INVERTER PLATFORM DEVELOPMENT FOR INVERTER PARALLELING

By

SAMUEL NOAH RAMRAJKAR

A thesis submitted to the

Graduate School-New Brunswick,

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Electrical And Computer Engineering

Written under the direction of

Hana Godrich

And approved by

_____

_____

_____

New Brunswick, New Jersey

May 2015

ABSTRACT OF THE THESIS

Inverter Platform Development For Inverter Paralleling

By SAMUEL NOAH RAMRAJKAR

Thesis Director:

Dr. Hana Godrich

With the advent of distributed power generation and FACTs, inverter paralleling is an indispensible part of modern power systems. The problem of inverter paralleling poses a unique requirement of sharing the required total power from the connected inverters. The following report describes the research for development of an open platform for inverter paralleling. This includes development of hardware and software for an inverter which can be used as building block for the further research of control algorithms used in power flow control.

In the wake of development of the platform, the project uses the novel "Direct Digital Synthesis" to generate the PWM signals. With a need for low cost inverter, the developed algorithm coupled with cost efficient circuit topology, provide the project with the needed cost effectiveness.

INDEX

Abstract Of The Thesis..... ii
Introduction..... 1
Inverter Hardware Design..... 4
Introduction:..... 4
Literature study:..... 4
The design:..... 6
1. Power Switches:..... 6
2. MOSFET drivers:..... 7
3. Snubber:..... 9
4. Boots Strap Circuit:..... 12
5. Feedback and Protection Circuits..... 15
Results..... 18
Direct Digital Synthesis for PWM..... 20
Introduction:..... 20
Literature Study:..... 20
Types of PWM generation:..... 22
DDS in PWM:..... 27
Results..... 31
Firmware Ideology..... 33
Introduction:..... 33
Project Structure..... 33
Some Concepts Implemented:..... 34
Algorithmic Flow..... 36
Fuzzy Logic Control: Frequency and Voltage Regulation..... 38
Introduction:..... 38
Literature Study:..... 38
Idea of Fuzzy logic:..... 41
The design..... 44

Result 50
Concepts In Inverter Paralleling 51
Introduction: 51
Literature Study: 51
Platform Readiness: 56
Results 58
Conclusion 60
Future Work 60
Appendix 1 61
Bibliography 66

LIST OF FIGURES

1. Figure 1:H bridge Inverter. Source: http://www.robotoid.com/my-first-robot/rbb-bot-phase2-part1.html..... 06
2. Figure 2: The block diagram and pinout for the chip..... 09
3. Figure 3: $_{VDS}$ voltage transients with no snubber..... 11
4. Figure 4: V_{DS} voltage transients with snubber in circuit..... 12
5. Figure 5:Half bridge driver..... 14
6. Figure 6:Boot strap circuit..... 15
7. Figure 7: Bus voltage feedback..... 19
8. Figure 8: Output voltage feedback..... 20
9. Figure 9: Bus voltage and output voltage ZCD..... 21
10. Figure 10: Output of inverter at the low voltage side of transformer..... 21
11. Figure 11: PWM signal. Source: Wikipedia http://en.wikipedia.org/wiki/Pulse-width_modulation..... 22
12. Figure 12: PWM generation using a sine triangular comparison method..... 24
13. Figure 13: PWM using delta method..... 25
14. Figure 14: Direct Digital Synthesis block layout. Source: Wikipedia http://en.wikipedia.org/wiki/Direct_digital_synthesizer..... 27
15. Figure 15: Positive half mapping for the pulse widths..... 31
16. Figure 16: Negative half mapping for the pulse widths..... 32
17. Figure 17: PWMs at 1A and 1B..... 33
18. Figure 18: Zoomed PWM for 1A and 1B..... 34
19. Figure 19: PWM for 1A and 2B..... 34

20. Figure 20: Firmware branching..... 35

21. Figure 21: Output of a P controller..... 41

22. Figure 22: Integral controller`s output..... 42

23. Figure 23: General Fuzzy control block diagram..... 45

24. Figure 24: Designed fuzzy sets for the frequency control..... 47

25. Figure 25: Designed fuzzy sets for the voltage control..... 47

26. Figure 26: Output of the inverter at High voltage side..... 52

27. Figure 27: inverters connected in parallel. Source: [10]..... 53

28. Figure 28: Standard Droop curves. Source [1]..... 57

29. Figure 29: Droop control block diagram. Source: [1]..... 58

30. Figure 30: No synchronization in the bus and inverter voltage..... 60

31. Figure 31: Synchronization achieved 61

LIST OF TABLES

1. Table 1: Positive half mapping 30
2. Table 2: Negative half mapping..... 31
3. Table 3: DDS algorithm parameters..... 33
4. Table 4: Fuzzy set definition..... 48
5. Table 5: Fuzzy logic implementation example..... 49
6. Table 6: Knowledge base for frequency control..... 50
7. Table 7: Knowledge base for voltage control..... 50

Introduction

In recent years, the power industry is shifting to integration of distributed energy resources (DERs) in addressing the ever-increasing need for power generation. Government incentives in developed countries are driving the adaptation of renewable energy option for DERs, setting up the stage for economically harvesting of solar, wind, geothermal and other sources with smaller carbon footprint such as fuel cells. A significant advantage of DERs integration at the power distribution level is the associated reduction in load over transmission lines and, thus, in losses. Additionally, with more generation delivered locally, growth requirements on ultra high voltage (HV) lines are minimized, offering lower investment cost and higher power efficiency for the inevitable increase in modern day power consumption [2].

DERs, like uninterruptable power supply (UPS) systems, can work in one of two modes of operation: i) Grid tie and ii) off grid or island operation. While in the former DERs will not be delivering power if the grid goes down in the later the DERs will support critical load, isolated from the grid. This type of operation enables micro-grid architecture. Micro-grid refers to a stand-alone small power grid with internal generation, transmission line, and control. A micro-grid can keep generating power locally and support load without a connection to the larger power grid (or utilities). With the development in energy storage systems (ESS) a steady increase in micro-grid implementation is seen, addressing both the need for increased power and higher resilience to weather conditions and large scale power outages. The technology that takes center stage in enabling micro-grid operation are power converter solutions [2] [3].

Flexible AC Transmission Systems (FACTs) [4] are the latest technology in transmission line support. They offer increased reliability, efficiency, and capacity to the transmission system through reactive power compensation and address some power quality issues. The static synchronous compensator (STATCOM) [5] belongs to the FACT family of products. The system is based on solid-state inverter capable of supplying and/or absorbing reactive power. This allows the control of the power angle and the voltage regulation over the lines.

FACTs opens an all new frontier in power electronics with the advent of custom power devices; which provide a mean for improving harmonic contents on the transmission line an at substations and improve overall power quality. One such system is the APF (Automatic power factor correction) [6]. Required current harmonic compensation and filtering bandwidth governs the rating of these devices and their operating frequency. In high power applications, not all harmonics may be filtered by a single converter. A counter solution is to use of parallel multi-level converters and meet the power rating for each of them connected in series. However, this makes the use of bulky and cost ineffective step down transformer at the output an indispensible part of the system. Thus multi converters in parallel connection are the natural choice to address such problems.

All the aforementioned applications have a core technology in common; they use static converters in parallel connection. The objective of this thesis was to design and build an inverter platform that can be used in research of new control paradigm for inverters in paralleling. The following report documents the efforts of developing an open platform technology, both hardware and software, for a modest 24v 50W inverter to be used for all

further efforts for paralleling. This is followed by a detailed study of the existing control algorithms and a discussion of how these can be ported on the developed environment.

Chapter one starts with a brief discussion about the inverter basics and then discusses in depth about the designed hardware with its specifications. Appendix 1 documents the schematic and the PCB layout. Chapter two talks about the general PWM techniques and then talks about the novel use of "Direct Digital Synthesis (DDS)" for the MOSFET firing. Chapter 3 presents the software ideology for the inverter.

Chapter four explain the frequency and voltage control using the Fuzzy logic control. Finally chapter 5 discusses the existing solutions for inverter paralleling and shows how to port them on the developed platform. This section also explains the exact question of inverter paralleling and then explains all the solutions.

Inverter Hardware Design

Introduction:

Inverters are solid state circuits that convert direct current/voltage to an alternating one. There are many types and topologies for inverters like the Voltage source inverters, current source inverters, multi level, resonance, etc. For the research at hand, I choose the voltage source inverter. The fully tested inverter is a 24v/50w software limited variant.

Literature study:

Broadly speaking inverters are of two types:

Voltage Source Inverters (VSI): In these types of inverters the output voltage waveform is controlled independently. Thus these act as voltage sources. The VSI generate waveforms with discrete voltage values. Thus the load sees a high dv/dt changes. For the load to see a sinusoidal current, the load should be inductive at the harmonic frequencies. If the load is capacitive, VSI will generate high current spikes. In such cases we need an inductive filter to smooth out the current waveform [7].

Current Source Inverters (CSI):): In these types of inverters the output current waveform is controlled independently. Thus these act as current sources. The CSI generate waveforms with discrete current values. Thus the load sees a high di/dt changes. Thus in contrast to the VSI, load needs to be capacitive at the harmonic frequencies. An inductive load will generate large voltage spikes, and a capacitive filter may be needed [7].

If the inverters are further investigated, they can be classified in various other ways. Thus inverters are classified as one of the following types: square wave, pure sine, modified sine, variable frequency, etc.

The inverter developed is a pure sine VSI. Thus hence forth the report covers only the VSI modified sine wave inverter.

Carrier based Pulse Width Modulation technique:

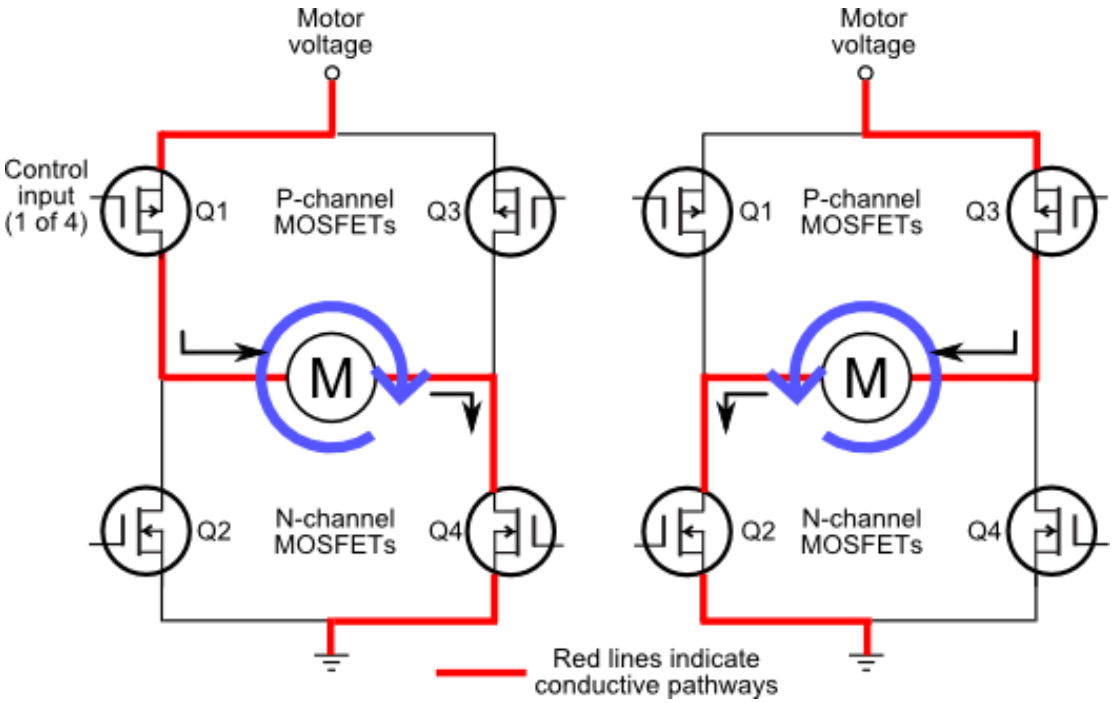


Figure 1:H bridge Inverter. Source: http://www.robotoid.com/my-first-robot/rbb-bot-phase2-part1.html

The fig 1, shows a single phase VSI. The major components for the VSI are:

Vdc: This is the "DC link", which is a capacitor bank with the capability of providing the inverter with the required voltage and the power to the load.

Switches: Q_1 thru to Q_4 are the 4 power switches used to control power and voltage waveform seen by the load.

Load: Finally the box with voltage V_o is the load to the inverter

The output voltage waveform should follow the desired shape of a sine wave. This achieved by switching the switches in a desired manner. The technique used for the inverter at hand was the carrier based PWM. In this method we define the 'ON' and the 'OFF' time of the switches by a predefined rule.

The predefined rule uses two waveforms: Sine wave (signal we want) as the modulating frequency (V_m) and a high frequency (at least 10 times higher in frequency) carrier wave(V_c). This special case is called the Sine PWM. If $V_m > V_c$ then the switch Q_1 and Q_2 is closed else the other two are closed.

A variation to this is when a pure sine wave is needed, as in our case. In this case Q_2 is always kept closed and Q_1 is kept closed if modulating signal is greater than the carrier signal. S_4 is switched complementary to Q_1. Q_3 is kept open all this while. This is followed for the positive half of the inverted output. For the negative half Q_4 is always closed and the switches Q_3 and Q_2 are switched complementary to each other when the if modulating signal is less than the carrier signal.

The design:

1. Power Switches:

Q_1 to Q_4 are the 4 power switches used for the single phase power inverter stage. Due to low switching requirements, small power rating and easy of development MOSFETs were selected. For the low power variant the IRF 530N or the IRF 540N are to be used.

The IRF 530N is a 100V rated MOSFET with on resistance of 90mΩ and rated I_D of 17A. The device can dissipate up to 70W. Also This device has a pulsed drain current of 60A.

The IRF 540N is the higher rated MOSFET and can be used in place of 530 if over rated design is needed. This device has a on resistance of 44mΩ and I_D 33A with pulsed current of 110A. With a dissipation of 110W rated, the device can deliver higher power performance when needed.

Both the selected devices work up to 13kHz of switching frequency. With a carrier frequency of PWM set to 10kHz this is well within the rated operating frequency.

The same topology and the MOSFET driver can drive a high voltage MOSFET. The entire power card was developed to accommodate the possibility of a 120V or a 240V output. For this design variant, the Infineon Cool MOS based SPP17N80C3 800V MOSFET was selected. It is rated for 17A continuous and 51A pulsed I_D. With a very low on resistance of 290mΩ it can dissipate up to 208W. This device is tested ok with 30V V_{ds} at 10kHz carrier frequency and found to be working well.

For all the MOSFETs a gate resistance of 47Ω is used as custom practice and is tested to be working well. These are the resistors R_1 to R_4.

