PortraitColourPos-eps-converted-to.pdf

SCHOOL OF ELECTRICAL ENGINEERING
AND TELECOMMUNICATIONS

A Single Stage Grid-connected PV System

by

Zhenning JIANG

Student ID: z5082223

Thesis submitted as a requirement for the degree

Master of Engineering (Electrical Engineering)

Submitted: May 30, 2017
Supervisor: Georgios Konstantinou

Abstract

The review of current design of single phase grid-connected photovoltaic system, initial model of the system and the concept of hardware-in-the-loop simulation are presented in this paper. A model of the system including solar cell model, MPPT controller, converter controller are developed both in Matlab and RT-LAB. Some initial results demonstrate the overall system performance.

Acknowledgements

I wish to thank all the people who have made contributions to the contents of this document; in particular Toan Phung and Julien Epps.

Abbreviations

BE Bachelor of Engineering

EE&T School of Electrical Engineering and Telecommunications

SSGPVC Single Stage Grid-connected PV Converter

SSGPVS Single Stage Grid-connected PV System

DSP digital signal processor

AC Alternating Current

DC Direct Current

PV Photovoltaic

PVS Photovoltaic System

MPPT maximum power point tracking

HIL hardware-in-the-loop

SIL software-in-the-loop

ADC analog to digital converter

DAC digital to analog converter

SOGI second order generalized integrator

GPIO general purpose input output

CPU central processing unit

FPGA field-programmable gate array

IC incremental conductance

PO perturb and observe

PLL phase-locked loop

PI proportional integral

RT real-time

Contents

Chapter 1

Introduction

Human activities reply on intensive use of energy generated from various source. Primarily, energy can further divided into two groups, renewable and non-renewable. Renewable energy which has drawn more and more attention in recent years, is energy supplied from renewable source. Renewable energy sources including solar, hydro, wind, biomass, tidal wave, ocean thermal and the like. They may be replenished in nature on human timescale, which mean there is little chance that the energy could be depleted. The four major areas which rely on renewable energy are: electricity production, heating/cooling, transportation, and rural energy services.

Solar power as one of the most important renewable energy is generated when energy from the sun (sunlight). Energy gather from sunlight is converted into electricity or used to heat air, water, or other fluids. There are two mainstream technology to harvest solar energy and put it into good use. The first technology is solar thermal which thermal energy is resulted from concentration of solar radiation. Air, water, or other fluid is widely used as carrier to carry harvested thermal energy which usually can be used directly for heating up space or generating electricity by means of steam and turbine. Solar thermal is commonly used in residential area to provide hot water for living. Solar thermal electricity, also known as concentrating solar power, is typically designed for large scale power generation. The Ivanpah Solar Electric Generating System is a good example of concentrated solar thermal plant in the Mojave Desert. Solar Photovoltaic (PV) converts sunlight directly into electricity using photovoltaic cells by means of photovoltaic effect. Photovoltaic System (PVS) can be mounted on rooftops, integrated into building designs and vehicles, or scaled up to multi-gigawatt scale power plants. PVS can also be used together with concentrating mirrors or lenses for large scale centralized power. It is

also not rare to combine solar thermal and PV technology into a single system which provide heat as well as electricity at the same time.

With great ambitions to create a green future for human beings and pressure to reduce production of greenhouse gas, many governments of developed and developing countries encourage people to increase the usage of renewable energy. The installation of PV have been increasing within recent decade. It is safe to say, the total number of installation would keep increasing for a long time before human being can find some better solutions to solve the thirst of reliable and clean energy.

Nowadays, majority of household electrical appliances rely on alternating current (AC), however, solar PV is only capable of generating direct current (DC) current or voltage. As a result, in order to make the most the solar power, various types of DC to AC converters are needed to meet different kinds of requirements for different applications. Microinverter, one of the most popular converters which features simplicity and great efficiency for solar applications has become more and more prevalent than ever. Although suffers from slightly higher upfront cost compared with PV strings with centralized inverter solutions. The market calls for cheaper and smaller microinverters in order to help integrate solar panels together with inverter as a single solution which would make building solar PVS more convenient and much easier.

The focus of this project is the development of a single stage grid-connected PV system (SSGPVS) which might be useful in real life. The development may be finished using simulation and no real circuit prototype is required due to limitation on time and budget. However, in order to make the project as close to real life product development as possible, hardware-in-the-loop (HIL) is used which enable the possibility to divide the system into two parts, simulated and actual part. The controller of the system would be implemented using a digital signal controller(DSC) or a microcontroller which is the real hardware in the system and the rest of the systme including PV panels, switching elements and grid are simulated using a Real-Time (RT) simulator. The simulator which is developed by Opal-RT is able to simulate any amount of time and generate the results using the same amount of time in real life while maintains a small enough time resolution. The closing stage of this project, I have managed to make....

Chapter 2 explains the background for this document. Chapter **??** states the style and submission related requirements to theses submitted at the school. Chapter **??** explains content related requirements to theses and how to avoid some commonly seen problems. Chapter 5 evaluates the thesis requirements template. Finally, Chapter 6 draws up conclusions and sug-

gests ways to further improve the thesis requirements template.

Chapter 2

Background

This chapter present more detailed introduction to solar PV and PVS in section 2.1. The following is context of HIL and its applications and advantages in section 2.2.

2.1 General Introduction

PVS is type of power system which convert solar power into electricity by means of photovoltaic effect. PVS have many advantages and become more and more popular nowadays compared with traditional fossil fuel based energy generation method. It has became a vital part of renewable energy or green energy, since it generate zero green house gas during operation. The module or panel price have experienced huge decline since middle of 2008[2], which makes the the whole PV industry prospered. According to the literature, PV cell is able to become an important alternate renewable energy source till 2040[12].

Power electronic is a vital part for solar power generation. Typically, a single stage grid-connected PV converter (SSGPVC) consist of a solar panel which absorb and convert sunlight into electricity and one or more stages of power converters that transform the electricity from DC into AC, as well as other accessories for mounting and connecting different components.
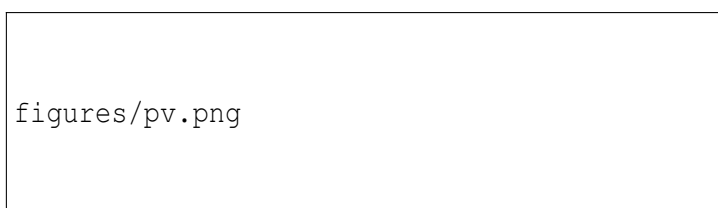
figures/pv.png

Figure 2.1: Typical PV system block diagram

9

Figure 2.1 on page 9 indicate a general configuration of PVS. Due to the output characteristic of a typical solar cell and the need to inject power into the grid, there are at least two stages in a grid-connected PVS conventionally. The use of optional DC/DC convert(s) is to extract maximum power from upstream solar cells or units, of which is also known as maximum power point tracking (MPPT). Additionally, it is able to provide required voltage for next stage inverter to work with a appropriate duty cycle. The functionality of DC/AC converter is to inject sinusoidal current whose phase may or may not be synchronized with grid voltage(depends on applications) into grid.

Currently, there are various configurations for grid-connected PVS including centralized structure, the string structure, the multi-string structure and ac-module structure.[23]. Especially, multi-string and ac-module structure have been used in a broad range of applications. Although multi-string configuration have many advantages like reducing loss due to solar panel unbalanced and a centralized MPPT over centralized configuration, it also suffers from lacking redundancy and scalability. With optional DC/DC converter, the overall system efficiency is compromised compared with a SSGPVC.

As a result, ac-module configuration which is based on SSGPVC is gaining more and more attention. The ac-module configuration is consist of only one power electronics devices which can integrate MPPT function and convert DC into AC at the same time. Current design of PVS requires for simplicity, modularity and efficiency. Those requirements are exactly the advantages of the single stage system. However, there are some issues related to the grid-connected system. The sinusoid power injected into the grid caused by the fluctuation in the dc link is one of the issue.[6] It can force the MPPT function to be voided so that the system can not operate at the maximum power point. Due to complexity of such converter and power rating limitations, SSGPVC is often used in residential PV system.

