

Project Report

Cascaded H-Bridge Inverter with 9 Levels in Voltage Waveform



Submitted by:

Usman IlamDin	2017-EE-119
Arslan Shafique	2017-EE-118
Khawaja Hammad	2017-EE-135
Hamza Naseer	2017-EE-149

Supervised by: Sir Aneeq Aslam

Department of Electrical Engineering
University of Engineering and Technology Lahore

Declaration

We declare that the work contained in this thesis/report is my own, except where explicitly stated otherwise. In addition this work has not been submitted to obtain another degree or professional qualification.

Signed: _____

Signed: _____

Signed: _____

Signed: _____

Date: _____

Contents

List of Figures	iv
Abstract	vi
1 Literature Review	1
1.1 Introduction	1
1.1.1 MOSFET	1
1.2 Hierarchy of Inverter	2
1.3 Voltage Source Inverter	2
1.4 Multilevel Inverters	3
1.5 H Bridge	4
1.6 Cascaded 9-level H Bridge	4
1.6.1 4 units of H bridge in series	6
1.6.2 2 units of H bridge in series	7
1.6.3 Features of Cascaded Inverter	8
1.6.4 Limitations of Cascaded Inverter	8
1.7 Modulation Techniques for Multilevel Inverter	9
1.8 Classification of Modulation Techniques	9
1.9 Sinusoidal PWM	10
1.9.1 Level-shifted carrier based PWM (LSCPWM) technique	10
1.9.2 Phase shifted carrier based PWM (PSCPWM) technique	10
2 Flow Chart and Block Diagram	12
2.1 Flowchart	12
2.2 Block Diagram	13
3 Circuit Design and Analysis	14
3.1 Terminologies	14
3.2 Design	15
3.3 Analysis	17
4 Detailed Circuit Design	22
4.1 DC Supplies	23
4.2 STM32F4x Microntroller	23
4.3 Gate triggering circuit	23
4.4 H-Bridge Circuits	25

5 Simulation Results	27
5.1 Gate Signals	27
5.2 Source Currents	28
5.3 Switch Currents	29
5.4 Load Current	31
5.5 Output Voltage	32
5.6 Output Generated by PWM Technique	34
5.7 Output Generated by Sine PWM Technique	35
5.8 AND Operation with PWM Signal	36
5.9 AND Operation with Sine PWM Signal	37
5.10 SHE PWM	38
5.11 Voltage Control	41
6 Hardware Results	42
6.1 Output Voltage	42
6.2 Verification of Conduction Angles	43
6.2.1 Verification of α_1	43
6.2.2 Verification of α_2	44
6.2.3 Verification of α_3	45
6.2.4 Verification of α_4	46
6.3 AND Operation with 10KHz PWM	47
6.4 PWM Operation	48
6.5 Frequency Control	48
6.6 Voltage Control	49
6.6.1 Table	56
6.6.2 Relation Between Duty Cycle and V_{rms}	56
6.7 Voltage and Current Parameters	57
7 Conclusion	58
A Code for Project	59
A.1 STM32F4 Code	59
A.2 MATLAB CODE	64
A.2.1 Switching Sequence Generator:	64
A.2.2 Sine PWM Generator	66
A.2.3 SHE PWM	67
A.2.4 Equations	67

List of Figures

1.1	Outputs of Inverters	1
1.2	MOSFET showing gate (G), body (B), source (S) and drain (D) terminals. The gate is separated from the body by an insulating layer (white)	2
1.3	Hierarchy of inverter	2
1.4	Generalized stepped waveform of multilevel inverters	4
1.5	Unit H Bridge	5
1.6	Quasi square wave	5
1.7	Cascaded Inverter Circuit(m=4)	6
1.8	Cascaded Inverter nine levels of output voltage	7
1.9	Classification of Modulation Techniques	9
2.1	Flowchart of Project Progress	12
2.2	Block Diagram of Project Progress	13
3.1	H Bridge circuit for 9 Levels Voltage wave-forms	16
3.2	Switching States for 9 Levels Voltage wave-forms	16
3.3	Graphical Representation of Switching Sequence with Minimum Transitions	17
3.4	9-Levels Voltage waveform when E=12 V	18
3.5	9-Levels Voltage Magnitude Spectrum	18
3.6	Better Approximated 9 levels Voltage waveform	20
3.7	Improved Fourier Spectrum	20
4.1	Circuit for 9 Levels Voltage wave-forms	22
4.2	Circuit for Constant DC supply	23
4.3	Circuit for TLP250	24
4.4	Inner circuit for TLP250	24
4.5	H-Bridge 1 With DC Source = E volts	25
4.6	H-Bridge 2 With DC Source = 3E volts	25
4.7	H-Bridge 1 With Gate Driver Circuit	26
4.8	H-Bridge 2 With Gate Driver Circuit	26
5.1	Gate Signals to H-Bridge 1	27
5.2	Gate Signals to H-Bridge 2	28
5.3	Source Currents (Resistive Load)	28
5.4	Source Currents (Inductive Load 0.8PF)	29
5.5	Switch Currents of H-Bridge 1 (Resistive Load)	29
5.6	Switch Currents of H-Bridge 2 (Resistive Load)	30
5.7	Switch Currents of H-Bridge 1 (Inductive Load 0.8PF)	30
5.8	Switch Currents of H-Bridge 2 (Inductive Load 0.8PF)	31

5.9	Load Current (Resistive Load)	31
5.10	Load Current (Inductive Load 0.8PF)	32
5.11	Load Voltage and Scaled Current (Inductive Load 0.8PF)	32
5.12	Output Voltage	33
5.13	Output Voltage FFT Analysis	33
5.14	Record of First 10 Harmonics	34
5.15	Output Signal Generated by PWM Technique	34
5.16	FFT of Output Signal Generated by PWM Technique	35
5.17	Output Signal Generated by Sine PWM Technique	35
5.18	FFT of Output Signal Generated by Sine PWM Technique	36
5.19	Output Signal after AND Operation	36
5.20	FFT of Output Signal after AND Operation	37
5.21	Output Signal after AND Operation	37
5.22	FFT of Output Signal after AND Operation	38
5.23	Output Signal (SHE PWM)	39
5.24	FFT of Output Signal (SHE PWM)	40
5.25	FFT Record Output Signal (SHE PWM)	40
5.26	Relation between Duty Cycle of PWM and V_{rms}	41
6.1	Output Voltage at 50Hz	42
6.2	Output Voltage with all Load Parameters	43
6.3	Verification of $\alpha_1 = 6^\circ$	44
6.4	Verification of $\alpha_2 = 22^\circ$	45
6.5	Verification of $\alpha_3 = 38^\circ$	46
6.6	Verification of $\alpha_4 = 60^\circ$	47
6.7	AND Operation with 10KHz PWM	47
6.8	PWM Operation at 10KHz	48
6.9	Frequency = 200Hz	48
6.10	Duty Cycle = 100%	49
6.11	Duty Cycle = 90%	49
6.12	Duty Cycle = 80%	50
6.13	Duty Cycle = 70%	50
6.14	Duty Cycle = 60%	51
6.15	Duty Cycle = 50%	51
6.16	Duty Cycle = 40%	52
6.17	Duty Cycle = 30%	52
6.18	Duty Cycle = 20%	53
6.19	Duty Cycle = 10%	53
6.20	Duty Cycle = 5%	54
6.21	Duty Cycle = 2%	54
6.22	Duty Cycle = 1%	55
6.23	Duty Cycle = 0%	55
6.24	Relation Between Duty Cycle and V_{rms}	56

Abstract

Cascaded H-Bridge Multilevel Inverters can generate high quality voltage waveforms close to sinusoidal. CHB Multilevel inverter that is constituted using dual H-bridge cells with MOSFETs on each phase legs. The output waveform is increased up to 9-level by using 2 separate DC sources defined at $V_{dc}:3V_{dc}$ ratio to obtain an asymmetrical output. The cascaded multilevel inverter is switched by the proposed SPWM modulator. The harmonic analysis of proposed inverter has been performed under several working conditions such as various switching frequencies and modulation indexes. The detailed comparisons are performed to determine the best working conditions of voltage source inverters (VSI) and presented in this report....

Chapter 1

Literature Review

1.1 Introduction

A power inverter, or inverter, is an electronic device that converts direct current (DC) to alternating current (AC). The input voltage, output voltage and frequency, and overall power handling depend on the design of the specific circuitry[1].

Inverters can be divided into two kinds:

- Voltage Source Inverters (VSI)
- Current Source Inverters (CSI)

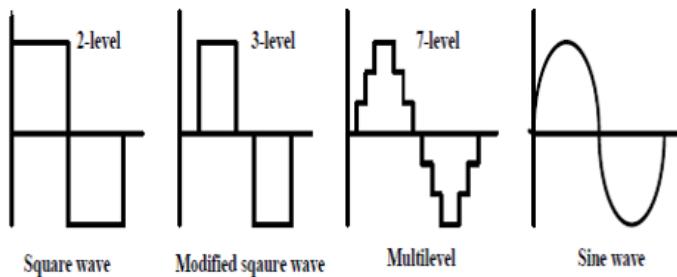


FIGURE 1.1: Outputs of Inverters

The basic component of an inverter is fast working switch like IGBT or MOSFET. However, later is more preferred because main advantage of a MOSFET is that it requires almost no input current to control the load current, when compared with IGBT.

1.1.1 MOSFET

The MOSFET^{4.2} is a type of field-effect transistor (FET), most commonly fabricated by the controlled oxidation of silicon. It has an insulated gate, whose voltage determines the conductivity of the device. This ability to change conductivity with the amount of applied voltage can be used for amplifying or switching electronic signals.

The Mosfet is further divided into n-channel Mosfet and p-channel Mosfet depending upon formation and has three modes of operation as :

- Cutoff mode
- Ohmic mode
- Active mode

For switching purposes Mosfet is used in active and cutoff mode.

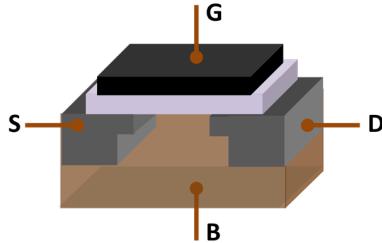


FIGURE 1.2: MOSFET showing gate (G), body (B), source (S) and drain (D) terminals.
The gate is separated from the body by an insulating layer (white)

1.2 Hierarchy of Inverter

General Hierarchy of inverter^{4.3} is shown below.Under the perspective of project the selective hierarchy will start from basic inverter to unequal dc sources cascaded h-bridge multilevel inverters.

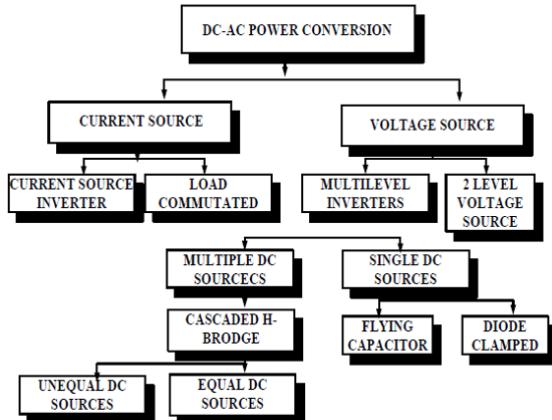


FIGURE 1.3: Hierarchy of inverter

1.3 Voltage Source Inverter

If the input dc is voltage source, the inverter is called a voltage source inverter.The VSI circuit has direct control over output (ac) voltage. Shape of voltage wave-forms output

by an ideal VSI should be independent of load connected at the output.

The simplest dc voltage source for a VSI may be a battery bank, which may consist of several cells in series-parallel combination. A voltage source is called stiff, if the source voltage magnitude does not depend on load connected to it. All VSI assume stiff voltage supply at the input. Some examples where voltage source inverters are used are: UPS units, adjustable speed drives (Automatic Star Delta) for ac motors, electronic frequency changer circuits etc.

The achievable magnitude of ac voltage is limited by the magnitude of input dc voltage. In some cases the inverter output voltage is stepped up using a transformer to meet the load requirement. VSI is further divided into two types:

- Multilevel Inverters
- 2 Level Voltage Source

1.4 Multilevel Inverters

A multilevel inverter is a PE device which is capable of providing desired alternating voltage level at the output using multiple lower level DC voltages as an input. Multilevel inverters also enables the use of low power application in renewable energy sources. These converters are suitable in high voltage and high power applications due to:

- Their ability to synthesize higher voltages with a limited maximum device rating
- Less harmonic distortion
- Producing of smaller common-mode voltage
- Less electromagnetic compatibility problems
- Attain higher voltage with a limited maximum device rating.

The range of the output power is a very important and evident limitation of two-level inverter. However, this problem can be overcome by introducing the concept of multilevel inverters.

The three most popular multilevel inverters are:

- Diode-Clamped Multilevel Inverter (DC-MLI)
- Flying-Capacitor Multilevel Inverter (FC-MLI)
- Cascaded H-Bridge Multilevel Inverter (CHBMLI)

First two use single voltage source for working while CHBMLI use multiple voltage sources.

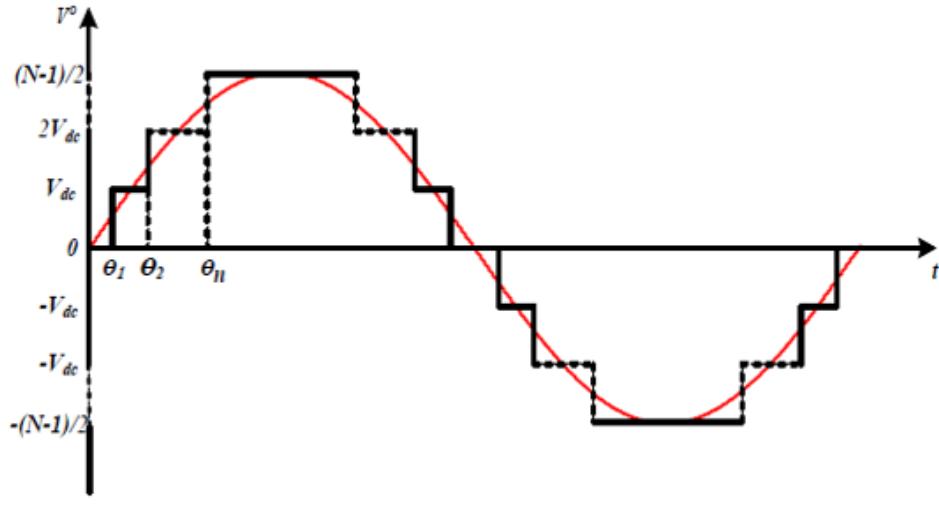


FIGURE 1.4: Generalized stepped waveform of multilevel inverters

1.5 H Bridge

An H bridge 4.8 enables a reverse voltage to be applied across a load.

These circuits are often used in multiple applications to allow bidirectional movement of DC motors, DC-to-AC converters, AC/AC converters, DC-to-DC *push-pull* converter and many other kinds of power electronics use H bridges.

An H bridge is built with four switches 4.2. When the switches S1 and S4 are closed (and S2 and S3 are open) a positive voltage will be applied across the motor and vice versa. The point to ponder is that switches S1 and S2 should never be closed at the same time, as this would cause a short circuit on the input voltage source as seen from the figure. The same applies to the switches S3 and S4.