2. MOSFET drivers:

The most important and a usual point of failure for MOSFETs are their drivers. It is very important that the MOSFETs switch with required speed and are complete saturation while conducting the load current. Improper or slow charge pump into the gate fails to charge the gate capacitance with the needed profile and the MOSFETs are destroyed due to thermal run away.

MOSFET drivers are the ICs the provide this profile for charge and charge removal of the MOSFET gate. They also serve as the method of isolation if needed. Keeping these core functions in mind, the MOSFET driver from Avago was selected. The part number is HCPL 3120, which is a 2.5A max output current opto coupled drive.

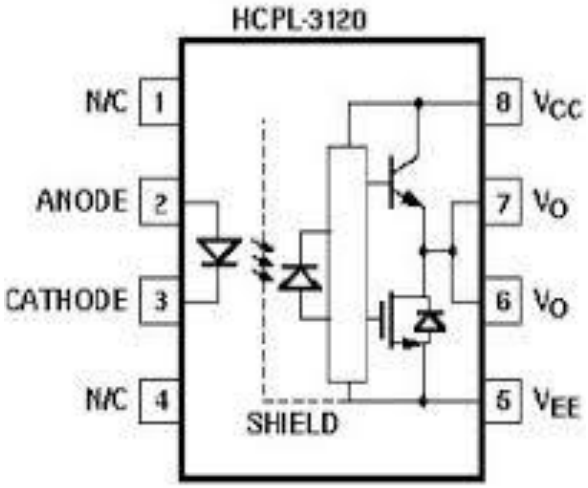


Figure 2: The block diagram and pinout for the chip

The input side is a optically isolated. The output is a complementary push pull. It has a under voltage shut down feature. This saves the MOSFET from thermal burnout. The other good reason to select this device is low component count which makes reliability high. This part removes the need for a opto isolator. Also this part works

Figure2:MOSFET driver. Source: HCPL3120 datasheet

well up to 500kHz of frequency, which was not the case in off the shelf opto isolators.

The input side of the isolator is fed directly from the micro controller`s pin thru a 300Ω resistance.

One added advantage of this device is the very low forward current for the LED to turn on. The Cortex M3 controller selected is a 3.3V device. each pin can source or sink 25mA with the sum of all currents on the port not exceeding 25mA. The design for the current limit was set with the following formula

$$R = \frac{V_{pin} - V_f}{7mA} = 300\ ohm$$

Here 7mA was the chosen forward current thru the LED which is a hard bias according to the figure 15: transfer characteristics and within the limit for the current rating for the Cortex M3. These are the resistors R_5 to R_8.

3. Snubber:

Power devices usually fail due to one of the following reasons:

a. Over Heating
b. Over Current
c. Excessive $\frac{di}{dt}$
d. Excessive $\frac{dv}{dt}$
e. Over voltage (usually during turn off)
f. Excessive switching losses

At low loads and with inductive load we do not need a $\frac{di}{dt}$ protection. However MOSFETs need a $\frac{dv}{dt}$ protection. Before explaining the design for the RC snubber used,

following picture shows the ringing stress on the MOSFET without a snubber during turn on.

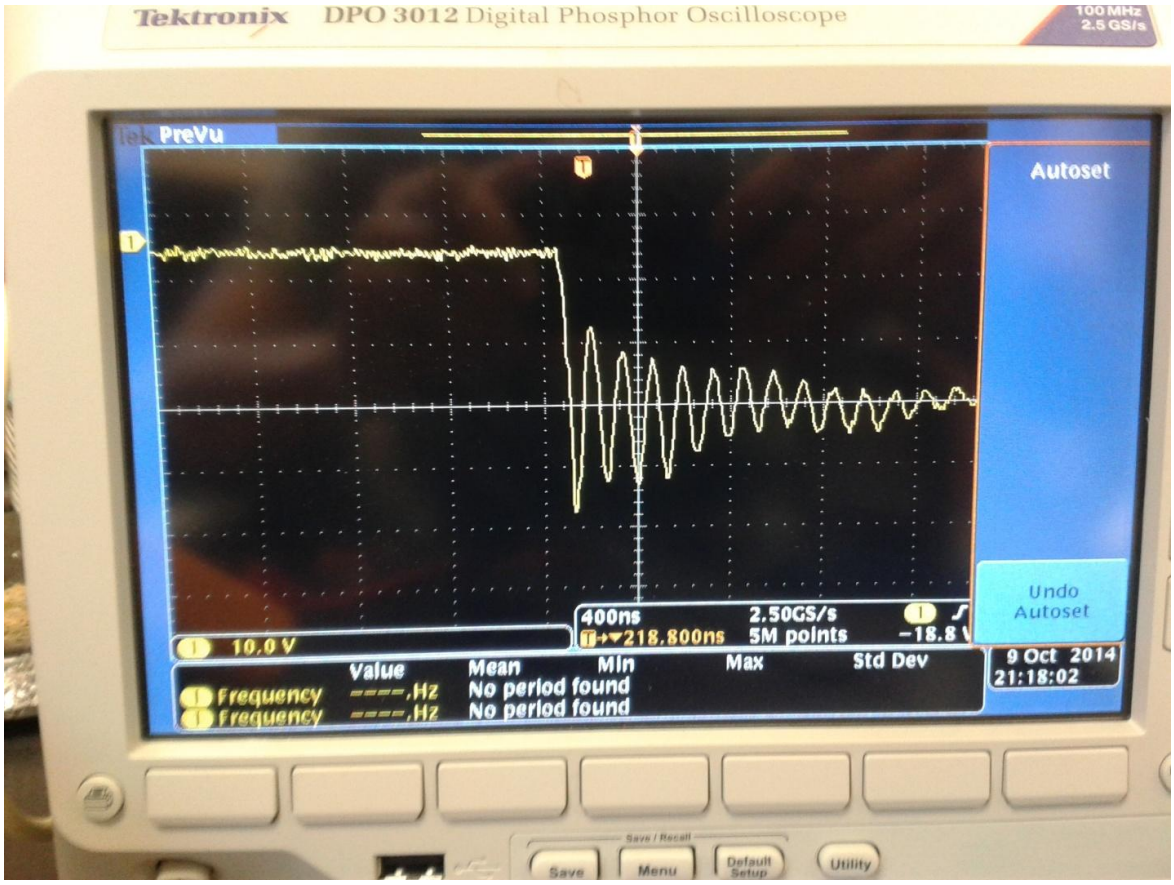


Figure 3: V_{DS} voltage transients with no snubber

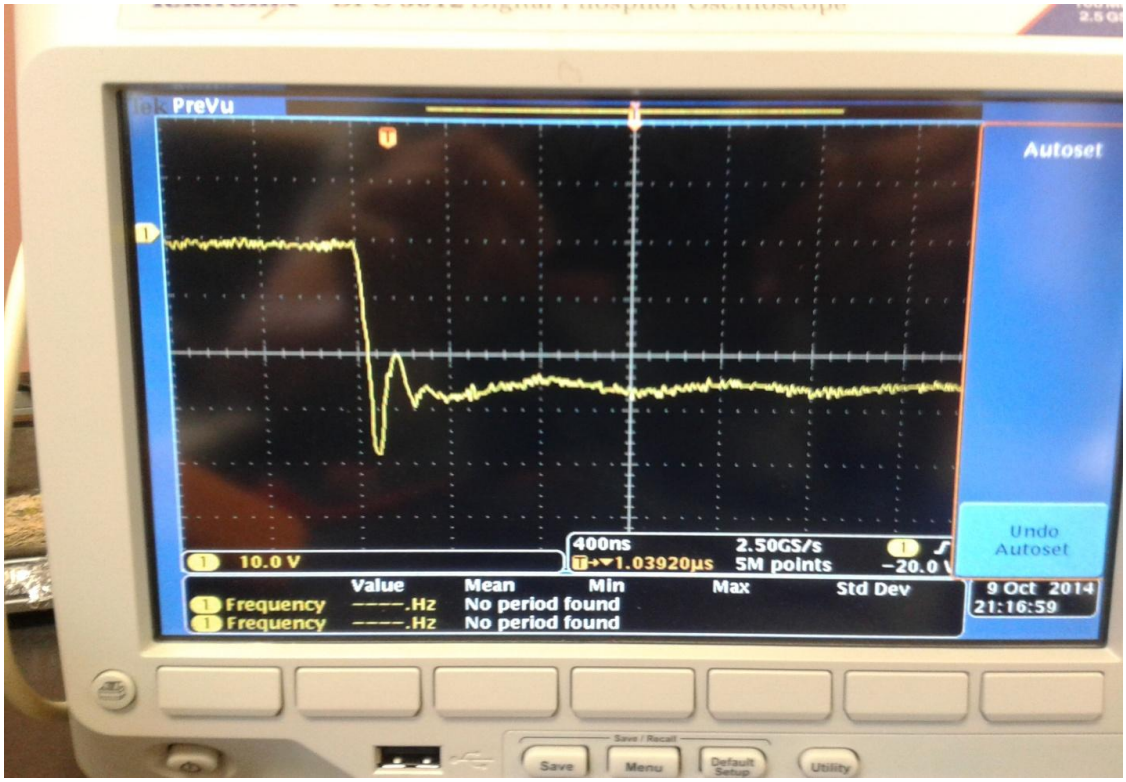


Figure 4: V_{DS} voltage transients with snubber in circuit

The phenomenon is a transient one caught on the scale of 400ns. This clearly shows the multiple ringing on the V_{ds} of the MOSFET during turn on.

To design a snubber circuit, TI suggests the following practical steps which were followed:

a. Measure the ringing frequency at the top of the MOSFET. Solder a good quality COG or film type capacitor from the switch to the ground. This causes the ringing frequency to be half. The effective capacitance is now 1/3 at the node. This is used to calculate the stray inductance using:

   $F_{ring} = \frac{1}{2\Pi\sqrt{L_S * C_T}}$ where C_T is the total capacitance.

b. In order to critically damp the ringing, a capacitor of double the size is used with a resistor in series sized by the formula

$$d = \frac{R}{2} \times \sqrt{\frac{L_S}{C_S}}$$

where d is the damping factor and is set to 1. C_S is the effective capacitance at the node.

There were some difficulties doing this method so we worked around this problem by using this as the lower bound and then designed a RC pole at the frequency counted [8].

4. Boots Strap Circuit:

Power converters employing bridge configurations function by the virtue of the high side power switch. In order to drive a device like a MOSFET or IGBT it needs to be supplied with a positive voltage with respect to its source or emitter. The high side switch does not have its source or emitter at system ground [7].

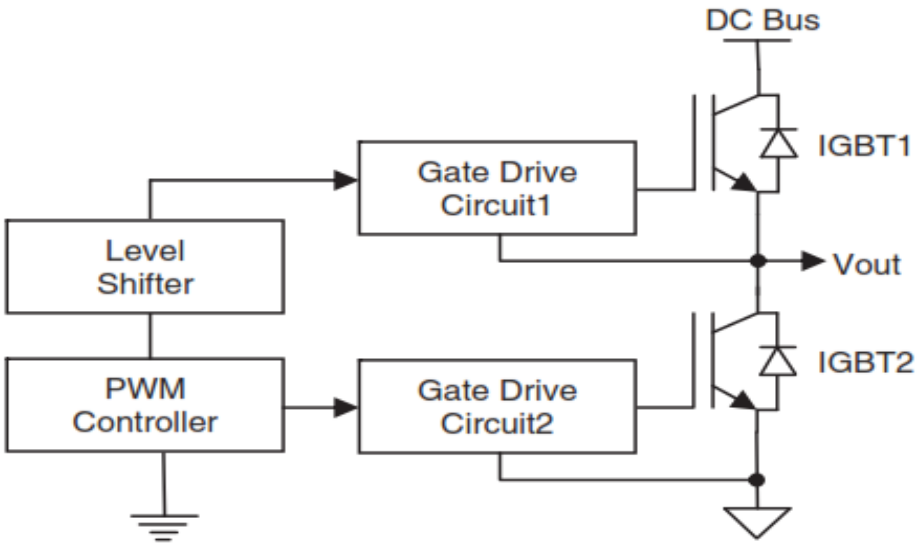


Figure 5:Half bridge driver

Now the emitter of the high switch is also the output terminal. Thus it will float anywhere between system ground to V_{dc} bus. To keep the high side switch in conduction, the gate bias should float from V_{bias} to $V_{bias}+V_{dc}$. This necessitates the use of what is called as a floating bias.

There are three types of floating supplies:

1. Isolated supply:

   This is the simplest way to generate a isolated supply. It uses a line transformer to generate a isolated supply. This is usually a bulky option and at higher power converters accounts to high losses in the transformer. The other way is to use a high frequency isolated DC-DC converter. This allows a smaller transformer.

2. Charge pump supply:

   This technique superimposes a voltage of one supply onto other. It is used to generate a boost voltage on to a higher voltage. However this is too complex for the higher side circuit[4].

3. Bootstrap:

   This is one of the common and simple technique used to generate the floating supply. It uses just one diode and a charge storage capacitor. This is a low cost solution for converters upto several kilowatts.

We have used this for the high side MOSFET drive.

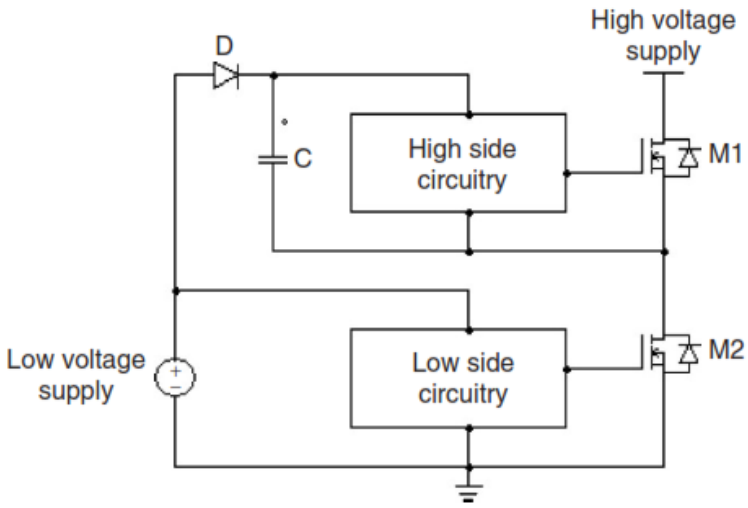


Figure 6:Boot strap circuit

The working for the circuit is very simple. When the low side switch M2 is on, the capacitor is charged almost to the supply voltage thru M2. The voltage across the capacitor is supply voltage minus the diode and MOSFET saturation voltage. When the time comes to fire the high side MOSFETs, the capacitor is used to power the driver. The voltage across the capacitor droops and is a function of current drawn, size of capacitor and the frequency of operation.

