Real-Time Simulation of a Smart Inverter

by

Ramanathan Thiagarajan

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2017 by the
Graduate Supervisory Committee:

Raja Ayyanar, Chair
Qin Lei
Jiangchao Qin

ARIZONA STATE UNIVERSITY

May 2017

ABSTRACT

With the increasing penetration of Photovoltaic inverters, there is a necessity for recent PV inverters to have smart grid support features for increased power system reliability and security. The grid support features include voltage support, active and reactive power control. These support features mean that inverters should have bidirectional power and communication capabilities. The inverter should be able to communicate with the grid utility and other inverter modules.

This thesis studies the real time simulation of smart inverters using PLECS Real Time Box. The real time simulation is performed as a Controller Hardware in the Loop (CHIL) real time simulation. In this thesis, the power stage of the smart inverter is emulated in the PLECS Real Time Box and the controller stage of the inverter is programmed in the Digital Signal Processor (DSP) connected to the real time box. The power stage emulated in the real time box and the controller implemented in the DSP form a closed loop smart inverter.

This smart inverter, with power stage and controller together, is then connected to an OPAL-RT simulator which emulates the power distribution system of the Arizona State University Poly campus. The smart inverter then sends and receives commands to supply power and support the grid. The results of the smart inverter with the PLECS Real time box and the smart inverter connected to an emulated distribution system are discussed under various conditions based on the commands received by the smart inverter.

ACKNOWLEDGMENTS

I express my sincere gratitude to my advisor, Dr. Raja Ayyanar for giving me a chance to work in the Power Electronics lab. His constant encouragement and kindness throughout my graduate studies has guided me through this journey. His passion for teaching and research has inspired me to pursue my thesis in power electronics. This work would not have been the same without his superior insight and knowledge. Gratitude is also due to my graduate thesis committee, comprising Dr. Qin Lei and Dr. Jiangchao Qin for their time and support.

I am grateful to the Department of Energy for funding my first year at graduate school through the Sunshot program for Reliability Studies of Module Level Power Electronics. I am also grateful to Lightworks at Arizona State University for funding my thesis work.

I would like to thank my lab colleagues Siddharth, Ziwei, Yinglai, Chenhao, Tong, Scott, Jinia, Pavan, Nikhil, Arkanatha, and Umit for their support and guidance throughout my stay at the Power Electronics lab. I am grateful to my collaborators, Dr. Jack Flicker and Prof. Govindasamy Tamizhmani, who helped me learn tremendously during the Reliability Studies of Module Level Power Electronics. I would also like to thank my previous advisors, Dr. Ganesan Kaliyaperumal and Dr. C.K. Subramaniam for helping me gain great knowledge on battery electronics. I would be forever indebted to my undergraduate professor, Prof. Ramesh Ramamoorthy for introducing me to the world of Power Electronics.

I am grateful to my late grandfather, Dr. S. Ramanathan, who has inspired me to achieve great things in my life. I am also grateful to my parents, Thiagarajan Ramanathan and Saraswathi Thiagarajan for their love and support. I would like to thank my uncle, K.N. Radhakrishnan and my cousin, Karthik for their continue support. I would also like to thank my aunts, Geetha Bennett, Padma Ramanathan, and Latha Radhakrishnan for their moral support. I would also like to thank my friends Vikram, Vaishnavi, Harinee, and Malathi for their support. My special thanks to my roommates Guru Vignesh, Sriram, Abhishek, and Ranjan for their support and cooperation during my graduate school.

Finally, I would like to thank my close friend, Sangita for her unabated support offered during the last six years. She has always stood by me and supported me during tough times. This journey would not have been possible without her.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Increasing interest in distributed energy resources in recent times can be attributed to technological advances in power conversion, environmental concerns and mandatory renewable energy standards (RES) to meet reduced carbon foot prints. As an inexhaustible and clean source of energy, solar energy has attracted the most among the renewable sources of energy [1], leading to the increased penetration of Photo Voltaic (PV) generation within the distribution system.

This chapter discusses the PV market growth and forecast, PV inverter classification, Distributed Generation, smart inverters, real time simulation and thesis organization.

1.1 PV market growth and forecast

As the electricity demand grows modestly every year, the primary reason for introducing more renewables into the generation can be attributed to Clean Power Plan (CPP) and near-term availability of renewable energy tax credits. The CPP is an administration policy aimed at combating global warming proposed by Environmental Protection Agency (EPA) in June 2014 [2], by planning to lower the carbon dioxide emitted by power generators.

The renewable energy tax credits also known as the investment tax credit (ITC) is an important federal policy regulation to support the deployment of solar energy in the United States. The ITC provides 30% tax credit for solar system installations on residential and commercial properties. This continues to drive growth in the industry and job creation

across the country [3]. The ITC has helped annual solar installation grow by over 1600% since its implementation in 2006, a compound growth of 76% every year. The existing credits of the ITC provides market security for companies to develop long term investments driving competition, which in turn, reduces the cost for the consumers. The recent ITC extension is expected to nearly quadruple solar deployment by the end of 2020.

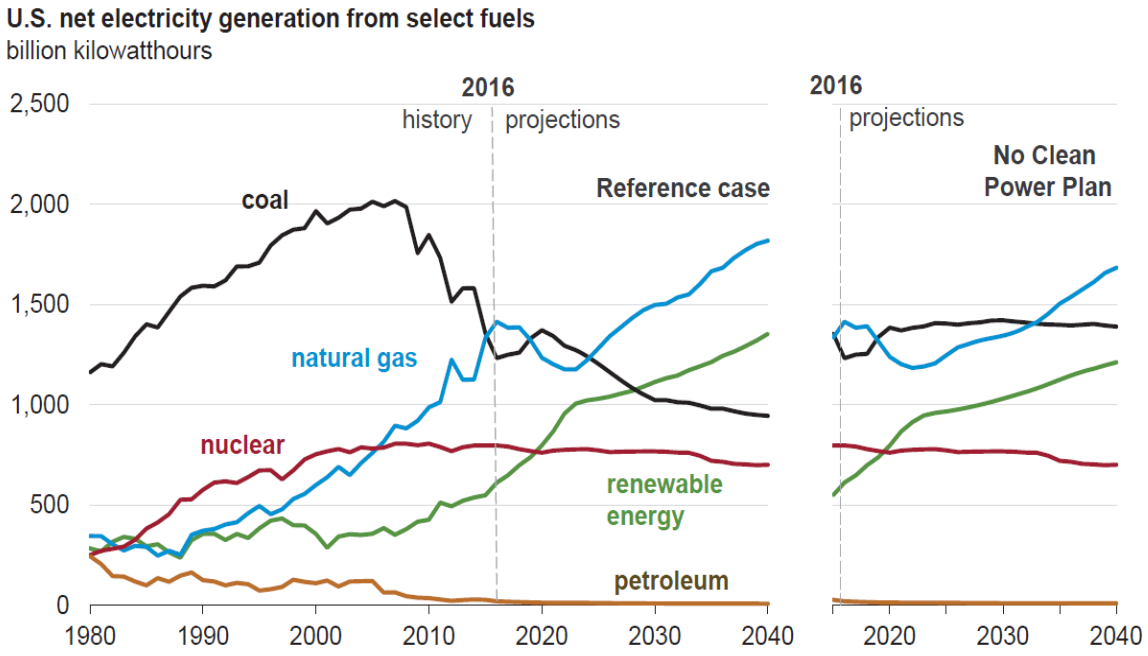

Figure 1.1 – Current and projected energy resource mix with and without CPP [4]

The projection of energy resource mix from year 2016 to 2040 is shown in Figure 1.1, taken from the AEO2017 presentation for the U.S. Energy Information Administration (EIA) [4]. It is mentioned that with natural gas prices rebounding from their 20-year lows which occurred in 2016, coal regains a larger generation share over natural gas through 2020. It is further mentioned that, longer term policy and unfavorable economic conditions compared with natural gas and renewables result in declining coal generation and growing

natural gas and renewables generation in the reference case. This prediction is shown in Figure 1.2.

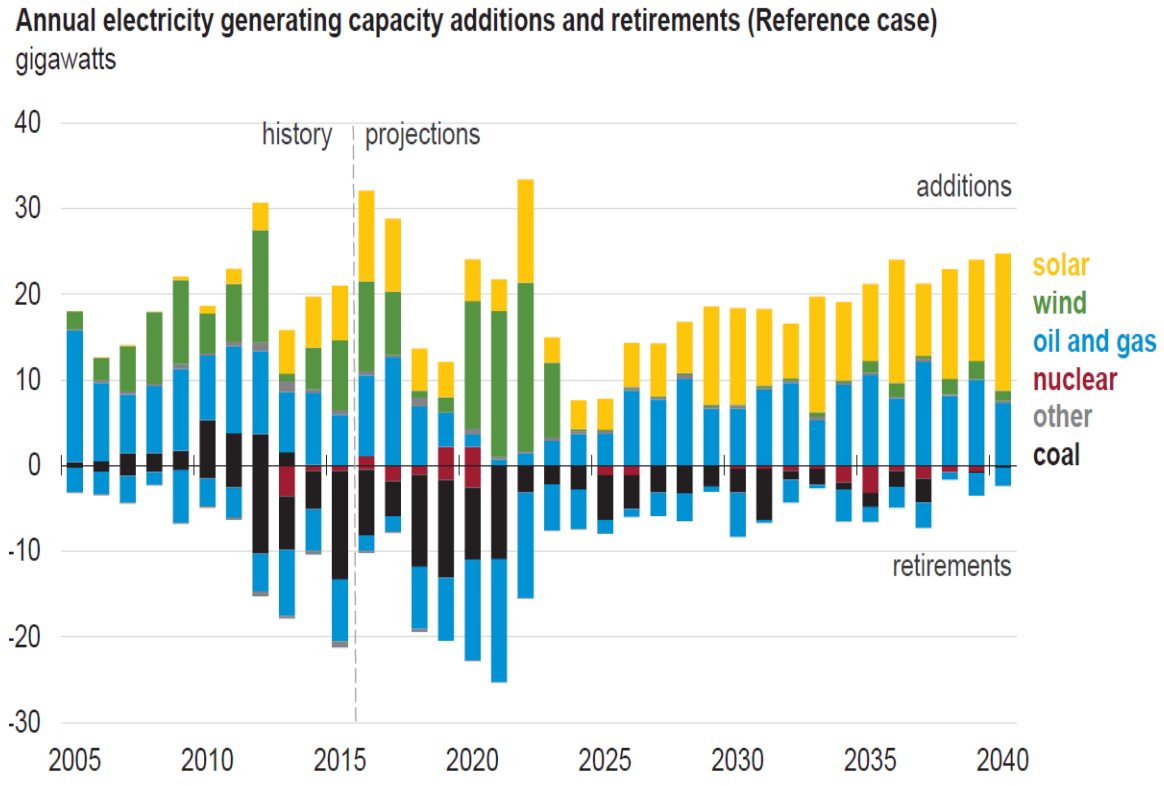

Figure 1.2- Annual electricity generating capacity additions and retirements [4]

The Solar Energy Industries Association (SEIA) has announced in its annual year in review of 2016 [5] with some key figures and data. The US market has installed 14,762 MW_{dc} of solar PV – nearly doubling its capacity installed in 2015. The PV market grew 97% over 2015, with 14.8 GW_{dc} from over 370,000 individual installations. For the first time, solar represented the largest new source of electricity generating capacity, beating out both natural gas and wind for new capacity additions. Figure 1.3 shows the annual U.S solar PV installations from year 2000-2016.

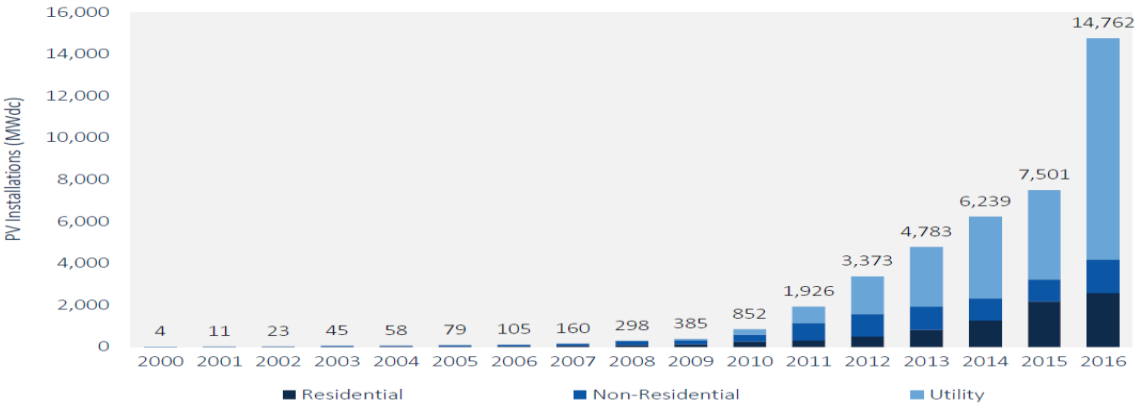

Figure 1.3 - Annual U.S solar PV installations from year 2000-2016 [5]

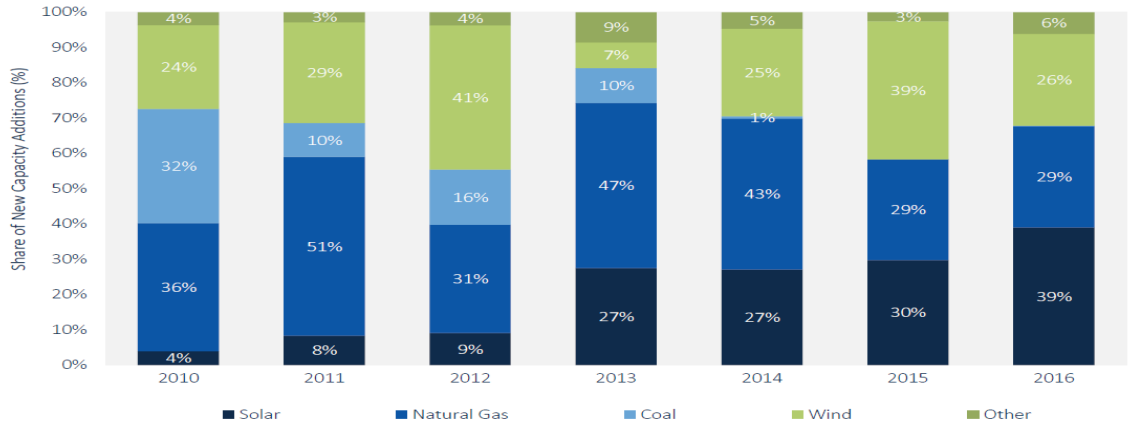

Figure 1.4 - New generation capacity additions from year 2010-2016 [5]

The year 2016 contributed to 39% of new electricity generation capacity in United States. The figure 1.4 shows the new generation capacity additions from year 2010-2016. With increase in PV penetration in residential, utility and non-commercial applications. The discussion will proceed further into the classification of PV inverters.

1.2 Inverter classification

Inverters can be classified based on multiple criteria like power levels, number of power stages, number of phases, grid connection and transformer isolation [6]. Inverters are

classified into single-phase and three-phase inverters based on the number of phases. Based on grid connection, inverters are classified into grid connected inverters and stand-alone inverters. Grid connected inverters are used typically for feeding power to the grid and stand-alone inverters are used for Unlimited Power Supply (UPS) and motor drive applications. The discussion here will be limited only to single phase PV inverters.

Based on the power levels, inverters are classified into three types. The three types classified based on power level are micro inverters, string inverters and central inverters. Micro inverters have a power rating of less than 500 Watts [7] with a typical input range of 9-40 volts. Each micro inverter is mounted at the back of a single PV panel. Since each inverter has only one PV panel, it has a higher energy yield compared to that of the string inverters. These inverters are connected in a series parallel modular system to achieve higher power rating. Figure 1.5 shows the block diagram of a two stage micro inverter.

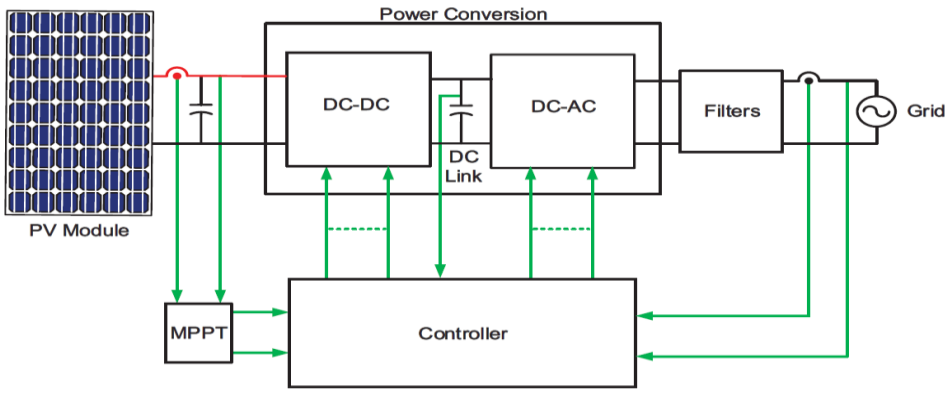

Figure 1.5 - Block diagram of a two stage micro inverter

Micro inverters are typically connected to the single phase 240V AC grid. For converting a PV input voltage between 9-40V to an AC voltage of 240V, two power stages are required.

The first stage is a DC-DC converter which boosts an input voltage of 9V to 40V to an output voltage of 400V to 450V. The DC-DC stage can be a transformer isolation based converter or a transformer-less converter. The DC link capacitor is maintained at a voltage near to that of the output voltage of DC-DC converter. The second stage is a DC-AC inverter circuit with filters at the grid side. The advantages of micro inverters are its increased modularity, input efficiency and reliability [8]. However, the cost is higher compared to string inverters and central inverters to achieve the same power rating.

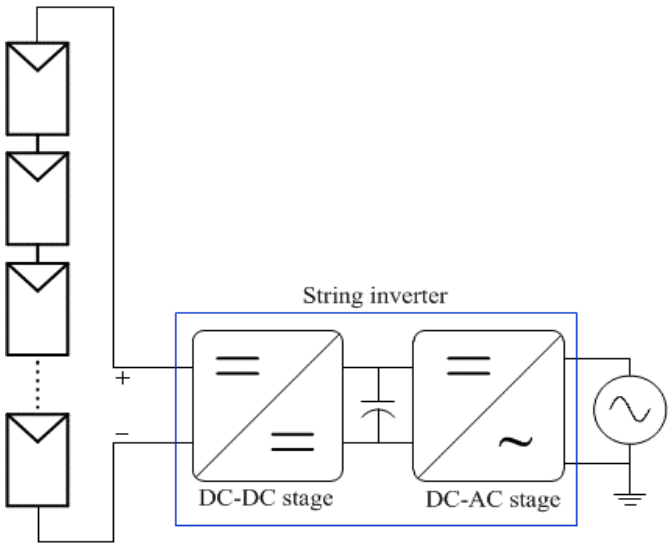

Figure 1.6 – Block diagram of string inverter

The block diagram of a string inverter is shown in Figure 1.6. String inverters have a power rating between 2 KW and 6 KW. The term string denotes series connection of PV panels to achieve higher input voltages. The input voltage range is typically between 250V to 550V. String inverters also have two power stages, one DC-DC stage and one DC-AC stage. The DC-DC stage can be a transformer based or transformer-less converter. The transformer based converter is further classified into low/line frequency transformer

converter and high frequency transformer converter. Transformer-less converter configurations were initially not allowed for safety purposes. However, with recent advances in transformer-less designs, they are gaining acceptance with changes in electric codes and interconnection standards. String inverters are most commonly used among the three configurations. Its advantages include simpler system design and simpler fault finding [10].

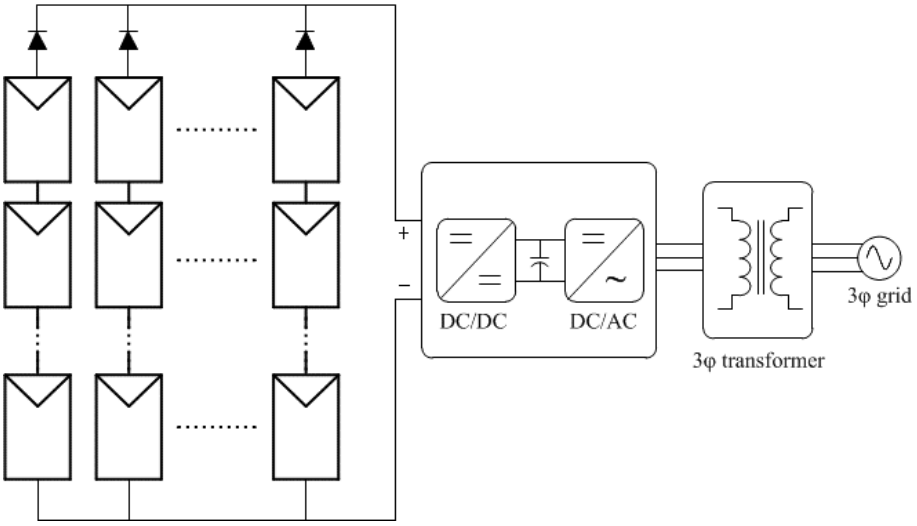

Figure 1.7 – Block diagram of a central inverter

Central inverters are used in utility scale and large commercial scale applications. These inverters are employed in large solar fields where a large number of PV panels are connected in series-parallel configuration for achieving very high power (250 KW to greater than 1MW). Though the array level Maximum Power Point Tracking (MPPT) is a drawback for central inverter, the losses are not very significant when a large, dedicated solar field is considered. Due to the use of a low frequency transformer, the losses in a central inverter are minimized and have an efficiency greater than 98%. Due to the high

power rating, central inverter has grid support features [10] for Volt/VAR control, active power control, frequency and inertial support and fault recovery support.

Central inverters are typically three phase inverters and they have two power stages, a DC-DC stage followed by a DC-AC stage and use interleaved boost DC-DC converters for the DC-DC stage. The interleaved boost DC-DC converter can have input voltage from a single PV array or multiple DC voltage inputs. In case of multiple PV panel, each PV panel is connected to one phase of the interleaved boost converter. This is shown in Figure 1.8. The central inverter can also be implemented using multi-level converters [11] [12].

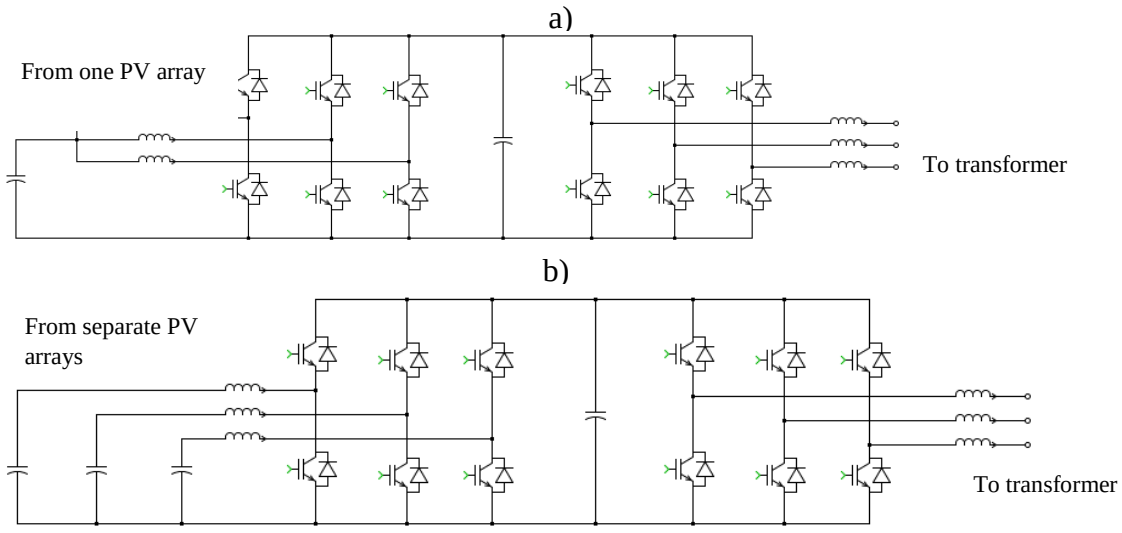

Figure 1.8 – a) interleaved boost DC-DC with single PV array input voltage b) interleaved boost DC-DC with separate PV arrays

1.3 Distributed Generation

Distributed generation (DG) refers to the generation of electric power at the point of consumption. DG has a power rating of 10MW or less and they are interconnected at

different levels (customer, substation, or distribution feeder) in the power system. DG technologies can run on waste heat, renewable energy resources such as PV and wind energy and fossil fuels.

