

FPGA Controlled 3 Phase VSI using SVPWM

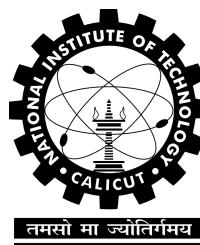
B.Tech. Major Project Report

*Submitted in partial fulfillment of
the requirements for the award of the degree of*

Bachelor of Technology
in
Electrical and Electronics Engineering

By

ABINAV SUBRAHMANIAN KRISHNA	B130784EE
GOPAVAJJULA KS ADITYA	B130346EE
GOVARDHAN ADITYA NAGESH	B130914EE
LOKESH SANCHETI	B130984EE



तमसो मा ज्योतिर्गमय

Department of Electrical Engineering
NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
Calicut, Kerala, India – 673 601

ACKNOWLEDGEMENT

During the course of our Major Project work on "**FPGA Controlled 3 Phase Voltage Source Inverter using Space Vector Pulse Width Modulation**" we received a lot of help, encouragement and invaluable guidance from our fellow students and faculty members alike. We would like to thank Dr. Jagadanand G. for agreeing to be our Project Guide and encouraging us to take this project. We thank him profusely for giving us guidance at all times. We would also like to thank other members of the evaluation panel for their invaluable suggestions and words of wisdom. We extend our deepest gratitude to our Head of Department, Dr. S Ashok for allowing us to use the department resources.

ABINAV SUBRAHMANIAN KRISHNA (B130784EE)

GOPAVAJJULA K S ADITYA (B130346EE)

GOVARDHAN ADITYA NAGESH (B130914EE)

LOKESH SANCHETI (B130984EE)

DECLARATION

"We hereby declare that this is submission of our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning except where due acknowledgement has been made in the text."

ABINAV SUBRAHMANIAN KRISHNA (B130784EE)

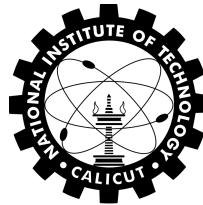
GOPAVAJJULA K S ADITYA (B130346EE)

GOVARDHAN ADITYA NAGESH (B130914EE)

LOKESH SANCHETI (B130984EE)

Place :Calicut

Date :April 28, 2017



तमसो मा ज्योतिर्गमय

CERTIFICATE

*This is to certify that the Major Project report entitled **FPGA Controlled 3 Phase VSI using SVPWM** is a bonafide record of the seminar presented by **Abinav Subrahmanian Krishna (B130784EE)**, **Gopavajjula KS Aditya (B130346EE)**, **Govardhan Aditya Nagesh (B130914EE)**, **Lokesh Sancheti (B130984EE)** in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in **Electrical and Electronics Engineering** from National Institute of Technology Calicut, Kozikode for the year 2017.*

तमसो मा ज्योतिर्गमय

Dr. Jagadanand G
Project guide

Dr. Ashok S
Professor and Head

Abstract

Inverters produce an AC output waveform from a DC source. Three-phase Voltage Source Inverters (VSIs) are used in applications that require sinusoidal voltage waveforms of variable magnitude as well as variable frequency. These inverters are controlled using control techniques like Sinusoidal PWM (SPWM) and Space Vector PWM (SVPWM). SVPWM is a sophisticated control technique for generating a fundamental sine wave that provides a higher voltage to the motor and lower total harmonic distortion (THD). Such sophisticated control algorithms become easier to be implemented with Field Programmable Gate Arrays(FPGAs) as one of the fundamental advantage is the freedom of parallelism as different parts of FPGA can be configured to perform independent functions simultaneously.

The objective of the project is to design and implement a three phase voltage source inverter using SVPWM control algorithm. The two level inverter topology is implemented on a Printed Circuit Board(PCB) and Field Programmable Gate Array(FPGA) is used for implementing SVPWM control algorithm. The control circuit is designed using an innovative methodology which significantly reduces the complexity of SVPWM implementation. The designed system is tested on a three phase induction motor.

The designed system is simulated using MATLAB Simulink software and obtained results are tested against expected output and verified to be consistent. Future scope of the project is discussed as well.

CONTENTS

Acknowledgement	i
List of Figures	i
List of Tables	ii
List of Abbreviations	iii
1 INTRODUCTION	1
1.1 Three Phase Inverters	1
1.2 Significance	3
1.3 Outline	3
2 LITERATURE SURVEY	5
2.1 Introduction	5
2.2 Literature Survey	5
3 SPACE VECTOR PWM	7
3.1 SVPWM Principle	7
3.2 Analysis of SVPWM Technique	8
3.3 Superior Performance of SVPWM	13
3.4 Conclusion	16
4 SYSTEM OVERVIEW AND SIMULATION	17
4.1 Control Circuit Simulation	17
4.2 Power Circuit Simulation	20
5 CONTROL CIRCUIT	22
5.1 Field Programmable Gate Array (FPGA)	22
5.2 Control Circuit Design	24
5.3 Control Circuit Implementation	27
5.4 Conclusion	28
6 POWER CIRCUIT	30
6.1 Voltage Regulator Circuit	30
6.2 Optocoupler Isolation	31
6.3 High Power Inverter Circuit	32
6.4 Snubber Circuit	33
6.5 Turn On and Turn Off of IGBT	34

6.6	Miller Capacitance	36
6.7	Other Design Considerations	37
6.8	Conclusion	37
7	OBSERVATION AND RESULTS	40
7.1	Initial testing	40
7.2	Testing with 3 phase Induction motor	42
7.3	Conclusion and future scope	45
A	APPENDIX - FPGA code	46
A.1	Modulating signals	46
A.2	Submodule-1	53
A.3	Submodule-2	54
A.4	Triangle Wave Generator	55
A.5	Comparator	57

REFERENCES

LIST OF FIGURES

1.1	Three Phase Inverter	1
1.2	Pulse Width Modulation	2
3.1	Voltage Source Inverter	7
3.2	Space Vector Transformation	8
3.3	Switching States and Voltage Vectors	9
3.4	Volt-Second Balance	10
3.5	Dwell Times - Method I	11
3.6	Dwell Times - Method II	12
3.7	Switching Pattern for Sector I	13
3.8	Additional DC Utilization by SVPWM	14
3.9	Current Ripple Vector over a subcycle	14
3.10	Current Ripple along q-axis and d-axis over a subcycle	15
3.11	THD vs. V_{REF}	15
4.1	Control Circuit Blocks	18
4.2	Control signals before delay circuit	18
4.3	Control signals after delay circuit	19
4.4	Carrier Wave	19
4.5	Power Circuit	20
4.6	Pole Voltages	20
4.7	Line Voltages	21
4.8	Phase Voltages	21
4.9	Speed vs Time and Torque vs Time	21
5.1	Basic Architecture of FPGA	23
5.2	Block Diagram of DE1-SoC Development Board	24
5.3	Modified Modulating Waves	25
5.4	Control Circuit Block Diagram	26
5.5	Delayed Circuit Logic Waveforms	27
6.1	Voltage Regulator Circuit	30
6.2	Selection of R_L and C_L	31
6.3	Optocoupler IC TLP250	32
6.4	High Power Inverter Circuit	33
6.5	MOSFET and IGBT Turn-off Characteristics	34
6.6	Turn-off Pattern of IGBT	35
6.7	Effect of R_G on Turn-off Losses	35

6.8	Miller Capacitance Effect	36
6.9	Resistance for Miller Capacitance Mitigation	36
6.10	Inverter Circuit Diagram	38
6.11	Power Circuit PCB Layout	39
7.1	Initial Lab Setup	40
7.2	Line Voltage at 30V DC Supply, No Load	41
7.3	Line Voltage at 50V DC Supply, No Load	41
7.4	Schematic of inverter setup for higher voltages	42
7.5	Setup for running motor	43
7.6	Voltage and current waveforms for 3 phase induction motor running at no load	43
7.7	Voltage and current waveforms for 3 phase induction motor running with high load	44
7.8	Voltage and current waveforms for 3 phase induction motor running with medium load	44

LIST OF TABLES

3.1	Switching States and Voltage Vectors	10
3.2	Forward and Reverse Sequences in Six Sectors	13
5.1	FPGA Resource Utilization	29

LIST OF ABBREVIATIONS

- ASIC - Application Specific Integrated Circuit
- CLB - Configurable Logic Block
- FPGA - Field Programmable Gate Array
- HDL - Hardware descriptive Language
- IGBT - Insulated Gate Bipolar Transistor
- MCU - Microcontroller Unit
- MOSFET- Metal oxide Semicondcutor Field Effect Transistor
- PCB - Printed Circuit Board
- RMS - Root Mean Square
- SPWM - Sinusoidal Pulse Width Modulation
- SRAM - Static Random Access Memory
- SVPWM - Space Vector Pulse Width Modulation

Chapter 1

INTRODUCTION

1.1 Three Phase Inverters

Because of advances in solid state power devices and microprocessors, variable speed AC Induction motors powered by switching power converters are becoming more and more popular. Switching power converters offer an easy way to regulate both the frequency and magnitude of the voltage and current applied to a motor. As a result much higher efficiency and performance can be achieved by these motor drives with less generated noises.

The energy that a switching power converter delivers to a motor is controlled by Pulse Width Modulated (PWM) signals applied to the gates of the power transistors. PWM signals are pulse trains with fixed frequency and magnitude and variable pulse width. There is one pulse of fixed magnitude in every PWM period. However, the width of the pulses changes from period to period according to a modulating signal. When a PWM signal is applied to the gate of a power transistor, it causes the turn on and turn off intervals of the transistor to change from one PWM period to another PWM period according to the same modulating signal.

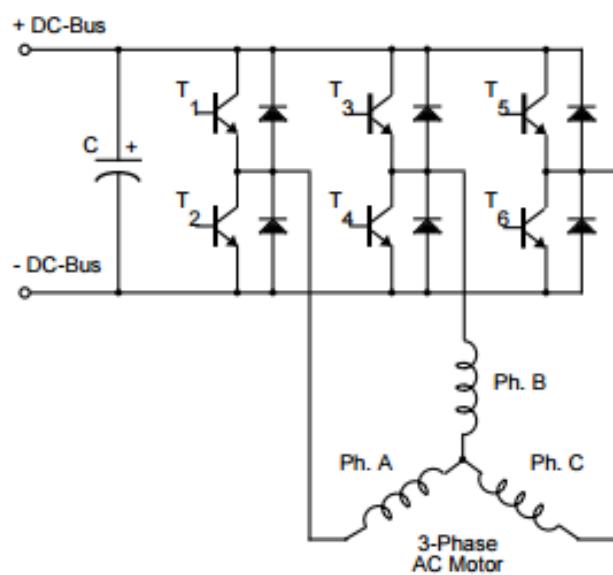


Fig. 1.1 Three Phase Inverter

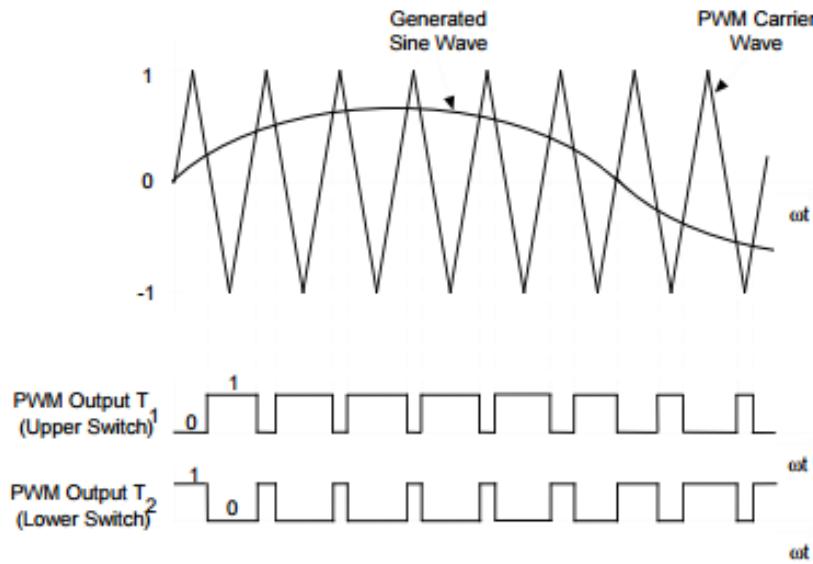


Fig. 1.2 Pulse Width Modulation

In adjustable speed applications the AC motors are powered by inverters. The inverter converts a DC power to AC power at required frequency and amplitude. The typical 3-phase inverter is illustrated in Fig. 1.1. The inverter consists of three half-bridge units where the upper and lower switch are controlled complementarily, meaning when the upper one is turned on, the lower one is turned off and vice versa. The output voltage is created by a Pulse Width Modulation (PWM) technique where an isosceles triangle carrier wave is compared with a fundamental frequency sine modulating wave, and the natural points of intersection determine the switching points of the power devices of a half bridge inverter. This technique is shown in Fig 1.2. The three phase voltage waves are shifted 120° to each other and thus a 3-phase motor can be supplied.

The most popular devices for motor control applications are Power MOSFETs and IGBTs. A Power MOSFET is a voltage controlled transistor designed for high frequency operation. It has low voltage drop and thus low power losses. However, the saturation and temperature sensitivity limit the MOSFETs application in high power circuits. An Insulated Gate Bipolar Transistor (IGBT) is a bipolar transistor controlled by a MOSFET on its base. The IGBT requires low drive current, has fast switching time and is suitable for high switching frequencies.

1.2 Significance

In 1958, Solid state power devices known as SCRs were developed which led to the availability of DC drives. In the early 1960's, the cost effectiveness of SCRs got improved which led to the better understanding of these applications. In late 1960's, Analog control circuitry using digital control and firing circuitry were developed. Development of phase locked loops for synchronization improved line noise immunity allowing DC drives to operate better. During the 1970's, large scale integrated circuit (LSI) technology was developed. Custom integrated circuitry improved the reliability and cost of current circuitry. Before 1985, SCR's/GTO's using six step technology led to the development of drives which are large, bulky and expensive. These were largely accepted in certain industries like petroleum/chemical and textile. During 1985-89 Bi-polar PWM technology, smaller and more economical drives evolved. There was a greater acceptance among users. From 1990-present IGBT technology was developed leading to smaller drive packages with micro drives for smaller hp motors. Switching frequency becomes ultrasonic. Micro drives have actually become commodities. In future, total motor drive compatibility will be achieved. Systems are sold as one. Energy is efficiently supplied across all industries and energy users. Motor development will parallel non-sinusoidal drive development.

For years, industrial motor control applications used general-purpose electronic devices such as microcontrollers (MCUs) and DSPs. These devices are designed with fixed hardware, leaving software as the only method for designers to update designs and limiting the development of application-specific functions. In comparison, FPGAs can integrate processor, Industrial Ethernet/fieldbus standards, custom motor interfaces, and DSP functions in one device. FPGAs give designers the freedom to create custom functions completely adapted to their specific application requirements by enabling both hardware and software customization. FPGAs provide the capability to implement functions in hardware, accelerating performance and simplifying the software porting effort. This additional freedom opens up new avenues of enhanced system performance, especially for motor control energy efficiency.

1.3 Outline

This chapter brought a overall review of the present scenario of the three phase inverter market and its development over time.

Chapter 2 reviews the material that was used to refer during design, implementation and testing of the inverter. Chapter 3 gives a profound introduction to the Space Vector PWM theory. It highlights its principles as well as its dominance over Sine PWM.

Chapter 4 gives a systematic overview of the inverter to get a more understandable and clearer view. It also presents the simulations tools that were used prior to implementing the design and gives a foresight about the results expected.

Chapter 5 and chapter 6 discuss in detail the design aspects of the whole inverter circuit and tries to bring out the various important aspects to be considered while designing an inverter. Chapter 5 deals with the control circuit while chapter 6 deals with the power circuit of the design.

Chapter 7 concludes the design and implementation with testing results and the conclusions drawn from the obtained results. It also discusses the future scope of the project.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

The implementation of an FPGA controlled 3 phase VSI requires gathering knowledge in a lot of domains including power electronics and digital electronics. Various research papers and application notes were referred to get acquainted with the steps involved in the design of VSI.

2.2 Literature Survey

The key to implementation of 3 phase VSI lies in the understanding of various features of the inverter module. Reference[1] gives a complete analysis of SVPWM, its implementation and the advantages it has over conventional Sine PWM. The tools required for the design of control algorithm was also referred from the paper.

The use of FPGA for the control circuit is a key feature of the project. The steps involved in FPGA design flow was referred from [2]. The key feature of the FPGA is its flexibility and parallelism which is exploited in motor drives. Article[3] discusses how FPGAs can be integrated with the power circuit for motor control and the advantages it has over Conventional MCU-based approach.

The DC-AC conversion involves switching of semiconductor devices. Application note[4] discusses the turn-on and turn-off properties of power IGBTs and the effect of gate resistance on the rise time of the IGBT. The gate resistance in the power circuit was designed according to guidance given in [4].

The voltage regulator is a key IC in the circuit. A 15V regulated DC voltage is required to power the opto-coupler IC. The design of input and output circuit parameters was based on the guidance given in application note[5].

The opto-coupler TLP250 which is required for driving the gate is a key component and its significance and application was given in application note[6].

Protection and safety becomes a necessity when it comes to high voltages and currents. The overvoltage protection and snubber circuit design was discussed in reference[7]. Mitigation of circuit faults that causes the arm to short circuit was discussed also discussed in [7].

Reference Material

- [1] *Space Vector PWM as a Modified Form of Sine-Triangle PWM for Simple Analog or Digital Implementation-* P Srikant Varma ang G Narayanan.
- [2] *FPGAs enable energy-efficient motor control-* Jason Chiang
- [3] *Drive Circuits for power MOSFETs and IGBTs-* B.Maurice, L.Wuidart.
- [4] *Linear Voltage Regulators-* Littelfuse, Inc.
- [5] *Smart Gate Driver Coupler TLP5214-* Toshiba Corporation.
- [6] *Protection Circuit Design-* Fuji Electric Co.,Ltd.
- [7] *Gate Drive circuit Design-* Fuji Electric Co.,Ltd.

Chapter 3

SPACE VECTOR PWM

Three phase voltage source inverters(VSI) are widely used in applications such as AC motor drives, uninterruptable power supplies (UPS), line side converters with power factor compensation and active power filters. VSI is a three-phase bridge consisting of six active switches as shown in Fig. 3.1.

3.1 SVPWM Principle

The stator windings of a three-phase ac machine (with cylindrical rotor), when fed with a three-phase balanced current produce a resultant flux space-vector that rotates at synchronous speed in the space. The flux vector due to an individual phase winding is oriented along the axis of that particular winding and its magnitude alternates as the current through it is alternating. The magnitude of the resultant flux due to all three windings is, however, fixed at 1.5 times the peak magnitude due to individual phase windings. The resultant flux is commonly known as the synchronously rotating flux vector.

Similarly, the space vector approach to PWM involves the use of a voltage space vector as reference vector has a constant magnitude (V_{ref}) and revolves with a constant frequency (f_1) in the anti-clockwise direction for phase sequence RYB. The line-side fundamental voltage is proportional to V_{ref} . The fundamental frequency of the line-side voltage is same as f_1 .

