POLITECNICO DI MILANO

LAUREA MAGISTRALE IN
ELECTRICAL ENGINEERING

TRACK: SMART GRID

CONTROL OF GRID INTERFACE INVERTERS FOR DISTRIBUTED POWER SYSTEM STABILIZATION

Master thesis of:
Muhammad Ahsan Idrees

Supervisor:
Prof. Alberto Berizzi

Co-supervisor:
Dr. Agamy Mohammed

Academic Year 2018-2019

"Stop acting so small. You are the universe in ecstatic motion and What you Seek is Seeking you"

- Jalal ad-Din Muhammad Rumi

Abstract

POWER Systems are having a breakaway and Distributed Generators (DGs) utilizing Renewable Energy Sources (RES) are a promising solution in achieving more sustainable supply of electricity. However, with the increase in number of Renewable Energy Sources based Distributed Generation, the control of the system is greatly affected which causes the grid dynamics to be changed hence, negatively affecting the performance of our system. Several strategies have been implemented in the past to control inverter based power systems. In this thesis we are trying to implement a strategy to provide voltage regulation, frequency support and develop virtual inertia in the system using control of inverter. The thesis is based on researching on different control strategies for inverter control and simulating them on PLECS|Plexim (The Simulation Platform for Power Electronic Systems). Finally the control is implemented using Dispatchable Virtual Oscillator Control (dVOC). After the results from PLECS, we moved towards Hardware in the Loop implementation. The controller logic is implemented in the Microcontroller from Texas Instruments using TI320F28069. I also tried to implement the control scheme using Hardware in the Loop (HIL).

Keywords: Hardware in the Loop(HIL), PLECS, Control Card, Micro-controller, Distributed Generation(DG), Frequency Control, Droop Control, Renewable Energy Sources, Storage System, Rate of change of frequency(ROCOF).

Sommario

POWER Systems sta avendo una svolta e i generatori distribuiti (DG) che utilizzano fonti di energia rinnovabile (FER) sono una soluzione promettente per raggiungere un approvvigionamento di elettricita piu sostenibile. Tuttavia, con l'aumento del numero di Generazioni distribuite basate su fonti di energia rinnovabile, il controllo del sistema e fortemente influenzato, il che provoca la modifica della dinamica della rete, influendo negativamente sulle prestazioni del nostro sistema. Diverse strategie sono state implementate in passato per controllare i sistemi di alimentazione basati su inverter. In questa tesi stiamo cercando di implementare una strategia per fornire regolazione della tensione, supporto di frequenza e sviluppare inerzia virtuale nel sistema usando il controllo dell'inverter. La tesi si basa sulla ricerca su diverse strategie di controllo per il controllo degli inverter e sulla loro simulazione su PLECS | Plexim (la piattaforma di simulazione per sistemi elettronici di potenza). Infine, il controllo viene implementato utilizzando il Controllo oscillatore virtuale dispacciabile (dVOC). Dopo i risultati di PLECS, siamo passati all'hardware nell'implementazione di Loop. La logica del controller e implementata nel microcontrollore di Texas Instruments

utilizzando TI320F28069. Ho anche provato a implementare lo schema di controllo usando Hardware in the Loop (HIL).

Parole chiave: Hardware in the Loop (HIL), PLECS, scheda di controllo, microcontrollore, generazione distribuita (DG), controllo di frequenza, controllo di caduta, fonti di energia rinnovabile, sistema di archiviazione, tasso di variazione di frequenza (ROCOF).

Acknowledgement

FIRST and foremost, I would like to thank God for giving me this ability and opportunity to undertake this study and courage to pursue this research. Thereafter, I would like to offer my distinctive gratitude to my supervisors: Prof. Alberto Berizzi in Politecnico di Milano and Prof. Dr. Agamy Mohammed in State University of New York at Albany. Their support, guidance and patient supervision throughout my thesis was invaluable.

I would also like to Thank Politecnico di Milano for giving me this opportunity of Thesis Abroad and at the same time to my Supervisor here in State University of New York who trusted in me and supported me from all aspects during my stay here in New York. Flexible working environment and thoughtful discussions with Dr. Agamy, his valuable advice were always fruitful for me. One could not wish for a better or more encouraging supervisor.

Finally, I would like to extend my distinctive gratitude to my Mother Fozia, my Father Idrees, to Fiza and Ahmad and my friends specially Zeshan and Faheem for always being there for me through thick and thin. I am grateful for their love and support, and their trust in me, and for encouraging me to pursue my dreams.

Contents

List of Figures

List of Tables

Introduction

The objective of this thesis is to explore the new dimension in the development and control of inverter based power grid. Conventional Power grid was composed of Synchronous generators. As of now, stability and system-wide synchronization of the grid is achieved with traditional synchronous generators and their controls. However with the increase in Renewable Energy Sources, inverter based control is the emerging development. This development of inverter based control, caters a lot of challenges among synchronization, load sharing and voltage regulation are a few of them.

The Research is carried out in State University of New York at Albany, United States in the Power Electronics Lab, under the Supervision of Dr. Agamy who has a wide experience in the domain of Power Electronics and Control. Thanks to the Funding opportunity provided by National Science Foundation (NSF), due to which the research was made possible. The Research is composed of different phases, however a brief summary is provided.

New York State National Grid is planning to amalgamate the grid with Distributed Generation mainly Renewable Energy Sources to utilize the potential of Clean Energy. Due to high penetration of RES, inverter based control is required and hence the control

strategies are needed to be investigated. On top, the inertia of the system will be decreased due to comparatively less number of Synchronous machines and increasing number of Renewable Energy Sources. Hence in order, not to utilize ancillary synchronous machines, a research is required on the feasibility of inverter based grid capable to manage the synchronization and inertia problems.

Thesis outline

This thesis consists of Six chapters. After introduction, some light is shed into the concept of Distributed Generation with the penetration of RES. Chapter two is dedicated to the explanation of Inverter Control strategies whereas Chapter three proposes our control algorithm. Chapter four emphasizes on the simulations performed in PLECS software. Chapter five introduces Hardware in the loop and test setup. Conclusions and future prospects are discussed in chapter six.

Contribution of thesis

The thesis contributes as follows:

- The study shed some light on evolution of the power grid and arising challenges with it.
- Investigated different control strategies regarding inverter control for a grid tried inverter.
- The control strategy "dispatchable Virtual Oscillator Control" is explained.
- Simulation of our inverter control strategy is presented in PLECS software along with results.
- Hardware in the loop implementation for testing of our inverter and virtual inertia approach is elaborated.
- Conclusions and future prospects are discussed.

Challenges faced in the study

The thesis lasted for about 6 months and the challenges faced to pursue the study are listed below:

- Inverter Control strategies were investigated thoroughly for the study. Author had to study the problems of synchronization and virtual inertia in depth and utilize it for the implementation of inverter control.

- Since there was a requirement of open source for the thesis, the language required was Embedded C programming and the programmable controller was also fixed. The author had to develop the competency in this regard.

- There was effectively no tutorials and help available for utilization of Hardware in the loop setup. The author had to invest a lot of time in understanding the Hardware in the Loop Architecture.

CHAPTER 1

Evolution of Power Grid

1.1 Conventional Power Grid

Early development of alternating current (AC) systems were in response of the shortcomings borne by the Thomas Alva Edison's direct current (DC) system. In 1885, invention of transformer changed the history of Electrical Systems. It solved the problem of voltage drops which was a problem in DC systems. Also the constraint of short distance transmission was overcome by increasing the voltage levels, hence reduction in current and resulting in reduced I^2R losses. AC transmission and distribution systems prevailed majorly during the twentieth century.

The power system mainly comprises of generation, transmission and distribution. In conventional power system the power flows only in one direction i.e from the power generating plants to the consumer which are mostly industrial, commercial and residential loads. These consumers require different voltage levels and hence transformers with different voltage and power ratings

are used for voltage and power compatibility. A conventional power grid infrastructure is shown in figure 1.1.

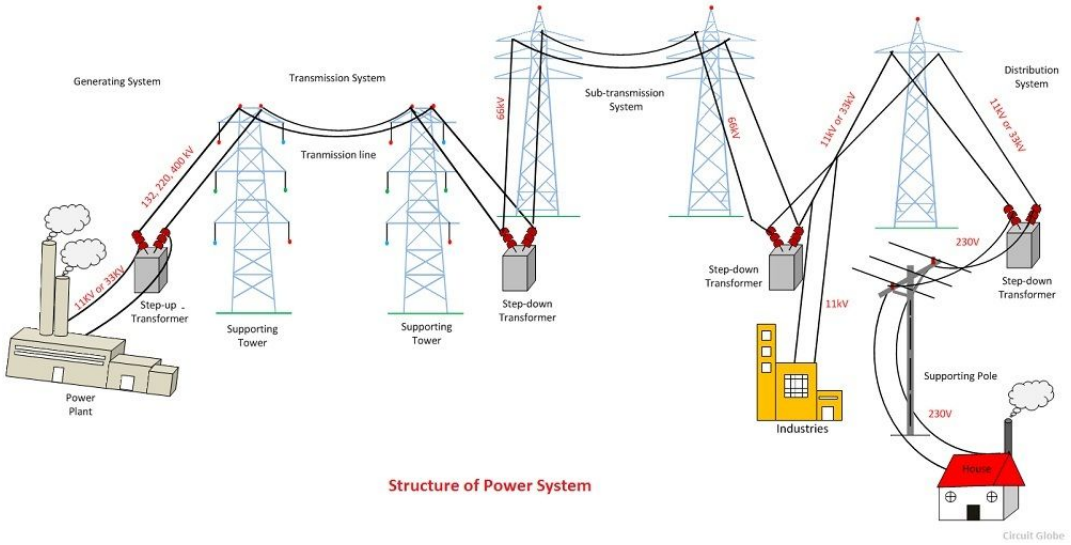


Figure 1.1: *A conventional power system infrastructure*

Power stations connected to grids are often located near energy resources such as a source of fuel which is economically viable. From the late 1800's until 2000, fossil fuels including coal, petroleum, and natural gas have been the major sources of energy in United States as depicted in figure 1.2. A synchronous grid or an "interconnection" is a group of distribution areas all operating with three phase alternating current (AC) frequencies synchronized (so that peaks occur at virtually the same time). This allows transmission of AC power throughout the area, connecting a large number of electricity generators and consumers. Multiple generating units are connected in a mesh, thus allowing a loss of power from one generator to be compensated by the others such that the stability of the system is maintained. However, one downside to a widely connected grid is thus the possibility of cascading failure and widespread power outage.

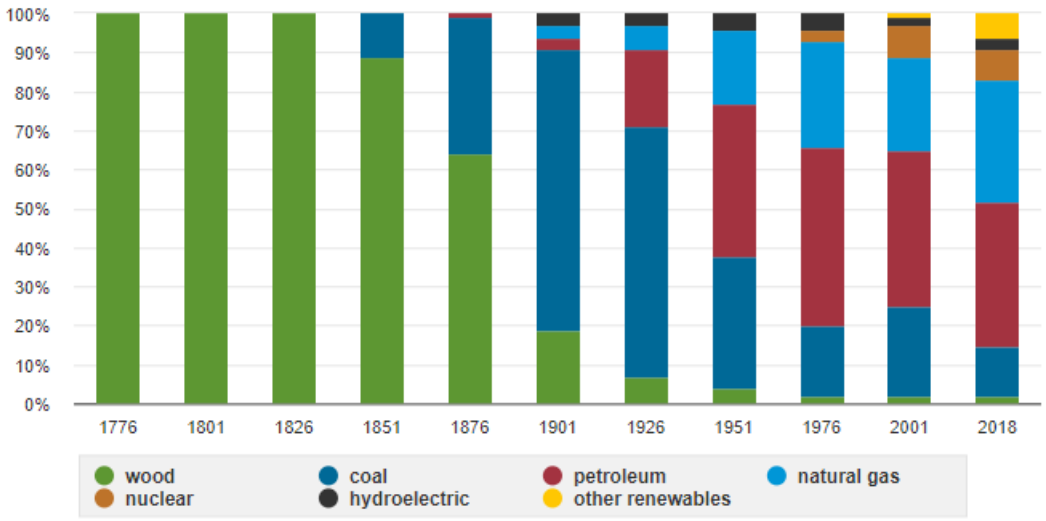


Figure 1.2: *U.S energy consumption by major sources*

1.2 Importance of Distributed Generation

In the traditional transmission and distribution grid, large sources provide power to huge numbers of residential, commercial, and industrial customers. Some of those consumers like industries live close to the centralized power plants. Other live far away, and sometimes very far. Generally, transmission of electricity from a power plant to a typical user wastes roughly 4.2 to 8.9 percent of the electricity as a consequence of losses and growing congestion.

Achieving universal access to electricity is essential for solving many global development challenges. According to new analysis by the International Energy Agency (IEA) published in World Economic Forum, 1.1 Billion people lack electricity in one way or another [1]. Decentralized renewable energy technologies have emerged as a viable solution. Distributed generation (DG) is an all encompassing term for any kind of power generation that occurs on a smaller scale, close to where the energy is used. This can mean solar panels installed on rooftops, fuel-cells, some geothermal plants, or micro-turbines just to name a few. A typical Distribution system is shown in figure 1.3.

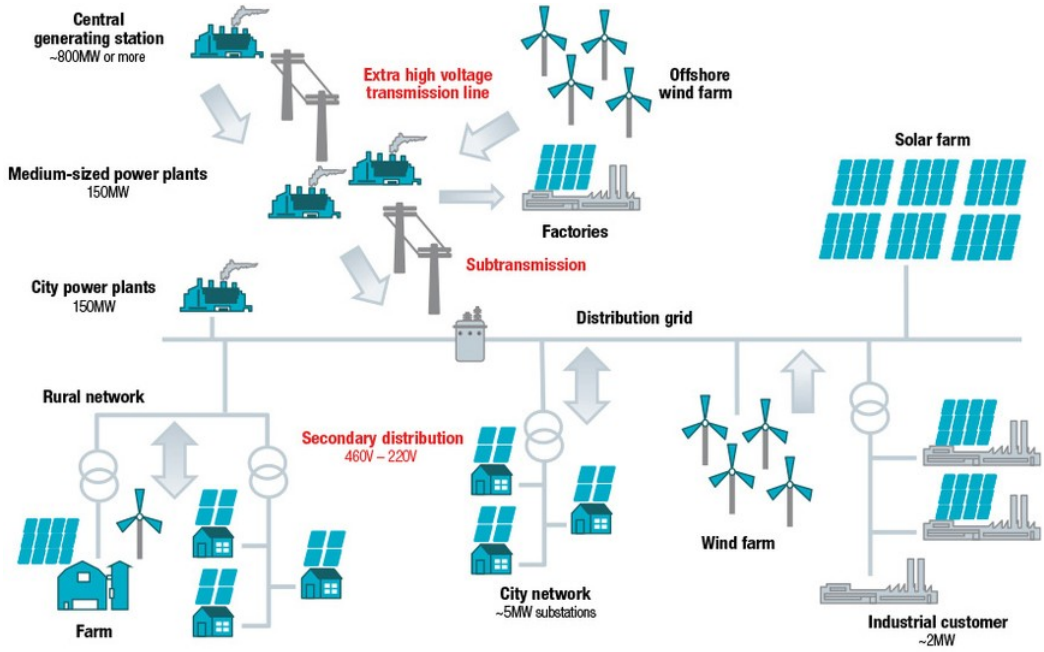


Figure 1.3: *A typical Distributed Generation(DG) configuration*

In recent years, distributed generation (DG) technology advancement has gained attention due to its wide range of benefits. The United States is committed to increase its distributed generation largely based on Renewable Energy Sources as a motive to clean and Green energy initiative. Renewable energy plays an important role in reducing greenhouse gas emissions. Using renewable energy can reduce the use of fossil fuels, which are major sources of U.S. carbon dioxide emissions.

The consumption of bio-fuels and other non-hydroelectric renewable energy sources in the United States more than doubled from 2000 to 2018, mainly because of state and federal government requirements and incentives to use renewable energy. The U.S. Energy Information Administration projects that U.S. renewable energy consumption will continue to increase through 2050 [2]. The energy source of United States as of 2018 is shown in pie chart 1.4.

Similarly in Europe, the growing environmental requirements

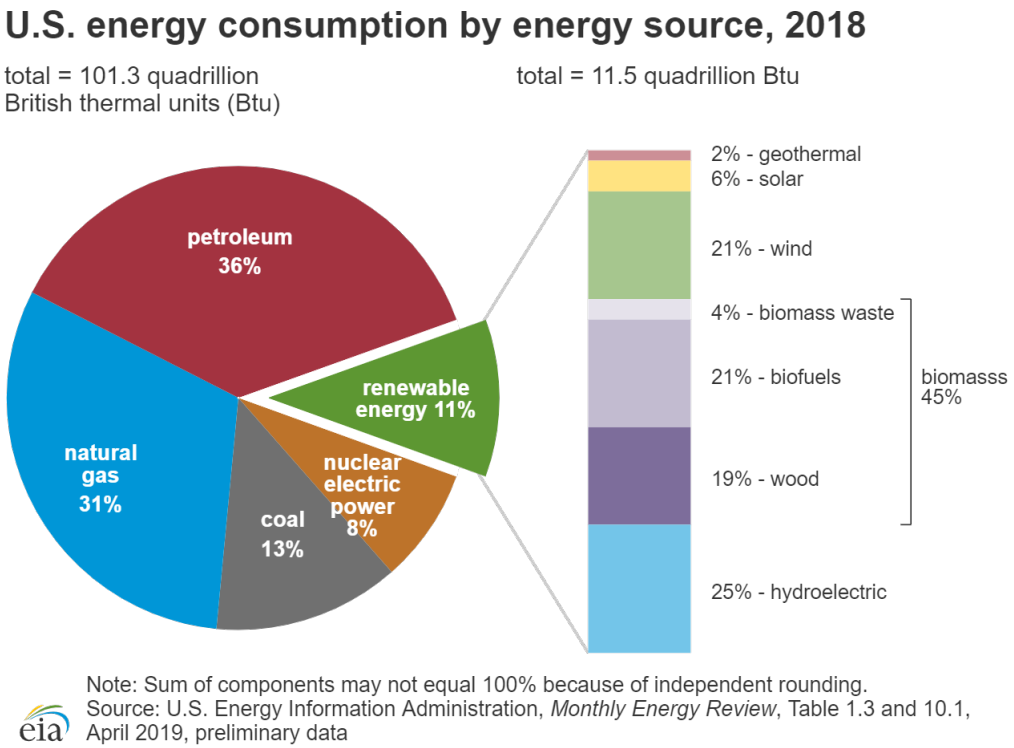


Figure 1.4: *U.S energy consumption 2018 [2]*

imposed on energy sector in 2020 climate and energy package, were introduced by European leaders in 2007 [3]. Key targets that were put in this perspective are, 20 percent cut in greenhouse gas emissions (from 1990 levels), 20 percent of EU energy from renewable, 20 percent improvement in energy efficiency. This legislation pushed countries to give incentives for renewable energy production. As a result, there was a boom in distributed renewable energy generation in EU distribution grids.

A good Distributed Generation(DG) based system reduces losses by incorporating multiple sources of Power mostly renewable. For instance, in case of Wind and solar Energy, both can be amalgamated in such a way that the cheaper solar power is utilized when there is plenty of sun and wind can be utilized when there is abundance of wind. A storage system can be used to store excess energy generated from these plants and therefore can be utilized by

consumers or added to the grid as applicable. This works similar to demand and response and the advantage of maximum utilization of resources can be largely achieved.

Distributed Generation also means lower number of Generators loss in case of catastrophic disaster as the Generation is distributed and not physically confined to very few locations. The loss of generation in case of contingency is a recurring process and can even lead to a blackout. The blackout of August, 2003 in United States is one of the examples which also leads to investigation in the area of Distributed Generation. The estimated economic loss estimated by Electricity Consumers Resource Council (ELCON) in February 2004 was approximately 6 Billion Dollars [4].

Distributed Generation also ensures clean energy when the source is renewable which avoids carbon dioxide emissions. One of the benefits of Distributed generation is its large scale-able technology. In order to make coal plants more efficient, the size of these plants needs to be up-scaled in order to move large turbines.In case of solar powered PV on the other hand, the technology remains the same at every scale. This inherent flexibility of the technology allows for simple scaling without additional breakthroughs.

1.3 Challenges in Distributed Generation

One of the biggest challenge in implementation of Distributed Generation is the compatibility of the conventional grid. Conventional Power system was not designed for bi directional flow of power, rather it was designed for unidirectional flow. This change has imposed restrictions on the conventional power system and there is a need for our system to be adaptive to changes. Smart Grid concept is deployed in order to solve the problem of Distributed Generation. A typical Smart Grid Network is shown in figure 1.5.

"Smart grid" technologies are made possible by two-way communication technologies, control systems, and computer process-

Figure 1.5: *A Smart Grid Topology [5]*

ing. These advanced technologies include advanced sensors that allow operators to assess measurements and grid stability, advanced digital meters that give consumers better information and automatically report outages, relays that sense and recover from faults in the substation automatically, automated feeder switches that re-route power around problems, and batteries that store excess energy and make it available later to the grid to meet customer demand.

In May 2015, the United States Senate Committee on Energy and Natural Resources presented a bill that amends the Energy Independence and Security Act of 2007 [6] to direct the Department of Energy (DOE), in collaboration with the National Institute of Standards and Technology of the Department of Commerce, the Institute of Electrical and Electronics Engineers, and the Smart Grid Interoperability Panel, to establish the Smart Grid Interoperability Working Group and the Smart Grid Regional Demonstration Initiative shall identify best practices for the implementation [18].

Distributed Generation specifically Solar and Wind are the generation technologies that are not market competitive with tradi-

tional power plants. Their Production cost is close to zero, however, their capital cost is quite high which makes the regulatory framework challenging.

The Distributed Generation mainly from RES is generally unpredictable and hence load patterns are less predictable. Although the energy is free from RES, however it is not always usable unless Electrical Storage Systems are used which makes the whole solution expansive. The advancement in Electrical Storage Technologies ensures the maximum utilization of these DG with high efficiency desired.

Mainstream renewable energy sources such as fuel cells and photovoltaics are inherently DC in nature. The output voltage and current they generate is DC. Moreover, AC inherent renewable source, wind generators also need DC links for the higher efficiency and reliable supply. Generation from Renewable Energy Sources as well as distributed generation (DG), both are connected with their respective AC Voltage levels, with suitable conversion techniques from DC to AC. This introduces the main challenge of RES and Distributed Generation to be integrated with in the grid.

In order for the Distributed Generation to be integrated within the grid, the dynamics have to be controlled. These include the network voltage from varying generation outputs as seen by the grid as well as the frequency synchronization. The voltage levels generated by DG's specifically from RES are of different magnitude and DC in nature; hence there is a need to convert these DC signals to AC by an inverter. After conversion to AC, the voltage levels from the output of the inverter and that of the grid need to be same. The same is true for the frequency.

The power injections in the grid may create variation in the grid frequency or variation in voltages or both, and there should be control of such parameters which is characterized by control of an inverter. Such a control is the building block of this thesis work.

Up til now, we have realized the potential of Distributed Generation through Renewable Energy Sources. Moreover, the evolving technologies of renewable and loads have made us constantly rethink about the changes to be made in power system to make it ready for rapidly changing consumer behavior e.g. access to the cheap PV panels for homes and economic electric vehicles availability in recent years. The next chapter will discuss about the control of Inverters generally and the approach of this thesis in detail.

CHAPTER 2

Problem Statement and Control of Inverter

2.1 What is inverter Control

In chapter 1 we have concluded with the importance of distributed generation through renewables and its impacts on economic viability and environment friendly nature. However, to use these Energy Sources we need to convert them into a suitable form which is compatible with our power system and grid infrastructure. Inverters are generally used to convert DC source into AC source. However, conversion from DC to AC caters a lot of challenges in itself. A typical block diagram of the process is shown in figure 2.1 and the general symbol used for an inverter is shown in figure 2.2.

The inverter does not produce any power; the power is provided by the DC source. A DC power source can be anything ranging from batteries to PV panels. The waveform obtained from the

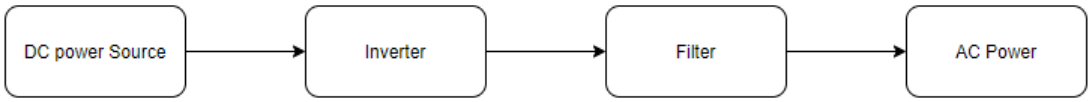


Figure 2.1: *General conversion from DC source to AC Power*

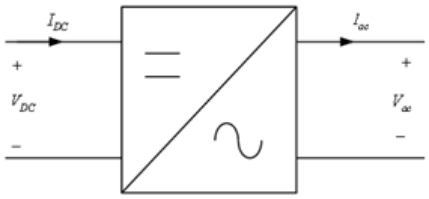


Figure 2.2: *Symbol used for an inverter*

output of an inverter is not purely sinusoidal and hence a filter is required to eliminate the harmonics from the waveform. After filtration, the output waveform is ready to be utilized for loads within constraints of power, voltage, current and frequency. The input voltage, output voltage and frequency, and overall power handling depend on the design of the specific device or circuitry.

Control of an inverter is generally constituted on the movement of switches. The control of these switches in a defined pattern produces the desired output waveform which is alternating in nature. An inverter can produce a square wave, modified sine wave, pulsed sine wave, pulse width modulated wave (PWM) or sine wave depending on circuit design. The electronic components used for switching are MOSFET's(metal-oxide-semiconductor field-effect transistor) or IGBT's(Insulated-gate bipolar transistor). These components have capability of fast switching and good power handling capabilities. A typical circuit diagram of an inverter along with LC filter and Resistive Load is shown in figure 2.3 [7].

Switches S1, S2, S3 and S4 are used to control the output waveform but this waveform is not sinusoidal in nature, rather it has many harmonics which needs to be filtered out.

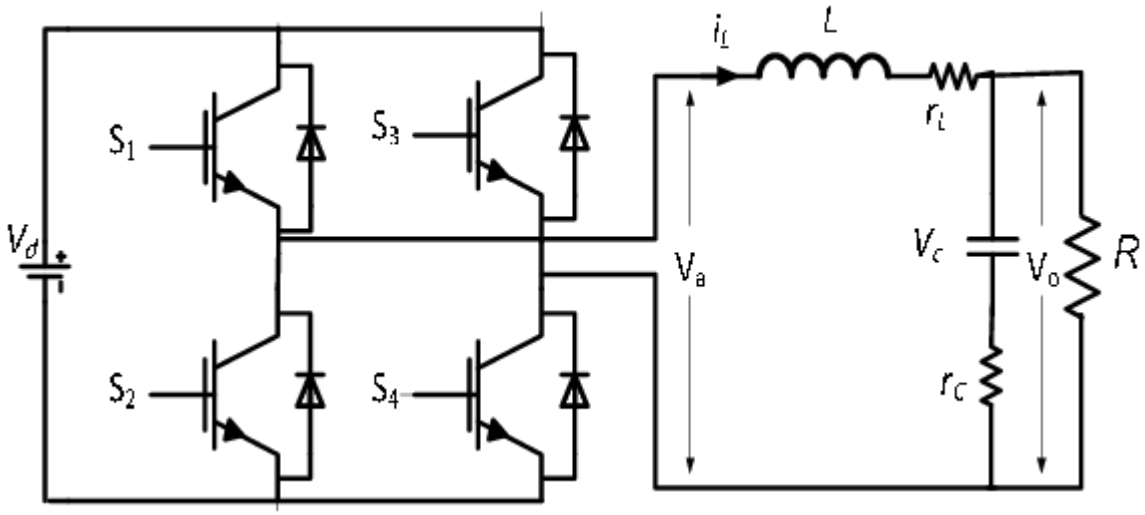


Figure 2.3: *Circuit diagram of a Single Phase Inverter with LC filter and Resistive Load*

2.1.1 Total Harmonic Distortion

In conversion from DC to AC, the output waveform should be sinusoidal. Total harmonic distortion (THD) is a measurement that tells you how much of the distortion of a voltage or current is due to harmonics in the signal. A voltage or current that is purely sinusoidal has no harmonic distortion because it is a signal consisting of a single frequency. The concept can be elaborated using figures 2.4 & 2.5 in which sinusoidal and square waveforms are compared with respect to THD. Hence we need to filter out these harmonic contents in order for the output of our inverter to be utilized according to standards.

Common types of inverters produce square waves or quasi-square waves. Since THD is one of the measure of the purity of a sine wave. A 50% duty cycle square wave is equivalent to a sine wave with 48% THD [8]. Technical standards for commercial power distribution grids require less than 3% THD in the wave shape at the customer's point of connection. IEEE Standard 519 recommends less than 5% THD for systems connecting to a power

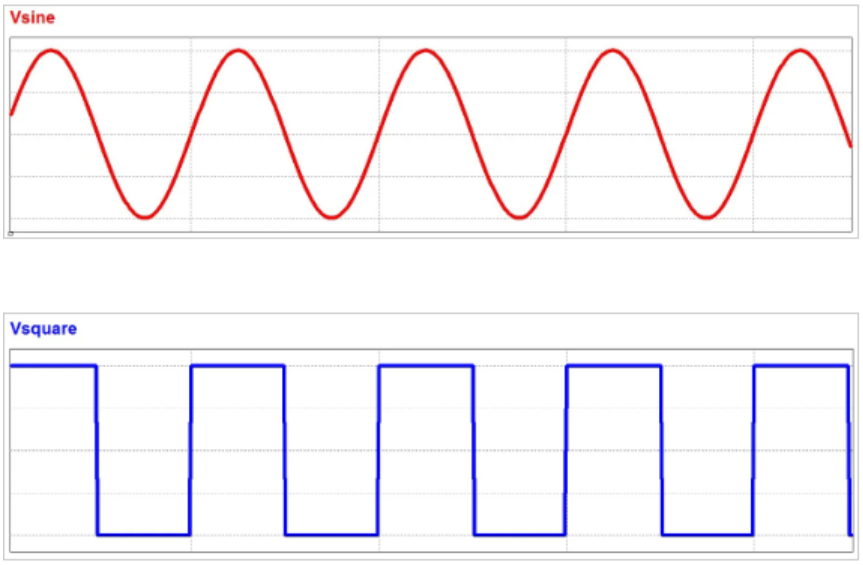


Figure 2.4: *A sinusoidal voltage and a square wave voltage in the time domain.*

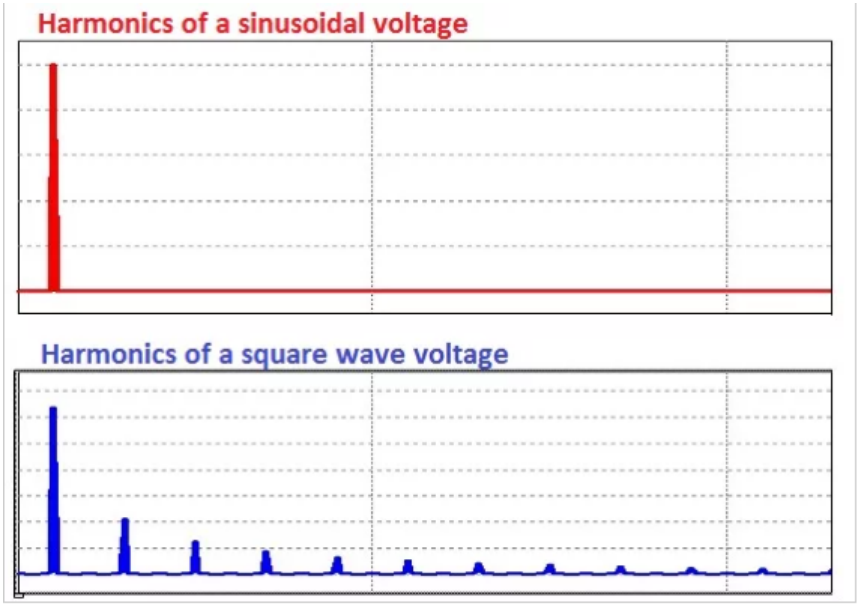


Figure 2.5: *A sinusoidal voltage and a square wave voltage in the frequency domain*

grid. The formula used to calculate THD is given in equation 2.1

$$THD = \frac{\sqrt{\sum_{n=2}^{\infty} V_{n\_rms}^2}}{V_{fund\_rms}} \tag{2.1}$$

Where:
$V^2_{n\_rms}$ is the RMS voltage of the nth harmonic.
V_{fund_rms} is the RMS voltage of the fundamental frequency.

2.2 The Scope of Problem

The Project of which this thesis is a part, is funded by National Science Foundation (NSF). The U.S government is appreciating its masses towards deployment of new Energy Integrations from Renwable Energy Sources. Among all of the Renewable technologies, Solar PV is highly dominant. A futuristic comparison of grid tied inverters with synchronous generators is shown in figure 2.6.

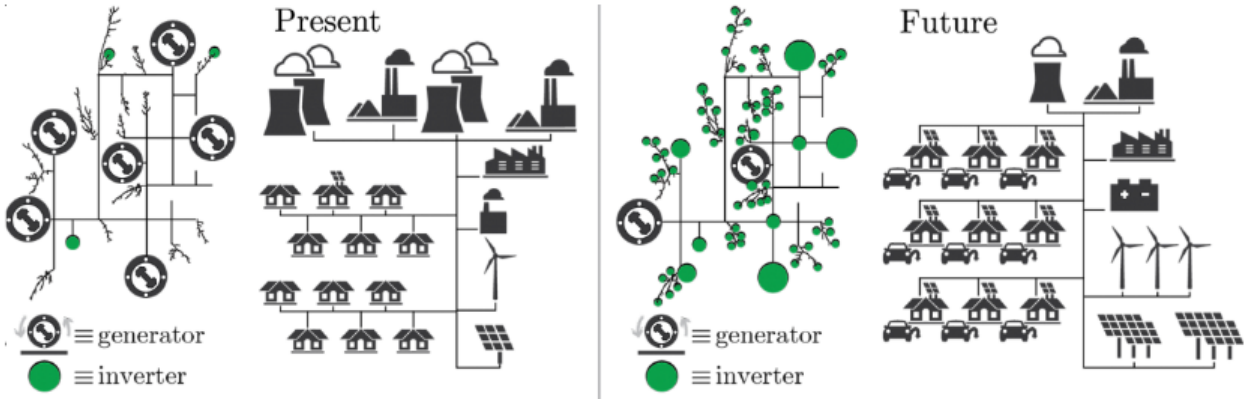


Figure 2.6: *A Present and futuristic vision of the synchronous generators and inverter dominant grid respectively [9]*

In the process of utilization of power from these renewable energy sources, a conversion is required and hence Inverters are required for the conversion of Power into a form which is compatible with the loads. The inverters have to be designed so that they can be integrated within the grid. The challenges require voltage regulation and frequency synchronization within the grid constraints.

Another challenge of RES (specifically PV and wind) and grid tied inverters is the loss of inertia. The Synchronous generators have rotating parts and hence a sudden change in frequency is

shortly compensated by the inertia stored in the system of synchronous generator. However the grid tied inverters are static in nature. They don't have movable parts to generate physical inertia. The New York state requires the control of grid tied inverters in such a way to compensate for this loss of inertia and provide grid stabilization with this increasing integration of Renewables.

Without this control, the state has to utilize the free running generators which are traded in the ancilliary service market for the compensation of inertia due to increase in grid tied inverters. This trade increases the cost and is not economically viable. The Research is based on the feasibility studies and control algorithm which can control a grid with penetration of RES without utilization of ancilliary Synchronous Generators (which are traded only for providing inertia in the system), hence creating an economic impact for the grid.

2.3 Importance of Inertia and Virtual Inertia Concept

The modern power system is progressing from a synchronous machine-based system towards an inverter-dominated system, with large-scale penetration of renewable energy sources (RESs) like wind and photovoltaics. The traditional approach of integrating them as grid following units can lead to frequency instability. The rapid development of RES is causing the modern power grid to gravitate towards an inverter-dominated system from a rotational generator-dominated system, as illustrated in figure 2.7.

