed();
```

```
get_adc_V();
```

```
get_frequency();
```

```
abc_dq_transformation();
```

```
switch (MODE)
```

```
{
case 1:
upperlimit_Id=_IQ18(10);
```

```
VSR_Regulation();
```

```
SVPWM_VSR();
```

```
break;
```

```
case 2:
```

```
speed_up();
```

```
upperlimit_Id=_IQ18(10);
```

```
VSR_Regulation();
```

```
SVPWM_VSR();
```

```
VSI_Regulation();
```

```
SVPWM_VSI();
```

```
break;
```

```
}
```

```
/*for waveforms display*/
```

```
graph_data[graph_index]= Iu>>12;
graph_data1[graph_index]=Iw>>12;
graph_data2[graph_index]= I_E_d>>12;
graph_data3[graph_index]= I_E_q>>12;
graph_data4[graph_index]=E_d>>10;
graph_data5[graph_index]=E_q>>10;
graph_data6[graph_index]=F;
graph_index++;
if (graph_index== 9000) graph_index=0;
if (graph_index_flag==1) {graph_index1++;}
if (graph_index1==6300) { graph_index1--
;graph_index--;}

/*sampling data of speed*/ //9000points for 60s
//if (graph_index1== 60)
//{graph_data[graph_index2]= torque_G_L>>8;
//graph_data5[graph_index2]= veloc_y>>11;
// graph_data4[graph_index2]= torque_shaft>>8;
//graph_index2++;graph_index1=0;}
//if (graph_index2==9000)
{graph_index1=0;graph_index2=0;graph_index_flag=0;}
```

```
/*sampling data of speed*/ //9000points for 60s
```

```
//if (graph_index1== 60)
```

```
//{graph_data[graph_index2]= torque_G_L>>8;
```

```
//graph_data5[graph_index2]= veloc_y>>11;
```

```
// graph_data4[graph_index2]= torque_shaft>>8;
```

```
//graph_index2++;graph_index1=0;}
```

```
//if (graph_index2==9000)
```

```
{graph_index1=0;graph_index2=0;graph_index_flag=0;}
```

```
watch2= EvbRegs.T3CNT;
```

```
if (watch2>watch1) {watch3=watch2-watch1;}
```

```
else {watch3=watch1-watch2;}
```

```
EvbRegs.EVBIFRA.bit.T3CNT=1;// Clear INT
```

```
PieCtrlRegs.PIEACK.all = PIEACK_GROUP4;//
```

```
Acknowledge interrupt to PIE
```

```
} /*End of t3cint_isr*/
```

```
/******
```

```
/*NAME:
```

```
EVB_CAPINT4_ISR(),EVB_CAPINT4_ISR()*/
```

```
/******
```

```
interrupt void EVB_CAPINT4_ISR(void) // CAP4
```

```
{
```

```
CAP_CNT0=EvbRegs.CAP4FBOT;
```

```
Angle_CNT0=EvaRegs.T2CNT;
```

```
EvbRegs.EVBIFRC.all=BIT0;
```

```
PieCtrlRegs.PIEACK.all=PIEACK_GROUP5;
```

```
}
```

```
interrupt void EVB_CAPINT5_ISR(void) // CAP5
```

```
{
```

```
CAP_CNT0=EvbRegs.CAP5FBOT;
```

```
Angle_CNT0=EvaRegs.T2CNT;
```

```
EvbRegs.EVBIFRC.all=BIT1;
```

```
PieCtrlRegs.PIEACK.all=PIEACK_GROUP5;
```

```
}
```

```
/******
```

```
/* NAME:      pdpinta_isr()
```



```

/* DESCRIPTION: Protection interrupt generated from
FPGA */
/*****/

interrupt void pdpinta_isr(void)
{
    Open_Relay ();
    Disable_Rec();
    Disable_Inv();

    INTERRUPT = 0;
    FAULT = 1;
    CONTACTOR = 0;
    Fault_info_A=*FAULT_DETECT_A;
    Fault_info_B=*FAULT_DETECT_B;
    Display_Fault ();
    PieCtrlRegs.PIEACK.all =
    PIEACK_GROUP1;// Acknowledge interrupt to PIE
    /*No clear of the interrupt flag*/
} /*End of Pdpinta_isr*/

/*****/
/* NAME: Protection_PLL() */
/* DESCRIPTION: Protection action when PLL is lost */
/*****/

void Protection_PLL (void)
{
    Open_Relay ();
    Disable_Rec();
    Disable_Inv();
    INTERRUPT = 0;
    FAULT = 1;
    CONTACTOR = 0;
    Fault_info_A=*FAULT_DETECT_A;
    Fault_info_B=*FAULT_DETECT_B;
    Display_Fault ();
} /*End of Protection_PLL*/

/*****/
/* NAME: get_theta() */
/* DESCRIPTION: phase for abc-dq transformation */
/*****/

void get_theta(void)
{
    Theta_con = *PHASE_A; /*phase info in counter
format*/
    Theta_err = *PHASE_LOCK_A; /*DPLL phase
error in counter format*/
    nmax = *N_MAX_A + 1; /* */
    /*PLL phase information*/
    if ((Theta_err>15)&&(Theta_err<480))
    {

```

```

        PLL = 0;
        //Protection_PLL();
        }
        else {PLL =1;}
        if (PLL==0) test_number++;
    }

/*****/
/* NAME: get_adc_V() */
/* DESCRIPTION: Line voltage and current
conditioning with multi-sampling */
/*****/

void get_adc_V(void)
{
    Uint16 i;

    Uint32 Sum_ch0=0;
    Uint32 Sum_ch1=0;
    Uint32 Sum_ch2=0;
    Uint32 Sum_ch3=0;
    Uint32 Sum_ch4=0;
    Uint32 Sum_ch5=0;
    Uint32 Sum_ch6=0;
    Uint32 Sum_ch7=0;
    Uint32 Sum_ch8=0;
    Uint32 Sum_ch9=0;
    Uint32 Sum_ch10=0;
    Uint32 Sum_ch11=0;
    //Uint32 Sum_ch12=0;
    //Uint32 Sum_ch13=0;
    //Uint32 Sum_ch14=0;
    //Uint32 Sum_ch15=0;

    for(i = 0; i<AVE_SAMPLE; i++)
    {
        Sum_ch0+=Sample_ch0[i];
        Sum_ch1+=Sample_ch1[i];
        Sum_ch2+=Sample_ch2[i];
        Sum_ch3+=Sample_ch3[i];
        Sum_ch4+=Sample_ch4[i];
        Sum_ch7+=Sample_ch7[i];