The capacitor is sized according to the following formula

$C_{Boot} = \frac{Q_{Total}}{\Delta V_{Boot}}$ Where, ΔV_{Boost} is the change in the boot strap voltage

The total charge is given by

$Q_{Total}=Q_{gate}+(I_{LKCAP}+I_{LKGS}+I_{QBS}+I_{LK}+I_{LKDIODE})*t_{ON}$

Here,

I_{LKCAP}=Capacitor leakage current

I_{LKGS}=Gate to Source leakage current

I_{QBS}=Bootstrap circuit quiescent current

I_{LK}=Bootstrap circuit leakage current

$I_{LKDIODE}$=Bootstrap circuit leakage current

Now the worst case is for the Infineon MOSFET SPP17N80C3. So the capacitor is sized for this MOSFET.

Q_{Total}=91nC+(3uA+100nA+5mA+0+5uA)*10us=141.081nC

For change of about 1.5mV in the supply for the upper drive circuit we get a capacitor requirement of 1000uf. I have sized it for 35V. Capacitor C1 and C2 are the bootstrap capacitors. Off the self 1N4007 was selected as the bootstrap diode.

5. Feedback and Protection Circuits

Inverters need some electrical protection and feedbacks to operate within limits and with intended functions. Few of the protection functions needed for safe operations are:

I. DC Bus over and under voltage
II. Output AC over and under voltage
III. Frequency upper and lower bounds
IV. Short circuit and over load protection

V. Thermal limit

1. DC link Feedback

   Resistors R_9 and R_{10} forms a voltage divider network. The resistors are so sized that the input voltage to the controller`s ADC pin is within 3.3V. R_9 is chosen to be 100k and R_{10} as 12k.

   This will divide the DC link voltage to

$$V_{DC\ link\ fb} = \frac{R_{10} \times V_{DC}}{R_9 + R_{10}}$$

   This works well for the 30V DC link of the low power converter. For the high power converter use V_{dc} as 400V and resize the resistor R_9 to 200k and R_{10} to 1.5k.

   Finally a ceramic capacitor C3 of 1.2nf is used across R_{10} for transient suppression during DC link charging.

2. Current Feedback

   R_{11} is a 0.04 Ω current sense resistance connected in series with the two legs of the inverter. This resistor gives a unipolar current feedback during both the halves of the output waveform.

3. Output voltage feedback:

   The heart of the voltage, zero crossing detector and frequency feedback is the 1V reference signal generated using the resistors R_{16} and R_{17}. These are sized using the reference design from Texas Instruments. This gives R_{16} as 10k Ω and R_{17} as 4.3k Ω. This is followed by the voltage buffer built around the IC LM324. This IC was selected as the feedback signals are all unipolar. This is achieved by the use

of the 1V signal on which the voltage signal rides at a quiescent value of 1V. The superimposition is achieved at the difference amplifier stage (U1.2) where the R_{19} and R_{20} are 100kΩ resistances used to step down the output voltage and the 2.4kΩ R_{18} resistance provides the quiescent bias. Finally the micro controller is fed the voltage feedback thru the RC pole of R_{22} and C_8 sized as 100Ω and 0.1uf. R21 is the feedback resistance. Its value decides the clipping in the feedback signal and is sized to 2.4kΩ by simulation and some hands on.

4. Zero crossing detector:

   The U1.3 stage of the Opamp IC is used as a ZCD. This is a simple comparator, that compares the output feedback signal going to the MCU and the 1V bias. This is true as the feedbacks are unipolar riding on the 1V, which is now the 'Zero' for the actual output wave.

5. Frequency Feedback:

   The ZCD`s output is a square wave of the same frequency as the output waveform. This is fed to a charge pump circuit, whose output is then proportional to the input frequency. When the output is low for the ZCD, capacitor C_9(sized at 0.1uf) is charged thru diode D_3. When the output for the ZCD goes high the pnp transistor Q_6 is turned on and charges the capacitor C_{10}. The output across the capacitor and the discharging resistor R_{24}(10kΩ sized by trial and error) is proportional to the output frequency. A little tuning is software gives the frequency output.

Results

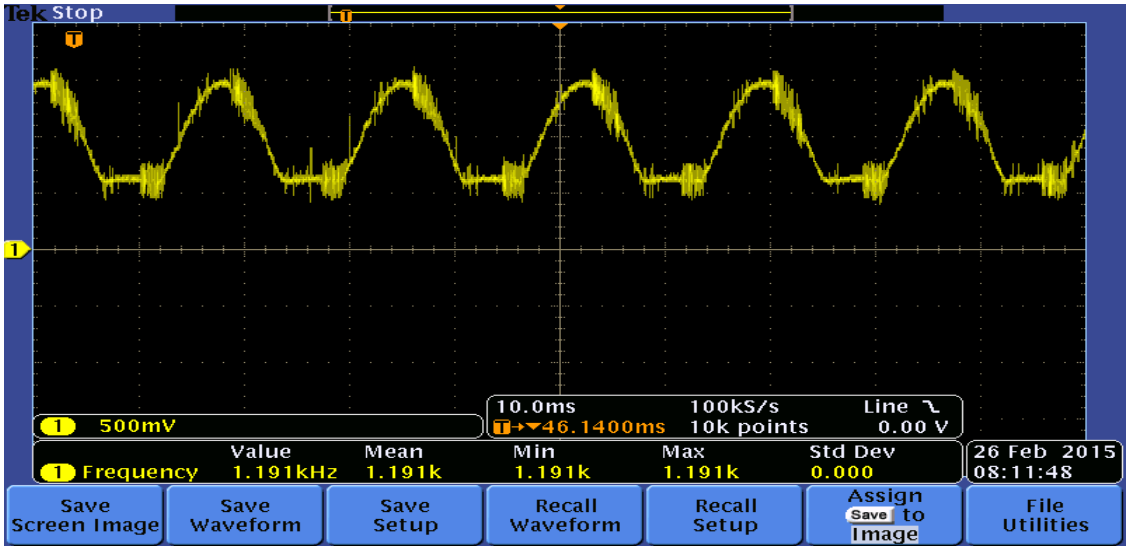


Figure 7: Bus voltage feedback

This is measured off the difference amplifier for the bus voltage feedback. The wave shape distortion is not a problem as only voltage peak is measured and averaged. Finally the amplitude is scaled to the actual value using a simple amplifying factor.

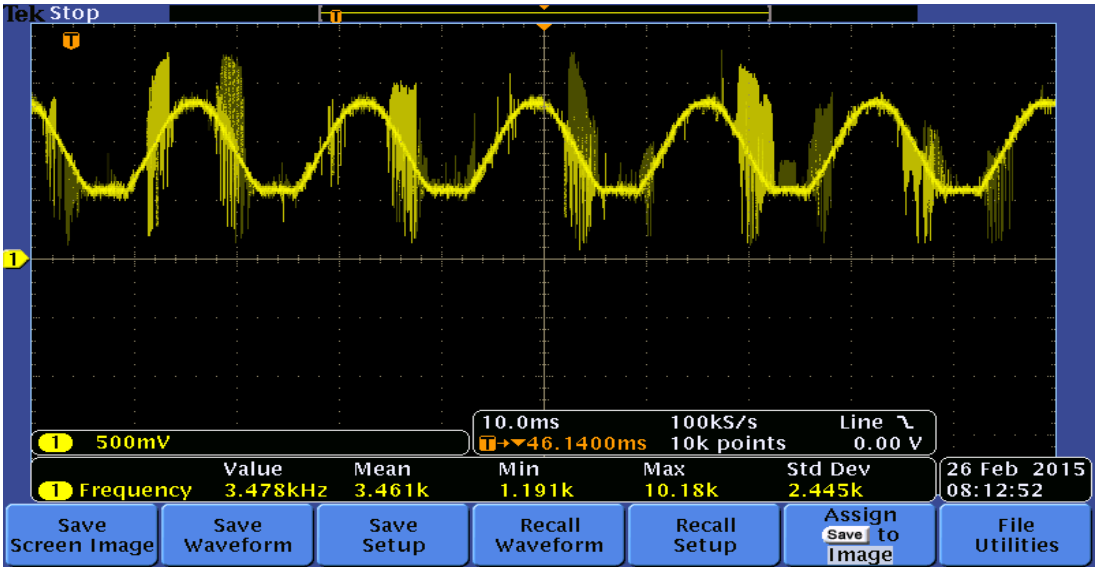


Figure 8: Output voltage feedback

This is a similar circuit for the inverter output voltage feedback. However this is corrupted by the conducted EMI at the PWM frequency. This is handled by a RC filter and further by software filter in the firmware. This is a simple average finding algorithm.

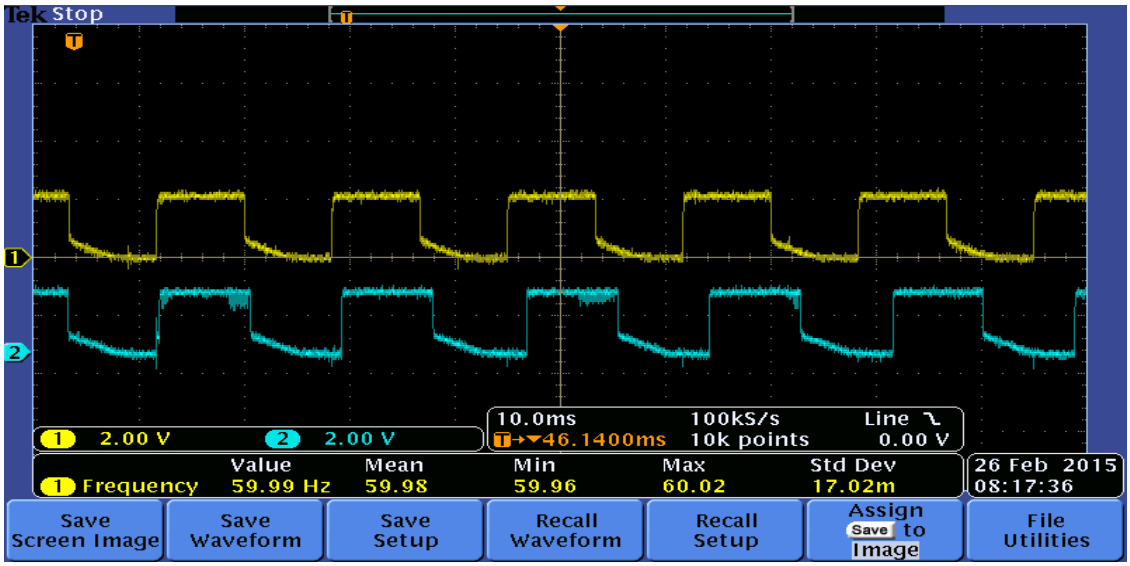


Figure 9: Bus voltage and output voltage ZCD

The output of the Zero Crossing Detectors for the bus voltage feedback (yellow) and for the inverter output voltage(blue). These signals are used as the interrupt signals to the Cortex M3 for the frequency measurement mechanisms and the phase control code.

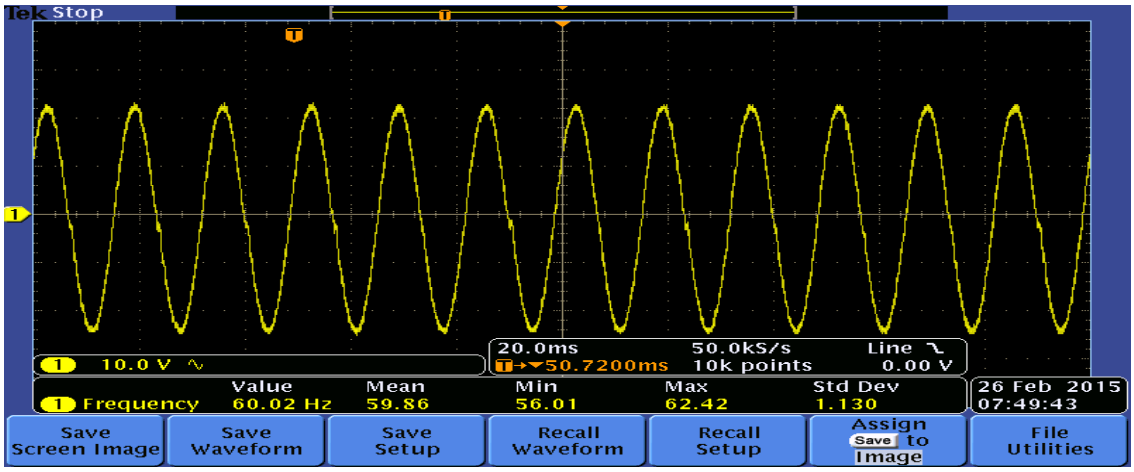


Figure 10: Output of inverter at the low voltage side of transformer

This is the inverter output voltage after it is latched in frequency, amplitude and phase.

Direct Digital Synthesis for PWM

Introduction:

Pulse Width Modulation (PWM) forms the bases of the inversion process. The following chapter introduces the general concept of PWM generation and the talks about the novel Direct Digital Synthesis (DDS) method used in this project to generate the PWM signals.

Literature Study:

Pulse width modulation is a method of modulation. Also its modulation technique can be used for encoding information. However its main use is to control power in the power electronics devices like the inverters, motor drives, power supplies, solar chargers for MPPT and power amplifiers better known as class D amplifiers [9].

The main advantage of PWM is that it uses the power device in its saturation or its cut off region of operation. This brings down the power dissipation in the device down to a very low value as compared to the controlled power.

For a quick recap of the basic nature of PWM consider the following waveform shown

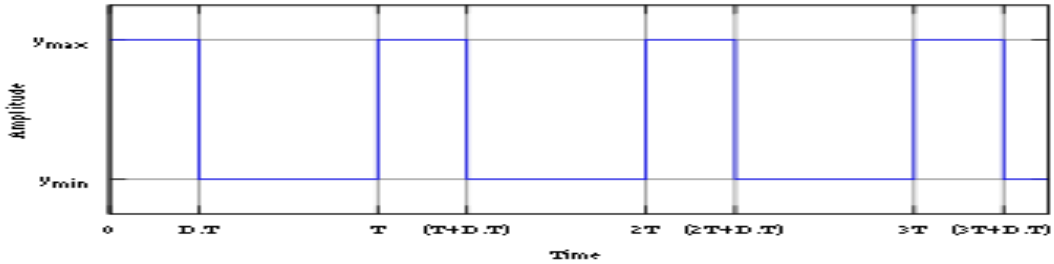


Figure 11: PWM signal. Source: Wikipedia http://en.wikipedia.org/wiki/Pulse-width_modulation

The PWM signal is a series of "ON" and "OFF" pulses. The time for which they are on as compared to the total time period of the waveform is called as the duty cycle of the PWM. Varying the "ON" time or better put as the duty cycle, the average value of the waveform can be varied between 0% to 100% of the amplitude of the pulse.