Although the SSGPVC has some limitations, it is worth to improve the system performance and make it a better solution. Its 'apply and plug' characteristic can be consider to be a general solution to various grid-connected PV applications.

2.2 Hardware-in-the-loop Simulation

Nowadays, the complexity of various digital algorithms for power electronic applications, circuit topology is rising incredibly, which makes the simulation and development of power elec-

tronics related product increasingly time-consuming.[20] In order to fulfill the need for fast prototyping, verified the controller functionalities without implementation of hardware, and avoid hazardous situation while testing the controller on a real hardware, the HIL simulation or more specific PIL simulation is deployed and used intensively for such applications.

High-performance embedded system enabled a power electronic revolution. The increasing demand for digital controlled power electronics system requires the development to verify the controller functionalities before the controllers are tested on a real hardware platform. PIL allow developers test and verify processors' performance on a simulator which host the virtual hardware, so it become a crucial part of developing a digital controlled power electronics system.

Compared with traditional software-in-the-loop (SIL) simulation, HIL and PIL simulation is another level of simulation which runs way faster than the SIL simulation. In conventional SIL simulation for power electronics application, the time it takes to run the simulation is often longer than the simulation time span of interest, which means, for example, people need to speed more than one second in order to generate one second of results. On the contrary, the HIL and PIL simulation run in real-time without sacrificing simulation time step resolution. If one second is spent on the simulation process, results for one second can be generated with corresponding resolution that preserves all the transient state of prototype under simulation.

Comparing HIL simulation with PIL simulation, the PIL simulation offers a more close-to-real results since in many applications the functionalities needed is hard to abstract and model using software. In PIL simulation, the controller functionalities are executed inside by a real controller, so the developers do not need to worry about how to implement a precise controller model inside software environment.

Fig.2.2 illustrate a basic setup for PIL simulation. The power converter operate inside the

figures/pil.png

Figure 2.2: The basic PIL system

11

RT computer. There are multiple ways that the real controller can communicate with the RT simulator. The simulator is capable of generating necessary voltage or current signal through digital to analog converter (DAC) so that the external controller can sense the signal and generate proper trig signal feeding back to the RT computer. The RT computer also have general purpose input output (GPIO) to read all the controller generated signal and carry them into runtime simulation. Another way mentioned in this papar[20] is to use direct file exchange and there is a solution support this method with any type of controller architecture.

For the sake of running PIL simulation, a power converter testbed needed to be implement inside the software environment. The RT computer has equipped with a powerful central processing unit (CPU) as well as field-programmable gate array (FPGA) acceleration card. For the sake of fully utilize the computation power of the simulator, the model should be construct complying to some rules and regulations. But converting an existing converter model into RT simulation compatible model requires a bit of work and experience which is described in detailed in Chapter 4 on page 23.

2.3   Previous work

PIL simulation has been used for PV application. In paper,[10] a two stages PVS was implemented using RT-LAB which is a software for controlling the RT simulator. In this paper, the simulation setup compose of a solar panel model, a grid-connected inverter and a DC/DC converter in the software environment. Also, there is a physical controller controlling DC/DC converter so that the author could perform PIL simulation. This paper proves that it is feasible to use PIL for PVS simulation and gave a detailed example of how to implement RT simulation model using the corresponding software. The major improvement to the results of the paper would improving the pulse width modulation (PWM) frequency of the controller since the state-of-art digital controlled converter's PWM can be as high as a few mega hertz while in the paper the PWM frequency is only 10 kHz. Also, making use of the FPGA acceleration card in the RT simulator would be beneficial to provide finer time resolutions and enable higher PWM processing capabilities.

Many researchers have already completed lots of research work regarding SSGPVC as well as PV panels as a whole system. In particular, Paper [12] presents approaches on modeling of single stage PVS for RT simulation. This paper focus on discrete time modeling of the whole

system, while at the same time it proposed a new MPPT method which is based on incremental conductance (IC) and had the new algorithm tested using RT simulation. In this paper, the author had the RT simulator investigated and concluded that the RT simulators have sufficient computation power to perform simulation for PV applications.

Chapter 3

A Single Stage PV System

3.1   The System

The system is shown in Figure 3.1 on the following page and can be divided into three major parts, solar panel/cells, H-bridge, controller and LCL filter. The transformer is optional depending on local rules and regulations. In some countries, it is compulsory to add the transformer for the purpose of isolation and protection, while in some countries it is not required. It is allowed to have transformerless system in Australia according to regulations[15]. The whole system design is fairly simple and straightforward. Detailed description on solar cell model can be found in section 3.2.

3.2   Modeling of Solar Cell

There are lots of solar cell models existed so far and the most widely used model is presented here. The equivalent circuit of a single solar cell is shown in Figure 3.2 on page 16. It is composed of a photo-generated current source, a parallel connected diode, a shunt resistor and a series connected resister[22]. In present of parasitic resistors, the solar cell output characteristic can be represent be the following equations[3][21].

$$I = I_L - I_o(e^{\frac{V+IR_2}{(\frac{nkT}{q})}} - 1) - \frac{V+IR_2}{R_1} \tag{3.1}$$

In this equation, I is the output current which flows out of the solar cell for positive value. It is possible produce negative current which is equivalent to current flows into cells, however, this situation should be avoided because . V is the terminal voltage of the cell. I_o is the parallel dido

saturation current. I_L is the light generated current. n is the number of cells in series. q is the elementary charge. T is the current ambient temperature of the cell. Equation (3.1) indicates that when the terminal voltage is low, almost all the current can flow out of the cell. When the voltage goes higher, the saturation current of the parallel diode begin to increase, thus reducing the output current. Eventually the output current can go to zero as raise of terminal voltage.

Temperature can affect the output characteristic of the solar cell as well, since under different temperature condition, electron have different electron mobility. Generally, the higher the temperature, the higher is saturation current of parallel connected diode, thus leading to decreasing of total output power. Figure 3.4 on page 17 illustrate a solar panel output characteristic with respect to various temperature.

The solar panel on the market usually consist of multiple solar cells connected in series or in parallel to increase the terminal open circuit voltage or maximum short circuit current. A typical output characteristic of a solar panel module is shown in Figure 3.2 on the next page. Due to the output characteristic of solar cell, there exist a maximum power point which have been marked in the figure with a red circle. If we want to extract the most power out of the panel or cells, MPPT algorithm is needed to archive this goal.
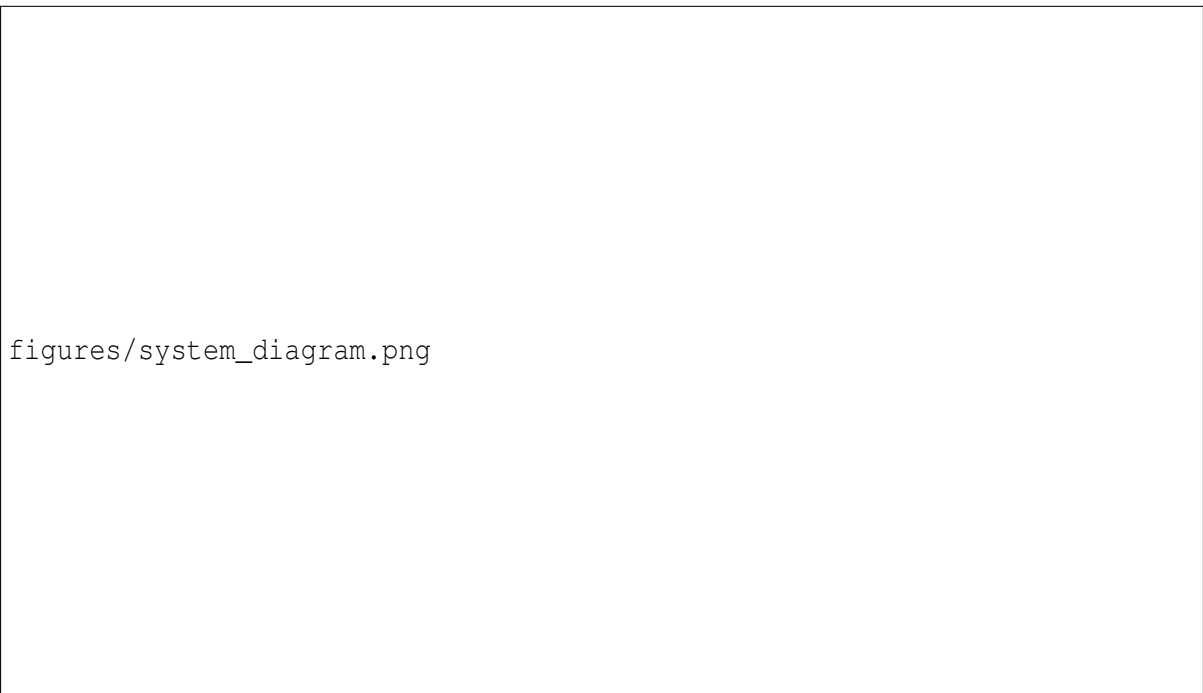