        Sum_ch9+=Sample_ch9[i];
        Sum_ch10+=Sample_ch10[i];
        Sum_ch11+=Sample_ch11[i];
        //Sum_ch12+=Sample_ch12[i];
        //Sum_ch13+=Sample_ch13[i];
        //Sum_ch14+=Sample_ch14[i];
        //Sum_ch15+=Sample_ch15[i];
    }
    for(i = 0; i<AVE_SAMPLE_DC; i++)
    {
        Sum_ch5+=Sample_ch5[i];
        Sum_ch6+=Sample_ch6[i];

```

```

Sum_ch8+=Sample_ch8[i];
}
/*****/

Ia2=Ia1;
Ia1=Ia;
Ic2=Ic1;
Ic1=Ic;
/*Chanel 1*/
Ia=Sum_ch0 - (AD_BIAS0<<(AVE_SHFT));
Ia=(Ia<<(18-AVE_SHFT)); //(iq18 format)
Ia=Ia/2035; /*2023 for Iag,2035 for Ia*/
Ia=-Ia*50; /*current sensor 5:1 */

/*Chanel 2*/
Ic=Sum_ch1 - (AD_BIAS1<<(AVE_SHFT));
Ic=(Ic<<(18-AVE_SHFT)); //(iq18 format)
Ic=Ic/2050; /*2002 for Icg,2050 for Ic*/
Ic=-Ic*50; /*current sensor 5:1 */

/*Chanel 3*/
Vab=Sum_ch2 - (AD_BIAS2<<AVE_SHFT);
Vab=(Vab<<(18-AVE_SHFT)); //(iq18 format)
Vab=Vab/2186; /*DSP ADC trim: 5V=3159 - 5V=1039*/
Vab=Vab*450; /*Voltage sensor 45:1 */

/*Chanel 4*/
Vbc=Sum_ch3 - (AD_BIAS3<<AVE_SHFT);
Vbc=(Vbc<<(18-AVE_SHFT)); //(iq18 format)
Vbc=Vbc/2065; /*DSP ADC trim: 5V=3156 - 5V=1042*/
Vbc=Vbc*450; /*Voltage sensor 45:1 */

/*Chanel 5*/
Vca=Sum_ch4 - (AD_BIAS4<<AVE_SHFT);
Vca=(Vca<<(18-AVE_SHFT)); //(iq18 format)
Vca=Vca/2153; /*DSP ADC trim: 5V=3170 - 5V=1026*/
Vca=Vca*450; /*Voltage sensor 45:1 */

/*Chanel 6*/
Vdc_P = Sum_ch5 - (AD_BIAS5<<(AVE_SHFT_DC));
Vdc_P = (Vdc_P<<(18-AVE_SHFT_DC)); //(iq18 format)
Vdc_P=Vdc_P/2204; /*DSP ADC trim: 5V=1041 -5V=3158*/
Vdc_P=-Vdc_P*300; /* sensor 30:1 */

/*Chanel 7*/
Vdc_N = Sum_ch6 - (AD_BIAS6<<(AVE_SHFT_DC));

```

```

Vdc_N = (Vdc_N<<(18-AVE_SHFT_DC)); //(iq18 format)
Vdc_N=Vdc_N/2274; /*DSP ADC trim: 5V=1015 -5V=3130*/
Vdc_N=-Vdc_N*300; /* sensor 30:1 */
Vdc0=Vdc;
Vdc=Vdc_P+Vdc_N;

/*Chanel 12*/
Iu=Sum_ch7 - (AD_BIAS7<<(AVE_SHFT));
Iu=(Iu<<(18-AVE_SHFT)); //(iq18 format)
//Iu=Iu-20972;
Iu=Iu/2510;
Iu=-Iu*50; /*current sensor 5:1 */

/*Chanel 13*/
Iw=Sum_ch8 - (AD_BIAS8<<(AVE_SHFT_DC));
Iw=(Iw<<(18-AVE_SHFT_DC)); //(iq18 format)
Iw=Iw/1990;
Iw=-Iw*50; /*current sensor 5:1 */

/*Chanel 14*/
Eab=Sum_ch9 - (AD_BIAS9<<AVE_SHFT);
Eab=(Eab<<(18-AVE_SHFT)); //(iq18 format)
Eab=Eab/2208; /*DSP ADC trim: 5V=3159 - 5V=1039*/
Eab=Eab*450; /*Voltage sensor 45:1 */

/*Chanel 15*/
Ebc=Sum_ch10 - (AD_BIAS10<<AVE_SHFT);
Ebc=(Ebc<<(18-AVE_SHFT)); //(iq18 format)
Ebc=Ebc/2171; /*DSP ADC trim: 5V=3156 - 5V=1042*/
Ebc=Ebc*450; /*Voltage sensor 45:1 */

/*Chanel 9*/
Eca=Sum_ch11 - (AD_BIAS11<<AVE_SHFT);
Eca=(Eca<<(18-AVE_SHFT)); //(iq18 format)
Eca=Eca/2171; /*DSP ADC trim: 5V=3170 - 5V=1026*/
Eca=Eca*450; /*Voltage sensor 45:1 */

/*Low pass filtering for Vdc*/
Vdc_L0=Vdc_L;
Vdc_L= _IQ18mpyIQX(LPFa_DC,28,(Vdc0+Vdc),18);
Vdc_L= _IQ18mpyIQX(LPFb_DC,28,Vdc_L0,18)+Vdc_L;

```

```

} /*End of Get_adc_V*/

/*****/
/* NAME:    get_speed() */
/* RETURNS: void */
/* DESCRIPTION: Generator speed w(rad/s)*/
/*****/
void get_speed(void)
{
    Angle_CNT0=Angle_CNT;

    Angle_index++;
    if (Angle_index==90)
        {Angle_CNT=EvaRegs.T2CNT;
        if (Angle_CNT<Angle_CNT0)
            Speed_CNT=Angle_CNT+14400-Angle_CNT0;
        else Speed_CNT=Angle_CNT-Angle_CNT0;
            Angle_index=0;
            Speed_n=_IQ18mpyIQX( _IQ8(Speed_CNT), 8,
            _IQ28(6000.0/14400), 28); //motor speed in (rpm)
        }

}

/*****/
/* NAME:    speed_up() */
/* DESCRIPTION: U/f control to speed up: f:0--60Hz in
20s */
/*****/

void speed_up(void)
{
    if (f<121)
        { f_counter++;
          if (f_counter==3000)
            {f=f+1; f_counter=0;}}

    else f=120;

    Theta_INV += _IQ28mpyIQX( _IQ18(f/2.0), 18,
    _IQ28(2*PI/9000.0), 28);
    if ( Theta_INV > _IQ28(2*PI) )
        Theta_INV = Theta_INV - _IQ28(2*PI);
    if ( Theta_INV < 0 )
        Theta_INV = Theta_INV + _IQ28(2*PI);
}

void get_frequency(void)
{
    count_f++;
    if ((Vab>0) && (Vab0<0)) {fL++;}
    Vab0=Vab;

```