If we consider the waveform in the diagram above an call it as f(t) with period T, low value as y_{min} and a high value or the amplitude as y_{max} and a duty cycle D

then the average value of the signal is given by [9]

$$y_{avg} = \frac{1}{T}\int_0^T f(t)dt$$

Now as f(t) is a pulse waveform, its value is y_{max} for 0< t< D.T and y_{min} for D.T< t <T . The above expression then becomes [9]:

$$\bar{y} = \frac{1}{T}\left(\int_0^{DT} y_{max}dt + \int_{DT}^{T} y_{min}dt\right)$$

$$= D.y_{max} + (1 - D).y_{min}$$

Here D is given by

$$D = \frac{T_{on}}{T}$$

For most of the PWM signals generated and for the one we have used in the project the y_{min} value is 0. This simplifies the average equation to [9]

$$y_{ave} = D.y_{max}$$

Types of PWM generation:

1. Sinusoidal PWM:

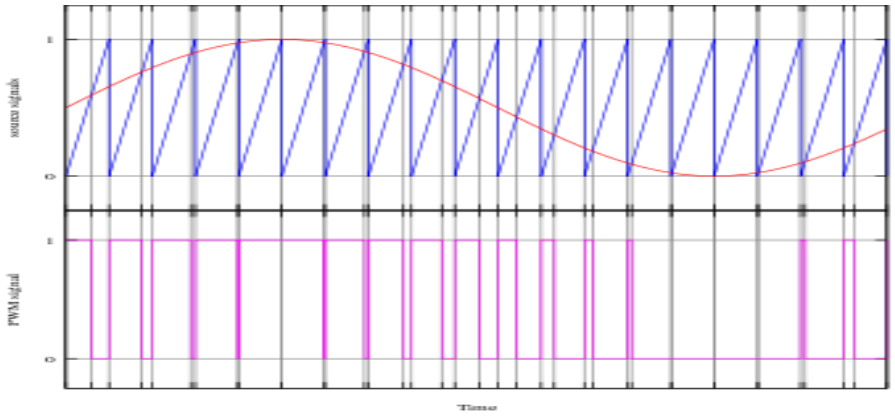


Figure 12: PWM generation using a sine triangular comparison method.

The simplest way to generate a PWM signal is the interceptive method, which requires only a saw tooth or a triangle waveform (easily generated using a simple oscillator) and a comparator. When the value of the reference signal (the red sine wave in figure above) is more than the modulation

waveform (blue), the PWM signal (magenta) is in the high state, otherwise it is in the low state. This is nothing but the modulation of the sine wave on to the carrier signal of saw tooth or triangular wave [9].

2. Delta

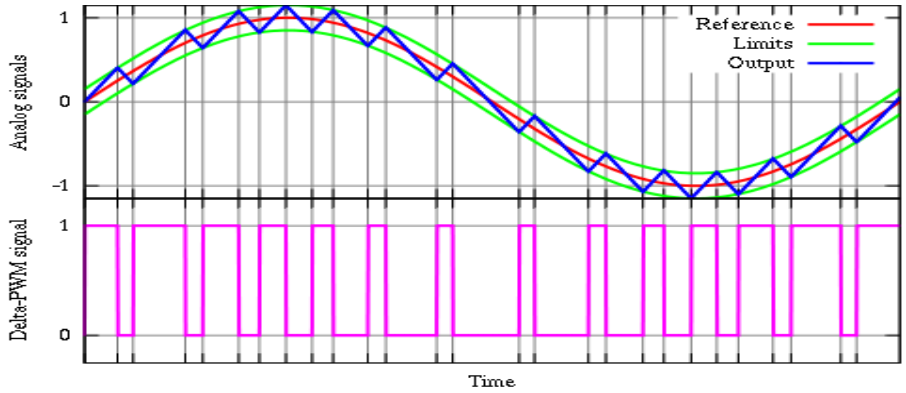


Figure 13: PWM using delta method

In this type of PWM the output signal is integrated over time and is checked with limits which is a reference signal with some off set on both sides of the reference value[6]. Consider the figure above, the red sine wave is the desired output signal. The green envelop is the limits enforced on the output. The idea is simple if the output crosses the limits on positive side turn off the power to the load side. If it falls below the envelop enable the power to the load. This scheme shown here is for a capacitor split DC link. In case of a 4 switch single phase we will need a new reference signal 180 degrees out of phase with the red signal and the load is connected as a floating load between the two half bridges [9].

3. Delta-sigma

This is a small modification to the Delta method, where a error signal is generated given by

$$V_e = V_{ref} - V_{out}$$

and the error signal is integrated and when it hits the limits the output changes polarity. This is more native to the concept of the delta modulation where we are trying to make output track the reference [9].

4. Space Vector Modulation

   This is an algorithmically intensive algorithm used for three phase converters. To keep things short and simple, this method of PWM generation is formed on the basic of the concept that a rotating magnetic field is generated by the 3 phases applied to a motor. Now the field can be expressed with one phasor. This phasor itself can be broken down into 2 resultant vectors.

   Now if we need a specific field in the motor, we can calculate its components and if we find the mapping of these 2 vectors back into the 3 electrical phases we can control the motor voltage. This concept is called the space vector modulation.

   The relation between the 3 phases and the 2 component vectors is given by the dq transforms and its inverse. To put it in strict words Space vector modulation is a PWM control algorithm for multi-phase AC generation, in which the reference signal is sampled regularly; after each sample, non-zero active switching vectors adjacent to the reference vector and one or more of the zero switching vectors are selected for the appropriate fraction of the sampling period in order to synthesize the reference signal as the average of the used vectors [9].

This brings us to the point where we know PWM techniques and we can review the basic DDS algorithm individually.

DDS is the method of producing a analog waveform by generating a time varying signal in the digital domain. The final output if needed is thru a digital to analog converter. Because operations within a DDS device are primarily digital, it can offer fast switching between output frequencies, fine frequency resolution, and operation over a broad spectrum of frequencies [10].

A basic direct digital synthesizer consist of the following basic components: frequency reference, numerically controlled oscillator (NCO), frequency control register and a digital-to analog converter (if needed). The frequency reference provides a time base for the algorithm and the NCO. This is one of the factor that decides the frequency of output.

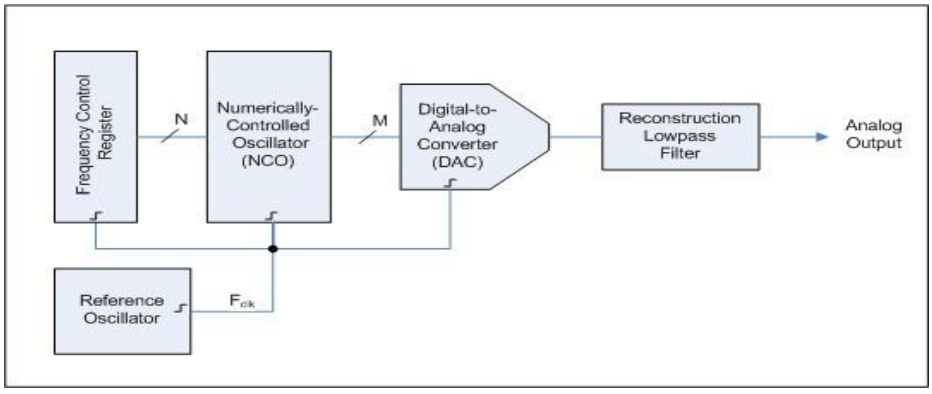


Figure 14: Direct Digital Synthesis block layout. Source: Wikipedia http://en.wikipedia.org/wiki/Direct_digital_synthesizer

The theory of operation can be divided into two parts a discrete time phase generator (the accumulator) and the phase to waveform converter. The implementation of DDS relies on

integer arithmetic. The size of the accumulator (N) is the length of the register in bits. Assuming that the period of the signal is 2Π rad, then the maximum phase represented will be 2^N. Let us say that the phase increment associated with the output frequency be Δ_{ACC}. Thus during one sampling time period the phase increments by Δ_{ACC}. So it will T_O. the time period of the waveform to reach 2^N, the maximum phase. Therefore we get [10]

$$T_O = \frac{1}{F_O} = \frac{2^N T_s}{\Delta_{\mathrm{ACC}}}$$

The above equation can be rewritten to get the output frequency or for the value of Δ_{ACC} required for the desired output frequency. Thus we get the basic operating formula for the DDS algorithm [10].

$$\Delta_{ACC} = F_O \frac{2^N}{F_S} + 0.5$$

By some arithmetic it can be proved that the minimum change in frequency is given by the formula [10]

$$\Delta F_o = \frac{F_S}{2^N}$$

Equipped with the above equations, following steps enlist the process of signal generation:

a) Store the phase data(the amplitude data) of the waveform to be generated in an M bits encoding. This forms a look up table of one full cycle of the waveform.
b) Use the top M bits of the phase accumulator which is of N bits (N>M) to look up into the table for the waveform. This gives the instantaneous value of the waveform.
c) Add the frequency control register to the accumulator to increment the phase of the waveform. This will give you the next sample to be output.

d) To change the output frequency, change the value of the frequency control register. Higher the number, quick will the lookup table be scanned giving higher frequency. For decrease in the output frequency opposite action is followed.

DDS in PWM:

To explain the use of DDS algorithm in the project, let us look at the other building blocks for the PWM method used. The basic idea of any PWM algorithm is to generate the duty cycles which are proportional to the values of the reference sine wave. This when averaged gives the instantaneous values of the output sine waveform.

The inverter uses the Cortex M3 implementation STM32F100RB from ST Microelectronics. This micro controller has the timer one implemented as the advanced timer. Following features of the timer have been used:

-16 bit timer with auto reload. This allows the peripheral to reload its timing values after every cycle of the firing signal.

-Compare match output for PWM generation with Complementary outputs. The basic switching from high to low and other way round is handled by the peripheral module itself with no intervention from the CPU. The complementary outputs allow for the firing of the MOSFETs in one branch of the H bridge.

-Dead Time Generation. One register allows a programmable dead time between the firing of the complementary MOSFETs in a H bridge. This saves the extra effort on the CPU`s side to insert this dead band.

The firing frequency is currently fixed at 4.8KHz. This allows for the easy filtering of the output waveform. With the core frequency of 24MHz and same clock in to the timer, the counting time chip for the timer is

$$\frac{1}{24MHz} = 41.667x10^{-9}s$$

Thus the timer is programmed to count till 5000 counts for the 4.8KHz. The output one is programmed to start as "High" and the output two as "Low". Now for the duty cycle for the PWM, the compare match registers for the two outputs need to be programmed in software to follow the reference sine wave.

This is achieved thru the use of DDS algorithm and linear mapping of the reference value onto the counts for the timer.

The mapping for one half of the output cycle is as follows:

For the positive half cycle the values of the unipolar reference sine wave are greater than 128. By some experimentation the minimum counts when the value is 128 were fixed at 64. The maximum count was fixed at 4890. This gave the mapping from the reference value to the counter value.

Sine reference	Timer counts
128	64
255	4890

Table 1: Positive half mapping

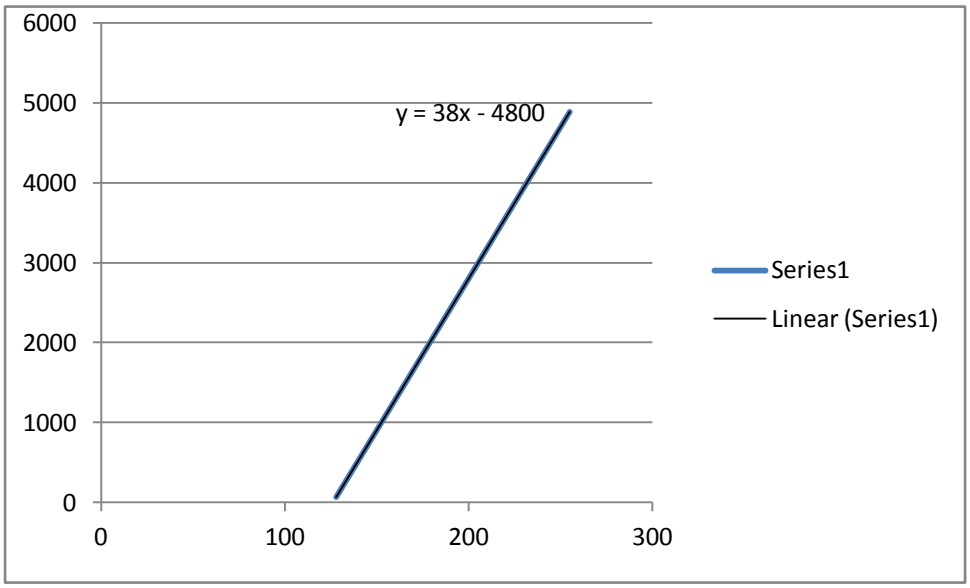


Figure 15: Positive half mapping for the pulse widths

For the negative half the mapping is exactly the inverse as we have the signal reference limits as 128 to 0 and the back to 128. Thus we get

Sine reference	Timer counts
128	64
0	4890

Table 2: Negative half mapping

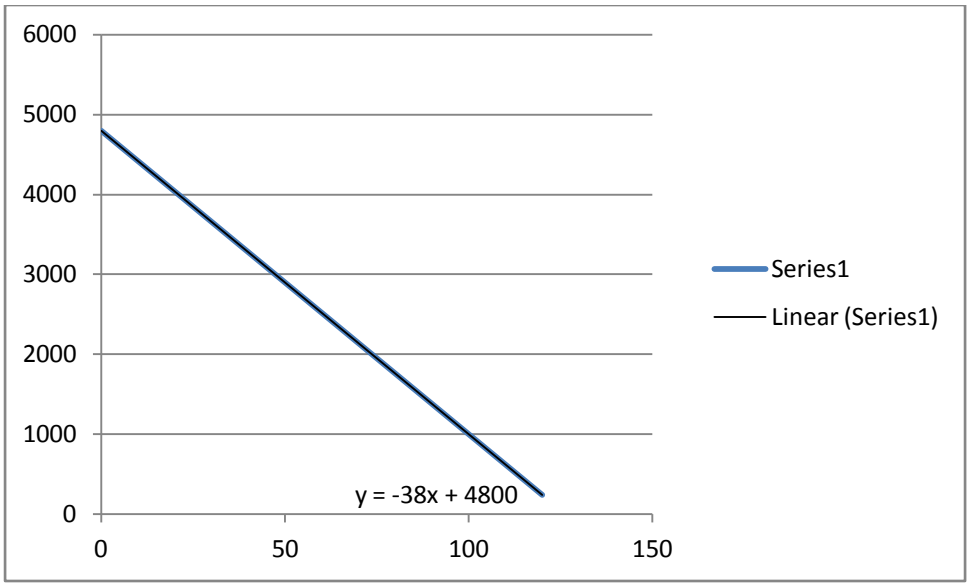


Figure 16: Negative half mapping for the pulse widths

Finally the DDS algorithm is implemented thru the use of the static compile time lookup table by the name "sinewave[256]". The implementation uses a 32 bit accumulator and frequency control register.