The high penetration of DG in the distribution systems have a lot of positive impacts [1] [13] [14]. The increased penetration of DG reduces the voltage sag improving the power quality. It also reduces the power loss. It also increases the distribution and transmission capacity. With more capacity, it also increases system reliability with DG acting as emergency backup during power outage.

The potential impacts of DG on distribution system include voltage regulation [15] - [18], power system losses [19] [20], voltage flicker [20], harmonics [21] – [24], increased level of fault current [25] [26] and impact on protection coordination [27] [28].

With these impacts, there is a necessity for PV inverters to include smartness for grid support features for supply reliability and security. These grid support features include Voltage/ Voltage Ampere Reactive power (VAR) support, active power control, frequency and inertial support, low-voltage and fault ride-through, fault recovery support and energy forecasting.

1.4 Smart inverter

With the increasing penetration of DG in the distribution system, DG systems are required to provide advanced grid functionality [29]. These functionalities are aimed at supporting grid during low / high voltage events, reactive power generation and low / high frequency ride through (LVRT /HVRT) as per the revised standards. [30] – [34]. This means inverters should have bidirectional power and communication capabilities.

These inverters should also have a scalable software platform incorporating a sophisticated performance monitoring capability. A smart inverter must be able to send and receive messages quickly, as well as share data with the owner, utility and other smart inverters. Such systems allow PV inverter installers and service technicians to diagnose operational and maintenance issues including predicting possible inverter or module problems. The technicians should also be able to remotely upgrade certain parameters in moments. These inverters must also include application-programming-interface (API) functionality that provides fleet owners and other partners a way to tie in their own software to create powerful enterprise-level tools. [35]

There are certain standards to which inverter manufacturers must adhere. Among these standards, IEC 61850 [36] is the important standard regarding smartness of the inverter. IEC 61850 is a standard for vendor-agnostic engineering of the configuration of Intelligent Electronic Devices for electrical substation automation systems to be able to communicate with each other. IEC 61850 is a part of the International Electrotechnical Commission's (IEC) Technical Committee 57 (TC57) reference architecture for electric power systems. The abstract data models defined in IEC 61850 can be mapped to a number of protocols. These protocols can run over TCP/IP networks or substation LANs using high speed switched Ethernet to obtain the necessary response times below four milliseconds for protective relaying.

Regarding direct control of an inverter’s power generation capability is concerned, there is a concern that only authorized personnel or entities are allowed to make changes to the operating settings. Hence, there is a necessity for secure communication [37] which includes authentication. Mod bus protocol is the preferred communication protocol

between the user, utility and the service personnel. The Mod bus standard does not include any mechanism to login or authenticate a master to a slave, or a slave to a master and has historically relied on physical security. However, there are common, commercial-off-the-shelf technologies and methods which can be used with Modbus/TCP connections:

- The inverter can have 'Permitted Host' lists, preventing control actions from unauthorized IP addresses.
- The inverter can limit control actions to Modbus/TCP clients connecting with SSL/TLS, which can include certificate-based authentication.
- Many cellular and broadband gateways include VPN end-point functionality, which allows remote host to securely tunnel standard Modbus/TCP via VPN link to remote sites.

1.5 Real time simulation

Real time simulation is the reproduction of output (voltage/current) waveforms that are representative of the behavior of the real systems being modelled. Digital Real-Time Simulation (DRTS) applied to the power systems can be classified into two categories:

1. Fully digital real-time simulation (model in the loop, software in the loop (SIL), or processor in the loop (PIL))
2. Hardware in the loop (HIL) real-time simulation

A fully digital real-time simulation has the entire system (control, protection, inputs, outputs and other accessories) to be modeled inside the real time simulator and does not involve external interfacing or analog and digital inputs/outputs (I/O). On the other hand, in the HIL simulation, parts of the digital real-time simulation have been replaced with

actual physical components. In the HIL simulation model, the device-under test (DUT) or hardware-under-test (HUT) is connected through I/O interfaces, e.g., filters, digital to analog (D/A) and analog to digital (A/D) converters and signal conditioners. Certain real-time controls of the simulation such as, closing or opening of switches can be executed with the user-defined control inputs.

If the HIL system involves real controller hardware that interacts with the rest of the simulated system, it is called controller hardware in the loop (CHIL). CHIL is used for rapid controller prototyping. In this method, no real power transfer takes place and the power system is modeled virtually inside the simulator, and the external controller hardware exchanges controller I/O with the virtual system. A newly designed controller is tested using this method, where the controller takes feedback signals from the simulator and processes them to produce the required output signals, which are then sent back to the system (inside the simulator). Such a setup of a controller prototyping or CHIL arrangement is shown in Fig. 1.9, where a power electronic converter is modeled inside the simulator and the real controller is connected to it through I/O.

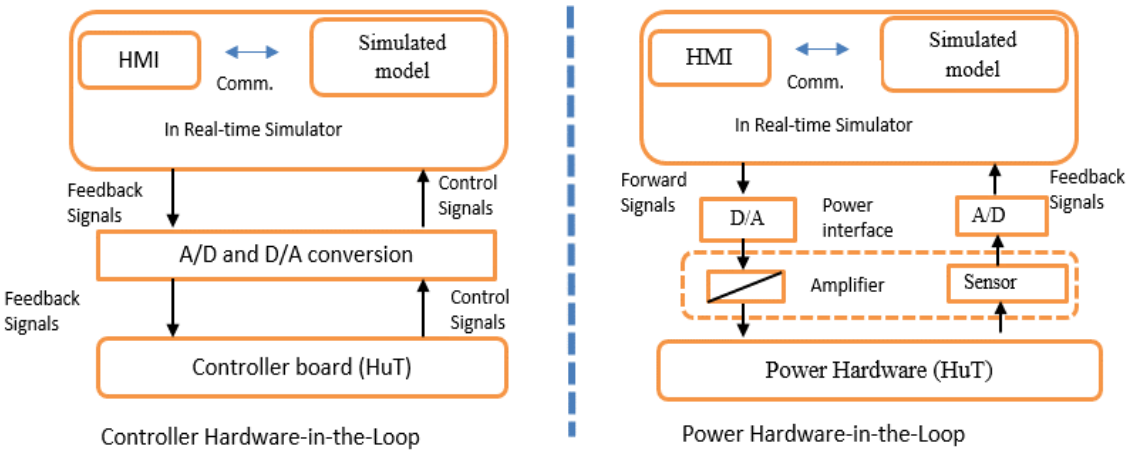

Figure 1.9 – Block diagram showing CHIL and PHIL [38]

HIL simulation involving real power transfer to or from the DUT is called power hardware in the loop (PHIL) (Fig. 1.9). In this case, part of the power system is internally simulated and the other part is the real hardware connected externally. A power source (connected through the PHIL interface) is needed for this setup, which will generate or absorb power. Reference signals are generated by the virtual system inside the real-time simulator and are sent to the power amplifier that produces voltages or currents to be applied to the DUT. Feedback signals obtained from the voltage/current measurements of the DUT are offset, scaled and then brought back to the simulator to complete the simulation loop.

In general, a fully digital simulation is often used for understanding the dynamic behavior of a system under specific circumstances, whereas an HIL simulation is used to minimize the risk of investment through the use of a prototype once the underlying theory is established with the help of a fully DRTS.

1.6 Thesis organization

The thesis describes the real time simulation of smart inverter connected to a distribution system, where the smart inverter is emulated in a PLECS Real-time box and the distribution system is emulated in the OPAL-RT. In Chapter 1, an introduction to PV market growth, inverter classification, Distributed Generation and its effects on the power distribution system, smart inverter and its associated standard and protocols, and real time simulation are discussed

Chapter 2 describes the grid tied inverter in detail. The power topology and the derivation of control voltages for a unipolar Pulse Width Modulation (PWM) is discussed in detail. Filter design and the procedure for choosing the value of inductor L is described.

Then the controller design for current controller, Phase Locked Loop, Real Reactive command follower is described in detail. Finally, the simulation results are discussed with the current and voltage waveforms.

Chapter 3 describes the real time simulation of smart inverter using PLECS RT box in detail. The PLECS RT box is described and the tools in the PLECS software used for real time simulation are explained in detail. Then the procedure for simulating an open loop and closed loop grid tied inverter are demonstrated. This chapter also describes the design of the Poly campus distribution system in the OPAL-RT system using the ePHASORsim tool in the RT-Lab software. Then, the tools used for receiving and sending analog signals in the OPAL-RT are discussed.

Chapter 4 describes the real time simulation results of a smart inverter including the voltage, current waveforms, switching pulses and comparison with the simulation results are explained in detail. Then the results and conclusions of the smart inverter system integrated with the Poly campus power system in the OPAL-RT are described. In addition, future work is proposed to extend the present work in line with the existing standards with improved communication standards.

Appendix A includes the C-code written for filter design, controller design implementation and PWM generation using Digital Signal Processor (DSP) TMS320F38377S for the smart inverter emulated in the PLECS RT box.

CHAPTER 2

GRID-TIED INVERTER

2.1 Introduction

Grid-tied inverter converts input direct current (DC) electric power into alternating current (AC) power into the utility grid. Since, the AC power is fed or absorbed from the grid, the phase of the voltage and current waveforms must match the phase of the grid voltage. The DC power source can be one of the renewable sources of energy like solar power (Photovoltaic cells), wind energy (wind turbine) or fuel cells. The power conversion of the inverter consists of two stages, the first stage being a DC-DC conversion followed by the second stage of DC-AC conversion, where the DC voltage is converted into AC voltage which is connected to the grid.

This chapter discusses the DC-AC conversion stage and its various subsystems. The subsystems of the DC-AC stages are power circuit topology, pulse width modulation, filter design, current control, phase locked loop (PLL) control and control of active and reactive power command following. This chapter also discusses the simulation results of a grid tied inverter.

2.2 Power circuit topology

DC-AC conversion is implemented using a two-pole converter, also known as full-bridge converter. Two-pole converters are normally used in DC-AC conversion, AC-DC conversion and in DC motor drive applications. These converters find major application in grid integration of solar energy with the use of Photovoltaic (PV) inverters into the power

distribution system [39]. These converters can operate in all four quadrants, implying that power flow between AC and DC sides can be bi-directional. Stand-alone inverters employ the same topology with capacitor filters at the AC terminals.

The schematic of a full-bridge converter topology is shown in Fig. 2.1. The two-pole operation enables it to act both as a rectifier (AC-DC converter) and an inverter (DC-AC converter). Power flows from the AC side to the DC side in the rectifier mode and from the DC side to the AC side in the inverter mode of operation.

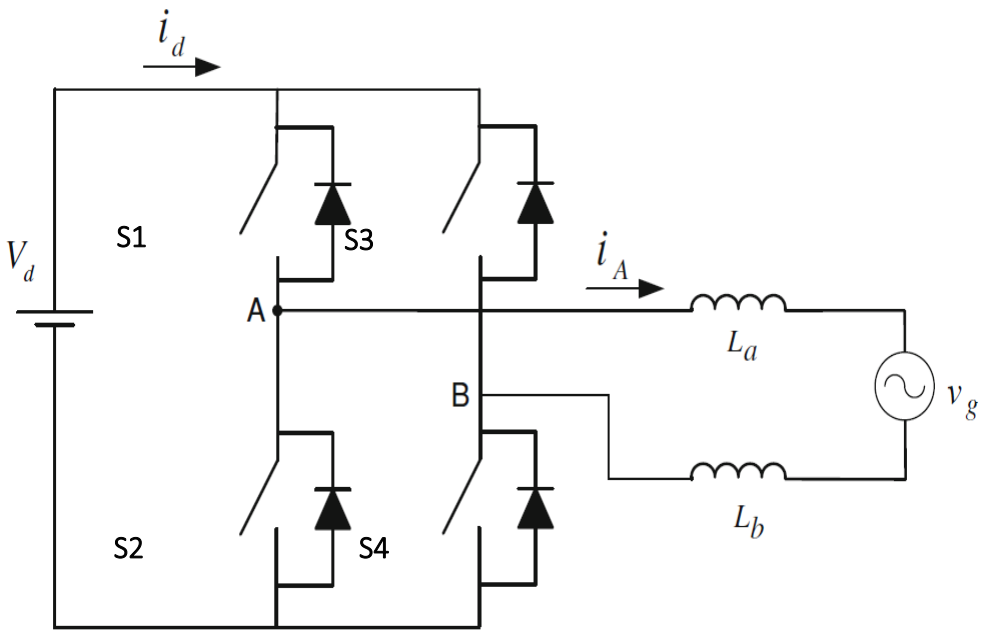

Figure 2.1 – Full-bridge converter topology used for grid-tied inverter

The circuitry compromises of DC voltage source V_d, switches with anti-parallel diodes *S1* to *S4*, inductor filters L_a and L_b and AC voltage source v_g. Two poles, A and B denote the poles of the converter. The current i_d denotes the current flowing from DC source to the grid.

The current i_A denotes the current flowing from pole A to the grid. The average model of the two pole converter is developed with the help of analysis of the two power poles A and B [40]. Each power pole is represented in an average model with the help of ideal transformer implemented in controlled voltage and current sources. The average model developed is shown in Fig 2.2.

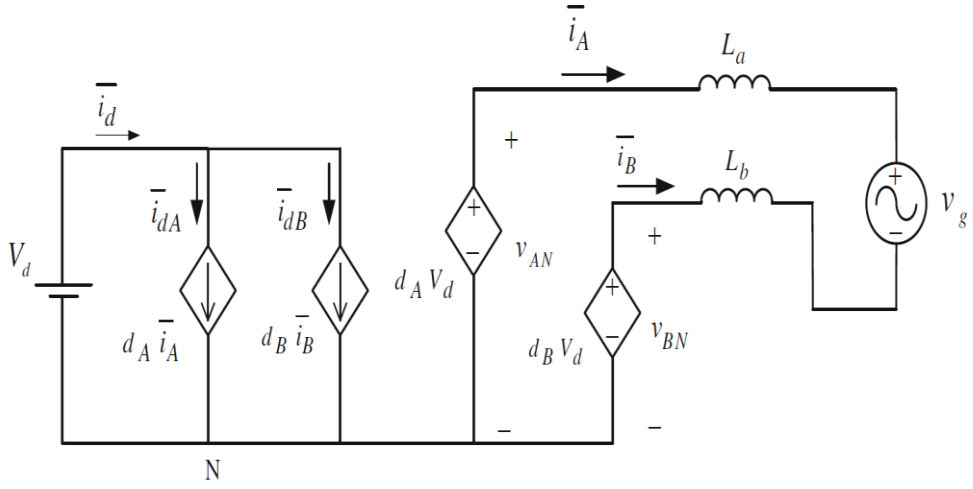

Figure 2.2 – Average model of a full-bridge converter

2.3 Pulse width Modulation

The pulse width modulation (PWM) strategy for a two pole converter is decided based on how power pole B is driven with respect to power pole A. The popular schemes used in industrial applications are unipolar PWM, bipolar PWM and variable duty ratio PWM [39]. In the unipolar PWM, the switching signals of the two power poles $q_A(t)$ and $q_B(t)$ are related as:

$$q_A(t) = -q_B(t) \tag{2.1}$$

In bipolar PWM, the switching signals of two power poles are related as:

$$q_A(t) = 1 - q_B(t). \quad (2.2)$$

In variable duty ratio PWM, one of the poles is driven at the switching frequency with variable duty ratio, and the other pole driven at the fundamental frequency with a fixed duty ratio of 0.5.

In unipolar PWM, the dominant high frequency components which have to be filtered lies at twice the chosen switching frequency. This reduces the size of filters to be designed to achieve a particular total harmonic distortion (THD). Hence, this method provides the best frequency spectrum for the input current, output current and the power pole voltages. These advantages of unipolar PWM makes it widely used at present and hence, this method will only be considered for further discussion in this work.

In unipolar PWM, the two control signals are compared with a triangular carrier. The triangular carrier is assumed to be $\widehat{V_{tri}} = 1V$. The unipolar PWM is defined in terms of the relationship between its switching signals of the power poles or the control voltages of the two power poles, v_{cA} and v_{cB}:

$$v_{cA}(t) = -v_{cB}(t) \quad (2.3)$$

In actual design practice, a control voltage $v_c(t)$ is generated by the controller based on the design parameters and system values. This control voltage is assigned to one of the power poles, say $v_{cA,}$ and v_{cB} is assigned a negative value of the control voltage.

$$v_c(t) = \quad v_{cA}(t) = -v_{cB}(t) \quad (2.4)$$

Figure 2.3 shows the waveforms of carrier waveform $\widehat{V_{tri}}$ along with the control voltages v_{cA} and $v_{cB,}$ the pole voltages V_{AN} and V_{BN} corresponding to poles A and B, and line V_{AB},

the voltage between poles A and B. It is observed that line voltage V_{AB} is positive in the positive half-cycle and negative in the negative half-cycle.

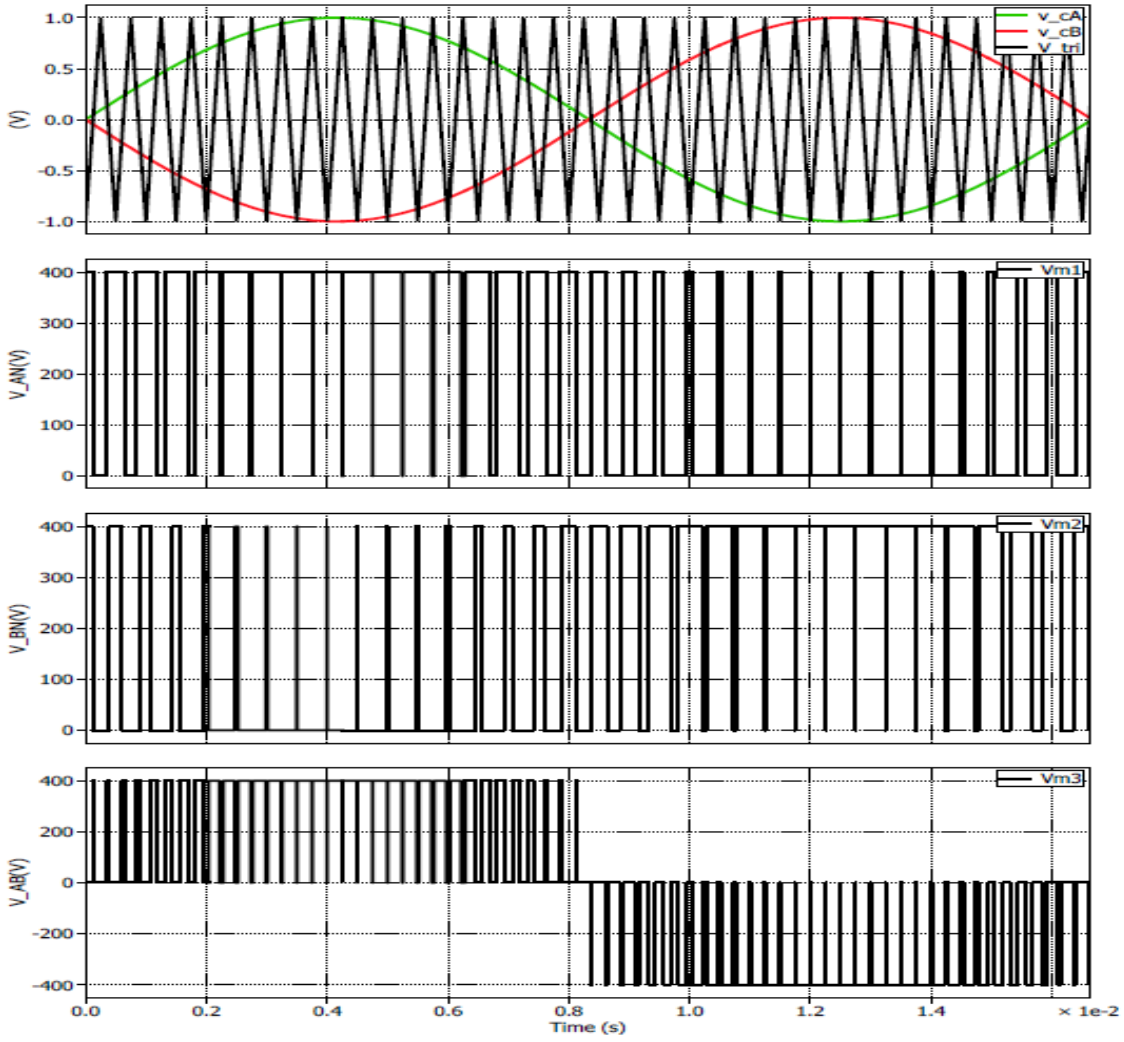

Figure 2.3 – Waveforms of unipolar PWM for a two pole converter

The duty ratio *d (t)* is defined with the help of control voltage v_c *(t)* and carrier voltage $\widehat{V_{tri}}$ *as*:

$$d\,(t) = \frac{1}{2} + \frac{1}{2 * \widehat{V_{tri}}} * v_{cA}\,(t) \qquad (2.5)$$

Using the relationship between Eqs. (2.4) and (2.5), and assuming $\widehat{V_{tri}} = 1V$, we get the following equations:

$$d_A\,(t) = \frac{1}{2} + \frac{1}{2} * v_{cA}\,(t) \tag{2.6}$$

$$d_B\,(t) = \frac{1}{2} + \frac{1}{2} * v_{cB}\,(t) \tag{2.7}$$

The duty ratio *d (t)* is equal to the difference of duty ratios of poles A and B, $d_A\,(t)$ and $d_B\,(t)$. The pole voltages V_{AN} and V_{BN} are related to $d_A\,(t)$ and $d_B\,(t)$ by equation below:

$$V_{AN}\,(t) = V_d * d_A\,(t) = \frac{V_d}{2} + \frac{V_d}{2} * v_{cA}\,(t) \tag{2.8}$$

$$V_{BN}\,(t) = V_d * d_B\,(t) = \frac{V_d}{2} + \frac{V_d}{2} * v_{cB}\,(t) \tag{2.9}$$

The line voltage V_{AB} is defined by the following equation:

$$V_{AB}\,(t) = V_{AN}\,(t) - V_{BN}(t) = \frac{V_d}{2} * (v_{cA}\,(t) - v_{cB}\,(t)\,) = V_d * v_c\,(t) \tag{2.10}$$

Hence, it is observed that the control voltage $v_c\,(t)$ is amplified by the DC voltage source V_d, which is the voltage stiff port of the two pole converter. This is the average model of the two pole converter represented with ideal transformer and dependent voltage and current sources.