PV systems and most modern wind turbines are interfaced through inverters. Although this is advantageous from the point-of-view of harvesting RES, the inverter-based generation does not provide any mechanical inertial response, and hence compromises frequency stability. In the state of Texas, United States the Electricity Reliability council has reported a decrease in inertial response of the system and hence demanded more inertial response. The importance of inertia can be comparatively understood by a

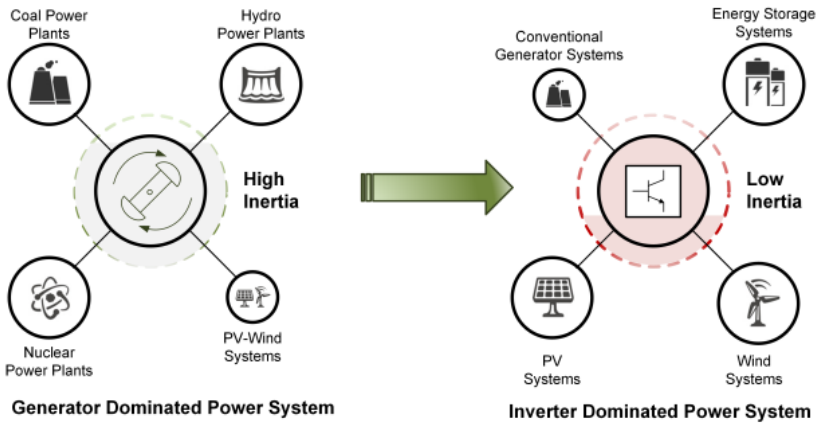


Figure 2.7: *Evolution towards Inverter dominant system*

figure 2.8 which is obtained from ERCOT (Electricity Reliability Council of Texas) [25].

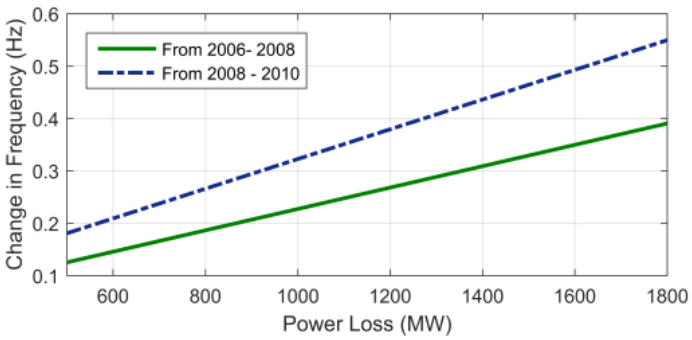


Figure 2.8: *Evolution towards Inverter dominant system [25]*

As a general rule, the loss in power will cause the frequency to drop and hence a change in frequency is observed. The two lines depict Power loss and Frequency change for two spans of time. Between 2006-2008 there were more Rotating Generators in comparison with inverter based connection. While from 2008-2010, the grid tied inverter based generation increases causing the inertia of the system to drop and hence the change in frequency is more for the same loss in power. A similar response in frequency violations is reported by ENTSO-E from the Nordic grid which is

highly densed with RES and grid tried inverters [26].

2.3.1 Frequency Response of a Power System

The purpose of a power system is to balance generation and loads however this is not always true and a flexible margin is always available. In the event, when generation is more than loads the system frequency increases while increase of loading as compared to generation leads to decrement in frequency. Hence the frequency is always fluctuating due to this imbalance. In order to maintain the power generation and load balance, various control actions are implemented in a power system which are initiated in different times. The equation that govern power balance is given in equation 2.2 , which is rephrased in equation 2.3 as:

$$\sum P_{Generated} - \sum P_{Consumed} = 0 \tag{2.2}$$

$$\sum P_{Generated} = \sum P_{Loads} + \sum P_{T\_loss} \tag{2.3}$$

Where:
$P_{Generated}$ is the Power generated;
P_{Loads} is the Power consumed by Loads;
P_{T_loss} is the Power consumed in transmission losses.
When a change in frequency occurs, the generator(s) have to take some action. Let's take an example where loading is more than generation, causing a drop in frequency. As a result, the generators have to generate more power to balance the load and maintain the frequency. The governor response is the primary control action which takes place within the first few seconds (typically 10-30 s) with a frequency change, and aims at balancing power and stopping the change in frequency.

Once the frequency is stopped from changing, the secondary control action which is automatic, takes place within minutes (typically 10-30 min) and restores the system frequency back to the

nominal value. The tertiary control action is the reserve control to handle present or future disturbances in the system. The frequency response with timing is shown in figure 2.9.

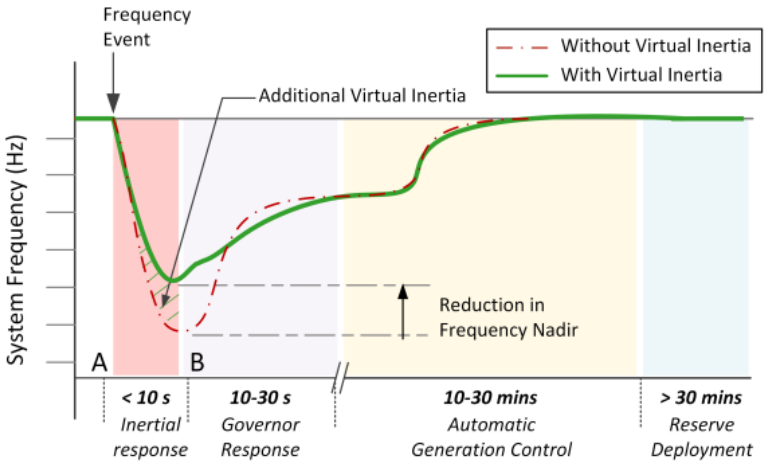


Figure 2.9: *Frequency response with different controls and timings*

Whenever there is a change of frequency in the system, the generators cannot respond instantaneously to arrest the system frequency, as the primary reserve comes into action in approximately 10 seconds. The rapid change in frequency is stopped by the kinetic energy stored in the rotors which is responsible for counteracting this imbalance through inertial response until the primary frequency control has been activated.

As we are introducing more RES which are controlled by inverters, the inertial response of the system further decreases. As a consequence the rate of change of frequency increases and frequency is touching the low minimum (frequency nadir) in a short duration of time. This is depicted in figure 2.9. It can also be observed that in the systems with low inertia, the frequency is touching minimum point in a short duration of time along with the fast rate of change of frequency.

The disadvantage of this problem is the possibility of unexpected tripping due to under frequency detection by frequency relays and in worst case can lead to cascaded outages in which

multiple generators will cut out from the system resulting in big loss of generation.

The solution to the above problem is to add inertia in the system. One way is to add rotating generators in standby which will only be utilized for inertial response, however this solution is expensive and these generators are mostly traded in ancillary services market. Another way is to add virtual inertial, i.e., inertia from the control of inverters.

The criterion for this generation of inertia is its fast response. The response time should be less than 10 seconds, i.e., before the primary control and it should be autonomous in nature. An approach utilized in this research is based on the fact that since primary control works with respect to the change in frequency, our control i.e. inertial control will act based on rate of change of frequency with time.

The success of virtual inertia deployment in the system would enhance system stability and encourage more penetration of Renewable Energy Sources (RES) through grid tied inverters.

2.4 Some Inverter Control Topologies

The high penetration of inertia-less PV and wind energy systems has a severe effect on the frequency stability. The rapid changes in the generation can cause frequency variations in the system that are outside standard limits and compromise the stability of the system.

In order to understand the control topologies, we first will refer to the basics of $Swing$ Equation as given below:

$$P_m - P_e = Jw_m \frac{\mathrm{d}\delta_m}{\mathrm{d}t}$$

Where P_m is the mechanical power, P_e is the electrical power, δ_m is the angular position in radians with respect to the synchronously rotating reference frame and J is the total moment of inertia of the rotor mass.

The frequency variation in a power system after a frequency disturbance can be approximated by the swing equation [10] and can be proceeded as:

$$P_{Gen} - P_{Load} = \frac{\mathrm{d}(K.E)}{\mathrm{d}t} \tag{2.4}$$

Where K.E = $\frac{1}{2}Jw_n^2$.
Putting in above equation gives:

$$P_{Gen} - P_{Load} = \frac{\mathrm{d}(\frac{1}{2}Jw_n^2)}{\mathrm{d}t} = Jw_n\frac{\mathrm{d}w_n}{\mathrm{d}t} \tag{2.5}$$

Where P_{Gen} is the generated power or the mechanical power, while P_{Load} is the electrical power demanded including losses. w_n is the grid/system frequency and J is the Inertia of the system. The inertia constant of the system is defined as the Kinetic Energy $\frac{1}{2}Jw_n^2$ normalized to nominal Power S_n given in equation 2.6.

$$H = \frac{\frac{1}{2}Jw_n^2}{S_n} \tag{2.6}$$

Rearranging equation 2.5:

$$Jw_n = \frac{P_{Gen} - P_{Load}}{\frac{\mathrm{d}w_n}{\mathrm{d}t}} \tag{2.7}$$

putting Equation 2.7 in equation 2.6, we get:

$$\frac{P_{Gen} - P_{Load}}{S_n} = \frac{2H}{w_n}\frac{\mathrm{d}w_n}{\mathrm{d}t} \tag{2.8}$$

Equation 2.8 can be represented in frequency (Hz) instead of angular frequency w_n (rad/sec):

$$\frac{P_{Gen} - P_{Load}}{S_n} = \frac{2H}{f}\frac{\mathrm{d}f}{\mathrm{d}t} \tag{2.9}$$

$\frac{\mathrm{d}f}{\mathrm{d}t}$ is called rate of change of frequency which is very important in frequency control. We can see that in order to keep Power

system balance, if we have less inertia in the system, more will be the rate of change of frequency which causes the frequency to go out of its limits in a short duration of time. Hence we need to inject more inertia.

The core idea behind virtual inertia implementation is the virtual inertia algorithm that presents the various energy sources interfaced to the grid through power electronics converters as Synchronous Generators. The control is based on voltage or current feedback from the inverter and consequently generate appropriate gating signals through algorithm to present these resources as SGs from the point-of-view of the grid. Hence Virtual inertia is a combination of control algorithms and Renewable Energy Sources along with power electronics that emulates the inertia of a conventional power system. The idea is shown in figure 2.10.

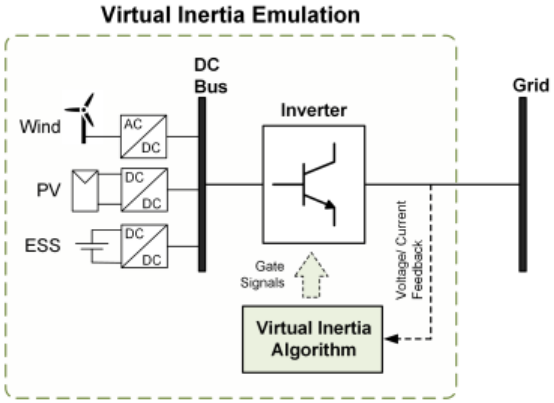


Figure 5. Concept of virtual inertia.

Figure 2.10: *Virtual Inertia Concept*

The most topologies studied so far of which, many try to mimic the behavior of the Synchronous Generator through a detailed mathematical model that represent their dynamics. Others try to simplify this by using just the swing equation to approximate the behavior of Synchronous Generator, while some of them employ an approach which makes the inverter controlled units responsive to frequency changes in the power system like in case of droop

control.

In an approach to implement virtual inertia, either dedicated Energy Storage Systems (ESS) can be used, or inertia can be emulated by operating PV/wind below their maximum power capacity [11]. Similar to inertia constant for Synchronous Generators as given in equation 2.6, we try to emulate the inertia of a Synchronous Generator by a PV source for which the virtual inertia co-efficient of a PV can be defined as 2.10:

$$H_{PV} = \frac{E_{K.E}}{S_{PV}} \tag{2.10}$$

Where $E_{K.E}$ is the Kinetic Energy to be emulated by the PV and S_{PV} is the rated power of the PV system.

The frequency dynamics of the PV system during inertia emulation can be defined similar to equation 2.9 as:

$$\Delta P_{I.Em} = \frac{2H_{PV}}{f_n}\frac{\mathrm{d}f}{\mathrm{d}t} \qquad (p.u) \tag{2.11}$$

Where $\Delta P_{I.Em}$ is the change of active power generation for virtual inertia emulation given in per unit. This is the change in Power that must be there in order to provide inertia to the system which is reciprocal to the inertia provided by the Synchronous Generator due to its Kinetic Energy. Hence the updated generation setting for the PV is given in equation 2.12.

$$P_{up\_gen} = P_{old\_gen} + \Delta P_{I.Em} \tag{2.12}$$

Where P_{old_gen} is the generation setting from upper level controller while P_{up_gen} is the updated generation setting for virtual inertia injection.

$\Delta P_{I.Em}$ is provided by PV panels and also Energy Storage System if applicable as given in equation 2.13.

$$\Delta P_{I.Em} = \Delta P_{PV} + \Delta P_{ESS} \tag{2.13}$$

Where ΔP_{PV} is the contribution from PV generation while ΔP_{ESS} is the possible contribution from the Energy Storage System towards inertia emulation. The power $\Delta P_{I.Em}$ needs to be reserved for the purpose of inertia injection and hence cannot be utilized for normal operation.

The concept is similar to the Synchronous generators in the sense that they have a margin of increasing or decreasing power for reserves (normally 10%). Likewise, the PV system has to have a reserve for inertia injection. The similarity is not true always, since the power produced by RES is highly dependent on external conditions(weather/temperature). Moreover, in the case when inertia contribution is not required, the useful energy of RES is wasted and hence not utilized while in case of Synchronous Generator, this energy is not wasted but available upon requirement. In case of high frequency due to less demand, energy is absorbed by Energy Storage Systems and hence support inertia of the system.

2.4.1 Synchronvertes

The idea of operating an inverter to mimic a synchronous generator is the main motivation and developed in this approach. If a synchronverter is connected to the utility grid and is operated as a generator, no difference would be felt from the grid side between this system and an Synchronous Generator. Frequency drooping mechanism is used to regulate the power output from the inverter similar to how the SG regulates its power output [12].

The author studies this topology from literature by Qing-Chang Zhong, Senior Member, IEEE, and George Weiss. [13]. The approach is based on modeling of the Synchronous Machine. The conclusion from the Electrical and Mechanical part Modeling is given in equation 2.14.

$$T_e = M_f i_f < i, \widetilde{\sin\theta} > \tag{2.14}$$

Where T_e is the electromagnetic torque of the synchronverter, M_f is the magnitude of the mutual inductance between the field coil and the stator coil, i_f is the field excitation current, θ is the angle between the rotor axis and one of the phases of the stator winding. Since this is three phase, the current i and $\widetilde{\sin\theta}$ are vectors given below:

$$i = \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} ; \widetilde{\sin\theta} = \begin{bmatrix} \sin\theta \\ \sin(\theta - \frac{2\pi}{3}) \\ \sin(\theta - \frac{4\pi}{3}) \end{bmatrix}$$

$< i, \widetilde{\sin\theta} >$ represents the standard inner product of two vectors in $\mathbb{R}^3$.

The figure representing the power part of this topology is shown in figure 2.11.

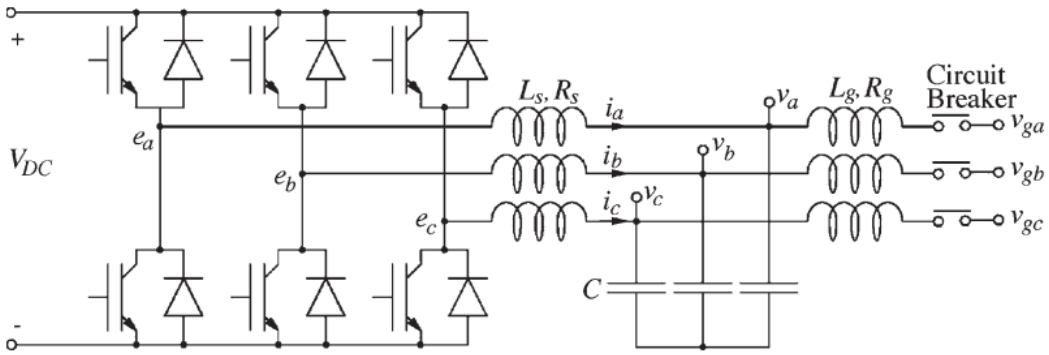


Figure 2.11: *Power part of a synchronverter including filter*

V_{DC} in the figure represents a general DC Voltage source which can be either a PV array or an Energy Storage. It is important to note that the capacitor voltages v_a, v_b, v_c mimic the terminal voltage terminal voltages of the imaginary Synchronous Generator. Furthermore, the impedance of the stator windings of the imaginary Synchronous Generator is represented by the inductance Ls and the resistance Rs of the left inductors is shown in figure 2.11.

The no load voltage e is given by equation 2.15.

$$e = \dot{\theta} M_f i_f \widetilde{\sin\theta} \tag{2.15}$$

where:

$$e = \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}$$

The generated real and reactive power is given as:

$$P =< i, e >; \qquad Q =< i, e_q >$$

where e_q is the same as e with the phase delayed by $\pi/2$. Hence reactive power is given by equation 2.16.

$$Q = -\dot{\theta} M_f i_f < i, \widetilde{\cos\theta} > \tag{2.16}$$

Where :

$$\widetilde{\cos\theta} = \begin{bmatrix} \cos\theta \\ \cos(\theta - \frac{2\pi}{3}) \\ \cos(\theta - \frac{4\pi}{3}) \end{bmatrix}$$

Equations 2.14, 2.15, 2.16 are first discretized and then solved in each control cycle in a digital controller to generate the gating signals for the Distributed Generation(DG) unit or inverter-based RES unit under consideration. Figure 2.12 shows the basic schematic of the synchronverter.

The inverter output current i and grid voltage v are the feedback signals utilized to solve the differential equations within the controller. Additionally, the desired moment of inertia J and damping factor D_p can be set as desired. Frequency and Voltage is controlled by the logic as shown in figure 2.13.

In the frequency loop, T_m which is the imaginary mechanical torque, is generated from the reference active power P^* based on the nominal angular frequency of the grid w_n. The virtual angular frequency of the synchronverter w is thus generated by this loop

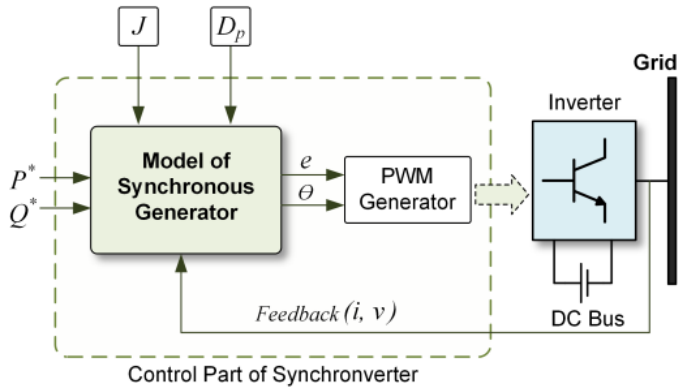


Figure 2.12: *Basic Control Schematic of a synchronverter*

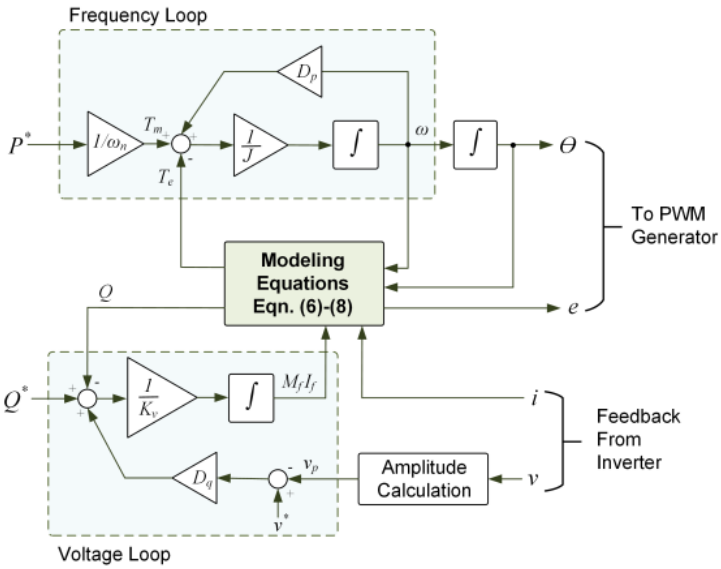


Figure 2.13: *Detailed Control Schematic of frequency and voltage*

which is integrated to calculate the phase command θ, and is used for the generation of pulse width modulated signal which is further used by the gate driver to operate inverter switches.

Similarly, in the voltage loop, the difference between the reference voltage v^* and and the amplitude of the grid voltage v is multiplied by a voltage drooping constant D_q. This is added to the error between the reference reactive power Q^* and the reac-

tive power Q calculated using equation 2.16. The resulting signal is then passed through an integrator with gain $1/k_v$ to generate $M_f I_f$. Finally the outputs of the controller are e and θ which are used for PWM generation.

The PWM generator compares a triangular carrier with a modulating signal to produce desired PWM signals.

Pros and Cons: The fact that the frequency derivative is not required for the implementation, is a major advantage as derivative terms often induce noise in the system. The synchronverter is able to somehow replicate the dynamics of a Synchronous Generator, however, the computation of differential equations is quite complex which can result in numerical instability.

2.4.2 Swing Equation Approach

The author referred this topology presented by Kelsey Sakimoto, Y. Miura and T. Ise in IEEE 8th International Conference on Power Electronics [14]. The idea is to emulate inertia similarly to synchronverter approach described previously, but instead of using a full detailed model of the Synchronous Generator, the topology solves the power-frequency swing equation. The approach of this topology is depicted in figure 2.14.

We can see from the figure that the current i and voltage v are sensed by the controller and they are used to calculate the grid frequency w_g and the output Power of the inverter P_{out} . These are the inputs to the control algorithm along with P_{in}. A Digital Signal Processor solves these equations and results in e and θ. The input power P_{in} mimics the prime mover input power in a Synchronous Generator is calculated as depicted in figure 2.15.

As part of the algorithm, the swing equation 2.17 is solved in every control cycle by DSP thus generating the phase command θ for the PWM generator. The typical swing equation of a Synchronous Generator as implemented in [4] is:

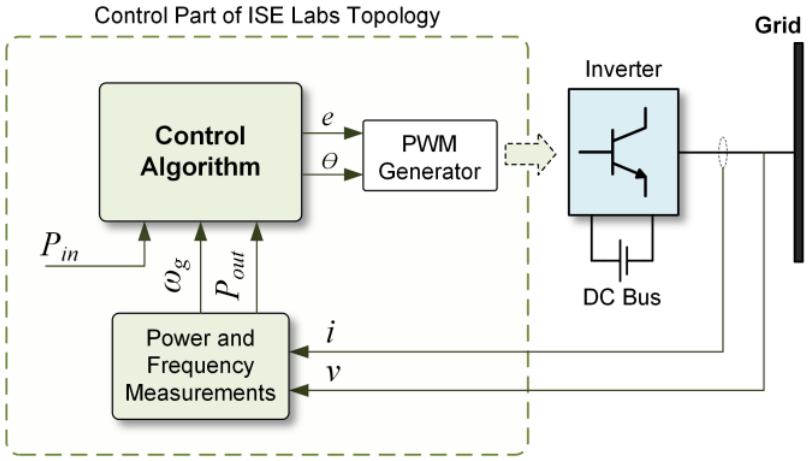


Figure 2.14: *Control Schematic for Swing equation approach*

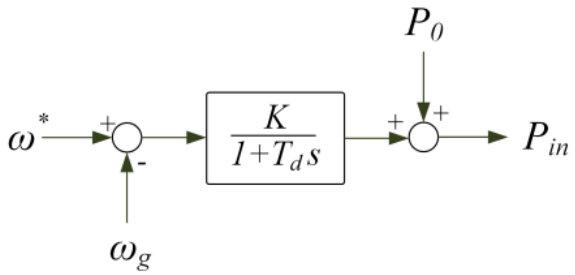


Figure 2.15: *Calculation of P_{in} using governor Model*

$$P_{in} - P_{out} = Jw_m \frac{\mathrm{d}w_m}{\mathrm{d}t} + D_p \Delta w \tag{2.17}$$

Where :

$$\Delta w = w_m - w_g$$

In the above equation:
P_{in} is the input power similar to the prime mover input power in Synchronous Generator.
P_{out} is the inverter output power.
w_m is the virtual angular frequency as to mimic the Synchronous Generator.

w_g is the grid angular frequency.
J is the moment of Inertia.
D_p is the Damping factor.

Input power P_{in} is computed based on the frequency deviation from a reference frequency w^*. More deviation from grid frequency means more power injection i.e. more P_{in}.

The governor is modeled as a first-order lag element with gain K and time-constant T_d. P_0 represents continuous power reference for the DG unit. One of the drawback of this approach is the delay in the governor model which leads to high rate of change of frequency. In a similar way, the voltage e is computed using Q-v droop approach as described in Research Papers [15],[16].

Pros and Cons: As we can see from the algorithm, no frequency derivative has to be calculated as they can introduce noise in the system which makes control to be difficult, however, the problem of numerical stability is the same in which a small error will deteriorate the process. Also improper tuning of parameters J and D_p, can lead to oscillatory system behavior.

2.4.3 Droop Based Approach

This is the last approach to be discussed in this chapter before we proceed to the one used in this research. The techniques used so far have been trying to mimic the behavior of a Synchronous Generator by Modeling the control algorithm. Droop based approach used frequency droop characteristics to control the inverter. Let's say the impedance of the grid is inductive, the active power-droop is defined by equation 2.18.

$$m_p = \frac{w^* - w_g}{P_{out} - P_{in}} \tag{2.18}$$

which can be re-arranged to give:

$$w_g = w^* - m_p(P_{out} - P_{in}) \tag{2.19}$$

Where w_g is the grid frequency, w^* is the reference frequency, P_{in} is the reference active power while P_{out} is the measured output power from the inverter. The voltage droop is provided by equation 2.20 as:

$$v_g = v^* - m_q(Q_{out} - Q_{in}) \tag{2.20}$$

Where v_g is the local grid voltage, v^* is the reference voltage, Q_{in} is the reference reactive power while Q_{out} is the measured output reactive power from the inverter. m_q is the reactive power droop.

The schematic of a frequency-droop controller based on Equation 2.19 is shown in Figure 2.16.

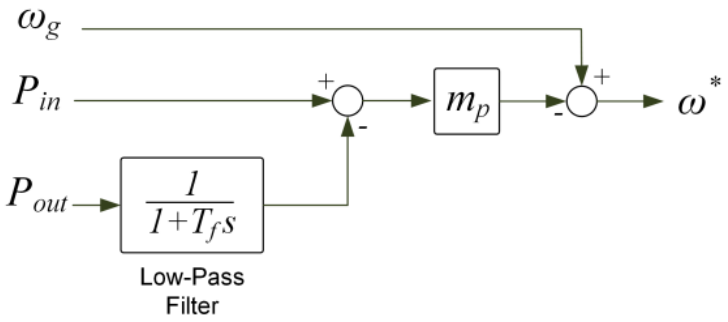


Figure 2.16: *Frequency-droop controller schematics*

We can see that a low pass filter with a time constant T_f is used when measuring the output power. This is performed in order to filter out high frequency components from the inverter [17]. Hence, the filter used for power measurements in this controller constitute a delay which is mathematically equivalent to virtual inertia, while the droop gain is equivalent to damping.

Pros and Cons: The concept of droop control is similar to traditional droop control in Synchronous Generator, however the droop controller has a slow transient response. It is clear that the objective of all the topologies is to provide dynamic frequency response through power electronic converters.

In the next chapter, the approach of dispatchable Virtual Oscillator Control is discussed.

CHAPTER 3

Proposed Algorithm using Dispatchable Virtual Oscillator Control

3.1 A brief Introduction

The electric power grid is undergoing a period of unprecedented change. A major transition is the replacement of bulk generation based on synchronous machines by distributed generation interconnected to the grid via power electronics devices fed by renewable energy sources. This implies that a part of grid may operate without conventional Synchronous Generators. As a consequence, the power grid faces great challenges due to the loss of rotational inertia and the self-synchronizing dynamics of Synchronous Generators. In order to face this challenge, the inverters need to be controlled in such a way as to compensate the effect of Synchronous Generators.

The approaches used to control inverters as discussed in Chap-

ter 2 are based on mimicking the physical characteristics and dynamic controls of a Synchronous Generator. The drawback in these strategies is that they are controlling a system (inverter) which has fast actuation but almost no inherent energy storage (no inertia) and mimicking with a system (Synchronous Generator), having slow actuation but significant energy storage (in terms of inertia). The energy Storage is used in context of inertia of a system. Moreover, PLL is used in most strategies, which has stability issues, in the case of disturbance in a system.

This approach discusses the control strategy, when one or more inverter based DG's have to be dispatched in the grid. Conceptually inverters are oscillators that have to synchronize; however the constraint of communication channel is a challenge in their feasibility. In this approach we will discuss a control that is based on local measurements. Dispatchable Virtual Oscillator Control (dVOC) is a promising decentralized control strategy that requires only local measurements to induce grid-forming behavior with programmable droop characteristics.

Conventionally, the power inverters are grid following. In this approach, the inverters are given pre-determined set point of power and the control regulates the current to achieve these power set points. In a highly dominated inverter-based grid, without synchronous generators, these control strategies are no longer suitable as they cannot maintain stability.

3.1.1 Grid Following and Grid Forming Inverters

Currently, most PV based inverters operate as grid- following (GFL) sources that control the inverter, by measuring the frequency or angle of the grid voltage using a phase-locked loop along with its voltage magnitude. The controller is defined in such a way, to match the voltage and frequency of the grid; irrespective of the disturbances. Hence they follow the grid voltage and frequency and do not contribute in frequency or voltage support to grid, in-

case of power imbalance. The main target of these PV based inverters is to maximize power after voltage and frequency stabilization.

A grid-forming inverter (GFM) on the other hand, actively controls its frequency and voltage output and contributes in frequency and voltage support of the grid. A more power demand by the grid means, the frequency is decreasing and frequency support is required to balance the power. Figure 3.1 and figure 3.2 represents block diagram of a grid following(GFL) and a grid forming(GFM) inverter respectively [19].

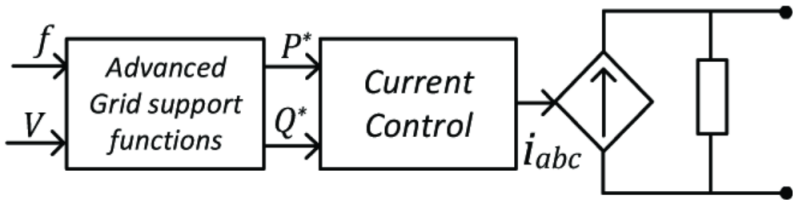


Figure 3.1: *Grid following inverter control*

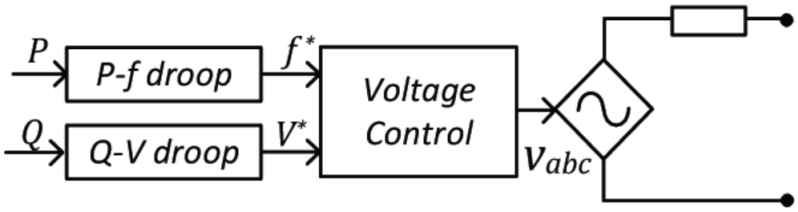


Figure 3.2: *Grid forming inverter control*

It can be seen that the grid following inverters track the power set points(which are maximized in general) by regulating their current while grid forming inverters track the frequency and voltage. The rotational speeds of synchronous generators is directly linked to the electrical output frequency hence they are controlling the frequency of the grid which make them grid forming elements.

Grid-forming inverters can help to stabilize the grid and in order to achieve frequency and voltage commands from real power measurements and reactive power measurements, droop control is widely used as shown is figure 3.3.

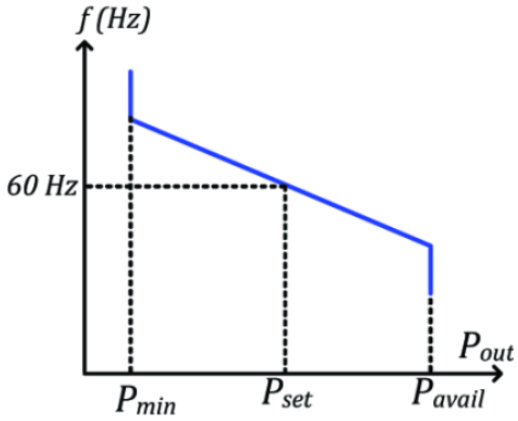


Figure 3.3: *Power frequency droop for grid forming inverters*

We can see that in grid forming inverter, the variable power is calculated by measuring output voltage and current. From this calculated active and reactive power, controlled voltage and controlled frequency variables are computed. In contrast to this, grid following inverters measure frequency and voltage and generate set points for active and reactive power using the frequency watt relation as depicted in figure 3.4.

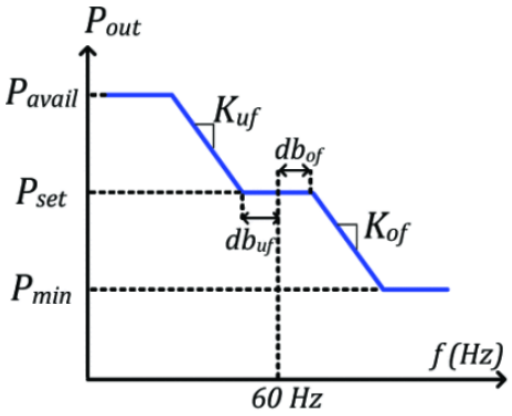


Figure 3.4: *Frequency-watt relationship for grid following inverters*

We can see that the grid following inverter in figure 3.4 shows a dead-band. This means that the inverter will not change power based on frequency variation, as the frequency is mostly balanced

by Synchronous Generators (SG) in the grid. Whereas, grid forming inverters will change their power based on frequency variation to actively contribute to stabilize the grid frequency.

The fundamental behavior of a grid-forming inverter is to operate as a voltage source while for the grid following inverter is to operate as a current source. Grid following inverters are used in grids with domination of synchronous generation where frequency response is not much desired from inverters, while grid forming inverters are required in grids with high penetration of inverter based Distributed Generation.

3.2 Modeling and Control Architecture

In order to model the control for our inverter, we have to take into account that the inverter is grid forming in nature. As discussed earlier in the previous section, the inverter will be modeled as a controllable voltage source. Let us consider that N such inverters are connected in a balanced three phase grid. The node voltage/inverter output voltage at each node is represented by v_k and belongs to $\mathbb{R}^2$ being in α, β co-ordinates. This transformation from a, b, c to α, β is called Clarke transformation and is represented in equation 3.1 [20]. The purpose of this transformation is to simplify our calculations by using two dimensional vector, instead of three-dimensional vectors for the voltage and current variables.

$$v_\alpha = \frac{2}{3}(v_a) - \frac{1}{3}(v_b - v_c); v_\beta = \frac{2}{\sqrt{3}}(v_b - v_c) \tag{3.1}$$

Similarly, the inverter output current from each node is represented by $i_{o,k}$ and belongs to $\mathbb{R}^2$ being in α, β co-ordinates and can be calculated using equation 3.1. Since the inverter is modeled as controllable voltage source, the system dynamic equation 3.2 is given as:

$$\frac{\mathrm{d}v_k(t)}{\mathrm{d}t} = u_k(v_k, i_{o,k}) \tag{3.2}$$

It can be seen from the equation that output voltage is fully controllable but it is based on the local measurements of current $i_{o,k}$ and voltage v_k. The controllable voltage source model for decentralized control is shown in figure 3.5.

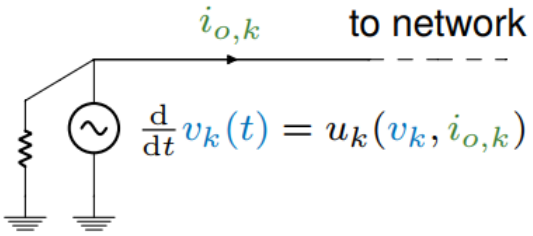


Figure 3.5: *Decentralized control setup of inverter*

We assume in this model that the transmission lines are homogeneous, i.e., the ratio between the inductance and resistance of each transmission line is constant, and there exist $\rho > 0$ so that:

$$\frac{\ell_{jk}}{r_{jk}} = \rho, \quad \forall(j,\ k)$$

where r_{jk} and ℓ_{jk} are the resistance and inductance of the line (j, k) respectively. This assumption is realistic due to structure(geometry) of transmission lines and also the transmission lines are constructed with the same material across the network.

3.2.1 Control Objectives

In order to implement equation3.2, the controller needs to be decentralized, which means that it should rely on local measurements of voltages and currents. The objectives that need to be achieved with this controller are frequency synchronization and voltage control without overloading the inverter and violating the power flow equations.