Results

Thus using the standard formulas for DDS we get the following values for the designed inverter:

Parameter	Formula	Value
Accumulator		32 bits
Phase		32 bits
Sampling freq		7.98Khz
PWM freq		10.00Khz
$\nabla F_{0\ min}$	$\frac{F_s}{2^N}$	1.858e-6 Hz
$F_{o\ max}$	$\frac{F_s}{2}$	3.99Khz
Number of bits in lookup representation	P	8

Table 3: DDS algorithm parameters

Following are the PWM signals generated at the output pins of the controller.

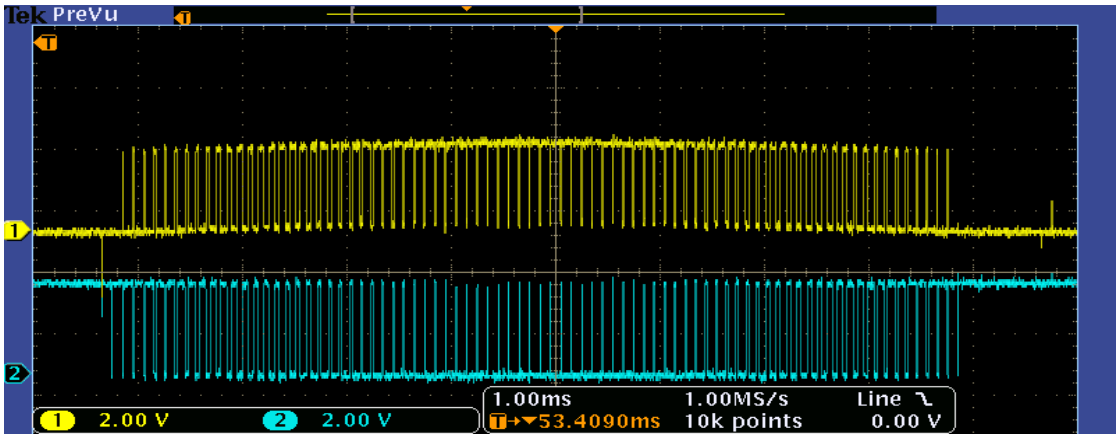


Figure 17: PWMs at 1A and 1B

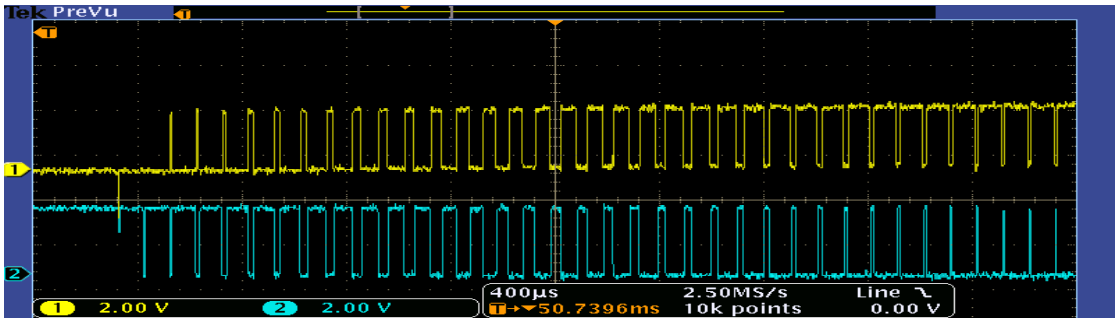


Figure 18: Zoomed PWM for 1A and 1B

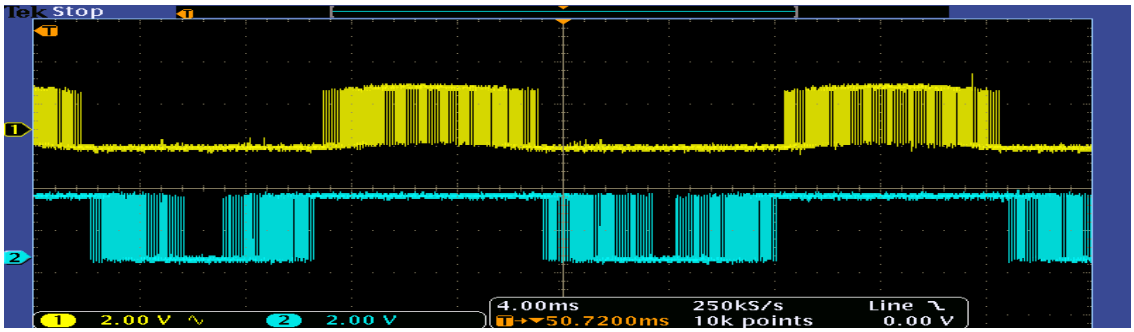


Figure 19: PWM for 1A and 2B

Firmware Ideology

Introduction:

The software developed is for the Cortex M3 STM32F100RB implementation from ST Microelectronics. The software is developed under the GCC compiler using the nonprofit version of the Keil IDE. The following chapter explains the basic software implementation flow charts and ideology which maps hardware to software. Please refer Appendix 3 for the complete source code.

Project Structure

The entire project is divided into the following files:

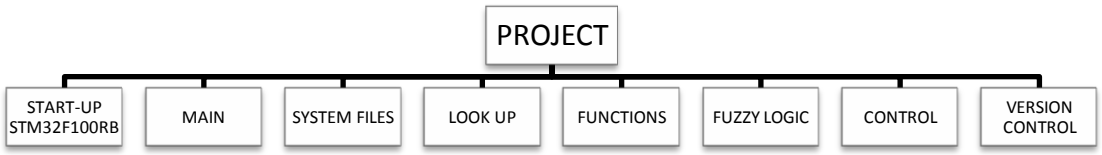


Figure 20: Firmware branching

- Start up file: This is the start up code for the cortex controller.
- Main: Contains the main routine. It initializes the system and then calls the routines for the inverter operation.
- System Files: These contain the Cortex M standard register description and peripheral function implementations.
- Look up: This is the file that has the static look up table for the DDS algorithm.
- Functions: Contains the routines which are called from main, namely the initializations for the peripherals.

- Fuzzy Logic: This file implements the Fuzzy controllers for the frequency and voltage control.
- Control: This file implements the necessary inverter control and fault handling routines.
- Version control: A text document that tells what version number is active and the features implemented or bugs fixed.

Some Concepts Implemented:

1. System Tick Timer:

   The cortex M3 implements a core level timer called the system tick timer. It is a 24 bit timer with a interrupt vector in the core interrupts. This is a down counter with a auto reload feature. This timer is programmed for 12000 counts at the system clock frequency of 24Mhz. This gives a regular interrupt of $5x10^{-4}$s. This is used as a system time base for the control algorithms or any other time synchronization need.

2. Direct Memory Access:

   The medium density chip used (STM32F100RB) implements a DMA controller. This is used to free up the CPU from the time consuming task of reading multiple ADC conversions at almost 500kSps.

   The DMA1 is configured on its channel one to read converted values from ADC1. After every conversion the ADC request the DMA to make a transfer to a predefined memory buffer. This buffer has a fixed position interpretation in the code.

As an example the ADC_data[10] is the buffer used. And its 0^{th} location is the converted value for the DC link voltage. This is fixed by virtue of the hardware connection to the controller`s ADC input pins.

One very keen advantage of the DMA in the used MCU is the fact that it can roll back after fixed number of conversions. Thus we can make sure we write back the values on every conversion scan to the same location where the firmware expects to find it. This feature is called a ring buffer.

3. Analog to Digital Converter:

   The ADC on chip is a 18 channels multiplexed 12bit successive approximation ADC. The ADC is used in the scan mode thus with no CPU intervention the ADC scans continuously the channels for the system health and status updates.

   Of the noteworthy features implemented the two most important are the programmable sample time and the analog watch dog feature.

   The programmable sample time features allows for the channels to be sampled for a duration according to the set parameter. As the channels are multiplexed, the internal sample and hold capacitor needs to be seldom charged and discharged thru its complete range. This introduces errors if done too quick due to partial discharge and pre charge of the capacitor. with the programmed sample for the channels kept high, specially for the slow pole of the frequency charge pump circuit, all errors in conversions are eliminated.

   The other feature implements a interrupt when the values on one of the selected channel go beyond certain value. This is implemented as a sample by sample

check for the load current; checking for all unacceptable transients and over load or short circuit conditions, reducing the stress on a downstream breaker.

4. Hardware Interrupts:

   These are the zero crossing detector outputs from the two feedbacks. For measuring frequency, Timers 6 and 7 were used. These were started on every rising edge of the ZCD, and the time between the 2 interrupts made for the time period of the measured signal. Finally it was inverted and frequency was found. This is then averaged over few cycles.

   For the phase measurement, rising edge of the bus signal is considered as reference and the Timer 2 is triggered. This is stoped at the rising edge of the inverter output`s zcd`s rising edge. This gives the time between the two signal`s zero crossings. At the locked frequency, phase can be easily calculated.

   One issue faced was the false triggering of the interrupt. This was due to the fact that the ISR was completed executing before the rising edge was over and was latched again. This was overcome by disabling the NVIC for the duration of the ISR and waiting for the input to be high before clearing the interrupt request on the line. Following lines of the code implement this feature

   ```
   while(!GPIOB->IDR&(1<<0));// Wait for the input to be stable high
   EXTI->PR|=(1<<0);                    // Clear the interrupt request on the line
   ```

Algorithmic Flow

The following section details the flow of program control from boot up. The code execution and the various algorithms implemented are also detailed out here.

1. Initialize: System, system timer, ADC, GPIO, Timer 1,2,6,7, DMA1 and external interrupts

2. Enable the interrupts at CPU level

3. Initialize all variables.

4. Enable all timers and ADC conversion

5. While 1 loop

-Implement the DDS algorithm.

-Check if the offset register makes the look up table access out of bounds and correct if needed.

-Map the sine reference value on to the PWM signals.

-Convert the ADC values to mapped value.

-Find the peak of output and bus voltage.

-Average the voltages and run the amplitude fuzzy logic

-Average the frequency and run the frequency fuzzy logic

-Re enable the interrupts for external interrupts.

Fuzzy Logic Control: Frequency and Voltage Regulation

Introduction:

Inverter operation needs tight control of frequency and output voltage. This is needed primarily to keep circulating current down to acceptable value and also to keep the quality of service high, with respect to harmonics on bus. While paralleling inverters to a infinite bus, inverters are treated as synchronous generators, which demands tight control over the frequency, voltage and phase. Following chapter describes the fuzzy control implemented for the regulator problem faced.

Literature Study:

The regulator problem in the control system demands that the controlled variable is held at a particular value called the set point. In the problem of inverter paralleling there are two stages to the operation- a. synchronization stage and b. power transfer.

In the 1st stage the inverter tries to control it output such that it is inverting at same frequency, voltage and phase. During this time the set points to the control system are the measurements from the infinite bus. And the system works as a regulator.

Traditionally the 3 basic types of feedback controllers are used:

Proportional Controller:

In this type of controller the output is proportional to the error between the current value of controlled variable and the set point. Thus the digital implementation of the proportional control can be expressed as

$$\Delta c(t) = c(t) + k_p * e(t)$$

where

c(t)=controlled variable (say modulation depth for voltage control or frequency control register for frequency control)

k_p=promotional gain and

e(t)= current value of controlled or plant variable and the set point.

Following graph shows the practical response of a P type controller with some physical limits on its outputs. One final point to note is that few P type controllers implemented have a proportional band rather than a constant gain given by

$$PB = \frac{100\%}{K_p}$$

This allows for the variation in the control action according to the zone of error in its sample space.

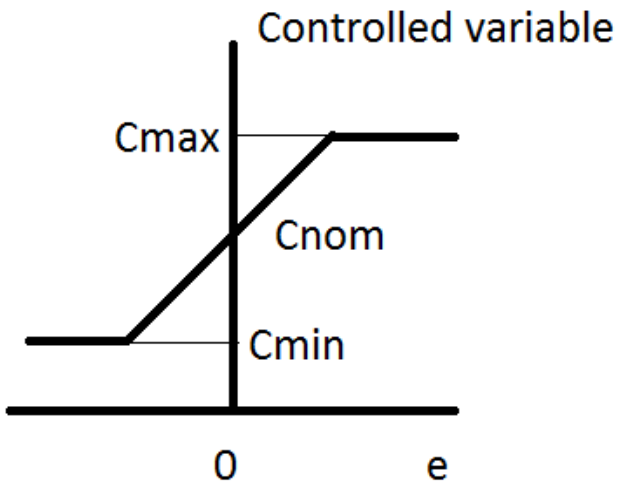


Figure 21: Output of a P controller

Proportional Integral controller:

The output of the controller is the dependent on the integral of the error in the process. The output is given by

$$\Delta c(t) = c(t) + K_p\left(e(t) + \frac{1}{\tau}\int_0^t e(\rho)d\rho\right)$$

where τ is the reset time for the error integration and its unit is in seconds. This type of controller will keep on acting till the time there is error in the system and hence reduces the steady state error to 0 which is not true for the proportional controller. The response of a PI controller to a step error input is as follows

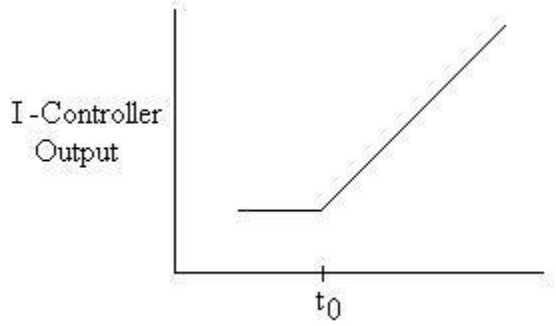


Figure 22: Integral controller`s output

As it is seen, the output of the controller builds up ever τ seconds, thus demonstrating the integral action to the persisting error.

Proportional Integral and Derivative Control

This action is also called the rate action. It tries to anticipate the future behavior of the error mechanism, and controls the output by responding to the rate of change of error. The output of such a controller is given by

$$\Delta c(t) = c(t) + K_p\left(e(t) + \frac{1}{\tau_i}\int_0^t e(\rho)d\rho + \tau_d\frac{de(t)}{dt}\right)$$

where τ_d is the derivative time and has units of time. The major drawbacks of the derivative action is that that output is zero for constant error and that noise can make the controller over compensate and the controller may break into oscillation.

Idea of Fuzzy logic:

Feed back control and state space based methods of design, depend heavily on the model of the system. What happens when the model is not precise or is not available. The model of the inverter system at hand with a LC filter is a 2nd order system and hence already difficult to model and design a control for it. Added to this if the load is non linear or what is most commonly seen, varying and unknown, modeling the system becomes unfeasible for any tangible system implementation. Also most state space methods may involve the use of complex linear algebra to solve for the correct control action, which may not be possible on low end embedded processors which actually implement the control systems at real time.