Thus, functioning of a single H-bridge is similar to that of conventional 2-level inverter. Each H-bridge requires an isolated dc source/capacitor to generate its corresponding output. The switches are activated in such a way that the output voltage across the load terminals is the aggregation of the voltage generated by the H-bridge.

1.6 Cascaded 9-level H Bridge

The concept of series H-bridge inverter was first proposed by R. H. Baker and L. H. Banister in 1975. To overcome the drawbacks of Neutral-Point Clamped and Flying Capacitor topologies such as extra clamping diodes and capacitors, Marchesoni have proposed CHB. This modular structure has been subsequently extended for three-phase applications.

CHB-MLI is formed by the series connection of two or more single-phase H-bridge inverters; hence the name H bridge is given . Each H-bridge corresponds to two voltage source phase legs, where the line-line voltage is the inverter output voltage. Therefore, three different voltage levels i.e. $+V$, $-V$ and 0 are generated using a single H-bridge

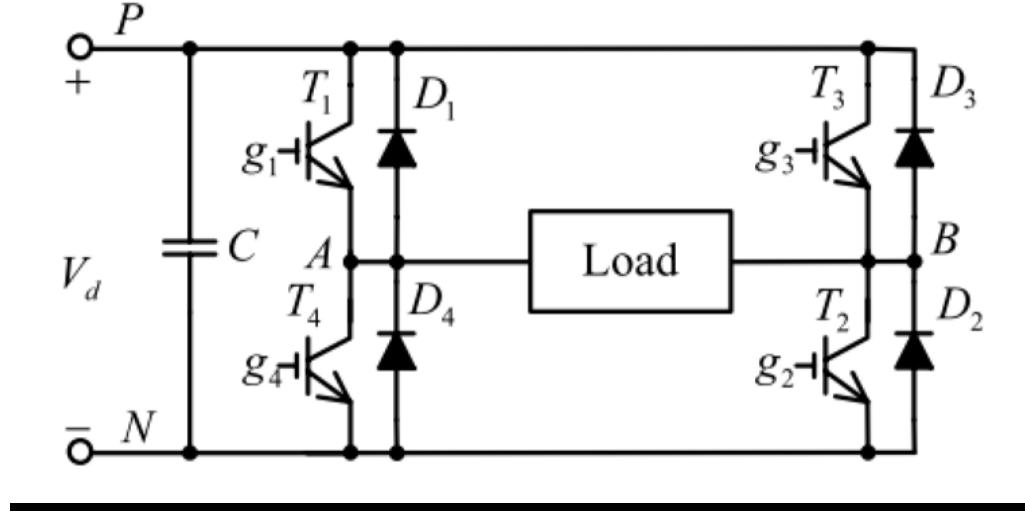
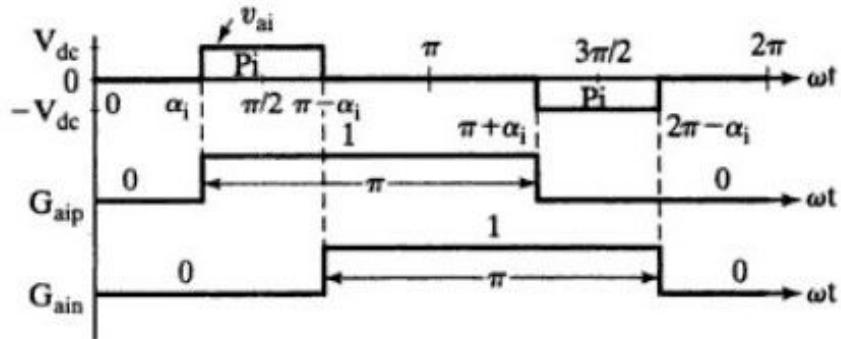


FIGURE 1.5: Unit H Bridge

inverter.

Series connection of N such bridges can produce $2N+1$ levels in the output. This connection is termed as cascaded H-bridge multilevel inverter. Each leg has only two possible switching states, to neglect dc-link capacitor short-circuit. For two legs, four different switching states are possible, although two of them have redundant output voltage.

If E Volts voltage source is connected with a unit H bridge, it gives 3 levels ($E, -E$ including 0 V) square wave, more precisely Quasi-square wave^{4.3} with voltage peak of E volts.



G_{aip}, G_{ain} is 1 if an upper switch is on and 0 if a lower switch is on

FIGURE 1.6: Quasi square wave

Considering above nomenclature in order to get 9-level staircase output two algorithms can be followed as described below.

Cascading 4 units of H bridge(equal source voltage) in series

Cascading 2 units of H bridge(unequal source voltage) in series

1.6.1 4 units of H bridge in series

In this circuit ?? 4 units of H bridge with equal value of source voltage are connected in series to get 9 levels of voltage as output such that when all the circuits give E outputs then we get total of 4E maximum and when all give -E volts as output we get -4E minimum collective. so we get a range of variable output from -4E to 4E.

However, this circuit is easy to apply but is large in size and has large no of switching resulting thus increase in temperature of the whole circuit. So, generally not reliable.

The ac outputs of each full-bridge inverter are connected in series such that the synthesized voltage waveform is the sum of the inverter outputs. For an nine-level inverter which contains four separate sources, the per phase voltage is given by:

$$V_{an} = V_{a1} + V_{a2} + V_{a3} + V_{a4}$$

The required circuit?? and expected waveforms 1.9 are as shown.

By increasing the number of levels we can get a more sinusoidal waveform. The

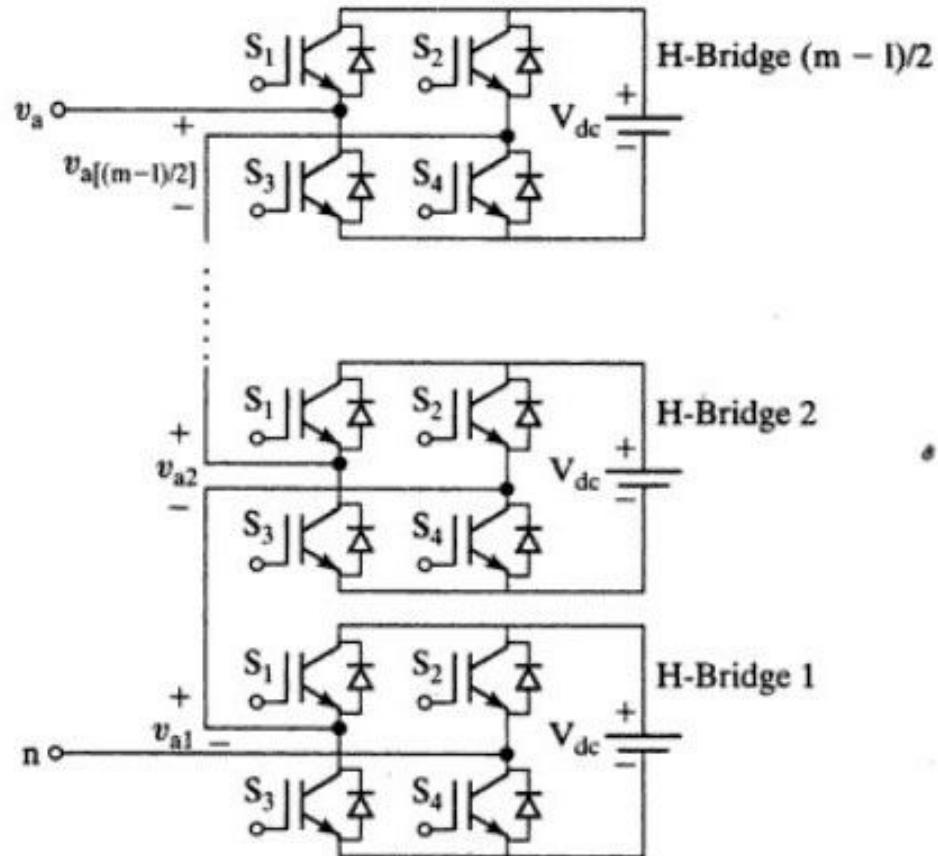


FIGURE 1.7: Cascaded Inverter Circuit($m=4$)

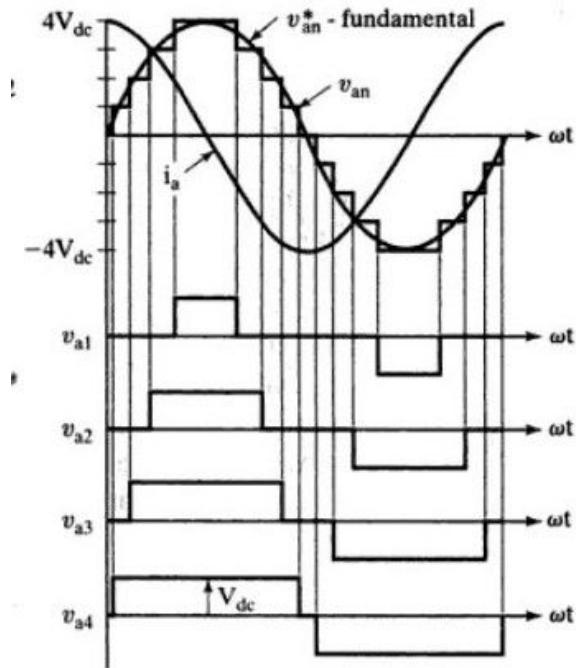


FIGURE 1.8: Cascaded Inverter nine levels of output voltage

conducting angles^{1.9} $\theta_1, \theta_2, \theta_3$ and θ_4 can be chosen such that the voltage total harmonic distortion is minimum. Generally, these angles are chosen so that predominant lower frequency harmonics 3, 5, 7, 11, and 13 are eliminated.

Thus, staircase waveform is obtained from the CHB multilevel inverter which can be nearly sinusoidal as the number of levels has increased, even without using filters. For a three-phase system, the output voltage of three single-phase cascaded converters can be connected in either \star or δ configurations.

1.6.2 2 units of H bridge in series

In this circuit 2 units of H bridge one with source voltage of E and other with 3E are connected in series to get 9 levels of voltage as output. We get a range of variable output from $-4E$ to $4E$ as shown:

Table 9 levels of voltage		
3E source voltage contained H bridge	E source voltage contained H bridge	Value of voltage as resultant output
on	on	4E
on	off	3E
on	inverted	2E
off	on	E
off	off	0
off	inverted	-E
inverted	on	-2E
inverted	off	-3E
inverted	inverted	-4E

The output voltage of inverter is approximately sine wave with a Total Harmonic Distortion(THD) of less than 5 percent as each switch is working at 180 degree at its fundamental frequency.

1.6.3 Features of Cascaded Inverter

- The number of possible output voltage levels say m is more than twice the number of dc sources say s ($m = 2s + 1$).
- Ability to reach higher output voltage and power levels.
- Capable of reaching medium output voltage levels using lower rating switching components.
- Repairing and replacement of faulty module is easy because of its high degree of modularity.
- Selecting an appropriate control strategy in the case of fault conditions can bypass the fault module and can ensure continuous current to the load, bringing an almost continuous over all availability.
- Ability to synthesize output voltage waveform with lesser value of total harmonic distortion.

1.6.4 Limitations of Cascaded Inverter

- Separate dc sources are required for each of the H-bridges.
- As the number of levels increases, more number of switching devices are required in this configuration. This requirement further increases in three-phase configuration.

1.7 Modulation Techniques for Multilevel Inverter

The inverter dc voltage input V_d is usually fixed while its ac output voltage $V_{(AB)}$ can be adjusted by different modulation schemes. Main objectives of modulation strategy are as follows:

- Capable of operating wide range of modulation index, preferably from 0 to 1
- Less switching loss with improved overall efficiency
- Less THD in output voltage
- Obtaining high magnitude of the output fundamental frequency component
- Easy for implementation for practical applications
- Less computational burden and time

However, for the inverters used in high power applications, THD, switching losses, switching capabilities and inverter efficiency are the critical issues that must be taken into account in performance evaluation.

1.8 Classification of Modulation Techniques

Based on switching frequency for computational work modulation techniques are classified as follows:

- Sinusoidal PWM and Space Vector PWM techniques for high switching frequency.
- Selective Harmonic Elimination and Space Vector Control for fundamental switching frequency

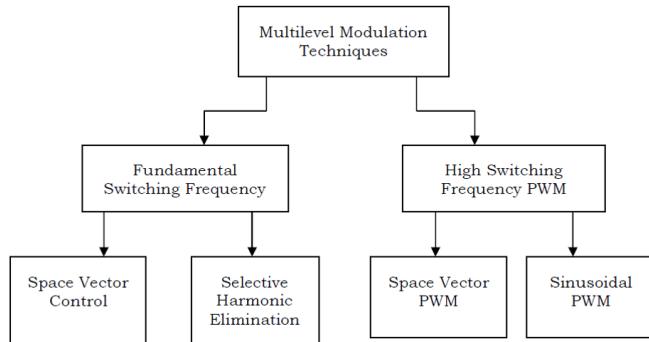


FIGURE 1.9: Classification of Modulation Techniques

1.9 Sinusoidal PWM

Carrier based sinusoidal PWM techniques which are employed in control CHBMLI can be generally classified into two categories:

- Phase-shifted carrier based pulse width modulation technique.
- Level-shifted carrier based pulse width modulation technique.

1.9.1 Level-shifted carrier based PWM (LSCPWM) technique

Level-shifted carrier based PWM (LSCPWM) technique is used to control neutral point clamped inverter and to control cascaded multilevel inverters. This technique has drawbacks such as uneven distribution of power among cells and high THD in output voltage and current wave forms. LSCPWM is further divided into three categories:

- Phase Disposition (PD-PWM): wherein all the carrier signals are in same phase.
- Phase Opposition Disposition (POD-PWM): wherein the carrier signals above the zero are out of phase with those below the zero by 180° .
- Alternative Phase Opposition Disposition (AOPD-PWM): wherein the adjacent carrier signals are out of phase by 180° .

1.9.2 Phase shifted carrier based PWM (PSCPWM) technique

Phase shifted carrier based pulse width modulation (PSCPWM) technique is commonly used modulation technique for control of cascaded multilevel inverters because of the following reasons:

- Better harmonic profile of output voltage and current wave-forms.
- Even power distribution among levels.
- Easy to implement independently.

These advantages made PSCPWM technique popular compared to LSCPWM technique to control CHB multilevel inverters.

Generally, a multilevel inverter with m -level voltage requires $m - 1$ triangular carriers. All the carriers have same frequency and same peak-to-peak amplitude with phase shift($\phi_{(cr)}$) between adjacent carrier waves and is given by

$$\phi_{(cr)} = \frac{360^\circ}{m - 1}$$

The modulating signal is usually a three-phase sinusoidal wave with adjustable amplitude and frequency. By comparing the modulated wave $V(mA)$ with the carrier waves gate

signals are generated. The fundamental voltage component in the inverter output voltage can be controlled by modulation index M_I . Modulation index is the ratio of maximum voltage value of modulating wave V_m to carrier wave voltage $V_{(cr)}$. The modulation index M_I is usually adjusted by varying V_m by keeping $V_{(cr)}$ fixed.

$$M_I = \frac{V_m}{V_{(cr)}}$$

Harmonics in the case the of high switching frequency modulation techniques appears as side-bands around carrier frequency produces high THD which results in trouble some filtering. For the project case multi-carrier sinusoidal PWM technique is used.