2.4 Filter design

The grid connected inverter has a filter on the AC grid side. This filter is used in the inverter circuit to meet the Total Harmonic Distortion (THD) requirements of the output current waveform. THD is defined as the ratio of Root Mean Square (RMS) value of the ripple current to that of the fundamental component of line current.

$$THD = \frac{\Delta I_{RMS}}{I_1} \tag{2.11}$$

Where, ΔI_{RMS} is the RMS value of line current and I_1 is the fundamental component of line current. Since, the voltage ripple is less than 1% in a grid connected inverter, only high frequency current ripple is considered for THD computation. The THD is expressed as a percentage value. The THD has to be below a certain value to meet the various standards by the manufacturers.

Filters can be classified into active and passive depending on the use of switching devices. Active filters involve use of semiconductor switching devices. Passive filters involve use of passive components like resistor, inductor and capacitor. Passive filters are further classified into L, LC and LCL filters, where L denotes inductor and C denotes Capacitor.

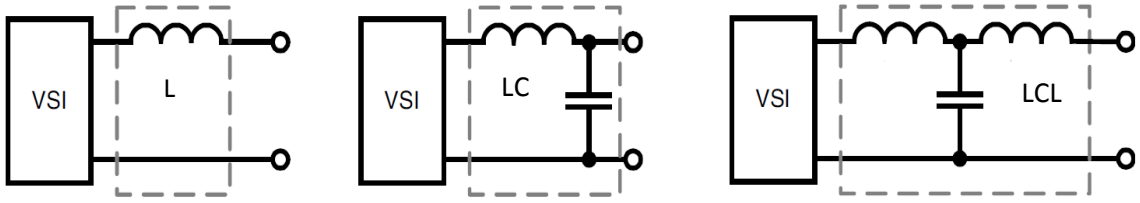

Figure 2.4 – Types of passive filters used in a voltage source inverter

Fig 2.4 shows the types of passive filters used in a voltage source inverter. L-filter is the simplest filter in terms of implementation and design. LC and LCL filter reduces the size of circuit and inductor values to achieve the same objective of filtering harmonics[41]. For the purpose of simplicity, L-filter is chosen for the single phase grid connected inverter for further discussion.

To design the filter inductor, time-domain analysis of the current ripple is used based on the work proposed by Xiaolin Mao [42] etc. The current waveform of an inverter consists of the fundamental component of the current and the current ripple, ΔI_{pk-pk}.

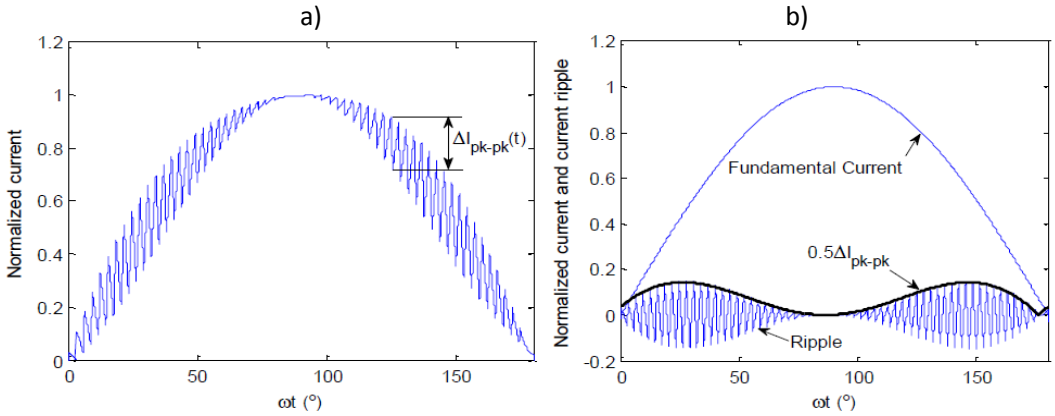

Figure 2.5 – Normalized current and current ripple a) current with ripple b) Fundamental current and ripple separately

Fig. 2.5 a) shows the normalized current with ripple for modulation index of 1 and Fig. 2.5 b) shows fundamental current component and ripple separately. The time domain ripple analysis is discussed in detail in [4] and an expression for peak to peak ripple current, ΔI_{pk-pk} is obtained as follows:

$$\Delta I_{pk-pk} = \frac{[V_d - \bar{v}_{AB}(t)]d(t)T_s}{L} = \frac{V_d T_s}{L}(1 - m_a sin\boldsymbol{\omega t})(m_a sin\boldsymbol{\omega t}) \text{ for } 0 \leq \boldsymbol{\omega t} \leq \boldsymbol{\pi} \qquad (2.12)$$

Where, V_d is the DC source voltage, m_a is the modulation index, $d(t)$ is the duty cycle, $\bar{v}_{AB}(t)$ is the converter output voltage and T_s is the time period. This time domain analysis is further extended to derive an expression for RMS value of the ripple current.

$$\Delta I_{RMS} = \frac{1}{\sqrt{12\pi}} \frac{T_s V_d}{L} m_a \sqrt{\frac{3\pi}{8} {m_a}^2 - \frac{8}{3} m_a + \frac{\pi}{2}} \qquad (2.13)$$

The worst-case value of m_a is at the value which maximum ripple occurs. From Equation 2.13, the worst case value of m_a is found to be 0.6168. Plotting normalized ΔI_{RMS} with respect to m_a , the curve is obtained as in Fig. 2.6. It is observed that maximum value of ripple is obtained at $m_a = 0.6168$.

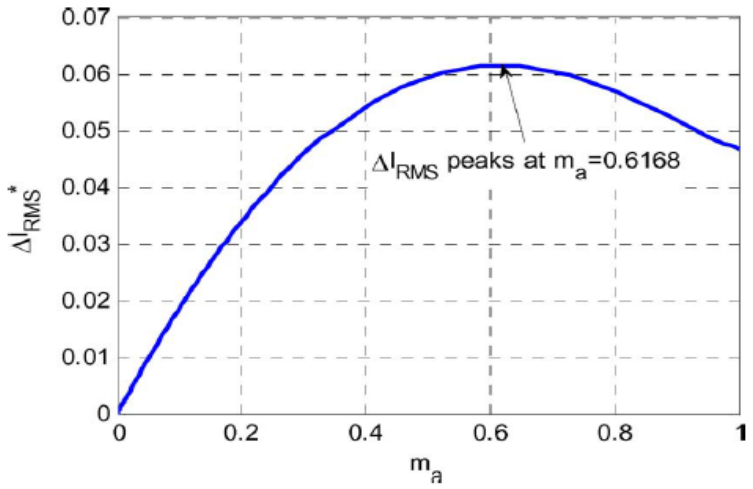

Figure 2.6 - Normalized ripple current versus modulation index

2.4.1 Filter design calculation

For the chosen 5KW string inverter system, L filter has to be designed to limit Total Harmonic Distortion (THD) within 3% of the fundamental value of inductor current. The grid voltage is chosen to be 240V RMS (340V peak value). The peak current value decides the value of inductor to be designed. The value of the inductor is chosen based on equations 2.13. The table 2.1 lists the specifications of the inductor designed for the chosen string inverter.

Table 2.1 Filter design values

Power rating	5KW
DC Voltage	400V
Grid Voltage	240V (RMS) 340V (pk-pk)
Peak inverter current	29.46A
Total Harmonic Distortion	3%
Inductor L value	1.55mH
Series Resistance of L	0.1Ω
Inductor ripple current	0.88A

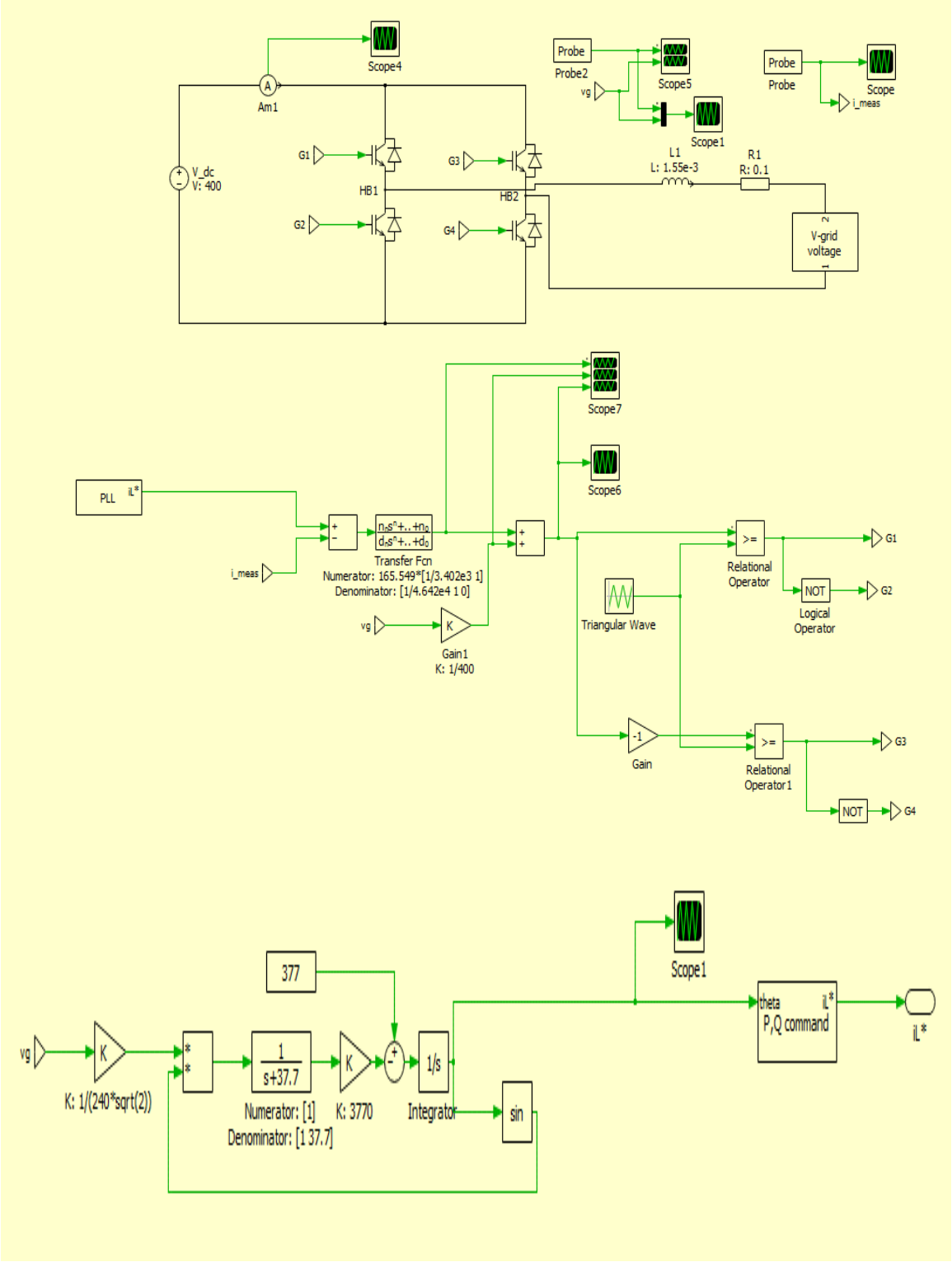

Fig 2.7 - Implementation of a closed loop grid tied inverter

2.5 Controller design

The controller design for the inverter stage consists of current control, phase locked loop (PLL) control and PQ (real-reactive) power control. In this control scheme, PLL control provides the phase component for the reference current in phase with the grid voltage. PQ command block provides the magnitude for the reference current used in current control. The inner current loop compares the reference current value with the measured value of inductor current and ensures that the measured current is equal to the reference current. This section will further discuss the following control schemes and blocks:

- Current control
- Phase locked loop (PLL) control
- PQ (real-reactive) power command following

2.5.1 Current Controller

Current controller is designed to control the current injected into the grid. The controller has a reference current command. The reference current command has a magnitude and phase component. The magnitude of the reference is based on the power rating and the insolation of the DC source. The phase component comes from the Phase Locked Loop (PLL) control. The PLL control will be discussed in the later sections. The current controller is an internal controller and hence it should be much faster (at least 10 times) than the external PLL control. For a grid tied inverter with L filter, two types of current controls are implemented namely Proportional Integral (PI) control with feed forward term and Proportional Resonant (PR) control.

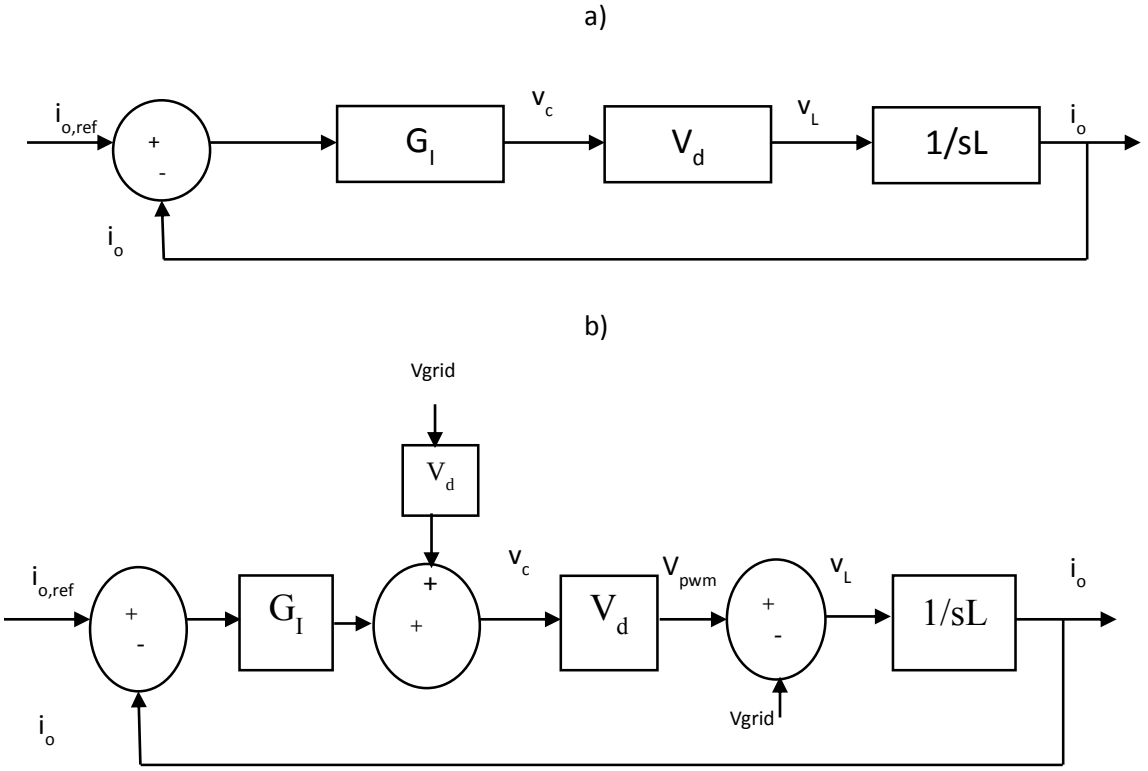

Figure 2.8 - PI controller implementation a) without the feed forward term b) with feed forward term

The PI controller is the most commonly used controller scheme. The implementation is shown Fig 2.8 with and without feed forward term. The feed forward term v_g is introduced to reduce the disturbance in input. The plant for designing the current controller is given by the expression below:

$$G_{iL} = \frac{1}{sL + r_L} \tag{2.14}$$

In equation (2.14), r_L is the internal resistance associated with the inductance L. The controller is designed based on the K-factor method proposed by HD Venable. Based on the values of the inductor and its internal resistance, a type 2 controller is used for controller design. A type 2 controller is a two pole one zero system. Equation

(2.15) shows the expression for a type 2 controller, where K_C is the magnitude of loop gain at cross-over frequency, ω_c is the cross-over frequency, ω_z is the zero frequency and ω_p is the pole frequency. The detailed controller design and calculations are shown in the appendix.

$$G_c(s)_{(s=j\omega_c)} = \frac{K_C}{s}\frac{(1+\frac{s}{\omega_z})}{(1+\frac{s}{\omega_p})} \tag{2.15}$$

Despite its simplicity, the PI controller has some disadvantages. The PI controller causes a large steady state error in the output from the current controller. Though the grid feedforward term reduces the error, its implementation is very complex. The noise and harmonics in the grid voltage affects the current controller and the feed forward term introduces delay in the voltage sensor.

The Proportional-Resonant (PR) controller significantly reduces the steady state errors in magnitude and phase without the feedforward term, since the gain at the fundamental frequency is very large [45]. The transfer function for the PR controller is given by equation (2.16), where K_p is the proportional constant, K_i is the resonant constant and $\boldsymbol{\omega}_o$ is the resonant frequency.

$$G_{PR}(s)_{(s=j\omega_c)} = K_p + \frac{K_i s}{s^2+\omega_o^2} \tag{2.16}$$

The resonant constant has a gain close to zero at frequencies well above and well below the resonant frequency. At these range of frequencies, the controller is only determined only by K_p, proportional constant and the plant gain. The proportional constant, K_p is calculated using the same procedure as in a PI controller using K-factor method.

Proportional constant, is given by the relation in equation (2.17) where $\left|G_{sys}\right|_{\omega_c}$ represents the magnitude of the system at cross over frequency

$$K_p = \frac{1}{\left|G_{sys}\right|_{\omega_c}} \tag{2.17}$$

The value of resonant constant is chosen to ensure sufficient loop gain (say 1000 at 59.3 Hz) at the minimum or maximum allowable grid frequency (60 ± 0.7 Hz) while ensuring a good phase margin. Figure 2.9 shows the bode magnitude and phase plots of a PR controller.

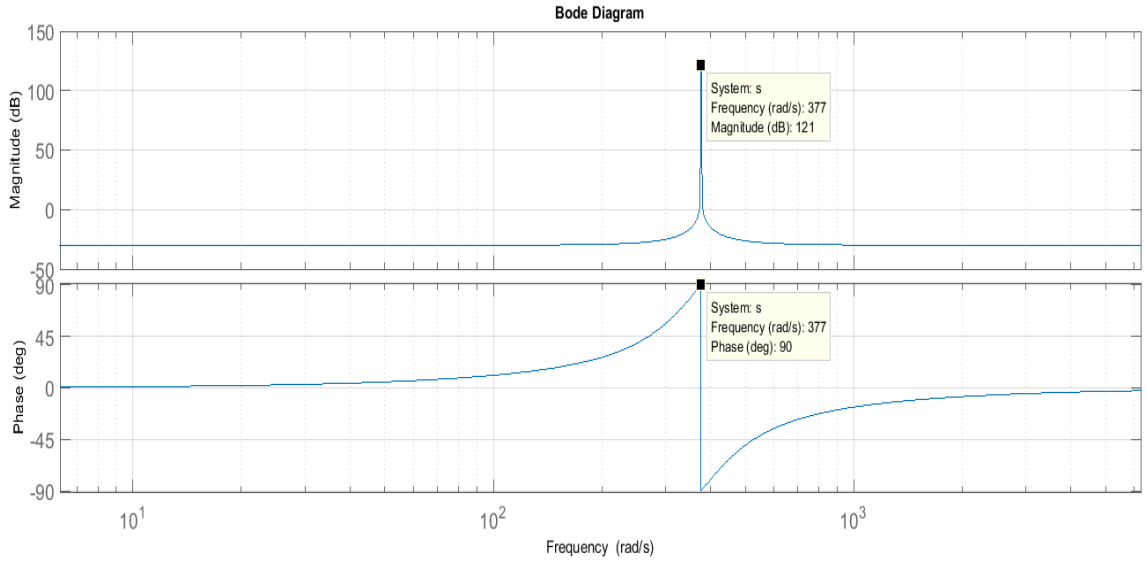

Figure 2.9 - Bode magnitude and phase plots of PR controller

2.5.2 Phase Locked Loop (PLL) control:

The PLL control is required for grid synchronization, the mechanism by which the internal reference signal generated by the inverter controller is maintained in phase with the grid voltage. Any error in grid synchronization will lead to uncontrolled power flow between the grid and inverter leading to overvoltage, overcurrent and tripping of protection devices.

The IEEE 1547 Standard for Interconnecting Distributed Resources with Electric Power Systems requires that grid synchronization shall not cause area Electric Power System(EPS) voltage fluctuation higher than 5%. It also states that synchronization device shall be capable of withstanding 1800 degrees out of phase [43].

There are several grid synchronization methods for distribution systems [44]. Zero-cross detection synchronization method is the simplest method for grid synchronization. However, this method suffers from sensitivities to noise in grid voltages. Hence, this method is not suitable for the standard requirements needed to be met by modern grid tied inverters. Phase Locked Loop (PLL) method is the predominant method used for grid synchronization of grid tied inverters.

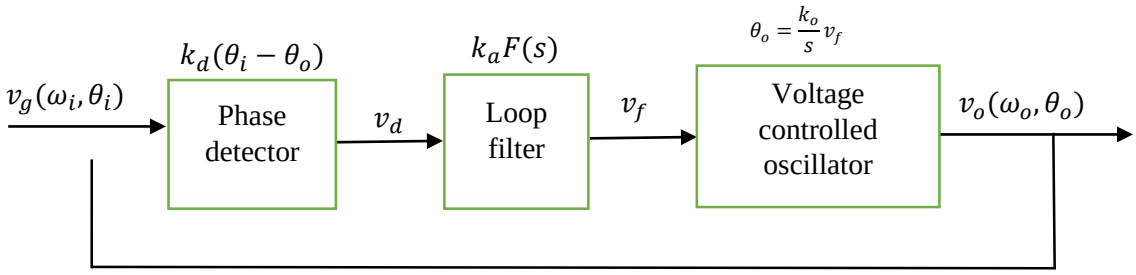

Figure 2.10 - Block diagram of Phase Locked Loop (PLL) implementation

Figure 2.10 shows the implementation of PLL control. The phase of the grid voltage is detected and it is compared with the phase generated from the internal voltage controlled oscillator. The phase detector compares the input phase and reference phase and generates an output voltage v_d proportional to the phase difference, along with the double frequency component of the ac voltage, $2f_{ac}$.

The loop filter is a first order filter or a PI controller acting as a low pass filter. The loop filter attenuates the high frequency ripple components in v_d and produces a filtered voltage

v_f. This voltage is fed into a voltage controlled oscillator (VCO). The oscillator generates sine and cosine signals with frequency and phase depending on its input DC voltage. The output of the VCO is fed back to the phase detector.