Thus it is clear there is a sense of ambiguity in even the most well modeled systems. This degree of uncertainty or partial certainty, the way you want to think of it as, needs a tool that in itself not as discreet as the binary system. for example "This is the tallest boy" is a certain event as I am stating the boy is the tallest. But the statement "This boy is very tall" has a sense of open end, which needs a truth validation.

For such representation Zadeh first introduced the mathematical foundation of Fuzzy sets and later Mamdani used it for the first time to control a system. Thus Fuzzy logic control is the method that uses these sets to represent the system variables and the dynamics of it.

As such Fuzzy logic is a design method that uses both intuition and engineering heuristics. It translates the elements of natural language to represent the system behavior and the control action (e.g. "If voltage is very low then increase pulse width to maximum") [11].

A generic fuzzy controller is depicted in the diagram below

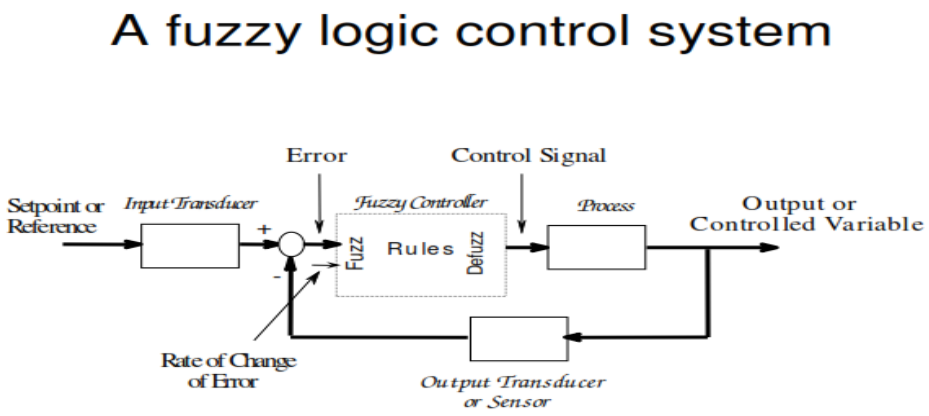


Figure 23: General Fuzzy control block diagram

All the ancillary components are same as a feedback control system. The Fuzzy controller is what is different. It is a usually a 2 input one output system, hence in a sense a MIMO.

It consist of the following components:

1. Fuzzifier: The unit that translate the input signals into fuzzy sets is called a fuzzifier. It also adds a level of confidence/truth level to the input representation telling how much does a instantaneous input value belongs to a fuzzy set [12].
2. Inference Engine: This is the decision making unit for the fuzzy controller. It accepts the fuzzified input variables and then uses the rule book to implement all the rules. It gives out multiple output values with a degree of confidence attached to them [12].
3. Defuzzifier: This unit accepts the multiple outputs and translate them to one crisp control action. The most common method is the moment method [12].

4. Knowledge base: This is the rule book that is translated from lingual definition for the control system`s behavior. This is structured as a "If....ELSE" rule base [12].

The following session discusses in depth the fuzzy control design implemented for frequency and voltage regulation.

The design

The design steps can be documented into 4 broad steps. The same design process is followed for both the voltage and the frequency control. The only difference between the two is the values of the adjustable parameters, which is the beauty of the fuzzy control; it translates lingual description into control algorithm.

1. Fuzzification:

The process starts with the basic step of choosing the input and the output variables. These could be any process variables, and the FLC in itself does not impose any restrictions on the selection process. However as we are facing the problem of regulation and not servo tracking, error in the controlled variable and the rate of change of error were selected to be the two inputs to the fuzzy system. Thus for the frequency controller the two inputs were

$$e_{Fout} = F_{measured} - F_{set\ point}$$

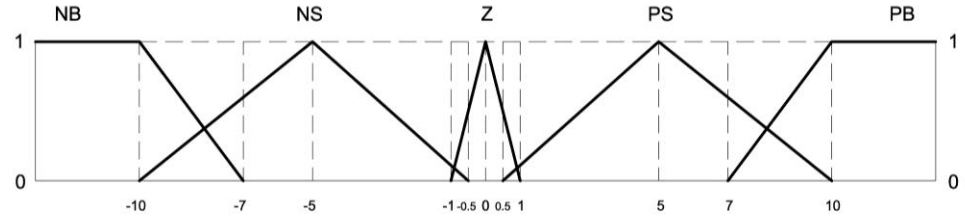
$$\Delta e_{Fout} = e_{Fout\ current} - e_{Fout\ last}$$

Similar process is followed for the voltage control

$$e_{Vout} = V_{measured} - V_{set\ point}$$

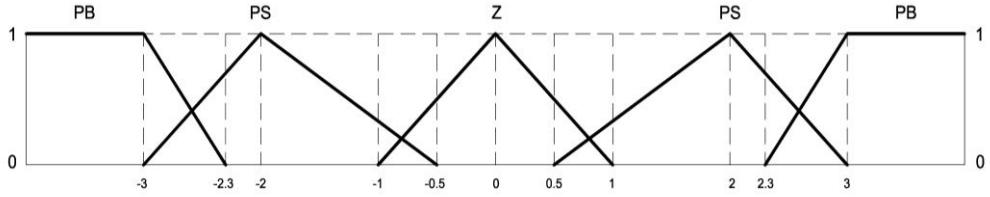
$$\Delta e_{Vout} = e_{Vout\ current} - e_{Vout\ last}$$

With the input variables decided we can now proceed to define the process to convert them to the fuzzy sets. The Fuzzy sets selected were of the type shown below:



FREQUENCY FUZZY SETS

Figure 24: Designed fuzzy sets for the frequency control



VOLTAGE FUZZY SETS

Figure 25: Designed fuzzy sets for the voltage control

As can be seen the sets selected is as follows:

Range of values	Name of fuzzy set
e_{Fout}<=-7	Negative Big
-10<= e_{Fout} <=-0.5	Negative small
-1<= e_{Fout} <=1	Zero
0.5<= e_{Fout} <=10	Positive small
e_{Fout} >=7	Positive big

Range of values	Name of fuzzy set
e_{Vout}<=-2.3	Negative Big
-3<= e_{Vout} <=-0.5	Negative small
-1<= e_{Vout} <=1	Zero
0.5<= e_{Vout} <=2.3	Positive small
e_{Vout} >=2.3	Positive big

Table 4: Fuzzy set definition

The widths of the fuzzy sets and the center points are selected after a judicious guess followed by the trial and error method to tune in on the desired system response. The values are chosen over the appropriate universes of discourse (which is limited by software to 0 to 70 Hz for frequency or -30V to +30V for voltage control).

A quick word on the fuzzy sets in control system. They are usually overlapped to about 25%. The overlapping allows for the smooth transition from the one control action to the other. On the other hand if there is too much overlap, the degree of fuzziness will increase to a level where there is a blur distinction in the control action. For simplicity all the sets can be same: triangular.

As a example of the fuzzification process consider these two input for the error in frequency:

e_{Fout}	$e_{Fout} \epsilon$ Fuzzy sets	Region and equation	Fuzzy value/s
-4	Negative small	(3) truth level=-0.22* e_{Fout}-0.11	0.77
-0.75	Negative small and zero	(3,4) truth level=-0.22* e_{Fout}-0.11 truth level= e_{Fout}+1	0.055 0.25

Table 5: Fuzzy logic implementation example

Thus it can be seen that a particular input can be a member to more than one fuzzy sets. This is the main power of the fuzzy control that the a single input can trigger multiple control action according to the degree of confidence of the membership to a set.

The same process is followed for the Δe_{Fout} variable; with different widths and center points for voltage fuzzy sets.

2. Knowledge Base:

This is the rule base for the fuzzy controller. It documents the controller`s response to different input conditions of the two inputs. It is represented with "IF....ELSE" statements. the fuzzy controllers implemented are both 25 rules based controllers that covers the entire discourse of the input variables. Following diagrams depict the controllers response to the following generic rule

"IF e=(some value) AND Δe=(some value) THEN control action=(some value from 0 to 1)"

ΔFe \ Fe	NB	NS	Z	PS	PB
NB	0.5	0.5	0.1	-0.5	-0.2
NS	1	0.5	0.1	-0.2	-0.5
Z	1	0.5	0	-0.2	-1
PS	1	1	0.1	-0.2	-1
PB	1	1	0.1	-0.5	-0.5

Table 6: Knowledge base for frequency control

ΔVe \ Ve	NB	NS	Z	PS	PB
NB	0.5	0.5	0.1	-0.2	-0.5
NS	1	0.5	0.1	-0.2	-0.5
Z	1	0.5	0.0	-0.2	-1
PS	1	1	0.1	-0.2	-1
PB	0.5	1	0.1	-0.2	-0.5

Table 7: Knowledge base for voltage control

3. Fuzzy Inference Engine:

   This is the block of the controller that actually looks up into the knowledge base and finds out the value of the membership function for each input and the uses some fuzzy set operators to return one truth value for a particular rule. The operation used in the generic rule is "AND". This is finding the minimum of the membership function for each rule of the truth value for the two inputs.

As an example consider the following values

e_{Fout}	$e_{Fout} \epsilon$ Fuzzy sets	Region and equation	Fuzzy value/s
-4	Negative small	(3) truth level=-0.22* e_{Fout}-0.11	0.77

and for the Δe_{Fout}

Δe_{Fout}	$\Delta e_{Fout} \epsilon$ Fuzzy sets	Region and equation	Fuzzy value/s
-3	Negative small	(3) truth level=-0.22* e_{Fout}-0.11	0.55

Then the fuzzy inference will be 0.55 membership for the rule error and delta error are both negative small.

4. Defuzzyfication:

   This is the final step where the fuzzy output is converted to a crisp value for a control action. This is also the step where result from multiple rules "add up" to give a final result. This is achieved by the use of the centroid method. The various outputs can be viewed as a set with the intersection of the individual rule`s output. The centroid is a method to find the average action needed to be taken.

   Thus for an n rule fuzzy system the centroid is calculated as

$$output = \frac{mf(1) * op(1) + mf(2) * op(2) + \cdots + mf(n) * op(n)}{mf(1) + mf(2) + \cdots + mf(n)}$$

   The final control action is scaled as needed with the following expression

$$c(t) = c(t) + (\text{control action} * \text{output})$$

   where mf=membership value to a nth fuzzy set.

For the details of the implemented fuzzy controllers please refer Appendix 3

Result

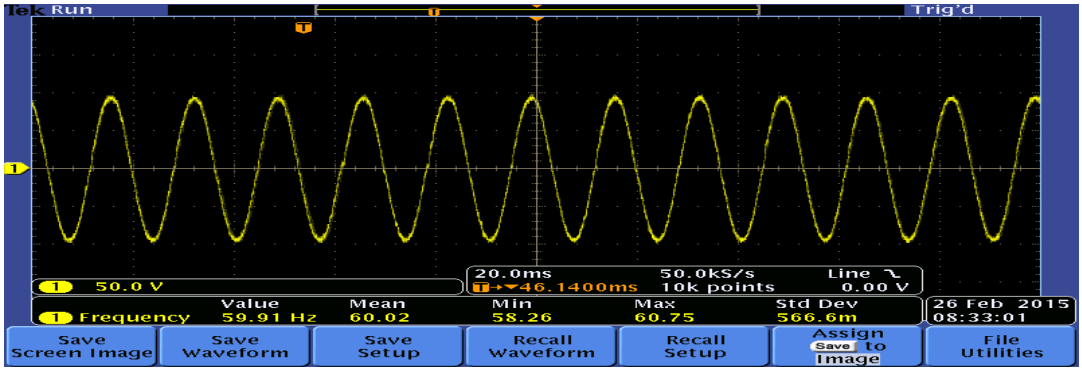


Figure 26: Output of the inverter at High voltage side

It can be seen that tight voltage and frequency regulation can be followed with set points on 60Hz and output voltage of 100V

Concepts In Inverter Paralleling

Introduction:

New control and methods for achieve inverter paralleling in distributed generation systems is the necessity of the hour. With more and more distributed generators being used with grid tie operation, control for the connection of the generator to the grid or micro grid is now much more stringent task more often than not. Primarily the task boils down to control of real and active power flow between the distributed generator and the grid with needed quality of service. The following chapter discusses the basics of the paralleling operation and how the developed inverter is ready as a test platform for further development in the field.

Literature Study:

Consider inverters connected to a infinite bus. This is depicted in the

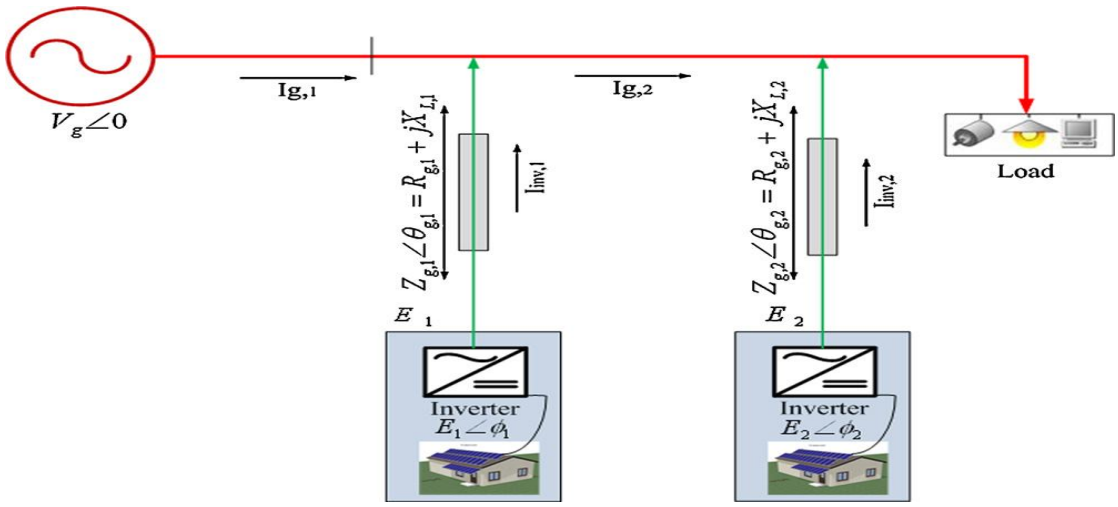


Figure 27: inverters connected in parallel. Source: [10]

For stable operation of the system there needs to be a balance between the generated and consumed active and reactive power. Let`s say P_1 and Q_1 be the real and reactive power transferred from the inverter to the grid. This is given by the expressions [1]

$$P_1 = \left[\left(\frac{E_1 V_g \cos \emptyset_1}{Z_{g,1}} - \frac{V_g^2}{Z_{g,1}}\right)\cos \theta_{g,1} + \frac{E_1 V_g}{Z_{g,1}} \sin \emptyset_1 \sin \theta_{g,1}\right]$$

$$Q_1 = \left[\left(\frac{E_1 V_g \cos \emptyset_1}{Z_{g,1}} - \frac{V_g^2}{Z_{g,1}}\right)\sin \theta_{g,1} - \frac{E_1 V_g}{Z_{g,1}} \sin \emptyset_1 \cos \theta_{g,1}\right]$$

Where E_1 is the inverter voltage, V_g is the grid voltage, $\emptyset_1$ is the phase difference between the grid and the inverter voltage. Thus it is clear that same value of E and V with difference of phase between the two will allow flow of real power. Different amplitude allows for reactive power flow. With both, amplitude and phase, different there is both real and reactive power flow [1].