Given the AC voltage v_k at a node and the AC current $i_{o,k}$ flowing out of the node, the instantaneous active and reactive powers are calculated using equation3.3 and equation3.4.

$$p_k = v_k^\top i_{o,k} \tag{3.3}$$

$$q_k = v_k^\top J i_{ok} \tag{3.4}$$

Where v_k represents the voltage vector in $\alpha\beta$ co-ordinate system, which can be calculated using Clark transform or given by equation 3.1. Similarly, $i_{o,k}$ represents the output current vector in $\alpha\beta$ co-ordinate system, which can be calculated using Clark transform or given by equation 3.1. In equation 3.4, J represents the 2-D Rotation matrix evaluated at $\pi/2$. The rotation matrix is defined in equation 3.5.

$$R(\kappa) = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) \\ \sin(\kappa) & \cos(\kappa) \end{bmatrix} \tag{3.5}$$

J is defined in equation3.6 as:

$$J = R(\pi/2) \Rightarrow J = \begin{bmatrix} \cos(\pi/2) & -\sin(\pi/2) \\ \sin(\pi/2) & \cos(\pi/2) \end{bmatrix} \tag{3.6}$$

Now we know that active and reactive power cannot be given to the inverter arbitrary, rather they needs to be consistent with the power flow equations given as 3.7 and 3.8:

$$p_k^\star = \sum_{(j,k)} \frac{v_k^{\star 2} r_{jk} - v_k^\star v_j^\star (r_{jk}\cos(\theta_{jk}^\star) + \omega_0 \ell_{jk}\sin(\theta_{jk}^\star))}{r_{jk}^2 + \omega_0^2 \ell_{jk}^2} \tag{3.7}$$

$$q_k^\star = \sum_{(j,k)} \frac{v_k^{\star 2} \omega_0 \ell_{jk} - v_k^\star v_j^\star (\omega_0 \ell_{jk}\cos(\theta_{jk}^\star) - r_{jk}\sin(\theta_{jk}^\star))}{r_{jk}^2 + \omega_0^2 \ell_{jk}^2}. \tag{3.8}$$

In equation 3.7 and 3.8, $v_k^\star$ is the magnitude of voltage set point for inverter k given as $v_k^\star = \sqrt{v_{k,\alpha}^2 + v_{k,\beta}^2}$, $v_j^\star$ is the magnitude of voltage set point for inverter j given as $v_j^\star = \sqrt{v_{j,\alpha}^2 + v_{j,\beta}^2}$, r_{jk} is the resistance of line between node k and node j, ℓ_{jk} is the inductance of line between node k and node j, $\theta_{jk}^\star$ is the required phase difference between two signal v_k and v_j and ω_0 is the grid nominal angular frequency.

After having the real power set point $p^\star$, reactive power set point $q^\star$, magnitude of voltage set point $v^\star$ and desirable frequency ω_0, which do not violate the power flow equations, we want our inverter to satisfy these conditions:

Frequency Synchronization: All voltage vectors $v_k, v_j, ...v_n$ of inverters which are in $\alpha\beta$ co-ordinate system, must behave as harmonic oscillators with frequency ω_0 at steady state given in equation 3.9.

$$\lim_{t\to\infty}\left[\frac{\mathrm{d}}{\mathrm{d}t}v_k(t) - \omega_0 J v_k(t)\right] = 0 \tag{3.9}$$

In other words, a voltage vector v_k at time t should be equal to the voltage vector at time 0, rotated with frequency ω_0. This equation is given in equation 3.10.

$$v_k(t) = R(\omega_0 t)v_k(0) \tag{3.10}$$

To prove equation 3.9, let assume that $v_\alpha(t) = V\cos\omega t$ and hence $v_\beta(t) = V\sin\omega t$.

$$v_k(t) = \begin{bmatrix} V\cos\omega t \\ V\sin\omega t \end{bmatrix}$$

Substituting $v_k(t)$ in equation 3.9 yields:

$$\omega = \omega_0$$

.

Voltage Magnitude: The second condition that the inverter will satisfy is the voltage magnitude. At steady state, all the vectors would have voltage magnitude set by the reference voltage. This condition is given in equation 3.11.

$$\|v_k\| = v_k{}^\star \tag{3.11}$$

Power flow: Each inverter will inject active power and reactive power at steady state, without violating the power flow equations. This condition is given in equation 3.12.

$$\begin{cases} p_k^\star = v_k^\top i_{ok} \\ q_k^\star = v_k^\top J i_{ok} \end{cases} \tag{3.12}$$

Also the relative phase angle $\theta_{jk}^\star$ should follow equation 3.13

$$v_k(t) = R(\theta_{jk}^\star) v_j(t) \tag{3.13}$$

The control challenges and required conditions are shown in figure 3.6 and 3.7. We can see that the two node voltage vec-

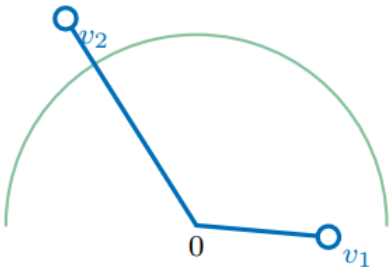


Figure 3.6: *Control challenges for two inverter system*

tors v_1 and v_2 represent different magnitudes, different angular frequency and large phase difference. Figure 3.7 represents the desired conditions at steady state as mathematically represented in equations 3.9-3.13. The voltages v_1 and v_2 follow reference voltage magnitude $v^\star$, they have same angular frequency as ω_0 and the phase angle is shown as $\theta_{12}^\star$.

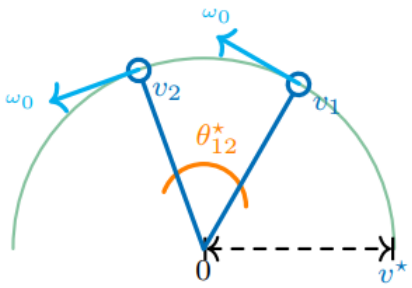


Figure 3.7: *Steady state conditions for two inverter system*

Hence the dynamics of the closed loop voltage controller which caters frequency matching, phase adjustment and voltage regulation is given in equation 3.14.

$$\frac{\mathrm{d}v_k}{\mathrm{d}t} = \begin{bmatrix} 0 & -\omega_0 \\ \omega_0 & 0 \end{bmatrix} v_k \; + \; c_1 e_{\theta,k}(v_k, i_{o,k}) \; + \; c_2 e_{\|v\|,k}(v_k) \quad (3.14)$$

This equation is obtained from a combination of 3.9-3.13 and cater all the controlling factors. First term in equation represents frequency synchronization by rotating v_k with frequency ω_0, second term $c_1 e_{\theta,k}(v_k, i_{o,k})$ represents phase synchronization and third term $c_2 e_{\|v\|,k}(v_k)$ represents voltage regulation. Figure 3.8 shows a pictorial representation of equation 3.14 in which the control is trying to synchronize frequency and phase along with voltage regulation.

Figure 3.8 shows an example of two inverter system, which are out of voltage limits and not synchronized. It can be seen that dark blue color represents the output voltage of inverter 1 and inverter 2, represented by v_1 and v_2 respectively. Light blue color represents the rotation of vectors v_1 and v_2 at the same nominal angular frequency ω_0.

The pink vectors $e_{\|v\|,1}(v_1)$ and $e_{\|v\|,2}(v_2)$ are trying to reduce the voltage error by reducing v_2 and increasing v_1. Similarly, the orange vectors $e_{\theta,1}(v)$ and $e_{\theta,2}(v)$ are trying to reduce the phase error as shown in figure 3.8.

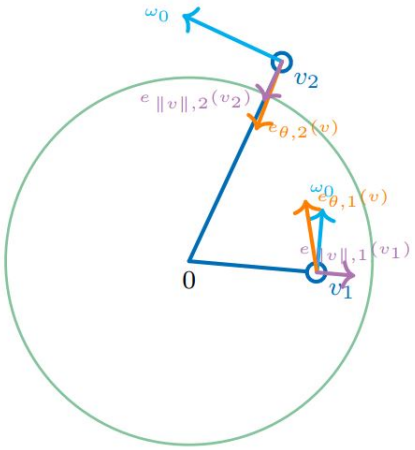


Figure 3.8: *Pictorial representation of equation 3.14*

3.3 The dVOC Control Law for Inverter Control

Dispatchable Virtual Oscillator Control (dVOC) is a dencentralized grid forming control strategy designed to achieve synchronization of an inverter-dominant grid, while maintaining a level of control on the power injections and voltage level of each inverter. dVOC has the following features:

- dVOC is dispatchable, i.e. it allows for the user to specify power set-points for each inverter.
- When no set-points are given, dVOC control shifts to VOC control and therefore it inherits all its favorable dynamical properties which are consistent with power flow equations.
- Since set-points are consistent with the AC power flow equations and other technical assumptions, dVOC makes the grid globally asymptotically stable with respect to the desired solution of the AC power-flow.

In order to find the control law, replace c_1 from equation 3.14 by η, and c_2 by $\eta\alpha$, where η and α are the design parameters for phase and voltage control. The first term of equation 3.14 is re-

placed by 3.9. Also the phase error term is given by equation 3.15 [21], where as magnitude error term is given by equation 3.16.

$$e_{\theta,k}(v_k, i_{o,k}) = K_k v_k - R(\kappa) i_{o,k} \tag{3.15}$$

$$e_{\|v\|,k}(v_k) = \phi_k(v_k) v_k \tag{3.16}$$

where K_k is given as:

$$K_k = \frac{1}{v_k^{\star 2}} R(\kappa) \begin{bmatrix} p_k^\star & q_k^\star \\ -q_k^\star & p_k^\star \end{bmatrix}.$$

$R(\kappa)$ is given as:

$$R(\kappa) = \begin{bmatrix} \cos(\kappa) & -\sin(\kappa) \\ \sin(\kappa) & \cos(\kappa) \end{bmatrix}$$

κ is defined as:

$$\kappa = \tan^{-1} \omega_0 \frac{\ell_{jk}}{r_{jk}}$$

Similarly in the magnitude error term in equation 3.16, $\phi_k(v_k)$ is defined as:

$$\phi_k(v_k) = \frac{v_k^{\star 2} - \|v_k\|^2}{v_k^{\star 2}}$$

where $\|v_k\|$ is the Euclidean norm or represents magnitude of the vector v_k. In the above definitions, $i_{o,k}$ and v_k represent the measured quantities where as $p^\star$, $q^\star$ and $v^\star$ represents the set points.

Substituting equation 3.9, 3.15 and 3.16 along with c_1 and c_2 in equation 3.14, the control law is given by equation 3.17 as:

$$\frac{\mathrm{d}v_k}{\mathrm{d}t} = \omega_0 J v_k \; + \; \eta(K_k v_k - R(\kappa) i_{o,k} \; + \; \alpha \phi_k(v_k) v_k) \tag{3.17}$$

Where $\omega_0 J v_k$ is the standard equation of a harmonic oscillator in rectangular coordinates. The parameter κ can be used to adjust the controller to adapt to the line parameters. $\kappa = 0$ corresponds to resistive lines and $\kappa = \pi/2$ corresponds to inductive lines.

Key points of control law: v_k and $i_{o,k}$ are in $\alpha\beta$ co-ordinates and in the case of single phase inverters, β component is constructed using Hilbert transform. We can see from equation of magnitude error 3.16 that if voltage of inverter v_k is equal to set point voltage $v^\star$, the term $\phi_k(v_k) = 0$ and hence the magnitude error $e_{\|v\|,k}(v_k)$ is zero. In order to find the condition for phase error to be zero, put equation 3.15 = 0 as:

$$e_{\theta,k}(v_k, i_{o,k}) = K_k v_k - R(\kappa) i_{o,k} = 0$$

putting parameters in above equation gives:

$$\frac{1}{v_k^{\star 2}} \begin{bmatrix} p_k^\star & q_k^\star \\ -q_k^\star & p_k^\star \end{bmatrix} v_k - i_{o,k} = 0$$

Upon solving, we conclude that the phase error term becomes zero when:

$$v_k^T i_{o,k} = p^\star$$
$$v_k^T J i_{o,k} = q^\star$$

Hence when the voltage and power injections corresponds to set points, the phase and magnitude error vanishes.

This clear that when all inverters connected with in a grid implement control law depicted in equation 3.17, and the set points are consistent with power flow equations along with technical assumptions, the grid will be asymptotically stable with respect to the power flows [21]. This implies that the inverters will synchronize and achieve set points irrespective of initial conditions.

In the case when set points are in-consistent with the power flow equations or not provided, this control presents droop like characteristics that enable grid synchronization.

3.4 Droop Characteristics of dVOC

In a conventional grid forming droop control as discussed earlier in this chapter, the controlling set points are voltage and frequency

which are obtained from active and reactive power mismatches as given in equation 3.18 and 3.19.

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \omega_k = \omega_0 + k_p(p_k^\star - p_k) \tag{3.18}$$

$$\frac{\mathrm{d} \parallel v_k \parallel}{\mathrm{d}t} = - \parallel v_k \parallel + v_k^\star + k_q(q_k^\star - q_k) \tag{3.19}$$

In the above equation, ω_k is the inverter output angular frequency, ω_0 is the grid nominal angular frequency, v_k is the inverter voltage where as $v_k^\star$ is the set point voltage. k_p and k_q are frequency and voltage droop co-efficients respectively.

Let us assume that the transmission line is inductive ,i.e., $r_{jk} = 0$ which yields $\kappa = \pi/2$. The harmonic oscillator equation $\omega_0 J v_k$ is given in equation 3.20 along with J and $R(\kappa)$ given as:

$$J = R(\kappa) = R(\pi/2) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$\omega_0 J v_k = \omega_0 \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} -\omega_0 v_\beta \\ \omega_0 v_\alpha \end{bmatrix} \tag{3.20}$$

Substituting $\kappa = \pi/2$; K_k becomes:

$$K_k = \frac{1}{v_k^{\star 2}} R(\kappa) \begin{bmatrix} p_k^\star & q_k^\star \\ -q_k^\star & p_k^\star \end{bmatrix} = \frac{1}{v_k^{\star 2}} \begin{bmatrix} q_k^\star & -p_k^\star \\ p_k^\star & q_k^\star \end{bmatrix}$$

$K_k v_k$ is given as:

$$K_k v_k = \frac{1}{v_k^{\star 2}} \begin{bmatrix} q_k^\star v_\alpha - p_k^\star v_\beta \\ p_k^\star v_\alpha + q_k^\star v_\beta \end{bmatrix} \tag{3.21}$$

$R(\kappa) i_{o,k}$ is given as:

$$R(\kappa) i_{o,k} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} -i_\beta \\ i_\alpha \end{bmatrix} \tag{3.22}$$

The phase error of equation 3.15 along with constant η becomes :

$$\eta e_{\theta,k}(v_k, i_{o,k}) = \eta(K_k v_k - R(\kappa) i_{o,k}) = \eta \begin{bmatrix} \frac{q_k^\star v_\alpha}{v_k^{\star 2}} - \frac{p_k^\star v_\beta}{v_k^{\star 2}} + i_\beta \\ \frac{p_k^\star v_\alpha}{v_k^{\star 2}} + \frac{q_k^\star v_\beta}{v_k^{\star 2}} - i_\alpha \end{bmatrix} \tag{3.23}$$

Similarly the magnitude error term along with other parameters is given as:

$$\eta\alpha\phi_k(v_k)v_k = \eta\alpha(\frac{v_k^{\star 2} - \|v_k\|^2}{v_k^{\star 2}}) \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix}$$

$$\eta\alpha\phi_k(v_k)v_k = \begin{bmatrix} \eta\alpha\phi_k(v_k)v_\alpha \\ \eta\alpha\phi_k(v_k)v_\beta \end{bmatrix} \tag{3.24}$$

Hence substituting 3.20, 3.23 and 3.24 in equation 3.17 gives the complete control law equation for dVOC.

$$\frac{\mathrm{d}v_k}{\mathrm{d}t} = \begin{bmatrix} \dot{v}_\alpha \\ \dot{v}_\beta \end{bmatrix} = \begin{bmatrix} \frac{\eta q_k^\star v_\alpha}{v_k^{\star 2}} - \frac{\eta p_k^\star v_\beta}{v_k^{\star 2}} + \eta i_\beta + \eta\alpha\phi_k v_\alpha - \omega_0 v_\beta \\ \frac{\eta p_k^\star v_\alpha}{v_k^{\star 2}} + \frac{\eta q_k^\star v_\beta}{v_k^{\star 2}} - \eta i_\alpha + \eta\alpha\phi_k v_\beta + \omega_0 v_\alpha \end{bmatrix} \tag{3.25}$$

Now using control law, we will show the droop characteristics for dVOC for which consider 3.18 and 3.19.

As we know:

$$\|v_k\| = \sqrt{v_\alpha^2 + v_\beta^2}$$

The derivative is calculated in equation 3.26as:

$$\frac{\mathrm{d}\|v_k\|}{\mathrm{d}t} = \frac{v_\alpha \dot{v_\alpha} + v_\beta \dot{v_\beta}}{\|v_k\|} \tag{3.26}$$

Where $\dot{v}_\alpha$ and $\dot{v_\beta}$ are extracted from equation 3.25 and given as:

$$\dot{v_\alpha} = \frac{\eta q_k^\star v_\alpha}{v_k^{\star 2}} - \frac{\eta p_k^\star v_\beta}{v_k^{\star 2}} + \eta i_\beta + \eta\alpha\phi_k v_\alpha - \omega_0 v_\beta$$

$$\dot{v_\beta} = \frac{\eta p_k^\star v_\alpha}{v_k^{\star 2}} + \frac{\eta q_k^\star v_\beta}{v_k^{\star 2}} - \eta i_\alpha + \eta\alpha\phi_k v_\beta + \omega_0 v_\alpha$$

Putting $\dot{v_\alpha}$ and $\dot{v_\beta}$ in equation 3.26 yields:

$$\frac{\mathrm{d}\|v_k\|}{\mathrm{d}t} = \frac{1}{\|v_k\|}[\frac{\eta q_k^\star v_\alpha^2}{v_k^{\star 2}} - \frac{\eta p_k^\star v_\beta v_\alpha}{v_k^{\star 2}} + \eta i_\beta v_\alpha + \eta\alpha\phi_k v_\alpha^2 - \omega_0 v_\beta v_\alpha$$
$$+\frac{\eta p_k^\star v_\alpha v_\beta}{v_k^{\star 2}} + \frac{\eta q_k^\star v_\beta^2}{v_k^{\star 2}} - \eta i_\alpha v_\beta + \eta\alpha\phi_k v_\beta^2 + \omega_0 v_\alpha v_\beta]$$

Upon simplification, we obtain:

$$\frac{\mathrm{d}\|v_k\|}{\mathrm{d}t} = \eta(\frac{q_k^\star}{v_k^{\star 2}} - \frac{q_k}{\|v_k\|^2})\|v_k\| + \frac{\eta\alpha}{v_k^{\star 2}}(v_k^{\star 2} - \|v_k\|^2)\|v_k\| \quad (3.27)$$

In the same fashion, we will obtain the frequency droop. As we know:

$$\theta = \tan^{-1}\frac{v_\beta}{v_\alpha}$$

Remember that v_α and v_β are time dependent. Hence:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = (tan^{-1}\frac{v_\beta}{v_\alpha})'$$

From trigonometric identities, we know that:

$$\frac{\mathrm{d}(tan^{-1}x)}{\mathrm{d}x} = \frac{1}{1+x^2}$$

Hence substituting $x = v_\beta/v_\alpha$ yields:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{v_\alpha\dot{v}_\beta - v_\beta\dot{v}_\alpha}{\|v\|^2} \quad (3.28)$$

Putting $\dot{v}_\alpha$ and $\dot{v}_\beta$ as obtained earlier, in equation 3.28 which yields:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{1}{\|v\|^2}(\frac{\eta p_k^\star v_\alpha^2}{v_k^{\star 2}} + \frac{\eta q_k^\star v_\beta v_\alpha}{v_k^{\star 2}} - \eta i_\alpha v_\alpha + \eta\alpha\phi_k v_\beta v_\alpha + \omega_0 v_\alpha^2$$

$$-\frac{\eta q_k^\star v_\alpha v_\beta}{v_k^{\star 2}} + \frac{\eta p_k^\star v_\beta^2}{v_k^{\star 2}} - \eta i_\beta v_\beta - \eta\alpha\phi_k v_\alpha v_\beta + \omega_0 v_\beta^2)$$

which on further simplification results in equation 3.29.

$$\frac{\mathrm{d}\theta_k}{\mathrm{d}t} = \omega_0 + \eta(\frac{p_k^\star}{v_k^{\star 2}} - \frac{p_k}{\|v_k\|^2}) \tag{3.29}$$

Equations 3.27 and 3.29 represent voltage droop and frequency droop respectively for the dVOC control of inverter satisfying power flows. We can see that dVOC is different from 3.18 and 3.19. In dVOC, $\omega - P$ droop is dependent on voltage which is different from conventional droop where as $V - Q$ droop is non-linear with less voltage drop by change of q. This droop control is valid for inductive lines i.e when $\kappa = \pi/2$. For resistive droop characteristics, $\kappa = 0$ which is similar to VOC [23].

Also in the case when power set points are not provided or they are inconsistent with power flow, the droop in equation 3.27 and 3.29 will become:

$$\frac{\mathrm{d}\|v_k\|}{\mathrm{d}t} = \eta(-\frac{q_k}{\|v_k\|}) + \frac{\eta\alpha}{v_k^{\star 2}}(v_k^{\star 2} - \|v_k\|^2)\|v_k\| \tag{3.30}$$

Similarly equation 3.29 will become:

$$\frac{\mathrm{d}\theta_k}{\mathrm{d}t} = \omega_0 + \eta(-\frac{p_k}{\|v_k\|^2}) \tag{3.31}$$

Equations 3.30 and 3.31 are similar to VOC control, however, unlike VOC, the dVOC can receive power and voltage set-points $p_k^\star$ $q_k^\star$, $v_k^\star$ and thus synchronizes the power system to the desired power-flow solution.

3.5 Implementation of dVOC

After developing the control law, the next step is finding a way to implement our strategy. A block diagram depiction of this strategy

is shown in figure 3.9.

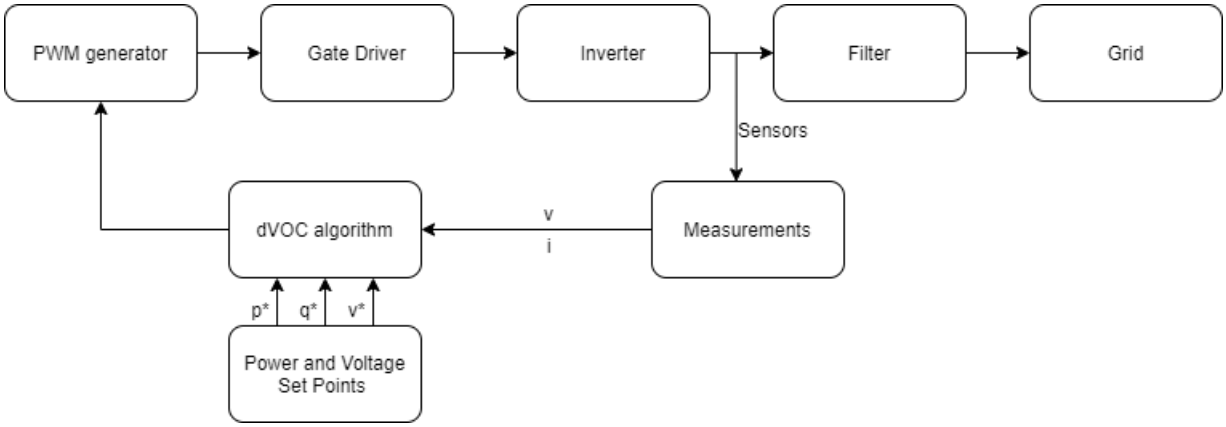


Figure 3.9: *Block diagram of proposed strategy*

It can be seen that the output voltage and current from inverter is measured using voltage and current sensors. Hall effect sensor is mostly used for voltage measurements. The measurement block synthesizes these current and voltage signals and perform alpha beta transformation.

The active power, reactive power and voltage set point are fed to the algorithm along with real measurements. The output of dVOC block is a voltage controlled signal $v_k(t)$ which is fed to the PWM block. The PWM block then generates a PWM signal and provide it to a gate driver which drives the switches of the Inverter. It is a closed loop system which tries to converge to the set points received by dVOC algorithm. The filter is used to filter out the higher harmonics and it serves us as an inductive network. The filtered output is then finally fed to the grid.

A more clear schematic of dVOC inverter for decentralized inverter-dominant grid is shown in figure 3.10.

It can be seen from figure 3.10, that the inverter output current is fed into an analog to digital converter(ADC) which converts the signal into digital domain. Then $\alpha\beta$ transformation is applied and the transformed signals along with set point $p^\star$, $q^\star$ and $v^\star$ is fed into dVOC algorithm which outputs voltage for the PWM

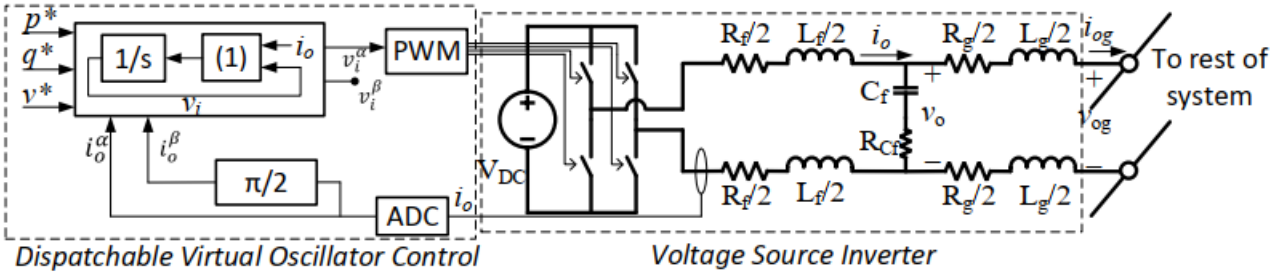


Figure 3.10: *Schematic of a dVOC inverter [24]*

generator.

The PWM module generates output signals by comparing the modulating signal with a triangular carrier. The output is then fed into a gate driver which drives the inverter switches. The inverter switches are mostly IGBT's which have capability of fast switching. All this functionality as shown by dotted lines in figure 3.10, is implemented by a Digital Signal Processor board. In this research, the board used is Texas Instrument 320F28069 and the details will be discussed in chapter 5.

The next chapter will discuss about simulations in PLECS software and their results.

CHAPTER 4

Simulation of dVOC Algorithm in PLECS

4.1 Introduction

PLECS (Piecewise Linear Electrical Circuit Simulation) is a software tool for system-level simulations of electrical circuits developed by Plexim. It is especially designed for power electronics but can be used for any electrical network [27]. For the current simulation, we are using the standalone version of PLECS that allows simulation of electrical circuits and control systems directly within the PLECS package.

In this chapter we will simulate the dVOC using control law given by equation 4.1 as:

$$\frac{\mathrm{d}v_k}{\mathrm{d}t} = \omega_0 J v_k \; + \; \eta[K_k v_k - R(\kappa) i_{o,k} \; + \; \alpha \phi_k(v_k) v_k] \qquad (4.1)$$

The explanation of this equation has already been discussed in Chapter 3, and only the implementation of this equation by simu-

lating in PLECS will be analyzed in this chapter. The schematic of the system is shown in figure 4.1.

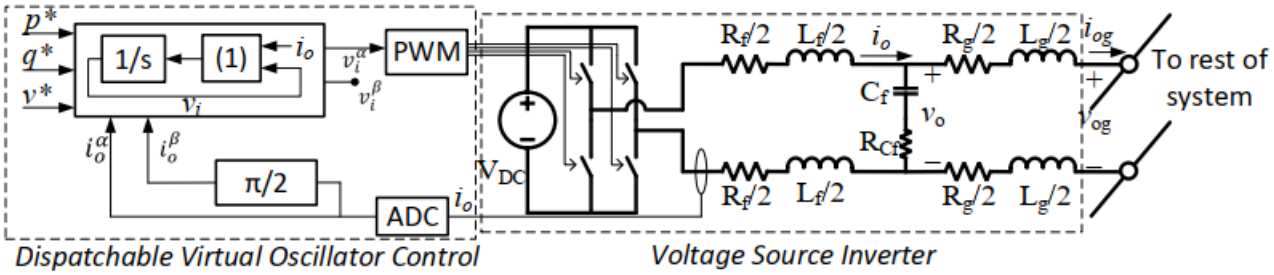


Figure 4.1: *Schematic of a dVOC inverter-based system [24]*

The system under consideration is a single phase system and the inverter is a single phase inverter for simplicity. There are different segments of control which will be implemented separately and will be discussed in this chapter. The different modules include active and reactive power calculations, voltage calculations, droop calculation and filter characteristics.

4.1.1 Importance of Filter

As we know that the output of an inverter is not a smooth sinusoidal waveform, rather it is pulsating and have a high value of Total Harmonic Distortion (THD). It has harmonics which needs to be filtered out before connecting it with the load or integrating to the grid. The harmonics can be eliminated using filters but they have different advantages and drawbacks. A filter is usually placed between the inverter and the grid to attenuate the switching frequency harmonics produced by the grid-connected inverter and the filter used in this simulation is a LCL filter. Compared with L filter, LCL filter has better attenuation capacity of high-order harmonics and better dynamic characteristic [30].

The three common filter schematics are shown in figure 4.2, where VSI represents output of Voltage Source Inverter.

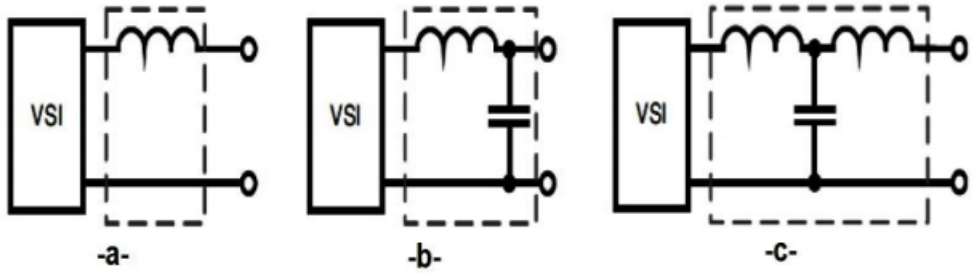


Figure 4.2: *Schematics of L, L-C and L-C-L filters [27]*

Filters comparison: The L-filter is the first order filter with attenuation 20 dB/decade. Therefore the application of this filter type is suitable for converters with high switching frequency, where the attenuation is sufficient.To reduce harmonics, a high value of input inductance should be used which makes this filter an expensive solution.

The LC-filter is a second order filter and it has better damping behavior than L-filter. However, higher inductance value is required to achieve demanded cut-off frequency of the filter which increases cost and size of the inductor. Also connection of this filter to the supply grid, will cause the resonant frequency of the filter, dependent on the grid impedance and therefore this filter is not suitable, too. The transfer function of L-C filter is given in equation 4.2.

$$F(s) = \frac{1}{1 + sL + s^2LC} \tag{4.2}$$

The attenuation of the LCL-filter is 60 dB/decade for frequencies above resonant frequency, therefore lower switching frequency for the inverter can be used. It also provides better decoupling between the filter and the grid impedance and lower current ripple across the grid inductor. The LCL filter has good current ripple attenuation even with small inductance values which makes it a cost effective solution too. The cut-off frequency of LCL filter can be calculated by equation 4.3, whereas the frequency response can

be calculated using equation 4.4.

$$f_{res} = \frac{1}{2\pi}\sqrt{\frac{L_f + L_g}{L_f L_g C_f}} \tag{4.3}$$

$$H(s) = \frac{1}{s(L_f + L_g) + s^3 L_f L_g C} \tag{4.4}$$

where L_f is the inverter side inductor while L_g is grid side inductor and C is the Capacitance of the filter capacitor.

4.1.2 Filter Design

A typical LCL filter is shown in figure 4.3.

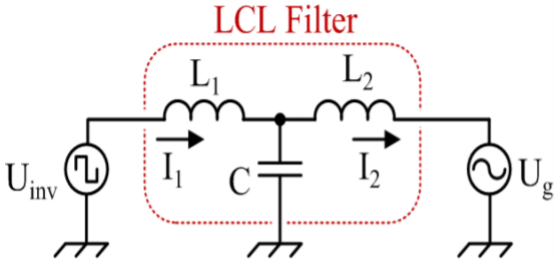


Figure 4.3: *A grid connected LCL filter [30]*

In figure 4.3, U_{inv} is the inverter output voltage while U_g is the grid voltage. The attenuation of a typical LCL filter is 60dB/decade. The cut-off frequency (fc) of the low pass filter is selected such that the output voltage THD(Total Harmonic Distortion) is less than 5% [29]. However, the filter cut-off frequency limits the control band-width of inverter systems. Increasing the control bandwidth is important for fast operation of the inverter system hence here is a trade-off between the attenuation effect and the control bandwidth. Generally, the resonant frequency should be greater than $10f_1$ and less than half of the switching frequency [31] as given by:

$$f_{res} > 10f_1$$

$$f_{res} < 0.5 f_{sw}$$

Combining both conditions of f_{res} yields:

$$10 f_1 < f_{res} < 0.5 f_{sw} \tag{4.5}$$

where f_1 is the grid frequency and and f_{sw} is the switching frequency. Another condition for filter design is the limitation on ripple current which should be $0.15 \sim 0.25$ of rated current. In this filter design, we selected the ripple current not to exceed 20% of the rated current. The equation for maximum current ripple is given in equation 4.6 [32].

$$\Delta I_{L1max} = \frac{U_{dc}}{8 L_1 f_{sw}} < 0.2 I_{rated} \tag{4.6}$$

where U_{dc} is the dc voltage, and the value in our simulation is $120V$. I_{rated} is the rated current given as:

$$I_{rated} = \frac{P_n \sqrt{2}}{U_n} \tag{4.7}$$

where U_n is the output rated voltage while P_n is the rated active power. The circuit parameters are given in table 4.1.

Table 4.1: *Circuit parameters*

Parameter	Value	Unit
f_{sw}	32000	Hz
f_1	60	Hz
U_{dc}	120	V
P_n	1000	W
U_n	120	V

Putting values of table 4.1 in equation 4.6 and 4.7 gives the value of inductance as:

$$L_1 = 1mH$$

Similarly, for grid side inductor, the value should be chosen 4-6 times lesser than inverter-side inductor as discussed in research paper [33]. Hence grid side inductor is given as:

$$L_1 = (4 \sim 6)L_2$$

In our design, we chose factor of 5 which gives value of L_2 as:

$$L_2 = 0.2mH$$

Substituting f_1 and f_{sw} in equation 4.5, gives range for f_{res} as:

$$600Hz < f_{res} < 16000Hz$$

In our design, we chose $2.5kHz$ as the cut-off frequency. Keep in mind that $L_1 = L_f$, $L_2 = L_g$ and $C = C_f$ as different notations are used in figure 4.3, and equations 4.3 and 4.4.

Hence putting values of L_f, L_g and f_{res} in equation 4.3 gives capacitance value to be:

$$C_f = 24\mu F$$

The filter parameter are also re arranged in table 4.2.

4.2 Schematics in the simulation environment

In this section, we will discuss the schematics of our system in PLECS software. The basic schematic of our system in PLECS is given in figure 4.4.

In figure 4.3, we can see that 4 switches of an inverter have to be controlled in such a way that the dVOC control can be implemented. The switches used are IGBT's which are capable of fast switching. "V_dc" is the DC voltage source which represents voltage from a PV (RES) source after the dc-dc converter stage.

The output of inverter is a pulsating signal having harmonics which needs to be eliminated by a filter. The filters are already discussed in Section 4.1 and in this configuration, LCL filter is

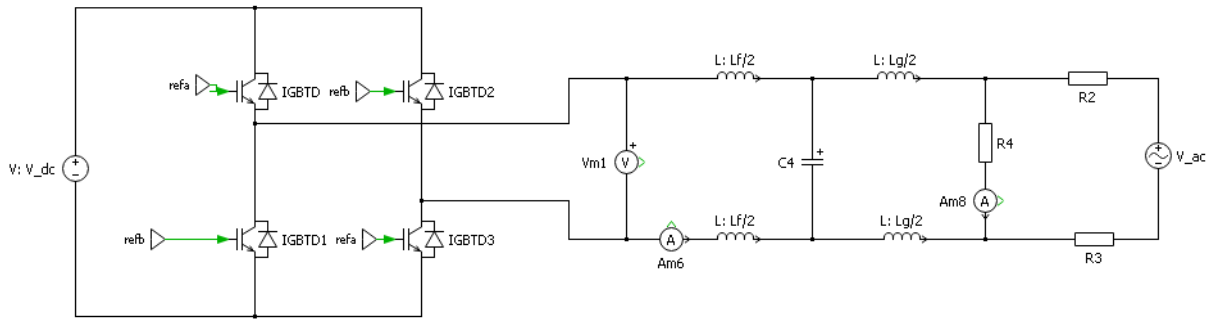


Figure 4.4: *Basic schematic in PLECS*

used. The values of the filter parameters are calculated above and given in table 4.2.