```

if (count_f==290) {count_f=0;F=fL*30;fL=0;}
}

/*****/
/* NAME:    abc_dq_transformation() */
/* DESCRIPTION: abc to dq transformation */
/*****/

void abc_dq_transformation(void)
{
    Theta_pu = (long)Theta_con<<16;
    Theta_pu = Theta_pu/nmax;
    Theta_pu = _IQ28(1)- (Theta_pu<<12); /*PLL is in
counting down mode*/
    Theta_line = _IQ28mpy( _IQ28(2*PI), Theta_pu);
    /*Va phase angle*/

    Theta_phase = Theta_line- _IQ28(PI/2); /*angle for
DQ transformation */
    Theta_phase = Theta_phase- angle_offset;

    if ( Theta_phase > _IQ28(2*PI) )
        Theta_phase = Theta_phase - _IQ28(2*PI);
    if ( Theta_phase < 0 )
        Theta_phase = Theta_phase + _IQ28(2*PI);

    sin_theta = _IQ28sin(Theta_phase);
    cos_theta = _IQ28cos(Theta_phase);

    /*Convert voltage from a-b-c to d-q*/
    Va = (Vab - Vca)/3;
    Vb = (Vbc - Vab)/3;
    Vc = (Vca - Vbc)/3;

    V_alpha = Va;
    V_beta = _IQ18mpy( _IQ18(sqrt13), (Vb - Vc));

    V_d = _IQ18mpyIQX(cos_theta, 28, V_alpha, 18);
    V_d += _IQ18mpyIQX(sin_theta, 28, V_beta, 18);
    V_q = _IQ18mpyIQX(-sin_theta, 28, V_alpha, 18);
    V_q += _IQ18mpyIQX(cos_theta, 28, V_beta, 18);

    /*Convert current from a-b-c to d-q*/

    Ib = -Ia-Ic;

    I_alpha = Ia;
    I_beta = _IQ18mpy( _IQ18(sqrt13), (Ib - Ic));

    I_d = _IQ18mpyIQX(cos_theta, 28, I_alpha, 18);
    I_d += _IQ18mpyIQX(sin_theta, 28, I_beta, 18);
    I_q = _IQ18mpyIQX(-sin_theta, 28, I_alpha, 18);

```

```

I_q += _IQ18mpyIQX(cos_theta, 28, I_beta, 18);

/*Inverter voltage from a-b-c to d-q*/

Ea = (Eab - Eca)/3;
Ea = Ea/170;
Eb = (Ebc - Eab)/3;
Eb = Eb/170;
Ec = (Eca - Ebc)/3;
Ec = Ec/170;

sin_theta = _IQ28sin(Theta_INV);
cos_theta = _IQ28cos(Theta_INV);

E_alpha = Ea;
E_beta = _IQ18mpy(_IQ18(sqrt13), (Eb - Ec));

E_d = _IQ18mpyIQX(cos_theta, 28, E_alpha, 18);
E_d += _IQ18mpyIQX(sin_theta, 28, E_beta, 18);
E_q = _IQ18mpyIQX(-sin_theta, 28, E_alpha, 18);
E_q += _IQ18mpyIQX(cos_theta, 28, E_beta, 18);

/*Inverter Current from a-b-c to d-q*/

Iu = Iu-Tu;
Iw = Iw+Tw;
Iv = -Iu-Iw;

Iuvw_alpha = Iu;
Iuvw_beta = _IQ18mpy(_IQ18(sqrt13), (Iv - Iw));

I_E_d = _IQ18mpyIQX(cos_theta, 28,
Iuvw_alpha, 18);
I_E_d += _IQ18mpyIQX(sin_theta, 28, Iuvw_beta,
18);
I_E_q = _IQ18mpyIQX(-sin_theta, 28,
Iuvw_alpha, 18);
I_E_q += _IQ18mpyIQX(cos_theta, 28, Iuvw_beta,
18);

}

/*****/
/* NAME: VSR_Regulation() & VSI_Regulation()
*/
/* DESCRIPTION: Converter side control */
/*****/

void VSR_Regulation(void)
{

```

```

/* Outer loop voltage PI control*/
Vdc_err = Vdc_ref-Vdc_L;
kp_dc= _IQ28(Kp_Udc); //kp=0.5
ki_dc= _IQ28(Ki_Udc/Sw_freq);//ki/fs=45/9k=0.005
if ((Vdc_err<_IQ18(1.2))&&(Vdc_err>_IQ18(-1.2)))

{kp_dc=_IQ28(Kp_Udc/4);ki_dc=_IQ28(Ki_Udc/Sw_fr
eq/4);Res_flag=1;}
//if ((Vdc_err<_IQ18(-
5))&&(Vdc_err>_IQ18(5)))
//{kp_dc=kp_dc<<1;ki_dc=ki_dc<<1;Res_flag=0;}

Vdc_err1 += _IQ18mpyIQX( ki_dc, 28, Vdc_err, 18);
if (Vdc_err1>upperlimit_Id)
Vdc_err1=upperlimit_Id;
if (Vdc_err1<_IQ18(-30)) Vdc_err1=_IQ18(-30);
I_d_ref= _IQ18mpyIQX( kp_dc, 28, Vdc_err,
18)+Vdc_err1;
if (test_flag1==1) I_d_ref= _IQ18(13.7);
if (I_d_ref>upperlimit_Id) I_d_ref=upperlimit_Id;
if (I_d_ref<_IQ18(-30)) I_d_ref=_IQ18(-30);

/* Inner loop current PI-Res control*/
I_d_in2 = I_d_in1;
I_d_in1 = I_d_err;
I_d_out2 = I_d_out1;
I_d_out1 = I_d_out;
I_d_err = I_d-I_d_ref;
//if (Res_flag==1)
//{I_d_out= _IQ28mpyIQX(Res_6a,28,(I_d_err-
I_d_in2),18);
//I_d_out=I_d_out-I_d_out2-
_IQ28mpyIQX(Res_6b,28,I_d_out1,18);}//resonant
regulator

I_d_err1 += _IQ18mpyIQX( ki_I, 28, I_d_err, 18);
if (I_d_err1>_IQ18(0.97)) I_d_err1=_IQ18(0.97);
if (I_d_err1<_IQ18(0.3)) I_d_err1=_IQ18(0.3);
V_d_O = _IQ18mpyIQX(kp_I, 28, I_d_err,
18)+I_d_err1;
if (V_d_O>_IQ18(0.97)) V_d_O=_IQ18(0.97);
if (V_d_O<_IQ18(0.3)) V_d_O=_IQ18(0.3);

V_d_O += _IQ18mpyIQX(_IQ28( XL), 28, I_q, 18);
//if (Res_flag==1) V_d_O +=I_d_out>>10;//add
resonant regulator output
//if (display_counter1<27)
//V_d_O += V_d_H;//for active damping
if (V_d_O>_IQ18(0.97)) V_d_O=_IQ18(0.97);
if (V_d_O<_IQ18(0.3)) V_d_O=_IQ18(0.3);