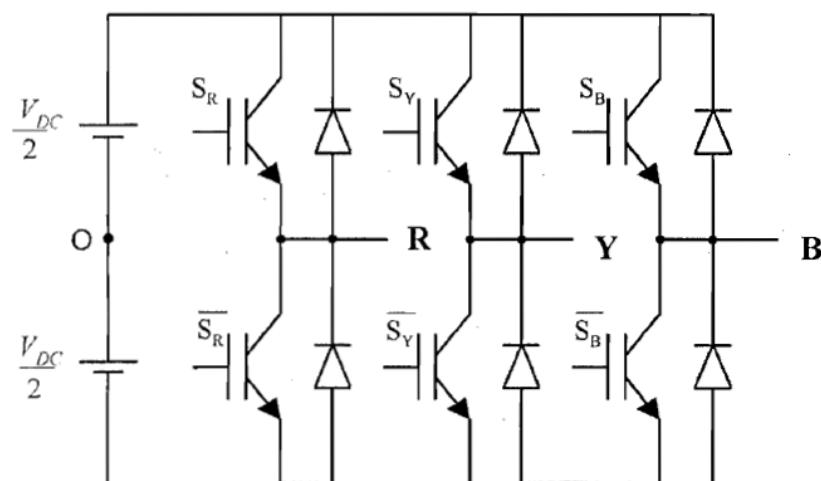


Fig. 3.1 Voltage Source Inverter

3.2 Analysis of SVPWM Technique

Three phase voltages (or currents) can be transformed into voltage space vectors (or current space vectors) using the space vector transformation, defined in equation (1). The axes three-phase axes and the two phase axes are (a-axis and b-axis) are illustrated in Fig. 3.2.

$$\begin{aligned} v_a &= 3v_{RN}/2 \\ v_b &= \sqrt{3}(v_{YN} - v_{BN})/2 \\ V_s &= v_a + jv_b \end{aligned} \quad (1)$$

3.2.1 Switching States and Voltage Vectors

Every leg of the VSI is a Single Pole Double Throw (SPDT) switch with the top and bottom devices switching in a complementary fashion. When the top device is ON, the pole voltage, measured with respect to the DC bus neutral, is $+0.5V_{DC}$. When the bottom device is ON, the pole voltage is $-0.5V_{DC}$.

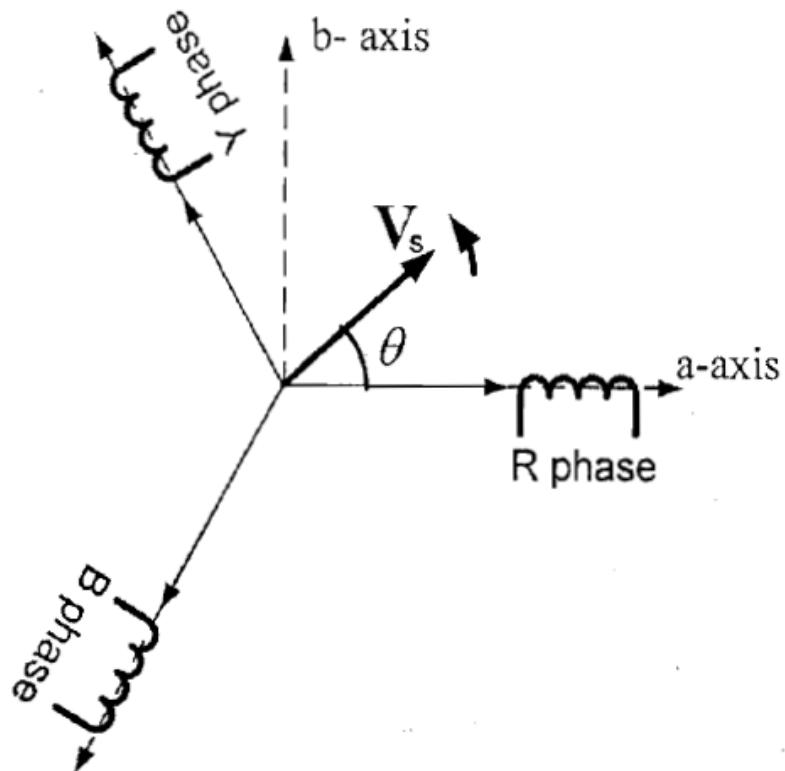


Fig. 3.2 Space Vector Transformation

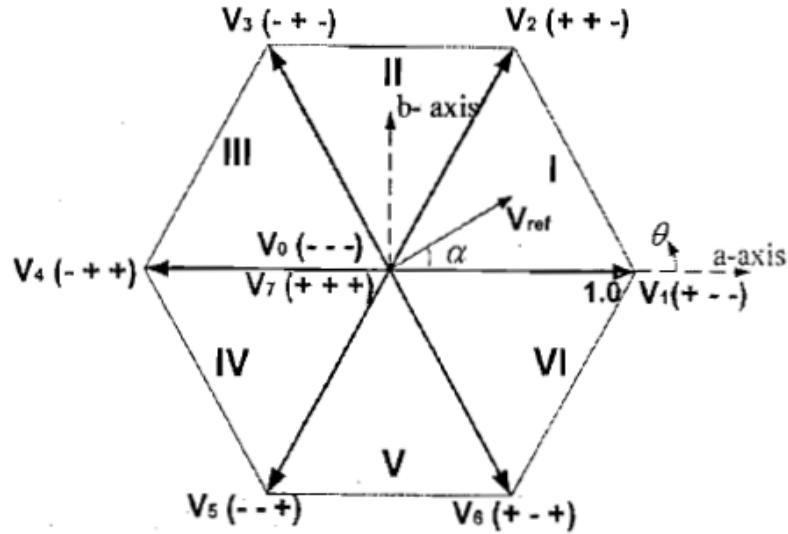


Fig. 3.3 Switching States and Voltage Vectors

With three such legs there are 2^3 or eight possible switching states as shown in Fig. 3.3. For each switching states the three-phase pole voltages (v_{RO} , v_{YO} , v_{BO}) are uniquely defined. Equation(2) gives the corresponding line-line voltages. Assuming a three-phase balanced star-connected load, the corresponding line-neutral voltages applied are as in equation(3). The space vector can now be expressed in terms of three-phase pole voltages as given in equation(4).

$$v_{RY} = v_{RO} - v_{YO}; v_{YB} = v_{YO} - v_{BO}; v_{BR} = v_{BO} - v_{RO} \quad (2)$$

$$\begin{aligned} v_{RN} &= (v_{RY} - v_{BR}) / 3; v_{YN} = (v_{YB} - v_{RY}) / 3; v_{BN} \\ &= (v_{BR} - v_{YB}) / 3 \end{aligned} \quad (3)$$

$$v_a = 3v_{RN} / 2 = (2v_{RO} - v_{YO} - v_{BO}) / 2$$

$$v_b = \sqrt{3}(v_{YN} - v_{BN}) / 2 = \sqrt{3}(v_{YO} - v_{BO}) / 2 \quad (4)$$

The eight inverter states and the corresponding three-phase voltages are tabulated in Table 3.1. The corresponding voltage vectors are also listed both in rectangular as well as polar co-ordinates.

When all the top devices are ON or all the bottom devices are ON, the three-phase load is shorted by the inverter. There is no transfer of power between the DC bus and the three-phase load. These two states are termed as the 'zero states' of the inverter. The two zero states lead to a voltage vector of zero magnitude as shown in Fig. 3.3.

Table 3.1 Switching States and Voltage Vectors

States	(v_{RO}, v_{YO}, v_{BO})	(v_{RY}, v_{YB}, v_{BR})	(v_{RN}, v_{YN}, v_{BN})	(v_a, v_b)	$v_s \angle \theta$
- - - (0)	$(-V_{DC}/2, -V_{DC}/2, -V_{DC}/2)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0)$	0
+ - - (1)	$(V_{DC}/2, -V_{DC}/2, -V_{DC}/2)$	$(V_{DC}, 0, -V_{DC})$	$(2V_{DC}/3, -V_{DC}/3, -V_{DC}/3)$	$(V_{DC}, 0)$	$V_{DC} \angle 0^\circ$
+ + - (2)	$(V_{DC}/2, V_{DC}/2, -V_{DC}/2)$	$(0, V_{DC}, -V_{DC})$	$(V_{DC}/3, V_{DC}/3, -2V_{DC}/3)$	$(V_{DC}/2, \sqrt{3}V_{DC}/2)$	$V_{DC} \angle 60^\circ$
- + - (3)	$(-V_{DC}/2, V_{DC}/2, -V_{DC}/2)$	$(-V_{DC}, V_{DC}, 0)$	$(-V_{DC}/3, 2V_{DC}/3, -V_{DC}/3)$	$(-V_{DC}/2, \sqrt{3}V_{DC}/2)$	$V_{DC} \angle 120^\circ$
- + + (4)	$(-V_{DC}/2, V_{DC}/2, V_{DC}/2)$	$(-V_{DC}, 0, V_{DC})$	$(-2V_{DC}/3, V_{DC}/3, V_{DC}/3)$	$(-V_{DC}, 0)$	$V_{DC} \angle 180^\circ$
- - + (5)	$(-V_{DC}/2, -V_{DC}/2, V_{DC}/2)$	$(0, -V_{DC}, V_{DC})$	$(-V_{DC}/3, -V_{DC}/3, 2V_{DC}/3)$	$(-V_{DC}/2, -\sqrt{3}V_{DC}/2)$	$V_{DC} \angle 240^\circ$
+ - + (6)	$(V_{DC}/2, -V_{DC}/2, V_{DC}/2)$	$(V_{DC}, -V_{DC}, 0)$	$(V_{DC}/3, -2V_{DC}/3, V_{DC}/3)$	$(V_{DC}/2, -\sqrt{3}V_{DC}/2)$	$V_{DC} \angle 300^\circ$
+ + + (7)	$(V_{DC}/2, V_{DC}/2, V_{DC}/2)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0)$	0

The other six states of the inverter are termed as 'active states', which lead to six active vectors of equal magnitude V_{DC} . Fig. 3.3 shows the vectors with their magnitudes normalized with respect to V_{DC} . The active vectors divide the space vector plane into six sectors of angle 60° as shown.

3.2.2 Calculation of Dwell Times and Switching Instants

The reference vector is sampled at equal intervals of time, termed as subcycle (T_s). Let V_{REF} be the sampled value of reference vector in a given subcycle. Let $V_x = 1\angle\theta_x$ and $V_y = 1\angle\theta_y$ be the two active vectors closest to V_{REF} . V_{REF} can be expressed as the sum of a fraction of V_x and a fraction of V_y as shown in Fig 3.4.

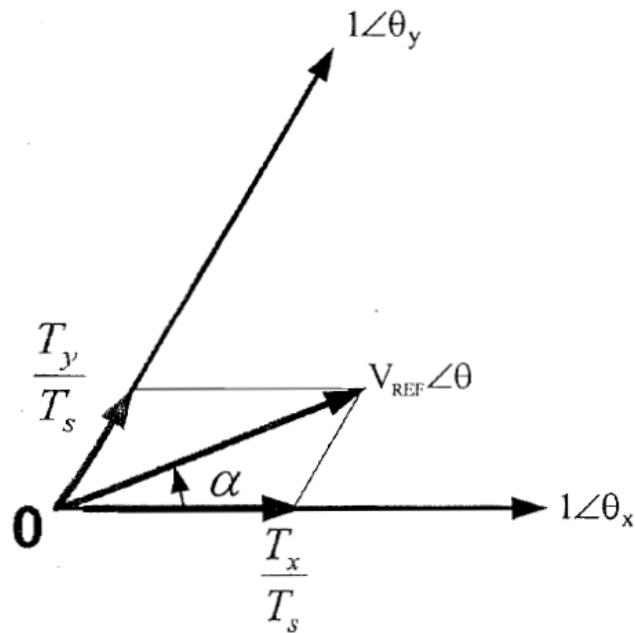


Fig. 3.4 Volt-Second Balance

If V_x is applied for a duration of T_x in the given subcycle, V_y is applied over another interval T_y , and the zero vector for the remaining duration T_z , the average vector applied over the subcycle is given by the RHS of equation(5a). If the durations T_x , T_y and T_z are appropriate, the average vector applied over the subcycle equals V_{REF} as shown in equation (5). In other words, the applied volt-seconds equal the reference volt-seconds. This is referred to as **volt-second balance**.

$$\mathbf{V}_{REF} = \mathbf{V}_x \frac{T_x}{T_s} + \mathbf{V}_y \frac{T_y}{T_s} + 0 \frac{T_z}{T_s} \quad (5a)$$

$$V_{REF} \angle \theta = \frac{T_x}{T_s} \angle \theta_x + \frac{T_y}{T_s} \angle \theta_y \quad (5b)$$

$$\theta_y = \theta_x + 60^\circ \quad (5c)$$

$$\theta = \theta_x + \alpha \quad (5d)$$

To derive expressions for the dwell times T_x , T_y and T_z , the vectors V_x , V_y and V_{REF} are resolved along the direction orthogonal to it as shown in Fig 3.5. The respective components can be equated as shown in equation(6).

$$\begin{aligned} V_{REF} \sin(\alpha) &= (T_y/T_s) \sin(60^\circ) \\ V_{REF} \cos(\alpha) &= (T_y/T_s) \cos(60^\circ) + (T_x/T_s) \end{aligned} \quad (6)$$

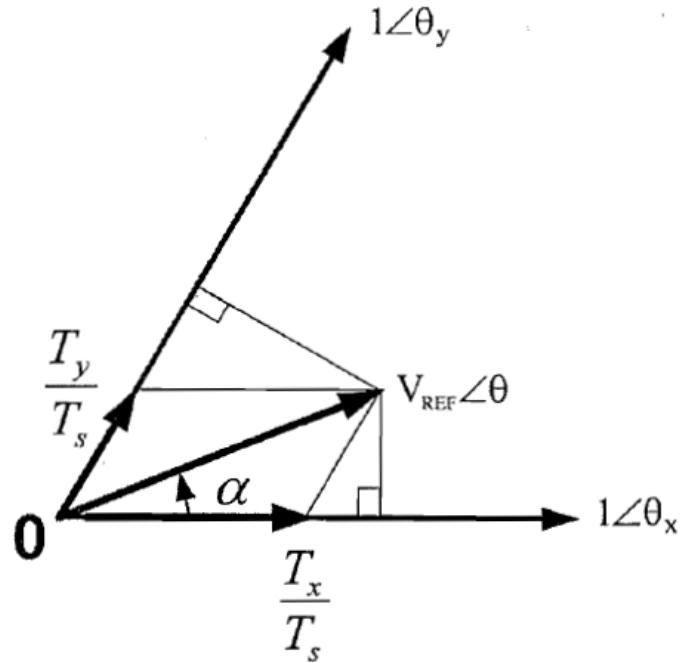


Fig. 3.5 Dwell Times - Method I

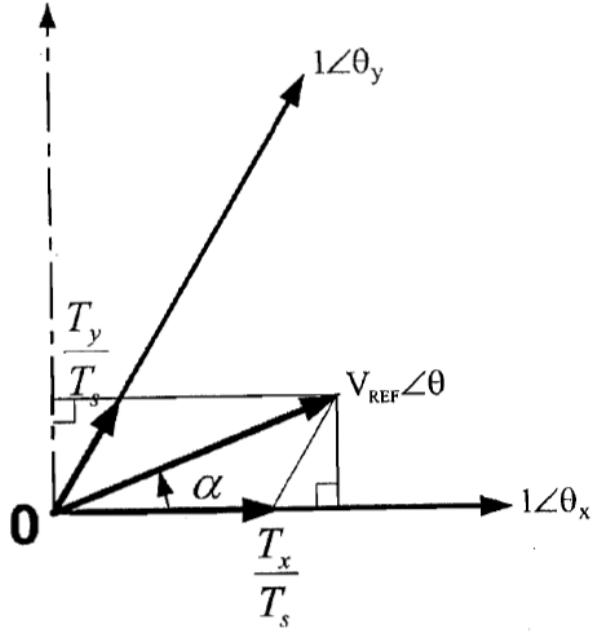


Fig. 3.6 Dwell Times - Method II

Alternatively, from the tip of V_{REF} , drop a perpendicular to V_x and another perpendicular to V_y as shown in Fig. 3.6. This construction yields equation(7).

$$\begin{aligned} V_{REF} \sin(\alpha) &= (T_y/T_s) \sin(60^\circ) \\ V_{REF} \sin(60^\circ - \alpha) &= (T_x/T_s) \sin(60^\circ) \end{aligned} \quad (7)$$

Solving equation(6) yields the expression for T_x and T_y as given in equation(8). These expressions are readily seen from equation(7).

$$\begin{aligned} T_x &= T_s V_{REF} \sin(60^\circ - \alpha) / \sin(60^\circ) = T_s V_{REF} \sin(\theta_y - \theta) / \sin(60^\circ) \\ T_y &= T_s V_{REF} \sin(\alpha) / \sin(60^\circ) = T_s V_{REF} \sin(\theta - \theta_x) / \sin(60^\circ) \\ T_z &= T_s - T_x - T_y \end{aligned} \quad (8)$$

The zero vector can be applied either using the zero state 0 or the zero state 7. The switching instants of the three phases depend on the apportioning of T_z between 0 and 7, and the switching sequence employed. SVPWM applies both the zero states equally for $0.5T_z$ as shown in Fig 3.7. The sequence of inverter states is 0-1-2-7 (forward sequence) and 7-2-1-0 (reverse sequence) in alternate subcycles in sector I. The forward and reverse sequences pertaining to different sectors are as shown in Table 3.2. Corresponding to every state sequence, the sequence in which the three phases switch is also given.

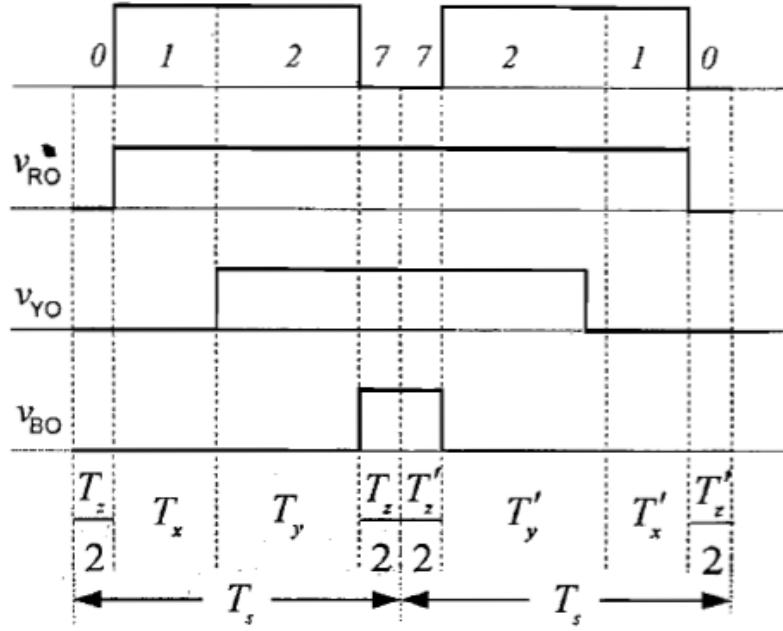


Fig. 3.7 Switching Pattern for Sector I

3.3 Superior Performance of SVPWM

The performance of SVPWM is considered superior to that of SPWM due to higher DC bus utilization and reduced Total Harmonic Distortion(THD).