Table 4.2: *Filter parameters*

Component	Value	PLECS constant	Unit
Lf	1	Lf	mH
Lg	0.2	Lg	mH
$C4$	24	Cf	μF

The output is then connected to a load "R4". The value of R4 is chosen to be 28.8Ω which is based on power consumption by the load, which is assumed to be $500W$. The transmission line/cable resistance is denoted by "R2" and "R4" to be 0.157Ω for approximately 1 mile length. The data is obtained from data-sheet for transmission lines [28]. It will be then connected to the grid represented by V_ac. The constant used to assign value to V_ac is "Vph_m" and its value is $120\sqrt{2}rms$. The frequency of the grid is given by variable "f" as 60Hz.

We must note that Voltmeter "Vm1" is used to measure the output voltage which contains harmonics. In-order to utilize this measurement by control algorithm, we have to apply the filter on this measured signal. In-built filters are available in PLECS and hence are utilized as shown in figure 4.5.

The three RC filters are single order filters combine to give 3rd order filter with a 60 dB/decade attenuation, for frequencies above

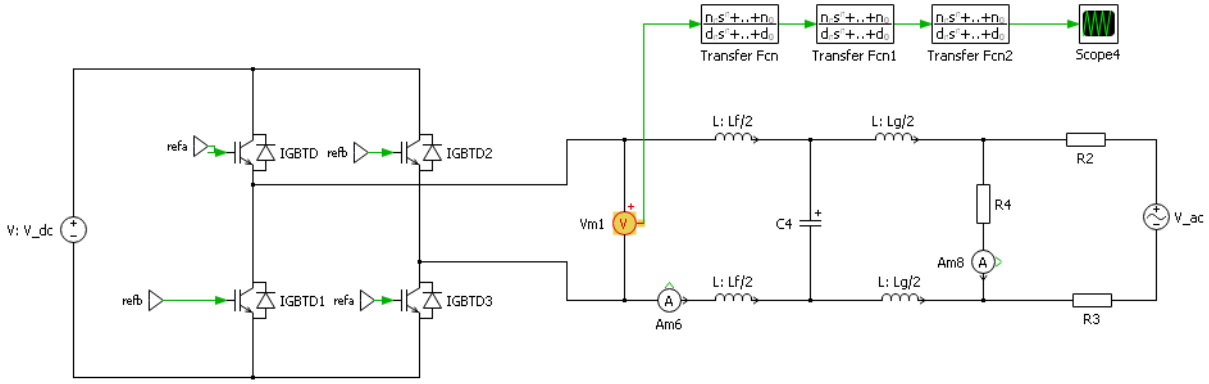


Figure 4.5: *Basic schematic with in-built filter*

resonant frequency. The output voltage signal after filtration is not yet the desired signal as it has to be transformed into $\alpha\beta$ domain to follow the dVOC control logic. The schematic with $\alpha\beta$ transformation block is shown in figure 4.6.

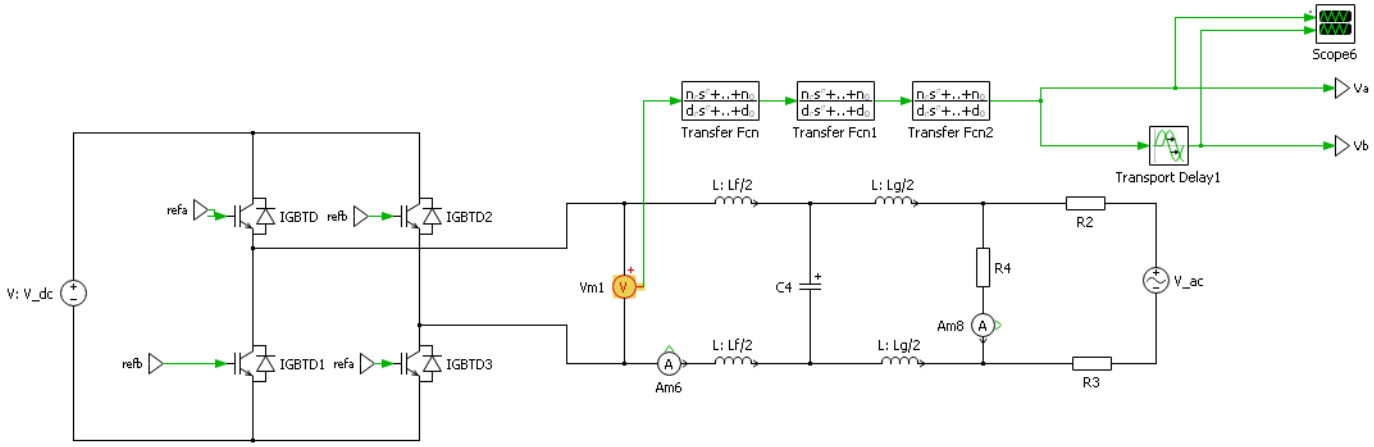


Figure 4.6: *Basic schematic with filter and $\alpha\beta$ transformation*

Transport delay block: The transport delay block delays a signal in time. Since we know that the β component is displaced by 90° or $\pi/2$ from α component. Let's assume that $v_\alpha(t) = V \sin \omega t$ then v_β is given as:

$$v_\beta(t) = V \sin(\omega t - \pi/2)$$

Hence the output voltage is given by equation 4.8 as:

$$v_k(t) = \begin{bmatrix} V \sin \omega t \\ V \cos \omega t \end{bmatrix} \tag{4.8}$$

As 360° corresponds to complete cycle or T (Time period) in the time domain , Similarly, 90° corresponds to $T/4$ or $1/4f$. Hence, the parameters of time delay block and output of $\alpha\beta$ transformation is shown in figure 4.7 and 4.8 respectively.

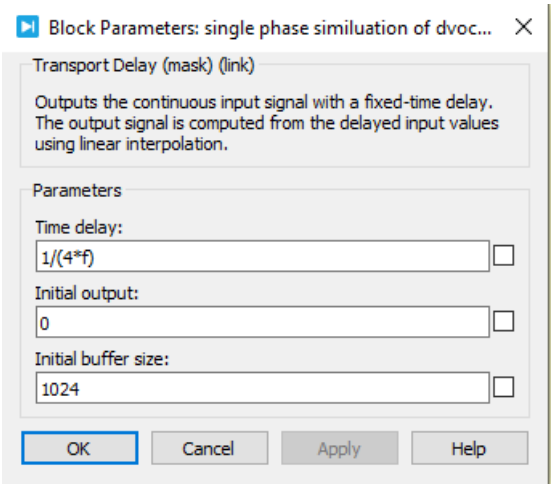


Figure 4.7: *Transport delay block parameters*

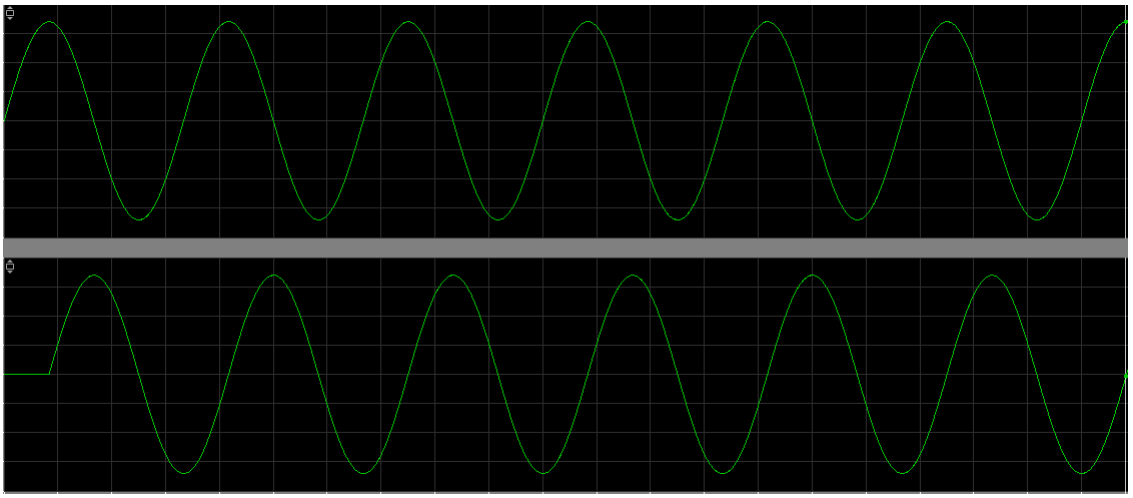


Figure 4.8: *Output of $\alpha\beta$ transformed signal*

Where f is 60Hz initialized at the beginning of simulation. First signal in figure 4.8 is the α signal whereas the other is represented by β signal displaced by $90°$. In the similar way, output current needs to be measured for dVOC algorithm and then transformed into $\alpha\beta$ domain. The inverter output current, measured by ammeter "Am6" along with $\alpha\beta$ transformation of current and voltage signals is shown in figure 4.9.

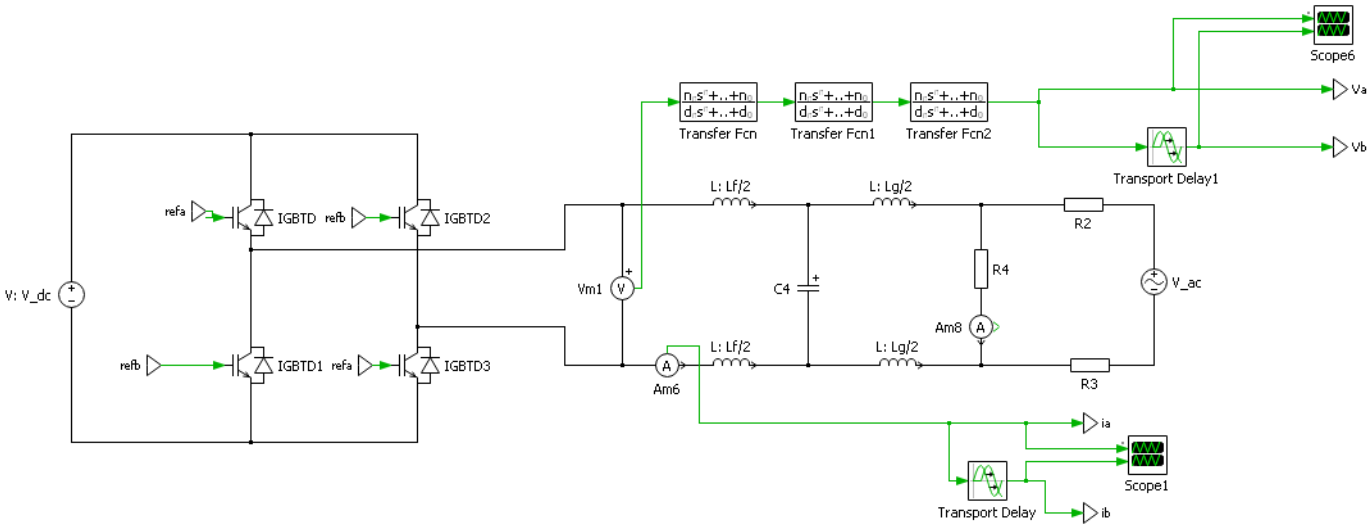


Figure 4.9: *Current measurement along with $\alpha\beta$ transformation*

It must be noted that current measurement "Am6", is opposite to the flow of current, hence a negative sign will be used in the control, to cater this problem. Power set point $p\star$ and $q\star$ are represented by p_ref and q_ref respectively and shown in figure **??**. Now we will move to the functionality blocks which are involved in the control of algorithm. Control blocks are shown in figure 4.11, while the whole schematic is shown in figure 4.12.

4.3 Control Blocks and their functionality

In this section, we will discuss the role of each block along with their implementation. The blocks used are :

- Reference Voltage.

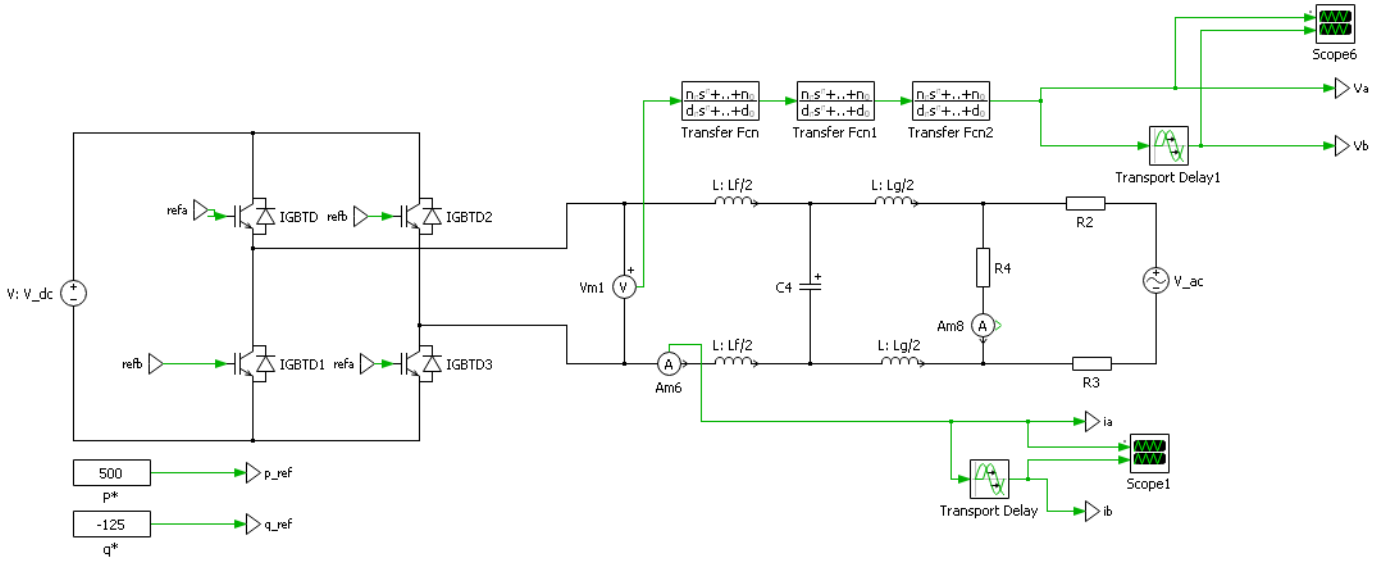


Figure 4.10: *PLECS schematic along with power set points*

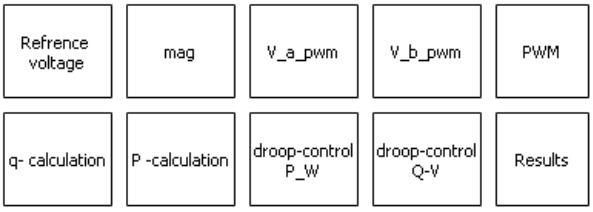


Figure 4.11: *Control blocks in simulation*

- mag.
- V_a_pwm.
- V_b_pwm.
- PWM.
- q-calculation.
- p-calculation.
- droop-control P_W.
- droop-control Q-V.
- Results.

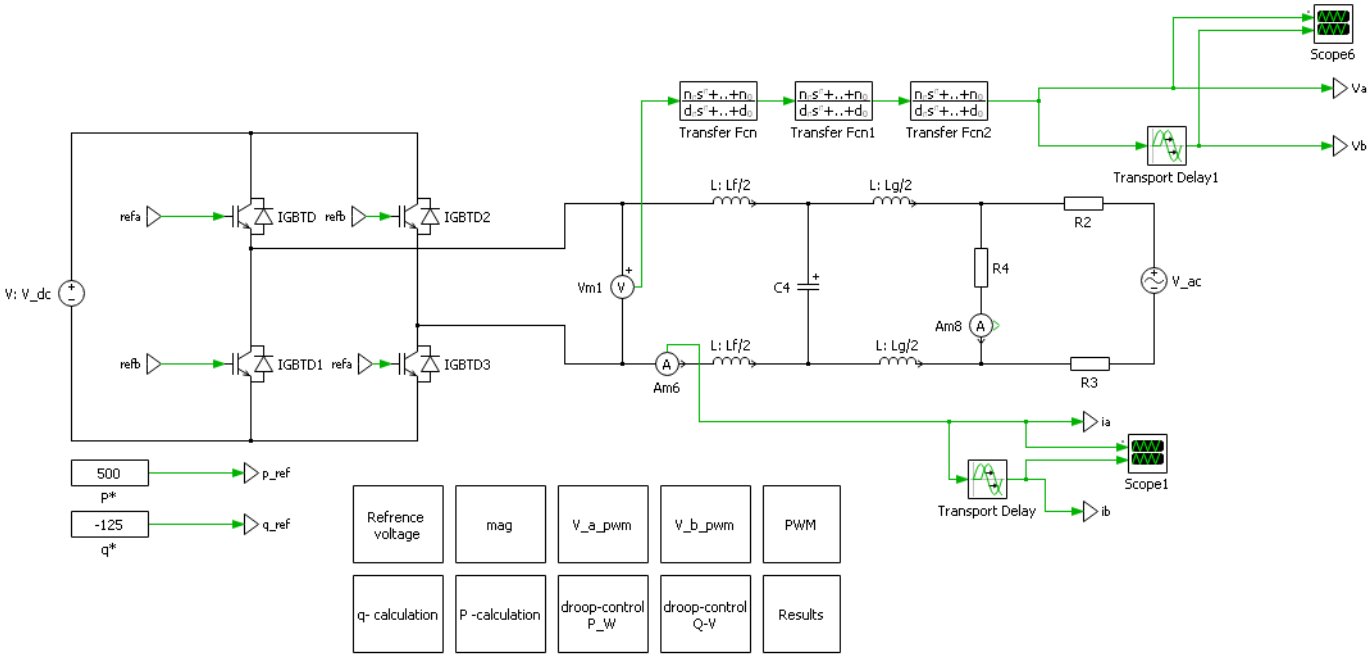


Figure 4.12: *Complete schematic of system in PLECS*

4.3.1 Reference Voltage Block

This block generates the reference signals to be used in equation 3.25. The reference signal is specified by the user in our simulation; however, the block transforms the voltage reference signals into $\alpha\beta$ domain to be utilized by the algorithm. The implementation is shown in figure 4.13 and the output in figure 4.14.

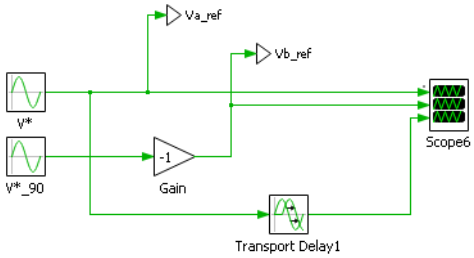


Figure 4.13: *Implementation of reference voltage block*

It can be seen that the transport block is just delaying the input signal, however, we need the β component value in that duration of time too. Hence we implemented a phase shift in 90° followed

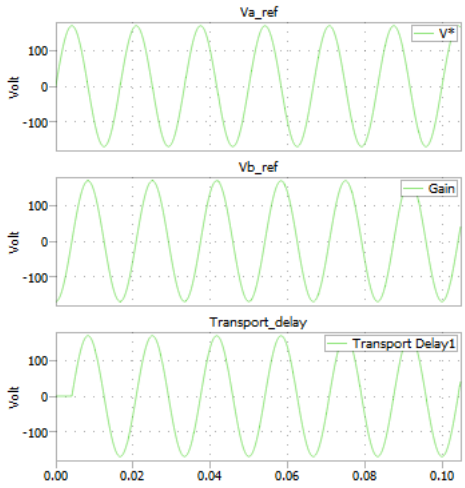


Figure 4.14: *Scope output of reference voltage*

by a gain of "-1" to get β component.

4.3.2 Magnitude Block

The magnitude block is used to calculate the magnitude of reference voltage, grid voltage and output voltage of inverter. The magnitude of these voltages are used in control law implementation given by equation 3.25. The implementation of the magnitude block is shown in figure 4.15 where "v_ref" is $v^{\star}$, "V" is the

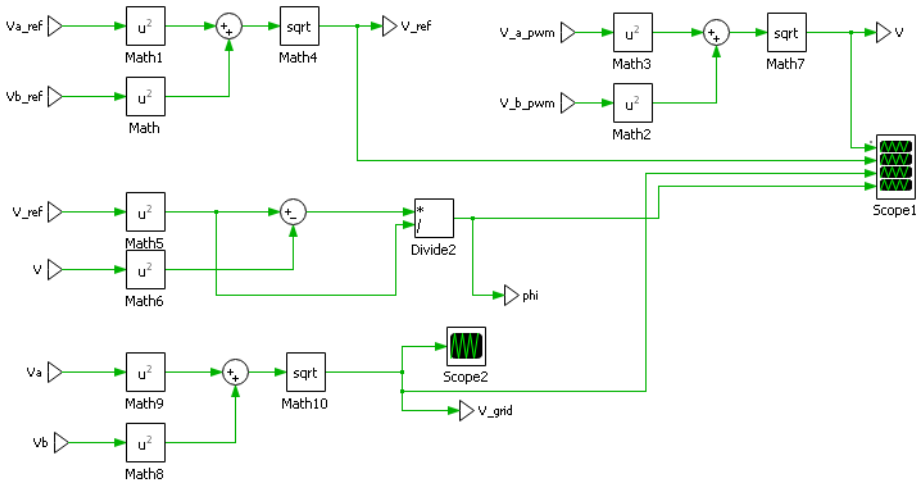


Figure 4.15: *Implementation of Magnitude Block*

controlled voltage magnitude v in equation 3.25. "V_grid" is the measured grid voltage and "phi" is the term defined in equation 3.16. The equations can be redefined here:

$$v^\star = \sqrt{v_\alpha^{\star 2} + v_\beta^{\star 2}}$$

equivalently:

$$v\_ref = \sqrt{va\_ref^2 + vb\_ref^2}$$

In the same way, "v_a_pwm" is equivalent to v_α and "v_b_pwm" is equivalent to v_β as shown in equation 3.25. Similarly "v_grid" is equivalent to measured grid voltage while "v_a" and "v_b" are v_α and v_β equivalents of the measured grid voltage respectively.

"phi" is equivalent to ϕ of equation 3.25 and ϕ is given as:

$$\phi = \frac{v^{\star 2} - \|v\|^2}{v^{\star 2}} \tag{4.9}$$

equivalently in simulation:

$$phi = \frac{V\_ref^2 - V^2}{V\_ref^2} \tag{4.10}$$

4.3.3 V_a_pwm Block

This block generates the signal to drive the inverter implementing dVOC. This is the main block in our simulation and implements equation 3.25 which is given as:

$$\frac{\mathrm{d}v_k}{\mathrm{d}t} = \begin{bmatrix} \dot{v}_\alpha \\ \dot{v}_\beta \end{bmatrix} = \begin{bmatrix} \frac{\eta q_k^\star v_\alpha}{v_k^{\star 2}} - \frac{\eta p_k^\star v_\beta}{v_k^{\star 2}} + \eta i_\beta + \eta\alpha\phi_k v_\alpha - \omega_0 v_\beta \\ \frac{\eta p_k^\star v_\alpha}{v_k^{\star 2}} + \frac{\eta q_k^\star v_\beta}{v_k^{\star 2}} - \eta i_\alpha + \eta\alpha\phi_k v_\beta + \omega_0 v_\alpha \end{bmatrix} \tag{4.11}$$

"V_a_pwm" is equivalent to v_α and the following equation is implemented in this block:

$$V\_a\_pwm = v_\alpha = \int \left(\frac{\eta q^\star v_\alpha}{v^{\star 2}} - \frac{\eta p^\star v_\beta}{v^{\star 2}} + \eta i_\beta + \eta\alpha\phi v_\alpha - \omega_0 v_\beta\right)$$

Where $\eta = 21.71\Omega rad/sec$ and $\alpha = 0.9722\mho$ obtained from [24]. The block schematic is given in figure 4.16.

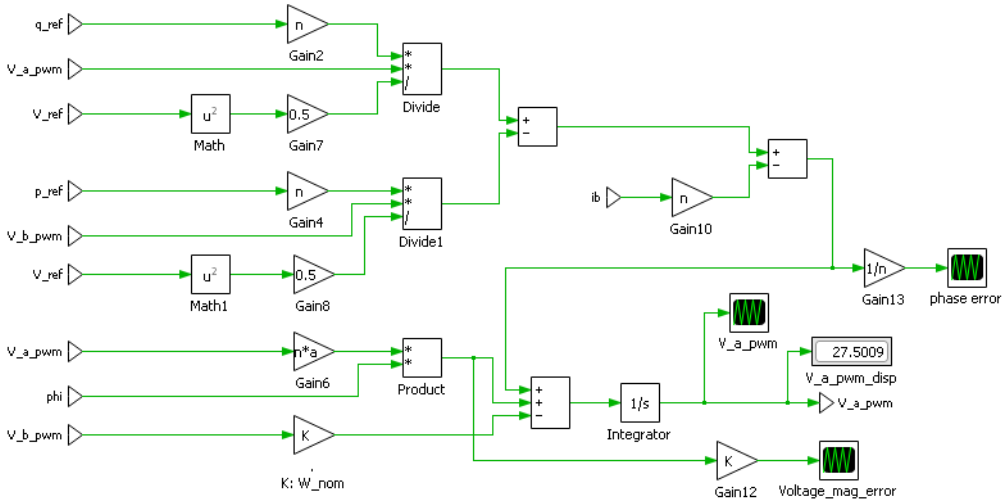


Figure 4.16: *Implementation of V_a_pwm Block*

4.3.4 V_b_pwm Block

As described in section 4.3.3, V_b_pwm is implemented in the same way. The equation that govern V_b_pwm is derived from 4.11 or 3.25.

$$V\_b\_pwm = v_\beta = \int (\frac{\eta p^\star v_\alpha}{v^{\star 2}} + \frac{\eta q^\star v_\beta}{v^{\star 2}} - \eta i_\alpha + \eta\alpha\phi v_\beta + \omega_0 v_\alpha)$$

The implementation of this block is given in figure 4.17.

4.3.5 PWM Block

After calculation of the voltage signals "V_a_pwm" and "V_b_pwm" for the control of inverter using dVOC technique, the next step is feeding the signal to the inverter. The inverter consists of switches and hence a digital signal is required to operated the switches(IGBT's). This is usually implemented using a PWM generator block in which digital signal is produced at the output according to input signal and carrier. Input signal is compared with a carrier signal and a switching output is generated. The carrier frequency in this simulation is chosen to be $f_s = 32kHz$. The implementation of PWM block is shown in figure 4.18.

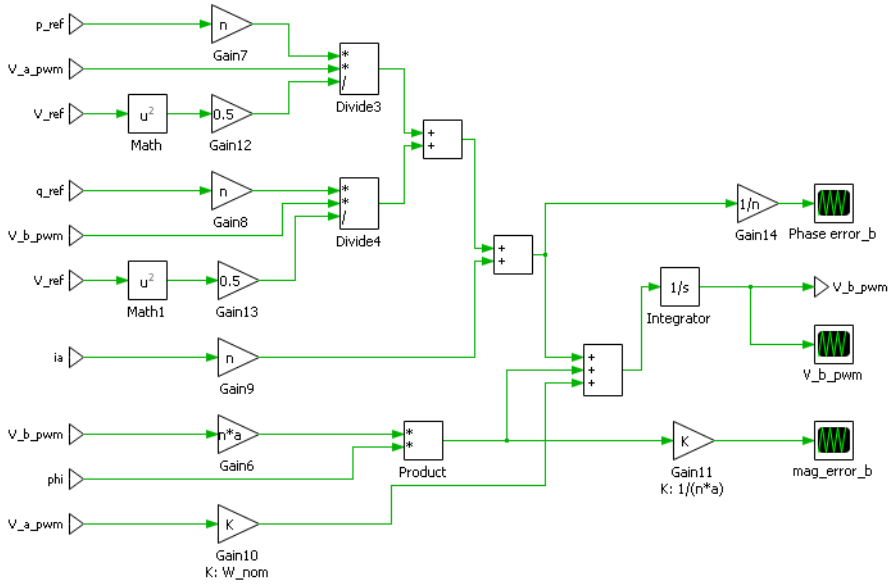


Figure 4.17: *Implementation of V_b_pwm Block*

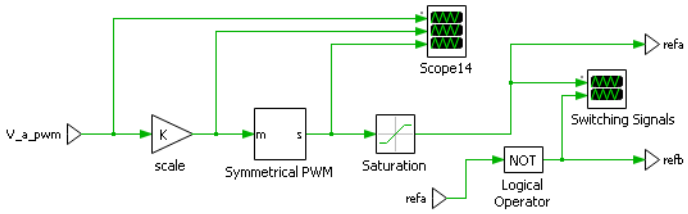


Figure 4.18: *Implementation of PWM Block*

Where "K" is the gain of "1/V_ph" which normalizes the input voltage so that the condition of input voltage for the Symmetrical PWM block can be satisfied, which is given as:

$$-1 \leq v_{in} \leq 1$$

V_a_pwm, normalized V_a_pwm and the corresponding PWM is shown in figure 4.19 and 4.20 This signal is then passed through a saturation block to get PWM signal between level 0 and level 1.

4.3.6 q-Calculation Block

In this section, reactive power is calculated based on equation given in Chapter 3 as:

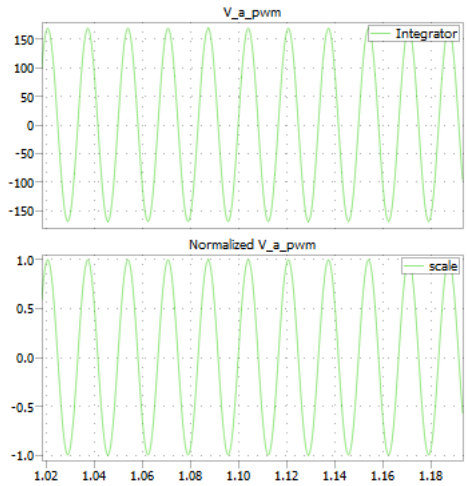


Figure 4.19: *V_a_pwm and normalized V_a_pwm of PWM Block*

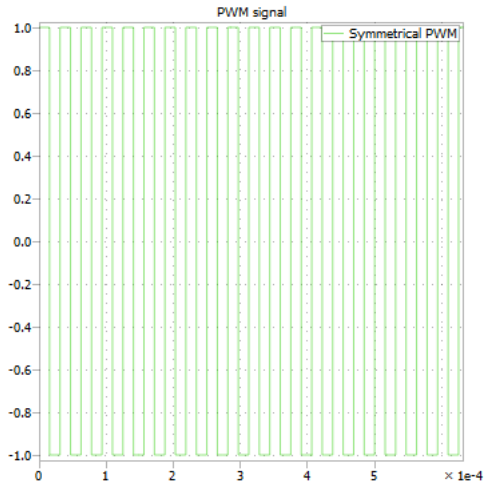


Figure 4.20: *Output PWM signal of PWM Block*

$$q = v^T J i$$

$$q = v_\beta i_\alpha - v_\alpha i_\beta$$

Gain "-1" is used for current adjustment due to measurement of current in the opposite direction, while gain "0.5" is used for rms and peak adjustment. The implementation is shown in figure 4.21

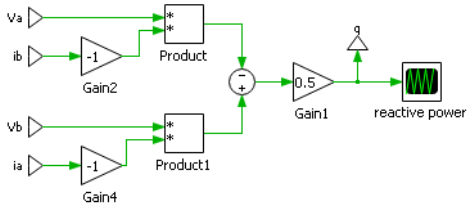


Figure 4.21: *Implementation of q-Calculation Block*

4.3.7 P-Calculation Block

In this section, active power is calculated, similar to q-Calculation block based on equation given in Chapter 3 as:

$$p = v^T i$$

$$p = v_\alpha i_\alpha + v_\beta i_\beta$$

In single phase, active power is also calculated as $v_\alpha i_\alpha$. The implementation is shown in figure 4.22.

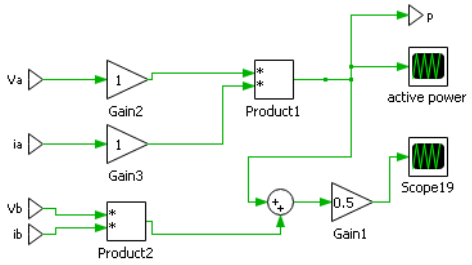


Figure 4.22: *Implementation of P-Calculation Block*

4.3.8 Droop Contol P-W Block

This block is implemented to prove the frequency droop characteristics of dVOC. The frequency droop characteristics are given by equation 3.29 from chapter 3 which is re-written as:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \omega_i = \omega_0 + \eta\left(\frac{p^\star}{v^{\star 2}} - \frac{p}{\|v\|^2}\right) \tag{4.12}$$

Where ω_i is angular frequency of output voltage and ω_0 is the grid nominal angular frequency. The Implementation of droop characteristics of equation 4.12 are given in figure 4.23.

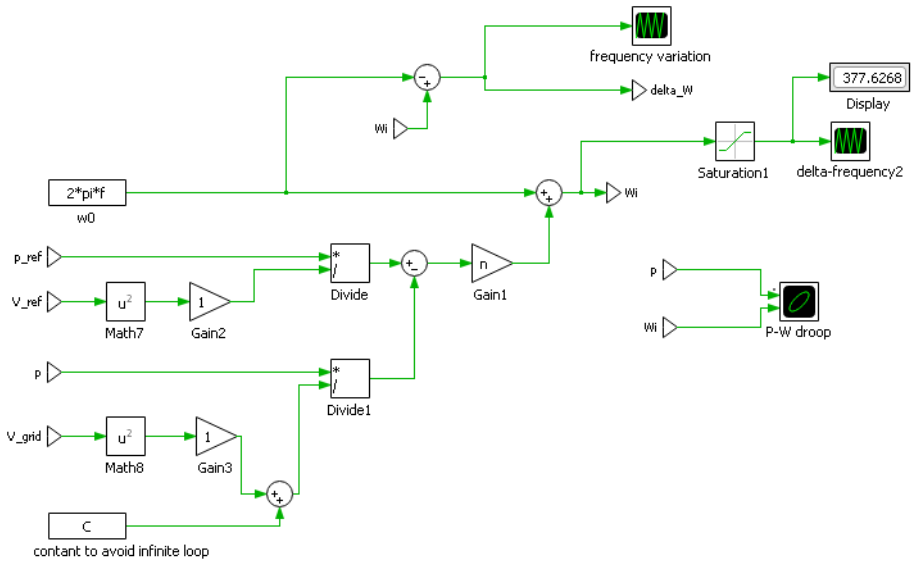


Figure 4.23: *Implementation of frequency droop using equation 4.12*

where constant C has a value "1" and is used to avoid divergence of the simulation. Another frequency droop characteristic is implemented with equation 4.13 using the assumption that $\|v\| \approx v^\star$. Equation 4.13 is given as:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \omega_i \approx \ \omega_0 + \eta(\frac{p^\star - p}{v^{\star 2}}) \tag{4.13}$$

The implementation of equation 4.13 is given in figure 4.24. We can see that the droop characteristics becomes linear with the approximation $v = v^\star$ given by equation 4.13.

4.3.9 Droop Contol Q-V Block

Similar to frequency droop characteristics of P-W block, Voltage droop characteristics are implemented in this block, to prove the voltage droop characteristics of dVOC. The voltage droop characteristics are given by equation 3.27 from chapter 3 which is re-

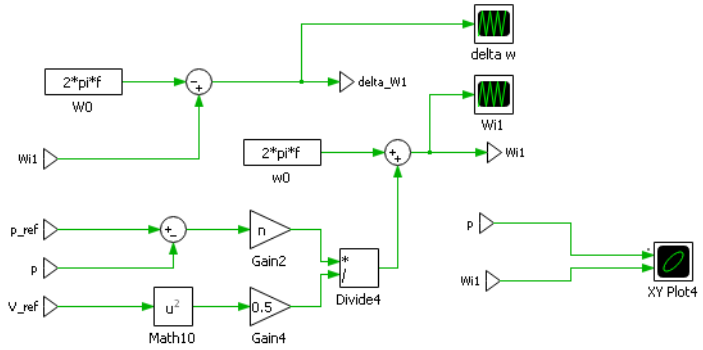


Figure 4.24: *Implementation of frequency droop using equation 4.13*

written as:

$$\frac{\mathrm{d}\|v\|}{\mathrm{d}t} = \eta\left(\frac{q^\star}{v^{\star 2}} - \frac{q}{\|v\|^2}\right)\|v\| + \frac{\eta\alpha}{v^{\star 2}}(v^{\star 2} - \|v\|^2)\|v\| \qquad (4.14)$$

The Implementation of droop characteristics of equation 4.14 are given in figure 4.25.