I_q_in2 = I_q_in1;
I_q_in1 = I_q_err;
I_q_out2 = I_q_out1;
I_q_out1 = I_q_out;
I_q_err = I_q-I_q_ref;

```

```

//if (Res_flag==1)
//{I_q_out= _IQ28mpyIQX(Res_6a,28,(I_q_err-
I_q_in2),18);
// I_q_out=I_q_out-I_q_out2-
_IQ28mpyIQX(Res_6b,28,I_q_out1,18);} //resonant
regulator

I_q_err1 += _IQ18mpyIQX(ki_I, 28, I_q_err, 18);
if (I_q_err1 > _IQ18(0.4)) I_q_err1 = _IQ18(0.4);
if (I_q_err1 < _IQ18(-0.4)) I_q_err1 = _IQ18(-0.4);
V_q_O = _IQ18mpyIQX(kp_I, 28, I_q_err,
18)+I_q_err1;
V_q_O += _IQ18mpyIQX(_IQ28(XL), 28, (-I_d), 18);
//if (Res_flag==1) V_q_O += I_q_out >> 10; //add
resonant regulator output
//if (display_counter1 < 27)
//V_q_O += V_q_H; //for active damping
if (V_q_O > _IQ18(0.4)) V_q_O = _IQ18(0.4);
if (V_q_O < _IQ18(-0.4)) V_q_O = _IQ18(-0.4);

/* for test */
// V_d_O = _IQ18(0.8);
// V_q_O = _IQ18(0);

//if ((E_d < _IQ18(80)) && (graph_index_flag == 0))

//{graph_index_flag = 1; data_shift_times = graph_index;} //
trigger for supply voltage sag

} /*End of Regulation*/

void VSI_Regulation(void)
{
    Ed_ref = _IQ18(f/120.0);
    Ed_err = Ed_ref - E_d;
    kp_E = _IQ28(Kp_E); //kp=0.07969
    ki_E = _IQ28(Ki_E); //ki=0.005

    kpp_E = _IQ28(Kpp_E); //kpp=4.74
    kii_E = _IQ28(Kii_E); //kii=3.735
    kdd_E = _IQ28(Kdd_E); //kdd=0.0058

    Ed_err1 += _IQ18mpyIQX(ki_E, 28, Ed_err,
18);
    if (Ed_err1 > _IQ18(0.97)) Ed_err1 = _IQ18(0.97);
    if (Ed_err1 < _IQ18(0.01)) Ed_err1 = _IQ18(0.01);
    Ed_D1 = Ed_D;
    Ed_D = _IQ18mpyIQX(kp_E, 28, Ed_err,
18)+Ed_err1;
    if (Ed_D > _IQ18(0.97)) Ed_D = _IQ18(0.97);
    if (Ed_D < _IQ18(0.01)) Ed_D = _IQ18(0.01);

    Ed_1 = Ed_0;
    Ed_0 = _IQ18mpyIQX(kpp_E, 28, Ed_D, 18);
    Ed_0 = Ed_0 - _IQ18mpyIQX(kii_E, 28, Ed_D1, 18);

```

```

    Ed_0 = Ed_0 - _IQ18mpyIQX(kdd_E, 28, Ed_1, 18);
    if (Ed_0 > _IQ18(0.97)) Ed_0 = _IQ18(0.97);
    if (Ed_0 < _IQ18(0.01)) Ed_0 = _IQ18(0.01);

    Eq_err = Eq_ref - E_q;
    Eq_err1 += _IQ18mpyIQX(ki_E, 28, Eq_err,
18);
    if (Eq_err1 > _IQ18(0.97)) Eq_err1 = _IQ18(0.97);
    if (Eq_err1 < _IQ18(0.01)) Eq_err1 = _IQ18(0.01);
    Eq_D1 = Eq_D;
    Eq_D = _IQ18mpyIQX(kp_E, 28, Eq_err,
18)+Eq_err1;
    if (Eq_D > _IQ18(0.97)) Eq_D = _IQ18(0.97);
    if (Eq_D < _IQ18(0.01)) Eq_D = _IQ18(0.01);

    Eq_1 = Eq_0;
    Eq_0 = _IQ18mpyIQX(kpp_E, 28, Eq_D, 18);
    Eq_0 = Eq_0 - _IQ18mpyIQX(kii_E, 28, Eq_D1, 18);
    Eq_0 = Eq_0 - _IQ18mpyIQX(kdd_E, 28, Eq_1, 18);
    if (Eq_0 > _IQ18(0.97)) Eq_0 = _IQ18(0.97);
    if (Eq_0 < _IQ18(0.01)) Eq_0 = _IQ18(0.01);

    /* for test */
    //Ed_0 = _IQ18(0.5);
    //Eq_0 = _IQ18(0);

    if (Ready == 1)
    {
        if
        ((I_E_d > _IQ18(1.0)) && (graph_index_flag == 0))

        {graph_index_flag = 1; data_shift_times = graph_index;}
    }

    /******
    /* NAME: SVPWM_7S() */
    /******

    void SVPWM_VSR(void)
    {
        _iq28 T1_sin, T2_sin;
        Uint16 comp1, comp2, comp3, comp4;

        Mmag = _IQ18mag(V_q_O, V_d_O); //PU value
        if (Mmag > _IQ18(0.98)) Mmag = _IQ18(0.98);
        Mmag = Mmag * (Ts >> 2); //count value of every half
        PWM period

        Theta = _IQ28atan2(V_q_O, V_d_O);
        Theta += Theta_phase; //Phase for SVPWM

        if (Theta > _IQ28(2*PI))
        Theta = Theta - _IQ28(2*PI);
        if (Theta < 0)
        Theta = Theta + _IQ28(2*PI);

```

```

section_number=Theta/_IQ28(PI/3);
Theta_sec= Theta-
_IQ28mpyl32(_IQ28(PI/3),(long)(section_number));

T1_sin=_IQ28sin(_IQ28(PI/3)-Theta_sec);
T2_sin=_IQ28sin(Theta_sec);
T1=_IQ18mpylQX(T1_sin, 28, Mmag,
18)>>16;// T1/2
T2=_IQ18mpylQX(T2_sin, 28, Mmag, 18)>>16;//
T2/2
T0=Ts-T1-T2;//// T0/2
comp1=T0>>1;
comp2=comp1+T1;
comp4=comp1+T2;
comp3=comp2+T2;
switch (section_number)
{
case 0:

EvbRegs.CMPR4 = comp1;
EvbRegs.CMPR5 = comp2;
EvbRegs.CMPR6 = comp3;
break;

case 1:

EvbRegs.CMPR4 = comp4;
EvbRegs.CMPR5 = comp1;
EvbRegs.CMPR6 = comp3;
break;

case 2:

EvbRegs.CMPR4 = comp3;
EvbRegs.CMPR5 = comp1;
EvbRegs.CMPR6 = comp2;
break;

case 3:

EvbRegs.CMPR4 = comp3;
EvbRegs.CMPR5 = comp4;
EvbRegs.CMPR6 = comp1;
break;

case 4:

EvbRegs.CMPR4 = comp2;
EvbRegs.CMPR5 = comp3;
EvbRegs.CMPR6 = comp1;
break;

case 5:

EvbRegs.CMPR4 = comp1;
EvbRegs.CMPR5 = comp3;
EvbRegs.CMPR6 = comp4;
break;

}
}

```

```

void SVPWM_VSI(void)
{
_iq28 T1_sin,T2_sin;
Uint16 comp1,comp2,comp3,comp4;

Mmag1=_IQ18mag(Eq_0,Ed_0);//PU value
if (Mmag1> _IQ18(0.98)) Mmag1=_IQ18(0.98);
Mmag1= Mmag1*(Ts>>2);//count value of every half
PWM period

Theta1=_IQ28atan2(Eq_0,Ed_0);
Theta1+=Theta_INV;//Phase for SVPWM

if ( Theta1 > _IQ28(2*PI) )
Theta1 = Theta1 - _IQ28(2*PI);
if ( Theta1 < 0 )
Theta1 = Theta1 + _IQ28(2*PI);

section_number=Theta1/_IQ28(PI/3);
Theta_sec= Theta1-
_IQ28mpyl32(_IQ28(PI/3),(long)(section_number));

T1_sin=_IQ28sin(_IQ28(PI/3)-Theta_sec);
T2_sin=_IQ28sin(Theta_sec);
T1=_IQ18mpylQX(T1_sin, 28, Mmag1,
18)>>16;// T1/2
T2=_IQ18mpylQX(T2_sin, 28, Mmag1, 18)>>16;//
T2/2
T0=Ts-T1-T2;//// T0/2
comp1=T0>>1;
comp2=comp1+T1;
comp4=comp1+T2;
comp3=comp2+T2;
switch (section_number)
{
case 0:

EvaRegs.CMPR3 = comp1-Tc;
EvaRegs.CMPR2 = comp2-Tc;
EvaRegs.CMPR1 = comp3-Tc;
break;

case 1:

EvaRegs.CMPR3 = comp4-Tc;
EvaRegs.CMPR2 = comp1-Tc;
EvaRegs.CMPR1 = comp3-Tc;
break;

case 2:

EvaRegs.CMPR3 = comp3-Tc;
EvaRegs.CMPR2 = comp1-Tc;
EvaRegs.CMPR1 = comp2-Tc;
break;

case 3:

```

```

    EvaRegs.CMPR3 = comp3-Tc;
    EvaRegs.CMPR2 = comp4-Tc;
    EvaRegs.CMPR1 = comp1-Tc;
    break;
case 4:

    EvaRegs.CMPR3 = comp2-Tc;
    EvaRegs.CMPR2 = comp3-Tc;
    EvaRegs.CMPR1 = comp1-Tc;
    break;
case 5:

    EvaRegs.CMPR3 = comp1-Tc;
    EvaRegs.CMPR2 = comp3-Tc;
    EvaRegs.CMPR1 = comp4-Tc;
    break;
}
}

/*****
/* NAME:
Enable_Rec(),Disable_Rec(),Enable_Inv(),Disable_Inv()
*/
/* DESCRIPTION: Enable or disable PWM output*/
*****/

void Enable_Rec(void)
{
    EvbRegs.COMCONB.bit.FCOMPOE=1;
} /*End of Enable_Rec*/

void Disable_Rec(void)
{
    EvbRegs.COMCONB.bit.FCOMPOE=0;
}

void Enable_Inv(void)
{
    EvaRegs.CMPR1 = 0x0000;
    EvaRegs.CMPR2 = 0x0000;
    EvaRegs.CMPR3 = 0x0000;
    EvaRegs.COMCONA.bit.FCOMPOE=1;
}

void Disable_Inv(void)
{
    EvaRegs.COMCONA.bit.FCOMPOE=0;
}

/*****
/* NAME:    Close_Relay(), Open_Relay*/
/* DESCRIPTION: Open and close relay for protection
*/
*****/

void Close_Relay(void)
{
    Uint16 i;

```

```

    GpioDataRegs.GPBCLEAR.bit.GPIOB12 = 1;//grid
side charging resistor by-pass relay open
    GpioDataRegs.GPACLEAR.bit.GPIOA7 = 1;//motor
side charging resistor by-pass relay open
    Delay_mili_second(400);
    GpioDataRegs.GPBDAT.bit.GPIOB7 = 1;//grid side
power relay close
    for (i=0; i<10; i++)
    {
        Delay_mili_second(100);
        LED_display();
    }
    GpioDataRegs.GPBDAT.bit.GPIOB12 = 1;//grid side
charging resistor by-pass relay close
    GpioDataRegs.GPADAT.bit.GPIOA7 = 1;//motor side
charging resistor by-pass relay close
    CONTACTOR = 1;
}

void Close_Relay1(void)
{
    Uint16 j;

    GpioDataRegs.GPBDAT.bit.GPIOB11 = 1;//inverter
side power relay close
    for (j=0; j<10; j++)
    {
        Delay_mili_second(100);
        LED_display();
    }

    CONTACTOR1 = 1;
}

void Open_Relay(void)
{
    GpioDataRegs.GPBCLEAR.bit.GPIOB7 = 1;//grid side
power relay open
    CONTACTOR = 0;
}

void Open_Relay1(void)
{
    GpioDataRegs.GPBDAT.bit.GPIOB11 = 0;//inverter
side power relay open
    CONTACTOR1 = 0;
}

/*****
/* NAME:    Key_input() */
/* DESCRIPTION: LCD display and push button
command */
*****/

void Key_input(void)
{

```

```

// key_value = *KEY_IN;

if(key_value)
{
// key_monitor++;
// if(key_monitor==300)
// {
// key_monitor1++;key_monitor=0;
// }
// if (key_monitor1==1000)
// {
// key_monitor1 = 0;
// key_monitor = 0;
// *KEY_IN |= 0xf;

switch (key_value)
{

case 1:
if (CONTACTOR==0)
{ SCREEN=1;Close_Relay
();*FAN_CONFIG = 0x1F;}
CONTACTOR=1;
break;

case 2:
if (CONTACTOR1==0)
{ SCREEN=2;Close_Relay1
();*FAN_CONFIG = 0x1F;}
CONTACTOR1=1;
break;

case 3:
if (CONTACTOR==1)
{ MODE=1;
Enable_Rec();}
break;

case 4:
//if (CONTACTOR==1)
//{ Enable_Rec();}

if (CONTACTOR1==1)
{ Enable_Inv();}
MODE=2;
break;

case 5:
switch (MODE)
{
case 0:
key1++; if
(key1>9){key1=0;}
break;
case 1:
key1++; if (key1>9){key1=0;}

```