3.3.1 DC Bus Utilization

For a given DC bus voltage, the highest line-side fundamental voltage is obtained with SVPWM when $V_{REF} = 0.866$ i.e, when V_{REF} equals the radius of the largest circle that can be inscribed inside the hexagon joining the tips of the six active vectors in the space vector plane as shown in Fig. 3.8. As far as the line-side fundamental voltage is considered, SPWM is equivalent to SVPWM with $V_{REF} = 0.75V_m/V_P$. When $V_m = V_P$, the magnitude of equivalent reference vector is 0.75

Table 3.2 Forward and Reverse Sequences in Six Sectors

Sector No.	Forward sequence	Reverse sequence	Switching sequence of phases for forward sequence	Switching sequence of phases for reverse sequence
I	0-1-2-7	7-2-1-0	R-Y-B	B-Y-R
II	7-2-3-0	0-3-2-7	B-R-Y	Y-R-B
III	0-3-4-7	7-4-3-0	Y-B-R	R-B-Y
IV	7-4-5-0	0-5-4-7	R-Y-B	B-Y-R
V	0-5-6-7	7-6-5-0	B-R-Y	Y-R-B
VI	7-6-1-0	0-1-6-7	Y-B-R	R-B-Y

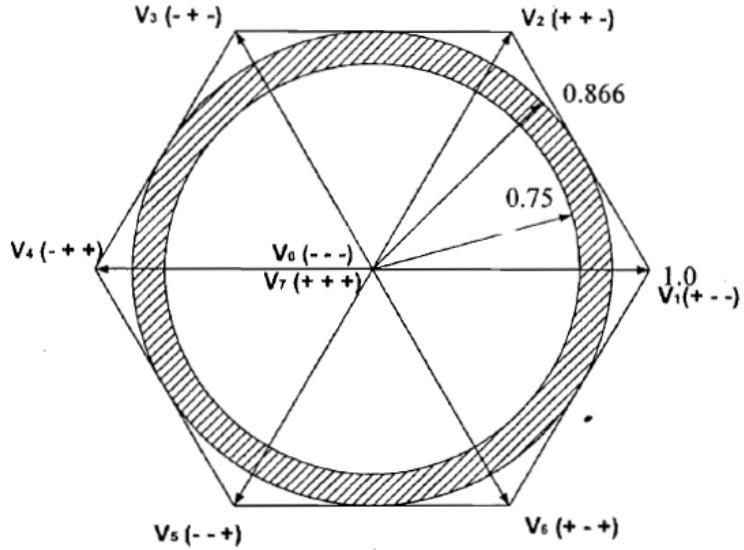


Fig. 3.8 Additional DC Utilization by SVPWM

as shown in Fig. 3.8. Thus the highest line-side voltage obtained with SPWM is only 0.866 time (i.e 0.75/0.866 times) of that obtained with SVPWM.

3.3.2 Total Harmonic Distortion

In every half cycle or subcycle, there is an instantaneous error between the applied voltage vector and the average voltage vector or the reference vector. This voltage is responsible for the harmonics in the line current. The time integral of the error voltage vector is termed as **static flux ripple vector**. This quantity is the measure of the ripple in the line currents. The trajectory of the tip of the stator flux ripple vector corresponding to SVPWM in a given subcycle is shown in solid line in Fig. 3.9 and the trajectory corresponding to SPWM in the same subcycle is shown in dashed lines in the same figure.

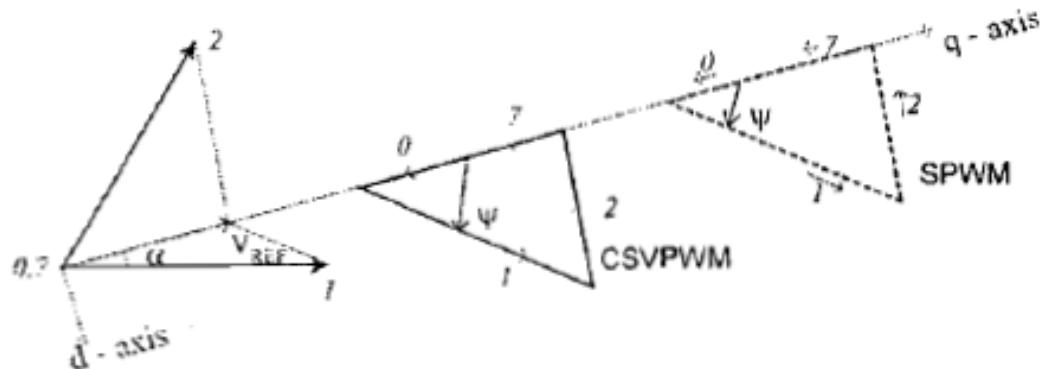


Fig. 3.9 Current Ripple Vector over a subcycle

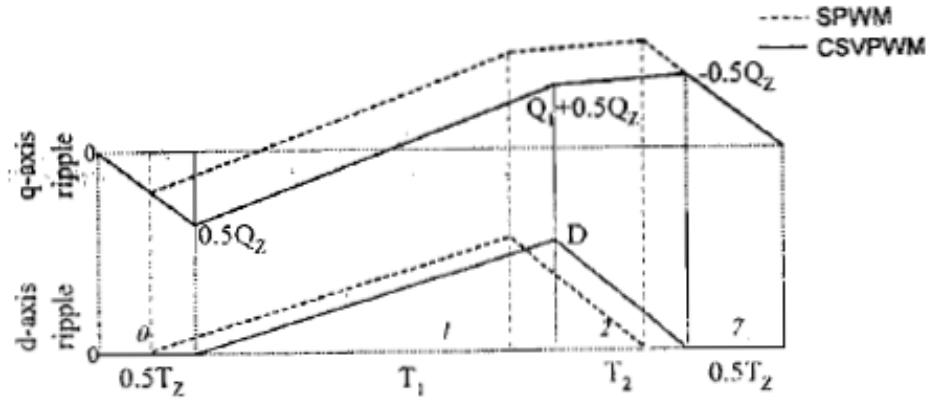


Fig. 3.10 Current Ripple along q-axis and d-axis over a subcycle

The flux ripple vector is considered in a synchronously revolving d-q reference frame. The d-axis and q-axis components of the ripple vector are shown in Fig. 3.10. The quantities Q_z , Q_1 , Q_2 and D in Fig. 3.10 are defined below.

$$\begin{aligned} Q_z &= -V_{REF}T_z \\ Q_1 &= (\cos(\alpha) - V_{REF})T_1 \\ Q_2 &= (\cos(60^\circ - \alpha) - V_{REF})T_2 \\ D &= \sin(\alpha)T_1 = \sin(60^\circ - \alpha)T_2 \end{aligned}$$

The d-axis ripple corresponding to SPWM and SVPWM are equal - both in terms of peak value as well as RMS value. However, the peak q-axis ripple is higher for SPWM due to unequal division of T_z as shown. The RMS q-axis ripple is also

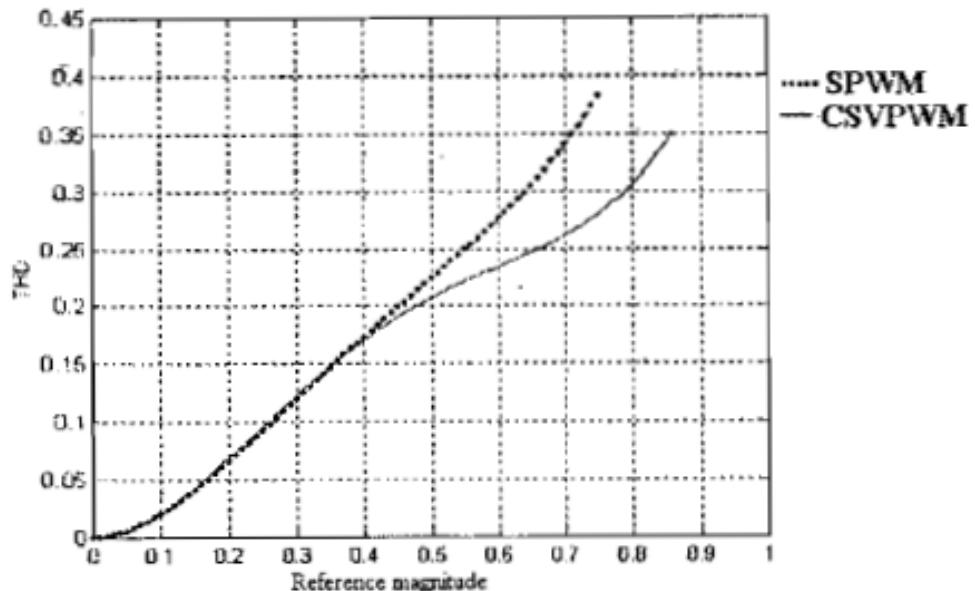


Fig. 3.11 THD vs. V_{REF}

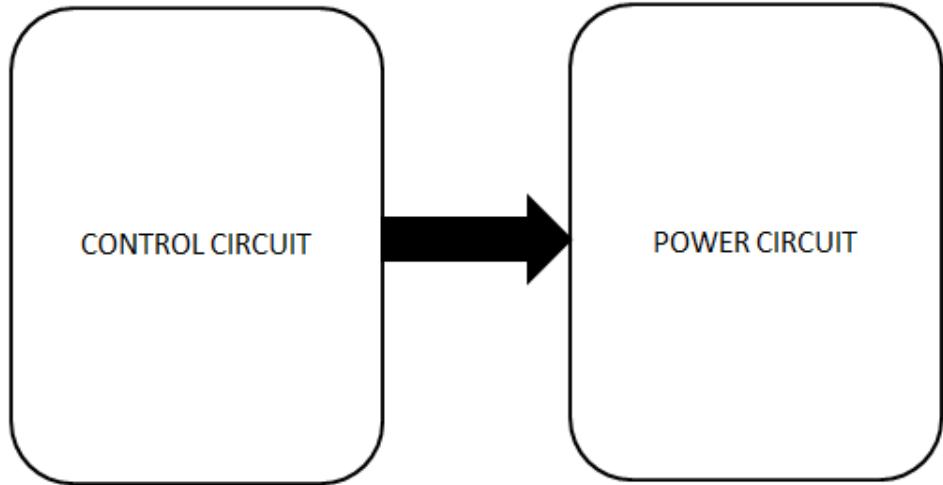
higher. Hence, the RMS current ripple over a subcycle is higher for SPWM. Consequently, the RMS current ripple over a fundamental cycle is higher for SPWM. Therefore, SPWM results in higher total harmonic distortion (THD) than SVPWM at a given fundamental voltage as shown in Fig 3.11.

3.4 Conclusion

Of the existing real-time PWM techniques, sine-triangle PWM (SPWM) and Space Vector PWM (SVPWM) are very popular and important. Compared to SPWM, SVPWM yields 15% higher line-side voltage for a given DC bus voltage. Conversely, for a given maximum line-side voltage, SVPWM requires less DC bus voltage. Consequently, the voltage stress on the semiconductor devices is less. Further, CSVPWM results in reduced harmonic distortion in the line currents over SVPWM, particularly at higher modulation indices.

Chapter 4

SYSTEM OVERVIEW AND SIMULATION



The inverter module consists of two circuits - Control and Power. The power circuit consists of six IGBTs arranged in a H-bridge configuration and are responsible for converting DC voltage to AC. The control circuit is designed on an FPGA which provides control signals to the gates of six IGBTs making them ON and OFF at appropriate times.

4.1 Control Circuit Simulation

The control circuit consists of three major blocks - modulating wave generator, carrier wave generator and delay circuit as shown in Fig. 4.1. The PWM signals are generated by comparing the carrier and modulating waves. The control signals for the lower limb IGBTs are made complementary to the upper limb IGBTs. Considering finite switch ON and OFF time of IGBTs a delay circuit is added to control circuit so that two IGBTs of the same limb do not turn ON at the same time.

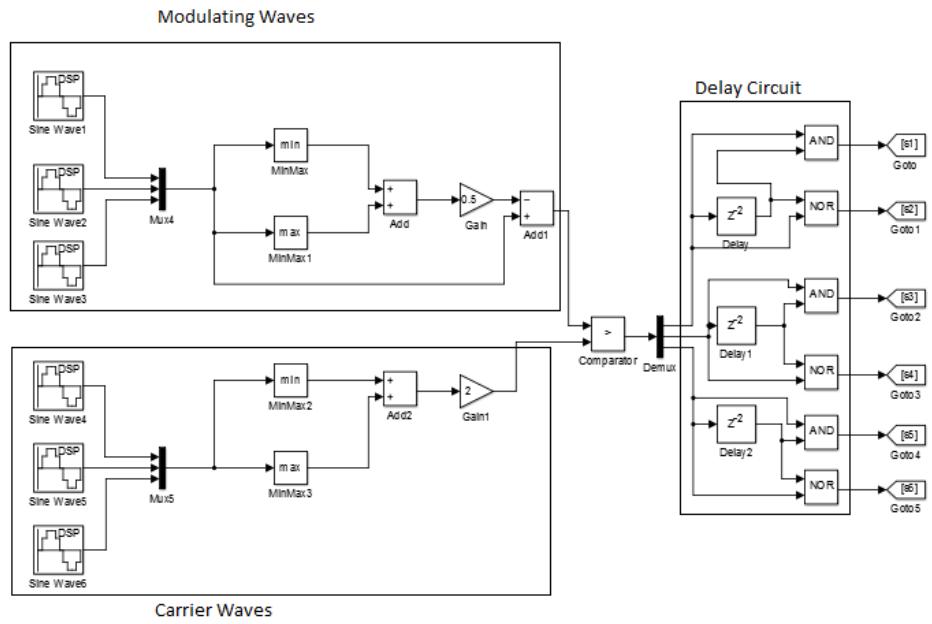


Fig. 4.1 Control Circuit Blocks

The waveforms obtained after comparison in comparator is given in Fig. 4.2 .

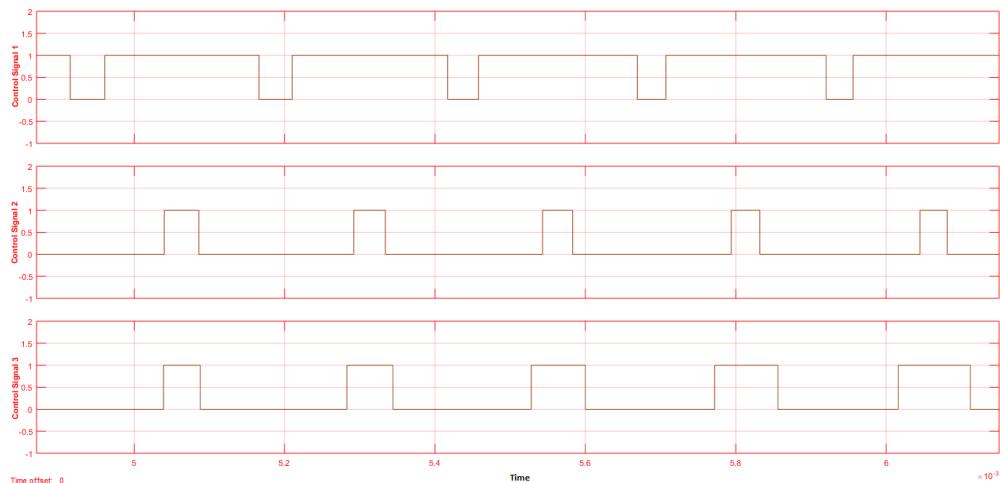


Fig. 4.2 Control signals before delay circuit

The control signals for the six IGBTs are obtained after giving a delay between two IGBTs of the same limb. The control signals for Switches 1 and 1' are shown in Fig. 4.3 showing the time interval in which both the switches are OFF.

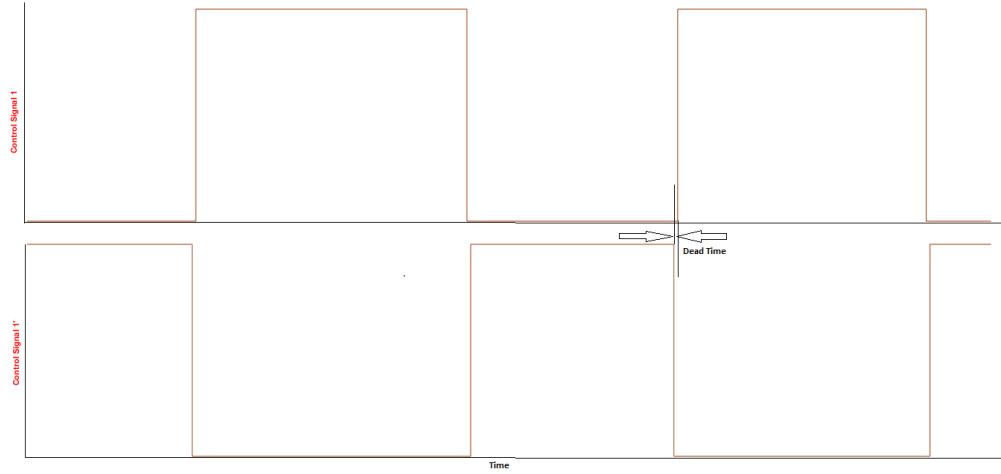


Fig. 4.3 Control signals after delay circuit

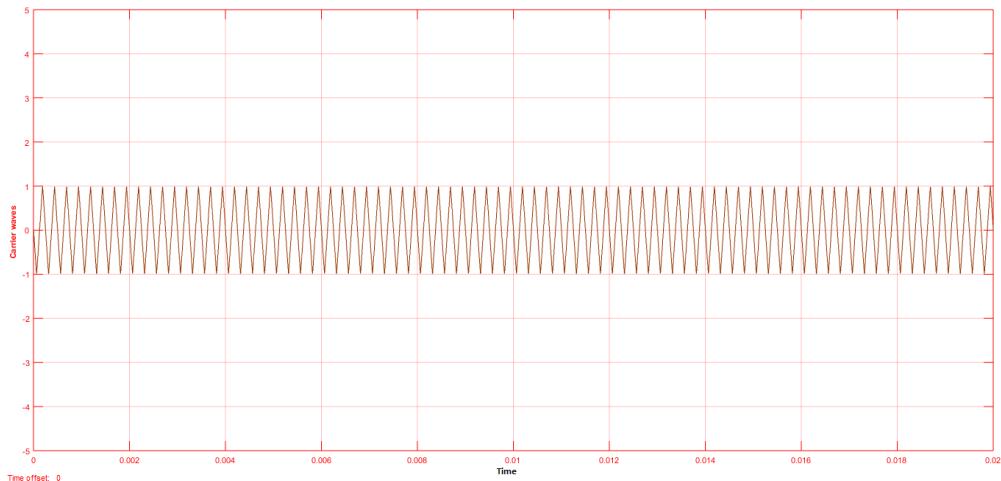


Fig. 4.4 Carrier Wave

The carrier wave which is compared with the modulating wave to generate the PWM signals is shown in Fig. 4.4.

4.2 Power Circuit Simulation

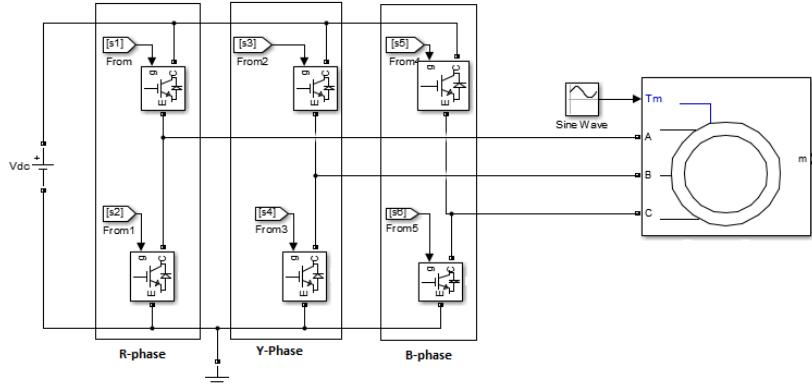


Fig. 4.5 Power Circuit

The power circuit is connected to DC supply of 600V and the control signals from the control circuit are given to the gates of the IGBTs. The AC output from the Power circuit is fed to a 3 phase induction motor. The line voltages are taken across any two phases. The Phase voltages are obtained by the equation:

$$\begin{aligned} v_{RN} &= (v_{RY} - v_{BR}) / 3; v_{YN} = (v_{YB} - v_{RY}) / 3; v_{BN} \\ &= (v_{BR} - v_{YB}) / 3 \end{aligned}$$

The Pole voltages, line voltages and phase voltages are given in Fig. 4.6, 4.7 and 4.8 respectively. Pole voltage has a peak to peak value of 600V. The peak value of Line voltage is equal to V_{dc} of 600V and peak value of phase voltage is $2*V_{dc}/3$ which is 400V.