figures/system_diagram.png

Figure 3.1: A single stage grid-connected PV system

3.2.1 MPPT

As mentioned before, MPPT is a method for extracting the most power out of solar cell. MPPT have been studied by many researchers many algorithms have been proposed, such as perturb and observe (PO), IC[4][7][11][13][24].

PO is one of the most classic and simple algorithm and have been widely used in many PVS. The method actively adjust the output voltage of the connected solar panel by a small amount ΔV and sense the ouput current to calculate the resulted change in output power ΔP by substrate current power with previous power. If ΔP is greater than zero, it means that output voltage may be set higher to gain more power. If ΔP is less than zero, it means that MPP has been passed and output voltage needed to be reduce. The power trajectory moves like climbing up a "mountain" which is the maximum power archivable under current sunlight condition. the whole algorithm can be described in Figure 3.6 on the following page.

The problem with PO method is the adjustment of power output based on perturb. The algorithm tries hard to get close to the maximum power point instead of staying at the maximum power point, of which lead to compromise of system efficiency. The system may operate around MPP depending on the size of perturb. Large perturb may lead to fast response when irradiance is changing dynamically but largely compromised steady state efficiency. Small perturb may result in high steady state efficiency, however, the algorithm may have little transient performance.
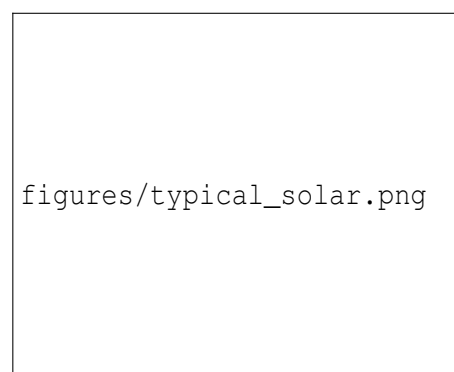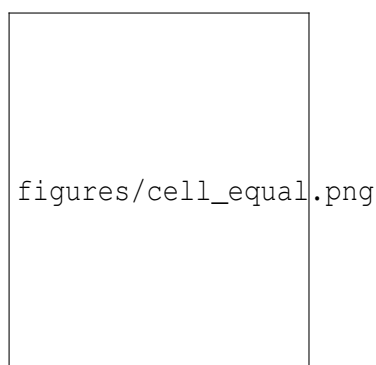

figures/cell_equal.png

figures/typical_solar.png

Figure 3.2: Equivalent circuit of single solar cell

Figure 3.3: Output characteristic with different sunlight

figures/typical_solar_temp.png
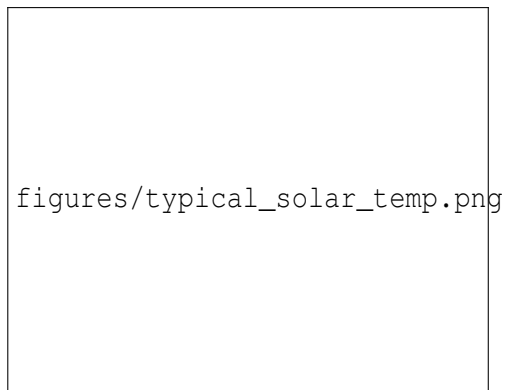
figures/topology.png

Figure 3.4: Output characteristic with different temperature

Figure 3.5: H-bridge with LCL filter

3.2.2 H-bridge and PWM

There are lots of different types of circuit topologies for inverter applications[1][5], but the simplest one is classical H-bridge configuration. Figure 3.5 shows a simple H-bridge with LCL
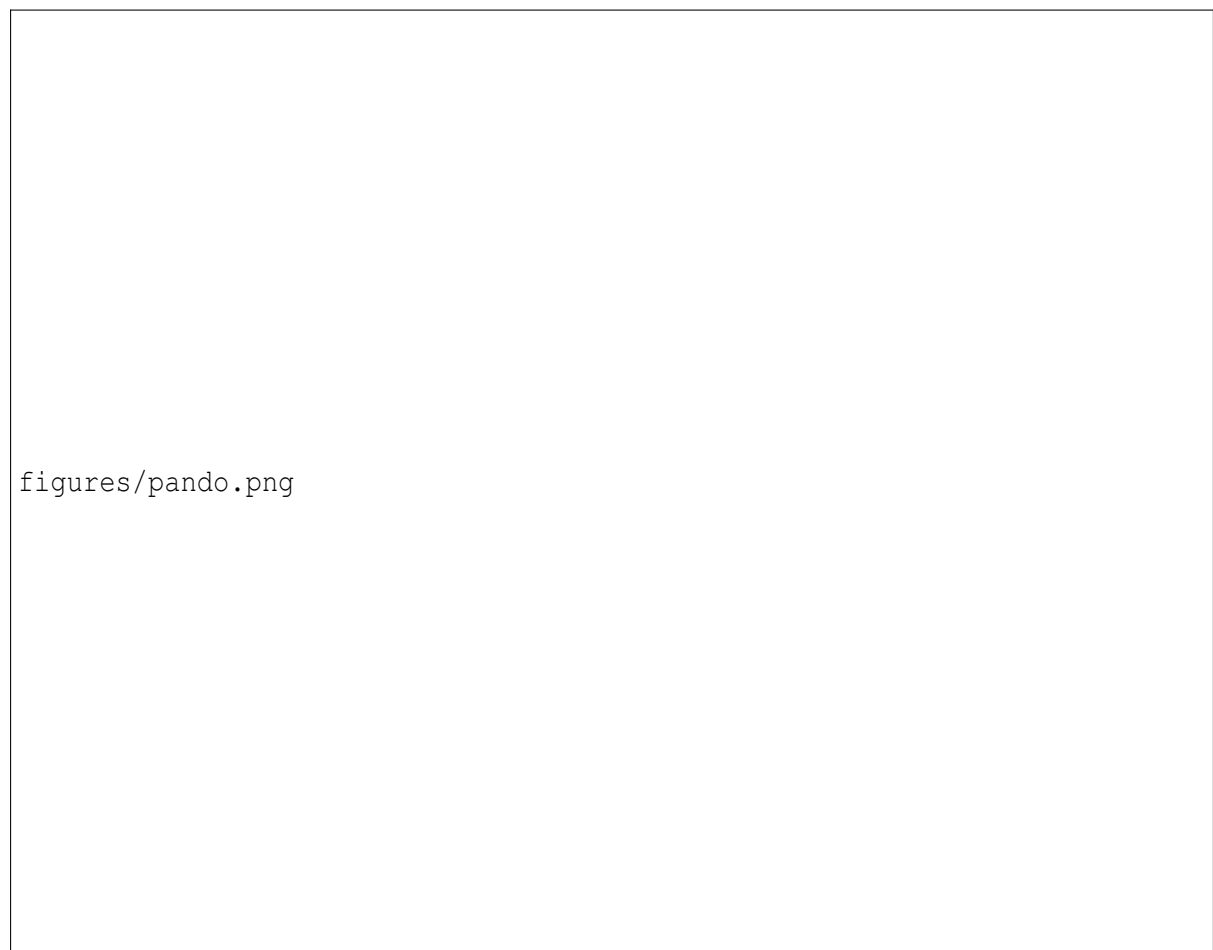
figures/pando.png

Figure 3.6: P&O Algorithm

filter configuration which would be used in the later simulation.