The PLL controller is the external controller in the loop. Hence, it has to be much slower than the internal current controller. The corner frequency is typically chosen as 1/20th of the expected double frequency component of the phase detector. For a 60 Hz system, the corner frequency is 2*60/20 i.e 6 Hz. With 6 Hz as the corner frequency. The transfer function of the first order low pass filter is given by $\frac{1}{s+37.7}$ .The implementation of the PLL in simulation is shown in the Figure 2.11

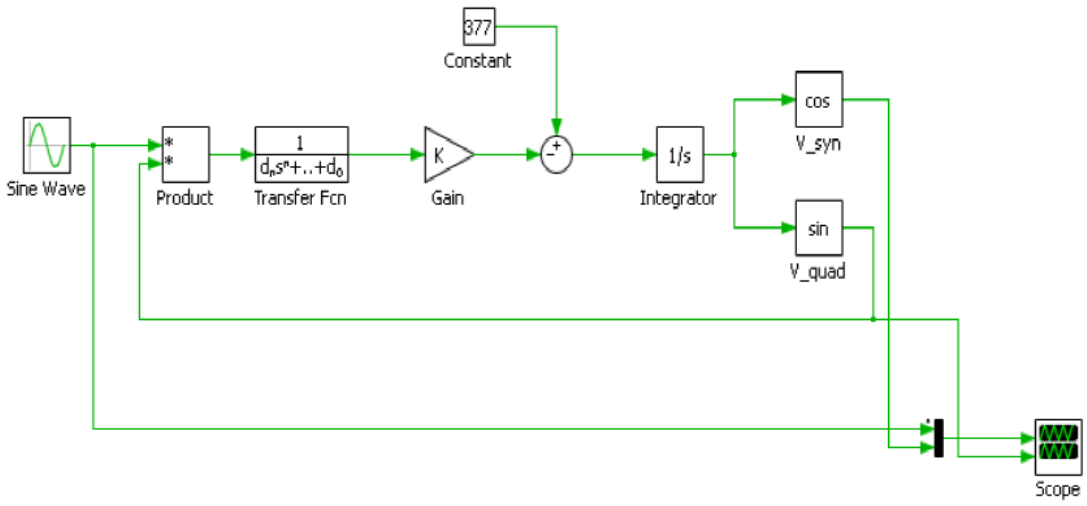

Figure 2.11 - PLECS implementation of a PLL

2.5.3 PQ power command following:

The PQ power command block receives the real (P) and reactive (Q) power references from the utility and calculates the magnitude of the current reference used in the current

controller. The block uses the sine and cosine component of the grid voltage v_g and uses the equation (2.18) to compute the current reference $\widehat{I_L}$.

$$\widehat{I_L} = \frac{P*cos\theta - Q*sin\theta}{\widehat{v_g}} \tag{2.18}$$

The implementation of PQ power command to generate $\overline{I_L}$ in PLECS is shown in the figure 2.12. The position θ is obtained from the PLL and then the sine and cosine of the angle are computed and then the calculation is done as per equation (2.18) to get $\widehat{I_L}$.

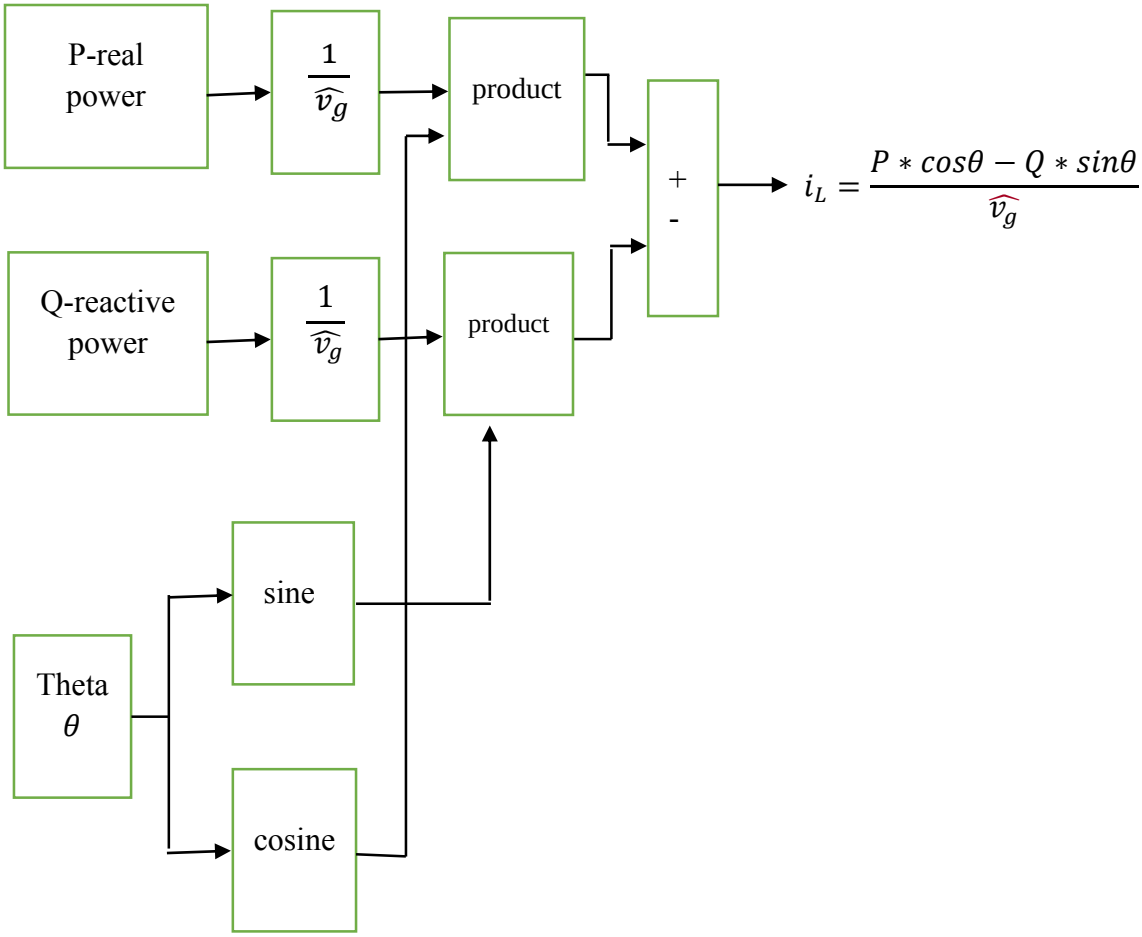

Figure 2.12 - PQ command block

2.6 Simulation results

PLECS simulation software is used for simulating the grid tied string inverter. The inverter has a DC voltage source of 400V and the inverter output is connected to the grid with 240V RMS value (340V pk-pk) at 60Hz. The design of power stage and the various controllers are discussed in the previous sections of the chapter. Figure 2.11 shows the implementation of closed loop grid tied inverter in PLECS.

To verify the design of the filter for meeting the THD requirements of the inverter, the current waveform is observed as shown in Fig. 2.13. The current fed into the grid is for a command of 5000W power at 240V RMS. We can observe that the peak current is 29.41A with a THD of 1.69%

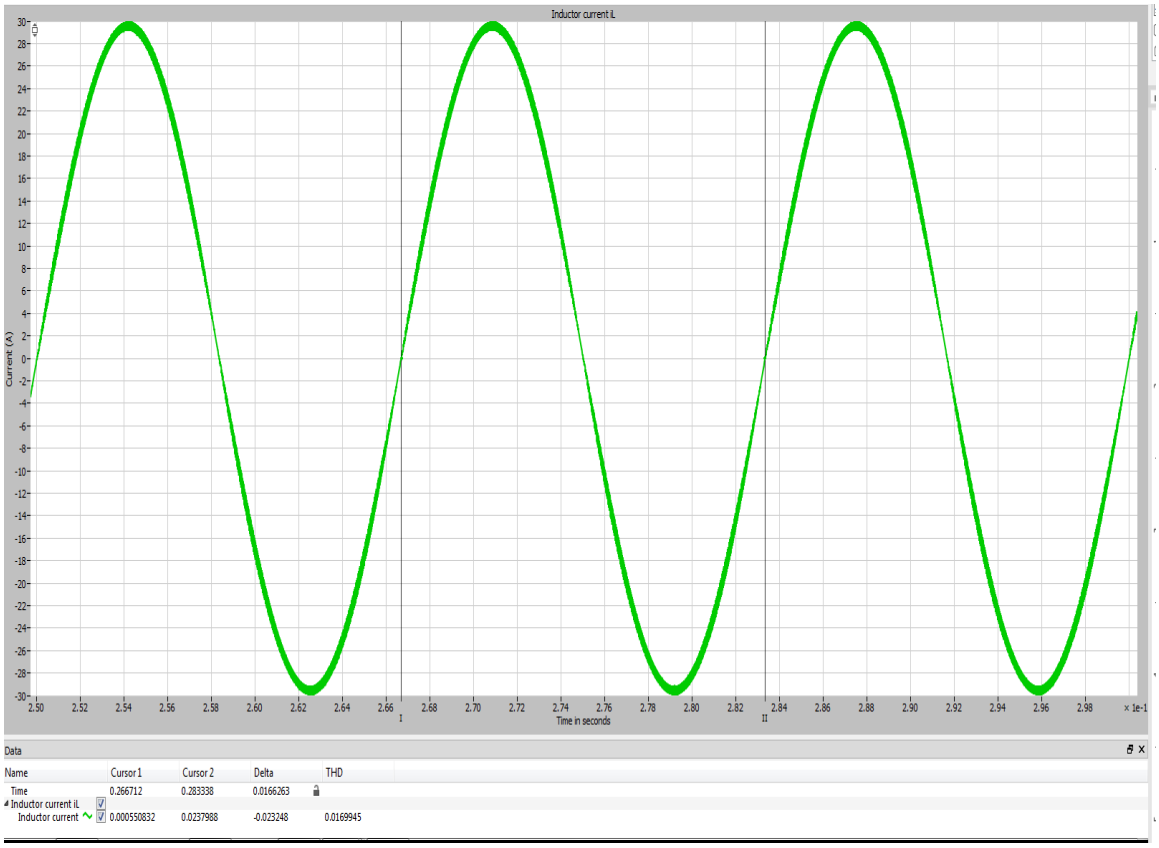

Figure 2.13 - inductor current waveform with THD value at the bottom

The function of Phase Locked Loop (PLL) is to ensure that the controller generated reference is in phase with the grid voltage, as stated in the previous sections. Figure 2.14 shows the combined waveforms of grid voltage and inductor current. The current reference is generated for a 5000W active power command for 240V RMS. It is observed that with reactive power command Q = 0, the inductor current is in phase with the grid voltage. When the reactive power command is of non-zero value, grid current will have phase lag or lead with respect to grid voltage.

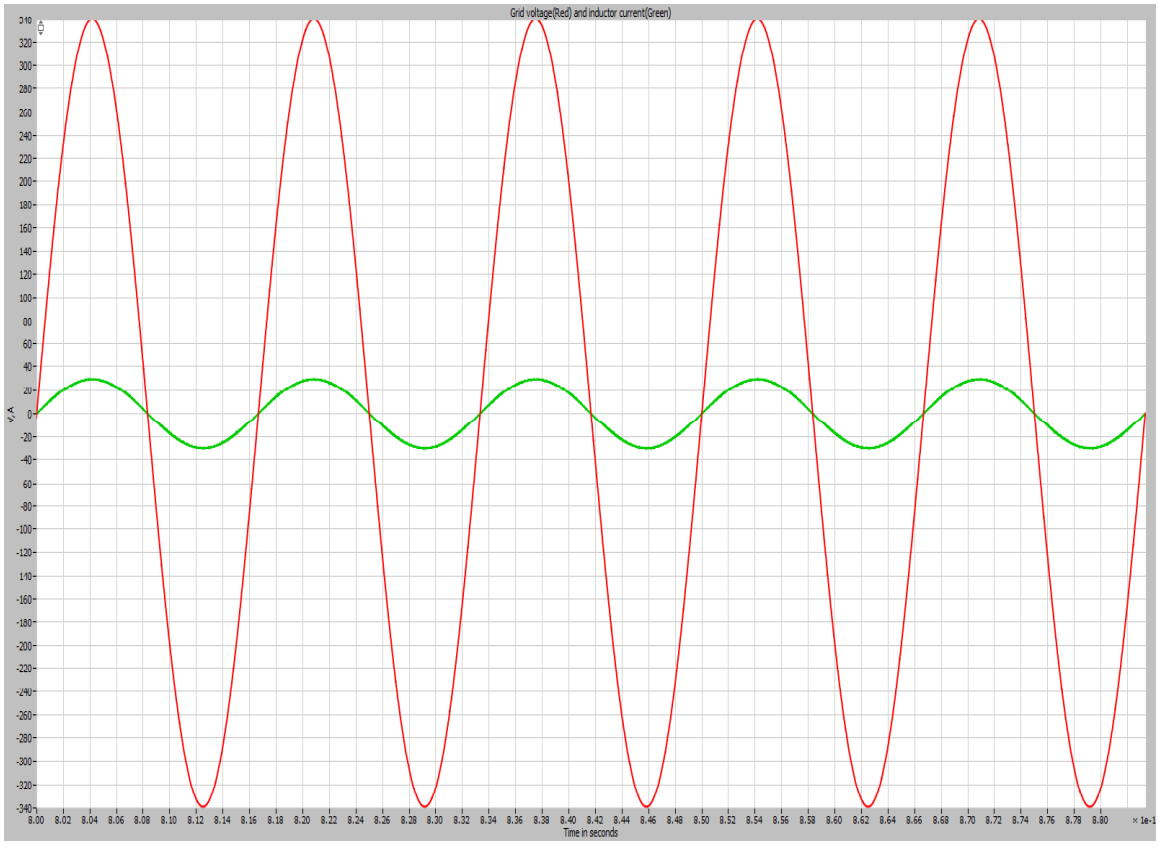

Figure 2.14 - Combined waveforms of grid voltage and inductor current

Fig. 2.15 shows the performance of the PI controller when current dynamically changes from 20A to 30A. The large steady state error associated with PI controller can be observed at the point where the current changes from 20A to 30A.

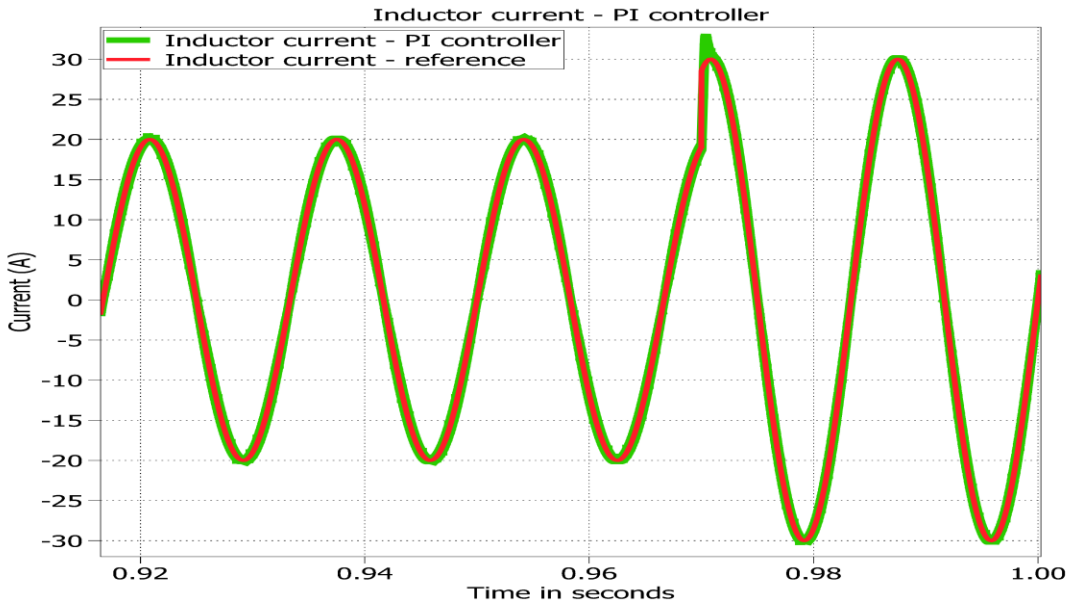

Fig. 2.15 - Performance of PI current controller

Fig 2.16 shows the performance of the PR current controller when current dynamically changes from 20A to 30A. It is observed that the large steady state error is reduced as compared to the PI controller

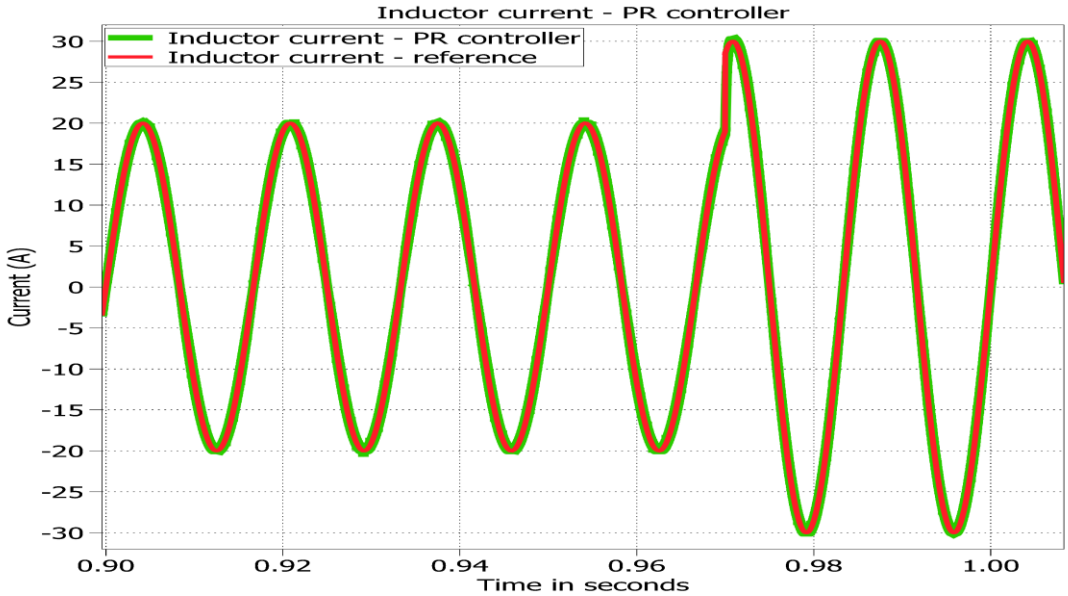

Fig. 2.16 - performance of PR current controller

With real and reactive commands determining the magnitude and phase of controller current reference, various values of P and Q are actual load requirements in a utility grid. The various values of P and Q determine the power factor, power factor angle, current magnitude and phase. Five cases of differing P and Q are considered and they are tabulated below in table 2.2

Table 2.2 - Relation between P, Q and power factor, power factor angle

Real power – P (W)	Reactive power- Q (VAR)	Apparent power – S (VA)	Power factor	Power factor angle (degrees)
5000	0	5000	1	0
4000	3000	5000	0.8 lead	36.86
4000	-3000	5000	0.8 lag	-36.86
0	5000	5000	0	90
0	-5000	5000	0	-90

Fig. 2.17 a) – e) shows the combined waveforms of grid voltage and voltage waveforms for values of real and reactive power commands mentioned in table 2.2.

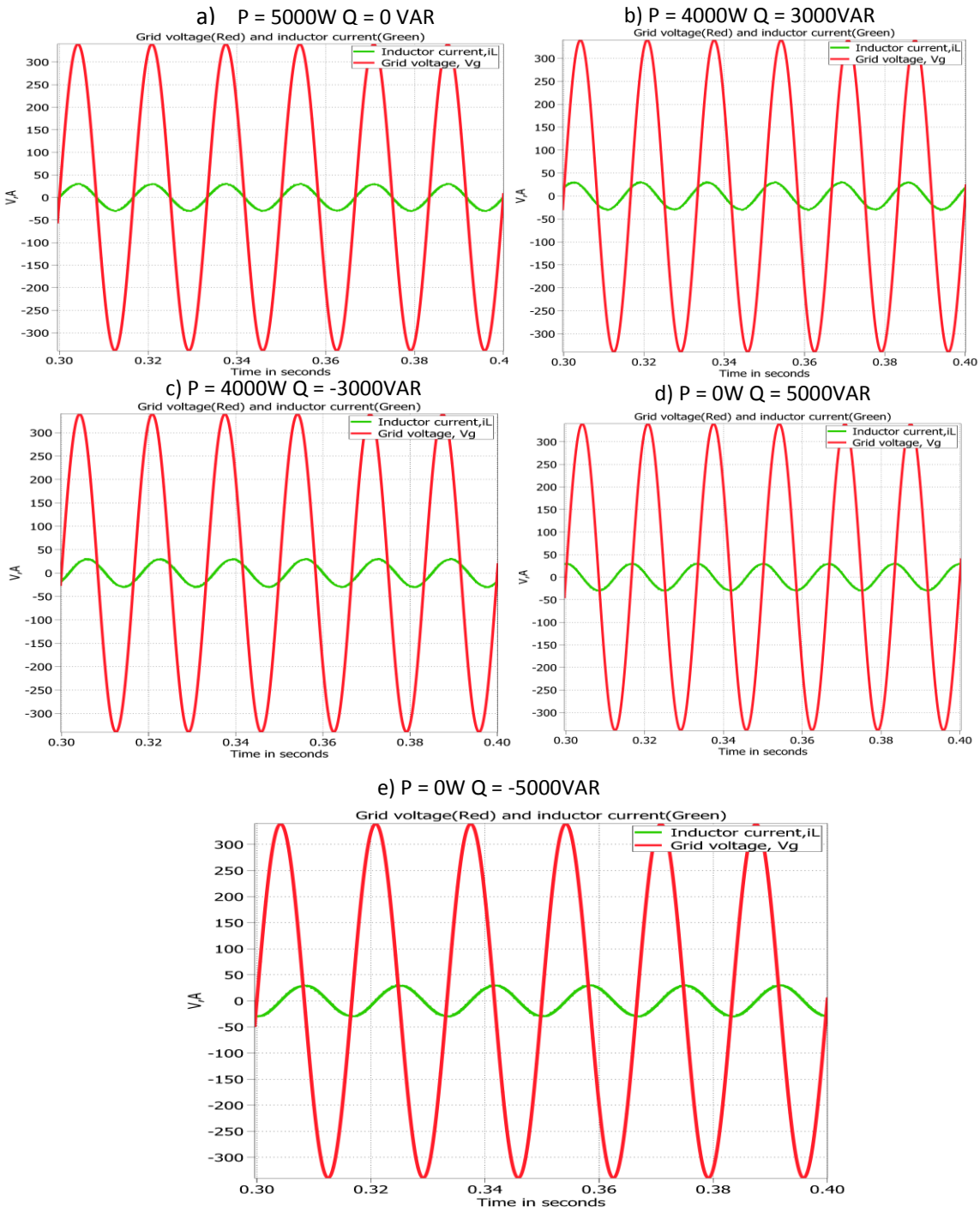

Fig 2.17 a) – e) - Combined waveforms of grid voltage and inductor current with various real(P) and reactive(Q) power combinations

CHAPTER 3

REAL-TIME SIMULATION OF A SMART INVERTER USING PLECS-RT BOX

Having analyzed the grid tied inverter in detail in the last chapter, this chapter deals with the real time simulation of the grid tied inverter with smart features implemented with PLECS-RT box. The implementation of the smart inverter is done with the Controller Hardware In the loop (CHIL) design. In CHIL mode, the PLECS RT box acts as the power stage of the inverter and the Digital Signal Processor (DSP) connected to the PLECS RT box acts as the controller for the power stage.

This chapter deals with the features of the PLECS RT box and its associated component libraries in the PLECS software, implementation of smart inverter with the RT Box, basic features of the OPAL-RT along with the implementation of the Poly campus distribution system in OPAL RT, and implementation of smart inverter connected to a distribution system using together OPAL RT and PLECS RT.

3.1 PLECS-RT box

The PLECS RT box is a real time simulator built by PLEXIM Inc [46]. It has 1 GHz dual-core central processing unit (CPU) along with provision for 32 analog inputs and outputs, and 64 digital inputs and outputs. This simulator can be used for Hardware In the loop (HIL) applications and rapid controller prototyping.

In HIL applications, PLECS-RT Box emulates the power stage of a power electronic system. The power stage could be as simple as a buck converter or as complex

as a motor drive system. In the HIL setup, The Device Under Test (DUT) is the controller connected to the PLECS RT box.

Figure 3.1- PLECS RT box along with the controller

Fig. 3.1 shows the PLECS-RT box along with the controller. This module has three hardware blocks

1. PLECS RT Box which emulates the power stage of a converter system.
2. Controller which is the DUT. The controller used for the smart inverter design is a Texas Instrument DSP, TMS320F28377S [47].
3. Breakout board, which acts as the interface between the controller and PLECS RT box.