```

break;
case 2:
//Iarm_ref= Iarm_ref+ _IQ18(0.5);
//if (Iarm_ref> _IQ18(20))
Iarm_ref= _IQ18(20);
// Speed_w_ref=Speed_w_ref+ _IQ18(5);
//if (Speed_w_ref> _IQ18(180)) Speed_w_ref=
_IQ18(180);
break;
}

break;

case 6:
switch (MODE)
{
case 0:
key1--;
if
(key1<0||key1>9){key1=9;}
break;
case 1:
key1--;
if
(key1<0||key1>9){key1=9;}
break;
case 2:
//Iarm_ref= Iarm_ref- _IQ18(0.5);
//if (Iarm_ref< _IQ18(-5))
Iarm_ref= _IQ18(-5);
//Speed_w_ref=Speed_w_ref- _IQ18(5);
//if (Speed_w_ref< _IQ18(0.1)) Speed_w_ref=
_IQ18(0.1);
break;
}

break;

case 7:
if (CONTACTOR==1)
{
Disable_Rec(); MODE =
0;Open_Relay (); SCREEN=0;
test_flag1=0;
Delay_mili_second(500);}

//if (data_shift_times>=2700)
data_shift_times=data_shift_times-2700;
//if (data_shift_times<2700)
data_shift_times=data_shift_times+6300;
//
Buffer_rotating(graph_data,data_shift_times);
// Buffer_rotating(graph_data1,data_shift_times);
//Buffer_rotating(graph_data2,data_shift_times);
//Buffer_rotating(graph_data3,data_shift_times);
//Buffer_rotating(graph_data4,data_shift_times);
// Buffer_rotating(graph_data5,data_shift_times);
Manu_display();*FAN_CONFIG = 0x1D;

```



```

        break;

case 8:
    if (CONTACTOR1==1)
        {Disable_Inv(); Open_Relay1
    );
        if (CONTACTOR==1)
            MODE=1;
            else MODE=0;
            Delay_mili_second(500);}

        break;
    }
}
key_value=0;//for control without keypad
} /*End of Key_input*/

/*****
/* NAME:    AD_bias()*/
/* DESCRIPTION: ADC dc bias calibration */
*****/

int AD_bias (unsigned int channel)
{
    int32 bias_data=0;
    Uint32 i;

    bias_data=0;
    for (i=0;i<65536;i++){
        while
(AdcRegs.ADCST.bit.INT_SEQ1== 0){}
        AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;    //
        Clear INT_SEQ1 bit
        switch (channel) {
            case 0:

                bias_data+=AdcRegs.ADCRESULT0>>4;
                    break;
                case 1:

                bias_data+=AdcRegs.ADCRESULT1>>4;
                    break;
                case 2:

                bias_data+=AdcRegs.ADCRESULT2>>4;
                    break;
                case 3:

                bias_data+=AdcRegs.ADCRESULT3>>4;
                    break;
                case 4:

                bias_data+=AdcRegs.ADCRESULT4>>4;
                    break;

```

```

                case 5:

                bias_data+=AdcRegs.ADCRESULT5>>4;
                    break;
                case 6:

                bias_data+=AdcRegs.ADCRESULT6>>4;
                    break;
                case 7:

                bias_data+=AdcRegs.ADCRESULT7>>4;
                    break;
                case 8:

                bias_data+=AdcRegs.ADCRESULT8>>4;
                    break;
                case 9:

                bias_data+=AdcRegs.ADCRESULT9>>4;
                    break;
                case 10:

                bias_data+=AdcRegs.ADCRESULT10>>4;
                    break;
                case 11:

                bias_data+=AdcRegs.ADCRESULT11>>4;
                    break;
                case 12:

                bias_data+=AdcRegs.ADCRESULT12>>4;
                    break;
                case 13:

                bias_data+=AdcRegs.ADCRESULT13>>4;
                    break;
                case 14:

                bias_data+=AdcRegs.ADCRESULT14>>4;
                    break;
                case 15:
                default:

                bias_data+=AdcRegs.ADCRESULT15>>4;
                    break;
            }
        }
        bias_data=bias_data>>16;
        return bias_data;
    } /*End of AD_bias*/

/*****
/* NAME:    AD_bias16() */
/* DESCRIPTION: LCD display and ADC dc bias
calibration */

```

```
/******
```

```
void AD_bias16(void)
```

```
{
    Text_LCD_clear_display();
    Text_LCD_printxy(0,1," Test the AD Bias");

    Text_LCD_printxy(0,2," Calculating CH 0");
    AD_BIAS0=AD_bias(0);//2059

    Text_LCD_printxy(0,2," Calculating CH 1");
    AD_BIAS1=AD_bias(1);//2066

    Text_LCD_printxy(0,2," Calculating CH 2");
    AD_BIAS2=AD_bias(2);//2053

    Text_LCD_printxy(0,2," Calculating CH 3");
    AD_BIAS3=AD_bias(3);//2059

    Text_LCD_printxy(0,2," Calculating CH 4");
    AD_BIAS4=AD_bias(4);//2059

    Text_LCD_printxy(0,2," Calculating CH 5");
    AD_BIAS5=2046;//AD_bias(5);

    Text_LCD_printxy(0,2," Calculating CH 6");
    AD_BIAS6=2049;//AD_bias(6);

    Text_LCD_printxy(0,2," Calculating CH 7");
    AD_BIAS7=AD_bias(7);//2047

    Text_LCD_printxy(0,2," Calculating CH 8");
    AD_BIAS8=AD_bias(8);//2050

    Text_LCD_printxy(0,2," Calculating CH 9");
    AD_BIAS9=AD_bias(9);//2055

    Text_LCD_printxy(0,2," Calculating CH 10");
    AD_BIAS10=AD_bias(10);//2051

    Text_LCD_printxy(0,2," Calculating CH 11");
    AD_BIAS11=AD_bias(11);//2047

    /*Text_LCD_printxy(0,2," Calculating CH 12");
    AD_BIAS12=AD_bias(12);

    Text_LCD_printxy(0,2," Calculating CH 13");
    AD_BIAS13=AD_bias(13);

    Text_LCD_printxy(0,2," Calculating CH 14");
    AD_BIAS14=AD_bias(14);

    Text_LCD_printxy(0,2," Calculating CH 15");
    AD_BIAS15=AD_bias(15);*/

    Text_LCD_printxy(0,2," Calculation DONE ");
```

```
    Delay_mili_second(100);
```

```
} /*End of AD_bias16*/
```

```
void Delay_mili_second (int delay)
```

```
{
    Uint16 i,j;
    i=delay*30;//150M Hz DSP clock
    for (j=0; j<1000; j++)
    {
        while (i){i--};
        i=delay*30;
    }
}
```