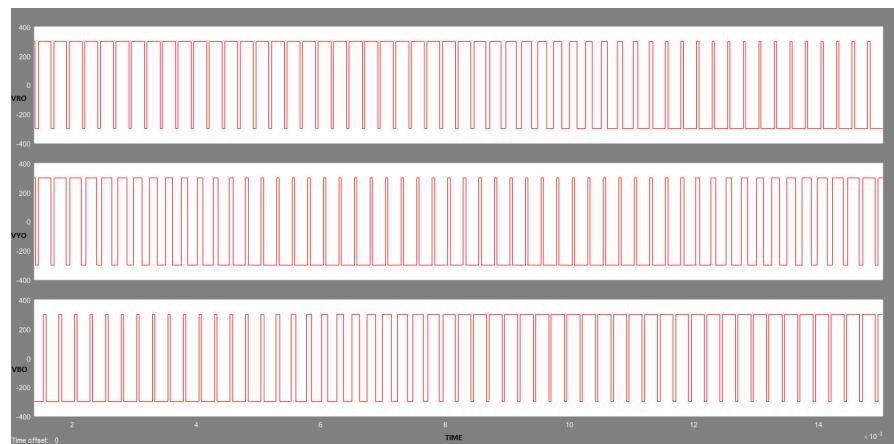


Fig. 4.6 Pole Voltages

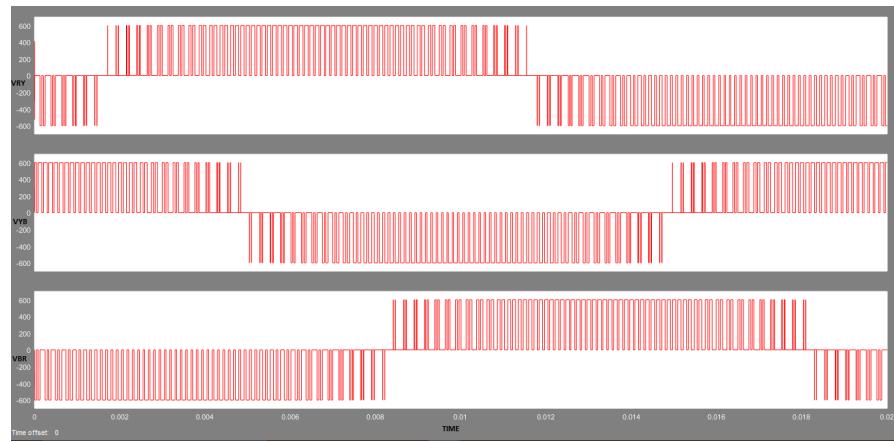


Fig. 4.7 Line Voltages

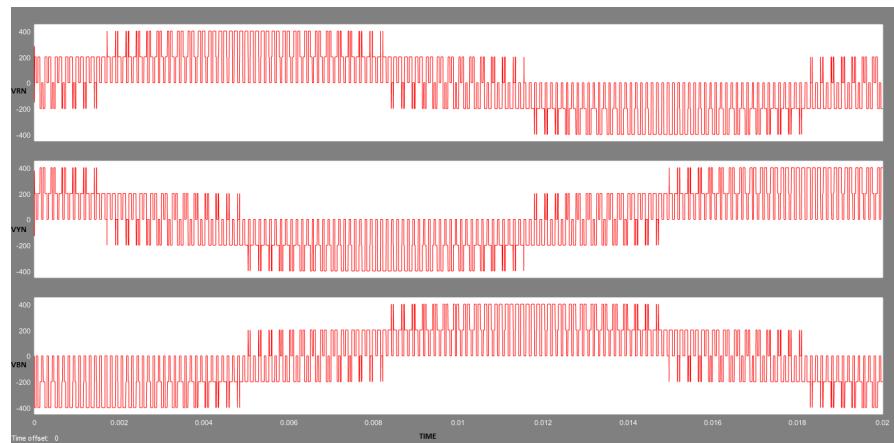


Fig. 4.8 Phase Voltages

The output is fed to a 3 phase, 3 hp, 415V, 4.5A, 1440rpm Induction motor. The simulation is carried out and the speed vs time and the torque vs time waveforms are recorded as shown in Fig. 4.9.

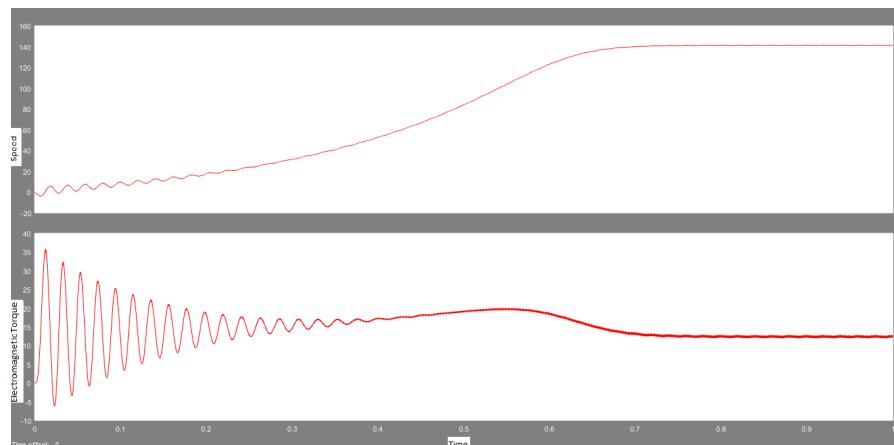
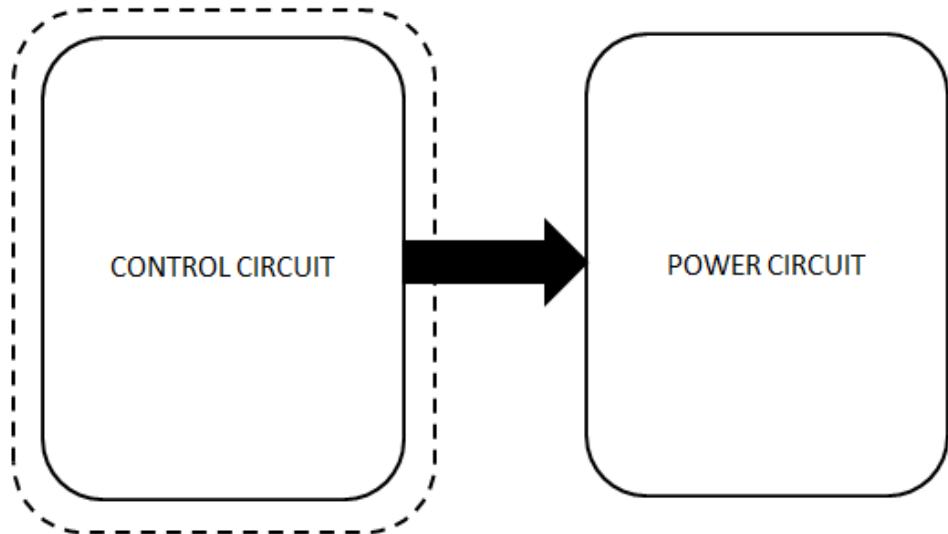


Fig. 4.9 Speed vs Time and Torque vs Time

Chapter 5

CONTROL CIRCUIT



The Space Vector PWM (SVPWM) control technique is implemented using a Field Programmable Gate Array(FPGA). Six control signals and one ground reference from FPGA is supplied to the power circuit. Following sections describe the design and implementation of the control circuit.

5.1 Field Programmable Gate Array (FPGA)

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. This feature distinguishes FPGAs from Application Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks.

A typical FPGA consists of an array of millions of logic blocks, surrounded by programmable input and output blocks and connected together via programmable interconnections. The basic architecture of an FPGA consists of an array of configurable logic blocks (CLBs), a variety of local and global routing resources, and inputoutput (I/O) blocks (IOBs), programmable I/O buffers, and an SRAM-based configuration memory, as shown in Fig 5.1. A typical FPGA logic block

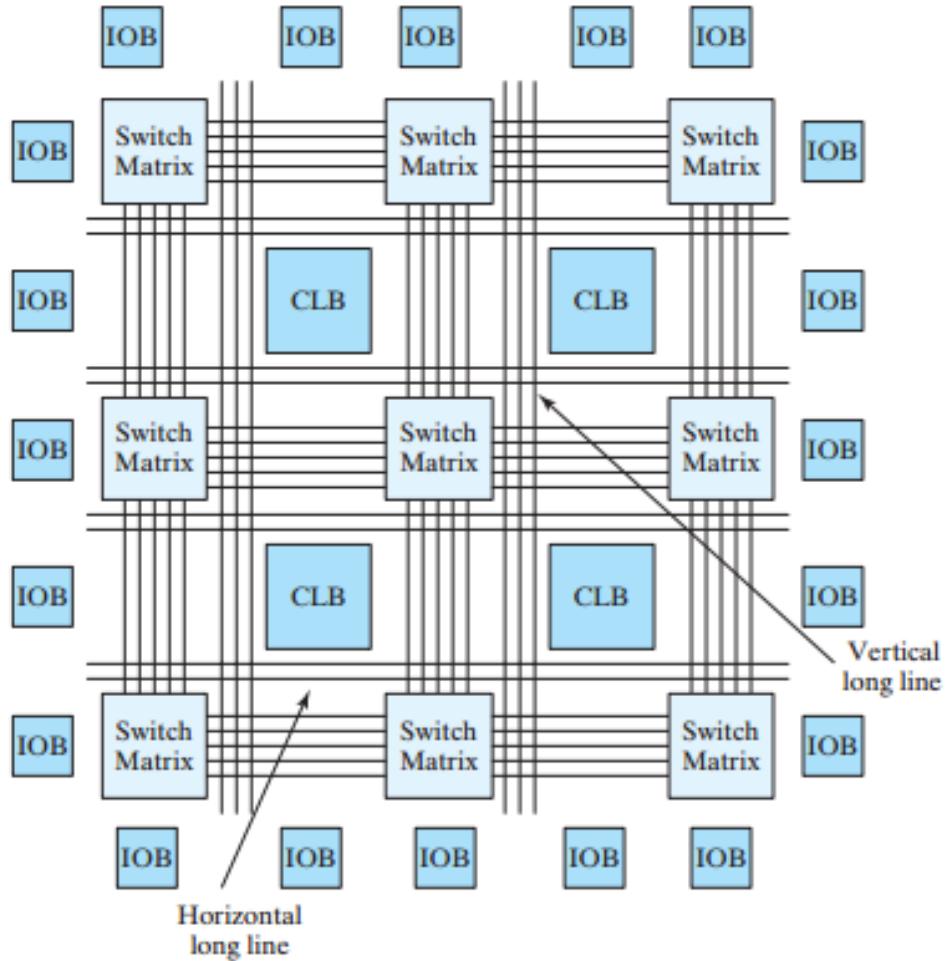


Fig. 5.1 Basic Architecture of FPGA

consists of lookup tables, multiplexers, gates, and flip-flops. A lookup table is a truth table stored in an SRAM and provides the combinational circuit functions for the logic block. The flip-flops act as a past memory element and provides the sequential circuit functions along with the other gates.

Development board used for control circuit implementation is **Terasic DE1-SoC Board**. Fig is the block diagram of the board. The development board consists of **Altera Cyclone V SE 5CSEMA5F31C6N** FPGA and other peripheral components for multi-purpose use.

A Hardware Description Language(HDL) is used for creating a digital circuit on the FPGA fabric. Famous examples of HDLs are VHDL and Verilog. The HDL code is converted into a system of configurable logic blocks(CLBs) and their interconnections which represent the desired digital circuit. This conversion is done using a software design suite. Typical examples are Xilinx Vivado Suite and

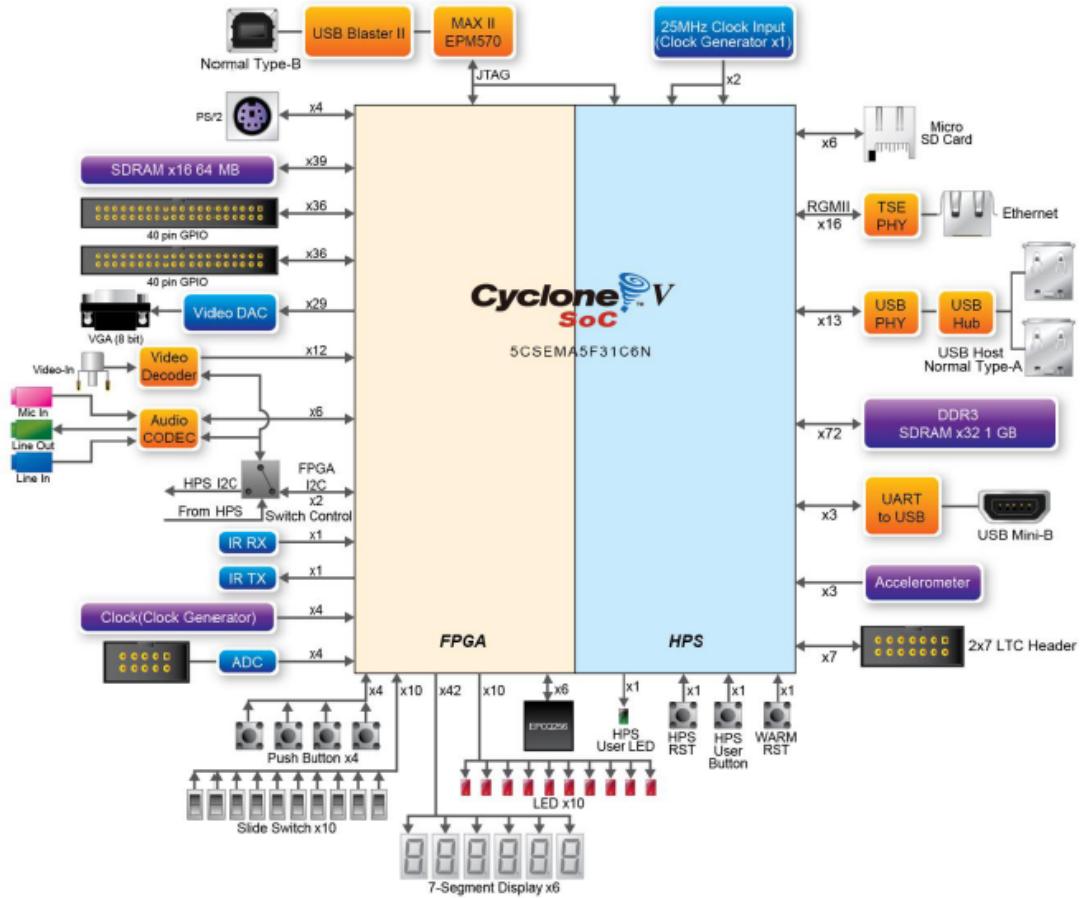


Fig. 5.2 Block Diagram of DE1-SoC Development Board

Altera Quartus Suite. HDL used for designing the control circuit is **Verilog** and software design suite used is **Quartus Prime 16.0**. The Verilog code is provided in Appendix A.

5.2 Control Circuit Design

The SVPWM algorithm explained in Chapter 4 requires digital implementation. For every sampled reference vector, the algorithm requires identifying the sector. The active vectors to be applied and the corresponding switching sequences depend on the sector in which the sample of the reference vector falls. Even within a sector the forward and reverse sequences are to be alternated. Moreover calculation of dwell times of the voltage vectors involves trigonometric functions, and require look-up table. These lead to complexity in the implementation of SVPWM in its traditional form.

This section attempts at modification of SPWM such that the waveforms generated by the modified technique are identical to those generated by SVPWM.

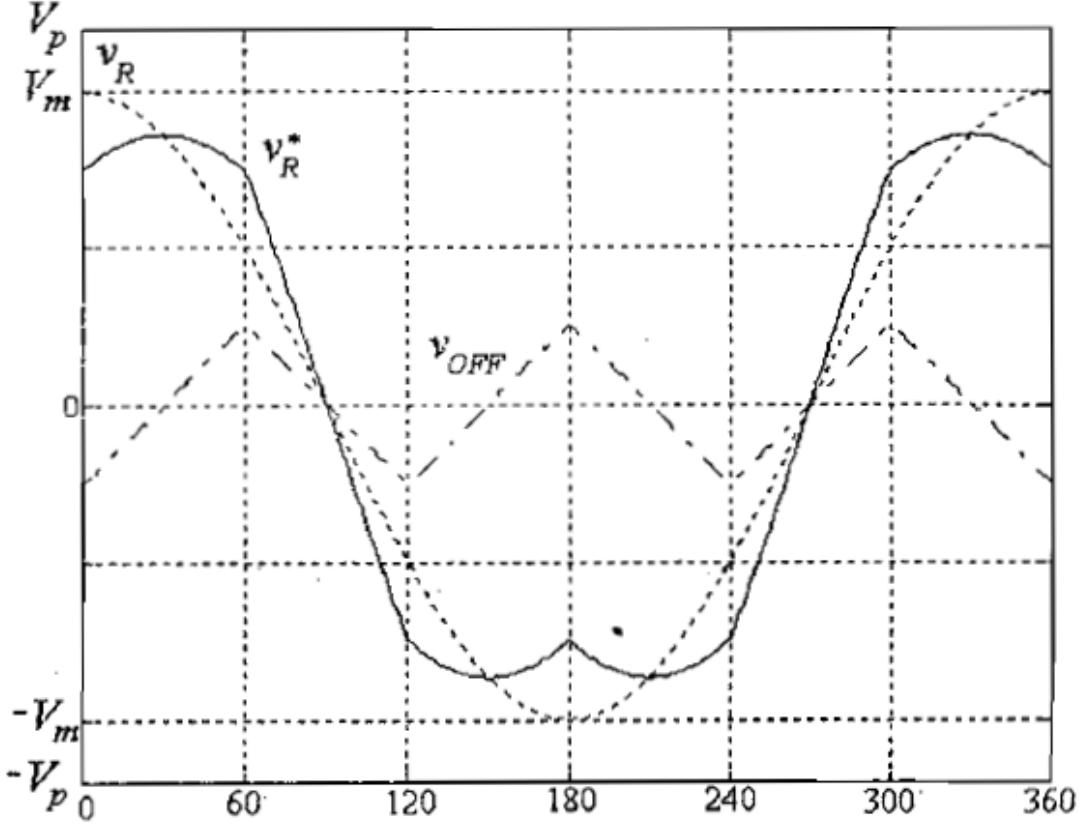


Fig. 5.3 Modified Modulating Waves

The highest of the three-phase references ($v_R[n]$, $v_Y[n]$, $v_B[n]$) in a given half-carrier cycle is designated as v_{MAX} , the lowest one as v_{MIN} and the middle valued one as v_{MID} . An offset voltage or a common-mode component (v_{OFF}) may be added to the three-phase references as shown in equation(5.1) subject to constraints in equation(5.2) where V_P is the carrier wave amplitude.

$$(v_{MAX}^*, v_{MID}^*, v_{MIN}^*) = (v_{MAX}, v_{MID}, v_{MIN}) + (v_{OFF}, v_{OFF}, v_{OFF}) \quad (5.1)$$

$$v_{MAX}^* \leq V_P, v_{MIN}^* \geq -V_P \quad (5.2)$$

If the modified voltage references (v_{MAX}^* , v_{MID}^* , v_{MIN}^*) are used for determining the switching instants instead of (v_{MAX} , v_{MID} , v_{MIN}), the active vector times are unchanged since $(v_{MAX}^* - v_{MID}^*) = (v_{MAX} - v_{MID})$ and $(v_{MID}^* - v_{MIN}^*) = (v_{MID} - v_{MIN})$. However, the zero vector dwell times T_0 and T_7 are changed as shown in equation(5.3), equation(5.4) and the condition is derived as shown in equation(5.7).

$$T_0 = T_s(V_p - v_{MAX}^*)/2V_p \quad (5.3)$$

$$T_7 = T_s(v_{MIN}^* + V_p)/2V_p \quad (5.4)$$

For equal dwell times, $T_0 = T_7$, leading to $(V_p - v_{MAX}^* = v_{MIN}^* + V_p)$ and thus,

$$v_{MAX}^* + v_{MIN}^* = 0 \quad (5.5)$$

$$v_{MAX} + v_{MIN} + 2v_{OFF} = 0 \quad (5.6)$$

$$v_{OFF} = -0.5(v_{MAX} + v_{MIN}) \quad (5.7)$$

The modified voltage references are obtained as shown in equation(5.8). The switching instants in the given half-carrier cycle are determined using the modified references.

$$(v_R^*[n], v_Y^*[n], v_B^*[n]) = (v_R[n], v_Y[n], v_B^*[n]) + (v_{OFF}[n], v_{OFF}[n], v_{OFF}[n]) \quad (5.8)$$

The variation of the offset voltage over the entire fundamental cycle is shown in Fig 5.3. The common-mode component appears almost like a triangular wave of peak $0.25V_m$ and frequency $3f_I$ as shown in Fig 5.3. The common-mode wave contains only triple harmonics. Addition of this offset voltage to the R-phase modulating wave results in the modified modulating wave for R-phase shown in Fig 5.3. Comparison of such three-phase modified modulating waves against a common triangular carrier produces PWM waveforms identical to those produced by SVPWM using the procedure in Chapter 3. In other words, SVPWM can be viewed as such a modified form of SPWM, and can be implemented as such in a simpler fashion.