3.1.1 PLECS-RT Box specifications

The PLECS-RT box can receive analog and digital inputs and transmit analog and digital outputs. The analog outputs can be in the form of measured parameters in the PLECS software, which are scaled and then offset to ensure the output voltage from PLECS is within the limits. The analog inputs can be in the form of voltage, current or position

sensor output. The analog inputs can be differential or single ended. The analog outputs can be in the form of measured parameters in the PLECS software, which are scaled and then offset to ensure the output voltage from PLECS is within the limits. The PLECS RT box has 16 analog inputs and 16 analog outputs. The digital inputs can be in the form of pulses generated from a controller. The digital outputs can be in the form of pulses generated from the PLECS-RT box. The RT box has 32 digital inputs and 21 digital outputs. The specifications of the PLECS-RT box are shown in table 3.1

Table 3.1 – Specifications of PLECS-RT box

Processor	Xilinx Zynq Z-7030	1 GHz
Analog inputs	Channels	16, simultaneous sampling
	16, simultaneous sampling	16 bit
	Voltage ranges	-10 to 10 V, -5 to 5 V
	Input type	Differential
	Sample rate	2 Msps, no cycle latency
	Connector	D-Sub 37 pin male
Analog outputs	Channels	16, simultaneous sampling
	Resolution	16 bit
	Voltage ranges	-10 to 10 V, 0 to 10 V -5 to 5 V, 0 to 5 V
	Sample rate	2 Msps, no cycle latency
	Output impedance	0Ω
	Connector	D-Sub 37 pin male
Digital inputs	Channels	32
	Logic levels	3.3 V (5 V tolerant)
	High-level input	min. 2 V
	Low-level input	max. 0.8 V
	Connector	D-Sub 37 pin male
Digital outputs	Channels	32
	Logic levels	3.3 V,5 V
	Connector	D-Sub 37 pin male
Connectivity	Ethernet	RJ-45, Gigabit

3.1.2 PLECS-RT box library

The PLECS software has separate components in the library for using the PLECS-RT box. There are four components in the PLECS library useful for real-time simulation [48]. These components include

- Analog In
- Analog Out
- Digital In
- Digital Out

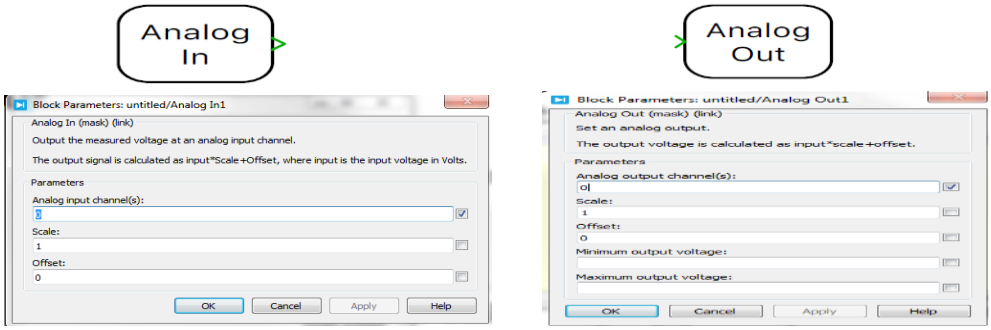

Figure 3.2 – PLECS Analog IO library component and its parameter settings

The PLECS components associated with analog IO are shown in Fig. 3.2. The analog voltage is set based on the formula of input*Scale + Offset. The scale factor is used for multiplying the input parameter with scale and the offset is used for adding or subtracting voltage from the scaled input. The input and output channels can be numbered from 0 to 16. Digital IO can be simply mentioned by the input and output channel number ranging from 0 to 31 as shown in Figure 3.3.

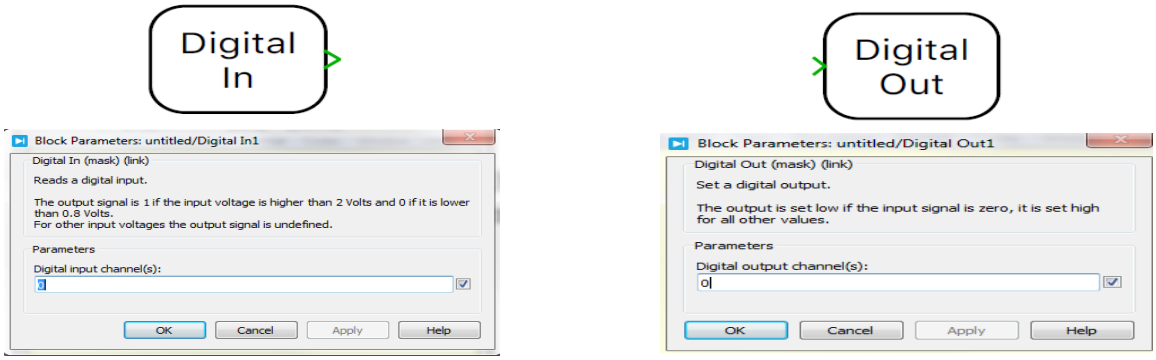

Figure 3.3 – PLECS Digital IO library component

3.1.3 DSP Controller

Texas Instruments DSP Delfino TMS320F28377S is used for designing the controller stage of the smart inverter. This DSP is a single CPU core 32-bit fixed-point processor with a clock frequency of 200 MHz belonging to the C-2000 family of processors. The DSP comes with twelve channels for Enhanced Pulse Width Modulation (EPWM) and six channels for High Resolution Pulse Width Modulation (HRWPM). The DSP also comes with 16 channels of ADC, with the resolution configurable to 16-bit or 12-bit. Three channels of ADC can also act as DAC in 12-bit resolution. Figure 3.4 shows the C2000 Launchpad development kit for the Delfino DSP.

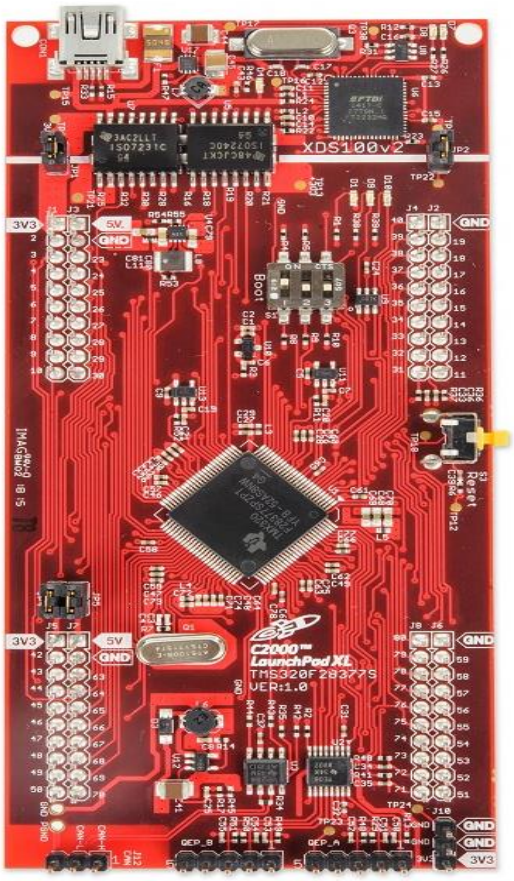

Figure 3.4 – C2000 Launchpad Development kit for the Delfino DSP

The DSP has provisions for IO communication with the features of Serial Peripheral Interface (SCI), Serial Communication Interface (SPI), Inter-Integrated Circuit module (I2C), Controller Area Network (CAN), Universal Serial Bus controller (USB), and Universal Asynchronous Receiver and Transmitter (UART). The DSP also has powerful features like Floating-Point Unit (FPU), Trigonometric Math Unit (TMU) and Viterbi Complex Math unit which reduces computational period by number of cycles.

3.1.4 Breakout Board

The breakout board acts as an interface between the PLECS RT box and the DSP. The breakout board connects the analog and digital IO of PLECS RT box with the individual pins in the DSP development kit. There are extra provisions to probe 8 ADC inputs directly to the oscilloscope. All the digital inputs and outputs also can be probed with the help of jumper wire connectors in the break out board.

3.2 Implementation of a smart inverter with PLECS-RT box

The implementation of smart inverter with PLECS-RT box is done in the CHIL mode. In CHIL mode, PLECS-RT box acts as the power stage of the inverter and the DSP acts as the controller. To emulate the power stage in PLECS-RT box, the power stage circuit is implemented in the PLECS software. The gate signals for the switches of the inverter are obtained as digital inputs from the DSP controller.

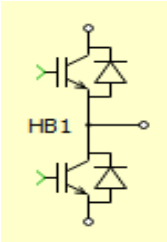

Figure 3.5 – IGBT half bridge used to represent single leg of the inverter

The single leg of the inverter is implemented using a switched model of IGBT half-bridge from the power modules library of PLECS software. The switches are assumed to be ideal in the switched model. There is also an averaged model for the IGBT half bridge, which is represented using a combination of controlled voltage and controlled current sources. Fig. 3.5 shows the IGBT half bridge available in the PLECS software library.

Having discussed analog and digital IO libraries in the previous sections, the IGBT half bridge is used for the two legs of the single phase inverter. In the PLECS software, the smart inverter was a modeled as a grid connected single phase inverter with a L filter. The value of inductor L was chosen based on the filter design mentioned in the earlier sections. Each leg only receives one digital input from the DSP controller for the top switch. The gate signals for the switches of the inverter are obtained as digital inputs from the DSP controller. The grid voltage and inductor current are given as analog inputs to the ADC of the DSP. A grid tied inverter with the control voltages generated from the DSP is shown in the Fig. 3.6

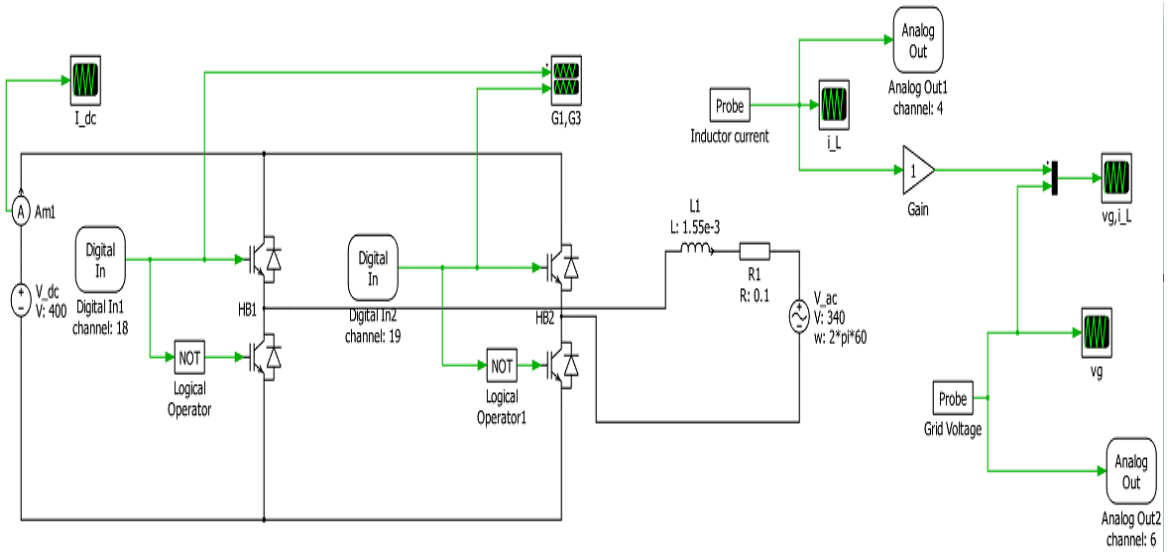

Fig 3.6 – Implementation of power stage of inverter in PLECS

The PLL and current controller designed for closed loop operation of the inverter is implemented in the DSP. The DSP receives grid voltage v_g and inductor current i_L from the inverter as analog inputs. The grid voltage v_g is fed into the PLL controller to get the "ωt" phase component. The PLL controller is implemented in the form of difference equations. With the phase generated from the PLL, the magnitude of the reference current is provided within the DSP as a float variable. The product of magnitude and phase component forms the current reference i_{L_ref}. The i_{L_ref} and $i_{L\_measured}$ is compared to get the error current. This is fed into the PI or PR current controller, which is also implemented in the form of difference equation. The output of the controller $v_{cA}\ (t)$ is modified using equation (2.5) to get duty $d\ (t)$. The duty $d\ (t)$ is multiplied with the Time-Base period (TBPRD) of the EPWM function. The product is compared with an up down counter of time period, TBPRD to get the digital pulses for the switches S1, S3 of the inverter modeled in the PLECS software. This process repeats every time the DSP goes through the ADC Interrupt Service Routine (ISR). The ADC service routine happens at the time period of switching frequency. The switching frequency is chosen as 20KHz. This is shown in Figure 3.7.

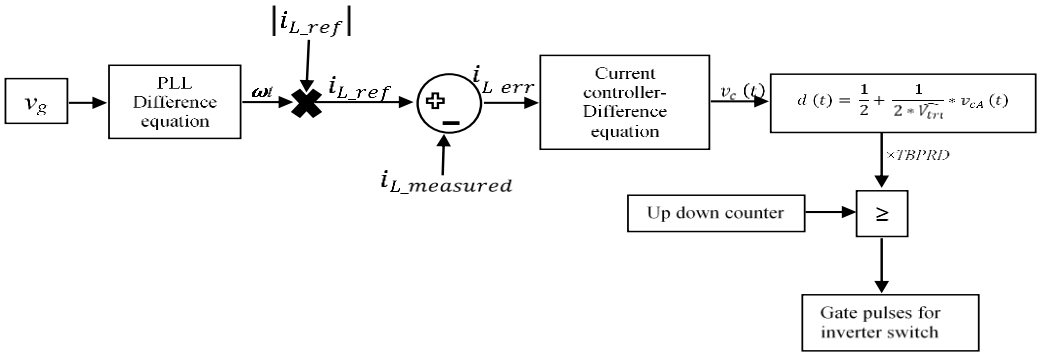

Fig. 3.7 – ADC Routine in the DSP for inverter controller

When the inverter does not receive any commands from the grid, the inverter delivers powers to the grid based on its input power source and inverter rating. When the inverter can receive commands from the utility operator or other inverters, it should be capable of changing the current reference i_{L_ref} based on the real and reactive power command it receives from the grid. The real power command, P and reactive power command, Q are given as analog inputs to the DSP from the PLECS software. The DSP receives the power commands and then changes the current reference i_{L_ref}. The current reference i_{L_ref} from P and Q commands is calculated using equation 2.18. The implementation of this scheme is shown in Fig. 3.8.

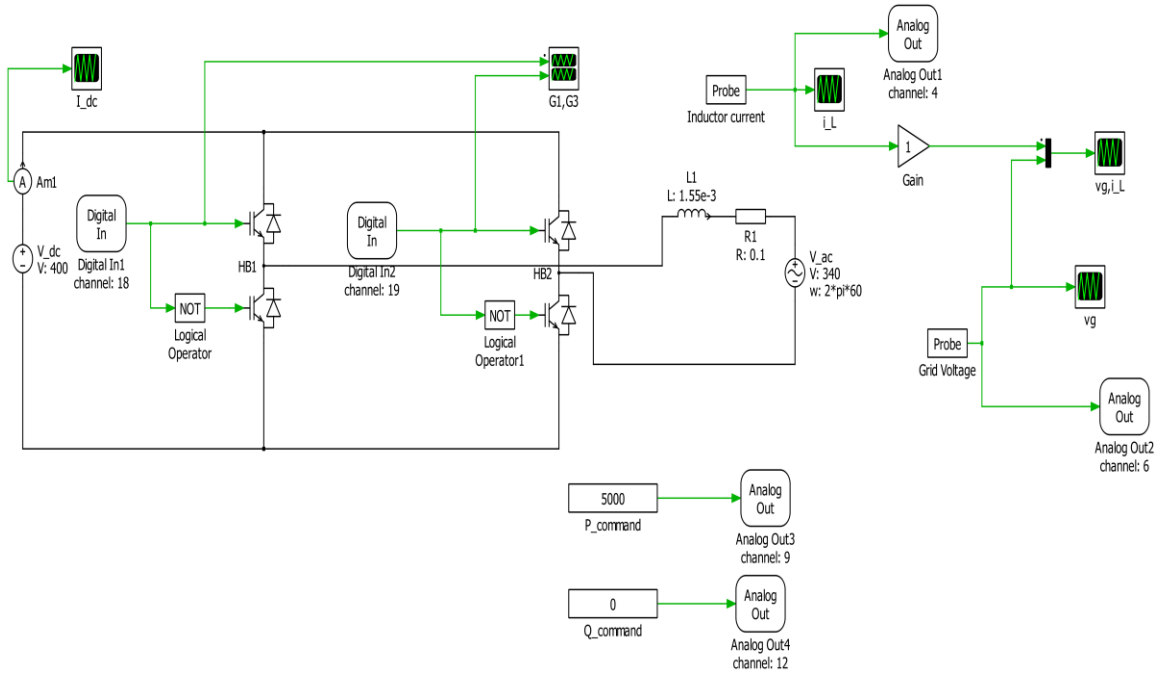

Fig. 3.8 – Power stage implementation with P, Q command

The controller is implemented in the DSP with the addition of P and Q commands. The i_{L_ref} is generated based on the equation 2.18. The PLL, current controller, EPWM

generation is similar to Fig. 3.7. Thus, the power stage and controller together form the smart inverter system.

3.3 OPAL-RT system and Poly campus distribution system

OPAL-RT is a real time simulator [49] with provisions for simulating applications in power system, power electronics, automotive and aerospace applications. It is also used for rapid control prototyping. It is integrated with MATLAB/SIMULINK. The system architecture is explained in [50] and is shown in Figure 3.9.

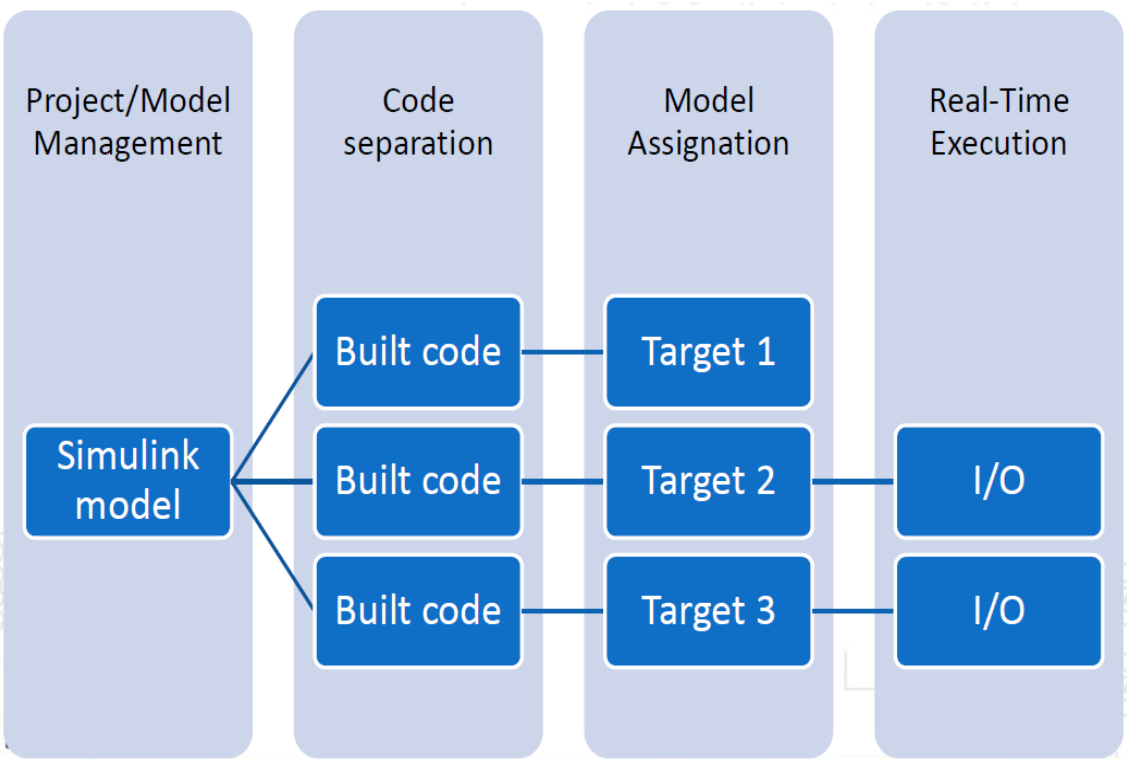

Fig. 3.9 – Architecture of OPAL-RT system

The OPAL-RT system is run on a Linux REDHAT operating system with a powerful real-time target with up to twelve 3.3 GHz Intel processor cores. It has up to 128 analog IO and 256 digital IO. It has provision for RJ-45 cables, DB-37 connectors and 4 PCI slots. It has front monitoring with access to all IO. It also has support for third party IO's including

IEC61850, Asynchronous TCP/IP, serial communication, CAN bus, MOD bus etc. The entire input and output section is based on Xilinx Virtex 7 FPGA.

3.3.1 ePHASORsim

ePHASORsim is an OPAL-RT based real-time transient stability simulation tool. The RT-lab software has a phasor solver in which settings can be configured for emulating the power system. The settings window for the ePHASORsim solver is shown in Fig. 3.10.

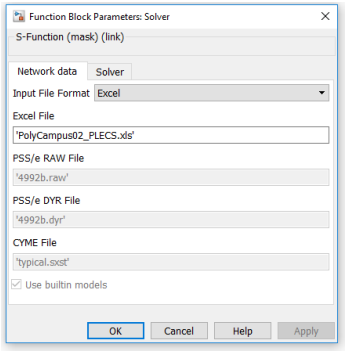

Fig. 3.10 – Solver settings for the ePHASORsim tool

The input file can be in the form of excel file (.xls) or a PSS/E Ver.32 file (.raw, .dyr). The Poly campus distribution system is modeled using an excel file. The online energy consumption of poly campus is shown in Fig. 3.11.