```
/******
```

```
/* NAME:      Init_variable()      */
```

```
/* DESCRIPTION: Initialization of
system control variables      */
```

```
/******
```

```
void Init_variable(void)
```

```
{
    /*ADC*/
    ConversionCount=0;

    /*PLL */
    Theta_con = 0;
    Theta_err = 0;
    nmax = 0;
    Theta_pu = 0; //phase in per unit value
    Theta_line = 0; //phase in radian value (0-2pi)
    angle_offset = 0;
```

```
/*Speed up*/
```

```
Theta_INV = 0;
f_counter= 0;
f= 0;
F= 0;
fL=0;
count_f=0;
```

```
/*Low pass filter*/
```

```
LPFa_DC=_IQ28(1-2.0/(2.0+w_DC/Sw_freq));
LPFb_DC=_IQ28(1)-(LPFa_DC<<1);
```

```
/*PI controller*/
```

```
kp_dc=_IQ28(Kp_Udc); //kp=0.5
ki_dc=_IQ28(Ki_Udc/Sw_freq);//ki/fs=45/9k=0.005
kp_I=_IQ28(Kp_I); //kp=0.04
ki_I=_IQ28(Ki_I/Sw_freq);//ki/fs=30/9k=0.0033
```

```
Vdc_ref=_IQ18(Udc_ref);
```

```

Vdc_err =0;
Vdc_err1 =0;

I_d_ref= 0;
I_d_err = 0;
I_d_err1= 0;

I_q_ref=0;//_IQ18(-0.3);
I_q_err =0;
I_q_err1 =0;

V_d_O=_IQ18(0.8);
V_q_O=0;

/*VSI LEAD-LAG controller*/
Ed_ref=0;

kp_E=_IQ28(Kp_E);
ki_E=_IQ28(Ki_E);

kpp_E=_IQ28(Kpp_E);//kpp=4.74
kii_E=_IQ28(Kii_E);//kii=3.735
kdd_E=_IQ28(Kdd_E);//kdd=0.0058

Ed_err=0; Ed_err0=0; Ed_err1=0; Ed_D=0;Ed_0=0;
Ed_D1=0;Ed_I=0;
Eq_err=0; Eq_err0=0; Eq_err1=0; Eq_D=0;Eq_0=0;
Eq_D1=0;Eq_I=0;

Tu=26614; Tw=17614;

/* Key input*/
key_value = 0;
key_monitor = 0;
key_monitor1 = 0;
key1=7;

/*system flags*/
CONTACTOR1= 0;
CONTACTOR = 0;
MODE= 0;
FAULT = 0;
PLL = 0;
Initial_over=0;

SCREEN = 0;
display_counter=0;
display_counter1=0;
} /*End of Init_variable*/

/*****/
/* NAME:      Gpio_select()          */
/* DESCRIPTION: GPIO configuration  */
/* initialization */
/*****/

```

```

void InitGpio(void)
{
    // sets GPIO Muxs as I/Os
    EALLOW;

    GpioMuxRegs.GPAMUX.all= 0x077F; // Configure
MUXs as digital I/Os or
    GpioMuxRegs.GPBMUX.all= 0x077F; // peripheral
I/Os
    GpioMuxRegs.GPDMUX.all=0x0000;
    GpioMuxRegs.GPEMUX.all=0x0000;
    GpioMuxRegs.GPFMUX.all=0x0030;
    GpioMuxRegs.GPGMUX.all=0x0030;

    GpioMuxRegs.GPADIR.all=0xF8FF; //
    GpioMuxRegs.GPBDIR.all=0xF8FF; // GPIO DIR
select GPIOs as output or input
    GpioMuxRegs.GPDDIR.all=0x0000;
    GpioMuxRegs.GPEDIR.all=0x0000;
    GpioMuxRegs.GPFDIR.all=0x3FDF;
    GpioMuxRegs.GPGDIR.all= 0x0010;

    GpioMuxRegs.GPAQUAL.all=0x0000; // Set GPIO
input qualifier values
    GpioMuxRegs.GPBQUAL.all=0x0000;
    GpioMuxRegs.GPDQUAL.all=0x000F;
    GpioMuxRegs.GPEQUAL.all=0x000F;

    EDIS;
} /*End of Gpio_select*/

/*****/
/* NAME:      InitXintfClocks(),InitPeripherals()*/
/* DESCRIPTION: System configuration */
/* NOTES: */
/*****/

void InitXintfClocks(void)
{
    XintfRegs.XTIMING0.all=0x000358AC;
    XintfRegs.XTIMING6.all=0x0003E746;//3D4A5;
    XintfRegs.XINTCNF2.all=0x00000007;
} /*End of InitXintfClocks*/

void InitPeripherals(void)
{
    //Text_LCD_initiate();
    InitAdc();
    InitEVA();
    InitEVB();
    init_fpga();
    Init_variable();
}

```

```

void InitEVA(void)
{
    // Configure EVA
    // EVA Clock is already enabled in InitSysCtrl();

    EvaRegs.EVAIMRA.bit.PDPINTA=1;//PDPINTA
    protection enable
    // Enable compare for PWM1-PWM6
    EvaRegs.CMPR1 = 0x0000;
    EvaRegs.CMPR2 = 0x0000;
    EvaRegs.CMPR3 = 0x0000;
    // Compare action control. Action that takes place
    EvaRegs.ACTRA.all = 0x0666;// active high
    EvaRegs.DBTCNA.all = 0x0FF4; // Enable deadband
    3.4uS
    EvaRegs.COMCONA.all = 0xA000;//update twice

    /* Initialize QEP circuits*/
    EvaRegs.CAPCONA.all = 0x9004;//disable capture
    1,2;select timer2;detect rising edge for capture 3.

    /* Initialize EVA Timer1*/
    EvaRegs.T1PR = Ts;// Timer1 period (for motor side
    switching frequency)
    EvaRegs.T1CNT = 0x0000; // Timer1 counter
    // Timer enable
    EvaRegs.T1CON.all = 0x0800; // (updowncount mode)
    //input clock(HSPCLK)/1
    //EvaRegs.EVAIMRA.bit.T1UFINT=1; //Enable
    Timer1 underflow interrupt

    /* Initialize EVA Timer2*/
    EvaRegs.T2CNT = 0x0000;
    EvaRegs.T2PR = 14399; // period register (for 3600*4
    pules every cycle of speed sensor)
    EvaRegs.T2CON.all = 0x1870; //(directional-
    u/dpowncount mode)input clock from QEP

    EvaRegs.T1CON.bit.TENABLE=1;//Timer enable
}

void InitEVB(void)
{
    // Configure EVA
    // EVA Clock is already enabled in InitSysCtrl();
    EvbRegs.EVBIMRA.bit.PDPINTB=1;//PDPINTA
    protection enable
    // Enable compare for PWM7-PWM12