Similar circuit is designed for generating the carrier waves of required frequency (similar to generating v_{OFF}) which are compared with the modified modulating waves to generate the control signals. Block diagram of the complete control circuit is shown in Fig 5.4.

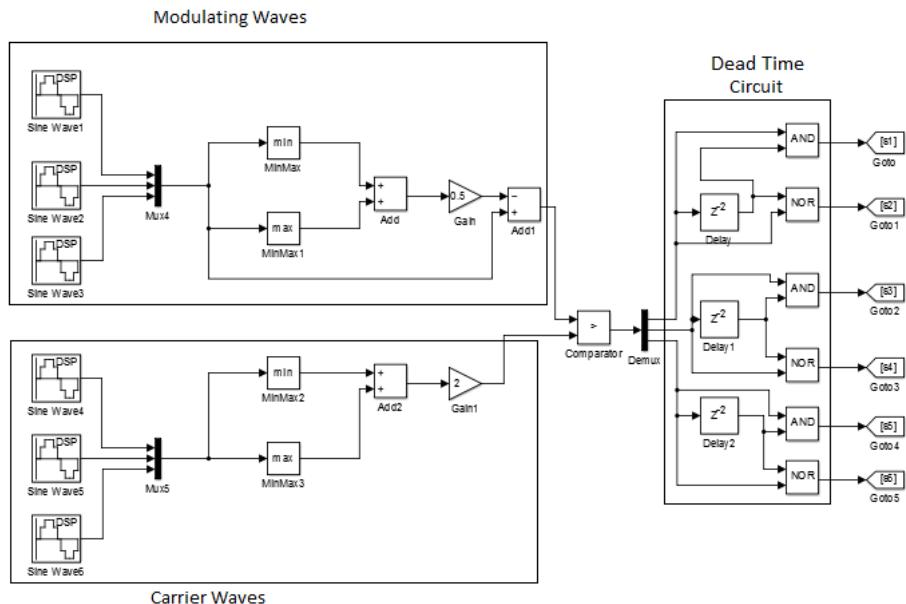


Fig. 5.4 Control Circuit Block Diagram

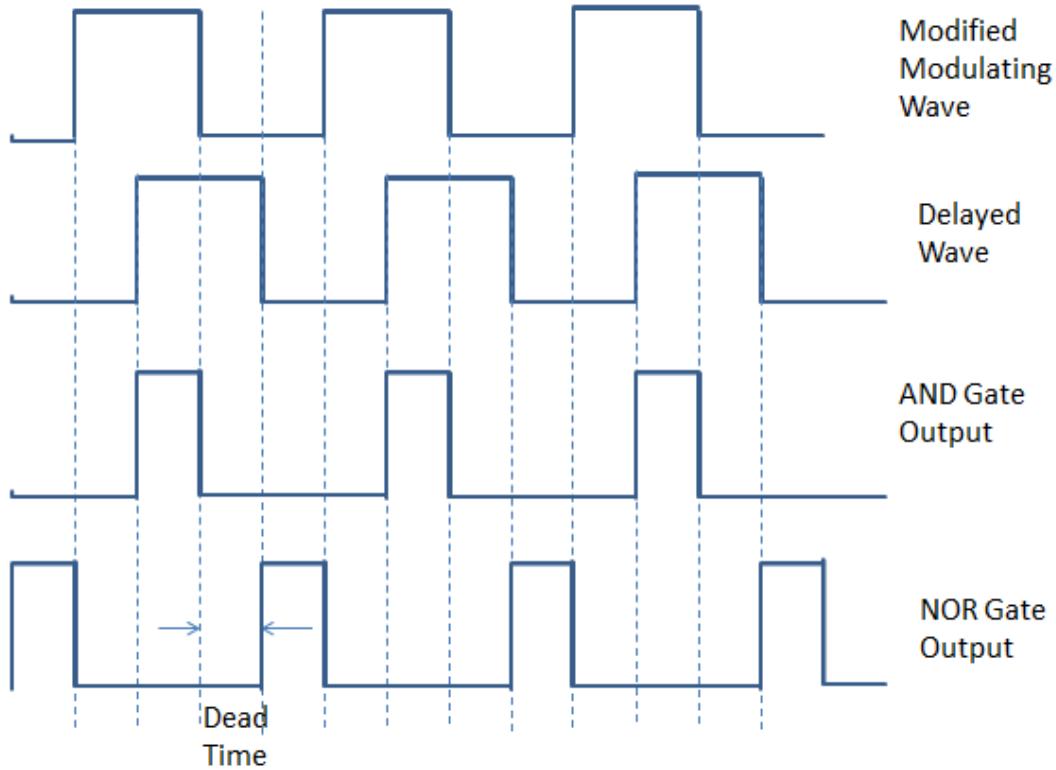


Fig. 5.5 Delayed Circuit Logic Waveforms

The dead time circuit is designed using a simple sequential circuit consisting of delay block, AND gate and NOR gate. The delay block generates a delayed version of the modified modulating waves and the amount of delay is equal to the dead time required. AND gate and NOR gate generate the subsequent control signals for the complementary power semiconductor devices of each leg as shown in Fig 5.5.

5.3 Control Circuit Implementation

The control circuit is implemented using Verilog HDL and the CLBs are configured on FPGA fabric using Altera Quartus Design Suite. Primary clock of FPGA operates at 50MHz but the control circuit is designed to operate at a **sampling rate of 1MHz**. This conversion is achieved by generating a psuedo clock which operates at 1MHz from the original clock at 50MHz. 50 cycles of FPGA clock implies 1 cycle of pseudo clock. Thus switching a bit after every 25 cycles generates the pseudo clock frequency of 1MHz.

Reference sine waves shown in Fig. 5.4 are generated using a look-up table of 20,000 samples per sine wave. Appropriate amplitude of 0.88 is chosen and

at a operating frequency of 50Hz. The three sine waves for generating the modified modulating waves are 120° out of phase with each other. A 16-bit wide register stores the instantenous amplitude of sine wave. These three registers are compared to obtain the minimum and maximum using min and max combinational blocks. These two extreme values are added using a adder circuit and multiplied by a gain of 0.5 using a multiplier as per equation(5.7) to generate v_{OFF} . Generated v_{OFF} is subtracted from the three registers storing instantenous sine wave values using a subtractor circuit. Thus three modified modulating waves are obtained and stored in three new registers which are 16-bit wide.

The carrier wave generator is designed using a similar approach. As seen in Fig. 5.3, v_{OFF} is a triangular wave of thrice the frequency and one-fourth the amplitude that of the reference sine wave. Thus the carrier triangular wave of 4 KHz is generated using three new set of look-up tables for generating the reference sine waves of $4000/3$ Hz. Appropriate amplitude of 1 is chosen and amplified following a similar procedure. v_{OFF} is generated which is a **triangular wave of 4 KHz** and amplitude greater than that of modified modulating waves. The instantenous look-up table values are stored in a 16-bit wide register.

The modified modulating waves and triangular waves are compared using a comparator circuit. The control signal is high when amplitude of modified modulating wave is greater than that of triangular wave and vice versa. Thus we obtain three PWM waveforms.

The three PWM signals obtained are passed further to dead time circuit as shown in Fig. 5.4. A dead time of $2\mu s$ is introduced in between the complementary control signals. This value is chosen as per the on and off time of IGBT and is explained in Chapter 6. Thus waveforms similar to Fig. 5.5 are thus generated for each comparator output and thus we obtain the six desired control signals. These six control signals are taken as output through GPIO pins of FPGA. The pin voltage of Cyclone V FPGA is 3.3V and thus we obtain six SVPWM signals with an amplitude of 3.3V.

5.4 Conclusion

Space Vector Pulse Width Modulation (SVPWM) is a computationally intensive method. As compared to Sine PWM where a sine wave of desired frequency is compared with a high frequency carrier triangular wave. SVPWM involves

Table 5.1 FPGA Resource Utilization

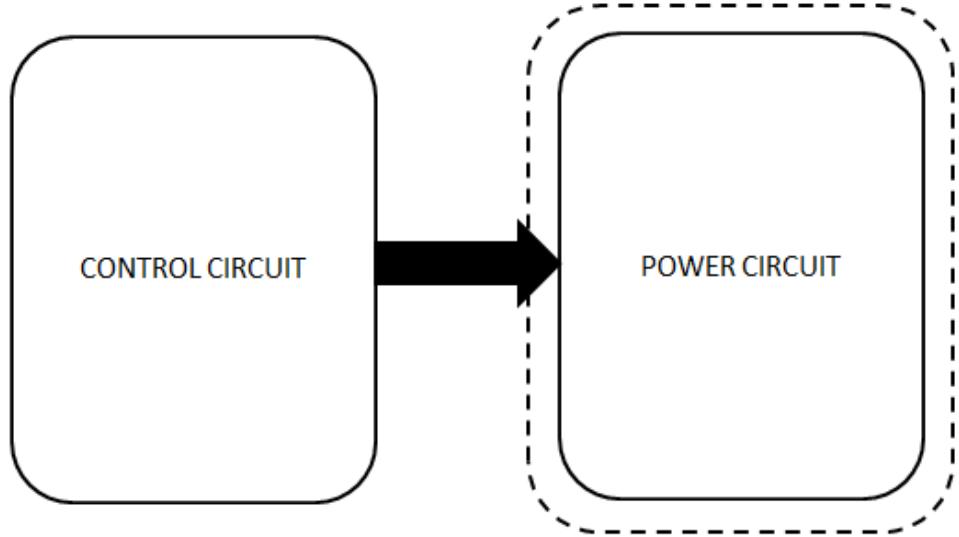
Fitter Status	Successful - Sun Apr 23 00:31:38 2017
Quartus Prime Version	16.0.0 Build 211 04/27/2016 SJ Lite Edition
Revision Name	NewDesign3
Top-level Entity Name	BlockDiagram
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	11,850 / 32,070 (37 %)
Total registers	108
Total pins	10 / 457 (2 %)
Total virtual pins	0
Total block memory bits	300 / 4,065,280 (< 1 %)
Total RAM Blocks	1 / 397 (< 1 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

multiple mathematical steps to be followed. However, benefits of SVPWM outweigh its computational burden. Additionally, the technological advancement has made this technique more efficient. One of these advancements is the adoption of FPGA instead of conventional microprocessor based systems. FPGA are programmable logic useful for parallel processing of signals as opposed to sequential nature of microprocessor instructions. As the hardware is implemented as per the requirements, the delay time involved is least and more robust circuits are obtained. Another advantage of FPGA is the flexibility of design revision over time. Naturally, FPGA are becoming an industry standard in the motor drive domain.

Thus, an FPGA is chosen as the hardware base for the implementation of control circuit design. Additionally, a design improvement which gives the same results as SVPWM but with lesser computational burden is chosen. This design improvement involves modification over Sine PWM and the only difference between them is the modulating signal generated. Table shows the resource utilization of the FPGA CLBs. High logic utilization is due to large number of look-up tables used.

Chapter 6

POWER CIRCUIT



Power circuit carries the actual work load of an inverter and is controlled by a low voltage control circuit which guides its conversion efficiency. However, numerous factors are involved in designing a power circuit which play an important role as well in the conversion efficiency. Thus the following sections focus on some of these factors as well as explains the design of the intended power circuit.

6.1 Voltage Regulator Circuit

The voltage regulator circuit is used to convert a 230V, 50Hz AC supply to a 15V DC supply as shown in Fig. 6.1. The 230V, 50Hz AC supply is stepped down to 14V-0-14V, 2A using a step down transformer. Secondary voltage between 15V and common is applied to bridge rectifier to obtain a fully rectified sine wave. The rectified output is passed through a parallel RC filter, where C is a 2200uF,

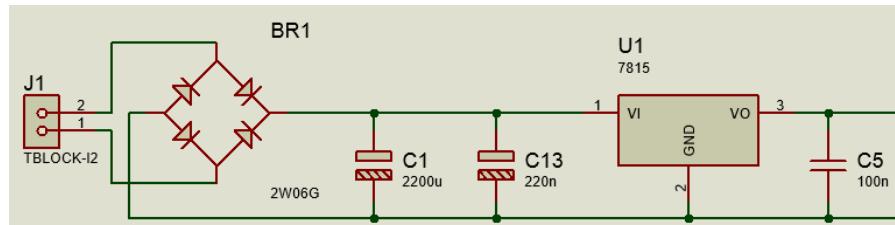


Fig. 6.1 Voltage Regulator Circuit

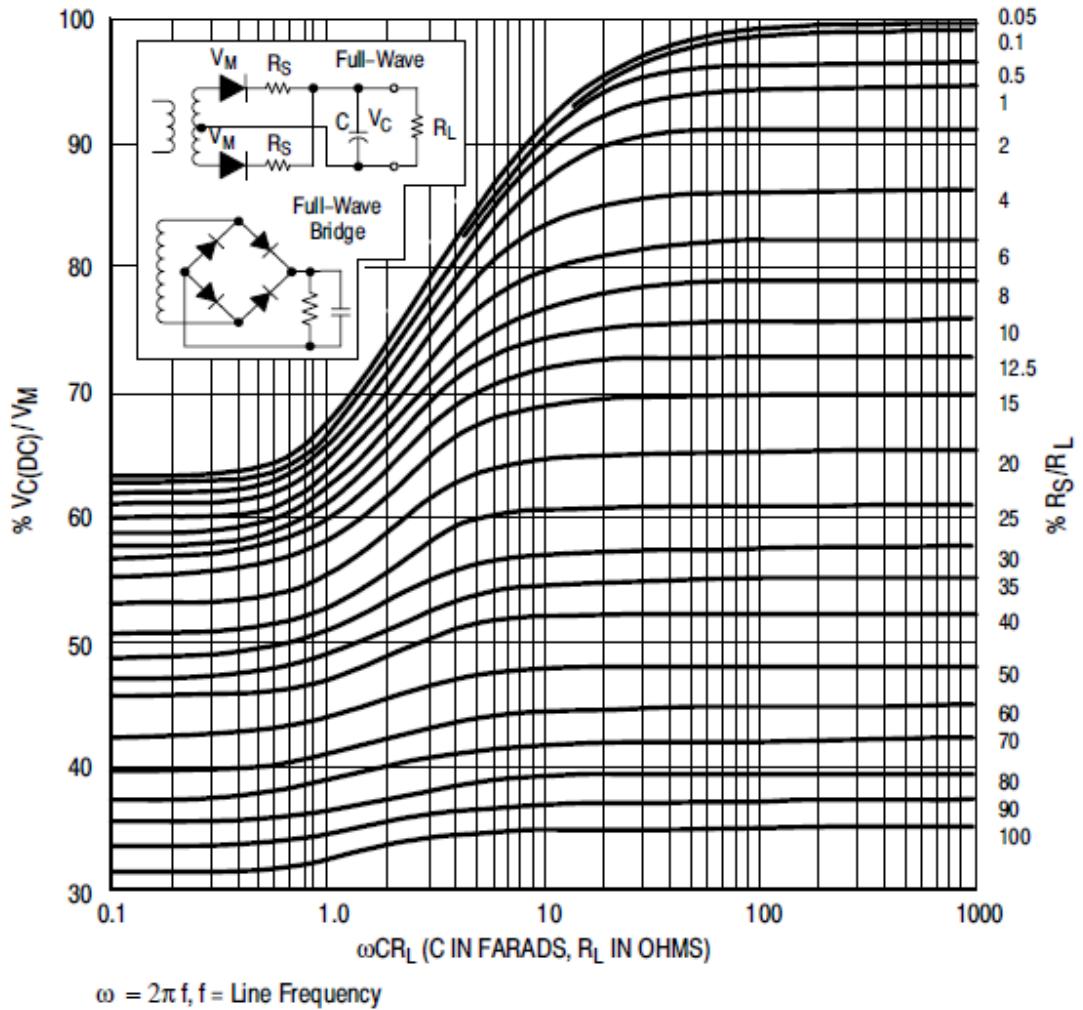


Fig. 6.2 Selection of R_L and C_L

25V capacitor and the remaining components are modelled as R. The value of R is assumed to be $1M\Omega$ and accordingly the capacitance value is calculated using $\omega C R_L = 30$

The voltage across 2200 μ F, 25V capacitor is applied to LM7815 voltage regulator IC, which provides the required 15V DC output. 220nF, 25V input capacitor is used to reduce the ripple voltage amplitude seen at the input of the module and 100nF, 25V capacitor is the output capacitor. The output capacitance of a switching regulator is a vital part of the overall feedback system. The energy storage inductor and the output capacitor form a second-order low-pass filter.

6.2 Optocoupler Isolation

The output voltage of voltage regulator circuit is applied to V_{CC} and GND of TLP250 optocoupler IC. As it can be seen from Fig 6.2, the optoisolator

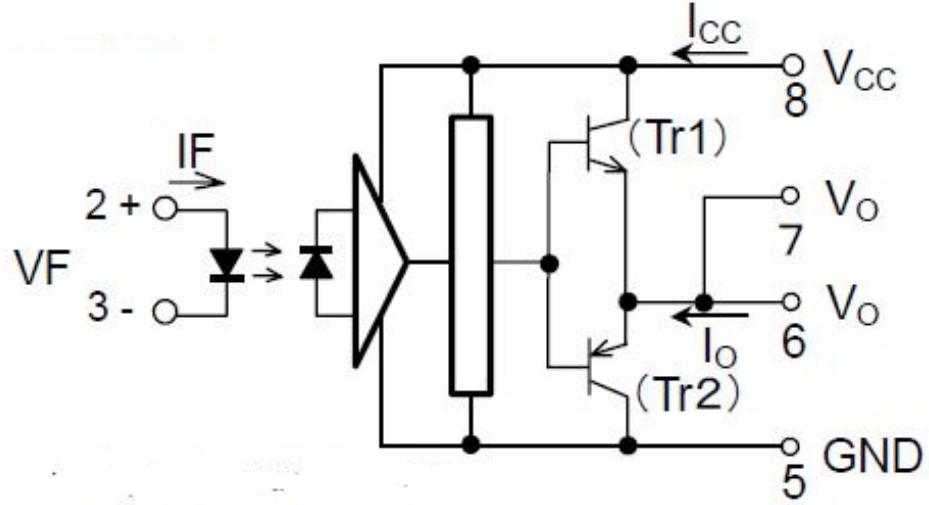


Fig. 6.3 Optocoupler IC TLP250

mechanism provides the required isolation between low voltage switching input and high voltage switching output. The push-pull mechanism converts the low voltage gate input to the switching devices to high voltage switching outputs.

The FPGA used for control circuit implementation provides the PWM signals of 3.3V amplitude. However, the forward voltage drop of the LED in TLP250 is 1.6V while its current rating is around 10mA.

$$R_{in} = \frac{3.3 - 1.6}{0.01} = 170\Omega$$

Thus commercially available 180Ω resistance is chosen. Other two important resistor values to be chosen are the one between TLP250 and IGBT gate (R_G) as well as the high resistance parallel to it. These values are chosen to be 10Ω and $1K\Omega$ and the reasons are discussed in detail in Section 6.5 and 6.6. A ceramic capacitor($0.1\mu F$) should be connected from pin 8 to pin 5 to stabilize the operation of the high gain linear amplifier. Failure to provide the bypassing may impair the switching property.