H-bridge, in this case, has four switching components which are represent by the symbol of MOSFET. By turning on and off the switching, three different levels of voltage can be sense from output terminals which are $\pm V_{dc}$ and zero. The state of the switches are controlled by desired PWM signals to generate desired voltage levels at between switching intervals.

PWM is particular important for success of converting DC to AC. PWM is a modulation technique to encode signal or signals into a pulse train. There exists two different types of modulation technique for a classic four switches H-bridge, bipolar and unipolar modulation for solar PV applications. The difference between the two is that in bipolar modulation, the output voltage can only be either $\pm V_{dc}$ compared with unipolar modulation. With one additional voltage level, unipolar modulation can produce higher efficiency and lower TOTAL Harmonic Distortion(THD).

One issue related with PVS is common mode voltage which usually cannot be found in other applications of DC to AC inverters. Since usually PV panels are mounted on large plates of mental, voltage potential difference, also known as common mode voltage between panels and ground, can build up due to capacitor effect leading to malfunctioning of circuit breaker at the grid side. One way to get around this issue is to use bipolar PWM if a H-bridge configuration is used. The reason why unipolar may suffer the issue is that when all the switches are turned off, the output terminals are flouting with respect to ground, thus lead to common mode voltage building up at the terminals. When the switches are turn on, closed circuit loop can be create with protection devices on grid side connected to earth ground. Anther way to get around the issue is to use the different inverter topologies to isolate panels and converter. One good example of such configuration proposed by this paper[5].

There is a limitation on the SSGPVC which is the voltage rating. The voltage on the input side need to be higher than the output voltage which means the overall voltage gain of H-bridge is always less than one. Interfacing with high voltage grid would results in large number of PV panels or cells connected in series on the input side to build up the voltage.

3.3   LCL Filter

As shown in Figure 3.5 on page 17, the circuit make use of a filter which is consist of two inductors and one capacitor. High frequency modulation signal requires a proper filter to eliminate
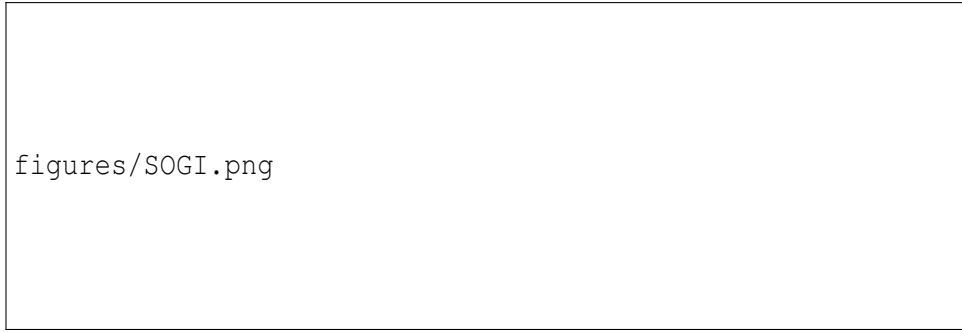
Figure 3.7: Basic structure of SOGI

any undesired high frequency harmonics to fulfill the requirement imposed by THD. The filter is a high orders passive filter which has superior performance compared with traditional single inductor filter in terms of harmonic elimination. However, high order filters are more difficult to control, it's resonance frequency can cause stability issue if not carefully deal with[8].

3.3.1 Grid Synchronization

AC voltage generated by the inverter should have identical phase with the grid voltage so that there is not cancellation of voltage which might cause instability of the grid. In order to archive synchronization with the grid, phase-locked loop (PLL) is used. The PLL served as a servo system controlling the signal phase of its output. The performance of it has a direct impact on control loop of grid-connected power applications. It should be able to reject source of errors including voltage unbalance, line notching, line dip, frequency jump. The most critical thing is to keep maintaining a clean and reliable phase lock to grid voltage. There are lots of PLL structures designed for various applications.

Among those PLL structures, second order generalized integrator (SOGI) have been widely discussed and used in many PVS. It can be selectively tuned to eliminate other frequency components other than grid frequency and its nominal frequency would not be affected by the frequency change[9]. The basic structure of SOGI is shown in Figure 3.7.

3.3.2 Feedback Control

$$T(s) = K_p + \frac{K_i}{s} \tag{3.2}$$

$$T(s) = K_p + \frac{K_i}{s^2 + \omega_1^2} \tag{3.3}$$

19

$$C(s) = \sum_{n=3,5,7}^{k} \frac{K_n}{s^2 + \omega_n} \qquad (3.4)$$

For current control of SSGPVC, proportional integral controller (PIC) and proportional-resonant controller (PRC) are common choices. The transfer function of a typical PIC is illustrated in Equation (3.2). In a negtive feedback system, PIC can be used to eliminate error between target and output value. The proportional term may help reduce the error quickly and the integral term may be used to eliminate steady state error. It features simplicity as well as reliability and has been used widely in many control engineering applications. Since PIC is a first order system, the performance to track a sinusoidal target is actually not very great and the error cannot be eliminated. It also suffers poor high frequency components rejection performance.

Equation (3.3) represents PRC where ω_1 is the fundamental frequency of the grid in radiance. ω_1 is either $50 \times 2\pi = 314.16$ rad/s for 50 Hz system or 377 rad/s for 60 Hz system. Equation (3.4) represents harmonic compensators for output current where ω_n are harmonics. Typically in a PVS, both PRC and compensator are used in conjunction to archive better performance. One good aspect offered by the compensator is that harmonics can be selectively eliminated by varying ω_n where $\omega_n = n \times \omega_1$ and $n = 3, 5, 7 \ldots$. The downside of the compensator is that it may result in high orders system which requires more computation power that some low end microcontrollers may struggle to offer.

3.4 Opal-RT Simulator

Opal-RT is a company based in Canada which provide RT simulators and its software solutions to make RT simulation possible. In this project, their simulator model OP4500 which is shown
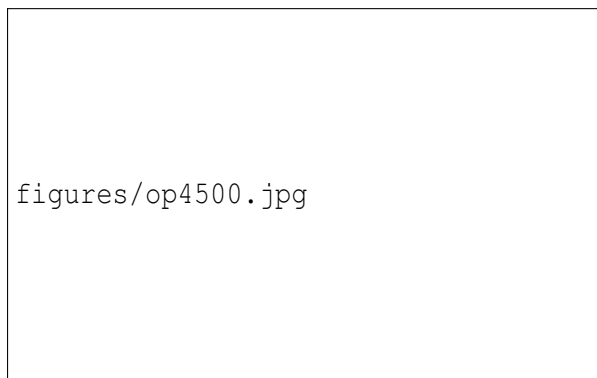

figures/op4500.jpg

Figure 3.8: OP4500 RT simulator

in Figure 3.8 on page 20 is used. OP4500 combine traditional X86 architecture and latest FPGA simulation technique to accelerate the process. The onboard Intel CPU can archive minimum time step of 7 μs without getting overrun and **eHS** FPGA based simulation solver is able to archive minimum time steps of 250 ns. It also multiple Input and Output(IO) modules including both analog and digital types, for interfacing external hardware to perform HIL simulation. The simulartor can be either a controller to run control algorithm and generate desired signal to control actual hardware or become a simulated plant to develop and verify functionalities of external controller.