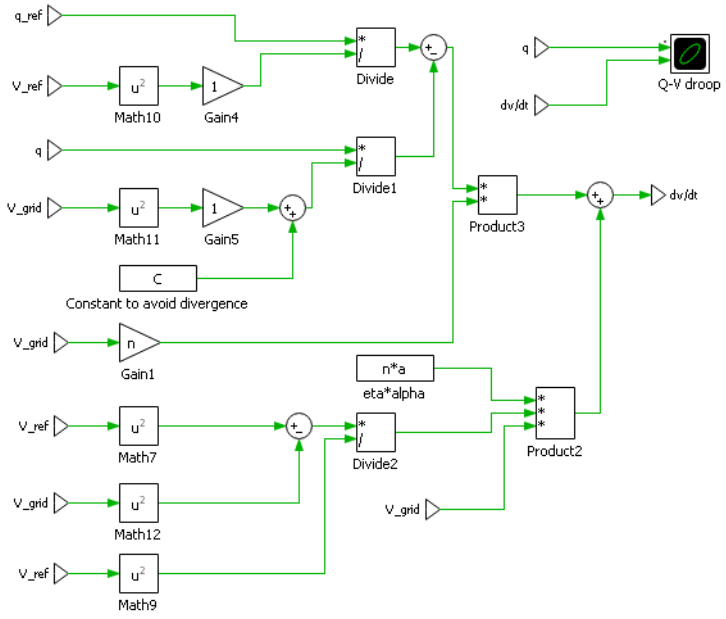


Figure 4.25: *Implementation of Voltage droop using equation 4.14*

where constant C has a value "1" and is used to avoid divergence of the simulation. Another voltage droop characteristic is implemented with equation 4.15 using the assumption that

$\|v\| \approx v^\star$, which makes the droop characteristics linear. Hence at steady state, equation 4.15 is given as:

$$\|v\| \approx v^\star + \frac{1}{\alpha v^\star}(q^\star - q) \tag{4.15}$$

The implementation of equation 4.15 is given in figure 4.26.

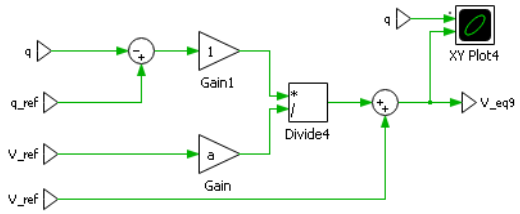


Figure 4.26: *Implementation of Voltage droop using equation 4.15*

4.3.10 Results Block

The Result block is used to accumulate all the scopes to display outputs. Figure 4.27 shows the implementation of this block.

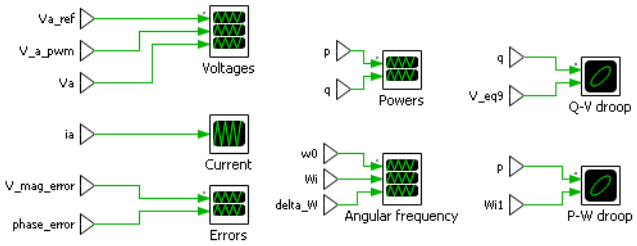


Figure 4.27: *Implementation of Results Block*

4.4 Simulation Results

In this section, we will discuss about the simulated results under different scenarios ranging from increasing power to adding and removing inverter. The effects on stability along with frequency variation and voltage regulation will be discussed.

4.4.1 Case 1: System Stability

In this case, the active power set point $p^\star$, reactive power set point $q^\star$, voltage set point $v^\star$ and nominal frequency f is given in table 4.3.

Table 4.3: *Circuit parameters*

Parameter	Value	Unit	PLECS const.
$p^\star$	500	W	p
$q^\star$	-125	Var	q
$v^\star$	120	$V(rms)$	va_ref
f	60	Hz	f

Reference voltage "va_ref", Control signal "v_a_pwm" as a result of dVOC control law, and grid voltage or output voltage "V_grid" is given in figure 4.28.

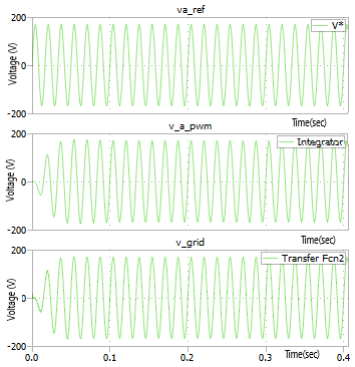


Figure 4.28: *va_ref, v_a_pwm and V_grid voltages*

In figure 4.28, x-axis represents time in seconds while y-axis represents voltage in Volts. It can be seen in the figure 4.28, that output voltage stabilizes and matches very closely to the given reference voltage and frequency within 100ms. The frequency measured is $60Hz$. Figure 4.29 represents magnitude error along with phase error as implemented in v_a_pwm block in figure 4.16. In figure 4.29, x-axis represents time in seconds while y-axis represents voltage in Volts in the first graph where as phase in radians

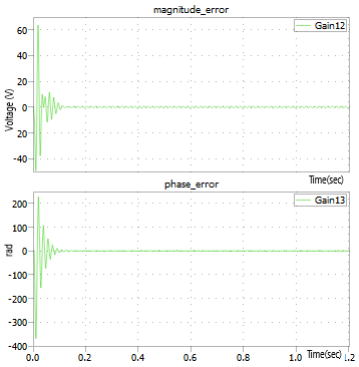


Figure 4.29: *Magnitude error and phase error*

in second graph. It can be seen that the magnitude error and phase error approximate to zero within 100ms as also shown in figure 4.28. The output active power p and reactive power q is shown in figure 4.30.

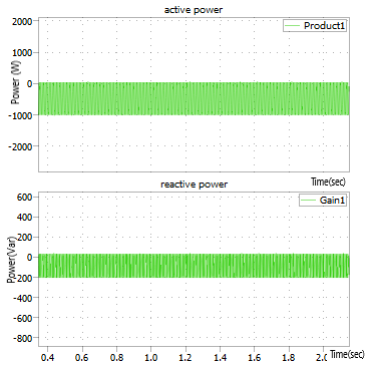


Figure 4.30: *Output instantaneous active and instantaneous reactive power*

In figure 4.30, x-axis represents time in seconds while y-axis represents active power in watt in the first graph where as reactive power in Var in second graph. It can be seen that active and reactive powers are instantaneous. However, their average value shows that they closely match with the reference active power "p_ref" and reference reactive power "q_ref". Transients occur due to integration of inverter with the grid, however the response is very fast. Reference angular frequency ω_0, output angular frequency ω and change in angular frequency $\Delta\omega$ is shown in figure

4.31.

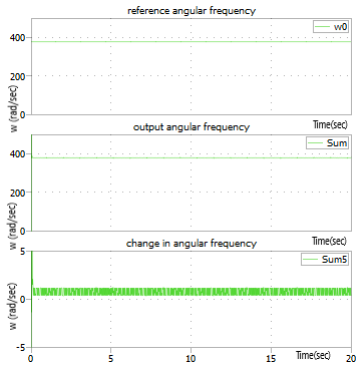


Figure 4.31: *Reference angular frequency ω_0, output angular frequency ω and change in angular frequency $\Delta\omega$*

In figure 4.31, x-axis represents time in seconds while y-axis represents frequency in Hz. We know that for a frequency of $60Hz$, the angular frequency is $377rad/sec$. It can be observed that the output frequency is very close to the reference and differs only by "$2rad/sec$" which is acceptable being in the limits of frequency deviation.

The frequency droop characteristics are shown in figure 4.32. In figure 4.32, x-axis represents active power in watts while y-axis

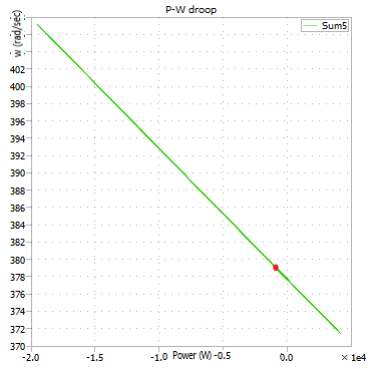


Figure 4.32: *$P - \omega$ droop characteristics*

represents angular frequency in rad/sec. The droop characteristics of equation 4.13 are represented in this figure. We can see that the characteristics are linear and the operating point with active power

496 W and angular frequency 379 rad/sec "(496 W,379 rad/sec)", represented by red dot is shown on the graph. Hence we can say that dVOC follows the frequency droop control.

Similar to frequency droop, Voltage droop characteristics are shown in figure 4.33.

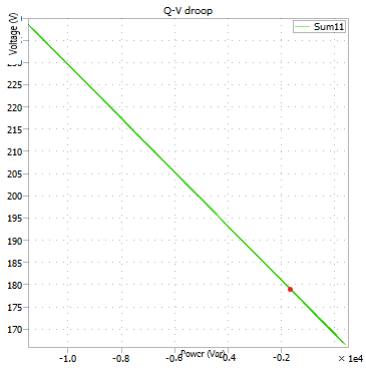


Figure 4.33: *$Q - V$ droop characteristics*

In figure 4.33, x-axis represents reactive power in var while y-axis represents voltage in Volts. The Voltage droop characteristics of equation 4.15 are represented in this figure. We can see that the characteristics are linear and the desired operating point is shown on the graph. Keep in mind that the voltage value on y-axis is peak to peak value and not rms. Figure 4.33 proves that dVOC follows voltage droop control.

4.4.2 Case 2: Adding an Inverter

As we have seen in the previous case that the inverter is following the voltage, frequency and power set points and stabilizing in less than 100ms. Also it meets the frequency and voltage droop characteristics as shown in figure 4.32 and 4.33 respectively.

In this case, we will observe the effects of adding another inverter on voltage and current. We consider two inverters with the same voltage, power and frequency set points shown in figure 4.34. It can be seen that two inverters with same characteristics are connected to the grid. Schematic of inverter A is shown in

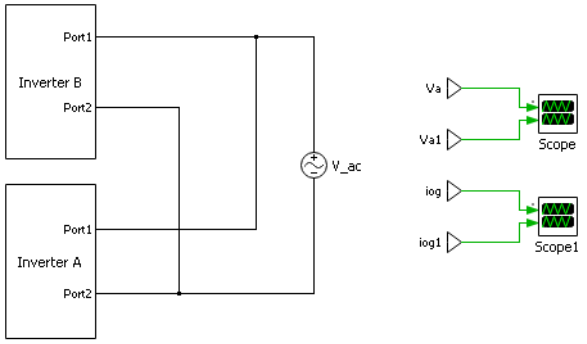


Figure 4.34: *Two inverters connected with the grid*

figure 4.35 This inverter in figure 4.35 is the same inverter con-

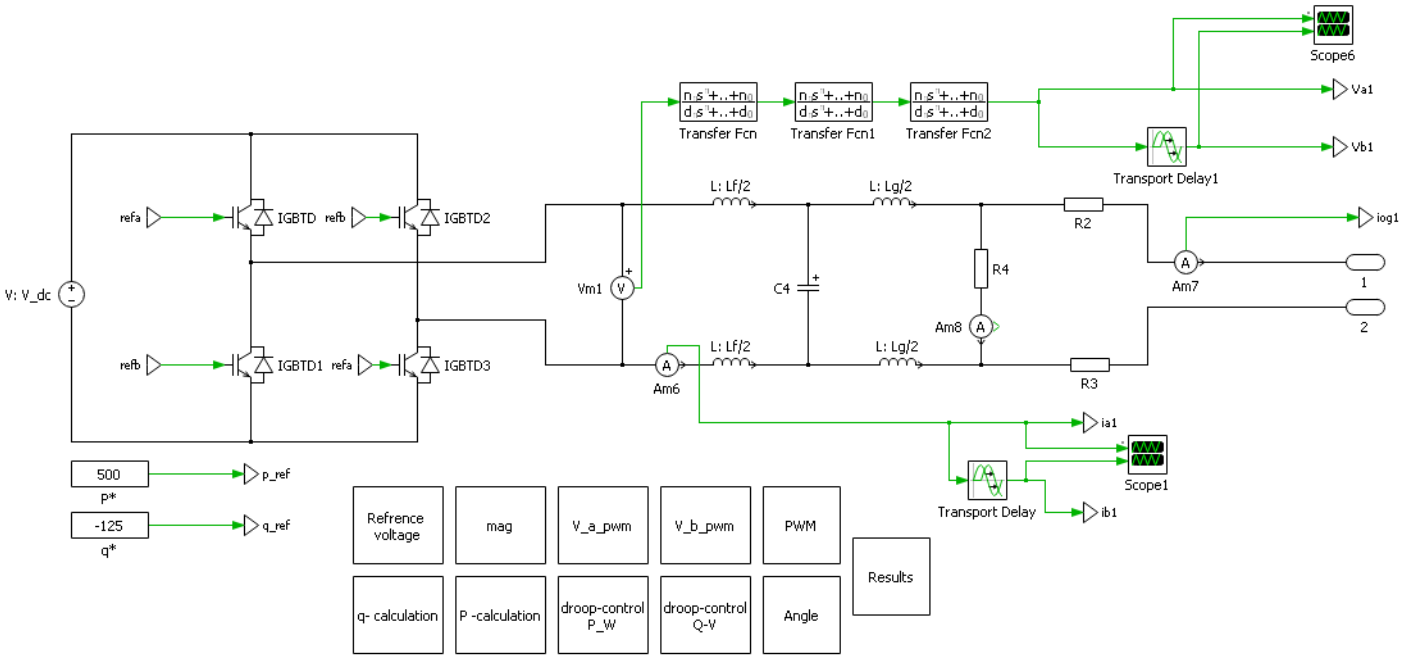


Figure 4.35: *Schematics of inverter A*

nected with the grid as defined in case 1. The output voltage is represented by "va1"and the current injected into the grid is denoted by "iog1". Schematics of inverter B (newly added inverter) is shown in the figure 4.36.

The output grid voltage is denoted by "va" whereas the current injection to the grid or current absorption by the grid is denoted

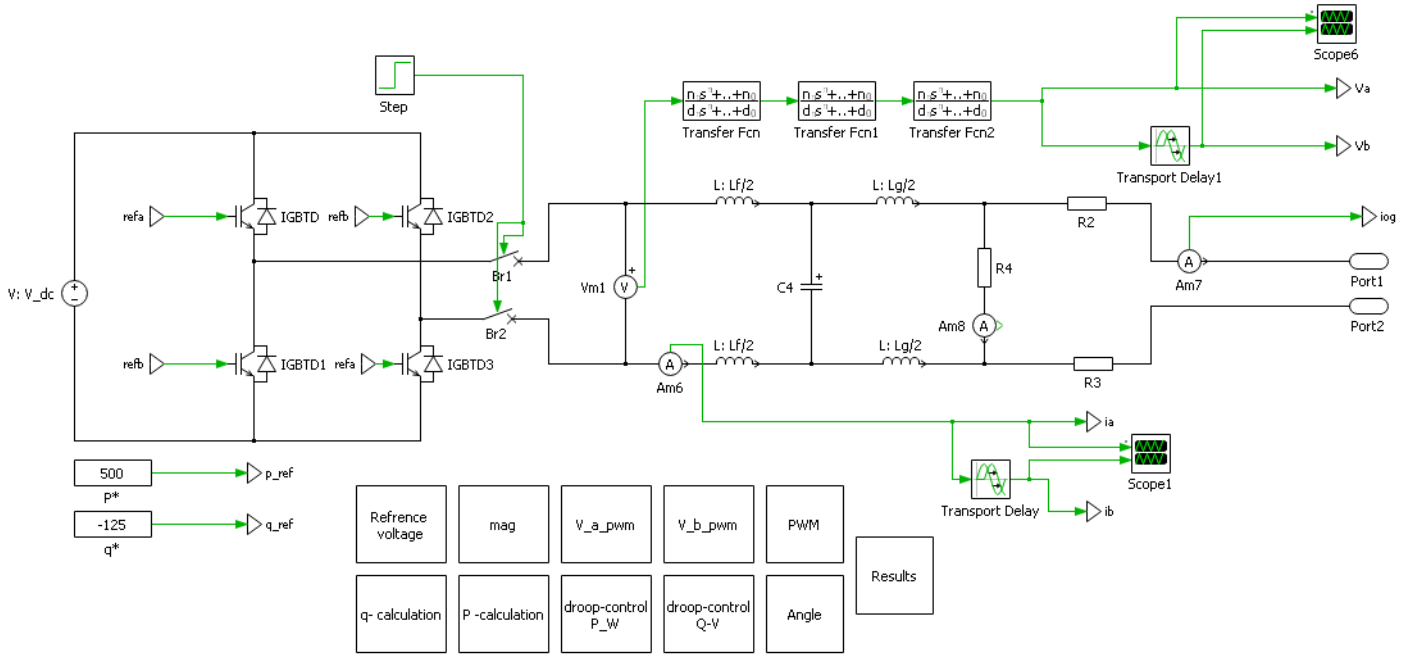


Figure 4.36: *Schematics of inverter B*

by "iog". It can be seen that the step block is used to operate the switches of the inverter. The new inverter (inverter B) is integrated into the grid at time instant "2". The voltages of the grid connected with the inverter are shown in figure 4.37.

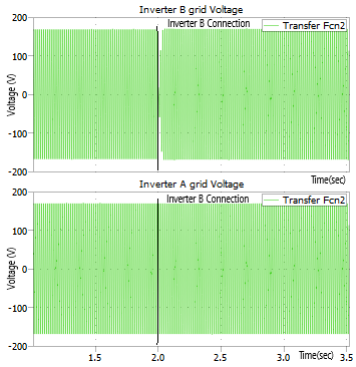


Figure 4.37: *Grid voltage with and without inverter integration*

In figure 4.37, x-axis represents time in seconds while y-axis represents voltage in Volts. It can be seen that inverter A is feeding the load and connected with the grid. The grid voltage fed by the

inverter is having the set point voltage and frequency. The load "R4", in case of inverter B, is fed by the grid and hence the voltage observed before time instant "2" is the voltage of the grid.

At time instant "2", inverter B is connected to the load and grid and we can see that it takes less than 100ms for the output voltage of this inverter to reach the desired voltage and frequency set points, and hence synchronize with the grid. It is proved, that the inverter dynamics and response is fast for grid integration.

The current fed or absorbed by the grid called the circulating current is shown in figure 4.38.

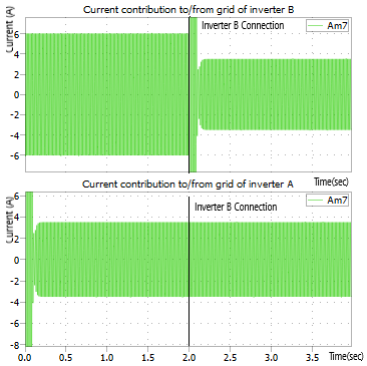


Figure 4.38: *Circulating current and power balance with and without inverter integration*

In figure 4.38, x-axis represents time in seconds while y-axis represents current in Ampere. It can be observed that before time instant "2", inverter A is feeding load and the grid. The grid is feeding the local load connected with inverter B, and hence the measured current "iog" is flowing opposite to the ammeter connection. The load "R4" required approximately $5.9A$ current which can be seen in current contribution of inverter B before inverter integration.

When inverter B is added to the system at time instant "2", the circulating current of both inverters becomes same, i.e., approximately $3.5A$. This proves that the inverters share equal power to the grid, based on their power set points.

4.4.3 Case 3: Adding an Inverter without grid

This case is exactly similar to case 2, except the system is not connected with the grid or the inverters are operating in islanding mode. The schematic diagram of this case is shown in figure 4.39. The measured voltages "va" and "va1", with and without inverter

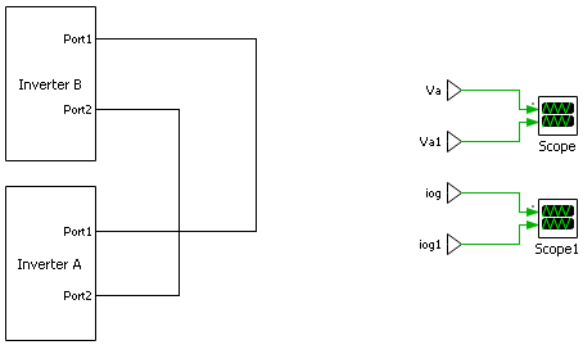


Figure 4.39: *Adding an inverter to a system without grid*

integration are shown in figure 4.40.

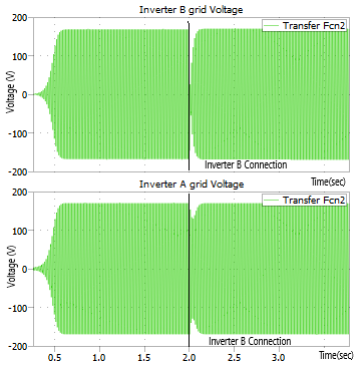


Figure 4.40: *Measured grid voltages with and without inverter integration*

In figure 4.40, x-axis represents time in seconds while y-axis represents voltage in Volts. It can be seen that the output voltage of inverter A becomes a source for local load connected with inverter B before time instant "2". At time instant "2", inverter B is integrated in the system and output voltage of inverter B is synchronized with the output voltage of inverter A in less than 100ms. The integration of inverter B, also causes fluctuation in inverter A

voltage as seen in figure 4.40. The circulating current, or current contribution to the grid with and without inverter integration is shown in figure 4.41.

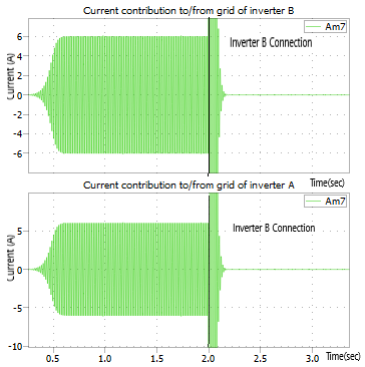


Figure 4.41: *Current contribution to the system, with and without inverter*

In figure 4.41, x-axis represents time in seconds while y-axis represents current in Ampere. It can be seen that the circulating current is flowing from inverter A, to feed the load of inverter B before time instant "2". It can be seen, that both circulating currents are same in value as current from inverter A is injected and the same current is absorbed by load of inverter B. After time instant "2", there is no circulating current as both of the inverters are feeding their local loads. This case proves that the inverters implemented by dVOC law are capable to operate in islanding.

4.4.4 Case 4: Removing an Inverter from the grid

In this case, we will observe the effects of removing an inverter on output voltage and circulating current(current contribution to/from the grid from an inverter). The implementation is similar to case3, where two inverters connected with the grid are shown in figure 4.34. Schematics of inverter A are shown in figure 4.35, while schematics of inverter B are depicted in figure 4.36. The inverter to be removed is "inverter B", which is removed at time instant "2". A step block is used to operate the switch. The inverter voltage to the grid is shown in figure 4.42.

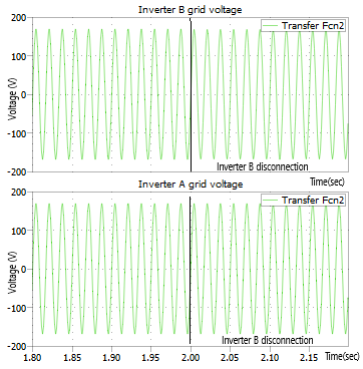


Figure 4.42: *Inverter voltage to the grid, with and without inverter*

In figure 4.42, x-axis represents time in seconds while y-axis represents voltage in Volts. It can be seen that before time instant "2", inverter A and inverter B, both are feeding the loads and the grid. We can see that both voltages synchronize with each other and the grid and the voltage set point is reached. However, after time instant "2", inverter B is disconnected and the load of inverter B is fed by the grid. Since, inverter B was synchronized with the grid before time instant "2", disconnection does not affect the grid voltage. The circulating current, or current contribution to the grid with and without inverter disconnection is shown in figure 4.43.

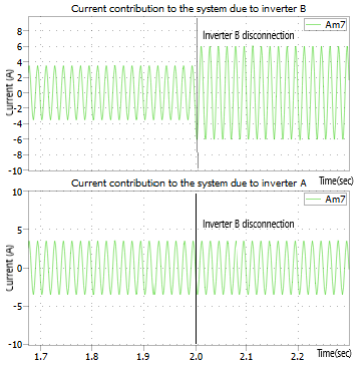


Figure 4.43: *Circulating current and power balance, with and without inverter disconnection*

In figure 4.43, x-axis represents time in seconds while y-axis represents current in Ampere. It can be seen that before time instant "2" or before disconnection of inverter B, the current shar-

ing towards the grid is same and hence equal power sharing is observed from both inverters. However, after disconnection of inverter B from the grid, the current contribution from inverter A remains the same, while the grid injects current to feed local load "R4" of inverter B. Hence it can be seen that a loss of inverter from the grid still stabilizes the system satisfying power set points.

4.4.5 Case 5: Removing an Inverter without the grid

This case is exactly similar to case 4, except the system is not connected with the grid or the inverters are operating in islanding mode. The schematic diagram of this case is shown in figure 4.39. The measured grid voltages "va" and "va1", with and without inverter disconnection are shown in figure 4.44.

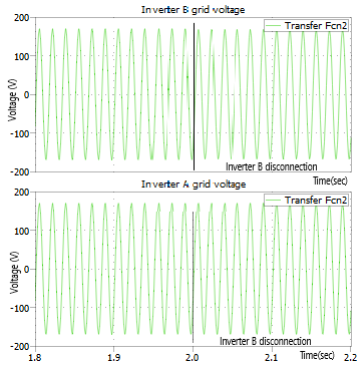


Figure 4.44: *Measured grid voltages with and without inverter disconnection*

In figure 4.44, x-axis represents time in seconds while y-axis represents voltage in Volts. We can see that before disconnection of inverter B, i.e. before time instant "2", grid voltages from inverter A and inverter B are synchronized and satisfy voltage set points. However, after disconnection of inverter B, i.e. after time instant "2", inverter A is feeding the local load of inverter B and hence the voltage seen by the grid of inverter B is same as voltage of inverter A. We can also see, that disconnection does not show us any dip in voltage.

The circulating current, or current contribution towards the system, with and without inverter disconnection is shown in figure 4.45.

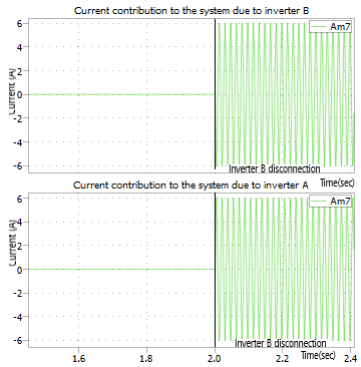


Figure 4.45: *Current contribution to the system, with and without inverter disconnection*

In figure 4.45, x-axis represents time in seconds while y-axis represents current in Ampere. It can be observed in figure 4.45 that, before disconnection of inverter B, there is no current flowing from inverter A to inverter B or vice versa. This shows that the inverters only feed their loads. However, after disconnection at time instant "2", inverter A feeds the local load of inverter B. We can see that the measured current "iog" is opposite to that of "iog1" which is "inverter A" current contribution to the system.

This case proves, that the inverters implemented by dVOC law are capable to operate in islanding mode, even with a loss of inverter.

In the next chapter, we will discuss about Hardware in the Loop implementation for control of our inverter.

CHAPTER 5

Implementation of Control Algorithm using Hardware in the Loop

5.1 Introduction

In the previous chapter, we have simulated the dVOC algorithm for two inverter system. It was evident from the results, that dVOC technique synchronizes the inverter output voltage along with the possibility to operate the system in islanding mode.

In this chapter, we will introduce the concept of hardware in the loop (HIL), its importance and implementation of dVOC for an inverter using HIL. Also, we will devise a technique to introduce virtual inertia in the system.

5.1.1 Hardware in the Loop (HIL)

HIL is a testing technique where real signals from a controller are connected to a test system that simulates reality, tricking the controller into thinking it is in the assembled product [34]. Test

and design iteration take place as though the real-world system is being used. You can easily run through thousands of possible scenarios to properly exercise your controller without the cost and time associated with actual physical tests. A description of a Hardware in the Loop system is shown in figure 5.1.

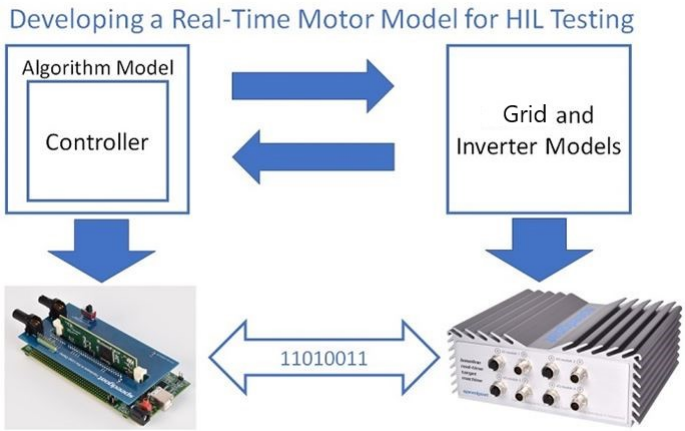


Figure 5.1: *General idea behind Hardware in the loop system [35]*

We can see that a physical controller (generally a DSP) implements the control algorithm and interacts with a physical hardware module which simulates the model of a grid and inverter. The signals generated by HIL system mimic the same response as expected from the real physical system and the controller responds as if connected with a real physical system (Grid and inverter). A detailed depiction of HIL simulation is shown in figure 5.2 In figure 5.2, Embedded Control System represents a Physical controller which interacts with HIL simulator using a physical interface. The Embedded control system process data of virtual sensors and run a control algorithm. The output is then fed to HIL simulator using a physical interface which will interact with the system.

In our implementation, Embedded Control system used is a DSP controller from Texas Instruments TI320F28069, the physical interface connecting the HIL with the controller is the wires,

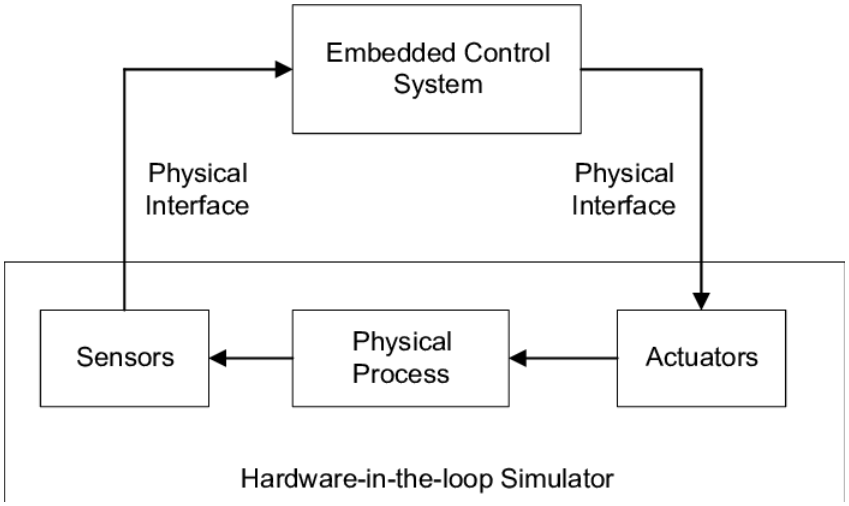


Figure 5.2: *Block diagram of a HIL simulation [36]*

The HIL simulation is run by "Typhoon HIL" which incorporates the model of the system.

Benefits of HIL Simulation: Cost saving and risk reduction are the major benefits of HIL simulations. The benefits are explained here:

- Reduce costs by increasing the speed and scope of continuous verification and validation.
- Perform tests beyond the range of normal parameters or plant capabilities without risking damage to equipment.
- Lower innovation costs by being able to continuously try new ideas, even when the actual plant is not available.
- Unlike physical plants, real-time simulators can easily be expanded with new I/O at any time.

5.2 Components Used in our Simulation

In this section, we will discuss about the components used in our HIL simulation along with their parameters. Figure 5.3 represents

the block diagram of HIL simulation implemented in this thesis. The Implementation consists of two parts, Physical controller and

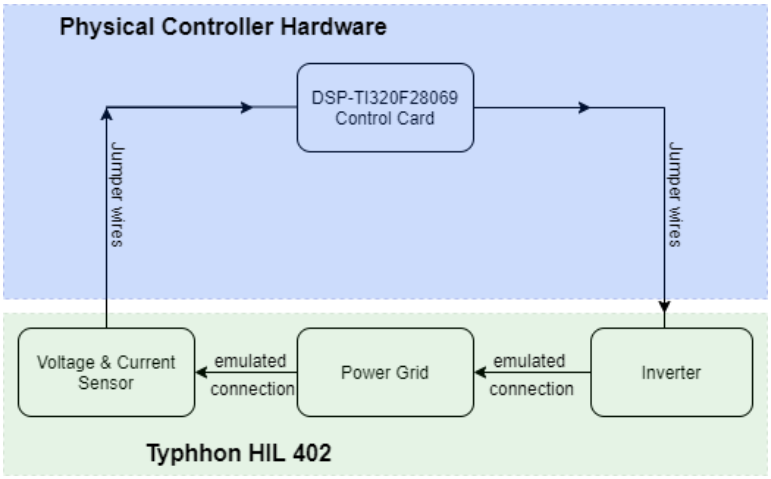


Figure 5.3: *Block diagram of process and components used in implementation of HIL simulation*

Software Simulation. HIL software is used to provide an interface between software Emulation and hardware controller.

5.2.1 Physical Controller

The Physical controller used in this implementation is Texas Instruments TMS320F28069 Control card. It belongs to family of C2000-32 bit microcontrollers. C2000 32-bit microcontrollers are optimized for processing, sensing, and actuation to improve closed-loop performance in real-time control applications such as industrial motor drives; solar inverters and sensing and signal processing [37].

The Analog to Digital converter(ADC) converts from 0 to 3.3-V fixed full-scale range and supports ratio-metric $\frac{VREFHI}{VREFLO}$ references. The ADC interface has been optimized for low overhead and latency. Figure 5.4 represents a control card along with Docking station. The Docking Station provides power and JTAG debug for the controller Card. Key device signals are accessible through header pins. dVOC control algorithm is implemented in this DSP

Figure 5.4: *Controller Card along with docking station*

micro-controller which is interfaced with Hardware in the loop simulation. Some Specifications of this micro-controller which are of interest are given as:

- High-efficiency 32-bit CPU with frequency of 90 MHz (11.11-ns cycle time).
- Floating-Point Unit (FPU) to deal quickly with decimal numbers.
- JTAG support for debugging purpose.
- Peripheral Interrupt Expansion (PIE) block that supports all peripheral interrupts.
- Three 32-bit CPU timers along with 16 PWM channels.
- 12-bit Analog-to-Digital Converter (ADC) with 16 channels and maximum sampling rate of 3.46 mega samples per second(MSPS).
- Up to 54 individually programmable, multiplexed General-Purpose Input/Output (GPIO) pins with input filtering

The Integrated Development Environment(IDE) or software used to program and debug this micro-controller is Code Composer Studio(CCS).

5.2.2 HIL Software

For HIL testing to be of value, the quality of the simulation software is of utmost importance. Simulation software must be paired

with hardware that not only accounts for system specifications such as connector type and I/O but also allows for fault insertion and the ability to test real-world scenarios. In our simulation, we used Typhoon HIL 402 to simulate our system and integrate with the micro-controller.

The HIL402 system provides a safe, high-fidelity environment for automatic test and verification of inverter based control systems. Figure 5.5 represents a complete HIL environment along with Typhoon HIL software.

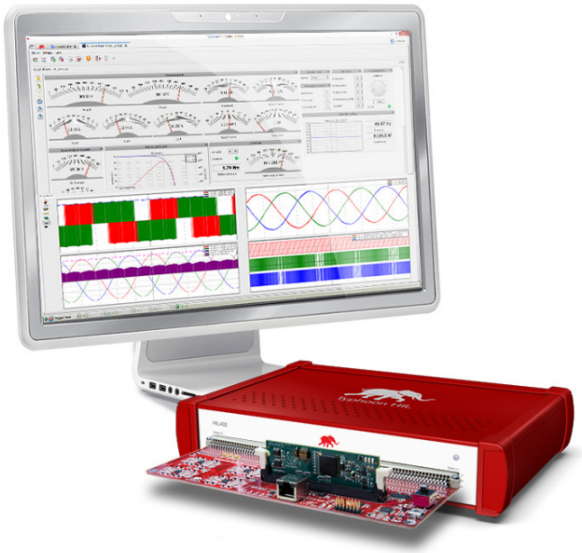


Figure 5.5: *A complete Hardware in the loop environment along with HIL software [38]*

Some specifications of interest of the Typhoon HIL are given below:

- 20 ns PWM sampling time for the signals from control card.
- Comprehensive model library of components to be used in the implementation of system.
- Built-in Scope and Capture to observe signals.
- 16 analog and 32 digital channels with 16 bit resolution to interact with the physical controller.