6.3 High Power Inverter Circuit

The high power inverter circuit is designed using two level voltage source inverter topology considering 600V DC as the supply input. 600V DC supply is provided to the three legs of the inverter through fuses of 5A. Thus the maximum allowable

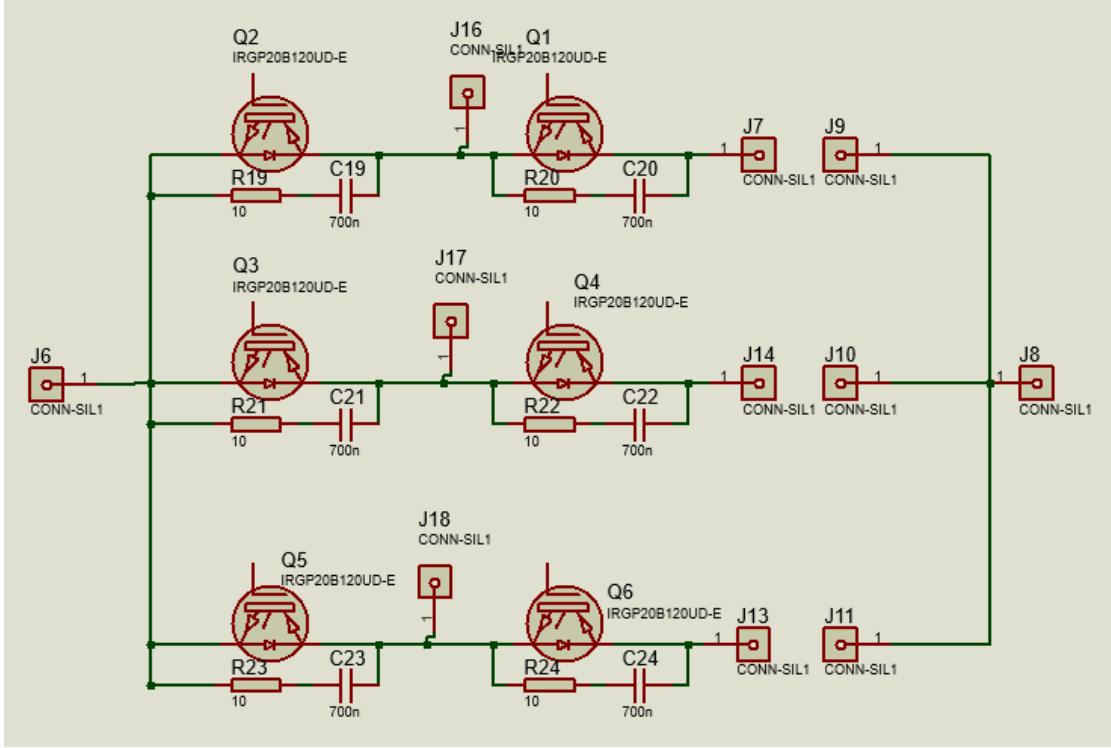


Fig. 6.4 High Power Inverter Circuit

current is limited to 5A considering the width of PCB trace. Higher current values can be used as load with wider trace widths. The three phase AC output is obtained at the pole positions of the legs as shown in Fig. 6.3. Power semiconductor switch used for implementation is GW38IH130D IGBT with 1300V, 33A rating. A basic heat sink is attached to the IGBTs for heat dissipation at higher powers.

6.4 Snubber Circuit

A simple design strategy is followed in designing the snubber circuit. The output capacitance of the chosen IGBT is 155pF. Thus, snubber capacitance C_s is given by:

$$C_s = 2 * \text{output_capacitance} = 310\text{pF}$$

$$R_s = \frac{E_o}{I_o} = \frac{600}{5} = 120\Omega$$

$$W = \frac{1}{2} * C_s * V^2 * f_c$$

$$W = \frac{1}{2} * 310 * 10^{-12} * 600^2 * 4000 = 0.22\text{W}$$

Thus the approximate design values for snubber capacitance is 310pF, 600V and snubber resistance is 120Ω, 0.22W.

6.5 Turn On and Turn Off of IGBT

When turned on under the same conditions, IGBTs and MOSFETs behave in exactly the same way, and have very similar current rise and voltage fall times. However, at turn-off, the waveforms of the switched current are different, as shown in Fig 6.5. At the end of the switching event, the IGBT has a tail current which does not exist for the MOSFET. This tail is caused by minority carriers trapped in the base of the bipolar output section of the IGBT causing the device to remain turned on. Unlike a bipolar transistor, it is not possible to extract these carriers to speed up switching, as there is no external connection to the base section, and so the device remains turned on until the carriers recombine naturally. Hence the gate drive circuit has no effect on the tail current level and profile. The tail current does however increase significantly with temperature.

The turn-off of an IGBT can be separated into two distinct periods, as shown in Figure 6.6. In the first period, its behaviour is similar to that of a MOSFET. The increase in drain voltage (dV/dt) is followed by a very fast fall of the switched current. Losses in this dV/dt period depend mainly on the speed of the voltage increase, which can be controlled by a gate drive resistor. The second tail current period is specific to the IGBT. As this period occurs while there is already a large voltage across the device, it causes losses at each turn-off.

The power involved in these two types of switching losses is linked to the switching frequency. Turn-off losses become critical when operating at high frequencies. In this case, the dV/dt can be increased (and hence losses reduced) by decreasing the

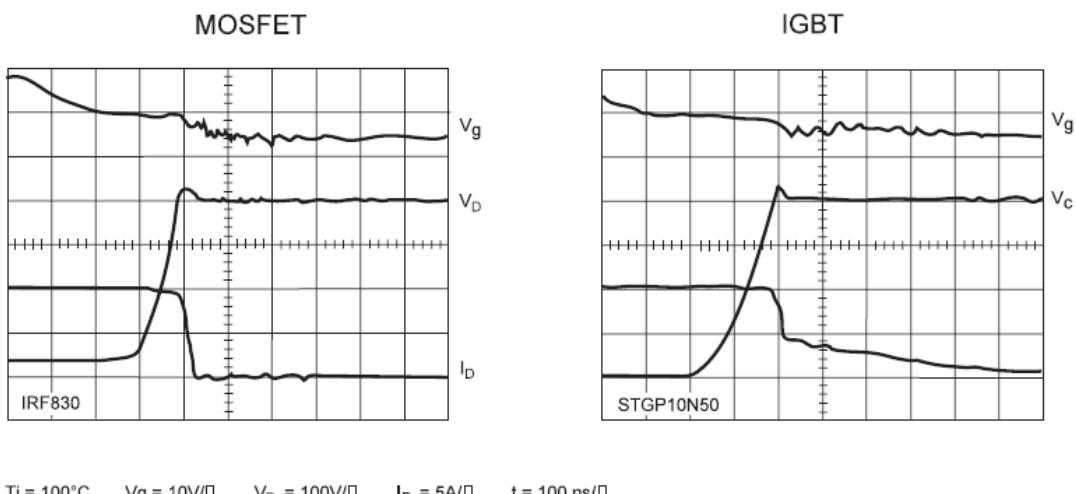


Fig. 6.5 MOSFET and IGBT Turn-off Characteristics

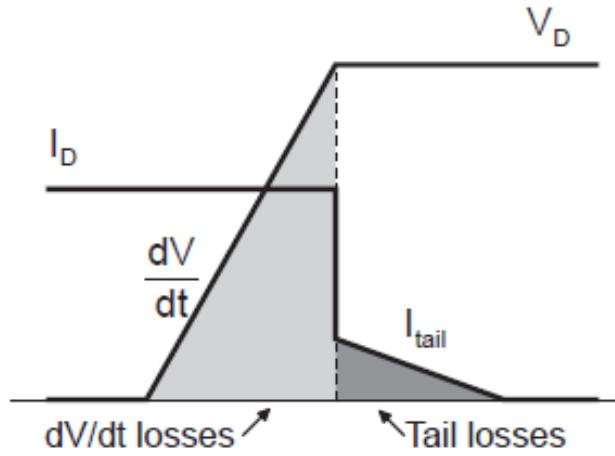


Fig. 6.6 Turn-off Pattern of IGBT

size of the gate drive resistor R_G , which will allow the gate to charge more quickly. The turn-off losses are proportional to the size of the gate resistor. However, it should be remembered that IGBT tail current losses are completely independent of the value of the gate resistor. Even though the tail current is constant, the losses in a system are often predominantly due to dV/dt , because the value of the gate resistance is often too high. In the example of Fig. 6.7, the total losses per cycle are reduced from 13mJ to 4mJ by decreasing the gate resistance from 100Ω to 10Ω .

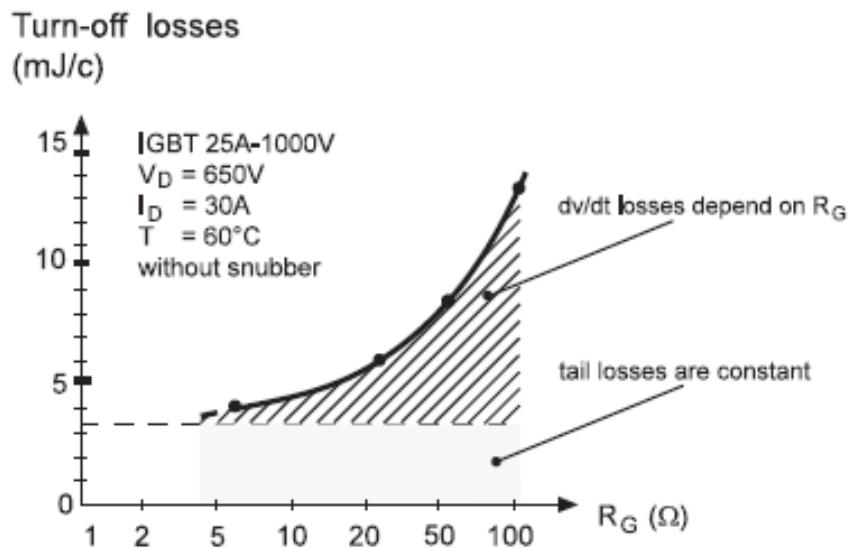


Fig. 6.7 Effect of R_G on Turn-off Losses

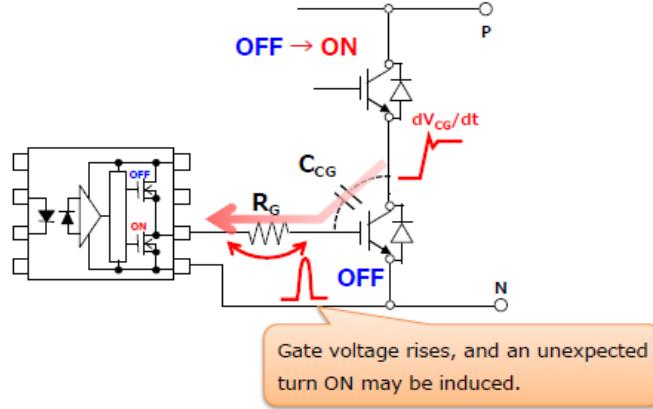


Fig. 6.8 Miller Capacitance Effect

6.6 Miller Capacitance

When the IGBT of the upper arm in the inverter circuit turns on, the V_{CE} of the IGBT of the lower arm rises sharply. At this time, the C_{CG} of the lower IGBT is introduced and the displacement current $I_S = C_{CG} \times (dV_{CG}/dt)$ is generated and flows in the direction of the photocouplers output. As the current passes through the gate resistor R_G of the circuit, the voltage drops and the gate voltage rises, generating a false ON condition of the IGBT, which induces a short circuit in the upper/lower arms. This is depicted in Fig. 6.8. The increasing in gate potential can be minimized by using a smaller gate resistance as shown in Fig. 6.9. Although this solution is less inexpensive than using a negative power supply, switching noise may increase.

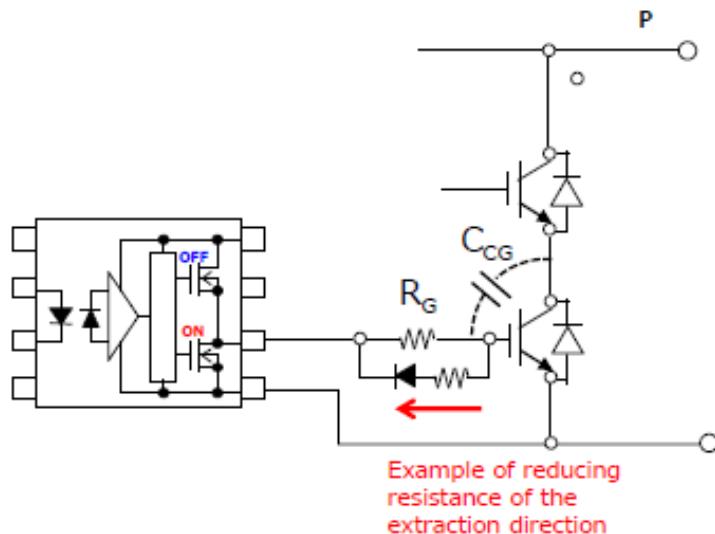


Fig. 6.9 Resistance for Miller Capacitance Mitigation

6.7 Other Design Considerations

One of the common problems in designing an inverter is the shoot through fault. When both the upper and lower IGBT are simultaneously ON due to non-zero turn-on and turn-off time, the DC Bus is short circuited and has a high possibility of damaging the circuit. This problem is eliminated through inclusion of dead time as explained in section 5.3.

Three phase inverter is a circuit with multiple isolations and multiple ground references. The AC input for transformer is isolated from the remaining circuit. The PWM signals are isolated from the power circuit using TLP250 IC. Three different transformers are used for upper three IGBTs so that they are not shorted through the common reference of a single transformer. On the other hand single transformer can be used for lower three IGBTs since their emitter pins are already shorted. Thus, only four transformers are required for supplying the gate voltage for the six IGBTs.

Other types of faults include overcurrent faults which are mitigated using current sensing feedback elements and overvoltage faults which are mitigated using snubber circuits.

6.8 Conclusion

Fig 6.10 shows the complete circuit diagram of the inverter. It excludes the FPGA and the transformer inputs to voltage regulator circuit.

Fig. 6.11 is the copper layout of the single layer PCB designed for power circuit. The width of the trace is thicker at the high voltage end and thinner at the low voltage end. As discussed in the previous sections, multiple fault possibilities prove to be a challenge in designing an efficient inverter, reiterating the fact that only control circuit doesn't determine the efficiency of the whole inverter.

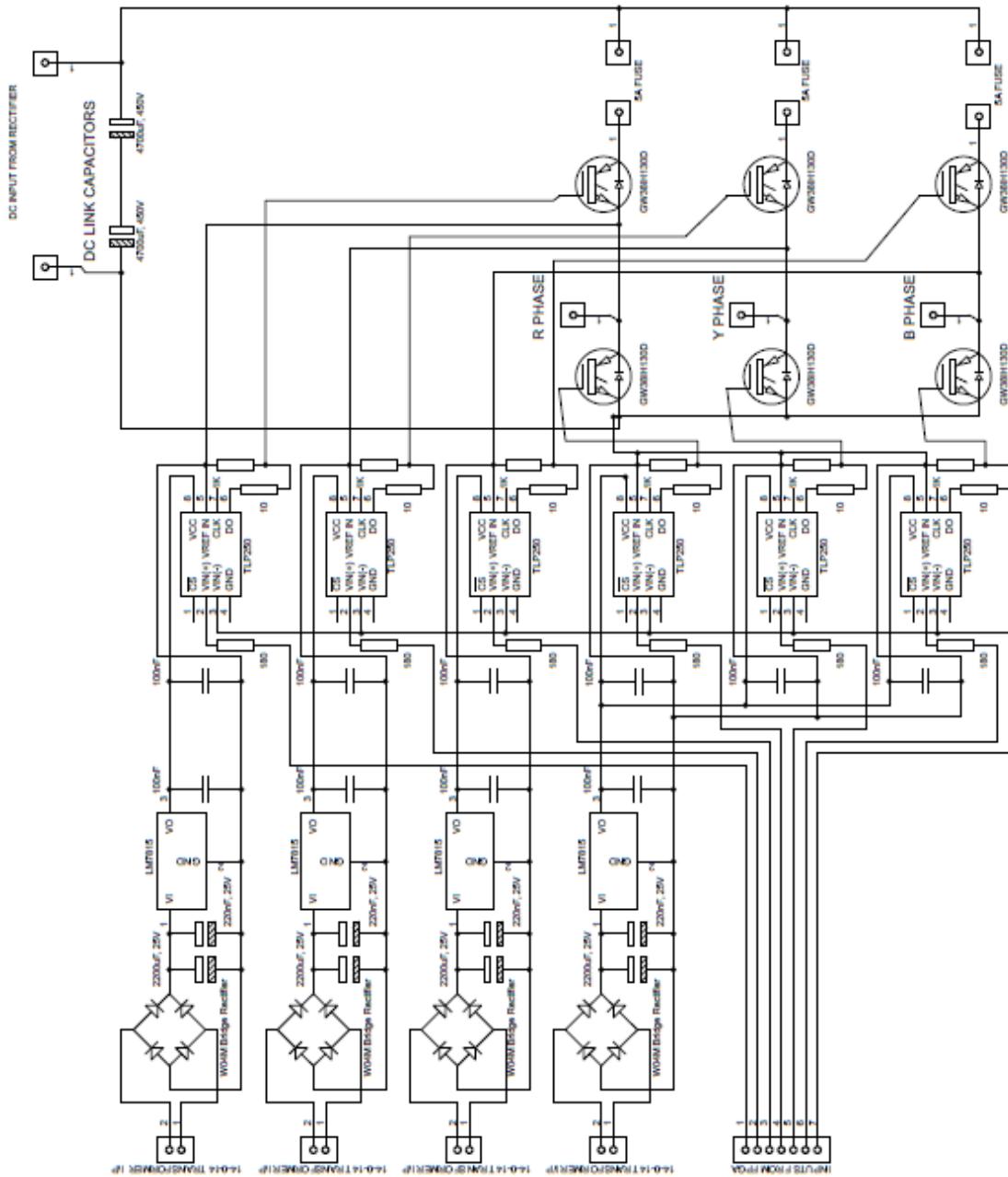


Fig. 6.10 Inverter Circuit Diagram

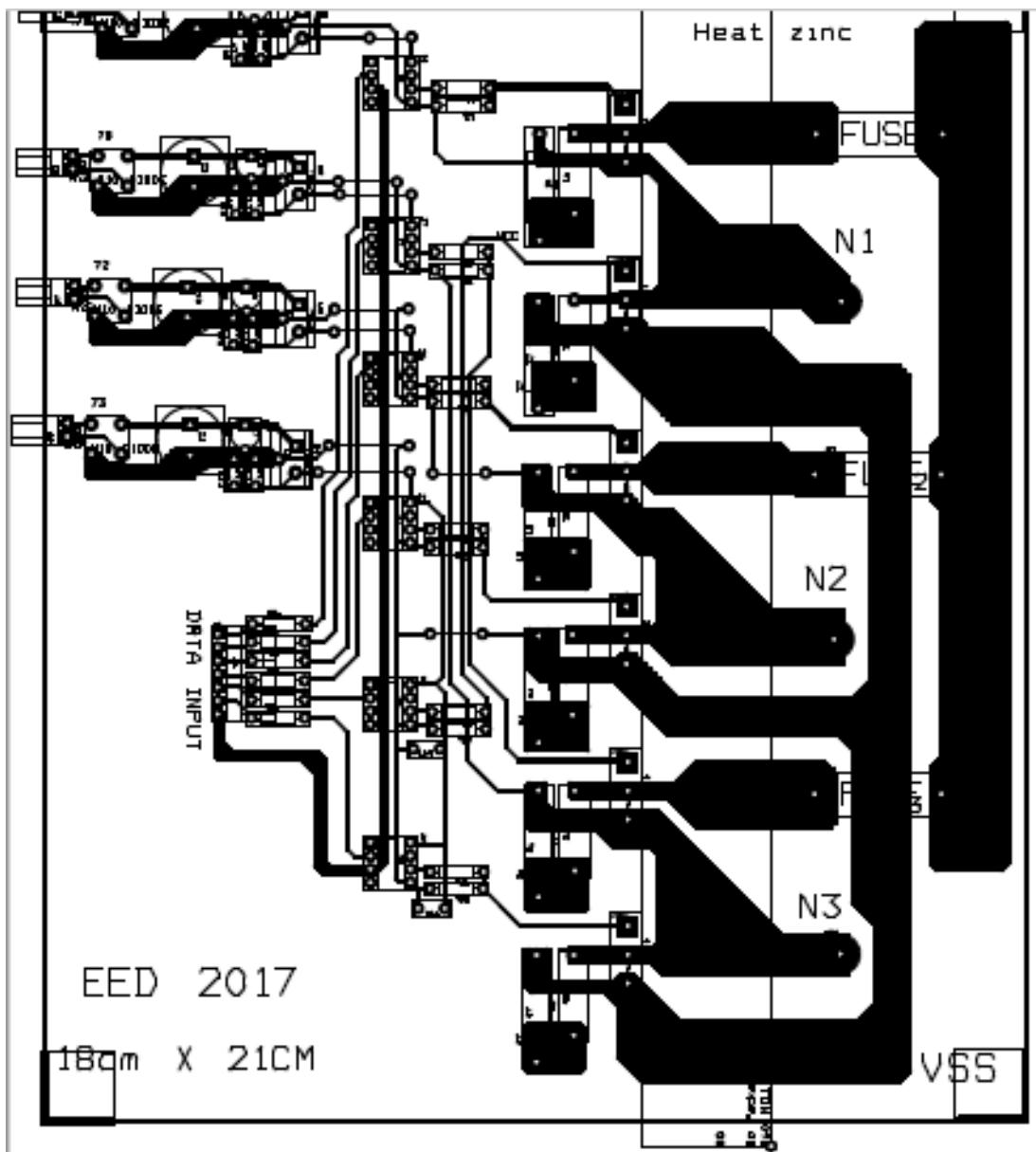


Fig. 6.11 Power Circuit PCB Layout

Chapter 7

OBSERVATION AND RESULTS

A three phase voltage source inverter is designed for 600V DC input. Following are the rated parameters of the designed inverter:

Rated Output Voltage	415V
Rated Output Frequency	50Hz
Maximum Output Current	5A
Input Voltage	600V
Sampling Frequency	1MHz
Carrier Wave Frequency	4KHz

7.1 Initial testing

The initial lab setup is shown in Fig. 7.1.