To get the power flow going, frequency variation allows for change in phase and thus the active power. Similarly with changes in the modulation depth of the inverter, the amplitude can be controlled for reactive power flow.

In parallel operation of inverters, the system is susceptible to load disturbances or disturbances from the other sources. Various control mechanisms are implemented for the control of power follow to counter such disturbance. Key features of such systems are [1]:

- Amplitude, frequency and phase synchronization
- Proper current distribution between inverters according to the capacities
- Flexibility

- hot swap

The conventional inverter paralleling methods can be classified into two main categories

1. Active load share/current distribution:

   In this method a reference is generated for each parallel-connected inverter. The method used to generate a reference signal further distinguishes type of control scheme

   i) Central limit control (CLC): In this method, inverters of same configuration are connected in parallel and every inverter tries to supply the average load current. A DSP based system can track the load current measuring the inductor current for the output filter.

   ii) Master-slave control (MSC): In the MSC configuration one inverter is chosen to be a master and provides a set point to other slave inverters to share the load. This system can fail if the master goes down and in practice, generally one other inverter is appointed as a stand by master to take over when the first one fails.

   iii) Average current sharing (ACS)/distributed logical control (DLC): This method over comes the problem of failing units, which can interrupt normal functioning in the CLC and MSC methods. Here, every module has a distributed control and measurement. Current control mode is used and the power flow is controlled by measuring and matching the active power of the system.

   iv) Circular chain control (3C): In this method every successive module tracks the current of the previous one to share the load current; and the first one tracks

the last module to form a chain. A coordinated strategy can be used to reduce the impact of circulating currents between the inverters due to unbalanced converters. This technique is further improved for non linear loads by sharing the harmonics currents too [1].

2. Droop control:

   In this method the output voltage of the inverters follow a amplitude and frequency droop curve to allow the load sharing as the load current increases. The basic idea of droop control is to control the output of the inverters without any interconnection or communications between the inverters.. Thus it is also called the 'wireless' control. This is achieved by measuring only the output of the inverter and the bus parameters. This method is suitable for the connection of inverters on bus which are separated by vast distances and communication between them is not possible or is not cost effective.

   In this mode of operation, inverters are run in voltage mode, and the phase(frequency) and the voltage amplitude are the controlled parameters. The controller introduces a droop in the frequency and the amplitude with the increase of active and reactive power respectively.

   The output of the inverter is predominantly inductive and the equations stated for P and Q before can be simplified to the following formula

$$P = \frac{EV}{X}\sin \emptyset$$

$$Q = \frac{EV\cos \emptyset - V^2}{X}$$

Where X is the output inductive reactance of the inverter, and ∅ is the phase angle between the inverter and the infinite bus. Thus it can be stated that the active power is dependent on the phase angle and reactive power on output amplitude. This is used by the controller by adding the droop to control power flow by using the following droop formula

$$\omega = \omega^* - m(P_{0i} - P_i)$$

$$E = E^* - n(Q_{0i} - Q_i)$$

where ω^* and E^* are the values of the angular frequency and voltage at no load and P_{0i}, P_i, Q_{0i} and Q_i represent the rated power and the actual powers drawn from the inverter. m and n are the droop coefficients that actually affect the amount of control action taken by each inverter. Higher the droop coefficients higher is the power flow and the cost of degrading voltage and frequency regulation. This fact can be over looked till the time the deviations for frequency is within 2% and for voltage within 5% [1]. The following image shows the droop characteristics in general

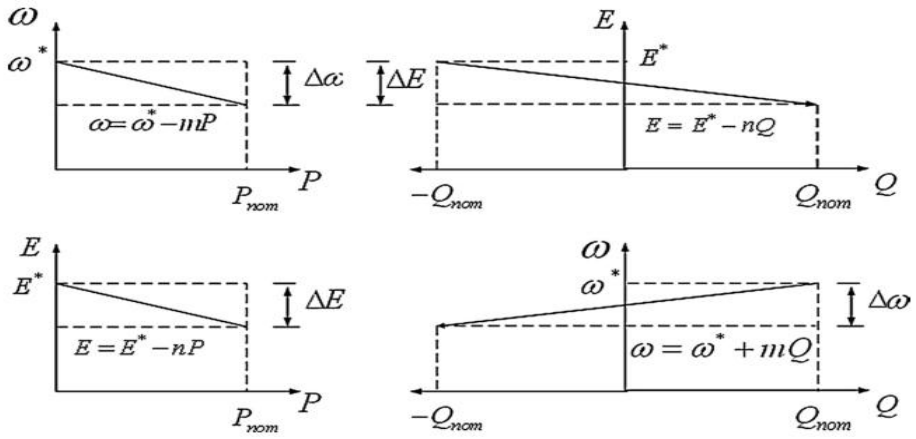


Figure 28: Standard Droop curves. Source [1]

Finally a simple topology to find out the needed reference signal to the PWM generator is given in the diagram below

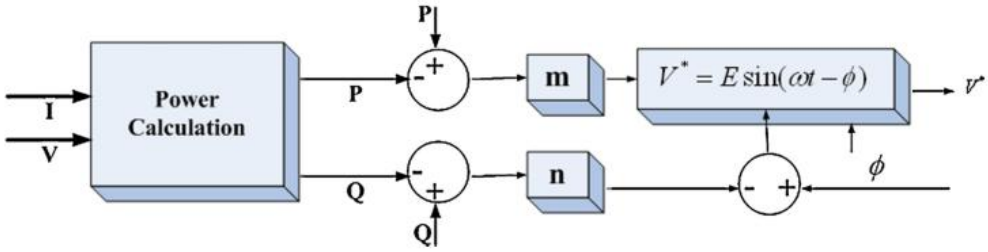


Figure 29: Droop control block diagram. Source: [1]

Platform Readiness:

The following section details how the developed platform is geared for the implementation of the algorithms mentioned in the preceding section. As mentioned the inverter paralleling problem is correct load sharing without large circulating currents. To achieve this end, the controlled parameters of the inverter are voltage amplitude, frequency and phase with respect to the grid.

The use of the DDS algorithm coupled with the fuzzy controller, makes the digital control of these parameter very easy. Following modes of actions are followed for each of the controlled variable:

1. Frequency: Frequency is controlled via the frequency control register in the DDS algorithm. This is a numeric integer value. Higher the value, higher is the frequency and otherwise for lower frequency. Thus change in this value is what is needed for the change of frequency. No other changes are needed for the PWM to react to this. This change in frequency is a native output the fuzzy controller are it will return the needed change of the control variable. Thus the output is simply

added to the frequency control register for the needed directed change of frequency. Following section of firmware handles this:

```
error=Fout_avg-(Fbus);                    //find the error in the output frequency
test=FIE_freq(error,(error-error_last));  //evaluate the control action
t=Phase;                                  //read current frequency control register
t+=(int)5000*test;                        //scale the %control action to a numeric
value
if(t!=0)
Phase=t;                                  //if the new sum does not result a 0 value for
frequency                                 control register then add
```

2. Voltage Amplitude: This is achieved thru the modulation depth and the implemented fuzzy control. The DDS algorithm gives the instantaneous value of reference sine value. This is them mapped to the pulse width control register. Here the modulation depth index is used as a value between 0.0 to 1.1 to allow the change in the multiplicative term. As the value of the reference signal is reduced by 1, the amplitude is reduced at the output, as the PWM width reduces.
   This is achieved by the following piece of firmware

   ```
   Am+=0.1*FIE_volt((Vpeak-Vpeakset),Verror_last);
   Verror_last=Vpeak-18;
   ```

   The explanation is similar for the frequency control except for the zero check.

3. Phase control: This is achieved by adding a constant value to the index of the look up table, or the phase accumulator. The method followed in the implemented is

$$sine\ value = @(index + offset)$$

This allows for a shift in just the phase of the waveform and does not affect the frequency or amplitude of the output. The output is first synchronized to the bus and then again shifted as needed by the P set point. The firmware for this is involved, please refer the appendix 2 for it.

Results

Following are the results for synchronization of the inverter to the grid. The first image shows that the grid`s zcd (yellow) is not in sync with the output voltage zcd (blue). The second image shows a latch on the bus voltage.

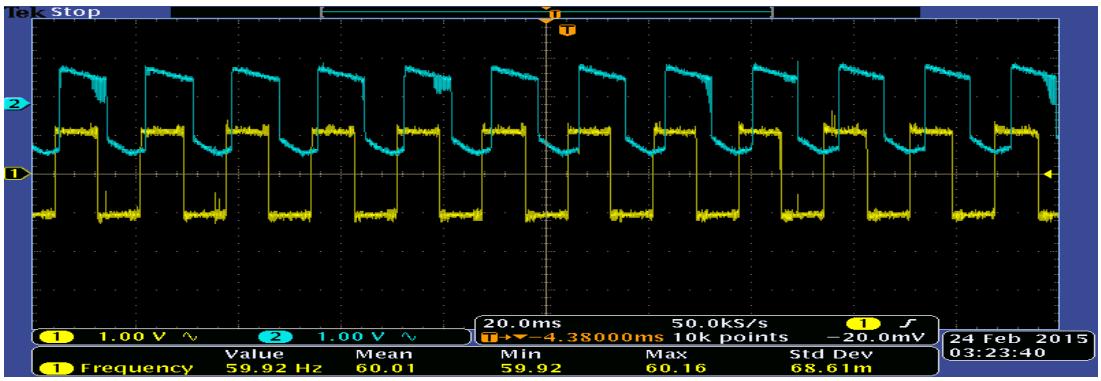


Figure 30: No synchronization in the bus and inverter voltage.

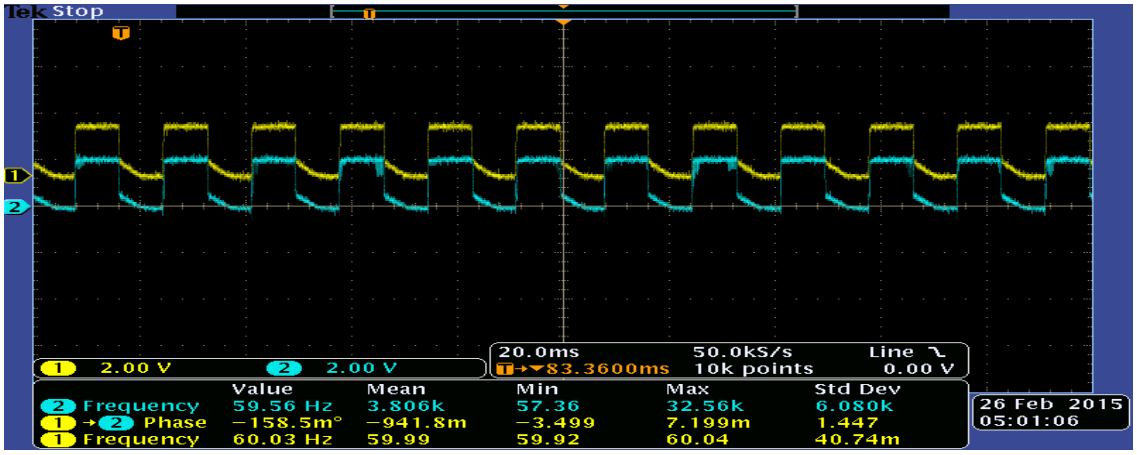


Figure 31: Synchronization achieved

Conclusion

A inverter platform was successfully developed and tested. As needed the platform was open source and cheap. It is easy to replicate when needed in bulk as all components used are off the shelf. The Direct Digital Synthesis was used successfully to implement the PWM firing algorithm. The fuzzy control for voltage and frequency was implemented and tested. The relation to the power flow in inverter paralleling was studied and verified that this platform can be used for practical implementation of the same.

Future Work

Following can be the most logical steps carrying forward from this point in project:

1. Fabricate a PCB.

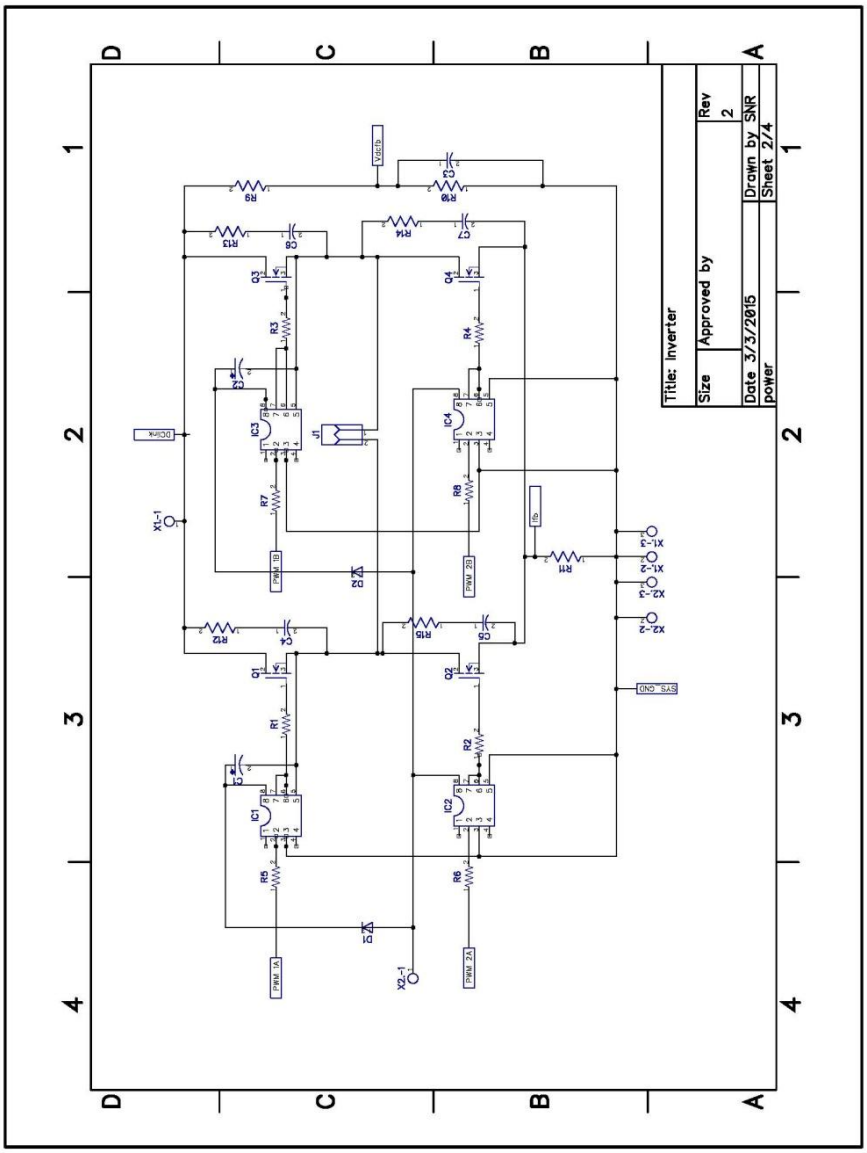
2.Develope the top level control for P and Q power flow.