Chapter 2

Flow Chart and Block Diagram

2.1 Flowchart

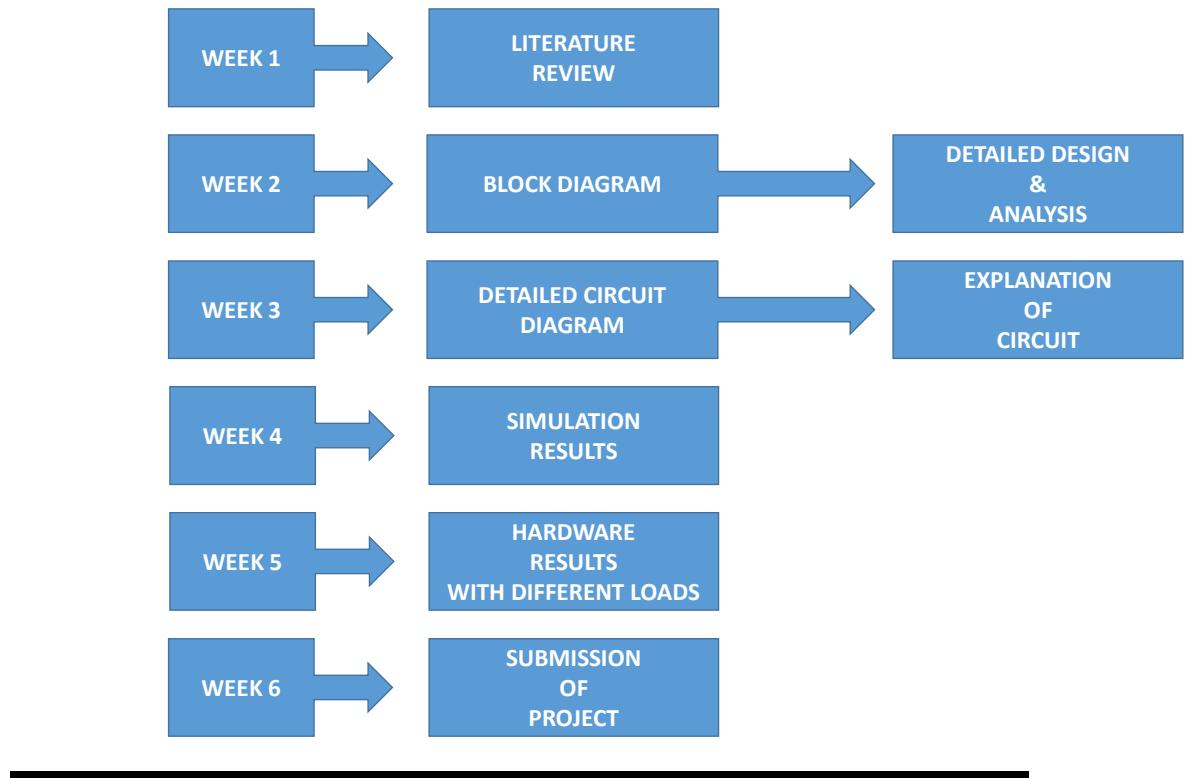


FIGURE 2.1: Flowchart of Project Progress

2.2 Block Diagram

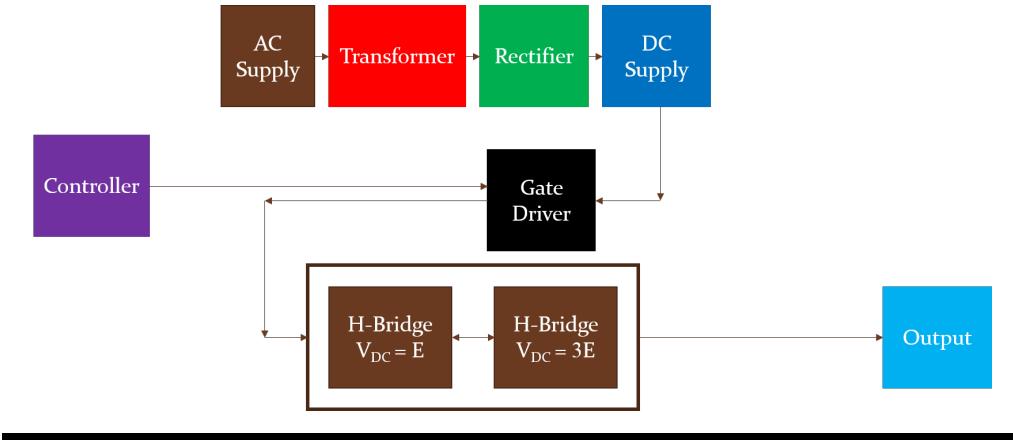


FIGURE 2.2: Block Diagram of Project Progress

Chapter 3

Circuit Design and Analysis

3.1 Terminologies

The average value of output load voltage is V_{dc} . The average value of output load current is I_{dc} . Thus, the output dc power:

$$P_{dc} = V_{dc}I_{dc}$$

The rms value of output load voltage is V_{rms} . The rms value of output load current is I_{rms} . Thus, the output ac power[3]:

$$P_{rms} = V_{rms}I_{rms} \quad (3.1)$$

The efficiency of system is defined as

$$\eta = \frac{P_{out}}{P_{in}} \quad (3.2)$$

The effective rms(ac component) of output voltage can be calculated as:

$$V_{ac} = \sqrt[2]{V_{rms}^2 - V_{dc}^2} \quad (3.3)$$

The Form factor FF, describes the shape of output voltage as:

$$FF = \frac{V_{rms}}{V_{dc}} \quad (3.4)$$

The Ripple factor RF,measures the ripple content of output voltage as:

$$RF = \frac{V_{ac}}{V_{dc}} \quad (3.5)$$

Also,

$$RF = \sqrt[2]{FF^2 - 1} \quad (3.6)$$

Next term is TUF (Transformer Utilization Factor) is defined as:

$$TUF = \frac{P_{dc}}{V_s I_s} \quad (3.7)$$

where V_s and I_s are rms values of source voltage and current respectively.

The Displacement Factor DF is defined as;

$$DF = \cos\phi \quad (3.8)$$

where ϕ is the angle between fundamental component of input current and input voltage. This ϕ is known as Displacement angle.

The Harmonic factor HF of input current is defined as:

$$HF = \sqrt{\left(\frac{I_s}{I_{s1}}\right)^2 - 1} \quad (3.9)$$

where I_{s1} is the fundamental component of input current I_s . Both are expressed in RMS.

The input Power factor PF is defined as:

$$PF = \frac{I_{s1} \cos\phi}{I_s} \quad (3.10)$$

Crest Factor CF which is of interest to specify the peak current ratings of devices and components is defined as:

$$CF = \frac{I_{speak}}{I_s} \quad (3.11)$$

where I_{speak} is peak input current.

Last one is THD Total Harmonic Distortion is measure harmonic distortion present in a signal and is defined as

“Ratio of sum of powers of all harmonic components to power of fundamental frequency.”

$$THD = \frac{\sqrt{\sum_{i=2}^n V_i^2}}{V_1} \quad (3.12)$$

3.2 Design

In general to obtain 9 levels in voltage waveform we need 4 cells of Independent H-Bridge. But by using two independent sources of $VDC1 = E$ and $VDC2 = 3E$ we can obtain 9 levels in output voltage. In circuit 4.1 we have:

$$VDC1 = E$$

$$VDC2 = 3E$$

By applying specific sequence on switches s1, s2, s3, s4, s5, s6, s7 and s8 we can obtain voltage waveform with 9-levels. We only need to design sequences for s1, s3, s5 and s7, because we can apply inverted signal on other switches s4, s2, s8 and s6 respectively.

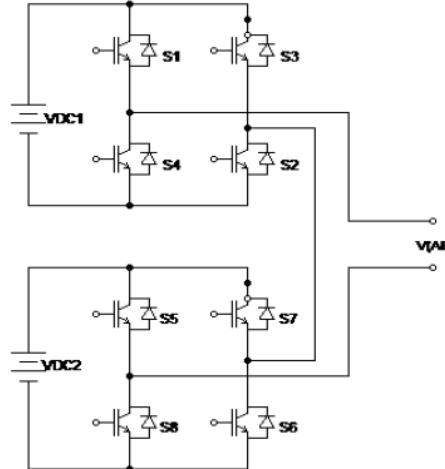


FIGURE 3.1: H Bridge circuit for 9 Levels Voltage wave-forms

Output	S1	S3	S5	S7	V1	V2
4E	1	0	1	0	E	3E
3E	1	1	1	0	0	3E
	0	0	1	0	0	3E
2E	0	1	1	0	-E	3E
E	1	0	0	0	E	0
	1	0	1	1	E	0
0	1	1	1	1	0	0
	0	0	0	0	0	0
	1	1	0	0	0	0
-E	0	0	1	1	0	0
	0	1	0	0	-E	0
-2E	1	0	0	1	E	-3E
-3E	0	0	0	1	0	-3E
	1	1	0	1	0	-3E
-4E	0	1	0	1	-E	-3E

FIGURE 3.2: Switching States for 9 Levels Voltage wave-forms

By using switching states as shown in 4.3 we can define switching sequence. The switching sequence is obtained by selecting switching states 4.2 so that minimum transitions

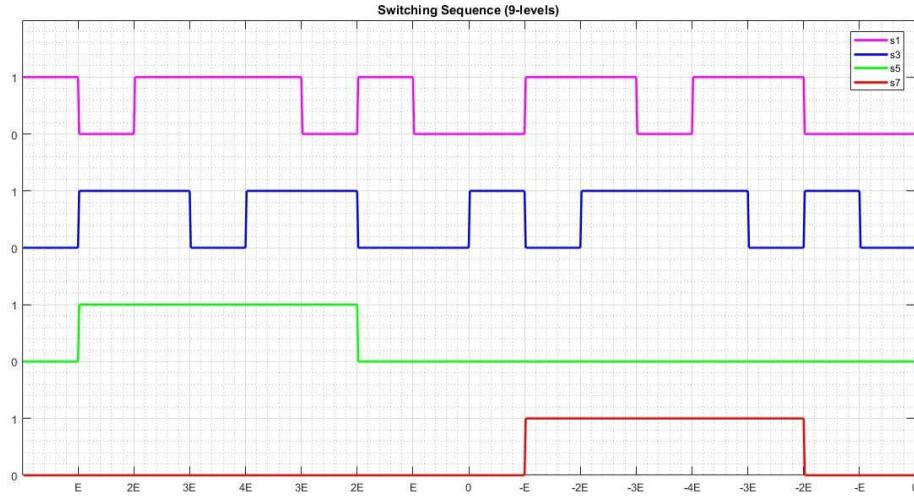


FIGURE 3.3: Graphical Representation of Switching Sequence with Minimum Transitions

occurs. By applying these sequences to respective switches s1, s3, s5 and s7 and inverted sequences on s4, s2, s8 and s6 respectively we get the output voltage waveform^{4.8}. To control output power we can use PWM techniques like Sinusoidal PWM as discussed before. To control harmonics in output waveform we can change conduction angles.

3.3 Analysis

Lower frequency harmonics like 3, 5, 7, 9.. can be removed by choosing conduction angles wisely. Higher frequency harmonics can be removed by filtering the output using Low Pass Filter.

The general Fourier series expression for ‘s’ steps in voltage waveform is given below:
For $\theta_1 = 14.8^\circ, \theta_2 = 30^\circ, \theta_3 = 48^\circ$ and $\theta_4 = 68^\circ$ taken by hit and trial. The output^{4.8}

$$V(\omega t) = \frac{4V_{dc}}{\pi} \sum_n [\cos(n\theta_1) + \cos(n\theta_2) + \dots + \cos(n\theta_s)] \times \frac{\sin(n\omega t)}{n}, \quad \text{where } n = 1, 3, 5, 7, \dots$$

of 9 levels CHBMLI and its magnitude spectrum^{3.5} is shown below. We can see form magnitude spectrum of output voltage that

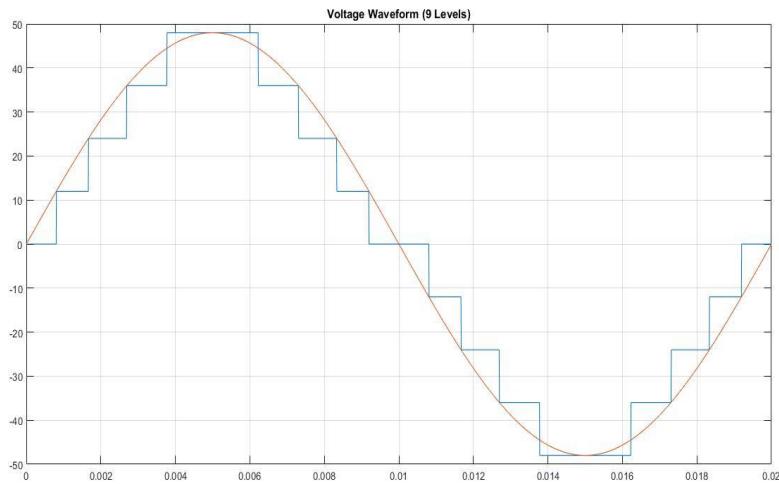


FIGURE 3.4: 9-Levels Voltage waveform when E=12 V

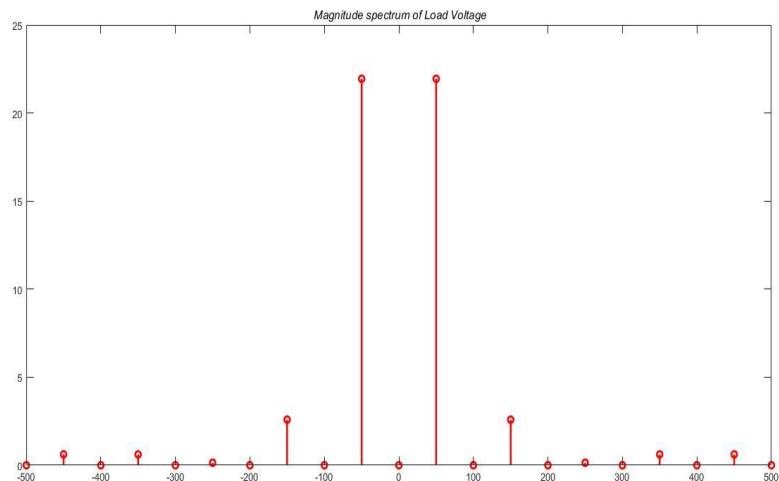


FIGURE 3.5: 9-Levels Voltage Magnitude Spectrum

- Fundamental component ($f = 50\text{Hz}$) has maximum contribution.
- Even harmonics have zero contribution.
- Odd harmonics are not zero, and 3rd harmonic ($f = 150\text{Hz}$) has maximum contribution after fundamental.