- Custom Graphical User Interface like SCADA to add gauges and meters to control parameters and monitor the output respectively.
- Advanced machine solver.

The control card implements the algorithm and interacts with the simulated model to test the response of the feedback system. An example of HIL implementation using typhoon HIL 402 and control card is shown in figure 5.6.

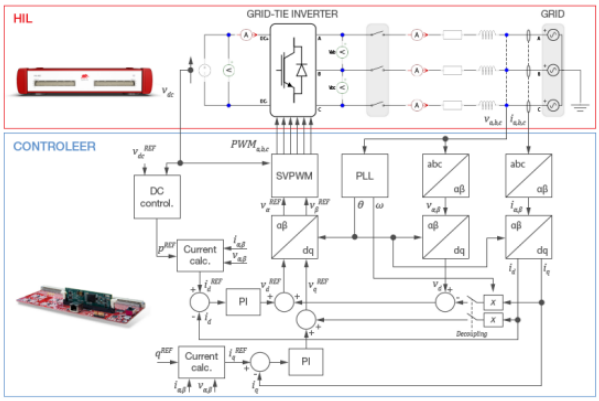


Figure 5.6: *An example of Typhoon HIL 402 implementation using control card 320F28069 [38]*

It can be seen, that the grid and inverter is implemented in HIL IDE , whereas the control algorithm to control inverter is implemented in 320F28069 controller. The HIL software is capable to scale down the voltage levels to be compatible with the controller voltage levels. Hence the actuators, sensors, voltage sources, grid and inverter are implemented in the HIL simulator while only the control algorithm is implemented in the controller 320F28069.

5.3 General testing using HIL

In this section, we will describe the initial testing procedures performed to understand the behavior of HIL. Instruments used in this validation are:

- RIGOL DS1104 Ultra Vision 4 Channel, 100MHz Oscilloscope with 1 Giga Samples Per Second(GSPS).
- SIGLENT SDG1025 Function waveform Generator with 125 Mega Samples Per Second(MSPS).
- Typhoon HIL 402.
- Control Card 320F28069 by Texas Instruments.

5.3.1 Latency test

Latency is one of the many problems that affect the performance of our system. As an example, let us consider an output signal generated by the controller to interact with the simulated system. We know that the simulation is running at a certain execution rate and their is a time delay between the injection of control signal and reception of the simulation. The idea is represented in figure 5.7.

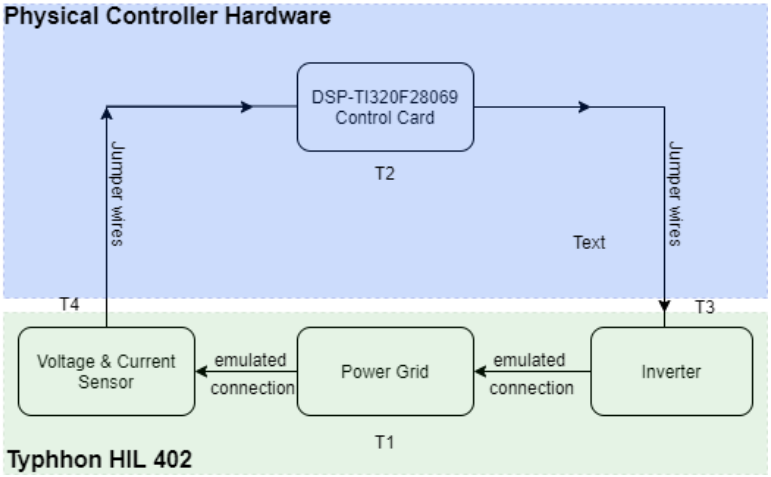


Figure 5.7: *Problem of latency*

It can be seen in figure 5.7, T1 represents the time taken by simulation, T2 represents the time taken by the Digital Signal Processor(DSP) to perform the algorithm, while T3 and T4 represents the latency between the two systems (DSP and Simulation). This

time delay is undesirable and can cause disturbances in the testing performance. Hence, it is important to measure the latency to be sure of the performance of our controller and the system. The approach used is given in figure 5.8.

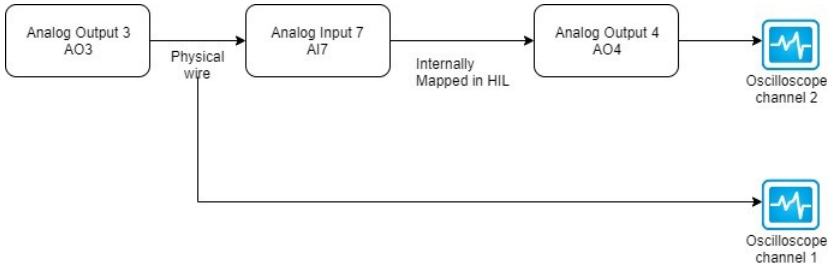


Figure 5.8: *Approach to measure the latency of our controller and simulated system*

In this approach we use the output from HIL(A03) to use it as an input for the HIL like a feedback. We will map this output coming from AO3 to analog input (AI7) through a physical wire. The input (AI7) is then internally mapped to analog output(A04) by HIL software which is then connected to RIGOL oscilloscope to measure the signal. Input signal block, modeled in HIL software along with generated signal on pin A03 is given in figure 5.9 and 5.10 respectively.

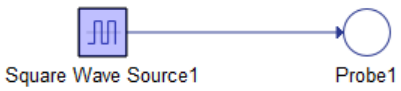


Figure 5.9: *Input signal block in HIL schematic editor*

The resulting output (A04) and input (A03) on channel one and channel two of oscilloscope with latency is shown in figure 5.11. It can be seen, that both signals of same amplitude, duty cycle and frequency only differ in time by 4 Micro-seconds. The same approach is used for real signals generated by Signal generator (Diglent SDG1025) is shown in figure 5.12.

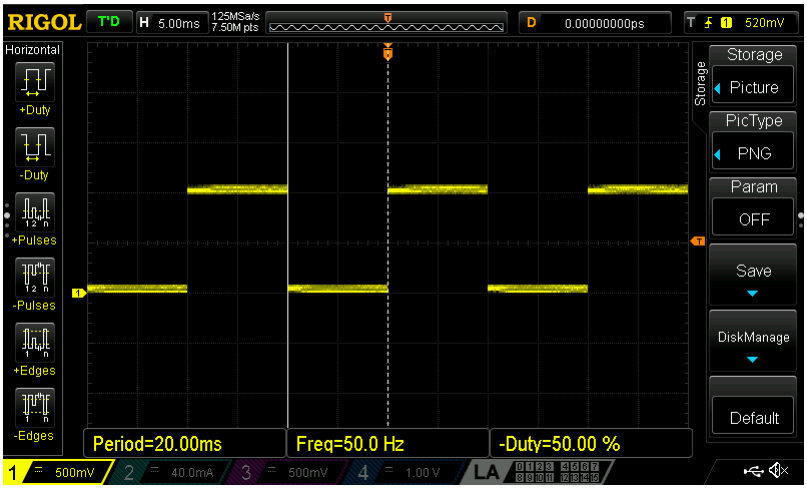


Figure 5.10: *Signal measured on output pin(A03) of HIL with oscilloscope*

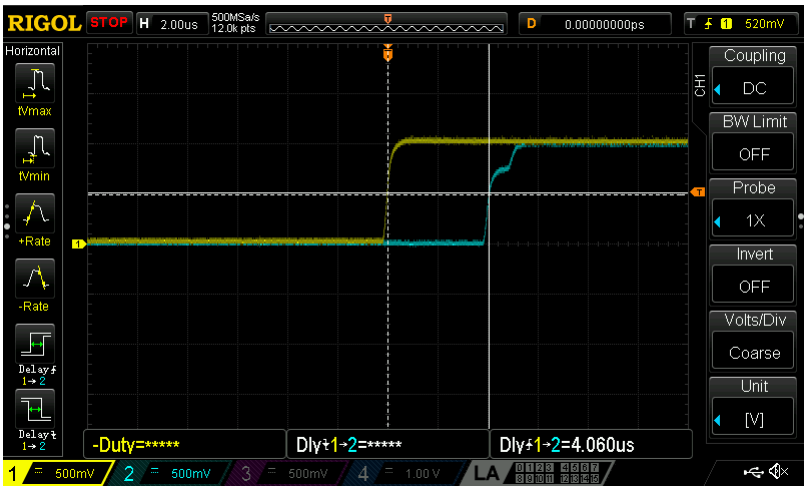


Figure 5.11: *Resulting output (A04) and input (A03) with latency information*

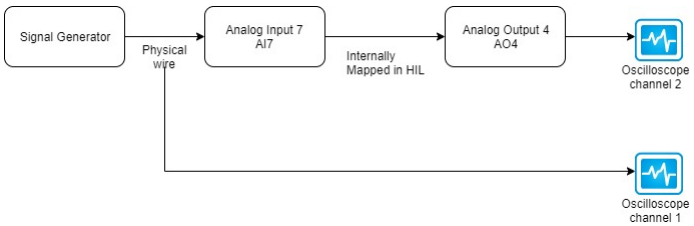


Figure 5.12: *An approach to calculate latency using real signals from signal generator(Diglent SDG1025)*

The input and output signals as measured by channel 1 and

channel 2 of oscilloscope are given in figure 5.13

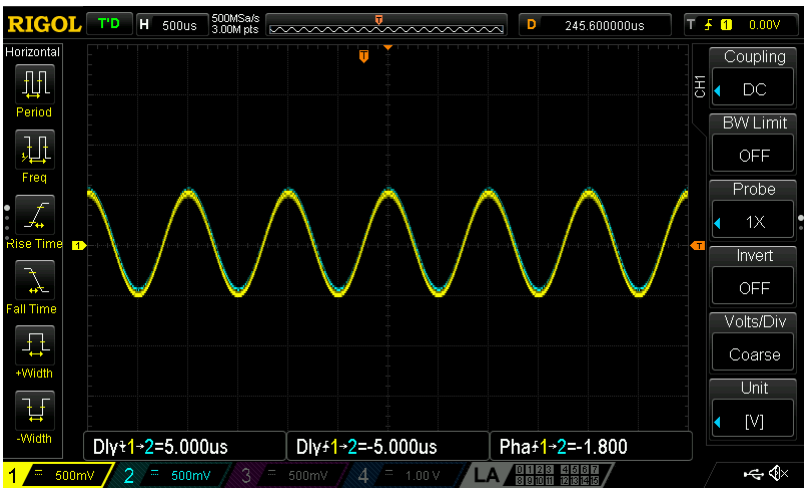


Figure 5.13: *Resulting output (A04) and input real signal with latency information at 1kHz frequency*

The signal injected in 5.13, is a sinusoidal signal with frequency of "1kHz". The delay is not significant in this case. The same test is performed on a signal with frequency 30kHz. The output is shown in figure 5.14.

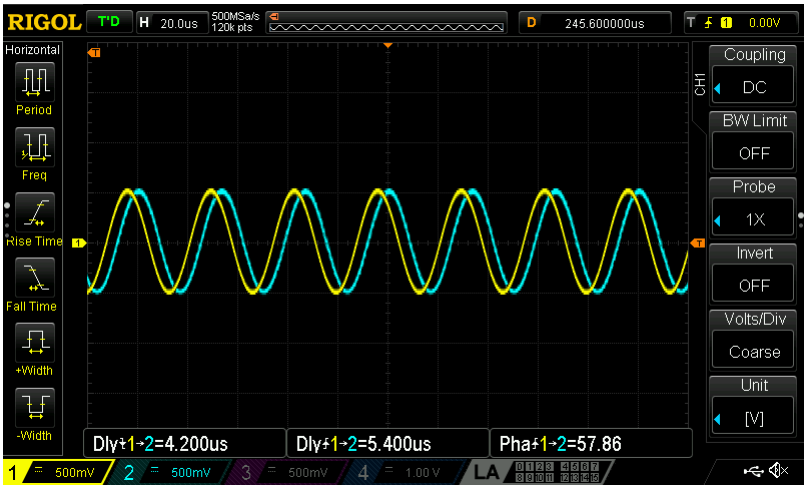


Figure 5.14: *Resulting output (A04) and input real signal with latency information at 30 kHz frequency*

We can see that the delay between the two waveforms in approximately $5\mu s$. The delay is not significant when the frequency is

low. However, when the frequency is high, the delay is more prominent, as the input is first sampled and then produced at the output. Hence we have to take it into consideration while performing the switching operations, that the injected input has a delay of almost $4 - 5\mu s$ to interact with model configuration.

5.3.2 Single Phase Inverter test

In this test, we try to familiarize ourselves with the inverter model and the possibilities to control it. In HIL simulation, the inverter can be controlled either by digital signals to the leg of inverter, switches of inverter or internal modulation. In this testing, we implement control of inverter, using digital input to the leg of inverter.

We are using a very basic single phase inverter whose output is measured. Load is resistive "1Ω" and DC source voltage is $10V$ for simplicity. We injected a PWM of 50Hz with 50% duty cycle to inverter leg 1 and the inverted PWM to the inverter leg 2. The model is shown in figure 5.15.

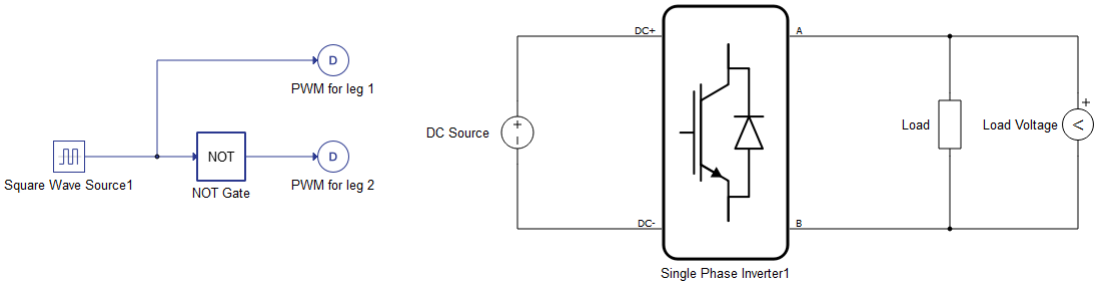


Figure 5.15: *Single phase inverter model in HIL simulation*

The approach used to inject input to the inverter legs in HIL is shown in figure 5.16.

The PWM signals generated for leg 1 and leg 2 of inverter are diverted to the digital pins 1 and 2 by software and are shown in figure 5.17. The output of the inverter is shown in figure 5.18.

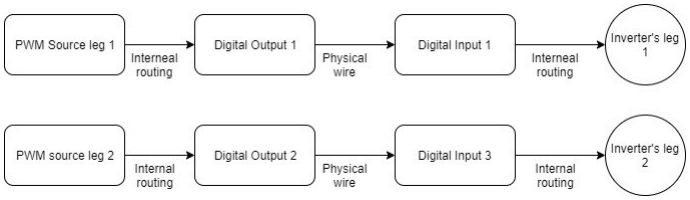


Figure 5.16: *Approach used to inject input signals to inverter legs*

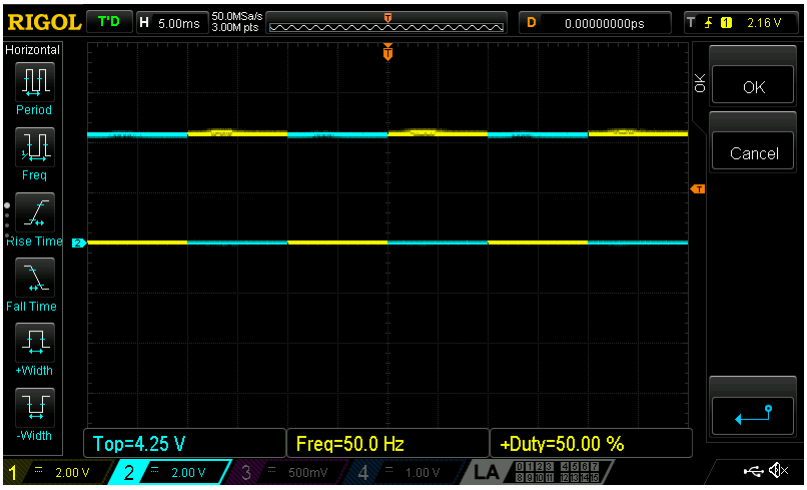


Figure 5.17: *Input signals to leg 1 and leg 2 of inverter in HIL simulation*

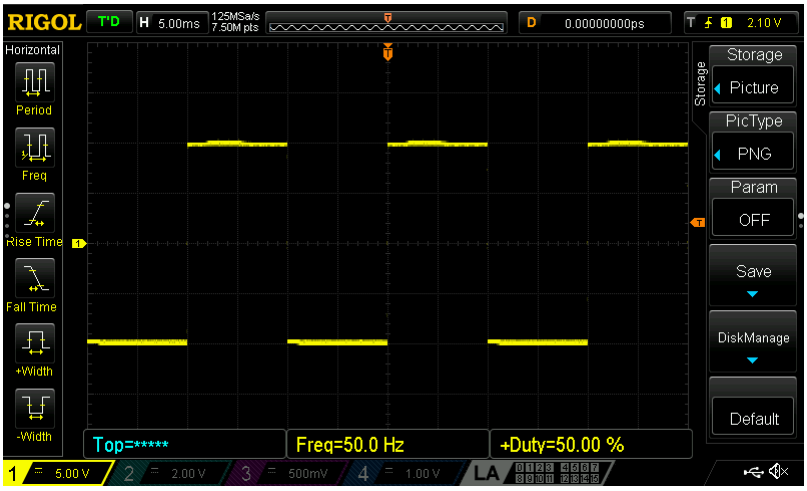


Figure 5.18: *Output of inverter in HIL simulation*

5.3.3 Inverter test using python

The inverter can also be controlled using script in python embedded in HIL. In this approach, we will use the inverter and all the individual switches are controlled by software using python. The output is observed both at SCADA Scope and at pin 1 of Analog Output (AO1) using original oscilloscope. A dead-time of $1ms$ is added for simplicity and the python code is shown in figure 5.19. The output of this inverter in SCADA, controlled by switching

```
# HIL API is imported as 'hil'
import time

hil.set_pe_switching_block_control_mode(blockName='Single Phase Inverter1', switchName='Sa_bot', swControl=True)
hil.set_pe_switching_block_control_mode(blockName='Single Phase Inverter1', switchName='Sa_top', swControl=True)
hil.set_pe_switching_block_control_mode(blockName='Single Phase Inverter1', switchName='Sb_top', swControl=True)
hil.set_pe_switching_block_control_mode(blockName='Single Phase Inverter1', switchName='Sb_bot', swControl=True)


hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sa_bot', value=0)

hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sa_top', value=0)

hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sb_bot', value=0)

hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sb_top', value=0)

time.sleep(1/1000)

i = 1
while i <=10:


    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sa_top', value=1)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sb_bot', value=1)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sb_top', value=0)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sa_bot', value=0)

    time.sleep(10/1000)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sa_bot', value=0)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sa_top', value=0)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sb_bot', value=0)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sb_top', value=0)

    time.sleep(1/1000)

    hil.set_pe_switching_block_software_value(blockName='Single Phase Inverter1', switchName='Sb_top', value=1)

```

Figure 5.19: *Python code for control of inverter in HIL*

all four switches of inverter using python code is shown in figure 5.20, where as the output observed on physical oscilloscope is shown in figure 5.21.

It can be observed from figure 5.20 and figure 5.21, that the SCADA output and the physical output are similar in nature and hence they are inter-operable. This testing proves that the HIL simulation provides nearly similar testing in a physical environment. The next section explains the dVOC testing in HIL.

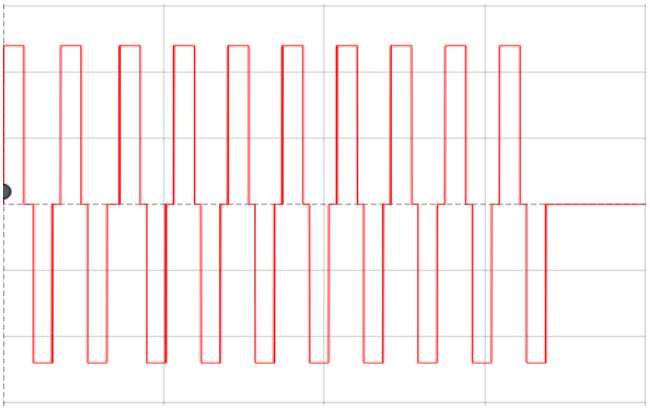


Figure 5.20: *SCADA output of inverter, controlled by python code in HIL*

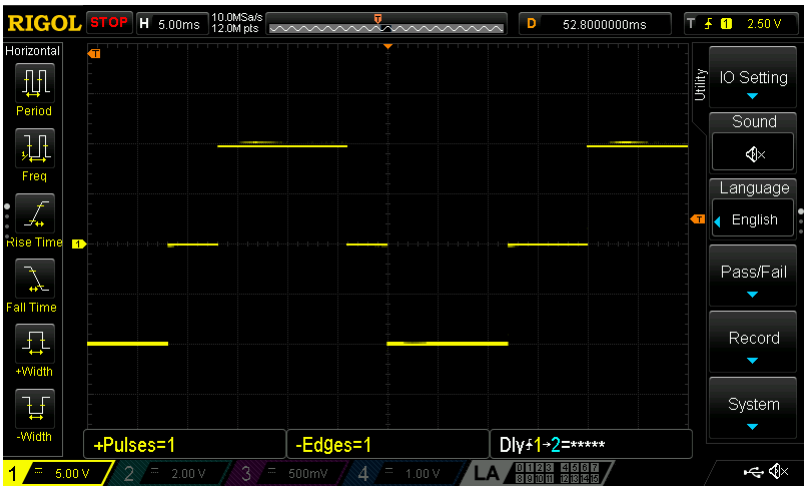


Figure 5.21: *Physical output of inverter, observed using RIGOL oscilloscope*

5.4 dVOC testing and virtual Inertia approach

This section explains the challenges in dVOC testing and the concept of virtual inertia injection in the system using HIL. The approach in this case is similar to PLECS simulation, however, the dVOC control algorithm is implemented in physical **DSP controller** which will interact with the inverter and grid model in HIL simulation. We will also introduce the virtual inertia approach in this section. The grid model along with filter and inverter is shown in figure 5.22. The inverter modeled in figure 5.22, has 4 switches that are individually controlled by the voltage control

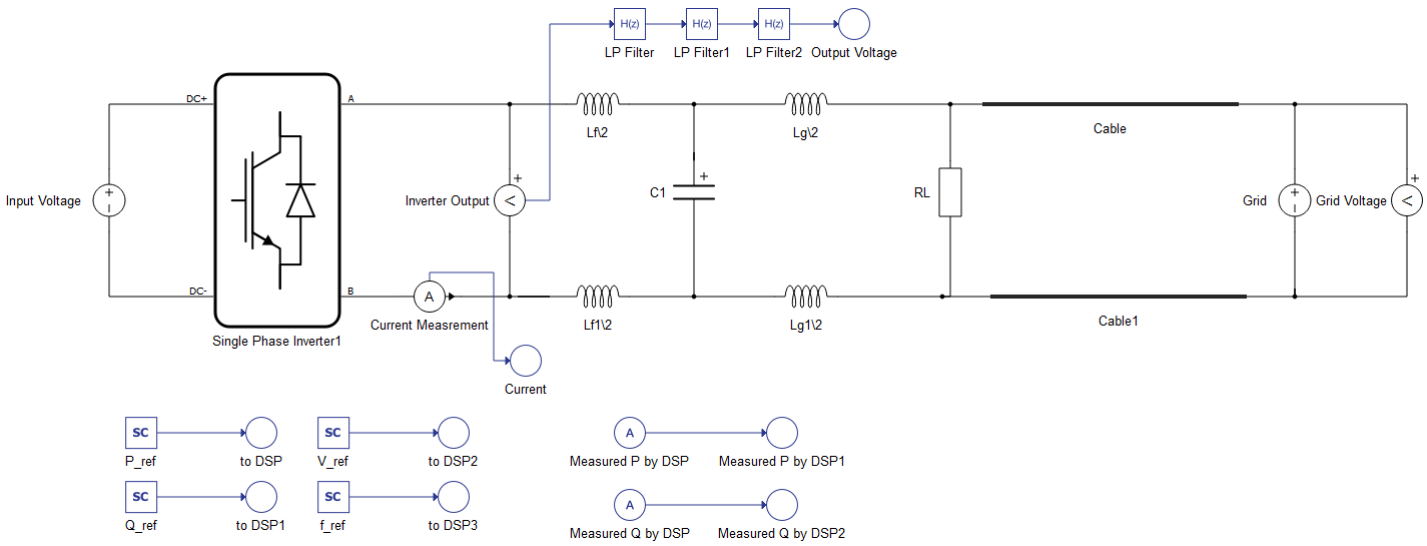


Figure 5.22: *Inverter and Grid model in HIL simulator*

signal of dVOC, similar to PLECS simulation in Chapter 4. Input voltage to inverter is a variable quantity in figure 5.22, which can be controlled in our SCADA model as shown in figure 5.24. Inverter output voltage is measured, filtered out of harmonics and then sent to the SCADA panel where it is displayed on an internal scope. Similarly the grid voltage and current are measured and sent to the SCADA model. The filter parameters are similar to table 4.2 given in Chapter 4. The values for components modeled in figure 5.22 are given in table 5.1:

Table 5.1: *Model Parameters of figure 5.22*

Model Parameter	Value	Unit
Lf	1	mH
$Lf1$	1	mH
Lg	0.2	mH
$Lg1$	0.2	mH
$C1$	24	μF
RL	28.6	Ω
fc	2500	Hz
$Grid$	120	V_{rms}

In table 5.1, f_c denotes the cut off frequency for the filter represented by "LP Filter". The transfer function for "LP Filter", "LP Filter1" and "LP Filter2" is given in figure 5.23.

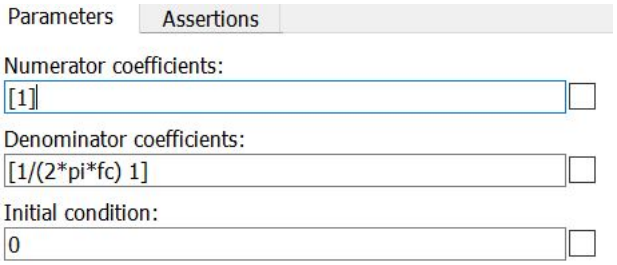


Figure 5.23: *Filter parameters for LP Filter*

The cable represented in figure 5.22 has a resistance of 0.157Ω. The data is obtained from data-sheet for transmission lines [28]. The SCADA model for HIL simulation of our model is shown in figure 5.24. The SCADA model provides a detailed graphical

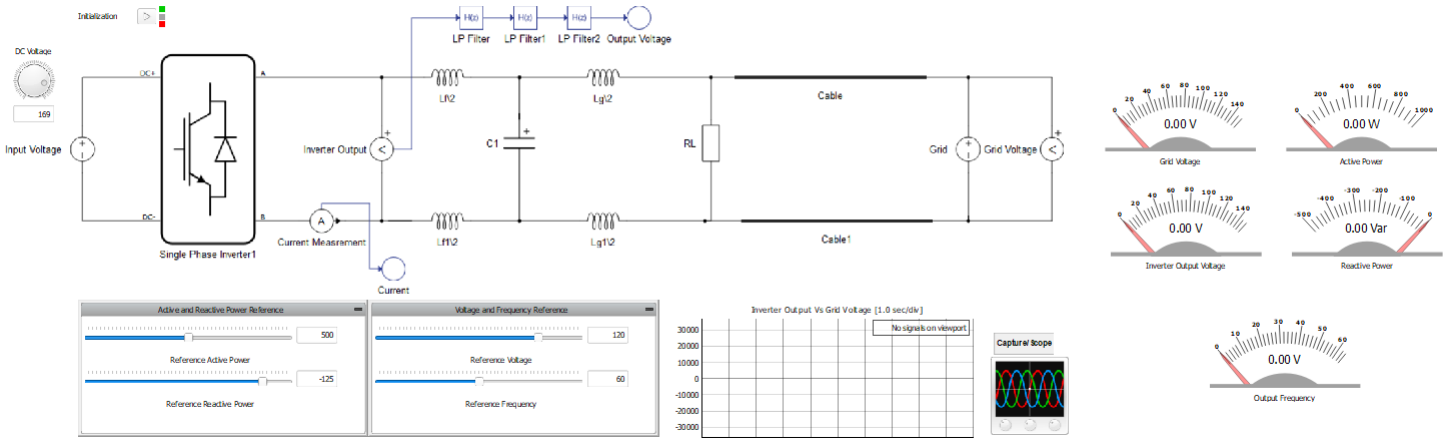


Figure 5.24: *The SCADA model for HIL simulation of figure 5.22*

overview of the HIL simulator with different control commands and measurements. The input voltage is controlled by a knob which can be seen in figure 5.25. The voltage can be varied during run time and hence can effectively mimic the variable input voltage source. The active power reference, reactive power reference, voltage reference and frequency reference can be provided

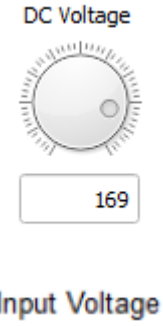


Figure 5.25: *Input Voltage control of inverter for HIL simulation of figure 5.22*

to the DSP controller using HIL SCADA. The voltage and frequency references can also be defined in DSP as a fixed data without the need of set points from HIL SCADA. The reference panels in SCADA are shown in figure 5.26. Similarly the magnitude of

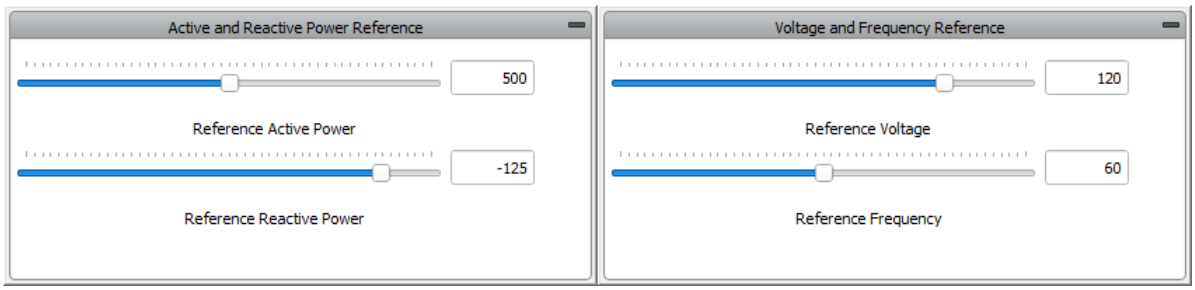


Figure 5.26: *Reference command panels in HIL SCADA*

output measurements are represented in the form of gauges which display output voltage, grid voltage, measured active and reactive power and output frequency. The output gauges are shown in figure 5.27.

Similarly the output voltage, grid voltage, measured current, active and reactive power measured by DSP is displayed by Scope of HIL SCADA, given in figure 5.28.

5.4.1 Approach to virtual inertia injection

This section explains the second part of our research in which virtual inertia is injected into the grid. The approach adopted,

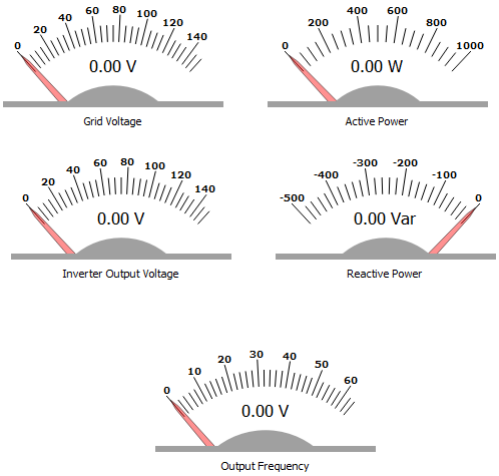


Figure 5.27: *Measurement gauges in HIL SCADA*

Figure 5.28: *Capture/Scope in HIL SCADA*

is based on dVOC along with detection of rate of change of frequency(ROCOF). dVOC stabilizes the grid in terms of voltage and frequency. Moreover, it follows the given power set points as we have seen in PLECS simulation in Chapter 4.

In virtual inertia approach, we are trying to slow down the fre-

quency change due to disturbances, by injecting power into the grid or absorbing power from the grid into Energy Storage System. Figure 5.29 shows the frequency drop, with and without virtual inertia.

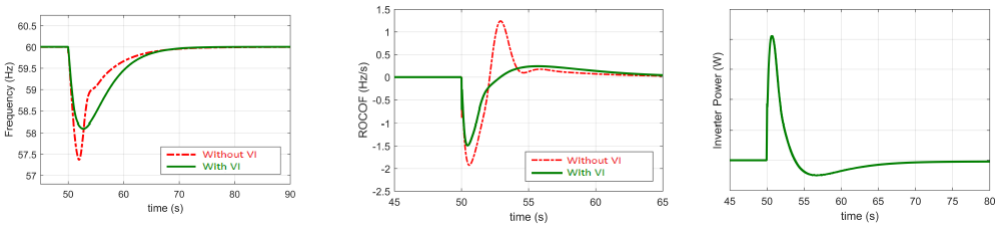


Figure 5.29: *Frequency drop, ROCOF and Inverter Power injection*

We can see from figure 5.29 that, in order to inject inertia in the system, the inverter has to inject power. The power injected by the inverter, to slow this frequency change is based on detecting the rate of change of frequency of the grid. To detect a frequency change, we have to first detect the frequency of the grid and then examine the rate of change of frequency. The injected power from inverter or power absorbed by the ESS "ΔP" is generally proportional to the ROCOF given as:

$$\Delta P \propto \frac{\mathrm{d}f}{\mathrm{d}t}$$

The relationship between ROCOF and power can be observed by a closed loop system given in figure 5.30.
It can be seen that the derivative of frequency difference with respect to time gives rate of change of frequency(ROCOF). The very small dead-band is added in the feedback to avoid the undesired action, due to small disturbances caused by measurement errors or transients. The $d\Delta f/dt$ value is then passed to a PI controller whose transfer function is given in equation 5.1:

$$H(s) = -K_p + K_I \frac{1}{s} \tag{5.1}$$

K_p and K_I are PI controller variables which are decided in DSP logic. A look-up table is normally used for the selection of these

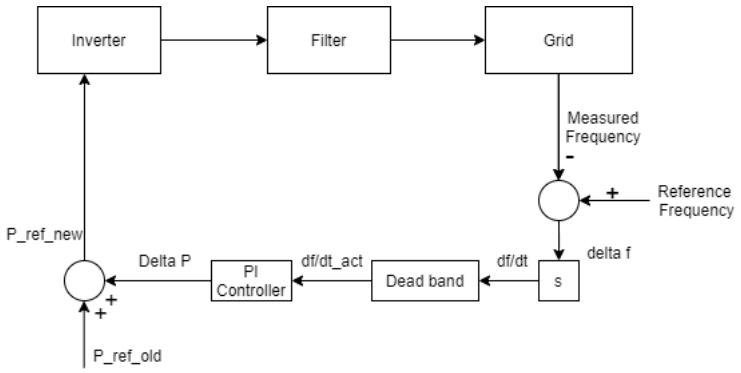


Figure 5.30: *Feedback control for virtual inertia injection*

values, which is obtained from simulation results. The value of K_p and K_I are chosen in such a way that in normal injection of power (without inertia injection demand), the value of "K_p" is kept smaller, however, for virtual inertia injection, the value of K_p is chosen to be large. K_p can be considered as a variable factor which decides the gain for power injection or power absorption. The negative sign shows that when the rate of change of frequency is negative ,i.e., frequency is dropping, the power has to be injected in the system.

If factor "K_p" is not a variable but a constant, then in case of frequency drop, rate of change of frequency is higher, hence higher amount of power is injected in the system to slow the frequency drop. However, during the process of restoration, factor K_p plays its role to slow down the restoration of frequency which is undesired. This is the reason why K_p is used as a variable in this approach.