    EvbRegs.CMPR4 = 0x0000;
    EvbRegs.CMPR5 = 0x0000;
    EvbRegs.CMPR6 = 0x0000;
    // Compare action control. Action that takes place
    // on a cmpare event
    EvbRegs.ACTRB.all = 0x0666;//active high

    EvbRegs.DBTCNB.all = 0x0FF4; // Enable deadband
    3.4uS
    //EvbRegs.COMCONB.all = 0x9000;//five segments
    EvbRegs.COMCONB.all = 0xA000;//0xA000;//seven
    segments and update twice

    /* Initialize the timers*/
    /* Initialize EVB Timer3*/

    EvbRegs.T3PR = Ts;// Timer3 PWM period(for
    computing fre of 8997Hz which is twice of rectifier
    switching frequency)
    EvbRegs.T3CMPR = Ts>>1; // Timer3 compare
    EvbRegs.T3CNT = 0x0000; // Timer3 counter
    // TMODE = continuous up/down
    // Timer enable
    EvbRegs.T3CON.all = 0x0802; // (updowncount mode)
    //input clock(HSPCLK)/1//timer compare enable
    EvbRegs.EVBIMRA.bit.T3CINT=1; //Enable Timer3
    compare interrupt

    /* Initialize EVB Timer4*/
    EvbRegs.T4CNT = 0x0000;
    EvbRegs.T4PR = Ts>>4; // Setup period register (for
    sampling feq of 8997Hz*8)
    EvbRegs.GPTCONB.bit.T4TOADC = 1; // Enable
    EVASOC in EVB (underflow starts ADC)
    EvbRegs.T4CON.all = 0x0880; // (updowncount
    mode) //input clock(HSPCLK)/1//start with T3

    EvbRegs.T3CON.bit.TENABLE=1;//Timer enable

    /* Initialize EVB Capture4&5*/
    EvbRegs.CAPCONB.all=0xA6F0;
    EvbRegs.EVBIMRC.all=0x0003;
    EvbRegs.EVBIFRC.all=0x0007;
}

/*****
/* NAME: InitAdc() */
/* RETURNS: void */
/* DESCRIPTION: ADC configuration, use Event
Manager A to start ADC */
/* NOTES: */
*****/

void InitAdc(void)
{
    /*Start ADC*/
    AdcRegs.ADCTRL3.bit.ADCBGRFDN = 0x3;
    // Power up bandgap/reference circuitry
    Delay_micro_second (ADC_usDELAY);
    // Delay before powering up rest of ADC
    AdcRegs.ADCTRL3.bit.ADCPWDN = 1;
    // Power up rest of ADC
    Delay_micro_second (ADC_usDELAY2);
}

```

```

/*Configure ADC*/
AdcRegs.ADCTRL1.bit.ACQ_PS = 0x1; //S/H width in
ADC module periods = 2 ADC clocks
AdcRegs.ADCTRL3.bit.ADCCLKPS = 0x5; //ADC
module clock = HSPCLK/5 = 25MHz
AdcRegs.ADCTRL1.bit.SEQ_CASC = 1; // 1
Cascaded mode
AdcRegs.ADCTRL1.bit.CONT_RUN = 0; //not
Continuous running
// AdcRegs.ADCTRL1.bit.SEQ_OVRD = 1; //
Enable Sequencer override feature (bit 5)

AdcRegs.ADCMAXCONV.all = 0x000B; // Setup 16
conv's on SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x1;
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x2;
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x3;
AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x4;
AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 0x5;
AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 0x6;
AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 0x7;
AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 0xC;
AdcRegs.ADCCHSELSEQ3.bit.CONV08 = 0xD;
AdcRegs.ADCCHSELSEQ3.bit.CONV09 = 0xE;
AdcRegs.ADCCHSELSEQ3.bit.CONV10 = 0xF;
AdcRegs.ADCCHSELSEQ3.bit.CONV11 = 0x9;
AdcRegs.ADCCHSELSEQ4.bit.CONV12 = 0x8;
AdcRegs.ADCCHSELSEQ4.bit.CONV13 = 0xE;
AdcRegs.ADCCHSELSEQ4.bit.CONV14 = 0xF;
AdcRegs.ADCCHSELSEQ4.bit.CONV15 = 0x9;

//AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 1; //
enable SEQ1 interrupt (every EOS)
AdcRegs.ADCTRL2.bit.EVB_SOC_SEQ = 1; //
enable EVBSOC to start SEQ1

} /*End of InitAdc*/

/*****
/* NAME: init_fpga() */
/* RETURNS: void */
/* DESCRIPTION: Initialization of FPGA registers */
/* NOTES: */
*****/

void init_fpga(void)
{
/*Hardware Protection*/
*FAULT_CLEAR_A = 0x0;
*FAULT_ENABLE_A = 0x1ff;

*FAULT_CLEAR_B = 0x0;
*FAULT_ENABLE_B = 0x1ff;

```

```

/*Thermal Protection (disabled)*/
*FAULT_THERMO_ENABLE = 0x0;

*FAN_PERIOD = 14999;
*FAN_COUNT = 0x0;
*FAN_COMPARE_A = 0x0;
*FAN_COMPARE_B = 12180;
//*FAN_CONFIG = 0x1F;

/*Fault signal distribution*/
*FAULT_DISTRIBUTE = 0xffff;

/*Zero-crossing detection (Zero-crossing starts
automatically)*/
*PHASE_SHIFT_A = 0x4DDc; // 0x4c2c //**shift
input to 90degree delay*/
// 0xb090 /*shift input to
180degree delay*/
//**PHASE_SHIFT_CONFIG_A = 0x0118;/**
divider=24+1,DPLL freq=Fclock/(divider+1); input
Vbc*/
*PHASE_SHIFT_CONFIG_A =
0x0118; //divider=24+1,DPLL
freq=Fclock/(divider+1); Vbc //0x0418 input Vvw
/*IO pin configuration*/
*GPIOA6_CONFIG = 0x0A; /*A side PLL
output*/
*GPIOB6_CONFIG = 0x0B; /*A side SYN
signal*/

/*XINT interrupt configuration*/
*XINT1_CONFIG = 0x0; /*Disable XINT1,
2*/
*XINT2_CONFIG = 0x0;

/*Key input Text LCD displace interface*/
*KEY_IN = 0xf; //clear all the registered key
input

*SWITCH_IN = 0x00;

/*FPGA LED*/
*FPGA_LED = 0x0;
} /*End of Init_fpga*/

void Buffer_rotating(int16 buffer[9000], Uint16
shift_times)
{
int16 a;
Uint16 i,j;

for (i=shift_times; i>0; i--)
{
a=buffer[0];
for (j=0; j<9000; j++)
{buffer[j]=buffer[j+1];}
}

```

```
buffer[8999]=a;  
}  
} /*End of Buffer_rotating*/
```

```
//=====//  
// End of WECS.c //  
//=====//
```