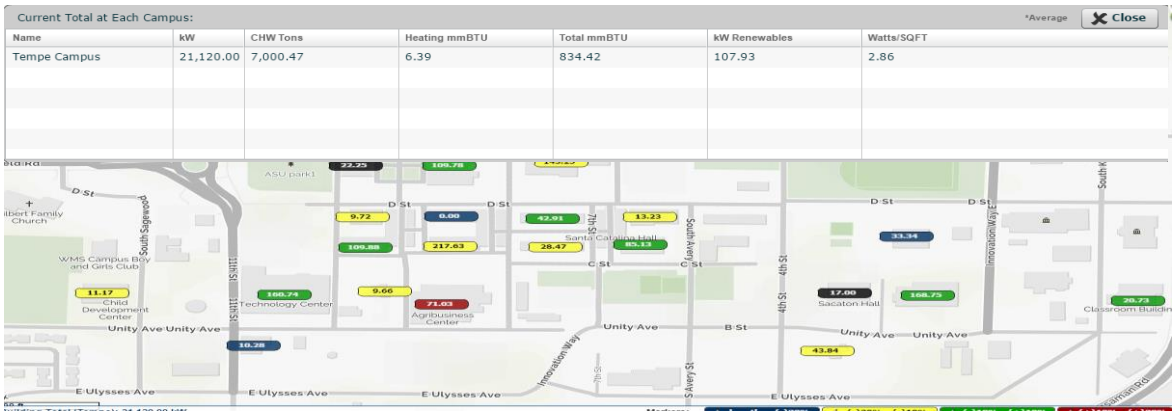

Fig. 3.11 – Online energy consumption of Poly campus

The poly campus power distribution data is fed into the excel file. This excel file is given to the solver file. The system is modeled as a 83 bus sytem with an infinite bus denoted as bus number 1. The smart inverter is modeled as a current source. The current source data is given as RMS magnitude of the current and steady state phasor component of the current. The smart inverter considered is injected into the phase A of bus number 61. The switches sw_a, sw_b and sw_c connect the rest of the buses in the distribution system to infine bus number 1. If the switches are disabled, then the distribution system acts in the islanded mode. The simulation shown below in Fig. 3.12 is performed in the offline mode in MATLAB

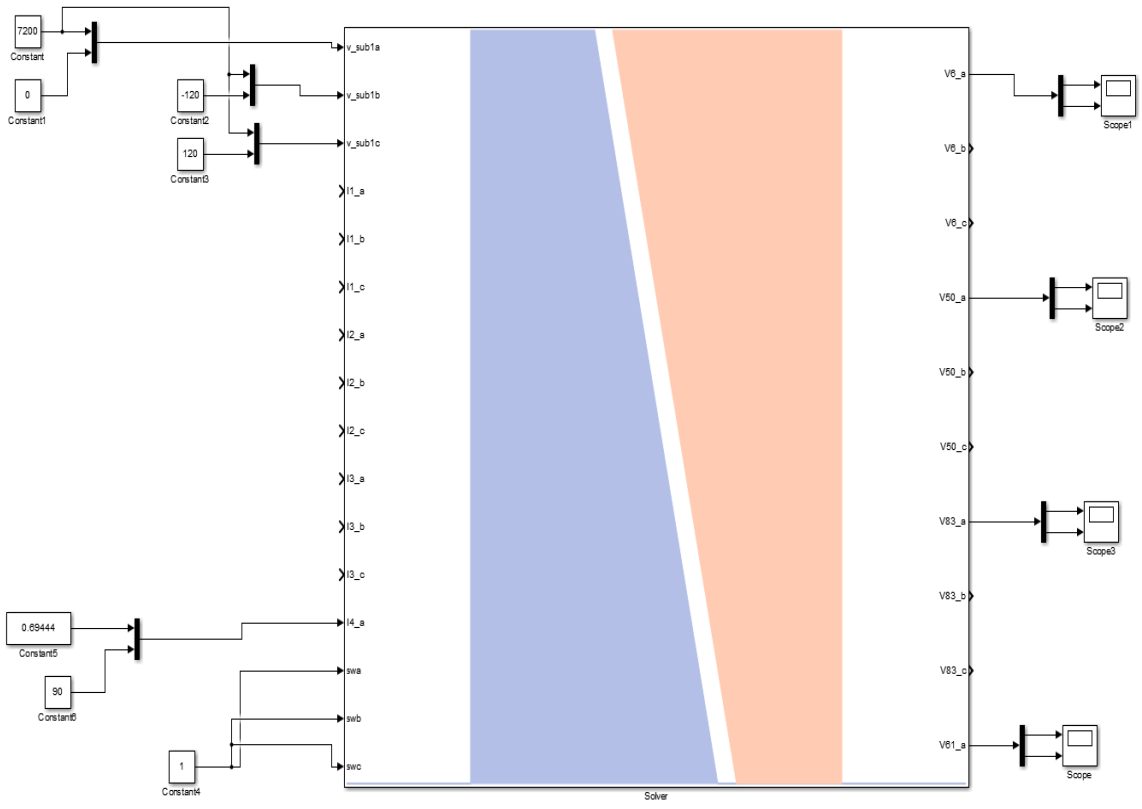

Fig 3.12 – Simulation of Poly campus dstribution system in offline mode in MATLAB

3.4 Smart inverter implemented with Poly campus ditribution system

The smart inverter modeled in PLECS RT box and controller is interfaced with the Poly campus distribution system emulated in the OPAL-RT box. The PLECS-RT box receives four analog inputs, P- real command, Q- reactive power command, v_g- grid voltage reference, θ_v - grid voltage angle. These commands are used to model the smart inverter and the DSP provides the gating signals to the inverter. The smart inverter gives the OPAL-RT model two analog outputs, $\overline{i_L}$ – inductor current reference, θ_i – inductor current angle. These two outputs are combined together which forms the current source input to the OPAL-RT model. Figure 3.13 shows the PLECS model with OPAL RT inputs and outputs.

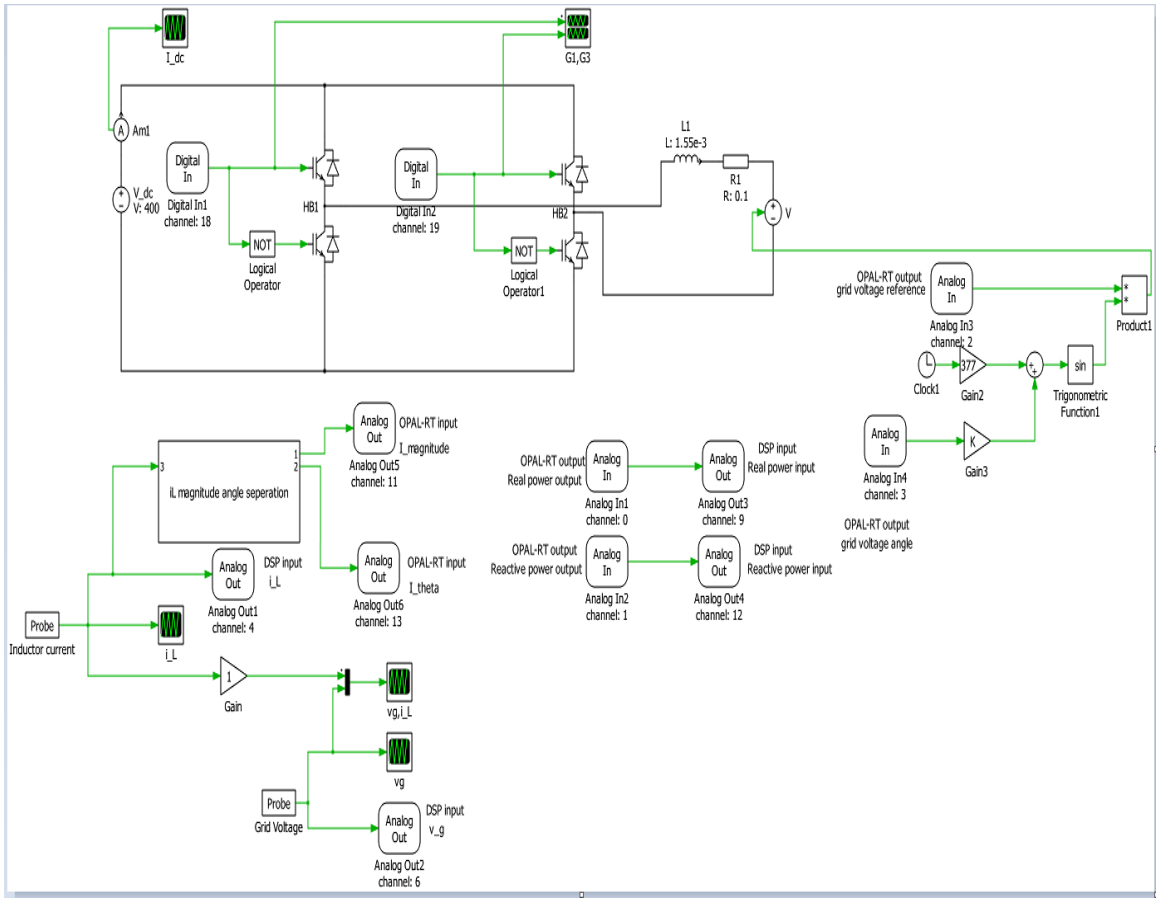

Fig. 3.13 – PLECS implementation of smart inverter with OPAL-RT

The OPAL-RT model is developed for this two input four output system. The various models are shown in Fig 3.14, 3.15 and 3.16.

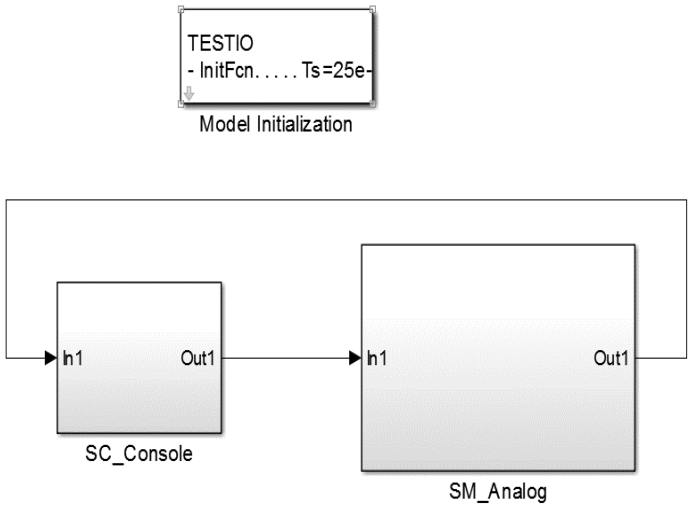

Figure 3.14 – Simplified OPAL-RT model for smart inverter stystem

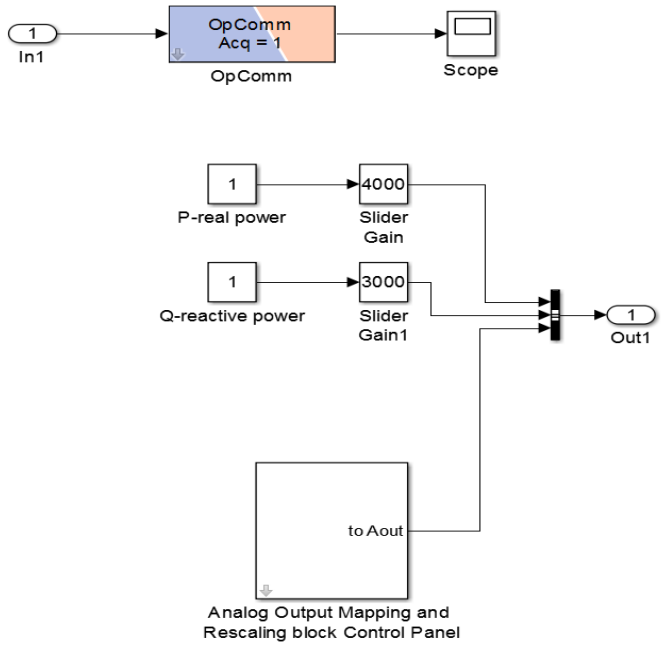

Figure 3.15 – GUI representation of OPAL-RT model for smart inverter system

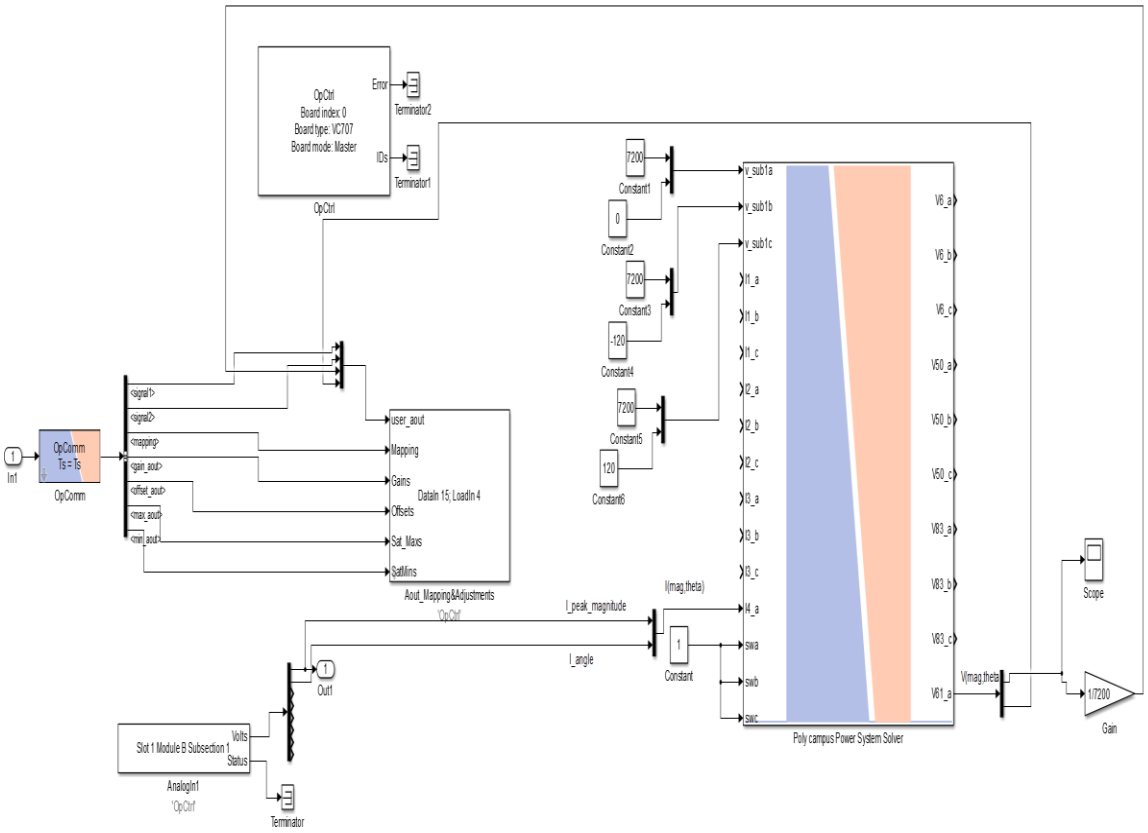

Figure 3.16 – Computation block of OPAL-RT model for smart inverter

The implementation of smart inverter with PLECS-RT and OPAL-RT were shown along with the pictures of simulation models.

CHAPTER 4

RESULTS AND CONCLUSION

Having discussed the implementation of smart inverter in PLECS-RT box and implementation of the distribution system of Poly campus system with the smart inverter in the OPAL-RT simulator, this chapter will discuss about the results and waveforms from smart inverter implementation with PLECS-RT box, OPAL-RT simulator and the DSP.

4.1 Results of DSP coding

The controller design for the smart inverter emulated in the PLECS-RT box was implemented in the DSP TMS320F28377S. The DSP was programmed in C-language with the help of Code Composer Studio (CCS) [51].

The DSP receives four commands from the PLECS software as analog inputs. These inputs are the real power command P, reactive power command Q, grid voltage v_g, and inductor current i_L. The DSP can only send digital outputs. These two outputs are the two gate signals for the inverter switches S1 and S3. The switches S4 and S2 receive the complementary inputs of S1 and S3 from the PLECS software respectively. The gate signals for switches S2 and S4 are not generated from the DSP due to the dead time problem. The PLECS RT box can only have a minimum dead time of 1 μs reducing the duty range from 100% to 96% for a 20KHz switching frequency. Hence, the complementary 'NOT' operator is used for generating gate signals for S2, S4 from S1 and S3 respectively.

Table 4.1 - Inputs and outputs of the DSP

Analog Input	Digital output
1) Real power command, P	1) Gate signal for switch S1
2) Reactive power command, Q	2) Gate signal for switch S3
3) Grid voltage, v_g	
4) Inductor current, i_L	

The procedure for generating gate signals from the analog inputs was explained in the last chapter. With the switching frequency chosen as 20 KHz, the ISR routine for ADC is chosen as the same. Figure 4.1 shows the gate signals of switch S1 and S3 along with the ADC ISR routine observed in the oscilloscope.

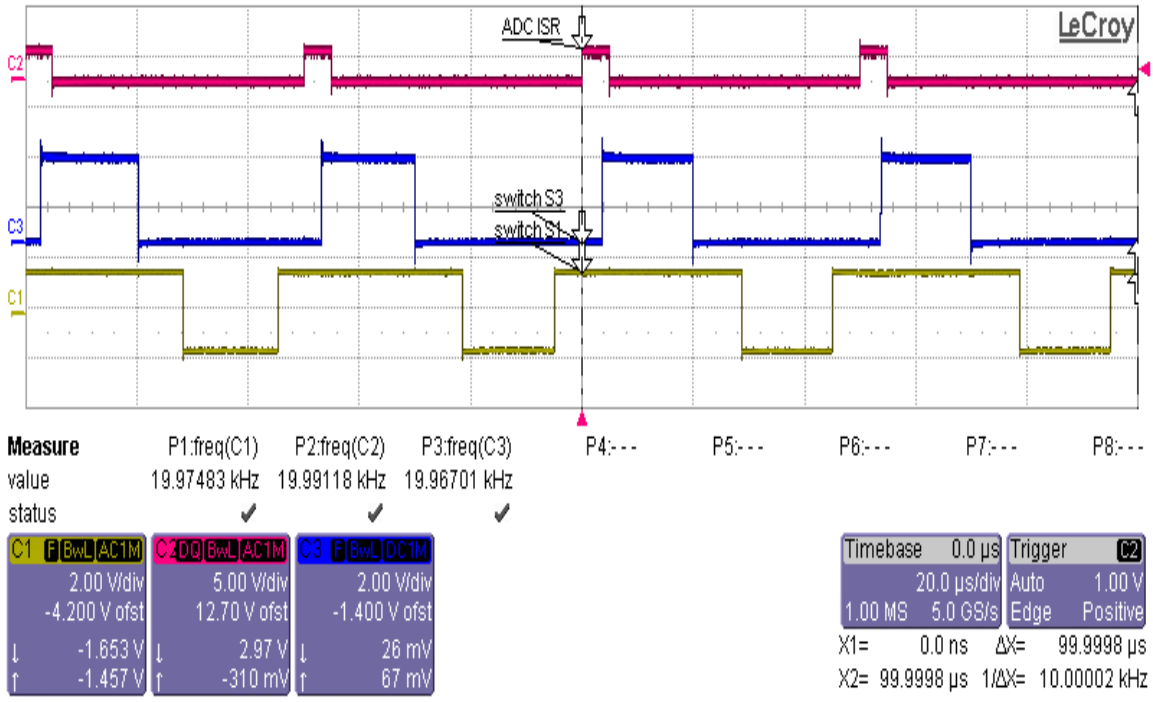

Figure 4.1 – Gate pulses for switches S1, S3 and the ADC ISR routine

The PLL required for generating the current reference to be in phase with the grid voltage is programmed in CCS. It is implemented as a difference equation. The program was verified with a sine wave input. The outputs are cosine wave and the maximum amplitude reference needed for PLL and current reference calculation. Figure 4.2 shows the sine, cosine, and maximum amplitude waveforms verified for the code written in the DSP controller.

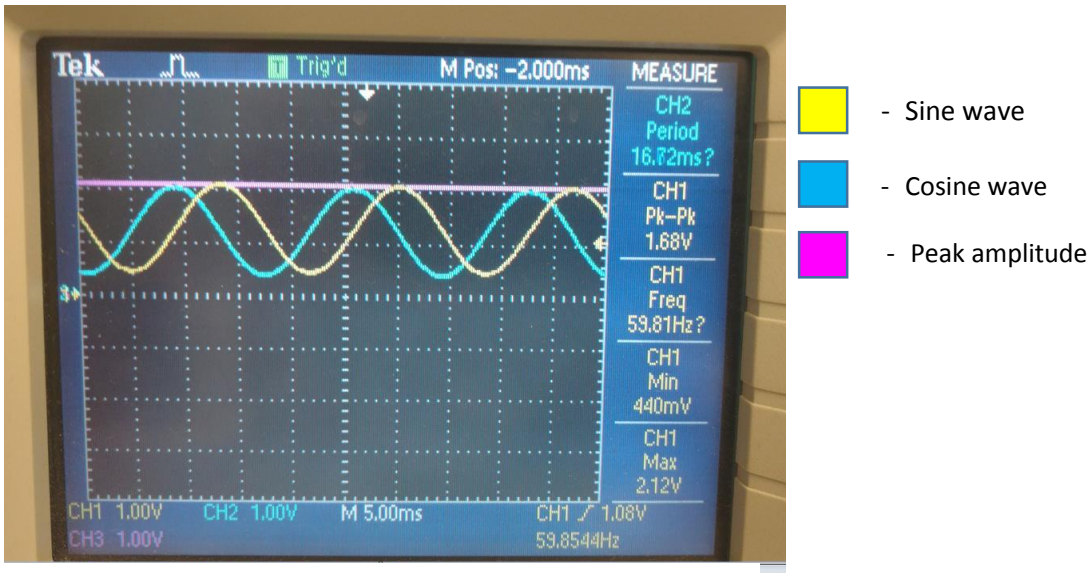

Figure 4.2 – Scope waveforms of Sine, cosine and peak amplitude for PLL verification

4.2 Results of Smart Inverter with PLECS-RT box

The PLECS-RT box connected with DSP controller acts as a smart inverter. The voltage and current waveforms of the inverter emulated in the PLECS-RT box can be observed in the PLECS software. The scaled grid voltage and inductor current waveforms for real power command of 5KW are shown in Figure 4.3.

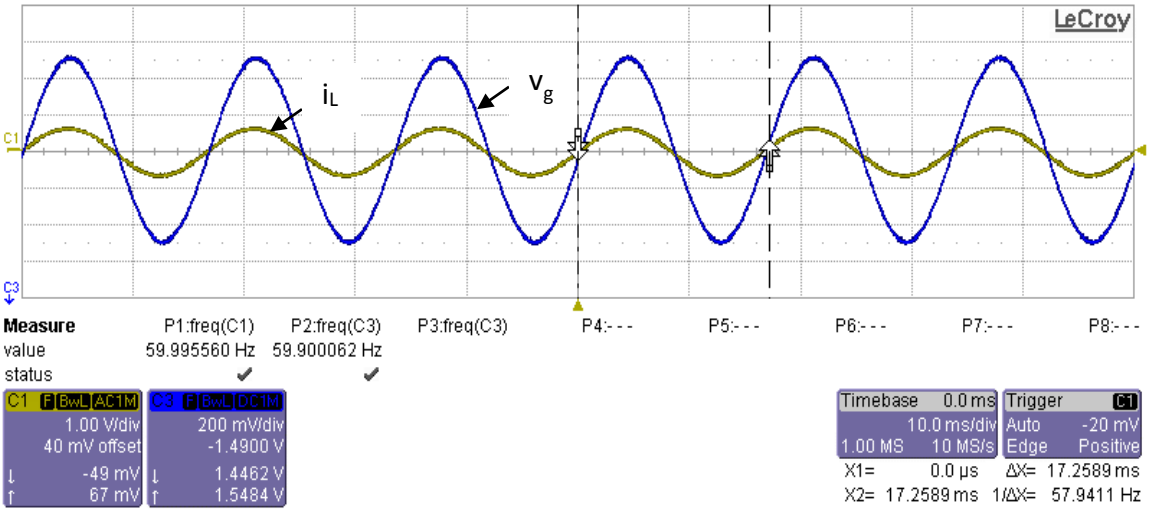

Figure 4.3 – Grid voltage and inductor current for a 5KW power command

The input DC current for the inverter depends on the real and reactive power commands. For a 5 KW real power, the DC current waveform of the inverter is shown in Figure 4.4.