Another factor that governs the values of K_p and K_I is the frequency change. We have to take into account the frequency change to determine the virtual inertia injection. To understand the dependence of K_p and K_I on frequency error, let us consider a situation in which frequency error is high enough, that we do not have enough room for significant change in frequency anymore as it may trip the frequency relay. In this case, if the ROCOF is not

high enough, still we will inject high power so that the frequency should not drop below the threshold and operate frequency relay. Hence the value of K_p and K_I will be chosen such that power injection is high. The scenario can be depicted in flowchart in figure 5.31

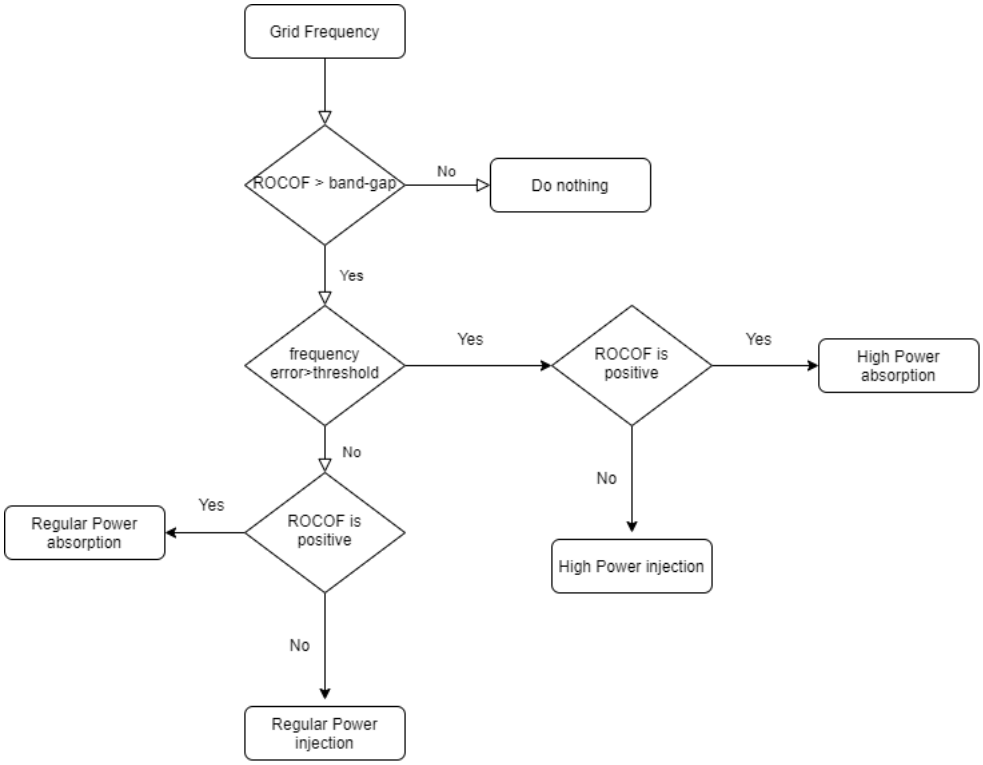


Figure 5.31: *Flow chart for PI controller parameter selection*

Hence the values of K_p and K_I of a PI controller are tuned depending on Δf and Rate of change of frequency(ROCOF). The threshold for frequency error, at which value of K_p should be high, irrespective of ROCOF is chosen to be $1.5Hz$ in this implementation. The dead band gap for ROCOF is chosen to $0.5Hz/s$. However, the system requirements for ROCOF is not to be less than $10Hz/s$. The DSP controller implements the logic, which determines the values of K_p and K_I, based on frequency error, ROCOF, frequency restoration, or providing inertia support to the

system.

5.4.2 The problem of appropriate voltage levels

The voltage levels in HIL simulation are order of magnitude of hundreds ($120V_{rms}$ in our implementation), and similar is true for power set points ($500W$ in our implementation). Whereas current is in order of magnitude of tens (6-12A in our implementation) along with frequency ($60Hz$). We have to import all this data to be processed by our DSP controller.

The DSP controller operates at the voltage levels of $0 - 3.3V$. Hence all the outputs from HIL simulator must be in this range. In order to scale the output signals from HIL simulator to DSP voltage levels, we have to calculate the minimum voltage that can be detectable by our DSP controller.

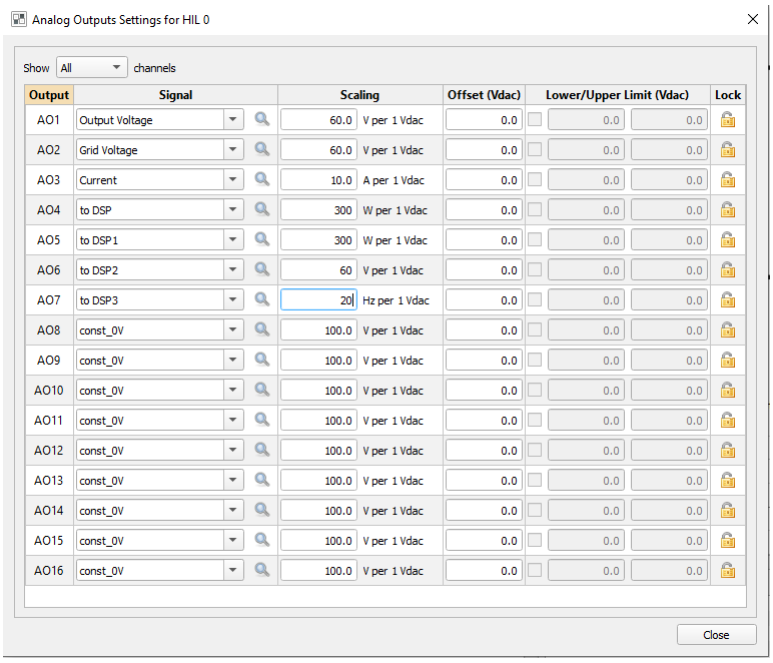
The resolution used by our Analog to Digital Converter (ADC) of DSP controller is 12-bit whereas the input range is $3.3V$. The step size is calculated using equation 5.2.

$$Step - size = \frac{Input\ Range}{Resolution} = \frac{3.3}{2^{12}} \tag{5.2}$$

Hence the step size is $0.806mV$. The minimum voltage level that can be detected by our DSP is $0.806mV$. Based on this calculation, we will calculate the appropriate scaling factors for signals from HIL simulator to DSP. We assume that voltage(rms) varies from $0 - 120V$ ($\pm 10\%$), hence a scaling factor of 60 is used. For an output value "$120V(rms)$" of HIL simulator, the DSP controller will receive "$2V(rms)$" which again will be scaled up in the DSP controller to "$120V(rms)$" for implementation of control algorithm.

Similarly for power set points, a scaling factor of 300 is used as we assume that the maximum power is "$1000W$". Even for the minimum power of "$1W$", the signal to DSP will be "$3mV$", which can be detected by the ADC. For the current, the maximum expected value is "$10A$" hence a scaling factor of 5 is used. In the

same way, a scaling factor of 20 is used for grid frequency. The scaling factors defined in HIL SCADA are shown in figure 5.32



Output	Signal	Scaling	Offset (Vdac)	Lower/Upper Limit (Vdac)	Lock
AO1	Output Voltage	60.0 V per 1 Vdac	0.0	0.0 0.0	
AO2	Grid Voltage	60.0 V per 1 Vdac	0.0	0.0 0.0	
AO3	Current	10.0 A per 1 Vdac	0.0	0.0 0.0	
AO4	to DSP	300 W per 1 Vdac	0.0	0.0 0.0	
AO5	to DSP1	300 W per 1 Vdac	0.0	0.0 0.0	
AO6	to DSP2	60 V per 1 Vdac	0.0	0.0 0.0	
AO7	to DSP3	20 Hz per 1 Vdac	0.0	0.0 0.0	
AO8	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO9	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO10	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO11	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO12	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO13	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO14	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO15	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	
AO16	const_0V	100.0 V per 1 Vdac	0.0	0.0 0.0	

Figure 5.32: *Scaling factors defined in HIL SCADA*

It should be noted that all the scaled down signals are eventually scaled up (only in value) to be used by the control algorithm of DSP. At the end, the signals injected from DSP controller to HIL simulator will be scaled up again for accurate visualization. The solution of scaling signals in HIL simulator and DSP controller is shown in 5.33.
It can be seen that the control algorithm takes regular original signal values(in software) and implements the logic. The resultant voltage control signals to drive the inverter gates are digital in nature and hence transmitted to digital pins of the HIL simulator using physical wires. Similarly, the computed powers are also transferred to the analog pins of the HIL simulator.

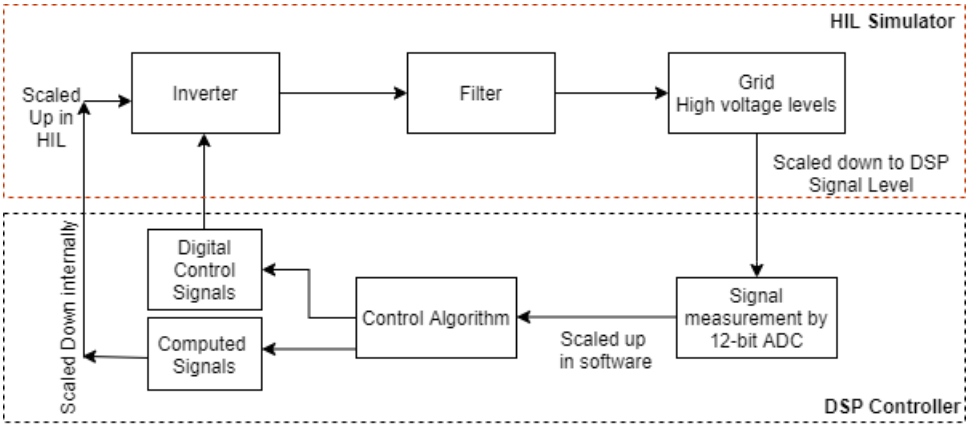


Figure 5.33: *Scaling of signals in HIL simulator and DSP controller*

5.4.3 Frequency Detection using DSP

In order to implement virtual inertia, the grid frequency has to be measured and consequently the rate of change of frequency(ROCOF). The constraints to be considered in frequency measurement for virtual inertia implementation are timing and accuracy. The frequency needs to be detected accurately and quickly, so that the adequate action can be taken.

In order to detect frequency, we consider a sinusoidal signal as shown in figure 5.34.
The signal given in figure 5.34 is grid voltage from HIL simulator to the DSP controller and scaled down by a factor of 60 as shown in figure 5.32 in real time. The signal is analog in nature and hence, has to be converted into digital domain using ADC of our DSP controller.

Accuracy of data: The ADC used is a 12 bit ADC with a step size of $0.806mV$. The ADC conversion is based on interrupt, which outputs result in an ADC output buffer. The output buffer is analyzed and based on frequency detection algorithm, frequency is measured. When sampled at a rate of 5kHz, the result obtained is shown in figure 5.35.
It can be seen from frequency measurement in figure 5.35, there

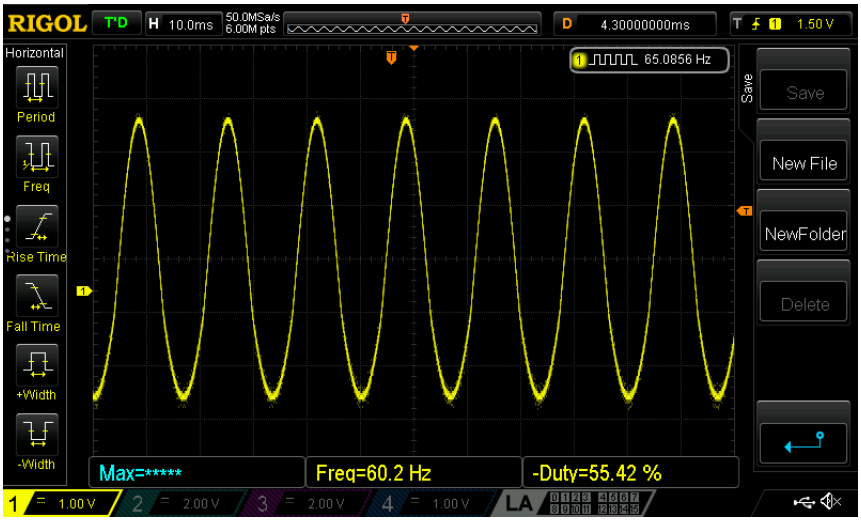


Figure 5.34: *Grid Voltage from HIL Simulator to DSP Controller*

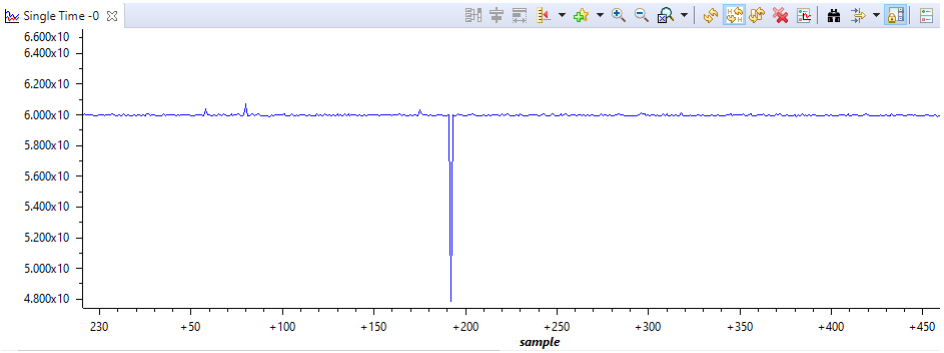


Figure 5.35: *Grid frequency measurement using ADC*

is a dip in frequency at sample 240 which is undesired. This is due to inaccurate measurement from ADC. In order to overcome this abrupt change or anomalous behavior of ADC, we sample the signal at a higher frequency of 30kHz along with devising a method of ADC array averaging. In this technique, the next sample is compared with the previous one and hence the ADC result is validated.

The grid frequency measurement at a high sampling frequency along with averaged ADC approach is shown in figure 5.36.
We can see that using higer sampling rate and ADC averaging approach, the data spikes are reduced and more accurate data is

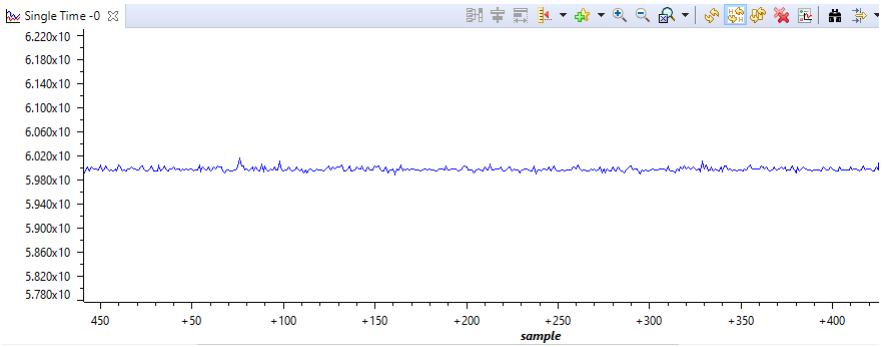


Figure 5.36: *Grid frequency measurement using averaged ADC and fast sampling*

obtained.

Timing Constraint: The grid frequency needs to be detected in a short duration of time in order to take an appropriate action. There is a time required by our ADC to convert analog signal into digital domain. Despite this time, the ADC channel and timers have to be energized which requires a few micro seconds. Moreover, the signal consists of positive and negative values and since our ADC is uni-polar, it cannot detect negative values. Hence an external circuit is required to level shift our signal and add a DC offset which results in our signal to be positive. The figure 5.37 shows a level shifter along with input and output.

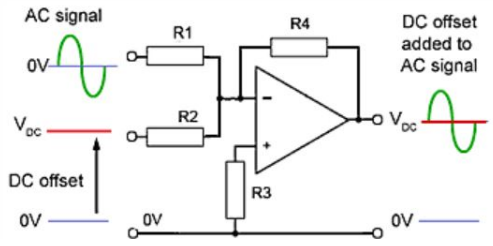


Figure 5.37: *Level shifter circuit for uni-polar ADC*

The level shifting incurs a delay which is undesired. Moreover, a physical external circuit is required to implement this level shifter, which will add cost. A signal of $60Hz$ requires minimum of

$16.66ms$. In total, the frequency is detected in more than $25ms$ which is calculated experimentally.

In order for our inertial response to be fast, the frequency needs to be detected in very short duration of time. Hence, a half cycle technique is adopted for frequency detection. In this technique, we will calculate the frequency based on half cycle of the signal. We will not use a shifter circuit and hence the negative part will not be converted into digital domain by our ADC. The scaled grid voltage along with ADC detected signal is shown in figure 5.38. The frequency detection technique is shown in figure 5.39. It can

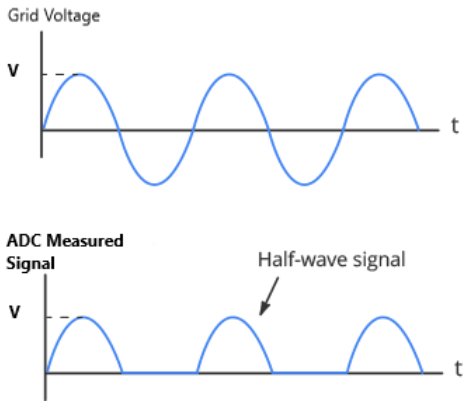


Figure 5.38: *Grid voltage signal from HIL simulator and ADC sampled signal by DSP controller*

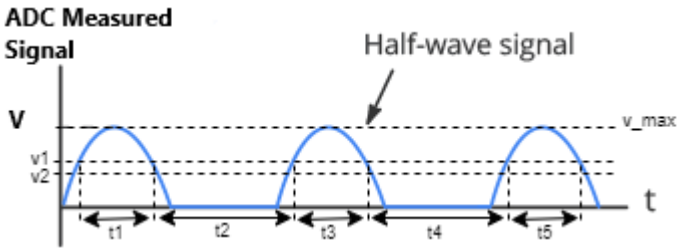


Figure 5.39: *Frequency measurement technique*

be seen from figure 5.39, that when the signal value is between "v1" and "v2", the DSP timer is started and stopped when signal value is again between "v1" and "v2". Hence t1 is measured.

Similarly, the timer starts and stops consecutively when the ADC value is between "v1" and "v2" which gives us t1, t2, t3, t4 and so on. The first instant of frequency measurement is achieved at 20 millisecond for the frequency of $60Hz$ including conversion and processing delays. Hence first value of frequency is obtained after time "T1" given as:

$$T1 = t1 + t2 + \Delta t$$

where Δt is the processing and conversion delay. The frequency is given as:

$$f_1 = \frac{1}{t1 + t2}$$

The second instant of frequency is obtained after time $t3 + \Delta t$ and the frequency is given as:

$$f_2 = \frac{1}{t2 + t3}$$

Hence we can see that the time required in frequency calculation is reduced to half as compared to "T1". The technique is experimentally tested and the time for first frequency measurement is 20 milliseconds whereas the next measurements are obtained in less than 10 milliseconds.

5.4.4 Programming the DSP Controller

This section discusses the programming of algorithm in DSP controller. The Integrated Development Environment (IDE) used is Code Composer Studio(CCS) where as the programming language is C.

The Program algorithm is similar to PLECS simulation described in Chapter 4 with an addition of virtual inertia control. The experimental setup for dVOC and virtual inertia implementation is shown in figure 5.40.

The first thing is the conversion of analog signals into digital signals using ADC. The DSP controller incorporates 16 ADC

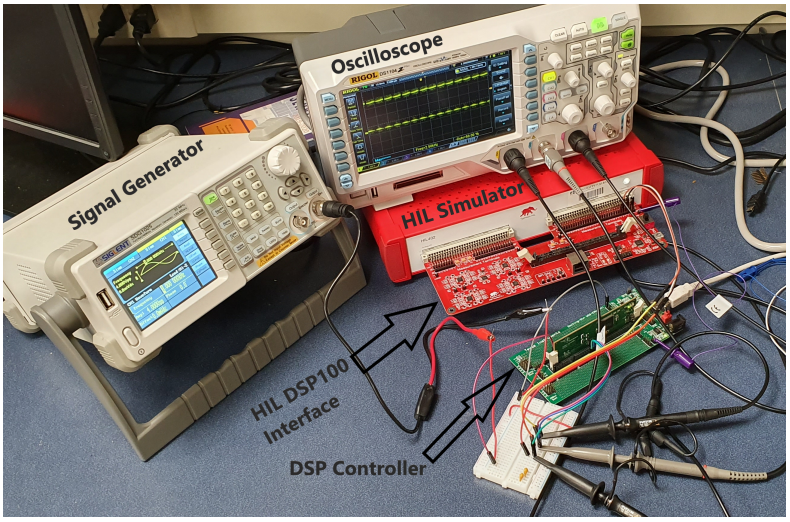


Figure 5.40: *Experimental setup for dVOC and virtual inertia implementation*

channels, hence all the analog signals can be sampled and processed without the channel constraint. The frequency detection is explained in previous section, however, in order to cater the virtual inertia control, rate of change of frequency has to be calculated. The ROCOF is calculated in DSP using equation 5.3.

$$ROCOF(n) = \frac{f(n) - f(n-1)}{T} \tag{5.3}$$

Where $ROCOF(n)$ is the rate of change of frequency at the current time step, $f(n)$ is the input frequency at the current time step, $f(n-1)$ is the input frequency at one time step delay and T is the sampling time. The Programming algorithm is explained in a block diagram given in figure 5.41.

It can be seen that the current and voltage signals are converted into $\alpha - \beta$ co-ordinates using Clark transformation. The dVOC implemented in DSP controller, is in-fact implementation of equation 3.25, which is re-written as:

$$\dot{v_\alpha} = \frac{\eta q_k^\star v_\alpha}{v_k^{\star 2}} - \frac{\eta p_k^\star v_\beta}{v_k^{\star 2}} + \eta i_\beta + \eta\alpha\phi_k v_\alpha - \omega_0 v_\beta$$

$$\dot{v_\beta} = \frac{\eta p_k^\star v_\alpha}{v_k^{\star 2}} + \frac{\eta q_k^\star v_\beta}{v_k^{\star 2}} - \eta i_\alpha + \eta\alpha\phi_k v_\beta + \omega_0 v_\alpha$$

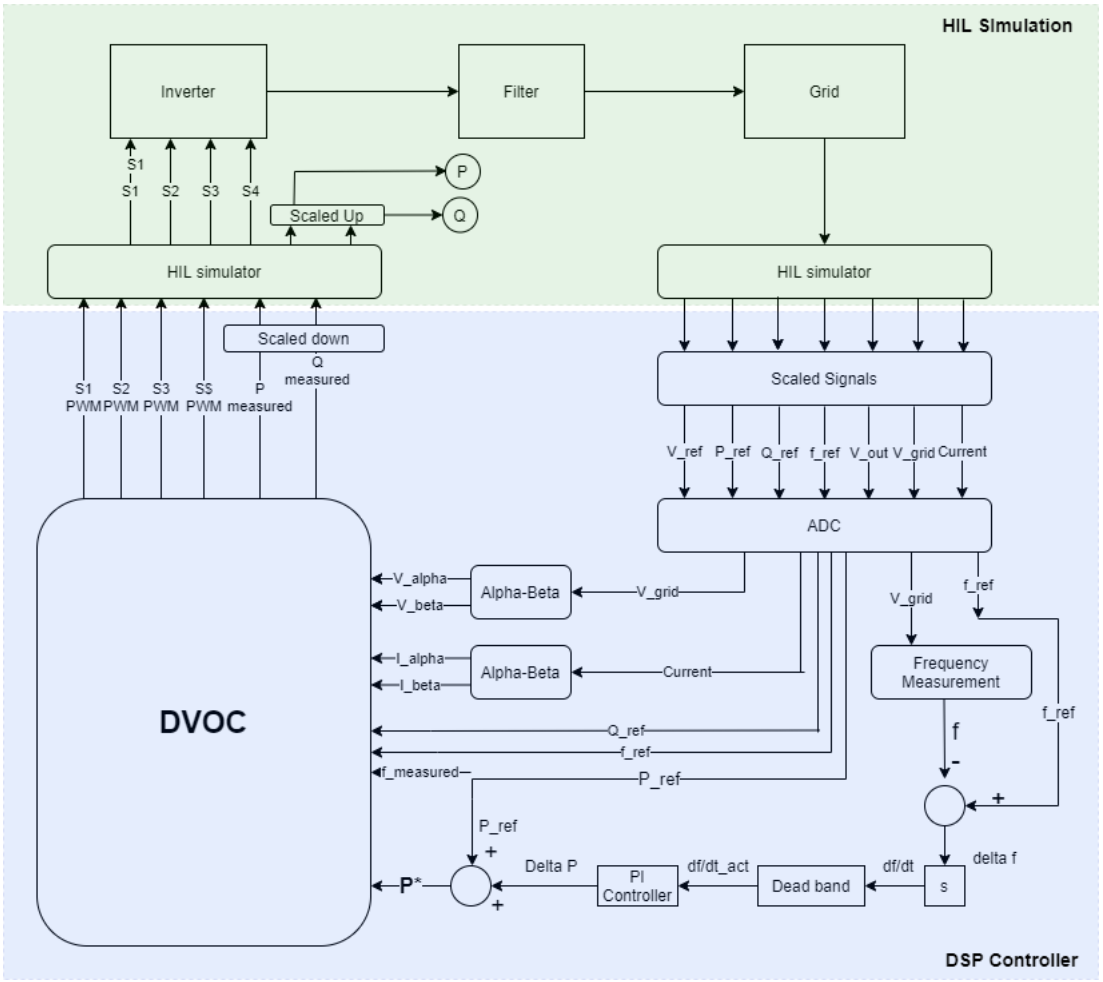


Figure 5.41: *DSP Controller algorithm*

When implementing this equation by DSP controller, all the time domain signals are converted into discrete domain where instead of time, we deal with samples. The main C-code for our DSP controller is given in Appendix. However, the results of this implementation are out of scope of this thesis.

CHAPTER 6

Conclusion and Future Work

6.0.1 Conclusion

We started with the idea of Distributed generation through Renewable energy sources due to their clean and environment friendly nature. Inverters are used to control these sources. Due to high penetration of Renewable energy sources, the grid is shifting from synchronous generators grid towards inverter based grid. In a system with large number of grid connected inverters, their control is also a challenge. Moreover, the inertia problem due to loss of synchronous generation is also discussed in detail in this thesis.

A detailed analysis among different control techniques for inverter control are discussed in chapter 2. It was shown that, fundamentally, the objective of all the topologies is to provide dynamic frequency response through power electronic converters. Grid following and grid forming approaches for inverter control are also discussed. Since, our inverter will contribute in grid stability, therefore we are focused on grid forming inverters.

Most grid following inverters are controlled based on PLL. The disadvantage of PLL is prominent in case of disturbances in the system. As a result, we find an approach that does not require PLL, rather the control is implemented based on local measurements. The concept of dVOC is described in chapter 3. In order to check stability, synchronization and power sharing capabilities, we simulated the dVOC algorithm using PLECS software which is described in detail in chapter 4.

We have observed from results of our simulation, that the inverter synchronizes itself with the grid in less than 100 milliseconds. Moreover, the frequency droop characteristics and voltage droop characteristics in steady state are also observed. The results shows that the dVOC follows the droop characteristics in steady state. Another advantage of dVOC is its power tracking capability. In simulation, we have proved that the dVOC control algorithm tracks the power set points very closely.

Another simulation, in which addition of an inverter to a normal grid is observed. The output shows the fast synchronization capability of dVOC in the presence of another inverter. It also shows the equal power sharing capabilities of dVOC controller. Another simulation without the AC grid is performed and the results shows us, that the dVOC control of inverter is valid for islanding operation.

The effects of loss of inverter on the system are observed in another simulation, which shows the similar characteristics of synchronization and load sharing. The loss of inverter in the absence of AC grid also concludes, that dVOC control algorithm can be used for islanding operation if power and current constraints are not violated.

In chapter 5, the dVOC control algorithm along with virtual inertia control is implemented in Hardware in the loop (HIL) testing environment. The algorithm is computed in DSP controller (TI320F28069) whereas the inverter, filter and grid is modeled in

HIL simulator. The proposed algorithm for virtual inertia is based on measurement of ROCOF along with values of K_p and K_I of PI controller. The results of this implementation are out of scope of this thesis.

To conclude, this research discusses the dVOC strategy and verifies its grid-forming functionality in an inverter-dominant electric power grid. Based only on local measurements, the dVOC inverters achieve almost instantaneous dynamic synchronization and load sharing. The analysis also verified the dVOC's embedded nonlinear droop law and how the inverters can be dispatched to optimize the power flow with programmable set-points. This research also discusses the idea of virtual inertia injection using HIL testing environment, which can be a promising solution for the inverter dominant grid.

6.0.2 Future work

Based on our dVOC control technique, we have concluded that the inverter can operate in islanding mode. Hence formation of mini and micro grids based on dVOC algorithm can be a futuristic vision. For this purpose, the dVOC has to be tested experimentally which can be the next step. In order to generate virtual inertia in the system, the algorithm proposed in Chapter 5 can be further validated by performing multiple testing scenarios in HIL simulator. After successful validation, an approach to increase power and current ratings should be devised to scale up the system.

As the research is funded by National Science Foundation(NSF)-USA, the goal is to find out the possibility and feasibility of replacing standby Synchronous Generators with inertial response of inverters. The service of virtual inertia will be traded in ancillary services market and hence the dependency on stand by Synchronous Generators only for inertial response will be reduced.

In the future, we can find different ways of virtual inertia sources. One major source of under-utilized energy lies in modern telcos

(telecommunication operators). Telcos need a high degree of reliability, and as a result, large amount of backup energy storage is deployed, which is unused during normal operating conditions. We can find different methods to utilize these resources using demand response techniques. This concept can be extended to telcos in specific due to non-utilization of their resources.

Using electric vehicles (EVs) to provide ancillary services can be a popular research area. Since, in idle, the electric vehicle can support the grid by injecting virtual inertia. Typically the control algorithm of the bidirectional converters in EVs can be modified in such a way as to implement virtual inertia support algorithm [39][40].

Another area of future research can be based on the fact, that virtual inertia emulated using energy storage systems and renewable energy sources is not constant as in the case of traditional synchronous generation. The available inertia in the system will depend upon whether RES units are online or not, and resource availability (wind speed, irradiance, and state of charge in case of ESS) [41]. System inertia estimation is thus going to be critical for planning purposes for system operators, in the future power systems with high RES penetrations. Furthermore, such estimates can provide helpful insights into the stable real-time operation of a power system.