Contribution of fundamental component is:

$$V_1 = 31.023V$$

also,

$$V_1 = 0.986V_o$$

where,

$$V_o = \left[\frac{4}{360} \left(\int_{14.8}^{30} E^2 dt + \int_{30}^{48} 4E^2 dt + \int_{48}^{68} 9E^2 dt + \int_{68}^{90} E^2 dt \right) \right]^{0.5}$$

$$V_o = 2.623E$$

$$V_o = 31.47V$$

Contribution of 3rd harmonic is:

$$V_3 = 3.6566V$$

$$V_3 = 0.116V_o$$

Contribution of 5th harmonic is:

$$V_5 = 0.1999V$$

$$V_5 = 0.006V_o$$

Contribution of 7th harmonic is:

$$V_7 = 0.866V$$

$$V_7 = 0.027V_o$$

Contribution of 9th harmonic is:

$$V_9 = 0.871V$$

$$V_9 = 0.027V_o$$

Thus,

$$THD = \frac{(V_0^2 - V_1^2)^{0.5}}{V_1}$$

$$THD = \frac{0.166V_o}{0.986V_o}$$

$$THD = 16.9\%$$

We can change the conduction angles for better approximation of output waveform towards a sinusoidal waveform as shown fig3.6 and fig3.7 for $\theta_1 = 6^\circ, \theta_2 = 22^\circ, \theta_3 = 38^\circ$ and $\theta_4 = 60^\circ$ taken by hit and trial.

$$V_o = \left[\frac{4}{360} \left(\int_6^{22} E^2 dt + \int_{22}^{38} 4E^2 dt + \int_{38}^{60} 9E^2 dt + \int_{60}^{90} E^2 dt \right) \right]^{0.5}$$

$$V_o = 2.901E$$

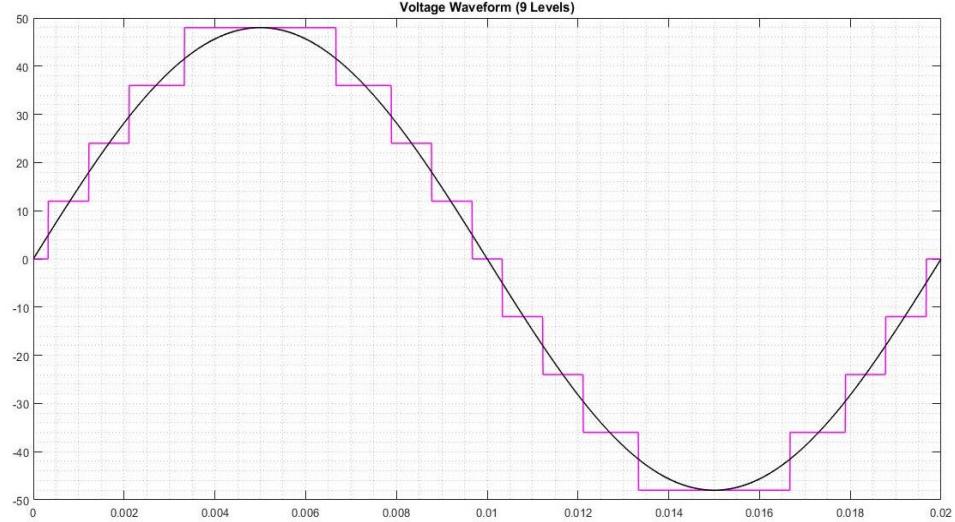


FIGURE 3.6: Better Approximated 9 levels Voltage waveform

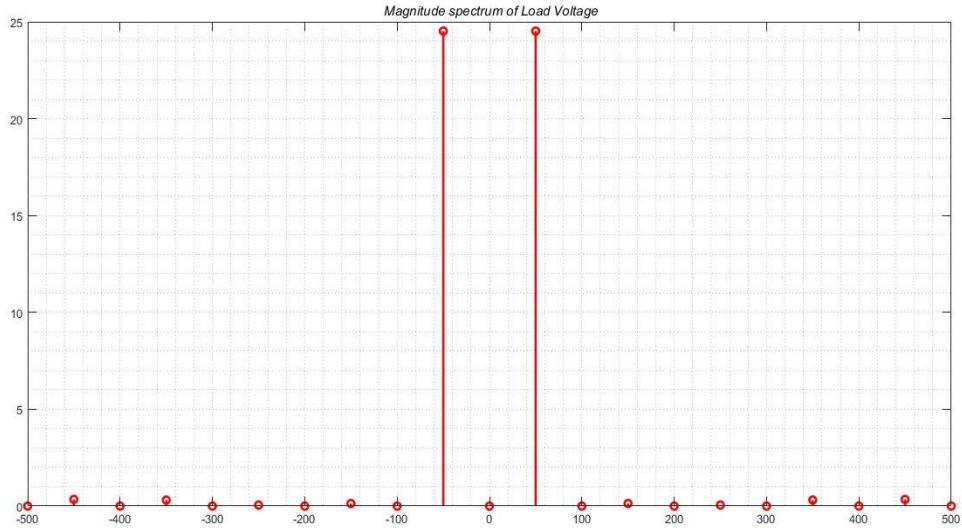


FIGURE 3.7: Improved Fourier Spectrum

$$V_o = 34.82V$$

Contribution of fundamental component is:

$$V_1 = 34.68V$$

$$V_1 = 0.996V_o$$

Thus,

$$THD = \frac{(V_o^2 - V_1^2)^{0.5}}{V_1}$$

$$THD = \frac{0.0893V_o}{0.996V_o}$$

$$THD = 8.97\%$$

Ratings		
Component	Voltage Rating	Current Rating
Isolated Supplies for Gate Driver	15V	0.3A
TLP250 Input (Gate Driver)	3.3V	20mA
TLP250 Output (Gate Driver)	35V	1.5A
IRF450 (Switch)	600V	12A
STM32F4 (GPIO Pins)	3.3V	20mA

Chapter 4

Detailed Circuit Design

The complete circuit diagram of the circuit is shown [4.1](#).

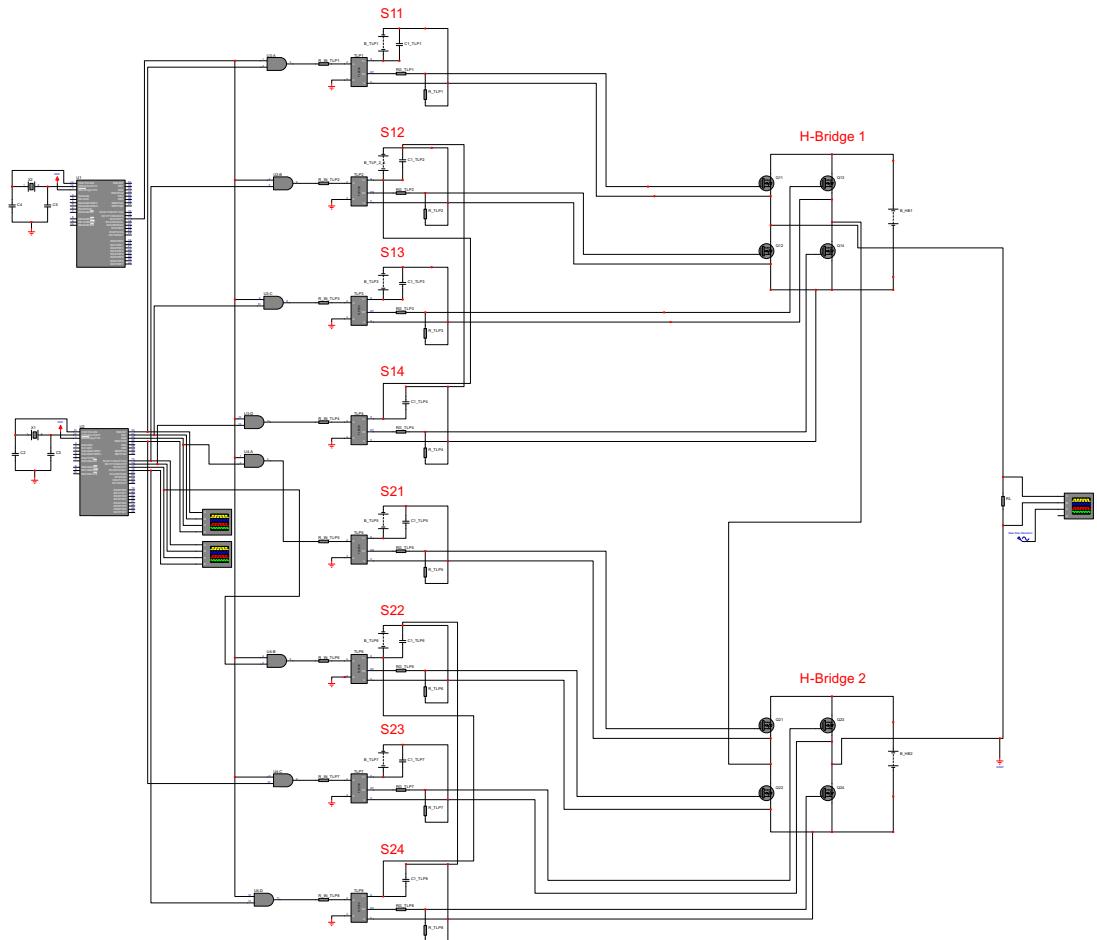


FIGURE 4.1: Circuit for 9 Levels Voltage wave-forms

The circuit^{4.1} is divided into 4 main portions as:

- DC Supplies.
- STM32F4x Micro-controller to apply switching.
- Gate Triggering Circuit using Opto-coupler TLP250.
- 2 Cascaded H-Bridge Inverter.

4.1 DC Supplies

In this circuit total 8 DC supplies are used to power the circuit, 6(Isolated) out of them have 15 Volts output to power the TLP 250 present in Gate triggering circuit and the other two having outputs ratio of 1:3 volts are used to get 9 levels output voltage waveform. Gate Drivers for ground connected Mosfets have given same supply.

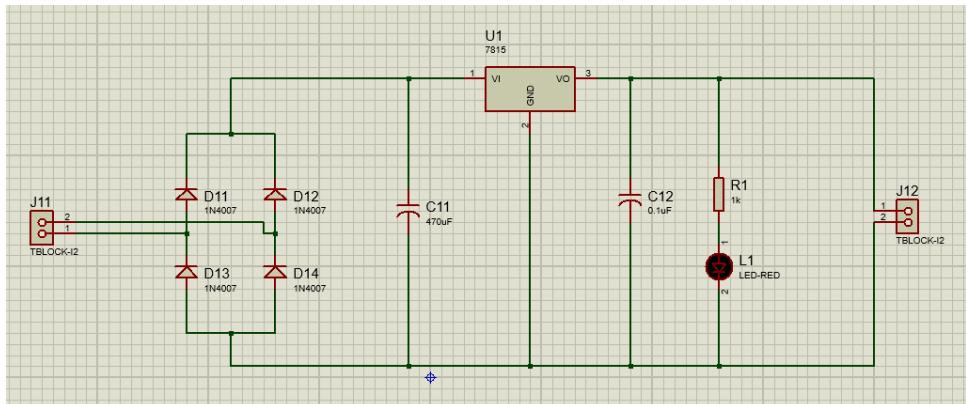


FIGURE 4.2: Circuit for Constant DC supply

4.2 STM32F4x Microntroller

It is a 32-bit ARM Cortex -M4 with FPU core, 1-Mbyte Flash memory and 192-Kbyte RAM. It has 6 GPIO ports and 82 I/O pins with operating voltage of 1.62-3.6v and current rating upto 25mA. It includes 4 user LEDs and 1 user switch. The board can be power up through USB, and it can supply 3v and 5v as external power supply. This is used in this project to generate pulses at frequency of 50Hz as input of TLP250.

4.3 Gate triggering circuit

TLP250 is 8 pin opto-coupler ic used for isolation and safety for Microcontroller also used as gate driver of switching device because of its opto coupling ability and ability to give discharging path.

As seen from the figure^{4.8} LED is connected with 2nd and 3rd pin and two BJTs are connected in series, top one to give output as V_E and lower one to give discharging

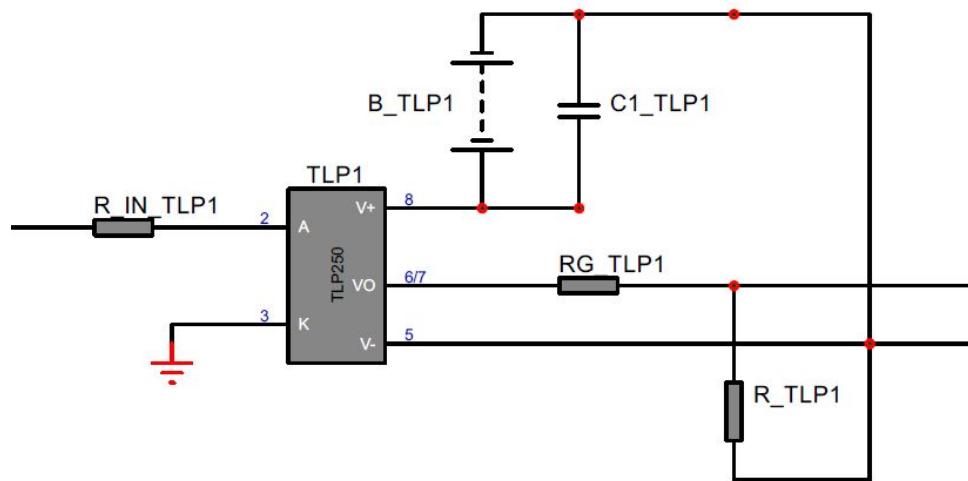


FIGURE 4.3: Circuit for TLP250

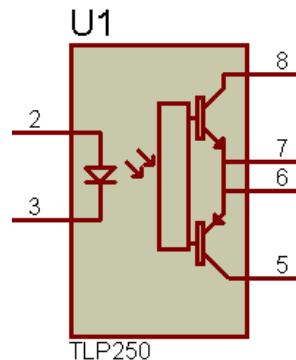


FIGURE 4.4: Inner circuit for TLP250

path. $R_i = 1k\text{ohm}$ and $R = 10\text{ohm}$. Capacitor is of 1nF 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. The pins of TLP 250 are labelled as:

- PIN: 2 = ANODE
- PIN: 3 = CATHODE
- PIN: 5 = V-/GROUND
- PIN: 6 and 7=OUTPUT
- PIN: 8 = V+/INPUT(15V)
- PIN: 1 and 4 = NC

4.4 H-Bridge Circuits

There are two H-Bridge circuits used in this project[5]. By adding two H-Bridges in series so that the upper H-Bridge has DC Source = E volts and lower H-Bridge has DC Source = $3E$ volts, and by applying a specific switching sequence to all switches we can obtain 9 levels in output waveform. Below the figures of all the parts are shown.