Incremental testing procedure was followed for testing the inverter. The inverter is first tested for 30V at no load conditions. The line voltage waveforms obtained

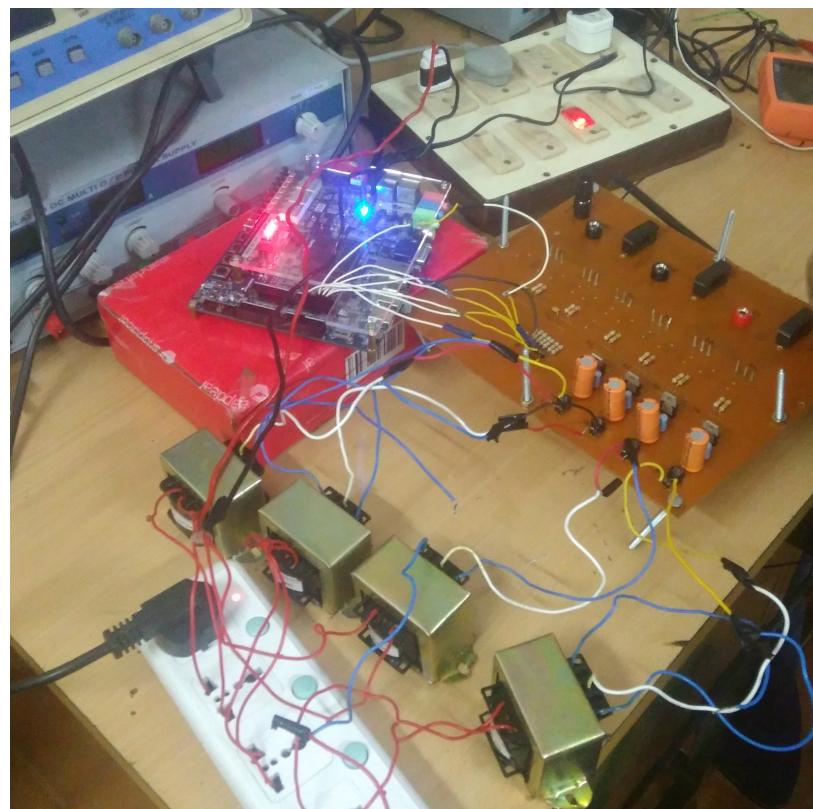


Fig. 7.1 Initial Lab Setup

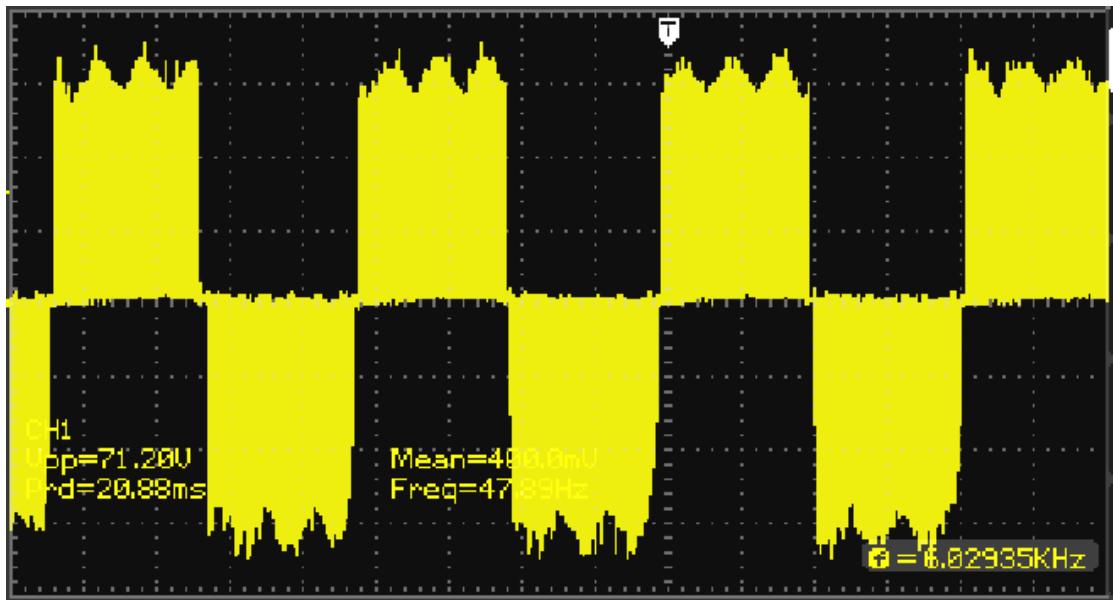


Fig. 7.2 Line Voltage at 30V DC Supply, No Load

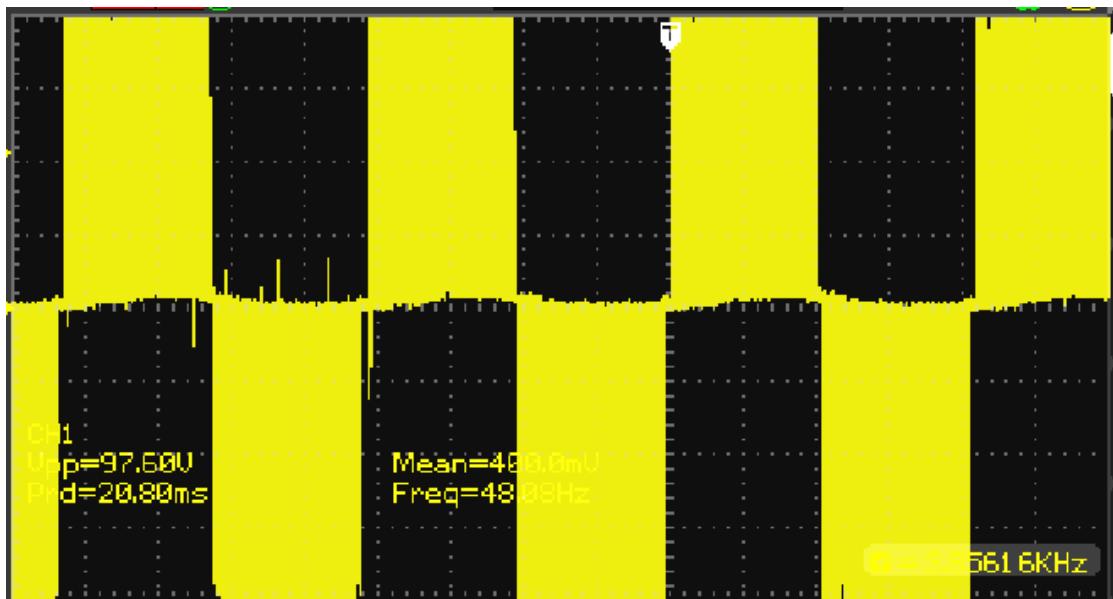


Fig. 7.3 Line Voltage at 50V DC Supply, No Load

at the CRO display is shown in Fig. 7.2.

The circuit is next tested for 30V DC with resistive load drawing 0.1A current. The output waveforms obtained are similar as shown in Fig. 7.2.

The DC bus voltage is increased from 30V to 50V and the line voltage waveform obtained is as shown in Fig. 7.3. The peak of the waveform is trimmed due to reaching the minimum volts/division limit of the CRO. The initial problems faced were due to DC input with high ripple content. This distorted the AC waveform

when the DC voltage was increased beyond 30V as shown in Fig 7.3. As a result there was a need for a DC link capacitor of high rating to reduce ripple content.

7.2 Testing with 3 phase Induction motor

After initial testing, modifications are made to the initial setup so that the inverter can be used to run at higher voltages. A 3 phase AC supply is connected to an auto-transformer and a full bridge rectifier module is connected to the auto-transformer to obtain a DC output. Two 4700uF, 450V capacitors are connected in series across the output of the rectifier to reduce ripples. The schematic of the setup is shown in Fig. 7.4.

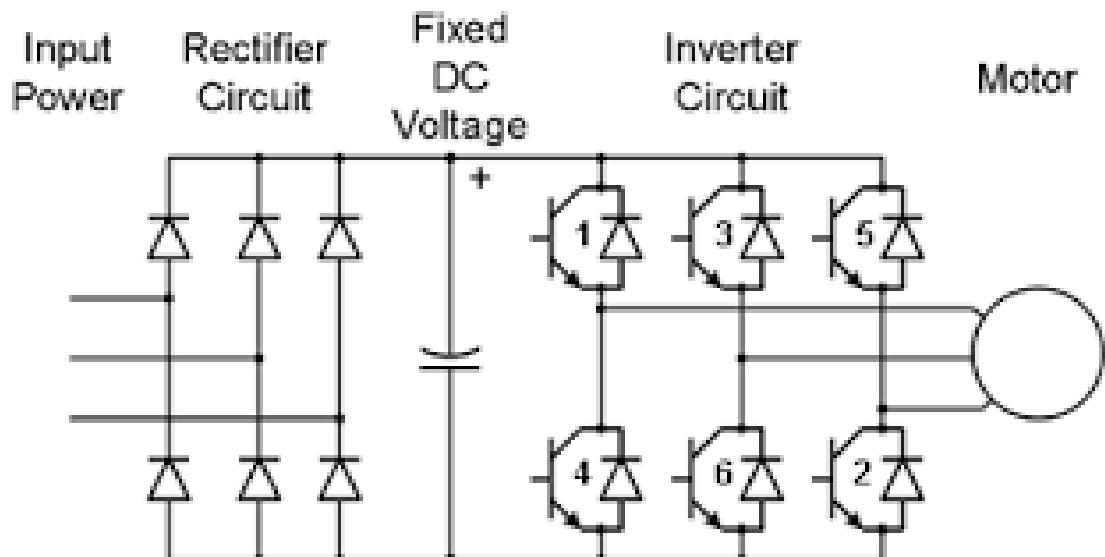


Fig. 7.4 Schematic of inverter setup for higher voltages

A 3.7KW 3 phase star connected squirrel cage induction motor is connected to the output of the inverter as shown in Fig 7.5. The DC input to the inverter is slowly increased from 0V to a maximum value of 400V using the 3 phase auto-transformer. The voltage and current waveforms are recorded on a CRO as shown in Fig. 7.6.

A line to line output voltage of 295V rms and a no load line current of 819mA is observed when DC input is 400V.

The motor is then connected in delta configuration and tested with load. The waveforms are recorded on the CRO as shown in Fig. 7.7. The line to line voltage and line current are found to be 394V rms and 3.77A respectively when DC voltage



Fig. 7.5 Setup for running motor

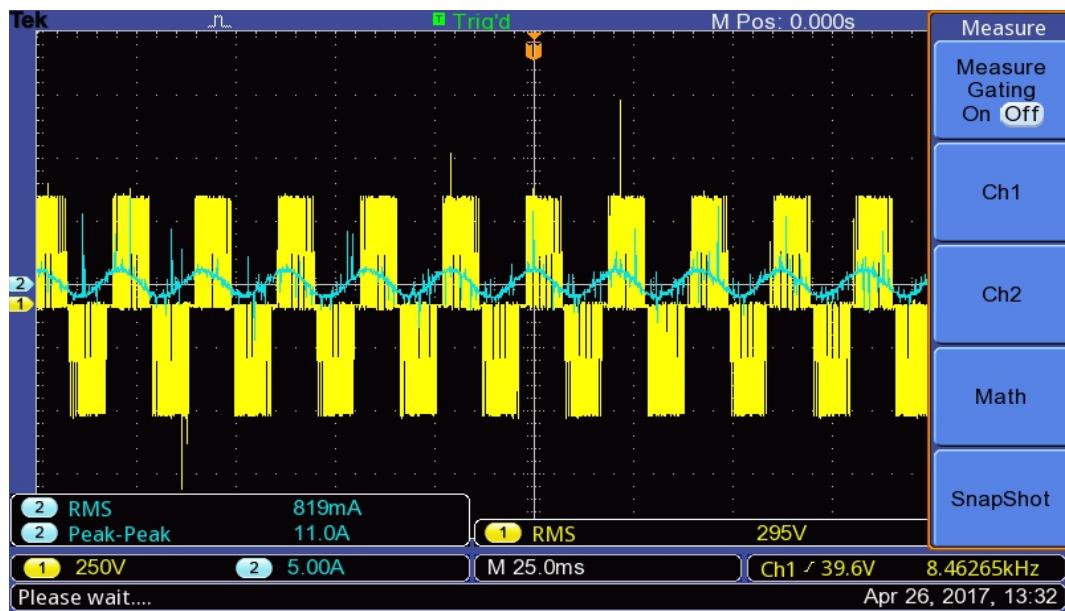


Fig. 7.6 Voltage and current waveforms for 3 phase induction motor running at no load

of 580V is given to the inverter. The same setup is used to run the motor at lesser load and the waveforms were obtained as shown in Fig. 7.8.

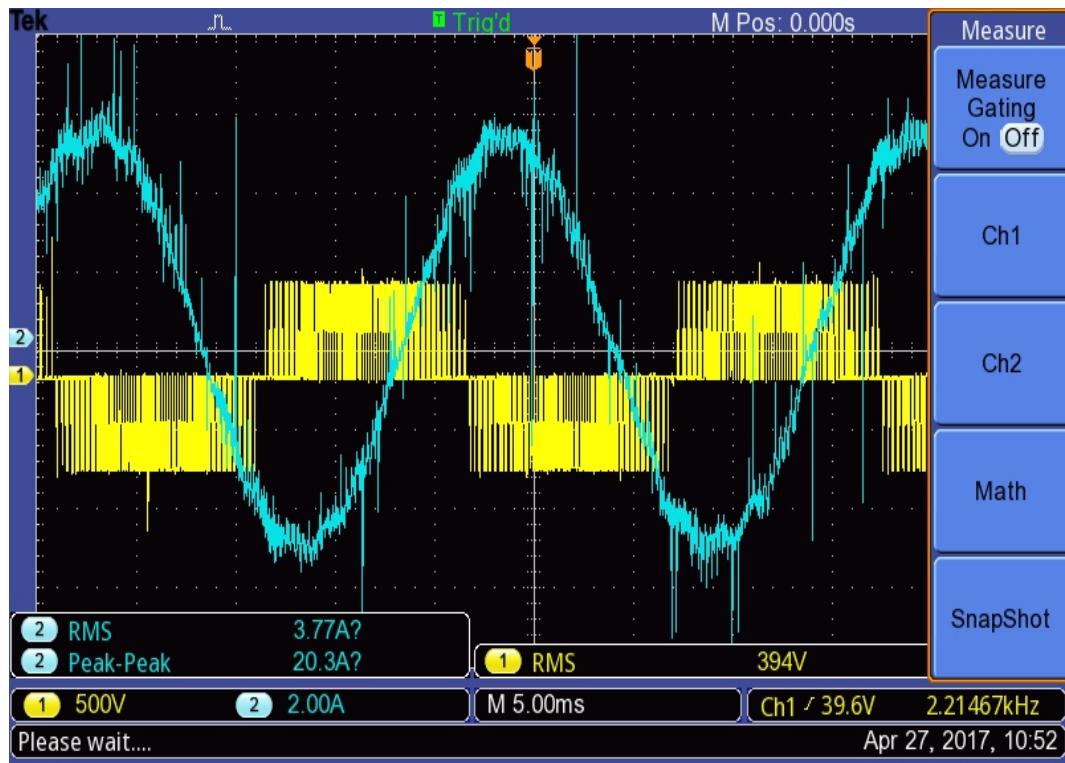


Fig. 7.7 Voltage and current waveforms for 3 phase induction motor running with high load

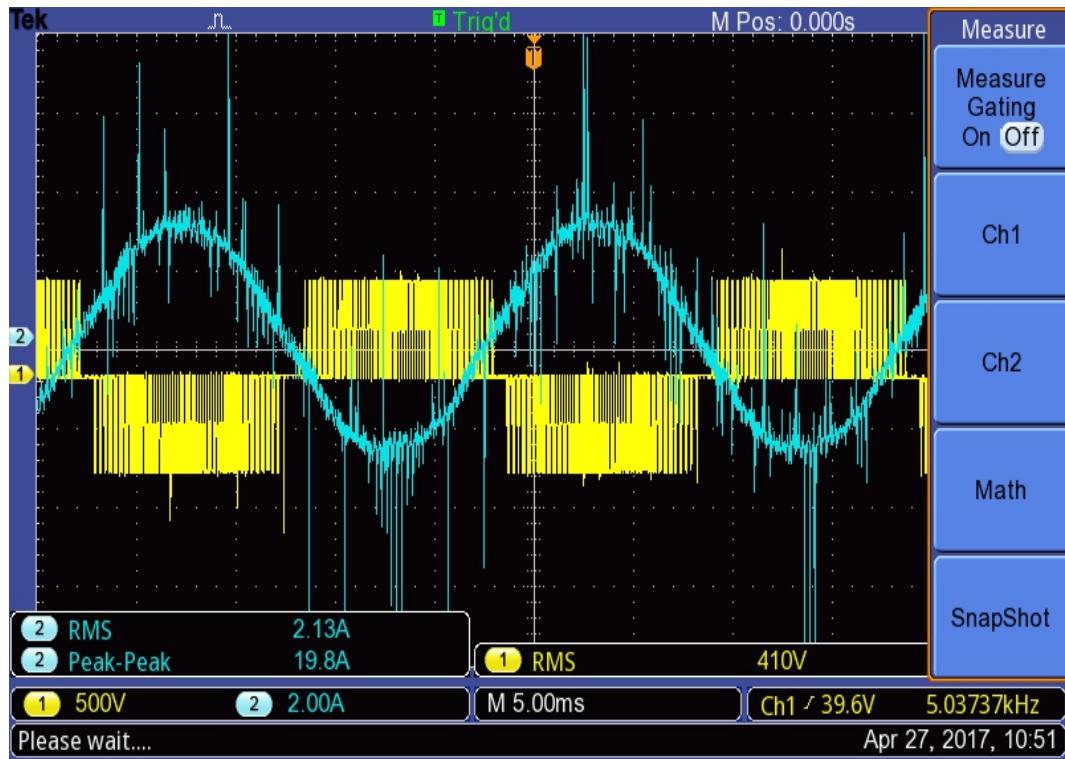


Fig. 7.8 Voltage and current waveforms for 3 phase induction motor running with medium load

7.3 Conclusion and future scope

3 phase Voltage Source Inverter was successfully designed and implemented on a PCB. Initially the simulation of the 415V, 10A 3 phase inverter running a 3 hp 415V, 4.5A 3 phase induction motor was carried out in MATLAB Simulink and waveforms obtained were recorded and analysed. The setup was implemented on a PCB and was used to run a 4hp 3 phase Induction motor and the results obtained were comparable to the simulation results.