RTLAB is the software running inside a host personal computer(PC) in order to be able to use the box for simulations. *RTLAB* currently is fully compatible with *MATLAB* and SIMULINK which make it easy to develop models for RT simulation. currently, *RTLAB* only supports up to *MATLAB 2014B* and not all models in SIMULINK are supported.

3.5 Controller Development Kit

Implementation of all controller functionalities is an vital part of this project. As stated in previous chapter, cost need to be considered since upfront cost of modular SSGPVS is still a bit higher at the moment. Microcontroller as the most important part of controller system, could be expensive sometimes depending on performance configurations. Obviously, higher the performance, greater the cost on microcontroller. The controller is implemented in using

Figure 3.9: TMS320F28027 Experiment Kit

a Texas Instrument C2000 microcontroller (TMS320F28027). The experiment controller kit is completely off-the-shelf and it is quite inexpensive to obtain. The volume price for this microcontroller is quite competitive considering the performance it can offer. This is a 32-bit fix-point microcontroller with clock frequency up to 60 Mhz and some necessary peripherals, of

which is designed for microinverter applications. The microcontroller structure block diagram is shown in Figure 3.10.

Figure 3.10: TMS320F28027 structure and on-chip peripherals

Chapter 4

Implementation

4.1 Breaking Down the Problem

Having examined the state of art developments of SSGPVS, what are essentials to successfully make a SSGPVS using RT simulators and microcontroller is pretty clear. Although theory behind SSGPVS has been studied for quite a long time and there has been existing products on the market already, documentations on development process of a RT or HIL simulation are still limited. So far the reviews have shed the light on tasks needed to complete the whole project.

The first task is to develop a SSGPVS in the pure software environment which may pave th way for developing a RT model for the simulator. The process involved is described in detail in Section 4.2. The second issue which is discussed in Section 4.3 on page 25 is to revise the model so that it can be load into RT simulator and runs in RT. The third issue is to implement the controller functionalities using a digital signal processor (DSP). The implementation involves some advanced C language programing which is something I enjoy doing. The details of implementation are presented in Section 4.4 on page 29. The fourth issue which discussed in Section 4.5 on page 34 is to perform PIL simulation. This issue is a little bit complicated since it requires me to be well familiar with the real time simulator. And I have not found much resources on this so far. It is a milestone to this project.

4.2 Building Model in SIMULINK

The purpose of a pure software simulation model is to verify the correctness of circuit topology, solar panel model and proper design of LCL filter. Luckily, SIMULINK has a build-in SSGPVS

model which can be found in *MALTAB 2015B* and later versions[14]. It provide a good starting point of this project. Since the project focus on performing RT simulations, any already known to work model can be used to accelerate the process of building the whole model and save some time avoid doing everything from scratch.

The model was quite well designed and simulation ran with not error. Minor adjustment may be made before turning it into a RT model. The system was design for North America market which result in a 60 Hz operation of grid frequency. Since the grid voltage in Australia is 50 Hz, the grid frequency of original model need to be change to 50 Hz. In order to change the frequency, the minimum time step, PWM frequency of the inverter and controller sampling time need to be changed as well. The reason for changing the minimum time step is that fixed time solver for solving ordinary differential equations(ODE) is used for simulation which means grid frequency, controller sampling time or any other components' sampling time need to be integer multiple of minimum time step. Section 4.2 is a summary of different configurations between models.

Items Changed	Origin Configuration	New Configuration
Minimum time step	$1.323e^{-6}$ s	$6.25e^{-7}$ s
Carrier frequency	3780 Hz	8000 Hz
Controller sampling	$2.6455e^{-5}$ s	$1.25e^{-5}$ s

Noticing that the switching frequency is set to 8 kHz because higher switching frequency may lead to less bulky inductors in real life which is less expensive. Instead of setting switching frequency to nearer 4 kHz, double of the frequency is chosen. In order to ensure the accuracy of simulation and avoid limited resolution on duty cycle, $6.25e^{-7}$ s of minimum time step has been chosen to ensure minimum stepping of duty cycle is 0.5%. The duty cycle resolution could affect SSGPVC creating high output current THD and undesired phase shift since duty cycle can only vary in discrete steps.

Changed made in PWM frequency lead to smaller minimum time step, thus, more time needed to generate one second simulation results. The time it takes to run the simulation depends on configurations of the host PC and it is common to run simulation for around five minutes to generate one second of results.

The controller design in the model is different from what has been mentioned previously. Bearing in mind that the purpose of using this model is to obtain a working plant for performing RT simulation in the next stage, as a result, the design of the controller is not thoroughly studied.

4.3   Convert to Real-time

After getting a working model running in SIMULINK, the model should be convert to *RTLAB* compatible model. Currently, *RTLAB* only support *MATLAB 2014B*, the first thing to do is to make sure the model can still work under *2014B*. Unfortunately, the model makes use of a generic solar PV model which is only rendered in *2015B*. In order to continue doing simulation, the PV model need to be replaced with a replica.

4.3.1   Custom Solar Cell

Since access to source code to *MALTAB*'s PV panel model is not possible, creating a replica of the *MALTAB*'s model is important so that it would be possible to verify the custom solar cell model is acutally correctly implemented. Based on what has been stated in Section 3.2 on page 14, the custom model was initially built up in a script in *MALTAB*. Since scripts or functions written in *MATLAB* is also compatible to run using SIMULINK, it should not be difficult to integrate the script into existing SSGPVS model.

The initial idea was to implement a generic PV cell which only relies on some key parameters that represent the external characteristic of the cell. The parameters include open circuit voltage, short circuit current and temperature coefficients of voltage/current. As for the internal serial and parallel connected resistors, they may be obtained via proper calculation. In this way, the model can be easily extended to simulate any cell or panel of interest. However, how to archive this goal is not clear at the time when the model was implemented since the resistors' value are usually acquired via measurements made by the manufacturers of of the cell/panel. Not many article which is available shed light on how to accurately estimate those parameters from external characteristic of cell/panel, which makes it difficult to fulfill this goal. As a workaround, the model was implemented under the condition that the two parameters are provided by the manufacturer, otherwise there is no way to implement the model.

The first version of the cell model was implemented as a completely standalone model (black box) which does not reply on any SIMULINK models. The first trial was based on [21] since the model it proposed is simple and easy to understand. However, the model in the article is a simplified one which shunt resister is not considered. Also, Newton's method was used to derive serial resistor value of the cell, of which is based on iteration. Newton's method may fail to converging to local minimum point when variables are trapped in saddle points. [3]

has more details on models which have different levels of simplification and equations are well presented. This makes it easier to understand and implement the model. Model presented in [3] were built-in native SIMULINK , however, using 'drag and drop' method has limitation on the maximum efficiency that one can archive. That's the reason that custom model implemented in this project is written as a script and the running results are presented in the later chapter.

It turned out that the idea to make the model as a black box is not a good idea because this approach makes it hard to interface with the rest of the system in SIMULINK . Interestingly, SIMULINK's way of modeling a circuit system is different from doing numerical calculations. It is not possible to directly interface circuit models with numerical calculation blocks. Custom components need to be implemented using controlled voltage or current source. Considering the design rules imposed by SIMULINK, there is no choice but to implement custom model partially based on SIMULINK native blocks (mainly controlled sources) to reuse most of work has been finished before.
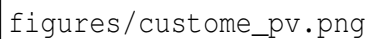
figures/custome_pv.png

Figure 4.1: Custom solar cell model

Figure 4.1 illustrates the final version of custom solar cell model and it is a replica of model that used before. Algebraic loop which is often the reason why simulation cannot run is a critical issue encountered during the process of making the model. Examining Equation 3.1 carefully, it is not hard to find that output current of solar cell depends on terminal voltage of the cell. The terminal voltage, in turn, depends on output current flow through external load. This means that in order to calculate either of them, at least one parameter has already been known,

which forms a classic algebraic loop. For the sake of running simulation, algebraic loop need to be broken by introducing delays into the system. The presence of a delay module in Figure 4.1 on page 26 is to break algebraic loop and setting up initial condition properly. Sometimes, extra delay may lead to unstable operation of the system, however, it is not a problem with custom model fortunately.