H-Bridge 1

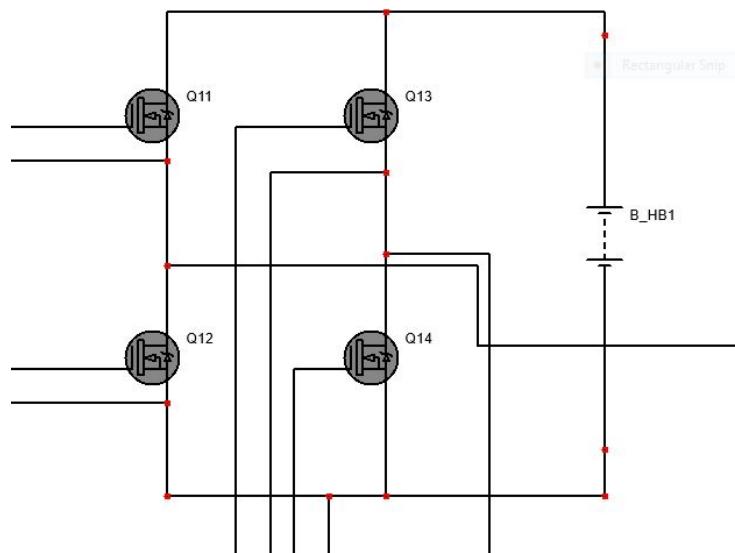


FIGURE 4.5: H-Bridge 1 With DC Source = E volts

H-Bridge 2

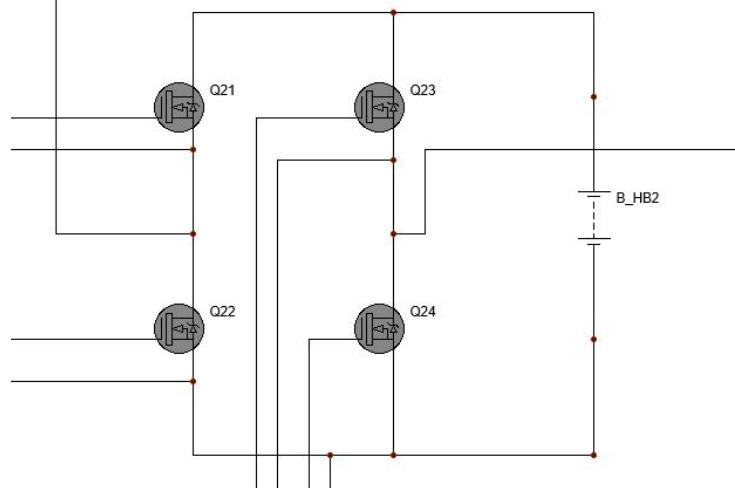


FIGURE 4.6: H-Bridge 2 With DC Source = $3E$ volts

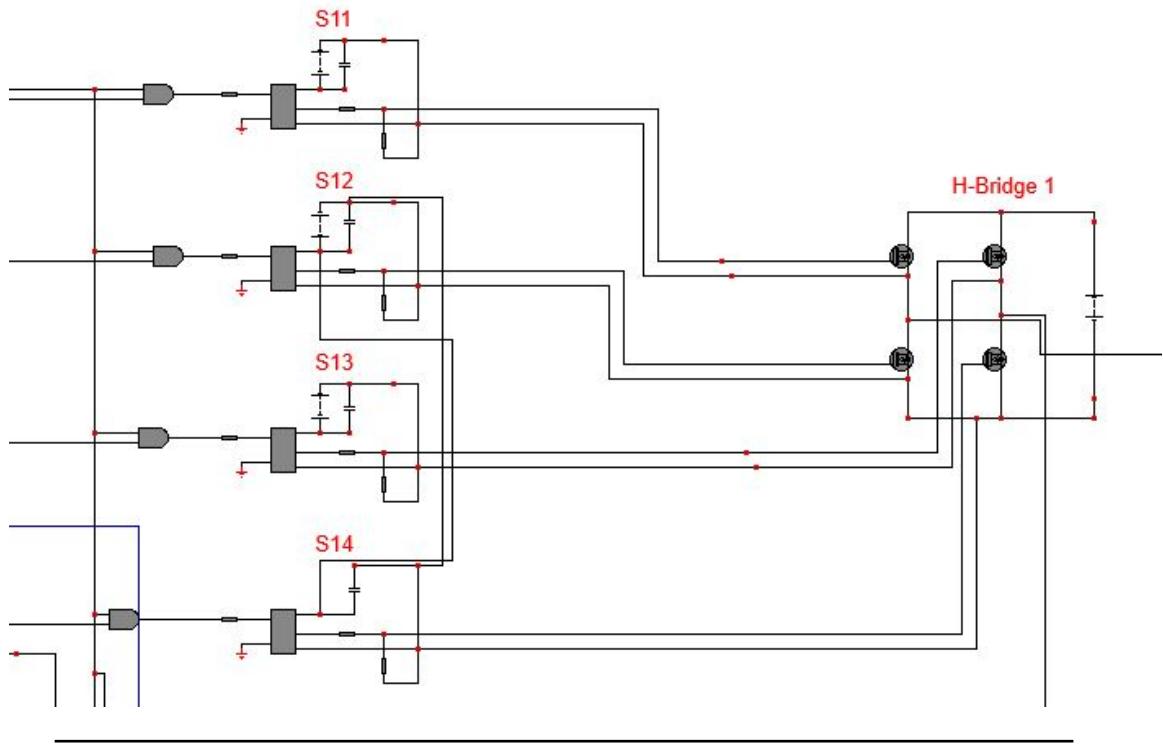


FIGURE 4.7: H-Bridge 1 With Gate Driver Circuit

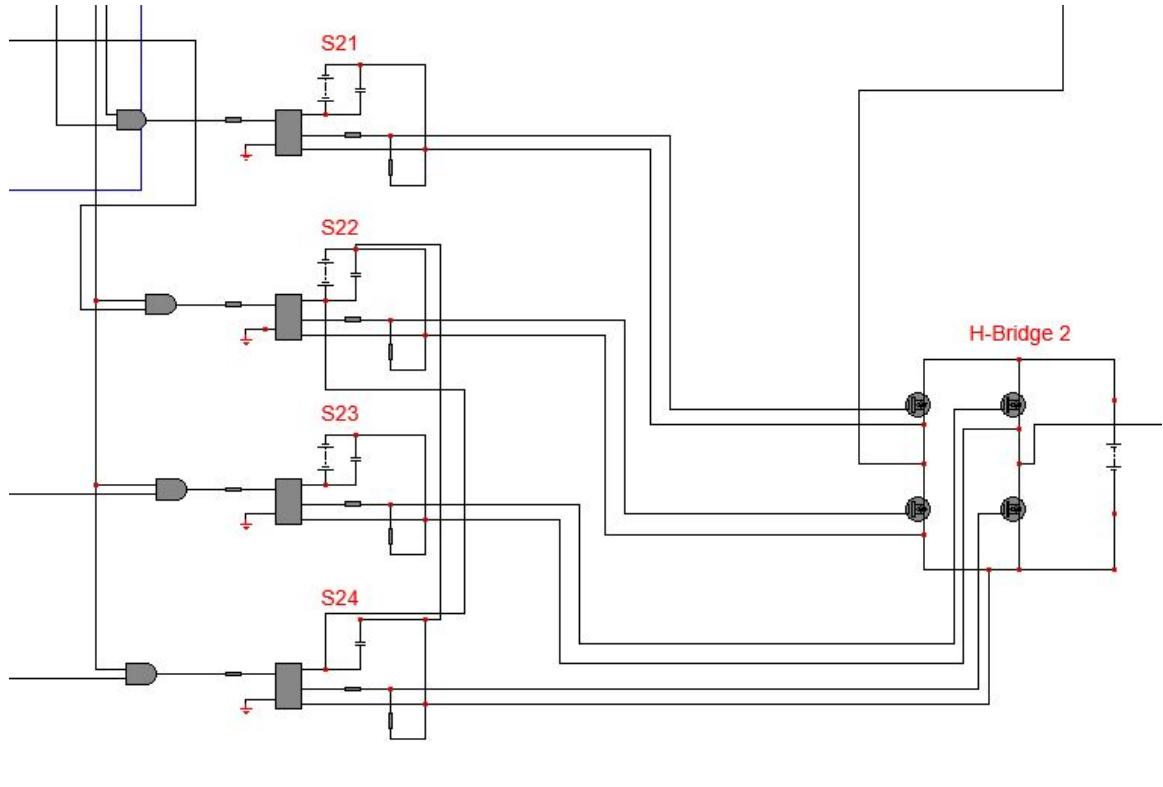


FIGURE 4.8: H-Bridge 2 With Gate Driver Circuit

Chapter 5

Simulation Results

5.1 Gate Signals

The Gating Signals applied to both H bridges in order to get 9 levels in output voltage waveform is shown below. Because the Input DC Source was unequal so we have to define specific sequences for all switches. The Conduction Angles are listed here:

Conduction Angles	
Angle	Value in Degrees
α_1	6°
α_2	22°
α_3	38°
α_4	60°

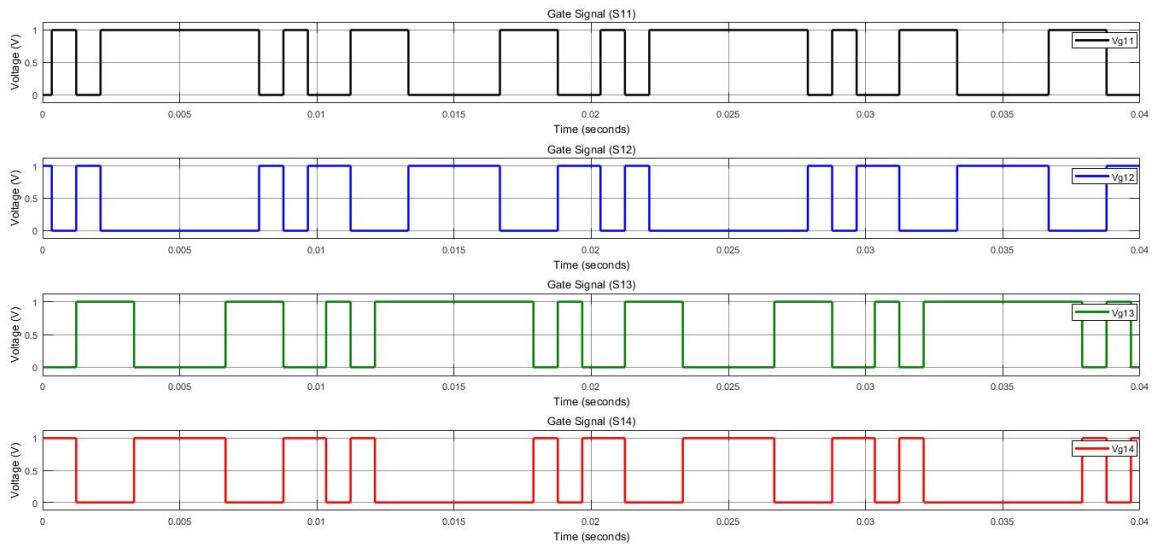


FIGURE 5.1: Gate Signals to H-Bridge 1

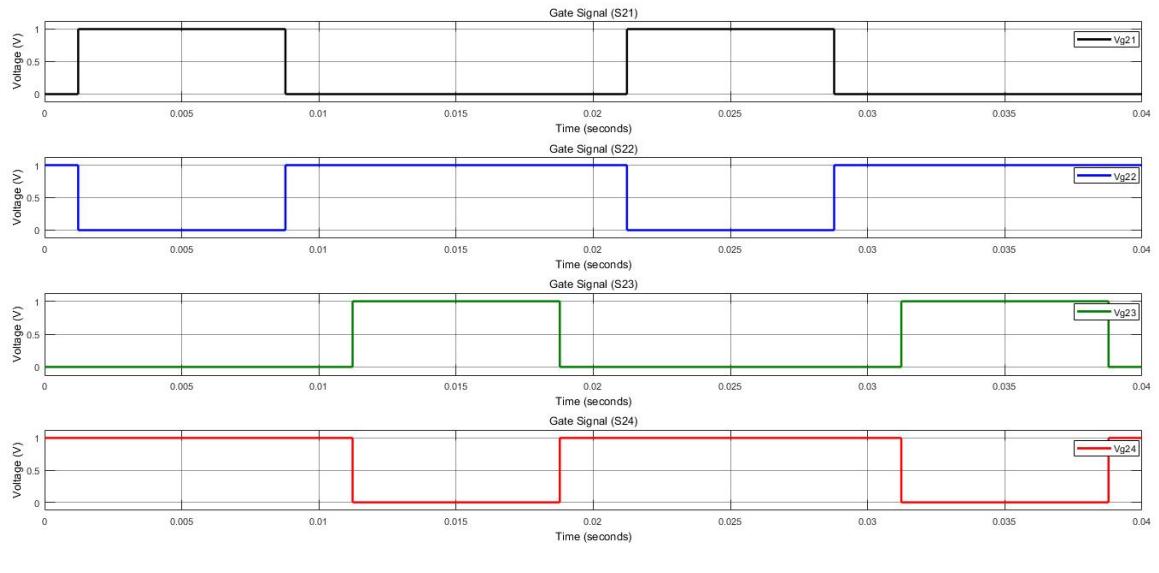


FIGURE 5.2: Gate Signals to H-Bridge 2

5.2 Source Currents

Below is the display of source currents when Resistive load is used:

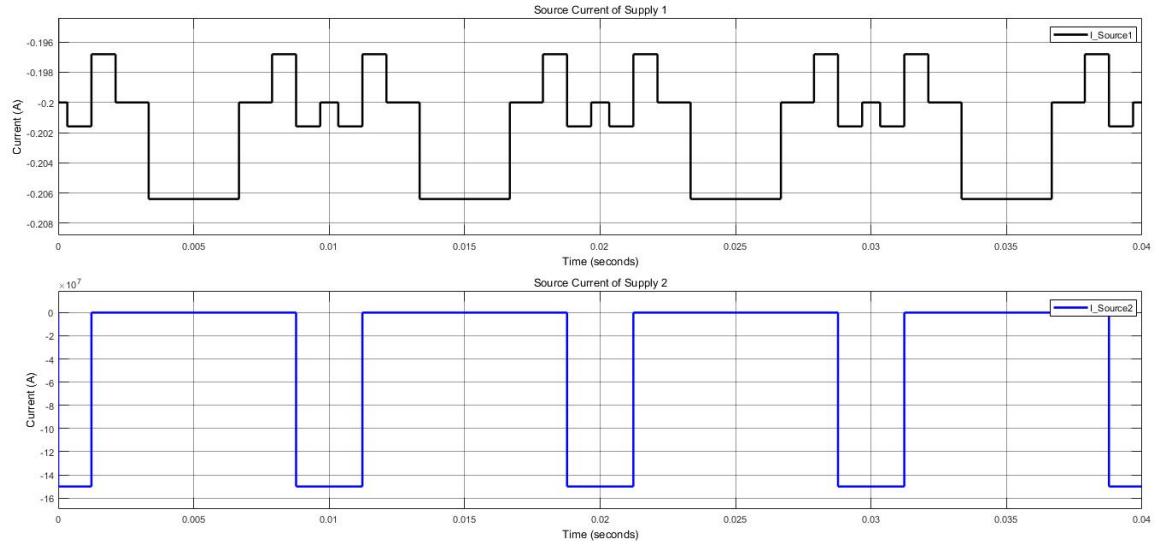


FIGURE 5.3: Source Currents (Resistive Load)

Below is the display of source currents when Resistive plus Inductive load is used at 0.8 power Factor:

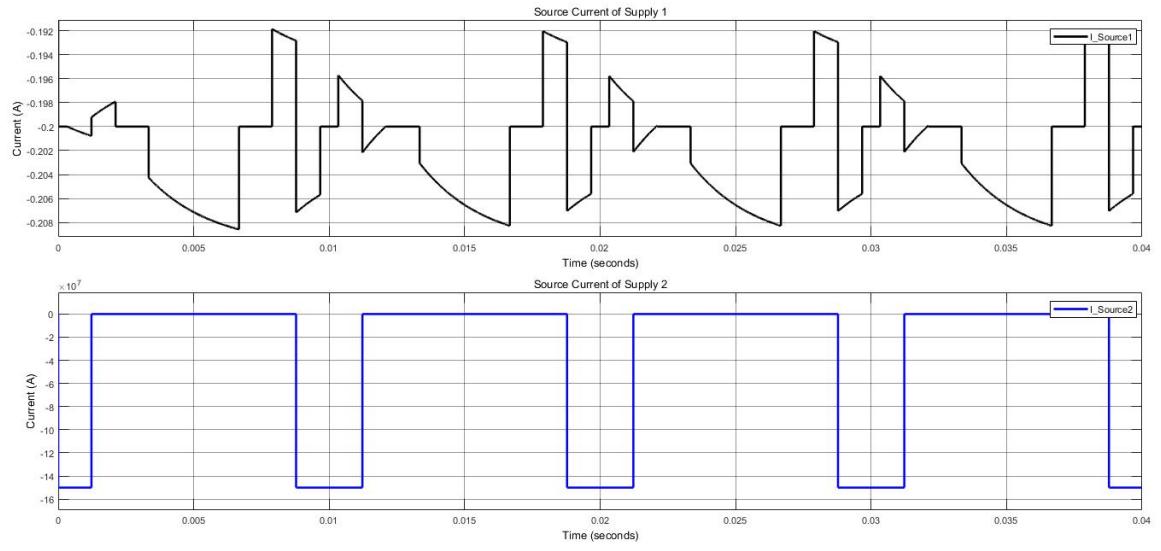


FIGURE 5.4: Source Currents (Inductive Load 0.8PF)