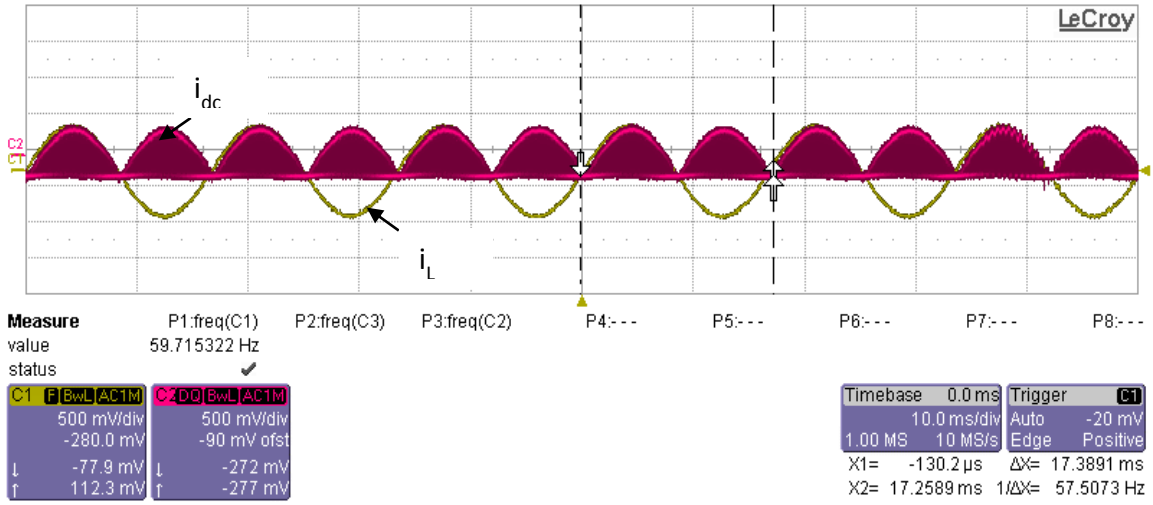

Figure 4.4 – Input current of the inverter for 5 KW real power command

The line voltage V_{AB} for the inverter is shown in Figure 4.5.

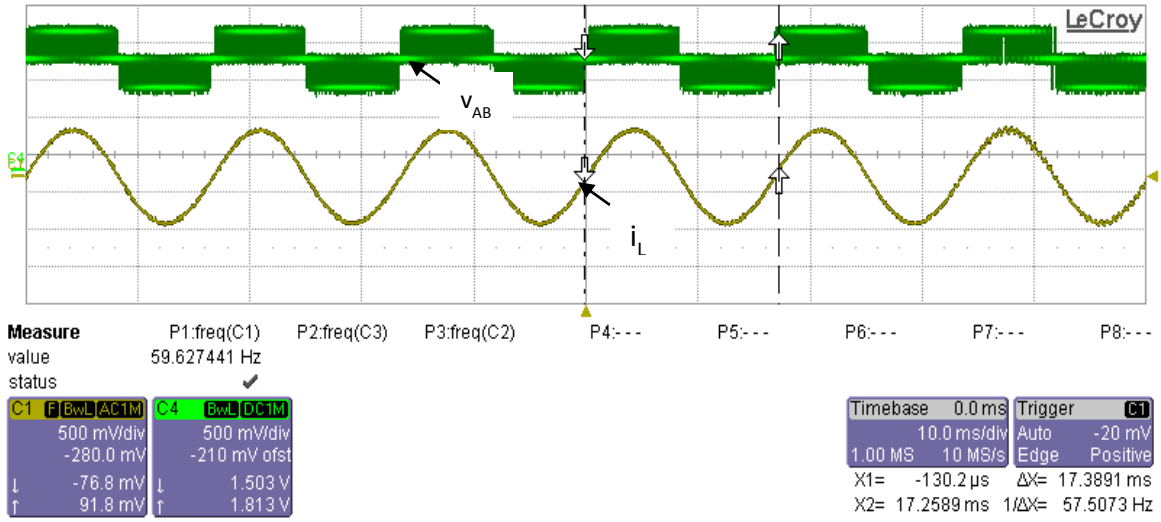

Figure 4.5 – Line voltage V_{AB} of the inverter

There is a current controller implemented within the DSP as a difference equation. The current controller has the input of difference between measured and reference current. The

output generated by the controller is control voltage $v_c(t)$. The performance of the controller is studied by its ability to track values.

The inductor current is given as input to the ADC of DSP. The current waveform is scaled and offset to ensure voltage stays within the ADC voltage limits from 0 to 3V. The scaled waveform is then probed in the digital oscilloscope. The transition is observed in the oscilloscope for change of active power from 2.5KW to 5KW which corresponds to change of current from 14.7A to 29.4A. The current waveform is offset and scaled using the formula used in equation (4.1). The reduced current waveform $i_{L\_reduced}$, which is reduced from i_L along with scale of 1/100 and offset of 1.5V.

$$i_{L\_reduced} = i_L*(1/100) + 1.5 \quad (4.1)$$

The voltage in inductor corresponding to the range of -30A to 30A is reduced 1.22V to 1.52V. The current change from 14.7A to 29.4A corresponding to power command change from 2.5 KW to 5 KW is shown in Figure 4.6

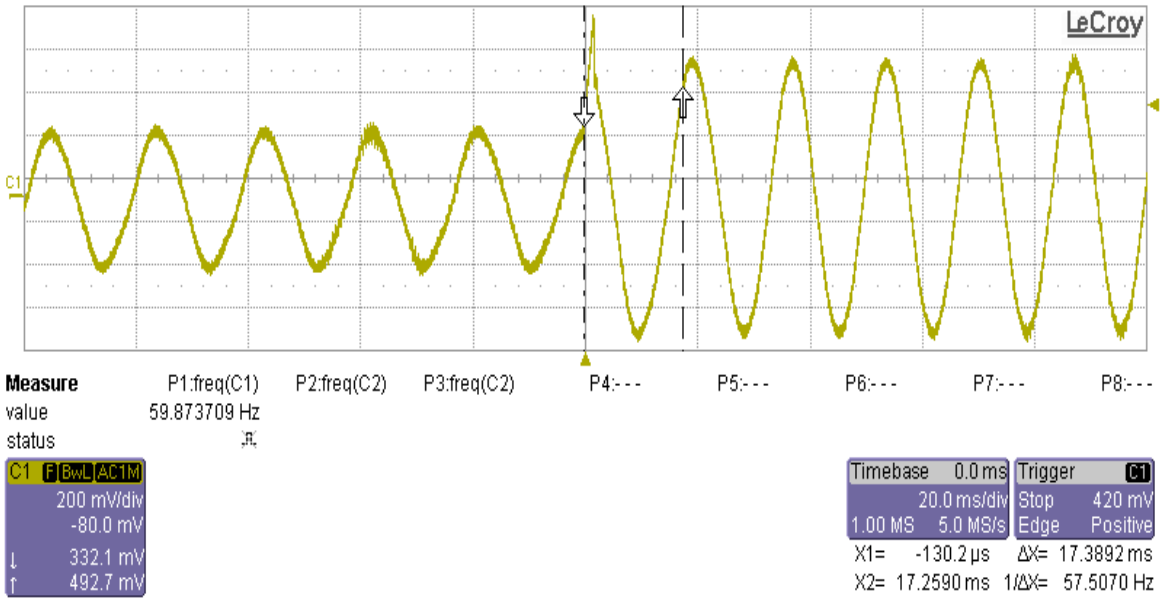

Figure 4.6 – Step change of inductor current from 14.7A to 29.4A

Figure 4.7 shows below the grid voltage, inductor current and input DC current for the inverter delivering 4000W of active power and 3000VAR of reactive power. It can be observed that inductor current is lagging behind the grid voltage and the power is getting delivered to the grid at 0.8 p.f lagging.

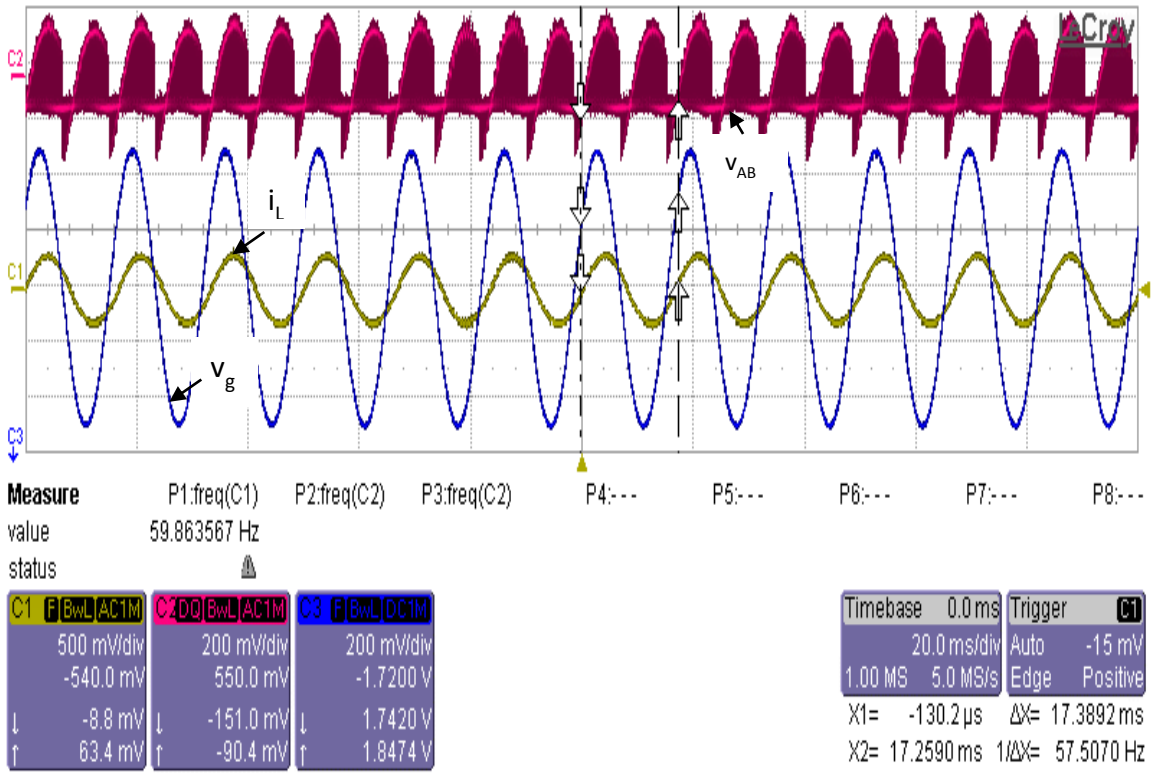

Figure 4.7 showing the grid voltage, inductor current and input DC current for the inverter receiving power command of P=4000W and Q=3000VAR

For the case of P=4000W and Q=-3000VAR, figure 4.9 shows below the grid voltage, inductor current and input DC current for the inverter delivering 4000W of active power and absorbing 3000VAR of reactive power. The P and Q commands are received from the OPAL-RT simulator. The waveforms show the inductor current leading the grid voltage

and the power is getting delivered to the grid from the inverter at the power factor of 0.8 p.f leading. This is shown in Figure 4.8.

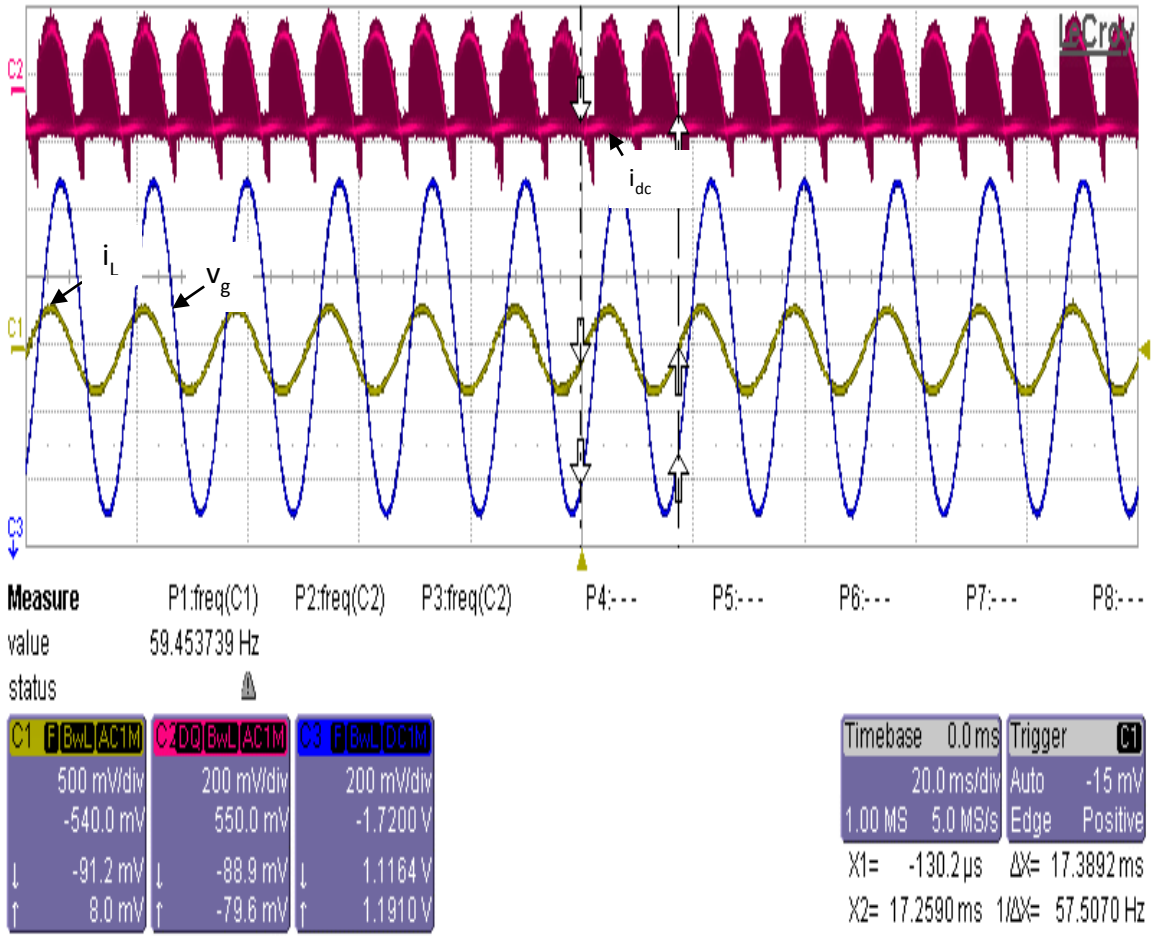

Figure 4.8 – Waveforms of grid voltage, inductor current and input DC current for the inverter receiving power command of P=4000W and Q=-3000VAR

4.3 Results of OPAL-RT integrated with PLECS-RT

The PLECS-RT system which acts as the smart inverter is connected to the OPAL-RT model containing the ePHASORsim model of the Poly campus. This set up is shown in the figure 4.7 below

Figure 4.9 – PLECS –RT box model integrated with the OPAL-RT model

As explained in the previous chapter, the OPAL-RT model has a slider gain option for the real and reactive power commands, which are received as analog inputs by the PLECS-RT box for the smart inverter. The power command change from P=3000W to P=3000W and Q=4000VAR is given with the help of slider gain from the OPAL-RT. This change can be observed as shown in Figure 4.10.

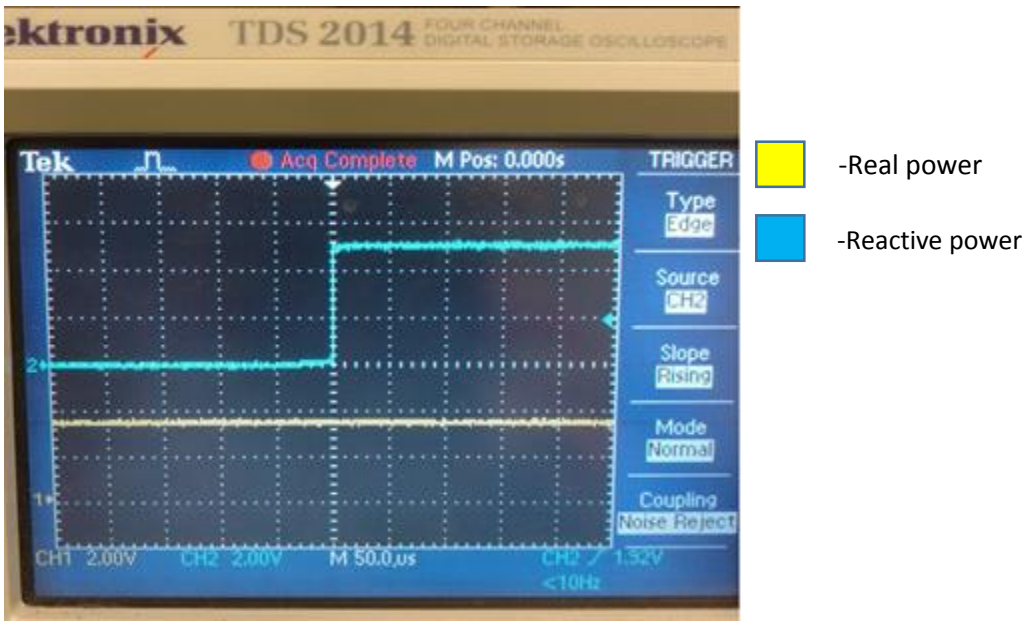

Figure 4.10 – Step change from P=3000W to P=3000W and Q=4000VAR

The corresponding change in the inductor current, input DC current for this step change is observed as shown in Figure 4.11.

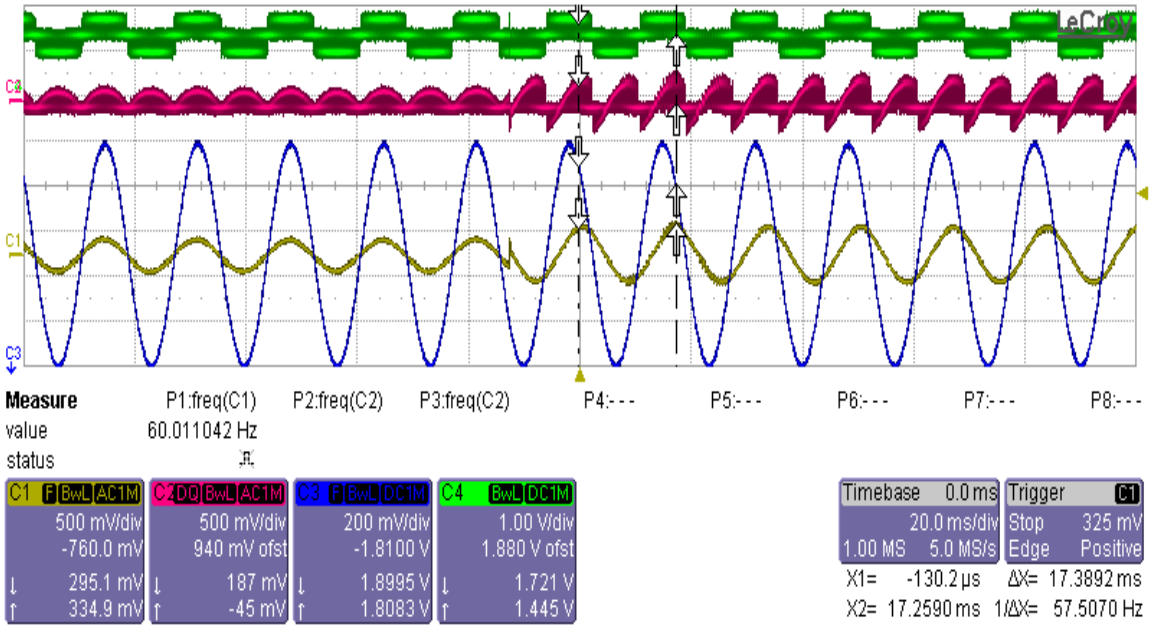

Figure 4.11 - grid voltage, inductor current, line voltage and input DC current waveforms for step change from P=3000W to P=3000W and Q=4000VAR

4.4 Conclusion and future research

PLECS-RT box was used to emulate the power stage with DSP acting as the controller for the power stage. CHIL scheme for smart inverter was implemented. The PLECS-RT inverter system was integrated with the OPAL-RT box which emulated the Poly campus distribution system. The waveforms of the grid voltage, inductor current, line voltage and input DC current were scaled and offset and observed at the oscilloscope for various possible real power and reactive power commands that the PLECS-RT box receives from the OPAL-RT simulator.

Future work includes establishing digital communication between OPAL-RT and PLECS-RT box based on Modbus over TCP/IP. The smart inverter model designed for a single phase system should be extended for emulating three phase inverter switching models and three phase motor drive systems in PLECS-RT and then integrating it with the OPAL-RT.

References:

[1] Philip P. Barker, "Determining the impact of distributed generation on power systems: part 1-radial distribution systems," *Power Engineering Society Summer Meeting*, vol. 3, pp. 1645-1656, July 2000.

[2] "Clean Power Plan for Existing Power Plants | Clean Power Plan | US EPA" available at:
https://www.epa.gov/cleanpowerplan/clean-power-plan-existing-power-plants

[3] "Investment Tax Credits, Rebates & Savings | Department of Energy" available at:
https://energy.gov/savings/business-energy-investment-tax-credit-itc

[4] Adam Sieminski, "AEO2017 Rollout Presentation", Johns Hopkins School of Advanced International Studies January 5, 2017 | Washington, D.C.

[5] "U.S. Solar Market, Executive Summary, 2016 Year in Review" available at:
http://www.seia.org/sites/default/files/Dn4u8Zl5snSMI2016YIR.pdf

[6] S. B. Kjaer, J. K. Pedersen and F. Blaabjerg, "A review of single-phase grid-connected inverters for photovoltaic modules," in *IEEE Transactions on Industry Applications*, vol. 41, no. 5, pp. 1292-1306, Sept.-Oct. 2005.

[7] Arkanatha Sastry Manchanahalli Ranganatha, "Solar Micro Inverter Modeling and Reliability," 2015, MS dissertation, Arizona State University.

[8] J. Flicker, G. Tamizhmani, M. K. Moorthy, R. Thiagarajan and Raja Ayyanar, "Accelerated Testing of Module-Level Power Electronics for Long-Term Reliability," in *IEEE Journal of Photovoltaics*, vol. 7, no. 1, pp. 259-267, Jan. 2017.

[9] J. M. A. Myrzik and M. Calais, "String and module integrated inverters for single-phase grid connected photovoltaic systems—A review," in *Proc. IEEE Bologna PowerTech Conf.*, vol. 2, 2003, pp. 430–437.

[10] S. Weckx, C. Gonzalez and J. Driesen, "Combined Central and Local Active and Reactive Power Control of PV Inverters," in *IEEE Transactions on Sustainable Energy*, vol. 5, no. 3, pp. 776-784, July 2014.

[11] Rodriguez J, Lai JS, Peng FZ (2002), "Multilevel inverters: a survey of topologies, controls, and applications," in IEEE Trans Ind Electron 49:724–738

[12] Malinowski M, Gopakumar K, Rodriguez J, Pérez MA (2010), "A survey on cascaded multilevel inverters," IEEE Trans Ind Electron 57:2197–2206

[13] Lisa Schwartz, "Distributed generation in Oregon: overview, regulatory barriers and recommendations," Oregon Public Utility Commission Report, February 2005.

[14] Tales M. de Britto, "Distributed generation impacts on the coordination of protection systems in distribution networks," *Transmission and Distribution Conference and Exposition: Latin America*, pp. 623-628, November 2004.

[15] Eurelectric, "Active distribution system management—a key tool for the smooth integration of distributed generation," *Eurelectric*, Feb. 2013

[16] F. A. Viawan and D. Karlsson, "Voltage and reactive power control in system with synchronous machine-based distributed generation," *IEEE Trans. Power Del.*, vol. 23, no. 2, pp. 1079-1087, Apr. 2008.