4.3.2 RTLAB

As menstioned before, RT model should be able to run using RTLAB because RTLAB is the only software that is capable of communicate with Opal-RT's simulators. The workflow to work with RTLAB is listed below.

1. **Create a new project**

   New project should be created with human readable project name. This would create a new folder to hold all the project related files including RT models and configurations and completions.

2. **Import exisitng model**

   Two options are available during this stage, one is template model provided by the Opal-RT, the other one is directly import the known to work model created previously. For a brand new pure software model, it is recommended to import the templates, then copy the pure software model into the template. In this way, users do not need to configure all the settings from scratch, however, caution need to be taken because some configurations may not be suitable for any models.

3. **Moidify model**

   Once imported, users should be able to edit the model via RTLAB launched MAT-LAB window. There are some requirements imposed by RTLAB and users need to make sure the model should be able to compile and run offline (without loading onto simulator). Since RTLAB is not smart enough at the moment, it needs users' help to separating model into different functional blocks so that each functional blocks can use one processing core in the simulator to maximize performance by running simulation in parallel. Different cores may communicate with each other synchronously or asynchronously depends on users' preference.

4. **Build RT model**

   The reason for this step is that simulator cannot run MATLAB model directly (and should not). In fact, SIMULINK as a high level development environment, has relatively low efficiency running time critical simulation. However, with the help of MATLAB coder, SIMULINK models can be compile into C/C++ which is a low level language that can be run of high efficiency. Generated files are stored in the project folder and once all completions are ready users may continue.

5. **Load model and perform RT simulation**

   In this stage, all completions are transferred from host PC to simulator and RTLAB would put simulator into ready state. Once users click on button to start simulation, simulator would start running and user may use auto-generated console window to interact directly with the simulator. Users may choose to pause or stop simulation at anytime via RTLAB.

4.3.3 eHS

eHSis a generic and reconfigurable FPGA-based electrical solver which provides a convenient user interface enabling users to bring into RTmodels created in the simulation tool of different mainstream simulation environment including SIMULINK, PSIM, PLECS Blockset, and Multisim. **eHS**operate at a much finer time resolution, so communication between **eHS**and CPU cores is archived by sampling. At the beginning of each calculation, **eHS**solver samples the values in CPU and perform calculations based on the sampled values.

In SIMULINK, **eHS**solver is represented as a block diagram and it contains many parameters that users may like to change. Here are some differences between building a regular RTLAB model and **eHS**model.

1. **Separate model into two files**

   RTLAB allows users to put models from *SimScape Power System SIMULINK toolbox* together with normal numerical calculation block under the same subsystem. The idea behind separation of simulation file is still the same. Because RTLAB and **eHS**are not that smart, they need manual process of separation. Briefly, **eHS**is able to handle majority of circuit models in *SimScape Power System toolbox*. Instead of separate those models into subsystem under the same file, they need to be separated into two different files. The rest of the system, apart from those power system models, become master of model and

28

interface with power system models with **eHS**block diagram.

2. **Changing naming of used power system models**

    eHSmakes use of pre-generated FPGA firmware that have special requirements on naming of each modules used in simulation model. By following the naming convention, model translator is able to map each modules onto different sections of digital circuit that take advantage of parallel computation power rendered by FPGA.

4.4 Implementation In Microcontroller

Figure 4.2 on the following page illustrates software flow implemented in TMS320F28027(F28027). Since objective of this project is to implement a simple SSGPVS and perform HIL simulation to prove the concept of fast prototyping and development without real hardware, only the necessary elements are implemented in the F28027. It is quite convenient to extend the program to archive much more functionalities depends on requirements. For example, communication via Universal Asynchronous Receiver/Transmitter(UART), power line communication, and temperature sensing can be added into background tasks section to make it more realistic and closer to actual product.

4.4.1 IQ math and Computation Budget

Due to the fact that our microcontroller is a fixed point microcontroller, there is no floating point processing in F28027. We cannot execute any floating point arithmetic using F28027, however, it is still possible to use float point numbers when writing codes. Compiler has the ability to handle the situation to perform float point calculations using software implemented floating point library. The performance is largely sacrificed because what the software floating point library does is to calculate floating numbers using fixed point using iterative numerical methods. Depends on implementation of software library, multiplications on microcontroller might take so long that in hard RT applications program can fail to meet computational deadline. Especially for digital control of power electronic applications, modern designs call for higher and higher PWM frequency in order to minimize size of passive components, thus, increased bandwidth requirements for controllers.

Fortunately, F28027 has 32x32 and 16x16 Multiply-accumulate(MAC) units inside that offer possibilities to do multiplication using less than 10 clock cycles, which make it promising
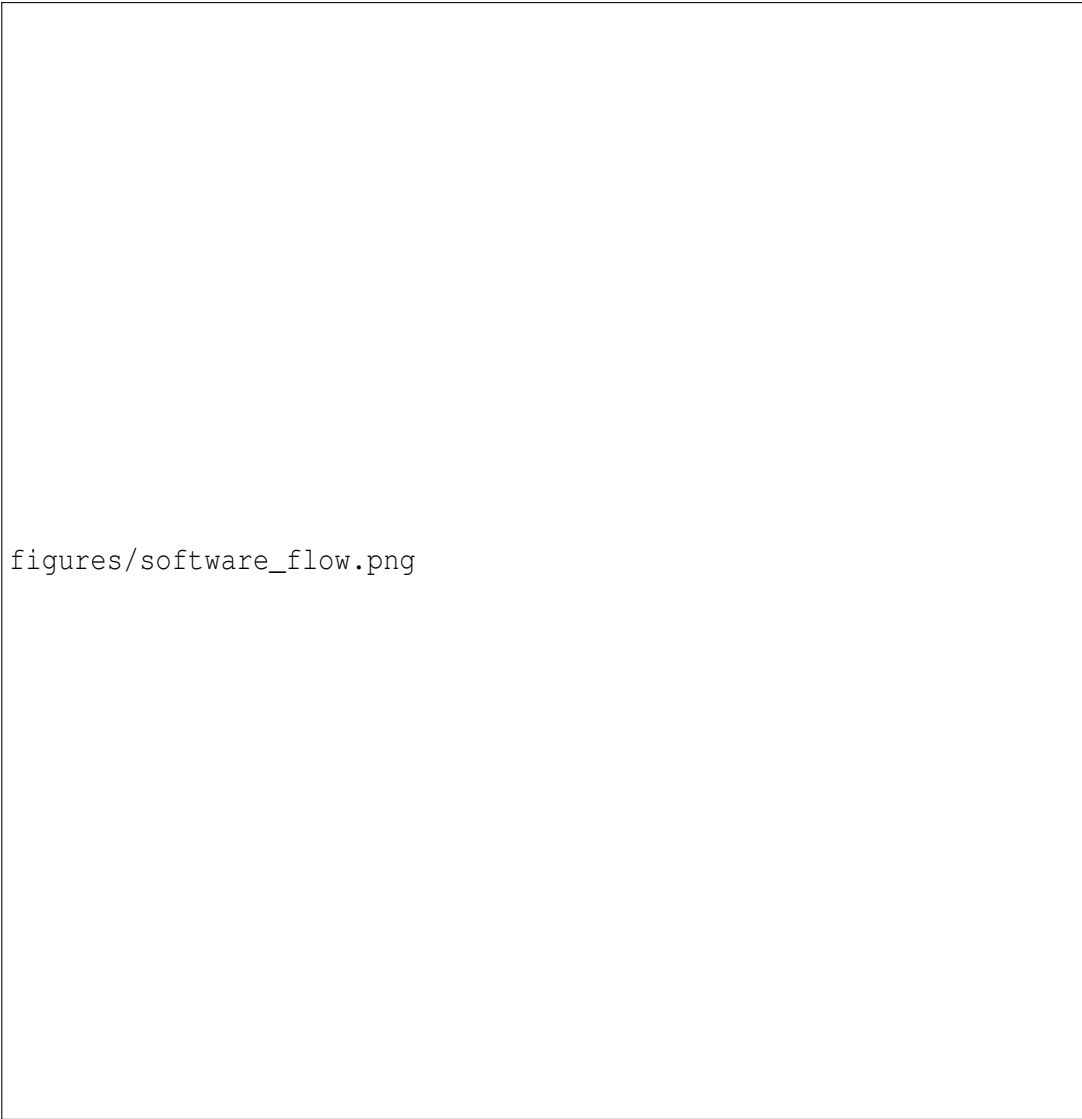
Figure 4.2: Software flowchat