5.3 Switch Currents

Below is the display of switch currents when Resistive load is used:

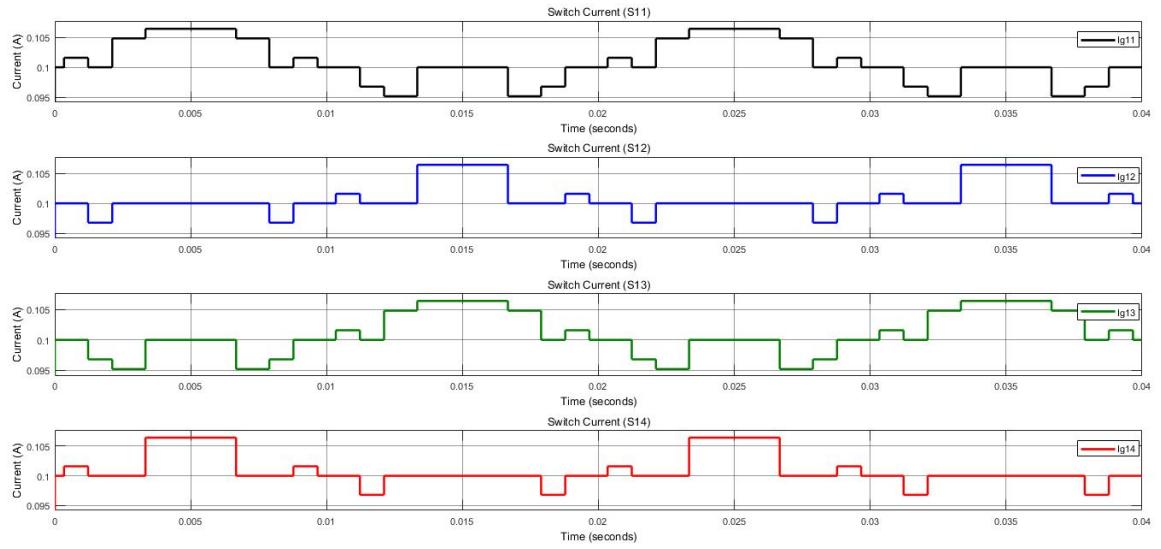


FIGURE 5.5: Switch Currents of H-Bridge 1 (Resistive Load)

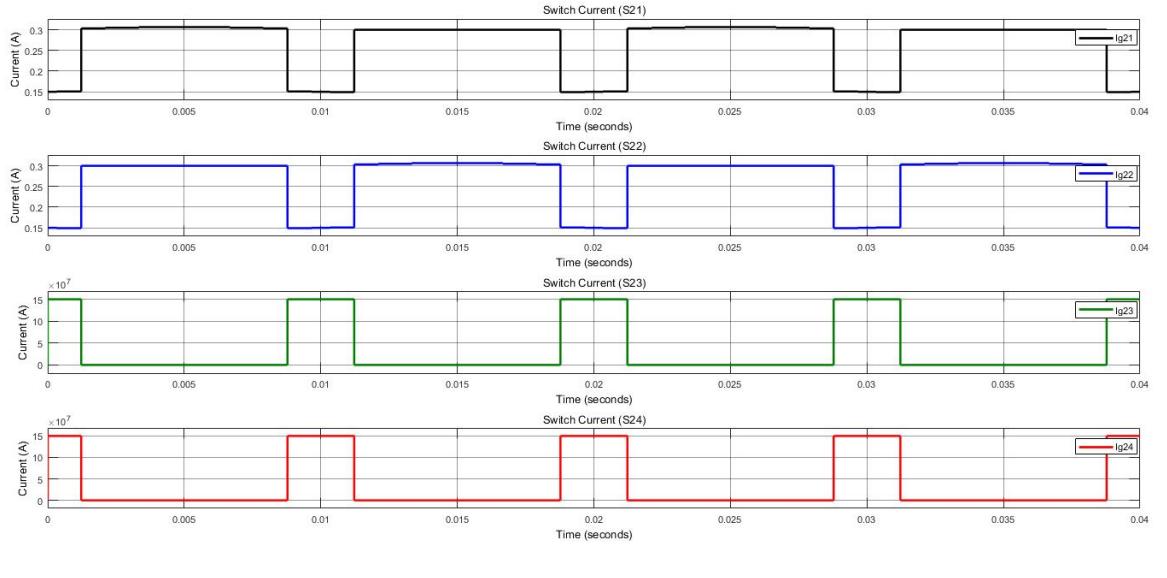


FIGURE 5.6: Switch Currents of H-Bridge 2 (Resistive Load)

Below is the display of switch currents when Resistive plus Inductive load is used at 0.8 power factor:

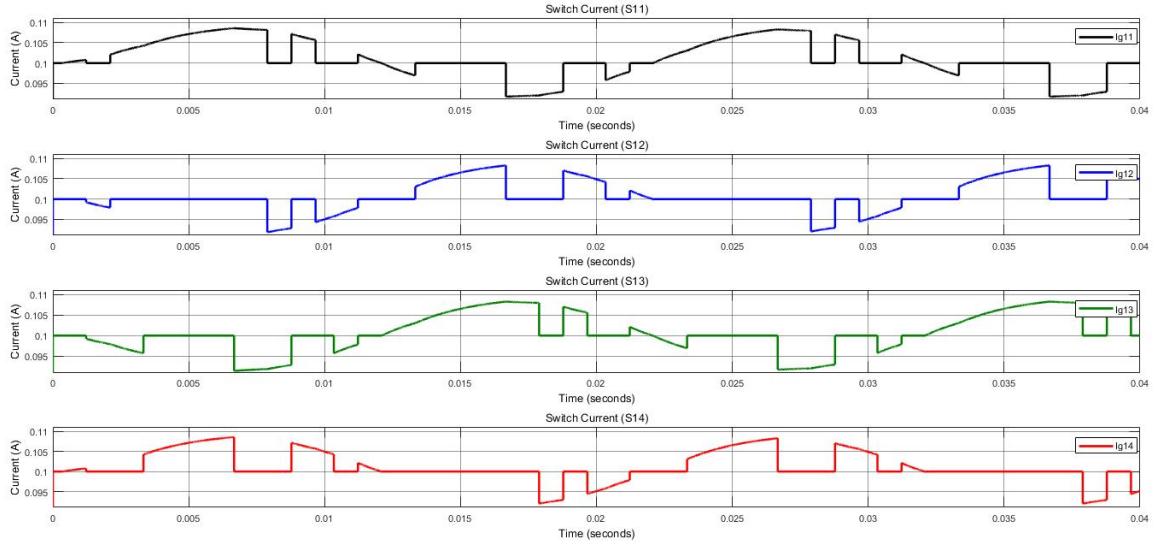


FIGURE 5.7: Switch Currents of H-Bridge 1 (Inductive Load 0.8PF)

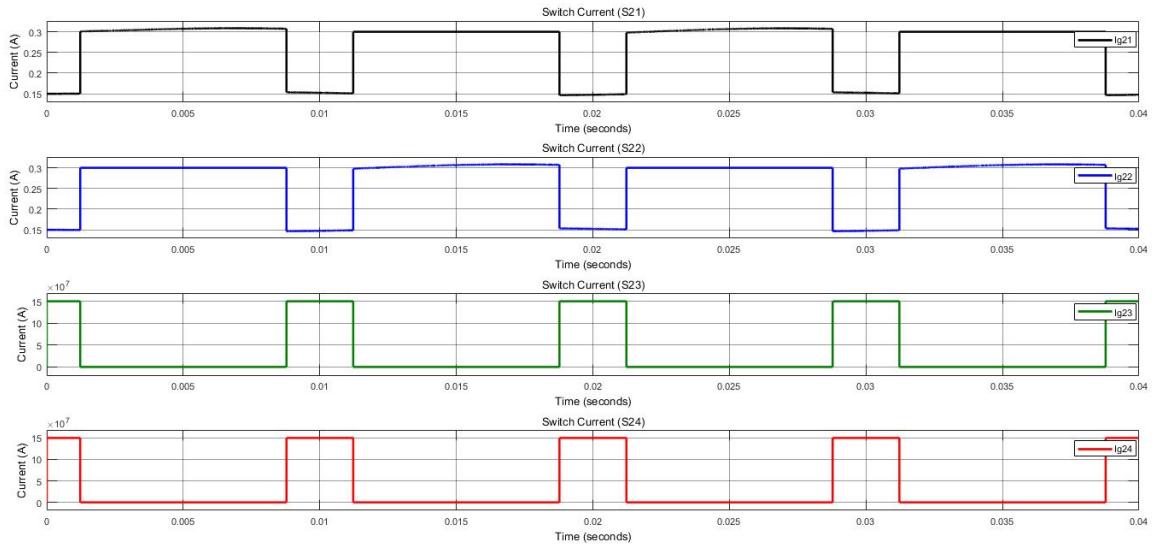


FIGURE 5.8: Switch Currents of H-Bridge 2 (Inductive Load 0.8PF)

5.4 Load Current

Load Current waveform when resistive load is used: Below is the display of load current

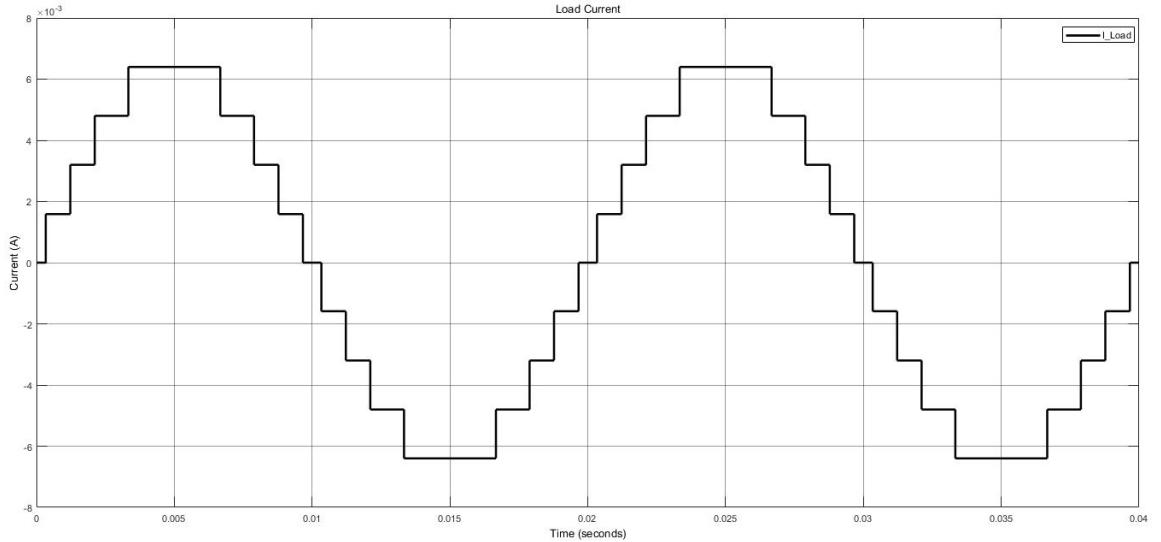


FIGURE 5.9: Load Current (Resistive Load)

when Resistive plus Inductive load is used at 0.8 power factor:

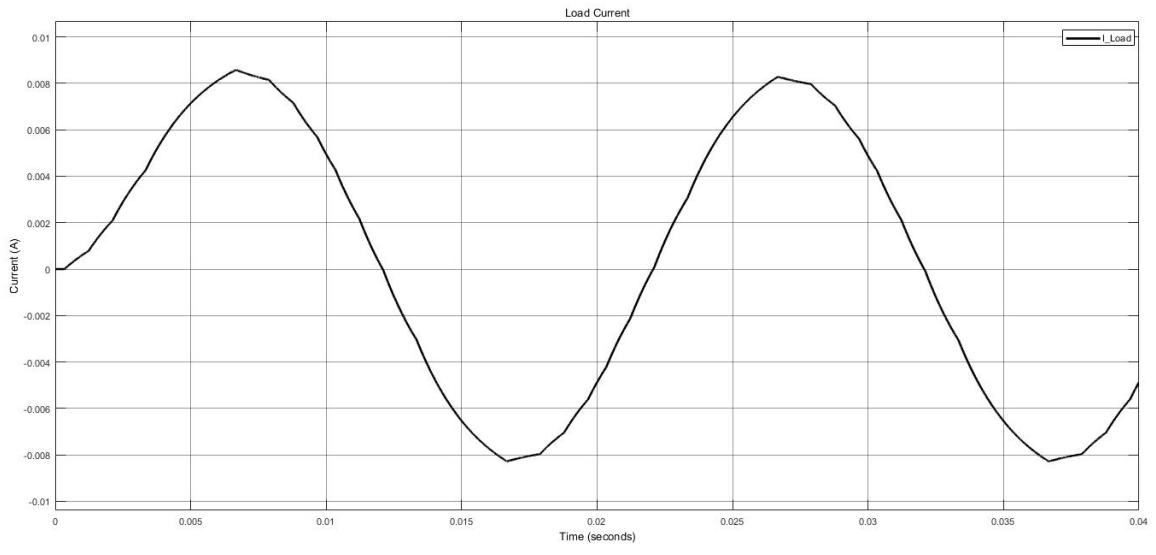


FIGURE 5.10: Load Current (Inductive Load 0.8PF)