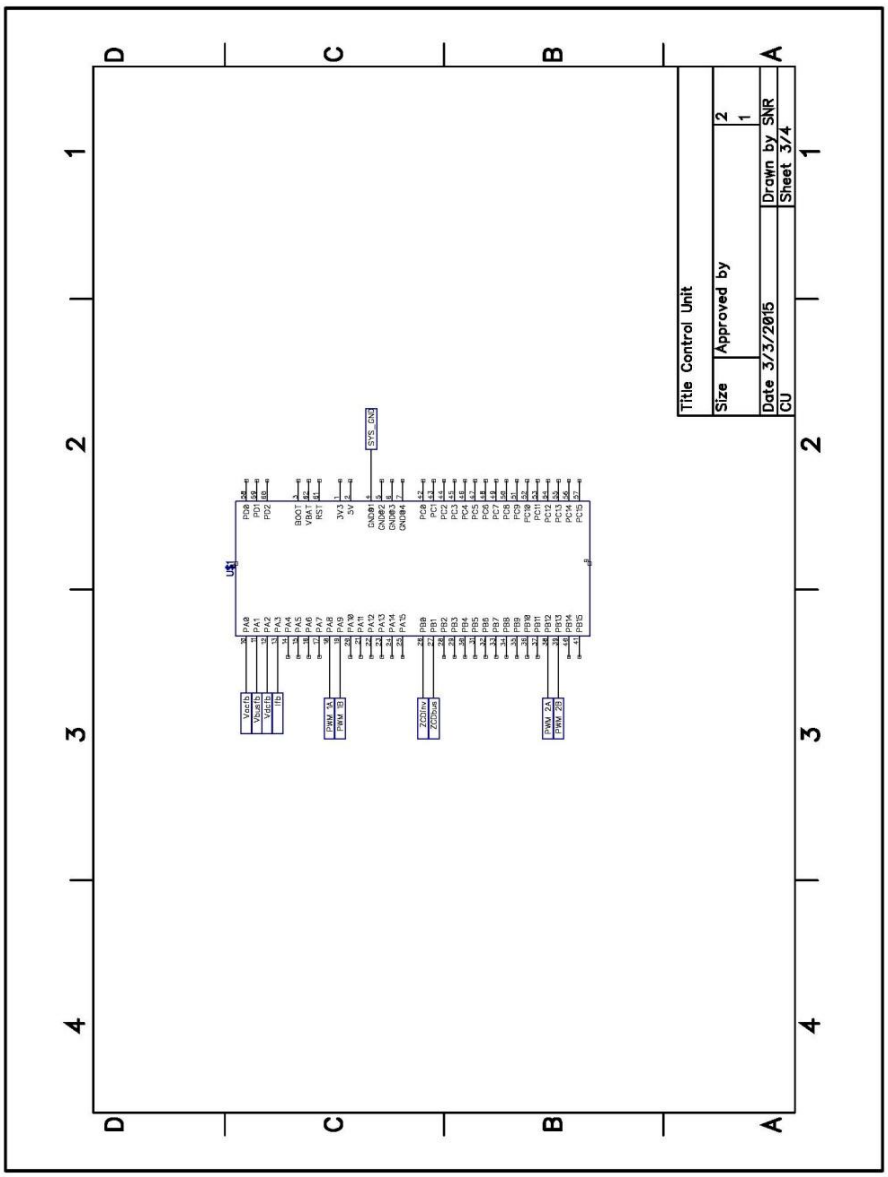
3. Test and feedback the results to the platform if any change is needed.

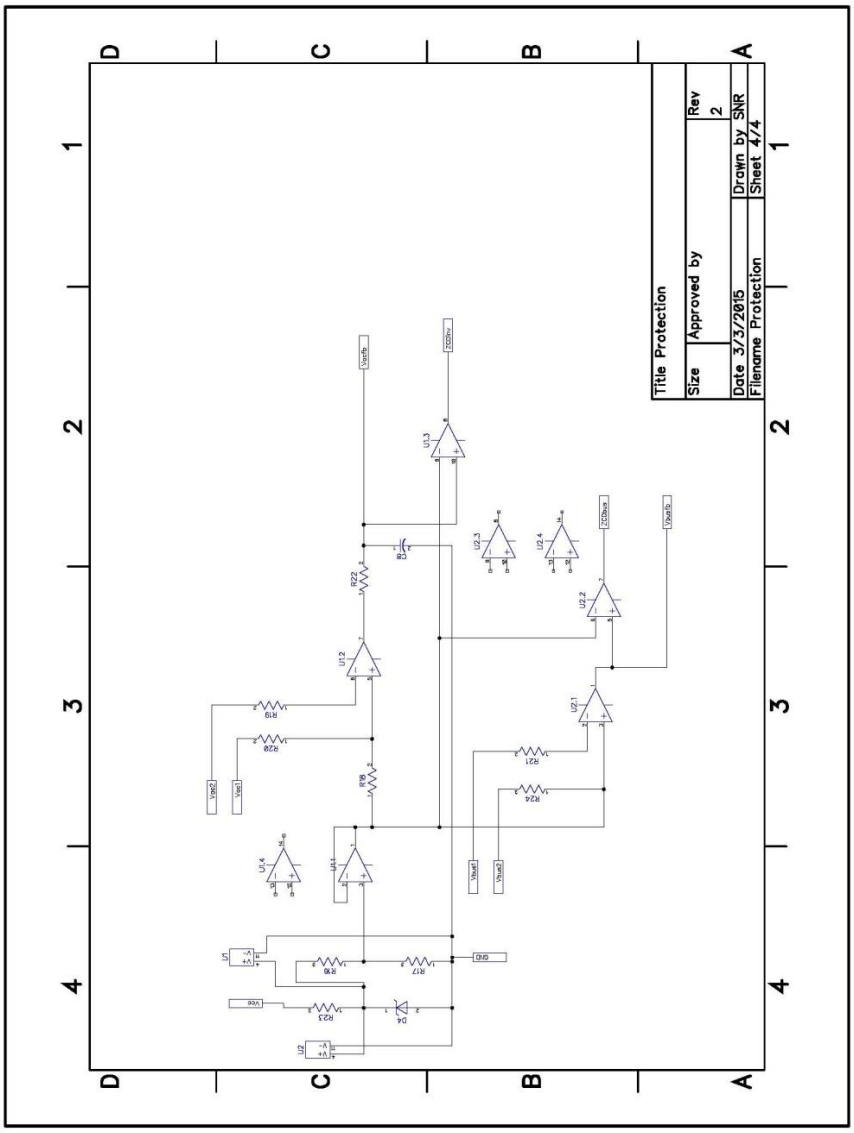
4. As a low priority, other hall effect current sensors can be probed.

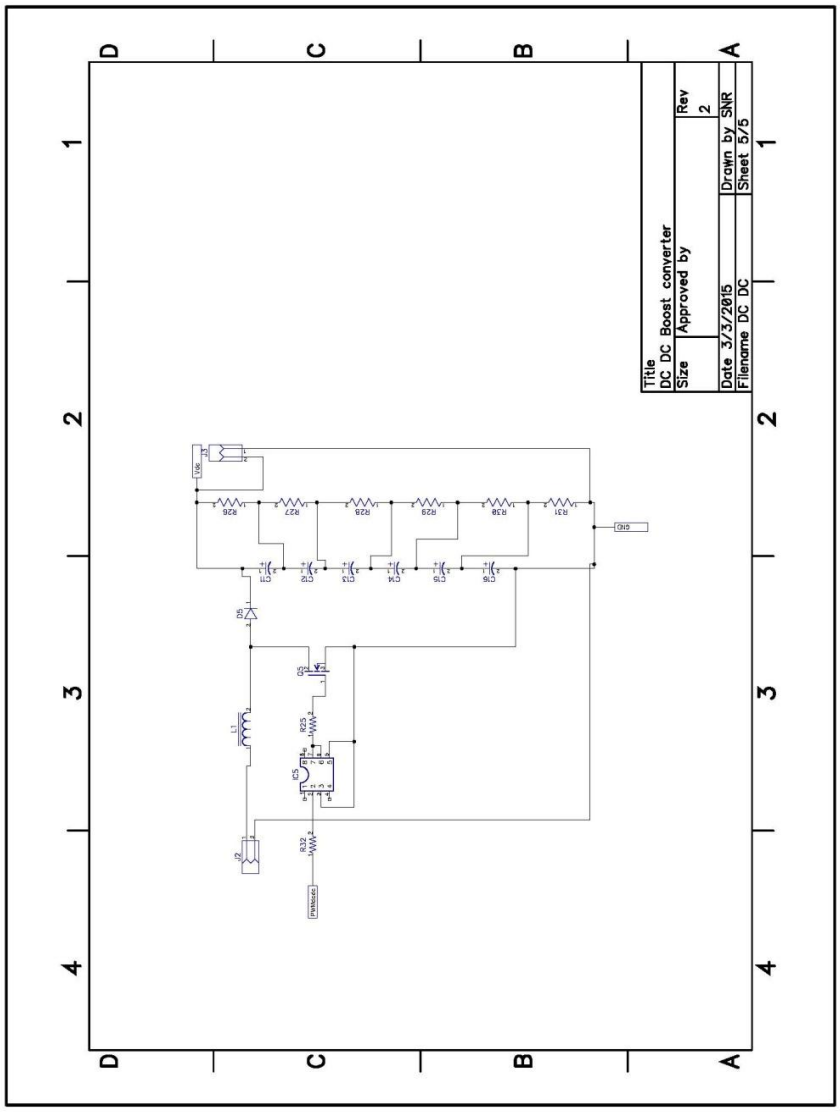
Appendix 1

-


Title: Inverter
Size
Approved by
Rev 2
Date 3/3/2015
Drawn by SNR
power
Sheet 2/4
Vdrib
R9
R10
C3
R13
R14
C7
Q3
Q4
R3
R4
C6
IC3
IC4
J1
DCin+
X1-1
R7
R8
PWM 1B
PWM 2B
D2
R11
X1-3
X1-2
X2-3
X2-2
R12
C4
R15
C5
Q1
Q2
SAS GND
R1
R2
C1
IC1
IC2
R5
R6
D1
PWM 1A
PWM 2A
X2-1


Title Control Unit
Size
Approved by
Date 3/3/2015
CU
Drawn by SNR
Sheet 3/4
A
B
C
D
1
2
3
4
U1
SYS_OSC
PWM_1A
PWM_1B
PWM_2A
PWM_2B
ZCDriver
ZCBack


Title Protection
Size
Approved by
Rev
2
Date 3/3/2015
Drawn by SNR
Filename Protection
Sheet 4/4


Title
DC DC Boost converter
Size
Approved by
Rev
2
Date 3/3/2015
Drawn by SNR
Filename DC DC
Sheet 5/5
GND
PWMout

	RefDes	Name	Value	Quantity
1	C1	CPOL-USE:	1000u	1
2	C2	CPOL-USE:	1000u	1
3	C3	CAP100	1.2n	1
4	C4	CAP		1
5	C5	CAP		1
6	C6	CAP		1
7	C7	CAP		1
8	C8	CAP100		0
9	C11	CAP1000A	470u	1
10	C12	CAP1000A	470u	1
11	C13	CAP1000A	470u	1
12	C14	CAP1000A	470u	1
13	C15	CAP1000A	470u	1
14	C16	CAP1000A	470u	1
15	D1	1N4007	1N4007	1
16	D2	1N4007	1N4007	1
17	D4	1N4728A	3.3	0
18	D5	DIODE		1
19	IC1	DIL8S	HCPL3120	1
20	IC2	DIL8S	HCPL3120	1
21	IC3	DIL8S	HCPL3120	1
22	IC4	DIL8S	HCPL3120	1
23	IC5	DIL8S	HCPL3120	1
24	J1	OSTTV021150		1
25	J2	1725656		1
26	J3	1725656		1
27	L1	INDI	660u	1
28	Q1	IRF540	IRF540	1
29	Q2	IRF540	IRF540	1
30	Q3	IRF540	IRF540	1
31	Q4	IRF540	IRF540	1
32	Q5	IRF540	IRF540	1
33	R1	R-US_020	47	1
34	R2	R-US_020	47	1
35	R3	R-US_020	47	1
36	R4	R-US_020	47	1
37	R5	R-US_020	300	1
38	R6	R-US_020	300	1
39	R7	R-US_020	300	1
40	R8	R-US_020	300	1
41	R9	RES500	100k	1
42	R10	RES500	12k	1
43	R11	RES500	0.04	1
44	R12	RES400		1
45	R13	RES400		1
46	R14	RES400		1

Bibliography

[1] M. B. a. M. C. Shafiuzzaman Khan Khadem, "Parallel Operation of Inverters and Active Power Filters in Distributed Generation System: A Review," *Renewable and Sustainable Energy Review,* 2011.

[2] *IEEE std 1547, IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems,* 2003.

[3] G. M. Masters, Renewable and Efficient Electric Power Systems, Hoboken: IEEE-Wiley, 2004.

[4] Wikipedia, "FACTs," [Online]. Available: http://en.wikipedia.org/wiki/Flexible_AC_transmission_system. [Accessed Jan 2015].

[5] Wikipedia, "Static Compensator," [Online]. Available: http://en.wikipedia.org/wiki/STATCOM. [Accessed Jan 2015].

[6] M. B. a. M. C. Shafiuzzaman Khan Khadem, "Parallel Operation of Inverters and Active Power Filters in Distributed Generation Systems: A Review.," 2011.

[7] P. Muhammad H. Rashid, Power Electronics Handbook: Devices, Circuits and Applications Third Edition, Elsevier.

[8] Texas Instruments, "Snubber Circuit Design - Practical Tips," [Online]. Available: http://www.ti.com/ww/en/analog/power_management/snubber_circuit_design.html.

[9] "Pulse-width modulation," 02 2015. [Online]. Available: http://en.wikipedia.org/wiki/Pulse-width_modulation. [Accessed 02 2015].

[10] L. Cordesses, "Direct Digital Synthesis:A Tool for Periodic Wave Generation (Part 1)," *IEEE SIGNAL PROCESSING MAGAZINE,* 2004.

[11] Wikipedia, "Fuzzy control system," [Online]. Available: http://en.wikipedia.org/wiki/Fuzzy_control_system. [Accessed Jan 2015].

[12] P. M. B. Khalid, "Fuzzy Control System Design," Unversity of Teknology Malaysia, Cairo.