for SSGPVS applications with a reasonable PWM frequency. In order to make use of MAC, software developers need to code in assembly because compiler might not be able to directly translate fixed point   multiplications into assembly efficiently. to squeeze every possible improvement on performance. Usually development in assembly is extremely time consuming and violate origin purpose of fast prototyping. Texas Instruments provides IQ math library as a solution to avoid dilemmas like this. Using the library can ensure that all multiplications can make the most of MAC available. The advantage of using this library is that it is straightforward and relatively simple. the disadvantage is that level of accuracy is not guaranteed. The following table has some key math operations and their benchmarks[16] which are used intensively in the program.

Math operations	Execution cycles	Program memory (words)
sin	46	49
cos	44	47
magnitude	86	96
square root	63	66
multiplication	7	N/A
division	63	71

It is a good practice to avoid as much division as possible because cost of doing division is ten times higher than doing multiplication according to Table 4.4.1. Usually, instead of doing division directly, same operation can be archived by doing reciprocal of divisor.

How IQ math work is quite common in Digital Signal Processing(DSP) applications. Basic idea is that processors treat all quantities as integers. Developers are free to choose any scaling factors to enlarger fractional part of a number to a integer. In order to simplify calculation, scaling factors are better to be exact power of 2 because multiplication with number of power of 2 can be archived by doing bit shifting. Bit shifting is much cheaper than normal multiplication in terms of computation. The developer must then bear in mind that the integer values are scaled versions of the original input samples, scaling the output back accordingly. To minimize the impact of integer rounding effects, it is desirable to scale the results produced by each processing step so that they utilize as many of the available bits as possible.

Based on the benchmarks, it is possible to allocate computation budget. After reviewing some of reference design of microinverter products, it is common that inverter loop running in the controller can archive bandwidth of 50 kHz. Although the microcontroller used in the reference design[18] has better performance compared with F28027, it would be still interesting to see whether it is possible to complete the jobs using a lower end microcontroller.

Firstly, the target bandwidth is the same as reference design, ie. 50 kHz. Assume F28027 operate at maximum clock speed which is 60 MHz, timer with 60 MHz clock input is used to generate interrupts at 50 kHz and all computations are complete in interrupt service routine(ISR). Counter value can be calculated using the following equation.

$$value = \frac{F_{clk}}{F_{loop}} \tag{4.1}$$

where F_{clk} is the clock frequency that drives the counter, F_{loop} is the control loop frequency. In this case, the value is 1200, which means all tasks in inverter control ISR need to be fin-

ished using clock cycles less than this value. Also, other tasks including background and slow loop ISR need clock cycles to execute. After finishing first version of the program, for inverter control ISR alone, there are 24 multiplications, two trigonometric operations, one magnitude operations and one division in total which sum up to execution clock cycle of 409. Other operations or factors need to be considered are summation, conditional operation, function switching overhead, analog to digital converter (ADC) sampling time and on-chip flash synchronization.

Flash synchronization may lead to excessive overhead because flash is considered to be a slow peripheral that operate at a much more slower speed compared with CPU in microcontroller. More than 3 clock cycles of overhead are expected each time the CPU perform a visit of flash. On-chip Static random-access memory(SRAM) is an ideal place to store instructions, constants and computation temporals because there is no clock cycles needed to fetch content stored in SRAM. However, integration of large SRAM in silicon is extremely expensive because it is huge in size compared with other parts of circuit, which means there is limited SRAM and it is not possible to put everything in SRAM for a low end microcontroller. Only time critical part of code may be put in SRAM to archive high performance.

Considering all have been stated above, the microcontroller may not has sufficient capabilities meet timing deadline and overruns are highly expected. In order to be safe, halved original frequency looks like a better choice now. F28027 would definitely have abilities to finish all tasks within time.

4.4.2 Peripherals Configurations

There are two important peripherals that are critical to the implementation of the controller, ADC and timers. ADC is used to convert analog signals(voltage and current) into digital signal so that CPU can use those sample values to perform calculations to control H-bridge. Timers are important because generation of PWM as well as ISR rely on proper functioning of timers.

According to [17], the core of the ADC contains a single 12-bit converter fed by two sample and hold (SH) circuits. Simultaneously as well as sequentially sampling of the SH is possible. SH are controlled by 16 identically Start-Of-Conversions(SOC) units. Developers only need to configure the SOC units and it would take care of the sampling. SOC can be configured to working with either SHs and there are multiple sources to trigger ADC sampling. In this project, since there are four analog channels need to be sampled which mean at least 4 SOCs are needed. During implementation, five SOCs in fact are used because the first value of ADC

sampling may be incorrect according to errata[19] published later. The first SOC is used to control SH to sample analog channel twice to ensure the result is correct. The trigger source of SOCs used is Timer One(T1) which is the same timer to generate PWM signal. When triggered at the same time, SOC with smallest identification number (SOC0) has the highest priority to access SH and do conversion. At the end of each conversion, an interrupt of end-of-conversion is generated and CPU has to response to the interrupt.

Timer of F28027 is extremely powerful, since our PWM frequency is relatvely low, the high resolution feature is not useful in this case. Each counter modules can be used to generate one pair of complementary PWM. Practically, two timers are needed to generate totally four PWM signals. However, bipolar modulated PWM is used in this case and one PWM can be used to control two switches, as a result, only one timer is used which is T1. In order to generate central aligned PWM to minimum THD, T1 is configured to work in up-down counting mode. One comparator is used to set and reset the output. If the counting register is equal to comparator register when counter is counting upward, output is set to high. When the counter register's value is equal to value in period register, the counter start to counter downward. When the counter is counting downward, output reset. The following equation is used to calculate duty cycle.

$$Duty = \frac{Comparator\ value}{Period\ value} \tag{4.2}$$

One advantage offered using up-down counting in this case is that SOC can be triggered when counter value is equal to period value, which means the moment that ADC start sampling is far away from switching moment. During switching transient period, the voltage or current are in transition, which inject undesired noise into system. Sampling during transient period can be extremely noisy and usually people would try this avoid this. Since output impulse is always center aligned with counter peak value. Figure 4.3 on page 35 illustrate how T1 works in this case.

In summary, ADC sampling rate of the system is the same as PWM frequency and controller bandwidth is also the same in this case since controller functions are called right after sampling finishes. Ideally controller can be set to independent with sampling frequency depending on applications and requirements.