[17] M. Venmathi, J. Vargese, L. Ramesh and E. S. Percis, "Impact of grid connected distributed generation on voltage sag," *Sustainable Energy and Intelligent Systems*, Chennai, pp. 91-96, 2011.

[18] L. Kojovic, "Impact DG on voltage regulation," *IEEE Power Engineering Society Summer Meeting*, vol. 1, pp. 97-102, 2001.

[19] L. F. Ochoa and G. P. Harrison, "Minimizing energy losses: optimal accommodation and smart operation of renewable distributed generation," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 198-205, Feb. 2011.

[20] T. Ackermann and V. Knyazkin, "Interaction between distributed generation and the distribution network: operation aspects," *IEEE Transmission and Distribution Conference and Exhibition*, vol. 2, pp. 1357-1362, 2002.

[20] G.C. Cornfield, "Definition and measurement of voltage flicker in lighting," *IEE Colloquium on Electronics in Power Systems Measurement*, pp. 1-4, Aug. 2002.

[21] Y. W. Li and J. He, "Distribution system harmonic compensation methods: an overview of DG-interfacing inverters," IEEE Industrial Electronics Magazine, vol. 8, no. 4, pp. 18-31, Dec. 2014.

[22] E. Vasanasong and E. D. Spooner, "The effect of net harmonic currents produced by numbers of the Sydney Olympic village's PV systems on the power quality of local electrical network," *International Conference on Power System Technology*, vol. 2, pp. 1001-1006, 2002.

[23] V. Khadkikar, R. K. Varma, R. Seethapathy, A. Chandra and H. Zeineldin, "Impact of distributed generation penetration on grid current harmonics considering non-linear loads," *Power Electronics for Distributed Generation Systems*, pp. 608-614, 2012.

[24] IEEE, “IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems," IEEE Standard 519-1992, pp. 1-112, April 1993.

[25] S. Boljevic and M. F. Conlon, "The contribution to distribution network short-circuit current level from the connection of distributed generation," *43rd International Universities Power Engineering Conference*, pp. 1-6, 2008.

[26] S. Boljevic and M. F. Conlon, "Fault current level issues for urban distribution network with high penetration of distributed generation," *6th International Conference on the European Energy Market*, pp. 1-6, 2009.

[27] A. Sinclair, D. Finney, D. Martin and P. Sharma, "Distance protection in distribution systems: how It assists with integrating distributed resources," *IEEE Transactions on Industry Applications*, vol. 50, no. 3, pp. 2186-2196, May 2014.

[28] F. Coffele, C. Booth, A. Dyśko and G. Burt, "Quantitative analysis of network protection blinding for systems incorporating distributed generation," *IET Generation, Transmission & Distribution*, vol. 6, no. 12, pp. 1218-1224, December 2012.

[29] M. Ebrahimi, S. A. Khajehoddin and M. Karimi-Ghartemani, "Fast and Robust Single-Phase DQ Current Controller for Smart Inverter Applications," in *IEEE Transactions on Power Electronics*, vol. 31, no. 5, pp. 3968-3976, May 2016.

[30] IEEE Draft Recommended Practice for Establishing Methods and Procedures that Provide Supplemental Support for Implementation Strategies for Expanded use of IEEE standard 1547, pp. 1–176, Sep. 2014.

[31] Advanced Functions for DER Inverters Modeled in IEC 61850-90-7, pp. 1–66, 2012.

[32] EPRI, “Common functions for smart inverters, version 3,” Electric Power Research Institute, Palo Alto,CA, USA, 2014.

[33] “California electric rule NO. 21 generating facility interconnections,” pp. 1–217, 2015.

[34] California Energy Commission. (2014). Recommendations for updating the technical requirements for inverters in distributed energy resources, pp. 1–91. [Online]. Available: http://www.energy.ca.gov

[35] Kathie Zipp, “What is a smart solar inverter?”, Solar Power World. Available at: http://www.solarpowerworldonline.com/2014/01/smart-solar-inverter/

[36] “*IEC 61850 Object Models for Photovoltaic, Storage and Other DER Inverters”*, IEC 61850-90-7, draft version 27a

[37] “SunSpec Inverter Models, SunSpec Alliance Interoperability Specification”, sunspec.org. Available at: http://sunspec.org/wp-content/uploads/2015/06/SunSpec-Inverter-Models-12020.pdf

[38] M. D. Omar Faruque *et al*., "Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis," in *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 2, pp. 63-73, June 2015.

[39] V. Vittal and R. Ayyanar, Grid Integration and Dynamic Impact of Wind Energy. New York, NY, USA: Springer, 2013.

[40] R. D. Middlebrook and S. Cuk, "A general unified approach to modelling switching-converter power stages," *1976 IEEE Power Electronics Specialists Conference*, Cleveland, OH, 1976, pp. 18-34.

[41] H. Cha and T. K. Vu, "Comparative analysis of low-pass output filter for single-phase grid-connected Photovoltaic inverter," *2010 Twenty-Fifth Annual IEEE APEC*, Palm Springs, CA, 2010, pp. 1659-1665.

[42] X. Mao, R. Ayyanar and H. K. Krishnamurthy, "Optimal Variable Switching Frequency Scheme for Reducing Switching Loss in Single-Phase Inverters Based on Time-Domain Ripple Analysis," in *IEEE Transactions on Power Electronics*, vol. 24, no. 4, pp. 991-1001, April 2009.

[43] IEEE 1547 Standard for Interconnecting Distributed Resources with Electric Power Systems

[44] S. Parhizi, H. Lotfi, A. Khodaei and S. Bahramirad, "State of the Art in Research on Microgrids: A Review," in *IEEE Access*, vol. 3, no., pp. 890-925, 2015.

[45] R. Teodorescu, F. Blaabjerg, M. Liserre and P. C. Loh, "Proportional-resonant controllers and filters for grid-connected voltage-source converters," in *IEE Proceedings - Electric Power Applications*, vol. 153, no. 5, pp. 750-762, September 2006.

[46] PLECS, electrical engineering software, available at: http://www.plexim.com/

[47] TMS320F28377S Single-Core Delfino Microcontroller, available at: http://www.ti.com/product/TMS320F28377S

[48] PLECS RT Box, Hardware-in-the-Loop (HIL) platform, available at: http://www.plexim.com/sites/default/files/flyers/flyer_hil_a4.pdf

[49] OPAL-RT technologies, available at: http://www.opal-rt.com/

[50] OPAL-RT user guide, available at:
https://www.opal-rt.com/wp-content/themes/enfold-opal/pdf/L00161_0407.pdf

[51] Code Composer Studio, available at:
http://www.ti.com/tool/ccstudio

APPENDIX A

DSP CODE FOR CONTROLLER IMPLEMENTATION OF SMART INVERTER

```
//###########################################################################
#####
// FILE:   epwm_updown_aq_cpu01.c
// TITLE:  Action Qualifier Module - Using up/down count.
//
//! <h1> EPWM Action Qualifier (epwm_updown_aq)</h1>
//!
//! This program designs the controller for smart inverter with setting of parameters
//! For epwm and adc, with definitions of pwm, adc and interrupt functions
//! The compare values CMPA and CMPB are modified within the adc's ISR.
//
//###########################################################################
#####
// $TI Release: F2837xS Support Library v160 $
// $Copyright: Copyright (C) 2014-2015 Texas Instruments Incorporated -
//            http://www.ti.com/ ALL RIGHTS RESERVED $
//###########################################################################
#####

#include "F28x_Project.h"     // Device Headerfile and Examples Include File


typedef struct
{
   volatile struct EPWM_REGS *EPwmRegHandle;
}EPWM_INFO;

// Prototype statements for functions found within this file.

void InitEPwm2Example(void);
void InitEPwm6Example(void);
void InitAdca(void);
interrupt void EPWM2_ISR(void);


// Global variables used in this example

EPWM_INFO epwm2_info;

// Configure the period for each timer

#define EPWM2_TIMER_TBPRD  1250  // Period register
#define EPWM2_CMPA     500
```

```
#define pi                              3.14159
#define TMR0       300000      // period of timer zero in micro seconds

int vg = 0;                                                 // pole A control signal
int ig = 0;
//int count = 0;
float ig_con=0;
float ig_err=0;
float ig_ph=0;
float ig_ref=0;
//float ig_ref1;
float ig_ff=0;
float vg_ff=0;
float vg_con=0;
float vg_ref;
float vg_ref1;
float vg_error;
float dA1=0;
float dA;
float dB;

//volatile int dB = 0;                                          // pole B control signal


// To keep track of which way the compare value is moving
#define EPWM_CMP_UP   1
#define EPWM_CMP_DOWN 0

void main(void)
{
// Step 1. Initialize System Control:
// PLL, WatchDog, enable Peripheral Clocks
// This example function is found in the F2837xS_SysCtrl.c file.
   InitSysCtrl();

// Step 2. Initialize GPIO:
// This example function is found in the F2837xS_Gpio.c file and
// illustrates how to set the GPIO to it's default state.
//    InitGpio();

   EALLOW;
                GpioCtrlRegs.GPAPUD.bit.GPIO19 = 1;   // disable pullup on GPIO19

                GpioCtrlRegs.GPAMUX2.bit.GPIO19 = 0; //Selecting pin from Mux
                GpioCtrlRegs.GPADIR.bit.GPIO19 = 1;  // GPIO19 = output
```

```
   EDIS;

//allocate PWM1, PWM2 and PWM3 to CPU1
//    CpuSysRegs.CPUSEL0.bit.EPWM1 = 0;
//    CpuSysRegs.CPUSEL0.bit.EPWM2 = 0;
//    CpuSysRegs.CPUSEL0.bit.EPWM3 = 0;

// enable PWM1, PWM2 and PWM3

   CpuSysRegs.PCLKCR2.bit.EPWM2=1;


// For this case just init GPIO pins for ePWM1, ePWM2, ePWM3
// These functions are in the F28M36x_EPwm.c file
         InitEPwm2Gpio();
         InitEPwm6Gpio();

// Step 3. Clear all interrupts and initialize PIE vector table:
// Disable CPU interrupts
   DINT;

// Initialize the PIE control registers to their default state.
// The default state is all PIE interrupts disabled and flags
// are cleared.
// This function is found in the F2837xS_PieCtrl.c file.
   InitPieCtrl();

// Disable CPU interrupts and clear all CPU interrupt flags:
   IER = 0x0000;


// Initialize the PIE vector table with pointers to the shell Interrupt
// Service Routines (ISR).
// This will populate the entire table, even if the interrupt
// is not used in this example.  This is useful for debug purposes.
// The shell ISR routines are found in F2837xS_DefaultIsr.c.
// This function is found in F2837xS_PieVect.c.
   InitPieVectTable();

// Interrupts that are used in this example are re-mapped to
// ISR functions found within this file.


         InitCpuTimers();                    // Init timer structure
```

```
        //ConfigCpuTimer(&CpuTimer0, 100, TMR0); // Configure Timer specified
period (in microseconds) and start


// For this example, only initialize the ePWM

        EALLOW;
        CpuSysRegs.PCLKCR0.bit.TBCLKSYNC =0;

        EDIS;

   InitAdca();
        InitEPwm2Example();
        InitEPwm6Example();

        EALLOW;
        CpuSysRegs.PCLKCR0.bit.TBCLKSYNC =1;

        EDIS;


// Enable CPU INT3 which is connected to EPWM1-3 INT:
        IER = IER = 0x0063;;
        IFR = 0x0000;

// Enable EPWM INTn in the PIE: Group 3 interrupt 1-3
        PieCtrlRegs.PIEIER3.bit.INTx1 = 1;
        PieCtrlRegs.PIEIER3.bit.INTx2 = 1;
        PieCtrlRegs.PIEIER3.bit.INTx3 = 1;
        PieCtrlRegs.PIEIER3.bit.INTx6 = 1;

// Enable global Interrupts and higher priority real-time debug events:
   EINT;  // Enable Global interrupt INTM
   ERTM;  // Enable Global realtime interrupt DBGM

   PieCtrlRegs.PIEACK.all = 0xFFFF;
// Step 5. IDLE loop. Just sit and loop forever (optional):
   for(;;)
   {
      asm ("         NOP");

   }
}
void InitEPwm2Example()
{
```

```
        asm(" EALLOW");                                 // Enable EALLOW
protected register access


        // Configure the prescaler to the ePWM modules.  Max ePWM input clock is 100
MHz.
                ClkCfgRegs.PERCLKDIVSEL.bit.EPWMCLKDIV = 1;             //
EPWMCLK divider from PLLSYSCLK.  0=/1, 1=/2
                // Must disable the clock to the ePWM modules if you want all ePWM
modules synchronized.
                CpuSysRegs.PCLKCR0.bit.TBCLKSYNC = 0;

                asm(" EDIS");                                   // Disable EALLOW
protected register access


        //---------------------------------------------------------------------
        //--- Configure ePWM2 to trigger ADC SOCA at a 50 kHz rate
        //---------------------------------------------------------------------
                asm(" EALLOW");                                                 //
Enable EALLOW protected register access
                DevCfgRegs.SOFTPRES2.bit.EPWM2 = 1;         // ePWM2 is reset
                DevCfgRegs.SOFTPRES2.bit.EPWM2 = 0;         // ePWM2 is released
from reset
                asm(" EDIS");                                           // Disable
EALLOW protected register access

                EPwm2Regs.TBCTL.bit.CTRMODE = 0x3;          // Disable the timer

                EPwm2Regs.TBCTL.all = 0xC033;               // Configure timer
control register
        // bit 15-14     11:     FREE/SOFT, 11 = ignore emulation suspend
        // bit 13        0:      PHSDIR, 0 = count down after sync event
        // bit 12-10     000:    CLKDIV, 000 => TBCLK = HSPCLK/1
        // bit 9-7       000:    HSPCLKDIV, 000 => HSPCLK = EPWMCLK/1
        // bit 6         0:      SWFSYNC, 0 = no software sync produced
        // bit 5-4       11:     SYNCOSEL, 11 = sync-out disabled
        // bit 3         0:      PRDLD, 0 = reload PRD on counter=0
        // bit 2         0:      PHSEN, 0 = phase control disabled
        // bit 1-0       11:     CTRMODE, 11 = timer stopped (disabled)

                EPwm2Regs.TBCTR = 0x0000;                               // Clear timer
counter
                EPwm2Regs.TBPRD = 2500;// Set timer period
```

```
        EPwm2Regs.TBPHS.bit.TBPHS = 0x0000;           // Set timer phase

        EPwm2Regs.ETPS.all = 0x0100;                  // Configure SOCA
    // bit 15-14    00:    EPWMxSOCB, read-only
    // bit 13-12    00:    SOCBPRD, don't care
    // bit 11-10    00:    EPWMxSOCA, read-only
    // bit 9-8      01:    SOCAPRD, 01 = generate SOCA on first event
    // bit 7-4      0000:  reserved
    // bit 3-2      00:    INTCNT, don't care
    // bit 1-0      00:    INTPRD, don't care

        EPwm2Regs.ETSEL.all = 0x0A00;                 // Enable SOCA to
ADC
    // bit 15       0:     SOCBEN, 0 = disable SOCB
    // bit 14-12    000:   SOCBSEL, don't care
    // bit 11       1:     SOCAEN, 1 = enable SOCA
    // bit 10-8     010:   SOCASEL, 010 = SOCA on PRD event
    // bit 7-4      0000:  reserved
    // bit 3        0:     INTEN, 0 = disable interrupt
    // bit 2-0      000:   INTSEL, don't care

        EPwm2Regs.TBCTL.bit.CTRMODE = 0x0;            // Enable the timer in
count up mode
        EPwm2Regs.CMPA.bit.CMPA = 200;
       // Set actions
       EPwm2Regs.AQCTLA.bit.CAU= 0x0002;
       EPwm2Regs.AQCTLA.bit.CAD= 0x0001;

    //-----------------------------------------------------------------
    //--- Enable the clocks to the ePWM module.
    //--- Note: this should be done after all ePWM modules are configured
    //--- to ensure synchronization between the ePWM modules.
    //-----------------------------------------------------------------
        asm(" EALLOW");                                       //
Enable EALLOW protected register access
        CpuSysRegs.PCLKCR0.bit.TBCLKSYNC = 1;         // TBCLK to ePWM
modules enabled
        asm(" EDIS");                                         // Disable
EALLOW protected register access


}

void InitAdca()
{
```

```
        asm(" EALLOW");                                 // Enable EALLOW
protected register access

//--- Reset the ADC.  This is good programming practice.
        DevCfgRegs.SOFTPRES13.bit.ADC_A = 1;// ADC is reset
        DevCfgRegs.SOFTPRES13.bit.ADC_A = 0;// ADC is released from reset

//--- Configure the ADC base registers
        AdcaRegs.ADCCTL1.all = 0x0004;          // Main ADC configuration
// bit 15-14    00:    reserved
// bit 13       0:     ADCBSY, ADC busy, read-only
// bit 12       0:     reserved
// bit 11-8     0's:   ADCBSYCHN, ADC busy channel, read-only
// bit 7        0:     ADCPWDNZ, ADC power down, 0=powered down, 1=powered up
// bit 6-3      0000:  reserved
// bit 2        1:     INTPULSEPOS, INT pulse generation, 0=start of conversion, 1=end of
conversion
// bit 1-0      00:    reserved

        AdcaRegs.ADCCTL2.all = 0x0006;          // ADC clock configuration
// bit 15-8     0's:   reserved
// bit 7        0:     SIGNALMODE, configured by AdcSetMode() below to get calibration
correct
// bit 6        0:     RESOLUTION, configured by AdcSetMode() below to get calibration
correct
// bit 5-4      00:    reserved
// bit 3-0      0110:  PRESCALE, ADC clock prescaler.  0110=CPUCLK/4

        AdcaRegs.ADCBURSTCTL.all = 0x0000;
// bit 15       0:     BURSTEN, 0=burst mode disabled, 1=burst mode enabled
// bit 14-12    000:   reserved
// bit 11-8     0000:  BURSTSIZE, 0=1 SOC converted (don't care)
// bit 7-6      00:    reserved
// bit 5-0      000000: BURSTTRIGSEL, 00=software only (don't care)

//--- Call AdcSetMode() to configure the resolution and signal mode.
//    This also performs the correct ADC calibration for the configured mode.
   AdcSetMode(ADC_ADCA, ADC_RESOLUTION_12BIT,
ADC_SIGNALMODE_SINGLE);

//--- SOC0 configuration
AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 7;// Trigger using ePWM2-ADCSOCA
AdcaRegs.ADCSOC0CTL.bit.CHSEL = 0;  // Convert channel ADCINA0 (Ch. 0)
        AdcaRegs.ADCSOC0CTL.bit.ACQPS = 19;// Acquisition window set to
(19+1)=20 cycles (100 ns with 200 MHz SYSCLK)
```

```
        //--- SOC1 configuration
        AdcaRegs.ADCSOC1CTL.bit.TRIGSEL = 7;                // Trigger using
ePWM2-ADCSOCA
   AdcaRegs.ADCSOC1CTL.bit.CHSEL = 1;                       // Convert channel
ADCINA1 (Ch. 1)
   AdcaRegs.ADCSOC1CTL.bit.ACQPS = 19;                      // Acquisition window
set to (19+1)=20 cycles (100 ns with 200 MHz SYSCLK)

        AdcaRegs.ADCINTSOCSEL1.bit.SOC0 = 0;                // No ADC interrupt
triggers SOC0 (TRIGSEL field determines trigger)
        AdcaRegs.ADCINTSOCSEL1.bit.SOC1 = 0;                // No ADC interrupt
triggers SOC0 (TRIGSEL field determines trigger)
        AdcaRegs.ADCSOCPRICTL.bit.SOCPRIORITY = 0;          // All SOCs handled
in round-robin mode

//--- ADCA1 interrupt configuration
        AdcaRegs.ADCINTSEL1N2.bit.INT1CONT = 1;             // Interrupt pulses
regardless of flag state
        AdcaRegs.ADCINTSEL1N2.bit.INT1E = 1;           // Enable the interrupt in the
ADC
        AdcaRegs.ADCINTSEL1N2.bit.INT1SEL = 0;              // EOC0 triggers the
interrupt

//--- Enable the ADC interrupt
        PieCtrlRegs.PIEIER1.bit.INTx1 = 1; // Enable ADCA1 interrupt in PIE group 1
        IER |= 0x0001;                                                          //
Enable INT1 in IER to enable PIE group

//--- Finish up
        AdcaRegs.ADCCTL1.bit.ADCPWDNZ = 1;                  // Power up the ADC
        DELAY_US(1000);                     // Wait 1 ms after power-up before using the
ADC
        asm(" EDIS");           // Disable EALLOW protected register access

} // end InitAdc()

void InitEPwm6Example()
{

   // Setup TBCLK
   EPwm6Regs.TBPRD = EPWM2_TIMER_TBPRD;        // Set timer period 801
TBCLKs
   EPwm6Regs.TBPHS.bit.TBPHS = 0x0000;         // Phase is 0
   EPwm6Regs.TBCTR = 0x0000;                   // Clear counter
```

```
   // Set Compare values
   EPwm6Regs.CMPA.bit.CMPA = dA*1250;    // Set compare A value
   EPwm6Regs.CMPB.bit.CMPB = dB*1250;    // Set compare B value

   // Set Compare B value

   // Setup counter mode
   EPwm6Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Count up and
down
   EPwm6Regs.TBCTL.bit.PHSEN = TB_DISABLE;       // Disable phase loading
   EPwm6Regs.TBCTL.bit.HSPCLKDIV = 0;      // Clock ratio to SYSCLKOUT
   EPwm6Regs.TBCTL.bit.CLKDIV = 0;

   // Setup shadowing
   EPwm6Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
   EPwm6Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
   EPwm6Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // Load on Zero
   EPwm6Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO;

   // Set actions
   EPwm6Regs.AQCTLA.bit.CAU= 0x0001;
   EPwm6Regs.AQCTLA.bit.CAD= 0x0002;
   EPwm6Regs.AQCTLB.bit.CBU= 0x0001;
   EPwm6Regs.AQCTLB.bit.CBD= 0x0002;

   //EPwm2Regs.AQCTLA.bit.CBU= 0x0000;
   //EPwm2Regs.AQCTLA.bit.CBD= 0x0000;
   //EPwm2Regs.AQCTLB.bit.CAU= 0x0000;
   //EPwm2Regs.AQCTLB.bit.CAD= 0x0000;
   // Set Interrupts
   EPwm6Regs.ETPS.bit.INTPRD= ET_1ST;
   EPwm6Regs.ETSEL.bit.INTSEL = 2;
   EPwm6Regs.ETSEL.bit.INTEN = 1; // enabled


}


// FILE:   current.c
// TITLE:  current controller with difference equation


#include "math.h"
#include "IQmathLib.h"
```

```
#define a 0.02836
#define b 0.004446
#define c -0.02392
#define d 0.9257
#define e 0.07429
#define pi 3.14159

static float       yn;
//static float    xn = 0;
static float    xn_1 = 0;
static float       yn_1 = 0;
static float    xn_2 = 0;
static float       yn_2 = 0;

float iL_ztf(float xn)
{
        //if(xn > 5) xn=0;
                //if(xn < -5) xn=0;
        yn = 0;                                 // clear accumulator
        yn += a*xn + b*xn_1 + c*xn_2;
        yn += d*yn_1 + e*yn_2;
        xn_2 = xn_1;                            // x(n-2) = x(n-1)
        xn_1 = xn;                              // x(n-1) = x(n)
        yn_2 = yn_1;                            // y(n-2) = y(n-1)
   yn_1 = yn;                                   // y(n-1) = y(n)

        return yn;
}
```