PCB design required considerable time for placing, routing and manufacturing and posed as a bottleneck in the completion of the project. However the PCB design used optimal space and modifications can be made to the FPGA terminals and transformer input terminals so that the issue of loose contact can be eliminated. Furthermore, use of filters at the output terminals of the inverter can reduce harmonics in the motor and can ensure smooth operation without much heating losses.

Future scope of this project involves inclusion of a closed loop control technique like V/f control or vector control. Also, a more sophisticated form of space vector PWM can be implemented. The inverter module can be tested for different setups of load variation.

Chapter A

APPENDIX - FPGA CODE

A.1 Modulating signals

```
1  /*This module is used for generating modulating signals.  
2   Input is a clock enable signal to switch on the clock,  
3   clock input signal and a active high reset signal to  
4   power down the control circuit  
5   Output are modulating waves  
6 */  
7  `timescale 1 ns / 1 ns  
8  
9  module ModulatingWaves  
10  
11  (clk,  
12  reset,  
13  clk_enable,  
14  ce_out,  
15  Out1_0,  
16  Out1_1,  
17  Out1_2  
18 );  
19  
20  
21  input  clk;  
22  input  reset;  
23  input  clk_enable;  
24  output ce_out;  
25  output signed [15:0] Out1_0; // int16  
26  output signed [15:0] Out1_1; // int16  
27  output signed [15:0] Out1_2; // int16  
28
```

```

30     wire enb;
31     reg signed [12:0] Sine_Wave_out1; // sfix13
32     reg signed [12:0] Sine_Wave1_out1; // sfix13
33     reg signed [12:0] Sine_Wave2_out1; // sfix13
34     wire signed [12:0] Mux_out1 [0:2]; // sfix13 [3]
35     wire signed [15:0] Data_Type_Conversion_out1 [0:2]; // int16 [3]
36     wire signed [15:0] MinMax_out1; // int16
37     wire signed [15:0] MinMax1_out1; // int16
38     wire signed [15:0] Add2_out1; // int16
39     wire signed [31:0] Gain_out1; // sfix32_En15
40     wire signed [15:0] Add3_v; // sfix16
41     wire signed [15:0] Add3_out1 [0:2]; // int16 [3]
42     reg [14:0] address_cnt; // ufix15
43     reg [14:0] address_cnt_1; // ufix15
44     reg [14:0] address_cnt_2; // ufix15
45
46     reg [4:0] count;
47     reg [4:0] count1;
48     reg [4:0] count2;
49     reg flip_bit;
50     reg flip_bit1;
51     reg flip_bit2;
52
53
54     assign enb = clk_enable;
55
56     /* This counter is used to take in a 50MHz
57      clock and output a 1MHz clock*/
58     always @ (posedge clk or posedge reset)
59     begin
60         if (reset == 1'b1) begin
61             count <= 5'b00000;
62             flip_bit <= 1'b0;
63         end
64         else begin
65             if (enb == 1'b1) begin
66                 if (count == 5'b11001) begin
67                     count <= 5'b00000;
68                     flip_bit <= ~flip_bit;
69                 end
70                 else begin
71                     count <= count + 5'b1;
72                 end
73             end
74         end
75     end
76

```

```

77  /* ADDRESS COUNTER
78      used to iterate through the look up table values using
79      an address register which is updated at a rate of 1MHz
80 */
81  always @ (posedge flip_bit or posedge reset)
82      begin: Sine_Wave_addrcnt_temp_process1
83          if (reset == 1'b1) begin
84              address_cnt <= 15'b0000000000000000;
85          end
86          else begin
87              if (enb == 1'b1) begin
88                  if (address_cnt == 15'b100111000011111) begin
89                      address_cnt <= 15'b0000000000000000;
90                  end
91                  else begin
92                      address_cnt <= address_cnt + 15'b1;
93                  end
94              end
95          end
96      end // Sine_Wave_addrcnt_temp_process1
97
98 // FULL WAVE LOOKUP TABLE - example shown, acutal 20,000 samples
99
100 always @(address_cnt)
101 begin
102     case(address_cnt)
103         15'b0000000000000000 : Sine_Wave_out1 = 13'b0000000000000000;
104         15'b0000000000000001 : Sine_Wave_out1 = 13'b0000000000000001;
105         15'b0000000000000010 : Sine_Wave_out1 = 13'b0000000000000010;
106         15'b0000000000000011 : Sine_Wave_out1 = 13'b0000000000000011;
107         15'b0000000000000100 : Sine_Wave_out1 = 13'b0000000000000100;
108         15'b0000000000000101 : Sine_Wave_out1 = 13'b0000000000000101;
109         15'b0000000000000110 : Sine_Wave_out1 = 13'b0000000000000110;
110         ...
111         ...
112         ...
113         default : Sine_Wave_out1 = 13'b11111111111111;
114     endcase
115 end

```

```

117  /* This counter is used to take in a 50MHz
118    clock and output a 1MHz clock*/
119    always @(posedge clk or posedge reset)
120      begin
121        if (reset == 1'b1) begin
122          count1 <= 5'b00000;
123          flip_bit1 <= 1'b0;
124        end
125        else begin
126          if (enb == 1'b1) begin
127            if (count1 == 5'b11001) begin
128              count1 <= 5'b00000;
129              flip_bit1 <= ~flip_bit1;
130            end
131            else begin
132              count1 <= count1 + 5'b1;
133            end
134          end
135        end
136      end
137
138  /* ADDRESS COUNTER
139    used to iterate through the look up table values using
140    an address register which is updated at a rate of 1MHz,
141    120 degree shifted sine wave
142 */
143  always @ (posedge flip_bit1 or posedge reset)
144    begin: Sine_Wave1_addrcnt_temp_process2
145      if (reset == 1'b1) begin
146        address_cnt_1 <= 15'b000000000000000000;
147      end
148      else begin
149        if (enb == 1'b1) begin
150          if (address_cnt_1 == 15'b100111000011111) begin
151            address_cnt_1 <= 15'b0000000000000000;
152          end
153          else begin
154            address_cnt_1 <= address_cnt_1 + 15'b1;
155          end
156        end
157      end
158    end // Sine_Wave1_addrcnt_temp_process2
159

```

```

162 // FULL WAVE LOOKUP TABLE - example shown, actual 20,000 samples
163     always @(address_cnt_1)
164     begin
165         case(address_cnt_1)
166             15'b0000000000000000 : Sine_Wave1_out1 = 13'b0110000110100;
167             15'b0000000000000001 : Sine_Wave1_out1 = 13'b0110000110011;
168             15'b0000000000000010 : Sine_Wave1_out1 = 13'b0110000110011;
169             15'b0000000000000011 : Sine_Wave1_out1 = 13'b0110000110010;
170             15'b00000000000000100 : Sine_Wave1_out1 = 13'b0110000110001;
171             15'b00000000000000101 : Sine_Wave1_out1 = 13'b0110000110001;
172             ...
173             ...
174             ...
175             default : Sine_Wave1_out1 = 13'b0110000110100;
176         endcase
177     end
181 /* This counter is used to take in a 50MHz
182 clock and output a 1MHz clock*/
183     always @(posedge clk or posedge reset)
184     begin
185         if (reset == 1'b1) begin
186             count2 <= 5'b00000;
187             flip_bit2 <= 1'b0;
188         end
189         else begin
190             if (enb == 1'b1) begin
191                 if (count2 == 5'b11001) begin
192                     count2 <= 5'b00000;
193                     flip_bit2 <= ~flip_bit2;
194                 end
195                 else begin
196                     count2 <= count2 + 5'b1;
197                 end
198             end
199         end
200     end

```

```

202  /* ADDRESS COUNTER
203      used to iterate through the look up table values using
204      an address register which is updated at a rate of 1MHz,
205      240 degree shifted sine wave
206 */
207  always @ (posedge flip_bit2 or posedge reset)
208      begin: Sine_Wave2_addrcnt_temp_process3
209          if (reset == 1'b1) begin
210              address_cnt_2 <= 15'b0000000000000000;
211          end
212          else begin
213              if (enb == 1'b1) begin
214                  if (address_cnt_2 == 15'b10011100001111) begin
215                      address_cnt_2 <= 15'b0000000000000000;
216                  end
217                  else begin
218                      address_cnt_2 <= address_cnt_2 + 15'b1;
219                  end
220              end
221          end
222      end // Sine_Wave2_addrcnt_temp_process3
227  // FULL WAVE LOOKUP TABLE - example shown, actual 20,000 samples
228  always @(address_cnt_2)
229      begin
230          case(address_cnt_2)
231              15'b0000000000000000 : Sine_Wave2_out1 = 13'b1001111001100;
232              15'b0000000000000001 : Sine_Wave2_out1 = 13'b1001111001100;
233              15'b0000000000000010 : Sine_Wave2_out1 = 13'b1001111001011;
234              15'b0000000000000011 : Sine_Wave2_out1 = 13'b1001111001011;
235              15'b00000000000000100 : Sine_Wave2_out1 = 13'b1001111001010;
236              15'b00000000000000101 : Sine_Wave2_out1 = 13'b1001111001001;
237              ...
238              ...
239              ...
240              default : Sine_Wave2_out1 = 13'b1001111001101;
241      endcase
242  end

```

```

246     assign Mux_out1[0] = Sine_Wave_out1;
247     assign Mux_out1[1] = Sine_Wave1_out1;
248     assign Mux_out1[2] = Sine_Wave2_out1;
249
250     assign Data_Type_Conversion_out1[0] = Mux_out1[0];
251     assign Data_Type_Conversion_out1[1] = Mux_out1[1];
252     assign Data_Type_Conversion_out1[2] = Mux_out1[2];
253
254
255 //Instantaneous values are compared using min and max submodules
256
257     MinMax    u_MinMax    (.in0_0(Data_Type_Conversion_out1[0]), // int16
258                             .in0_1(Data_Type_Conversion_out1[1]), // int16
259                             .in0_2(Data_Type_Conversion_out1[2]), // int16
260                             .out0(MinMax_out1) // int16
261 );
262
263     MinMax1   u_MinMax1   (.in0_0(Data_Type_Conversion_out1[0]), // int16
264                             .in0_1(Data_Type_Conversion_out1[1]), // int16
265                             .in0_2(Data_Type_Conversion_out1[2]), // int16
266                             .out0(MinMax1_out1) // int16
267 );
268
269
270 //obtained min and max values are added
271 assign Add2_out1 = MinMax_out1 + MinMax1_out1;
272
273
274 //addition output is divided by 2
275 assign Gain_out1 = {{2{Add2_out1[15]}}, {Add2_out1, 14'b0000000000000000}};
276
277
278 //obtained output is subtracted from sine waves, modulating signals obtained
279 assign Add3_v = Gain_out1[30:15];
280 assign Add3_out1[0] = Data_Type_Conversion_out1[0] - Add3_v;
281 assign Add3_out1[1] = Data_Type_Conversion_out1[1] - Add3_v;
282 assign Add3_out1[2] = Data_Type_Conversion_out1[2] - Add3_v;
283
284
285
286 assign Out1_0 = Add3_out1[0];
287
288 assign Out1_1 = Add3_out1[1];
289
290 assign Out1_2 = Add3_out1[2];
291
292 assign ce_out = clk_enable;
293
294 endmodule // ModulatingWaves

```

A.2 Submodule-1

```
1  /*This module is a submodule of the modulating signal
2   generator. It compares the three sinusoidal signal
3   instantaneously and the output is the the minimum value
4   of the three signals*/
5
6  `timescale 1 ns / 1 ns
7
8  module MinMax
9    (
10      in0_0,
11      in0_1,
12      in0_2,
13      out0
14    );
15
16
17  input  signed [15:0] in0_0; // int16
18  input  signed [15:0] in0_1; // int16
19  input  signed [15:0] in0_2; // int16
20  output signed [15:0] out0; // int16
21
22
23  wire signed [15:0] in0 [0:2]; // int16 [3]
24  wire signed [15:0] MinMax_stage1_val [0:1]; // int16 [2]
25  wire signed [15:0] MinMax_stage2_val; // int16
26
27
28  assign in0[0] = in0_0;
29  assign in0[1] = in0_1;
30  assign in0[2] = in0_2;
31
32  // ---- Tree min implementation ----
33  // ---- Tree min stage 1 ----
34  assign MinMax_stage1_val[0] = (in0[0] <= in0[1] ? in0[0] :
35  in0[1]);
36  assign MinMax_stage1_val[1] = in0[2];
37
38
39
40  // ---- Tree min stage 2 ----
41  assign MinMax_stage2_val = (MinMax_stage1_val[0] <=
42  MinMax_stage1_val[1] ? MinMax_stage1_val[0] :
43  MinMax_stage1_val[1]);
44
45
46
47  assign out0 = MinMax_stage2_val;
48
49 endmodule // MinMax
```

A.3 Submodule-2

```
1  /*This module is a submodule of the modulating signal
2   generator. It compares the three sinusoidal signal
3   instantaneously and the output is the the maximum value
4   of the three signals*/
5
6  `timescale 1 ns / 1 ns
7
8  module MinMax1
9    (
10      in0_0,
11      in0_1,
12      in0_2,
13      out0
14    );
15    input  signed [15:0] in0_0; // int16
16    input  signed [15:0] in0_1; // int16
17    input  signed [15:0] in0_2; // int16
18    output signed [15:0] out0; // int16
19
20    wire signed [15:0] in0 [0:2]; // int16 [3]
21    wire signed [15:0] MinMax1_stage1_val [0:1]; // int16 [2]
22    wire signed [15:0] MinMax1_stage2_val; // int16
23
24
25    assign in0[0] = in0_0;
26    assign in0[1] = in0_1;
27    assign in0[2] = in0_2;
28
29    // ---- Tree max implementation ----
30    // ---- Tree max stage 1 ----
31    assign MinMax1_stage1_val[0] = (in0[0] >= in0[1] ? in0[0] :
32    in0[1]);
33    assign MinMax1_stage1_val[1] = in0[2];
34
35
36
37    // ---- Tree max stage 2 ----
38    assign MinMax1_stage2_val = (MinMax1_stage1_val[0] >=
39    MinMax1_stage1_val[1] ? MinMax1_stage1_val[0] :
40    MinMax1_stage1_val[1]);
41
42
43
44    assign out0 = MinMax1_stage2_val;
45
46  endmodule // MinMax1
```

A.4 Triangle Wave Generator

```
1  /*This module is used for generating triangular waves.
2   Input is a clock enable signal to switch on the clock,
3   clock input signal and a active high reset signal to
4   power down the control circuit
5   Output are triangular waves of 4KHz waves
6 */
7
8  `timescale 1 ns / 1 ns
9
10 module TriangularWaves
11   (
12     input  clk,
13     input  reset,
14     input  clk_enable,
15     output ce_out,
16     output Out5
17   );
18
19
20   input  clk;
21   input  reset;
22   input  clk_enable;
23   output ce_out;
24   output  signed [15:0] Out5; // int16
25
26   wire enb;
27   wire signed [15:0] Constant_out1; // int16
28   reg [15:0] Counter_Limited_count; // ufix16
29   wire [15:0] Counter_Limited_out1; // uint16
30   wire signed [15:0] Add_out1; // int16
31   wire switch_compare_1;
32   wire signed [15:0] Counter_Limited_out1_dtc; // int16
33   wire signed [15:0] Switch_out1; // int16
34   wire signed [15:0] Constant1_out1; // int16
35   wire signed [15:0] Add1_out1; // int16
36
37
38
39   assign Constant_out1 = 16'sd12499;
```

```

43     assign enb = clk_enable;
44
45     // Count limited, Unsigned Counter
46     // initial value      = 0
47     // step value        = 1
48     // count to value    = 12499
49     always @(posedge clk or posedge reset)
50         begin : Counter_Limited_process
51             if (reset == 1'b1) begin
52                 Counter_Limited_count <= 16'b00000000000000000000;
53             end
54             else begin
55                 if (enb) begin
56                     if (Counter_Limited_count == 16'b0011000011010011) begin
57                         Counter_Limited_count <= 16'b0000000000000000;
58                     end
59                     else begin
60                         Counter_Limited_count <= Counter_Limited_count + 16'b1;
61                     end
62                 end
63             end
64         end
65
66     assign Counter_Limited_out1 = Counter_Limited_count;
67     //counter value instantaneously subtracted from constant value 12499
68     assign Add_out1 = Constant_out1 - $signed({1'b0, Counter_Limited_out1});
69
70
71
72
73     //output of the two opposing counters is selected as per the following
74     assign switch_compare_1 = (Add_out1 >= 16'sb0001100001101001 ? 1'b1 :
75                                         1'b0);
76
77
78
79     assign Counter_Limited_out1_dtc = Counter_Limited_out1;
80
81
82     //output is selected from previous comparison
83     assign Switch_out1 = (switch_compare_1 == 1'b0 ? Counter_Limited_out1_dt
84                                         Add_out1);
85
86
87
88     assign Constant1_out1 = 16'sd9374;

```

A.5 Comparator

```
1  /*This module is used to compare the input modulating signals
2   and triangular carrier waves to generate the required IGBT
3   gate control signals*/
4   `timescale 1 ns / 1 ns
5
6   module Comparator
7   (
8     In1, //Modulating Signals
9     In2, //Triangular waves
10    Out1 //IGBT gate control signals
11  );
12  input  signed [15:0] In1; // int16
13  input  signed [15:0] In2; // int16
14  output Out1;
15
16
17  wire Relational_Operator_relop1;
18
19
20  assign Relational_Operator_relop1 = (In1 > In2 ? 1'b1 :
21  1'b0);
22
23
24
25  assign Out1 = Relational_Operator_relop1;
26
27  endmodule // Comparator
```

REFERENCES

- [1] *Space Vector PWM as a Modified Form of Sine-Triangle PWM for Simple Analog or Digital Implementation-* P Srikant Varma ang G Narayanan.
- [2] *FPGAs enable energy-efficient motor control-* Jason Chiang
- [3] *Drive Circuits for power MOSFETs and IGBTs-* B.Maurice, L.Wuidart.
- [4] *Linear Voltage Regulators-* Littelfuse, Inc.
- [5] *Smart Gate Driver Coupler TLP5214-* Toshiba Corporation.
- [6] *Protection Circuit Design-* Fuji Electric Co.,Ltd.
- [7] *Gate Drive circuit Design-* Fuji Electric Co.,Ltd.
- [8] *Digital design, 4th edition-* M.Morris Mano.
- [9] *Power Electronics, 2nd edition-* M. D. Singh.