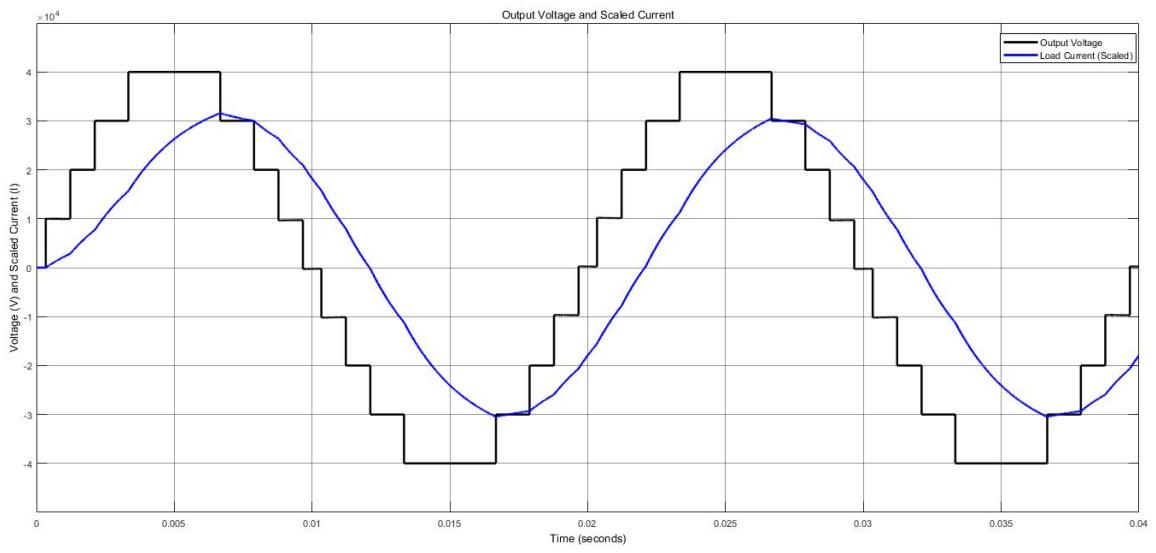


FIGURE 5.11: Load Voltage and Scaled Current (Inductive Load 0.8PF)

5.5 Output Voltage

The output voltage and its FFT analysis is shown below:

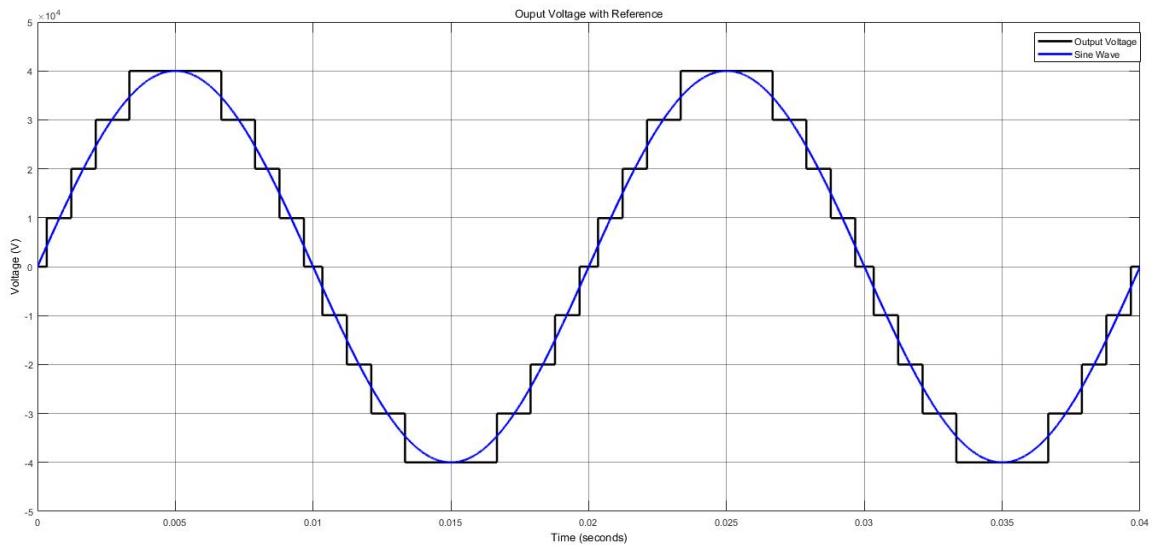


FIGURE 5.12: Output Voltage

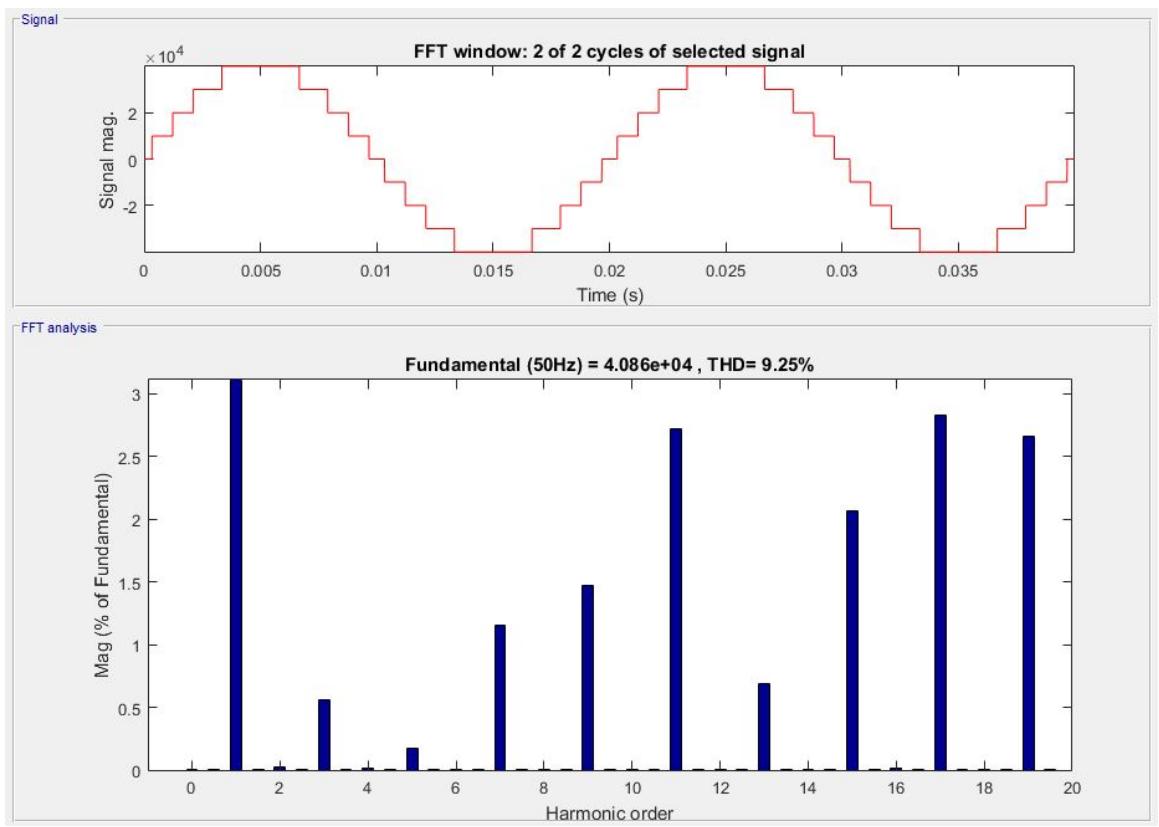


FIGURE 5.13: Output Voltage FFT Analysis

THD = 9.25%			
0 Hz (DC) :	0.00%	90.0°	
25 Hz	0.01%	193.9°	
50 Hz (Fnd) :	100.00%	-0.0°	
75 Hz	0.01%	154.8°	
100 Hz (h2) :	0.02%	188.6°	
125 Hz	0.01%	193.2°	
150 Hz (h3) :	0.56%	179.1°	
175 Hz	0.01%	189.5°	
200 Hz (h4) :	0.01%	195.8°	
225 Hz	0.00%	34.8°	
250 Hz (h5) :	0.18%	-1.4°	
275 Hz	0.01%	190.5°	
300 Hz (h6) :	0.01%	79.2°	
325 Hz	0.00%	-23.6°	
350 Hz (h7) :	1.16%	-0.4°	
375 Hz	0.01%	157.2°	
400 Hz (h8) :	0.01%	-28.4°	
425 Hz	0.00%	-61.0°	
450 Hz (h9) :	1.47%	179.4°	
475 Hz	0.00%	240.3°	
500 Hz (h10) :	0.01%	-20.4°	
525 Hz	0.00%	59.6°	

FIGURE 5.14: Record of First 10 Harmonics

5.6 Output Generated by PWM Technique

The output voltage generated by programming based PWM technique is shown below, the switch frequency is 10kHz and duty is 50 percent.

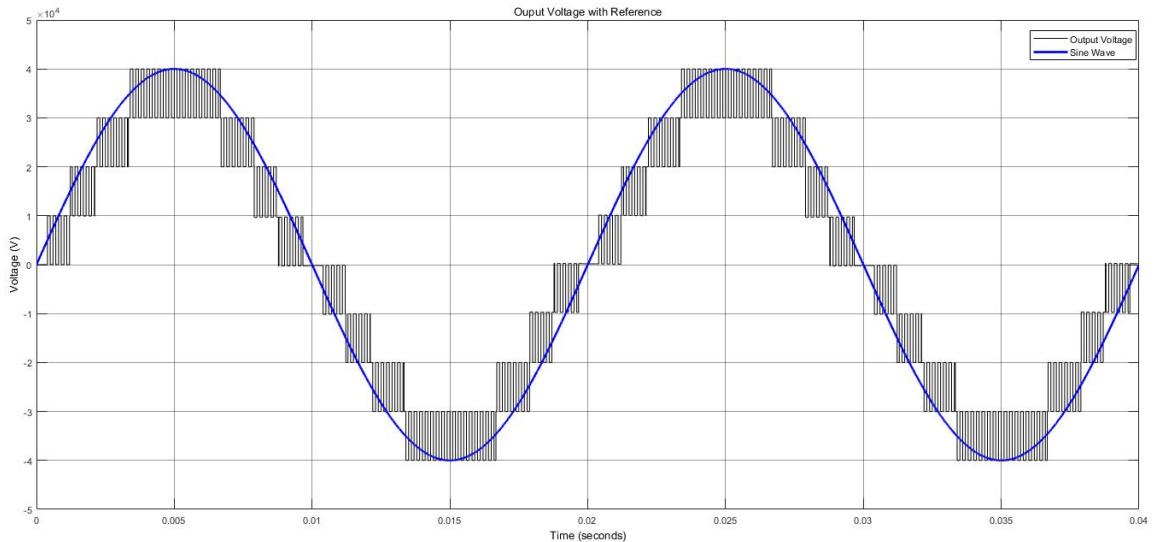


FIGURE 5.15: Output Signal Generated by PWM Technique

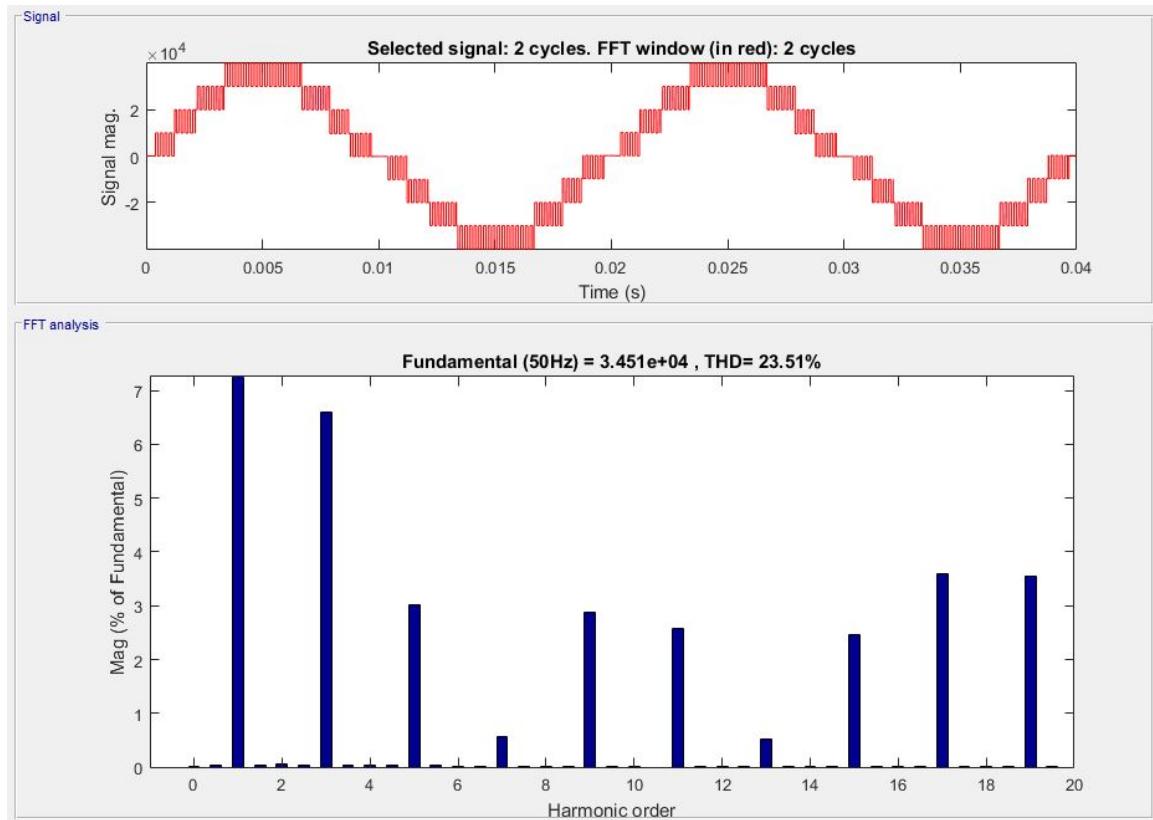


FIGURE 5.16: FFT of Output Signal Generated by PWM Technique

5.7 Output Generated by Sine PWM Technique

The output voltage generated by programming based PWM technique is shown below, the switch frequency is 10kHz, m_f is 200 and m_a is 0.8.