Appendix

```
1 // TI File $Revision: /main/3 $
2 // Checkin $Date: March 15, 2020   17:25:54 $
3 //###########################################################################
4 //
5 // FILE:    dVOC_VI.c
6 //
7 // TITLE:  dVOC and Virtual Inertia Control
8 //
9 // ASSUMPTIONS:
10 //
11 //    This program requires the F2806x header files.
12 //
13 //    Make sure the CPU clock speed is properly defined in
14 //    F2806x_Examples.h before compiling this example.
15 //
16 //
17 //     $Boot_Table:
18 //
19 //     While an emulator is connected to your device, the
       TRSTn pin = 1,
20 //     which sets the device into EMU_BOOT boot mode. In this
       mode, the
21 //     peripheral boot modes are as follows:
22 //
23 //       Boot Mode:        EMU_KEY          EMU_BMODE
24 //                         (0xD00)          (0xD01)
25 //       -----------------------------------------
26 //       Wait              !=0x55AA         X
27 //       I/O               0x55AA           0x0000
28 //       SCI               0x55AA           0x0001
29 //       Wait              0x55AA           0x0002
```

```
30 //        Get_Mode            0x55AA            0x0003
31 //        SPI                 0x55AA            0x0004
32 //        I2C                 0x55AA            0x0005
33 //        OTP                 0x55AA            0x0006
34 //        ECANA               0x55AA            0x0007
35 //        SARAM               0x55AA            0x000A     <-- "Boot
       to SARAM"
36 //        Flash               0x55AA            0x000B
37 //        Wait                0x55AA            Other
38 //
39 //    Write EMU_KEY to 0xD00 and EMU_BMODE to 0xD01 via the
       debugger
40 //    according to the Boot Mode Table above. Build/Load
       project,
41 //    Reset the device, and Run example
42 //
43 //    $End_Boot_Table
44 //
45 //
46 // Description:
47 //
48 //    This example sets up the PLL in x16/2 mode.
49 //
50 //    For 80 MHz devices (default)
51 //    (assuming a 10Mhz input clock).
52 //
53 //    Interrupts are enabled and the ePWM1 is setup to
       generate a periodic
54 //    ADC SOC - ADCINT1. Two channels are converted, ADCINA4
       and ADCINA2.
55 //
56 //    Watch Variables:
57 //
58 //          freq_Val      frequency value
59 //          PI_look_up[100]      for Kp and Ki values
60 //          signal_look_up[100]  Generating PWM signal.
       signal_look_up is modulating signal
61
62 //
63 //
64 //###########################################################################
65 // $TI Release: 2806x C/C++ Header Files and Peripheral
       Examples V1.00 $
66 // $Release Date: January 11, 2011 $
67 //###########################################################################
68
69 #include "DSP28x_Project.h"      // Device Headerfile and
       Examples Include File
70 #include "stdbool.h"
71 #include "stdlib.h"
```

```
72 #include "math.h"
73
74 // Prototype statements for functions found within this file.
75 interrupt void adc_isr(void);
76 interrupt void  epwm2_isr(void);
77 interrupt void  epwm3_isr(void);
78 interrupt void  epwm4_isr(void);
79 interrupt void  epwm5_isr(void);
80 void Epwm3_Config(void);
81 void Epwm4_Config(void);
82 void Epwm5_Config(void);
83 void Adc_Config(void);
84
85 #define UPPER_THRESHOLD 1060
86 #define LOWER_THRESHOLD 1040
87 #define CYCLE_MEASUREMENT_PRD 500
88 #define SYS_FREQUENCY 80000000          //80MHz
89 #define ADC_VREFLO 0
90 #define ADC_VREFHI 3.3
91 #define ADC_MAX_VALUE 4095
92 #define SCALE_FACTOR_HIL 300
93
94
95 //***Defing MACROS for TIMERS
96
97 #define PWM1_FREQ 2000    // Should be more than 1.3kHz
98 #define PWM2_FREQ 2000
99 #define PWM3_FREQ 2000
100
101 #define PHASE_SHIFT_AB 120
102 #define PHASE_SHIFT_AC 240
103 #define FULL_CYCLE      360
104
105 //*** MODULATNG SIGNAL / RFERENCE SIGNAL
106
107 #define MOD_SIG_FREQ 60
108
109 // Global variables used in this example:
110 Uint16 LoopCount;
111 Uint16 ConversionCount;
112 Uint16 Voltage1[1000];
113 Uint16 Voltage2[1000];
114 Uint16 timing_events[100];
115 Uint16 timing_count = 0;
116 Uint16 Previous_Voltage;
117 Uint16 Present_Voltage;
118 bool interrupt_occured = false;
119 Uint16 interrupt_counter = 0;
120 //int Sample_difference_array[1000];
121 int Sample_Difference = 0;
```

```
122 Uint16 counter= 0;
123 Uint16 freq_occurance[1000];
124 Uint16 freq_sample = 0;
125 Uint16 Sample_difference_exceed = 0;
126 Uint16 Adc_Chanel;
127 Uint16 Adc_Count = 0;
128 Uint16 adc_zero_count = 0;
129 Uint16 freq_timing = 0;
130 Uint16 first_cycle = 0;
131 Uint16 second_cycle = 0;
132 Uint16 full_cycle_array[1000];
133 Uint16 full_cycle_index = 0;
134 bool previous_cycle = false;
135 bool cycle_detected = false;
136 bool first_cycle_detected = true;
137 Uint16 half_cycle_timing = 0;
138 double freq_Val = 0;
139 double div_Val   = 0;
140 double Voltage = 0;
141 double Vref = ADC_VREFHI-ADC_VREFLO;
142 Uint16 i = 0;
143
144 double cosine90;                    // For alpha beta single
      phase
145 double sin90;                       // For alpha beta single
      phase
146 double PI_look_up[100];             // For PI look up table
147 double signal_look_up[100];         // For control signal
      look up table
148 double eta;
149 double alpha;
150 double v_ref;
151 double v_ref_alpha;
152 double v_ref_beta;
153 double v_grid;
154 double v_grid_alpha;
155 double v_grid_beta;
156 double v_out;
157 double v_out_alpha;
158 double v_out_beta;
159 double current;
160 double current_alpha;
161 double current_beta;
162 double p_ref;
163 double q_req;
164 double f_ref;
165 double freq_measurement;
166 //** Defining Variables for timers
167 Uint16 Tmr3_PRD    = SYS_FREQUENCY/PWM1_FREQ;
168 Uint16 Tmr4_PRD    = SYS_FREQUENCY/PWM2_FREQ;
```

```
169 Uint16 Tmr5_PRD    = SYS_FREQUENCY/PWM3_FREQ;
170 float Pwm1_Duty    = 0.5;        // By Default
171 float Pwm2_Duty    = 0.5;        // By Default
172 float Pwm3_Duty    = 0.5;        // By Default
173 Uint16 Tmr3_CMPA;
174 Uint16 Tmr4_CMPA;
175 Uint16 Tmr5_CMPA;
176 Uint16  PWM_Cycles;
177 Uint16 j = 0;
178
179 Uint16 Phase_Shift1 =
        ((double)PHASE_SHIFT_AB*SYS_FREQUENCY)/((double)PWM1_FREQ*FULL_CYCLE);
180 Uint16 Phase_Shift2 =
        ((double)PHASE_SHIFT_AC*SYS_FREQUENCY)/((double)PWM1_FREQ*FULL_CYCLE);
181 double angle_spacing;
182 double angle_rad = 0;
183 double angle_degree = 0;
184 double sin_look_up[PWM1_FREQ/MOD_SIG_FREQ];
185 double look_up[PWM1_FREQ/MOD_SIG_FREQ];
186
187 static void sig_lookup_fn(void);
188 void clark_trans(double sig);                    //This function
        performs Clark transformation
189 double dvoc_alg(double volt, double freq, double current);
190 double active_p(double v,double i);
191 double reactive_p(double v,doublei);
192
193 Uint16 epwm3_inst = 0;
194 Uint16 epwm4_inst = 0;
195 Uint16 epwm5_inst = 0;
196 void main()
197 {
198
199     PWM_Cycles            = PWM1_FREQ/MOD_SIG_FREQ;
200     angle_spacing         = (double)360/PWM_Cycles;
201
202 sig_lookup_fn();
203
204
205
206
207     //Intialize the CMPA values for Timer
208
209 //     Tmr3_CMPA   = ((float)Tmr3_PRD*Pwm1_Duty)/100;
210 //     Tmr4_CMPA   = ((float)Tmr4_PRD*Pwm2_Duty)/100;
211 //     Tmr5_CMPA   = ((float)Tmr5_PRD*Pwm3_Duty)/100;
212
213     Tmr3_CMPA   = ((float)Tmr3_PRD*Pwm1_Duty);
214     Tmr4_CMPA   = ((float)Tmr4_PRD*Pwm2_Duty);
215     Tmr5_CMPA   = ((float)Tmr5_PRD*Pwm3_Duty);
```

```
216
217
218 // Step 1. Initialize System Control:
219 // PLL, WatchDog, enable Peripheral Clocks
220 // This example function is found in the F2806x_SysCtrl.c
      file.
221    InitSysCtrl();
222
223
224 // Step 2. Initialize GPIO:
225 // This example function is found in the F2806x_Gpio.c file
      and
226 // illustrates how to set the GPIO to it's default state.
227 // InitGpio();  // Skipped for this example
228
229 // Step 3. Clear all interrupts and initialize PIE vector
      table:
230 // Disable CPU interrupts
231    DINT;
232
233 // Initialize the PIE control registers to their default
      state.
234 // The default state is all PIE interrupts disabled and flags
235 // are cleared.
236 // This function is found in the F2806x_PieCtrl.c file.
237    InitPieCtrl();
238
239 // Disable CPU interrupts and clear all CPU interrupt flags:
240    IER = 0x0000;
241    IFR = 0x0000;
242
243 // Initialize the PIE vector table with pointers to the shell
      Interrupt
244 // Service Routines (ISR).
245 // This will populate the entire table, even if the interrupt
246 // is not used in this example.  This is useful for debug
      purposes.
247 // The shell ISR routines are found in F2806x_DefaultIsr.c.
248 // This function is found in F2806x_PieVect.c.
249    InitPieVectTable();
250
251 // Interrupts that are used in this example are re-mapped to
252 // ISR functions found within this file.
253    EALLOW;  // This is needed to write to EALLOW protected
         register
254    PieVectTable.ADCINT1 = &adc_isr;
255    PieVectTable.EPWM2_INT = &epwm2_isr;
256    PieVectTable.EPWM3_INT = &epwm3_isr;
257    PieVectTable.EPWM4_INT = &epwm4_isr;
258    PieVectTable.EPWM5_INT = &epwm5_isr;
```

```
259
260   // PieVectTable.EPWM3_INT = &epwm3_isr;
261    EDIS;    // This is needed to disable write to EALLOW
          protected registers
262
263 // Step 4. Initialize all the Device Peripherals:
264 // This function is found in F2806x_InitPeripherals.c
265 // InitPeripherals(); // Not required for this example
266    InitAdc();  // For this example, init the ADC
267
268 // Step 5. User specific code, enable interrupts:
269
270 // Enable ADCINT1 in PIE
271    PieCtrlRegs.PIEIER1.bit.INTx1 = 1;    // Enable INT 1.1 in
          the PIE
272    PieCtrlRegs.PIEIER3.bit.INTx2 = 1;
273    PieCtrlRegs.PIEIER3.bit.INTx3 = 1;
274    PieCtrlRegs.PIEIER3.bit.INTx4 = 1;
275    PieCtrlRegs.PIEIER3.bit.INTx5 = 1;
276
277    IER |= M_INT1;                        // Enable CPU
          Interrupt 1
278    IER |= M_INT3;                        // Enable CPU
          Interrupt Group 3
279    EINT;                                 // Enable Global
          interrupt INTM
280    ERTM;                                 // Enable Global
          realtime interrupt DBGM
281
282    LoopCount = 0;
283    ConversionCount = 0;
284
285 // Configure ADC
286     EALLOW;
287     AdcRegs.ADCCTL2.bit.ADCNONOVERLAP = 1;  // Enable
          non-overlap mode
288     AdcRegs.ADCCTL1.bit.INTPULSEPOS = 1;    //ADCINT1 trips
          after AdcResults latch
289     AdcRegs.INTSEL1N2.bit.INT1E     = 1;    //Enabled ADCINT1
290     AdcRegs.INTSEL1N2.bit.INT1CONT  = 0;    //Disable ADCINT1
          Continuous mode
291     AdcRegs.INTSEL1N2.bit.INT1SEL   = 0;    // setup EOC1 to
          trigger ADCINT1 to fire
292     AdcRegs.ADCSOC0CTL.bit.CHSEL    = 4;    // set SOC0
          channel select to ADCINA4
293 //    AdcRegs.ADCSOC1CTL.bit.CHSEL    = 4;    // set SOC1
      channel select to ADCINA2
294     AdcRegs.ADCSOC0CTL.bit.TRIGSEL  = 5;    // set SOC0 start
          trigger on EPWM1A, due to round-robin SOC0 converts
          first then SOC1/
```

```
295 //     AdcRegs.ADCSOC1CTL.bit.TRIGSEL  = 5;     // set SOC1
       start trigger on EPWM1A, due to round-robin SOC0 converts
       first then SOC1
296      AdcRegs.ADCSOC0CTL.bit.ACQPS     = 9;     //set SOC0 S/H
           Window to 7 ADC Clock Cycles, (6 ACQPS plus 1)
297 //     AdcRegs.ADCSOC1CTL.bit.ACQPS     = 6;     //set SOC1 S/H
       Window to 7 ADC Clock Cycles, (6 ACQPS plus 1)
298      EDIS;
299
300 // Assumes ePWM1 clock is already enabled in InitSysCtrl();
301     EPwm1Regs.ETSEL.bit.SOCAEN    = 1;          // Enable SOC on
           A group
302     EPwm1Regs.TBCTL.bit.HSPCLKDIV= 0;
303     EPwm1Regs.ETSEL.bit.SOCASEL   = 2;          // Select SOC
           from CMPA on upcount
304     EPwm1Regs.ETPS.bit.SOCAPRD    = 1;          // Generate pulse
           on 1st event
305 //    EPwm1Regs.CMPA.half.CMPA      = 0x00FF;     // Set compare
       A value
306     EPwm1Regs.TBPRD               = 0x160;    // Set period for
           ePWM1
307     EPwm1Regs.TBCTL.bit.CTRMODE   = 0;          // count up and
           start
308
309 // ***Defining EPwm2 for timing purposes
310
311     EPwm2Regs.TBPRD                 = CYCLE_MEASUREMENT_PRD;
312     EPwm2Regs.TBCTL.bit.CLKDIV      = 0;
313     EPwm2Regs.TBCTL.bit.HSPCLKDIV   = 0;
314 //    EPwm2Regs.TBCTL.bit.CTRMODE     = 0;
315     EPwm2Regs.ETSEL.bit.INTEN       = 1;
316     EPwm2Regs.ETSEL.bit.INTSEL      = 2;
317     EPwm2Regs.ETPS.bit.INTPRD       = 1;
318
319
320     //** Configuring EPwm3
321
322     Epwm3_Config();
323     Epwm4_Config();
324     Epwm5_Config();
325
326     //****** COnfiguring GPIO pins for inverter switches leg 1
327
328     // GPIO Pins for Epwm3
329        EALLOW;
330         GpioCtrlRegs.GPAMUX1.bit.GPIO4   = 1;
331         GpioCtrlRegs.GPAMUX1.bit.GPIO5   = 1;
332
333         GpioCtrlRegs.GPADIR.bit.GPIO4    = 1;
334         GpioCtrlRegs.GPADIR.bit.GPIO5    = 1;
```

```
335
336     EDIS;
337     //** Initializing GPIO PINS for Epwm3
338
339     GpioDataRegs.GPASET.bit.GPIO4    = 1;
340     GpioDataRegs.GPACLEAR.bit.GPIO5 = 1;
341
342 //****** COnfiguring GPIO pins for inverter switches leg 2
343
344     // GPIO Pins for Epwm4
345        EALLOW;
346        GpioCtrlRegs.GPAMUX1.bit.GPIO6   = 1;
347        GpioCtrlRegs.GPAMUX1.bit.GPIO7   = 1;
348
349        GpioCtrlRegs.GPADIR.bit.GPIO6    = 1;
350        GpioCtrlRegs.GPADIR.bit.GPIO7    = 1;
351
352     EDIS;
353     //** Initializing GPIO PINS for Epwm4
354
355     GpioDataRegs.GPASET.bit.GPIO6    = 1;
356     GpioDataRegs.GPACLEAR.bit.GPIO7 = 1;
357
358     //****** COnfiguring GPIO pins for inverter switches leg 3
359
360        // GPIO Pins for Epwm5
361           EALLOW;
362           GpioCtrlRegs.GPAMUX1.bit.GPIO8   = 1;
363           GpioCtrlRegs.GPAMUX1.bit.GPIO9   = 1;
364
365           GpioCtrlRegs.GPADIR.bit.GPIO8    = 1;
366           GpioCtrlRegs.GPADIR.bit.GPIO9    = 1;
367
368        EDIS;
369        //** Initializing GPIO PINS for Epwm4
370
371        GpioDataRegs.GPASET.bit.GPIO8    = 1;
372        GpioDataRegs.GPACLEAR.bit.GPIO9 = 1;
373
374
375 // ***Defining EPWM3 for ADC timing purposes
376 /*
377     EPwm3Regs.TBPRD                  = 0x32;
378     EPwm3Regs.TBCTL.bit.CLKDIV       = 0;
379     EPwm3Regs.TBCTL.bit.HSPCLKDIV    = 0;
380     EPwm3Regs.ETSEL.bit.INTEN        = 1;
381     EPwm3Regs.ETSEL.bit.INTSEL       = 2;
382     EPwm3Regs.ETPS.bit.INTPRD        = 1;
383
384 // GPIO Pins enabled on counter
```

```
385     EALLOW;
386     GpioCtrlRegs.GPAMUX1.bit.GPIO0   = 1;
387     GpioCtrlRegs.GPAMUX1.bit.GPIO2   = 1;
388     GpioCtrlRegs.GPAMUX1.bit.GPIO4   = 1;
389
390     GpioCtrlRegs.GPADIR.bit.GPIO0    = 1;
391     GpioCtrlRegs.GPADIR.bit.GPIO2    = 1;
392     GpioCtrlRegs.GPADIR.bit.GPIO4    = 1;
393 EDIS;
394 //    GpioDataRegs.GPASET.bit.GPIO2 = 1;
395     EPwm1Regs.AQCTLA.bit.PRD = 3;
396     EPwm2Regs.AQCTLA.bit.PRD = 3;
397     EPwm3Regs.AQCTLA.bit.PRD = 3;
398 */
399         EPwm3Regs.TBCTR = 0;
400         EPwm4Regs.TBCTR = 0;
401         EPwm5Regs.TBCTR = 0;
402 // Wait for ADC interrupt
403
404
405
406     while(1)
407     {
408         if(EPwm3Regs.TBCTR >= Phase_Shift1){
409             EPwm4Regs.TBCTL.bit.CTRMODE    = 0;
410
411         }
412
413         if(EPwm3Regs.TBCTR >= Phase_Shift2){
414                     EPwm5Regs.TBCTL.bit.CTRMODE    = 0;
415
416                 }
417
418         Voltage =
                ((double)Present_Voltage/(double)ADC_MAX_VALUE)*(double)Vref;
419
420        LoopCount++;
421       if(Present_Voltage>LOWER_THRESHOLD &&
             Present_Voltage<UPPER_THRESHOLD){
422           EPwm2Regs.TBCTL.bit.CTRMODE    = 0;
423
424        }
425       if(cycle_detected == true){
426
427           if(first_cycle_detected == true){
428       first_cycle = half_cycle_timing;
429       first_cycle_detected = false;
430           }
431           else{
432
```

```
433         second_cycle = half_cycle_timing;
434         first_cycle_detected = true;
435             }
436         freq_timing = first_cycle+second_cycle;
437  //       i++;
438
439  //       tatti = 1334000.0;
440          div_Val    =
                (double)CYCLE_MEASUREMENT_PRD*(double)freq_timing ;
441          freq_Val = (double)SYS_FREQUENCY/div_Val;
442         cycle_detected= false;
443         }
444
445
446 /*
447         if(full_cycle_index>=2 && cycle_detected == true){
448         first_cycle = full_cycle_array[i];
449         second_cycle = full_cycle_array[i+1];
450         freq_timing = first_cycle+second_cycle;
451         i++;
452
453  //       tatti = 1334000.0;
454          div_Val    =
                (double)CYCLE_MEASUREMENT_PRD*(double)freq_timing ;
455          freq_Val = (double)SYS_FREQUENCY/div_Val;
456         cycle_detected= false;
457         }
458    */
459
460
461 /*       if(freq_sample>0 && freq_occurance[freq_sample-1]>100){
462           if(previous_cycle == false){
463           first_cycle = freq_occurance[freq_sample-1];
464           previous_cycle = true;
465           }
466           if(previous_cycle == true){
467
468              second_cycle = freq_occurance[freq_sample-1];
469              previous_cycle = false;
470           }
471
472         freq_timing = first_cycle+second_cycle;
473
474       }
475
476 */
477
478
479        /*if (ConversionCount>0 && interrupt_occured == true){
480             Sample_Difference =
```

```
                abs(Present_Voltage-Previous_Voltage);
481             Sample_difference_array[interrupt_counter] =
                    Sample_Difference;
482             Previous_Voltage = Present_Voltage;
483
484             if(interrupt_counter == 999)
485               {
486                 interrupt_counter = 0;
487               }
488               else interrupt_counter++;
489             interrupt_occured = false;
490             if(Sample_Difference > 500){
491                     Sample_difference_exceed++;
492
493               }
494
495         }*/
496
497
498     }
499
500 }
501
502
503 interrupt void  adc_isr(void)
504 {
505 ConversionCount++;
506   //  interrupt_occured = true;
507   Present_Voltage = AdcResult.ADCRESULT0;
508   if(Present_Voltage>LOWER_THRESHOLD &&
        Present_Voltage<UPPER_THRESHOLD){
509       EPwm2Regs.TBCTR = 0;
510
511
512             freq_occurance[freq_sample] = counter;
513             if(counter>100){
514                 half_cycle_timing = counter;
515                 //full_cycle_array[full_cycle_index]=counter;
516                 //full_cycle_index++;
517                 cycle_detected = true;
518             }
519             freq_sample++;
520             counter = 0;
521         }
522
523
524
525   AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;     //Clear ADCINT1
        flag reinitialize for next SOC
526   PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;    // Acknowledge
```

```
        interrupt to PIE
527 //  timing_events[timing_count] = counter;
528 //  timing_count++;
529   return;
530 }
531
532 interrupt void  epwm2_isr(void)
533 {
534
535 counter++;
536
537   EPwm2Regs.ETCLR.bit.INT = 1;        //Enables other interrups
        can be generated
538   PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;    // Acknowledge
        interrupt to PIE
539
540   return;
541 }
542
543 interrupt void  epwm3_isr(void)
544 {
545
546     Tmr3_CMPA  = ((float)Tmr3_PRD*sin_look_up[epwm3_inst]);
547     EPwm3Regs.CMPA.half.CMPA     =  Tmr3_CMPA ;
548     if(epwm3_inst < PWM_Cycles){
549
550         epwm3_inst++;
551     }
552     else{
553
554         epwm3_inst = 0;
555     }
556   EPwm3Regs.ETCLR.bit.INT = 1;        //Enables other interrups
        can be generated
557   PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;    // Acknowledge
        interrupt to PIE
558
559   return;
560 }
561
562 interrupt void  epwm4_isr(void)
563 {
564     Tmr4_CMPA  = ((float)Tmr4_PRD*sin_look_up[epwm4_inst]);
565     EPwm4Regs.CMPA.half.CMPA     =    Tmr4_CMPA;
566     if(epwm4_inst < PWM_Cycles){
567
568         epwm4_inst++;
569     }
570     else{
571
```

```
572             epwm4_inst = 0;
573         }
574
575       EPwm4Regs.ETCLR.bit.INT = 1;        //Enables other interrups
              can be generated
576       PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;     // Acknowledge
              interrupt to PIE
577
578       return;
579 }
580
581 interrupt void  epwm5_isr(void)
582 {
583
584         Tmr5_CMPA  = ((float)Tmr5_PRD*sin_look_up[epwm5_inst]);
585         EPwm4Regs.CMPA.half.CMPA     =     Tmr5_CMPA;
586
587         if(epwm5_inst < PWM_Cycles){
588
589             epwm5_inst++;
590         }
591         else{
592
593             epwm5_inst = 0;
594         }
595
596       EPwm5Regs.ETCLR.bit.INT = 1;        //Enables other interrups
              can be generated
597       PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;     // Acknowledge
              interrupt to PIE
598
599       return;
600 }
601
602 void Epwm3_Config(void){
603
604         // *** Defining EPwm3 for PWM generation for inverter
605             EPwm3Regs.TBPRD                 = Tmr3_PRD; // For
                  frequency of 2kHz
606             EPwm3Regs.TBCTR                 = 0;
607             EPwm3Regs.TBCTL.bit.CLKDIV      = 0;
608             EPwm3Regs.TBCTL.bit.HSPCLKDIV   = 0;
609             EPwm3Regs.TBCTL.bit.CTRMODE     = 0;
610             EPwm3Regs.CMPA.half.CMPA        = Tmr3_CMPA;
611         //    EPwm3Regs.CMPB                  = 39999;
612         //    EPwm3Regs.AQCTLA.bit.CBU        = 2;                //
              When CTR = CMPB; then make the GPIO_EPWM3 = High;
613             EPwm3Regs.AQCTLA.bit.CAU        = 1;                //
                  When CTR = CMPA; then make the GPIO_EMPWM3_S1 = LOW;
614             EPwm3Regs.AQCTLA.bit.PRD        = 2;                //
```

```
            When CTR = PRD; then make the GPIO_EPWM3_S1 = HIGH;
615         EPwm3Regs.AQCTLA.bit.ZRO        = 2;                    //
            When CTR = ZRO; then make the GPIO_EPWM3_S1 = HIGH;
616
617         EPwm3Regs.AQCTLB.bit.CAU        = 2;                    //
            When CTR = CMPA; then make the GPIO_EMPWM3_S2 =
            HIGH;
618         EPwm3Regs.AQCTLB.bit.PRD        = 1;                    //
            When CTR = PRD; then make the GPIO_EPWM3_S2 = LOW;
619         EPwm3Regs.AQCTLB.bit.ZRO        = 1;                    //
            When CTR = ZRO; then make the GPIO_EPWM3_S2 = LOW;
620
621         EPwm3Regs.DBCTL.bit.HALFCYCLE = 0;
622         EPwm3Regs.DBCTL.bit.IN_MODE    = 0;                     //
            EPWMA is the trigger for both rising and falling
            edges
623         EPwm3Regs.DBCTL.bit.POLSEL     = 2;                     // If
            we are using dead band then please refer to fig
            3-29 for understanding
624         EPwm3Regs.DBCTL.bit.OUT_MODE  = 3;                      //
            Dead band is applied both for falling and rising
            edges
625
626         EPwm3Regs.DBRED                = 10;                    //
            Dead band time delay for rising edge = 10
            (12.5ns*10)
627         EPwm3Regs.DBFED                = 10;                    //
            Dead band time delay for falling edge = 10
            (12.5ns*10)
628
629         //** Enabling Interrupts
630
631         EPwm3Regs.ETSEL.bit.INTEN       = 1;
632         EPwm3Regs.ETSEL.bit.INTSEL      = 2;
633         EPwm3Regs.ETPS.bit.INTPRD       = 1;
634
635         return;
636 }
637
638 void Epwm4_Config(void){
639
640     // *** Defining EPwm3 for PWM generation for inverter
641             EPwm4Regs.TBPRD                 = Tmr4_PRD; // For
                frequency of 2kHz
642             EPwm4Regs.TBCTR                 = 0;
643             EPwm4Regs.TBCTL.bit.CLKDIV      = 0;
644             EPwm4Regs.TBCTL.bit.HSPCLKDIV = 0;
645 //            EPwm4Regs.TBCTL.bit.CTRMODE    = 0;
646             EPwm4Regs.CMPA.half.CMPA        = Tmr4_CMPA;
647          //    EPwm3Regs.CMPB                  = 39999;
```

```
648         //    EPwm3Regs.AQCTLA.bit.CBU       = 2;
                // When CTR = CMPB; then make the GPIO_EPWM3 =
                High;
649             EPwm4Regs.AQCTLA.bit.CAU       = 1;                //
                When CTR = CMPA; then make the GPIO_EMPWM3_S1 =
                LOW;
650             EPwm4Regs.AQCTLA.bit.PRD       = 2;                //
                When CTR = PRD; then make the GPIO_EPWM3_S1 =
                HIGH;
651             EPwm4Regs.AQCTLA.bit.ZRO       = 2;                //
                When CTR = ZRO; then make the GPIO_EPWM3_S1 =
                HIGH;
652
653             EPwm4Regs.AQCTLB.bit.CAU       = 2;                //
                When CTR = CMPA; then make the GPIO_EMPWM3_S2 =
                HIGH;
654             EPwm4Regs.AQCTLB.bit.PRD       = 1;                //
                When CTR = PRD; then make the GPIO_EPWM3_S2 =
                LOW;
655             EPwm4Regs.AQCTLB.bit.ZRO       = 1;                //
                When CTR = ZRO; then make the GPIO_EPWM3_S2 =
                LOW;
656
657             EPwm4Regs.DBCTL.bit.HALFCYCLE = 0;
658             EPwm4Regs.DBCTL.bit.IN_MODE    = 0;                //
                EPWMA is the trigger for both rising and
                falling edges
659             EPwm4Regs.DBCTL.bit.POLSEL     = 2;                //
                If we are using dead band then please refer to
                fig 3-29 for understanding
660             EPwm4Regs.DBCTL.bit.OUT_MODE   = 3;                //
                Dead band is applied both for falling and
                rising edges
661
662             EPwm4Regs.DBRED                = 10;               //
                Dead band time delay for rising edge = 10
                (12.5ns*10)
663             EPwm4Regs.DBFED                = 10;               //
                Dead band time delay for falling edge = 10
                (12.5ns*10)
664
665             //** Enabling Interrupts
666
667             EPwm4Regs.ETSEL.bit.INTEN      = 1;
668             EPwm4Regs.ETSEL.bit.INTSEL     = 2;
669             EPwm4Regs.ETPS.bit.INTPRD      = 1;
670
671
672     return;
673 }
```

```
674
675 void Epwm5_Config(void){
676
677     // *** Defining EPwm3 for PWM generation for inverter
678             EPwm5Regs.TBPRD                 = Tmr5_PRD; // For
                frequency of 2kHz
679             EPwm5Regs.TBCTR                 = 0;
680             EPwm5Regs.TBCTL.bit.CLKDIV      = 0;
681             EPwm5Regs.TBCTL.bit.HSPCLKDIV = 0;
682 //            EPwm5Regs.TBCTL.bit.CTRMODE    = 0;
683             EPwm5Regs.CMPA.half.CMPA        = Tmr5_CMPA;
684         //    EPwm3Regs.CMPB                  = 39999;
685         //    EPwm3Regs.AQCTLA.bit.CBU        = 2;
              // When CTR = CMPB; then make the GPIO_EPWM3 =
              High;
686             EPwm5Regs.AQCTLA.bit.CAU        = 1;                //
                When CTR = CMPA; then make the GPIO_EMPWM3_S1 =
                LOW;
687             EPwm5Regs.AQCTLA.bit.PRD        = 2;                //
                When CTR = PRD; then make the GPIO_EPWM3_S1 =
                HIGH;
688             EPwm5Regs.AQCTLA.bit.ZRO        = 2;                //
                When CTR = ZRO; then make the GPIO_EPWM3_S1 =
                HIGH;
689
690             EPwm5Regs.AQCTLB.bit.CAU        = 2;                //
                When CTR = CMPA; then make the GPIO_EMPWM3_S2 =
                HIGH;
691             EPwm5Regs.AQCTLB.bit.PRD        = 1;                //
                When CTR = PRD; then make the GPIO_EPWM3_S2 =
                LOW;
692             EPwm5Regs.AQCTLB.bit.ZRO        = 1;                //
                When CTR = ZRO; then make the GPIO_EPWM3_S2 =
                LOW;
693
694             EPwm5Regs.DBCTL.bit.HALFCYCLE = 0;
695             EPwm5Regs.DBCTL.bit.IN_MODE    = 0;                 //
                EPWMA is the trigger for both rising and
                falling edges
696             EPwm5Regs.DBCTL.bit.POLSEL     = 2;                 //
                If we are using dead band then please refer to
                fig 3-29 for understanding
697             EPwm5Regs.DBCTL.bit.OUT_MODE  = 3;                  //
                Dead band is applied both for falling and
                rising edges
698
699             EPwm5Regs.DBRED                 = 10;               //
                Dead band time delay for rising edge = 10
                (12.5ns*10)
700             EPwm5Regs.DBFED                 = 10;               //
```

```
                Dead band time delay for falling edge = 10
                (12.5ns*10)
701
702             //** Enabling Interrupts
703
704             EPwm5Regs.ETSEL.bit.INTEN       = 1;
705             EPwm5Regs.ETSEL.bit.INTSEL      = 2;
706             EPwm5Regs.ETPS.bit.INTPRD       = 1;
707
708
709     return;
710 }
711 // To Generate control signal pulsating waveform; Look up
    table in used.
712 static void sig_lookup_fn(void){
713
714     for(j = 0; j < PWM_Cycles; j++){
715         look_up[j] = (double)sin(angle_rad);
716
717         signal_look_up[j] = fabs(look_up[j]);
718         angle_degree = angle_degree + angle_spacing;
719
720         angle_rad = ((double)3.14*angle_degree/180);
721
722     }
723
724
725 }
```

dVOC.c

Bibliography

[1] Daly, H., 2018. 1.1 Billion People Still Lack Electricity. This Could Be The Solution. [online] World Economic Forum. Available at: https://www.weforum.org/agenda/2018/06/1-billion-people-lack-electricity-solution-mini-grid-iea/

[2] Eia.gov. 2018. Renewable Energy Explained - U.S. Energy Information Administration (EIA). [online] Available at: <https://www.eia.gov/energyexplained/renewable-sources/>

[3] Climate Action - European Commission. (2018). 2020 climate & energy package - Climate Action - European Commission. [online] Available at: https://ec.europa.eu

[4] Nrc.gov. 2003. The Economic Impacts Of The August 2003 Blackout. [online] Available at: https://www.nrc.gov/docs/ML1113/ML111300584.pdf

[5] Graber, G. An Interconnected And Integrated Electricity System Of The Future. [online] Research Gate. [Image] Available at: <https://www.researchgate.net/figure/An-interconnected-and-integrated-electricity-system-of-the-future-source-Infineon_fig19_303629402>

[6] Energy.gov. 2007. Grid Modernization And The Smart Grid. [online] Available at: <https://www.energy.gov/oe/activities/technology-development/grid-modernization-and-smart-grid>

[7] Azli, Naziha & Md, Shahrin & Mortezaei, Ali. (2011). Design of a current mode PI controller for a single-phase PWM inverter. 10.1109/IAPEC.2011.5779864.

[8] Williams, D., 2017. Understanding, Calculating, And Measuring Total Harmonic Distortion (THD) - Technical Articles. [online] Allaboutcircuits.com. Available

at: <https://www.allaboutcircuits.com/technical-articles/the-importance-of-total-harmonic-distortion/>

[9] Mather, B., n.d. Virtual Oscillator Controls | Grid Modernization | NREL. [online] Nrel.gov. [Image] Available at: <https://www.nrel.gov/grid/virtual-oscillator-controls.html>

[10] Tielens, P.; Hertem, D.V. The relevance of inertia in power systems. Renew. Sustain. Energy Rev. 2016,55, 999-1009.

[11] Pal Verma, Y., 2017. Inertial Response From Renewable Energy Sources. [online] idr.nitkkr.ac.in. Available at: <http://idr.nitkkr.ac.in>

[12] Kundur, P.; Balu, N.J.; Lauby, M.G. Power System Stability and Control; McGraw-Hill: New York, NY, USA, 1994

[13] Zhong, Q.C. Virtual Synchronous Machines: A unified interface for grid integration. IEEE Power Electron. Mag. 2016, 3, 18-27.

[14] Sakimoto, K.; Miura, Y.; Ise, T. Stabilization of a power system with a distributed generator by a Virtual Synchronous Generator function. In Proceedings of the 8th International Conference on Power Electronics (ECCE Asia), Jeju, Korea, 30 May-3 June 2011; pp. 1498-1505.

[15] Liu, J.; Miura, Y.; Ise, T. Dynamic characteristics and stability comparisons between virtual synchronous generator and droop control in inverter-based distributed generators. In Proceedings of the International Power Electronics Conference (IPEC-Hiroshima 2014-ECCE ASIA), Hiroshima, Japan, 18-21 May 2014; pp. 1536-1543.

[16] Sakimoto, K.; Miura, Y.; Ise, T. Characteristics of Parallel Operation of Inverter-Type Distributed Generators Operated by a Virtual Synchronous Generator. Electr. Eng. Jpn. 2015, 192, 9-19.

[17] Guerrero, J.M.; de Vicuna, L.G.; Matas, J.; Castilla, M.; Miret, J. A wireless controller to enhance dynamic performance of parallel inverters in distributed generation systems. IEEE Trans. Power Electron. 2004, 19, 1205-1213.

[18] Sen., W., 2015. S.1232 - 114Th Congress (2015-2016): Smart Grid Act Of 2015. [online] Congress.gov. Available at: <https://www.congress.gov/bill/114th-congress/senate-bill/1232>.

[19] D. Pattabiraman, R. H. Lasseter. and T. M. Jahns, "Comparison of Grid Following and Grid Forming Control for a High Inverter Penetration Power System," 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, OR, 2018, pp. 1-5.

[20] Microsemi.com. 2013. Park, Inverse Park And Clarke, Inverse Clarke Transformations. [online] Available at: <https://www.microsemi.com/document-portal/doc_view/132799-park-inverse-park-and-clarke-inverse-clarke-transformations-mss-software-implementation-user-guide>.

[21] D. Gro, M. Colombino, J.-S. Brouillon, and F. Dorfler, "The effect of transmission-line dynamics on grid-forming dispatchable virtual oscillator control," arXiv preprint arXiv:1802.08881, 2018.

[22] M. Colombino, D. Gro, J.-S. Brouillon, and F. Dorfler, "Global phase and magnitude synchronization of coupled oscillators with application to the control of grid-forming power inverters," Submitted to IEEE Transactions of Automatic Control, avialable online: arXiv:1710.00694, 2018.

[23] B. Johnson, M. Rodriguez, M. Sinha, and S. Dhople, "Comparison of virtual oscillator and droop control," in IEEE Workshop on Control and Modeling for Power Electronics, 2017.

[24] G. Seo, M. Colombino, I. Subotic, B. Johnson, D. Gro, and F. Dorfler, "Dispatchable virtual oscillator control for decentralized inverter-dominated power systems: Analysis and experiments," 2019, arXiv:1811.08842

[25] Electricity Reliability Council of Texas (ERCOT). Future Ancillary Services in ERCOT; ERCOT: Taylor, TX, USA, 2013. Available online: http://www.ercot.com/content/news/presentations/2014/ERCOT_AS_Concept_Paper_Version_1.1_as_of_11-01-13_1445_black.pdf.

[26] Poolla, B.K.; Bolognani, S.; Dorfler, F. Optimal placement of virtual inertia in power grids. In Proceedings of the American Control Conference, Boston, MA, USA, 6-8 July 2016.

[27] Kahlane A. E.W.H., Hassaine L and Kherchi M, "LCL Filter Design for Photovoltaic Grid Connected Systems", the Journal of Renewable Energies pp. 227-232.

[28] Egr.unlv.edu. 2013. EE 740 - Transmission Lines. [online] Available at: <http://www.egr.unlv.edu/≈eebag/TRANSMISSION

[29] H. M. Kojabadi et al., "A Novel DSP-based currentcontrolled PWM strategy for single phase grid connected inverters," IEEE Transactions on Power Electronics, vol. 21, no. 4, pp. 985-993, July 2006

[30] Hojabri, Mojgan & Hojabri, Mehrdad & Toudeshki, Arash. (2015). PASSIVE DAMPING FILTER DESIGN AND APPLICATION FOR THREE-PHASE PV GRID-CONNECTED INVERTER. International Journal of Electrical, Electronics and Data Communication (IJEEDC). 3. 2320-2084.

[31] H. Kim and S.K Sul, "A novel filter design for output LC filters of PWM inverters," Journal of Power Electronics., vol. 11, no.1, pp. 74-81,2011.

Bibliography

[32] Z.Ye, G. Sinha, X.Yuan and T. C.Wang, Output filter design for a grid-interconnected three-phase inverter, IEEE Annual Conf, 2003.

[33] X. Renzhong, X. Lie, Z. Junjun, and D. Jie .,Design and research on the LCL filter in three-phase PV grid-connected inverters, International Journal of Computer and Electrical Engineering, vol. 5, no. 3, June 2013.

[34] Ni.com. 2020. What Is Hardware-In-The-Loop? - National Instruments. [online] Available at: https://www.ni.com/en-us/innovations/white-papers/17/what-is-hardware-in-the-loop-.html

[35] Speedgoat.com. n.d. Motor Control And Power Electronics | Speedgoat. [online] Available at: <https://www.speedgoat.com/applications-industries/industries/power-electronics>

[36] Duman, Erkan. (2008). Evaluating of a Fuzzy Chip by Hardware-in-the-Loop (HIL) Simulation. International Review on Computers and Software. 3.

[37] ti.com. n.d. TMS320F28069. [online] Available at: <https://www.ti.com/product/TMS320F28069>

[38] Typhoon-hil.com. n.d. Typhoon HIL DSP Interface. [online] Available at: <https://www.typhoon-hil.com/doc/brochures/HIL-DSP-100-Interface.pdf>

[39] Almeida, P.R.; Soares, F.; Lopes, J.P. Electric vehicles contribution for frequency control with inertial emulation. Electr. Power Syst. Res. 2015, 127, 141-150.

[40] . Meng, J.; Mu, Y.; Wu, J.; Jia, H.; Dai, Q.; Yu, X. Dynamic frequency response from electric vehicles in the Great Britain power system. J. Mod. Power Syst. Clean Energy 2015, 3, 203-211.

[41] Zhang, Y.; Bank, J.; Wan, Y.H.; Muljadi, E.; Corbus, D. Synchrophasor Measurement-Based Wind Plant Inertia Estimation. In Proceedings of the IEEE Green Technologies Conference (GreenTech), Denver, CO, USA, 4-5 April 2013; pp. 494-499.