4.4.3 Inplement Algorithms

SOGI presented before is in continuous form which is not very helpful since microcontroller works in discrete time. Continuous transfer functions need to be discretize using various method. As mentioned in [9], trapezoidal approximation has a better performance compared with other discretization methods. Discretization operator can be represented using Equation 4.3.

$$s = \frac{2(z-1)}{T_s(z+1)} \tag{4.3}$$

After doing some math, the discrete time system can be summarize as following.

$$H_d(z) = \frac{\frac{x}{x+y+4} + \frac{-x}{x+y+4}z^{-2}}{1 - \frac{2(4-y)}{x+y+4}z^{-1} - \frac{x-y-4}{x+y+4}z^{-2}} = \frac{b_0 + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \tag{4.4}$$

$$H_q(z) = \frac{\frac{ky}{x+y+4} + \frac{2ky}{x+y+4}z^{-1} + \frac{ky}{x+y+4}z^{-2}}{1 - \frac{2(4-y)}{x+y+4}z^{-1} - \frac{x-y-4}{x+y+4}z^{-2}} = \frac{qb_0 + qb_1 z^{-1} + qb_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \tag{4.5}$$

where $y = (\omega_n T_s)^2$ and $x = 2k\omega_n T_s$. ω_n is the fundamental frequency, in this case, $\omega_n = 2\pi 50 = 314.2$. k is scaling factor determine the transient response of PLL. In the program, k is set to be 0.5. T_s is the sampling frequency or the time interval between two runs of PLL.

The current PR controller given in previous chapter is also defined in continuous time domain. Same method, ie trapezoidal approximation, has been used to derive the discrete time system.

$$G_p r = K_p + \frac{K_i \frac{2(z-1)}{T(z+1)}}{\frac{4(z-1)^2}{T^2(z+1)^2} + \omega_n^2} \tag{4.6}$$

$$= \frac{(K_p + \frac{2K_i T}{(4+T^2\omega^2)}) +}{y} \tag{4.7}$$

4.5 Performing PIL Simulation

figures/timer_wave.png

Figure 4.3: Timer waveform

Chapter 5

Evaluation

This chapter is mainly provided for the purpose of showing a typical thesis structure. There are no more thesis requirements described.

5.1 Results

The result of this work is the present document, being both a LaTeX template and a thesis requirement specification.

5.2 Discussion

The dual function of this document somewhat de-emphasises the primary purpose of the document, namely the thesis requirements. It would be better, perhaps, if these could be stated on a few concise pages.

Chapter 6

Conclusion

A thesis requirements/template document has been created. This serves the dual purposes of giving students specific requirements to their theses — both style and content related — while providing a typical thesis structure in a LaTeX template.

6.1 Future Work

Extract the requirements from the template in order to have very concise requirements.

Bibliography

[1] O. Abdel-Rahim, H. Funato, and J. Haruna. Pseudo single stage flyback current source inverter for grid connected pv applications. In *Industrial Electronics Society, IECON 2015 - 41st Annual Conference of the IEEE*, pages 000001–000006, Nov 2015.

[2] Morgan Bazilian, Ijeoma Onyeji, Michael Liebreich, Ian MacGill, Jennifer Chase, Jigar Shah, Dolf Gielen, Doug Arent, Doug Landfear, and Shi Zhengrong. Re-considering the economics of photovoltaic power. *Renewable Energy*, 53:329 – 338, 2013.

[3] Habbati Bellia, Ramdani Youcef, and Moulay Fatima. A detailed modeling of photovoltaic module using matlab. *NRIAG Journal of Astronomy and Geophysics*, 3(1):53–61, 2014.

[4] M. A. G. de Brito, L. Galotto, L. P. Sampaio, G. d. A. e Melo, and C. A. Canesin. Evaluation of the main mppt techniques for photovoltaic applications. *IEEE Transactions on Industrial Electronics*, 60(3):1156–1167, 2013.

[5] P. Chamarthi, M. Rajeev, and V. Agarwal. A novel single stage zero leakage current transformer-less inverter for grid connected pv systems. In *Photovoltaic Specialist Conference (PVSC), 2015 IEEE 42nd*, pages 1–5, June 2015.

[6] Y. M. Chen, H. C. Wu, Y. C. Chen, K. Y. Lee, and S. S. Shyu. The ac line current regulation strategy for the grid-connected pv system. *IEEE Transactions on Power Electronics*, 25(1):209–218, Jan 2010.

[7] Lee Chia-Tse, Liou Jian-Nai, Ma Hsi-Pin, Huang Po-Chiun, and Cheng Po-Tai. Implementation of the digital control algorithm for the maximum power point tracking dc module. In *Control and Modeling for Power Electronics (COMPEL), 2012 IEEE 13th Workshop on*, pages 1–6.

[8] M. Ciobotaru, R. Teodorescu, and F. Blaabjerg. Control of single-stage single-phase pv inverter. In *Power Electronics and Applications, 2005 European Conference on*, pages 10 pp.–P.10.

[9] Mihai Ciobotaru, Remus Teodorescu, and Frede Blaabjerg. A new single-phase pll structure based on second order generalized integrator. In *IEEE Power Electronics Specialists Conference (PESC)*, pages pp. 361–366. IEEE.

[10] O. Cr, x, ciun, A. Florescu, S. Bacha, I. Munteanu, and A. I. Bratcu. Hardware-in-the-loop testing of pv control systems using rt-lab simulator. In *Power Electronics and Motion Control Conference (EPE/PEMC), 2010 14th International*, pages S2–1–S2–6.

[11] K. Ding, X. Bian, H. Liu, and T. Peng. A matlab-simulink-based pv module model and its application under conditions of nonuniform irradiance. *IEEE Transactions on Energy Conversion*, 27(4):864–872, 2012.

[12] Javad Khazaei, Zhixin Miao, Lakshan Piyasinghe, and Lingling Fan. Real-time digital simulation-based modeling of a single-phase single-stage pv system. *Electric Power Systems Research*, 123:85–91, 2015.

[13] S. B. Kj. Evaluation of the "hill climbing" and the "incremental conductance" maximum power point trackers for photovoltaic power systems. *IEEE Transactions on Energy Conversion*, 27(4):922–929, 2012.

[14] MATLAB. Single-Phase, 240 Vrms, 3500 W Transformerless Grid-Connected PV Array, 2017.
https://au.mathworks.com/help/physmod/sps/examples/
single-phase-240-vrms-3500-w-transformerless-grid-connected-pv-array.
html [Accessed: 2017-05-21].

[15] Standards Australia. AS/NZS 5033:2014 Installation and safety requirements for photovoltaic (PV) arrays, 2014.
https://infostore.saiglobal.com/store/details.aspx?ProductID=
1764523[Accessed: 2017-05-21].

[16] Texas Instruments. *C28x IQmath Library - A Virtual Floating Point Engine*, August 2011. V1.6.0.

[17] Texas Instruments. *TMS320x2802x, 2803x Piccolo Analog-to-Digital Converter (ADC) and Comparator Reference Guide*, Dec 2011.

[18] Texas Instruments. *Digitally Controlled Solar Micro Inverter using C2000 Piccolo Microcontroller*, April 2014. Version 1.0.

[19] Texas Instruments. *TMS320F28027/28026/28023/28022/28021/28020/2802x0 Piccolo MCU Silicon Errata*, Dec 2015. Rev. M.

[20] H. Vardhan, B. Akin, and H. Jin. A low-cost, high-fidelity processor-in-the loop platform: For rapid prototyping of power electronics circuits and motor drives. *IEEE Power Electronics Magazine*, 3(2):18–28, June 2016.

[21] Geoffrey R. Walker. Evaluating mppt converter topologies using a matlab pv model. In *AUPEC 2000 : Innovation for Secure Power*, pages 138–143, 2000.

[22] S.R. Wenham. *Applied Photovoltaics*. Earthscan, 2007.

[23] L. Zhang, K. Sun, Y. Xing, L. Feng, and H. Ge. A modular grid-connected photovoltaic generation system based on dc bus. *IEEE Transactions on Power Electronics*, 26(2):523–531, Feb 2011.

[24] Yan Zhou, Fei Liu, Jinjun Yin, and Shanxu Duan. Study on realizing mppt by improved incremental conductance method with variable step-size. In *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pages 547–550.

Appendix 1

In the source code for this document, some commonly used LaTeX operations can be found. The typesetting opportunities with LaTeX are vast, and there exist an enormous collection of packages that give added functionality. Modern LaTeX distributions come with many of the most common packages (some used in this document) and many more can be found on the web.