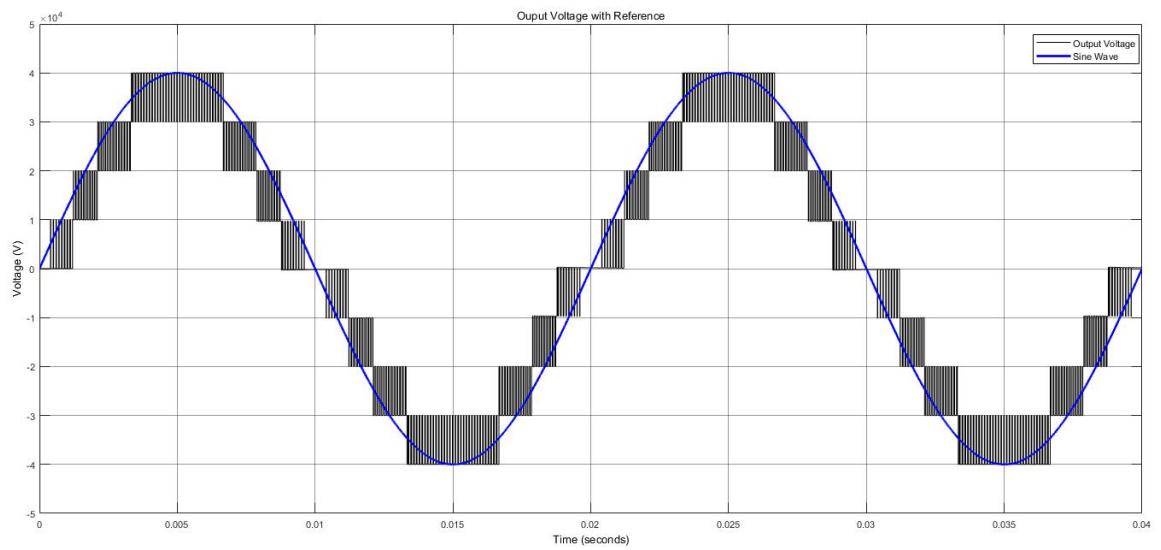


FIGURE 5.17: Output Signal Generated by Sine PWM Technique

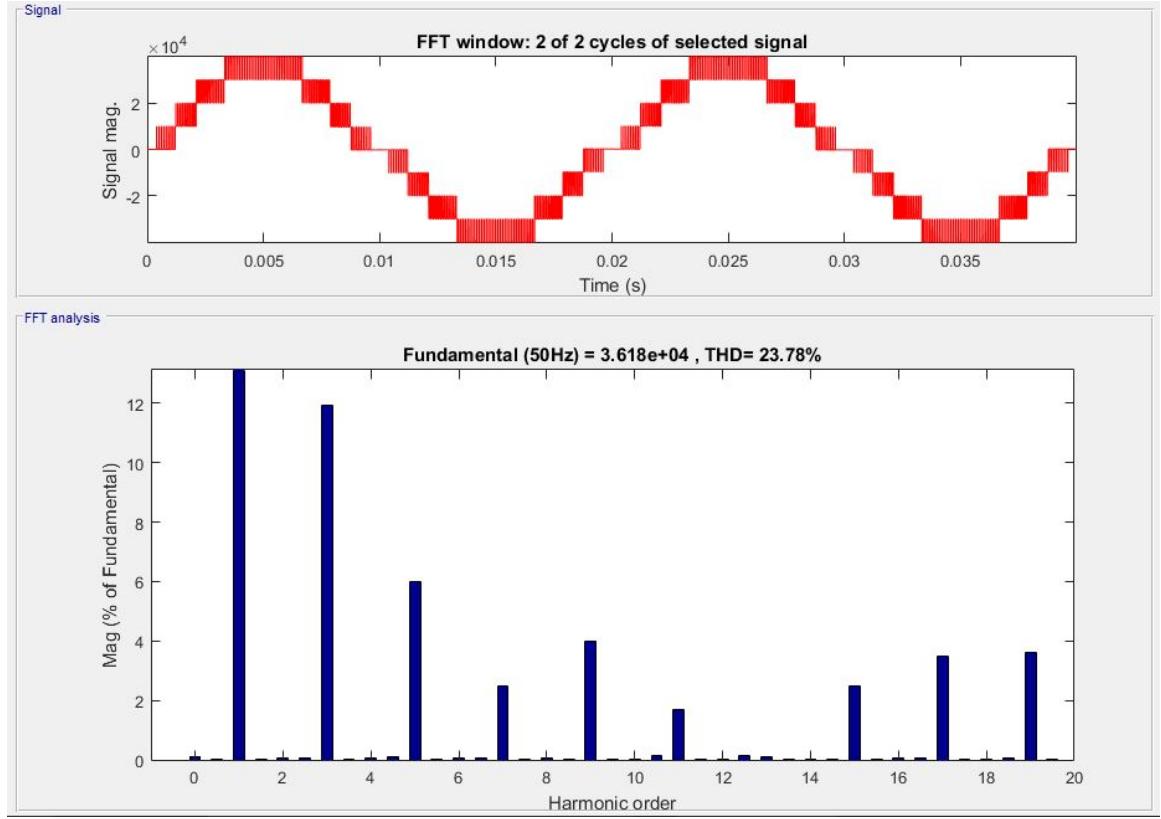


FIGURE 5.18: FFT of Output Signal Generated by Sine PWM Technique

5.8 AND Operation with PWM Signal

A PWM Signal of 10kHz frequency is generated and AND operation is performed with all gating signals, the output is shown below:

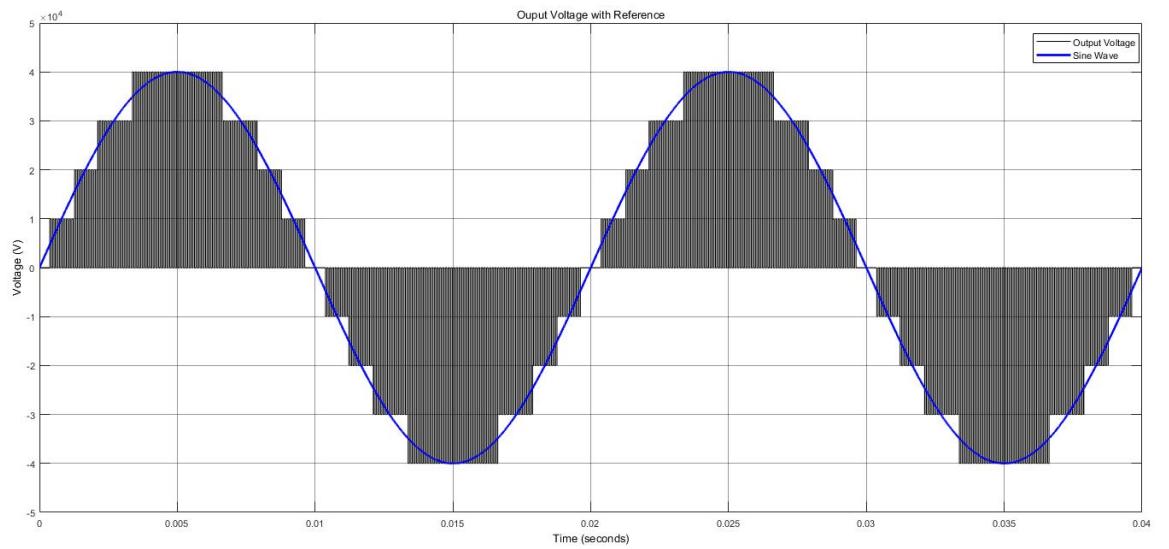


FIGURE 5.19: Output Signal after AND Operation

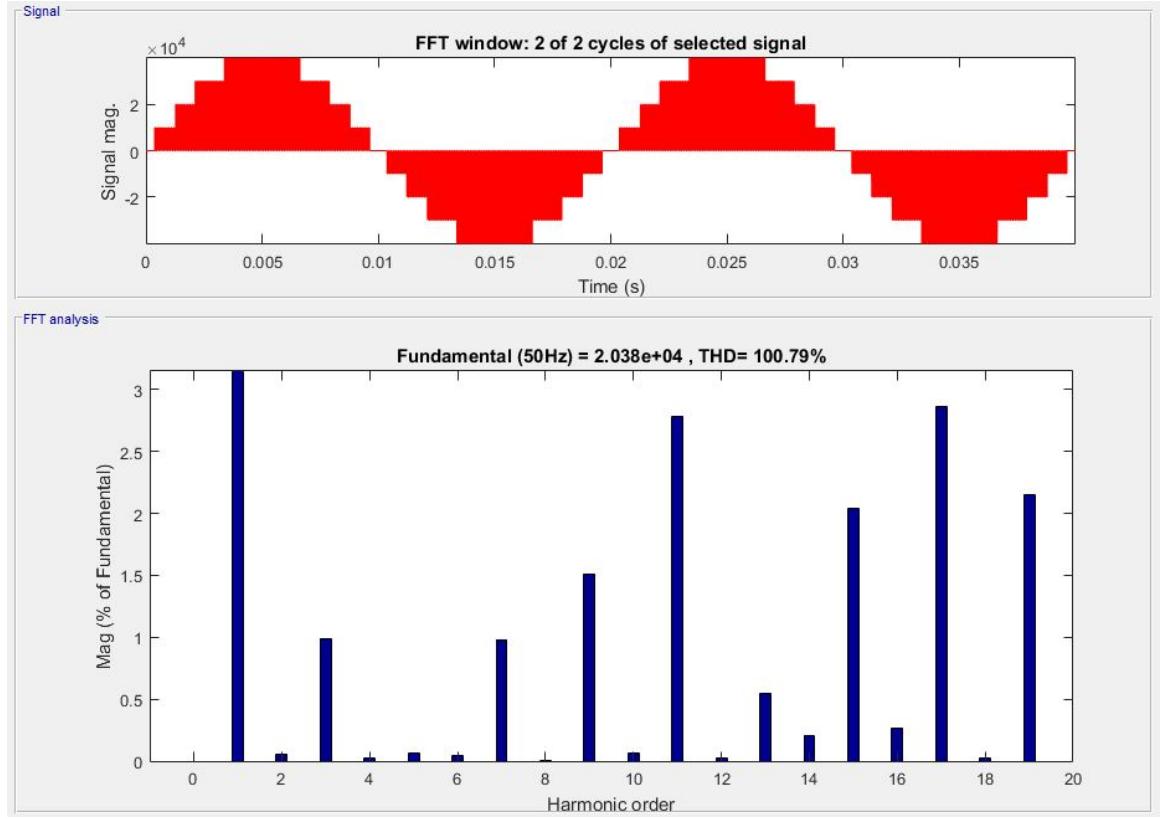


FIGURE 5.20: FFT of Output Signal after AND Operation

5.9 AND Operation with Sine PWM Signal

A Sine PWM Signal of 200 m_f and 0.8 m_a is generated and AND operation is performed with all gating signals, the output is shown below:

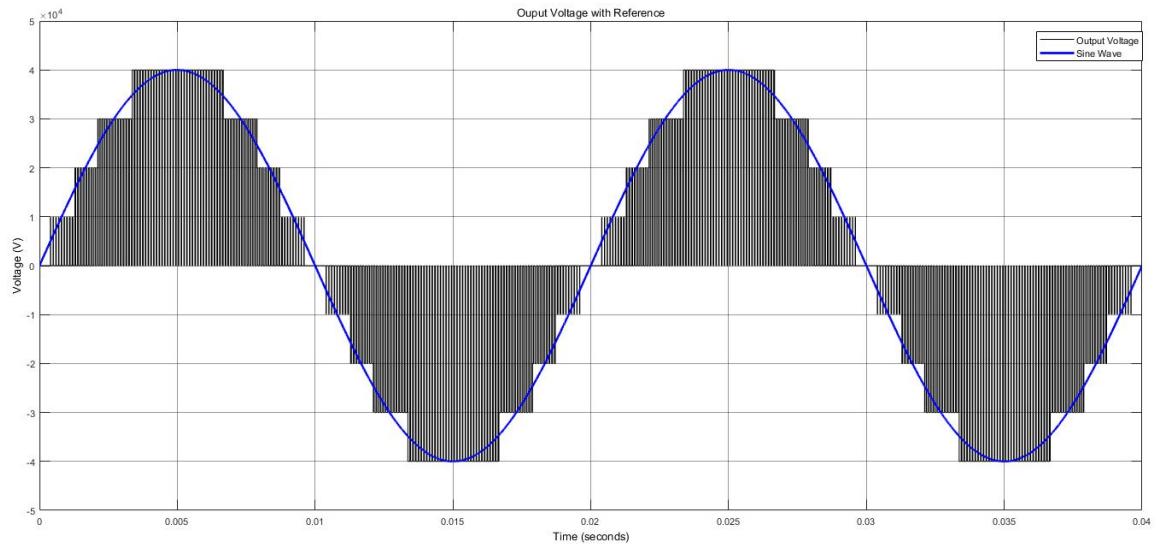


FIGURE 5.21: Output Signal after AND Operation

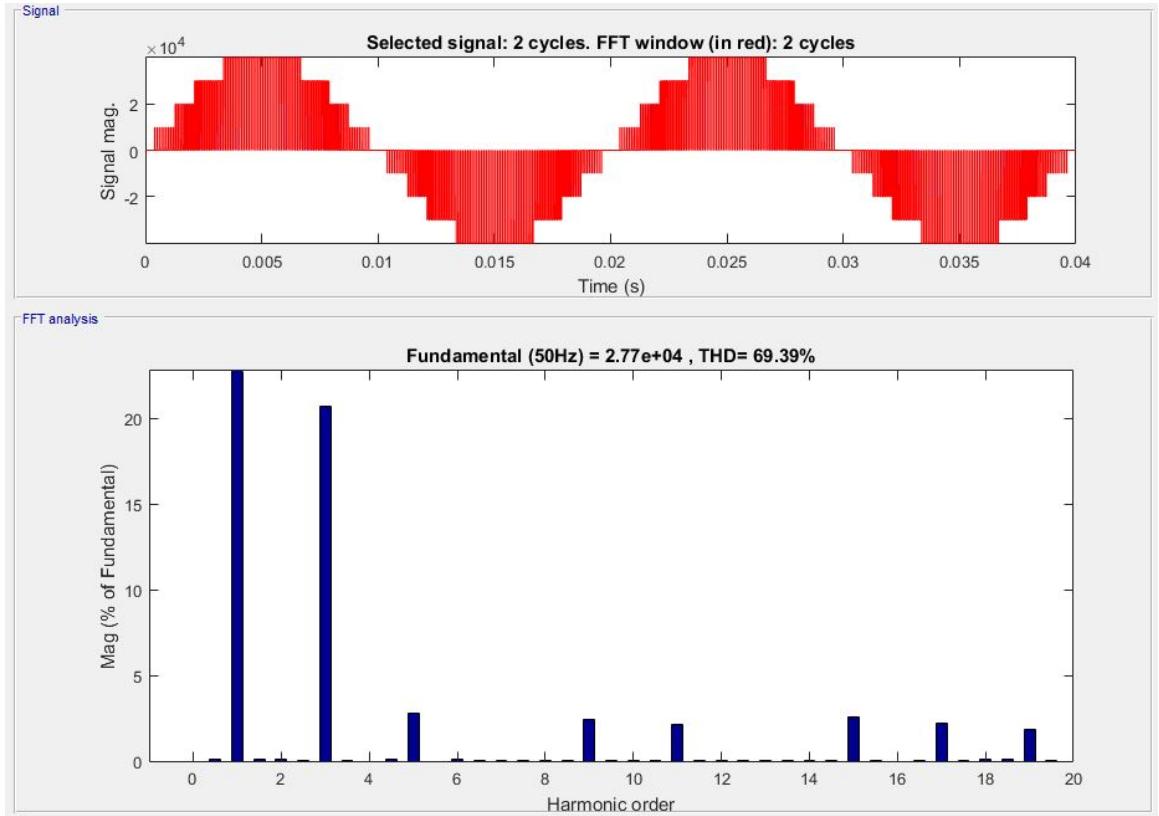


FIGURE 5.22: FFT of Output Signal after AND Operation

5.10 SHE PWM

Selective Harmonic Elimination is also very useful technique. For this project we have removed 5th, 7th and 13th harmonics form output waveform at 0.8 Modulation index. To do this we need to solve 4 non linear Equations:

$$\cos(\alpha_1) + \cos(\alpha_2) + \cos(\alpha_3) + \cos(\alpha_4) = 3.2 \quad (5.1)$$

$$\cos(5\alpha_1) + \cos(5\alpha_2) + \cos(5\alpha_3) + \cos(5\alpha_4) = 0 \quad (5.2)$$

$$\cos(7\alpha_1) + \cos(7\alpha_2) + \cos(7\alpha_3) + \cos(7\alpha_4) = 0 \quad (5.3)$$

$$\cos(11\alpha_1) + \cos(11\alpha_2) + \cos(11\alpha_3) + \cos(11\alpha_4) = 0 \quad (5.4)$$

The previous conduction angles that was found by hit and trial method are:

Conduction Angles	
Angle	Value in Degrees
α_1	6°
α_2	22°
α_3	38°
α_4	60°

By using fsolve command in MATLAB the following solution is found. And the new conduction angles are:

Conduction Angles for SHE PWM	
Angle	Value in Degrees
α_1	9.8409°
α_2	20.3828°
α_3	38.4054°
α_4	60.4164°

Below the results of SHE PWM are shown, it can be verified that the 5th, 7th and 11th harmonics are removed.

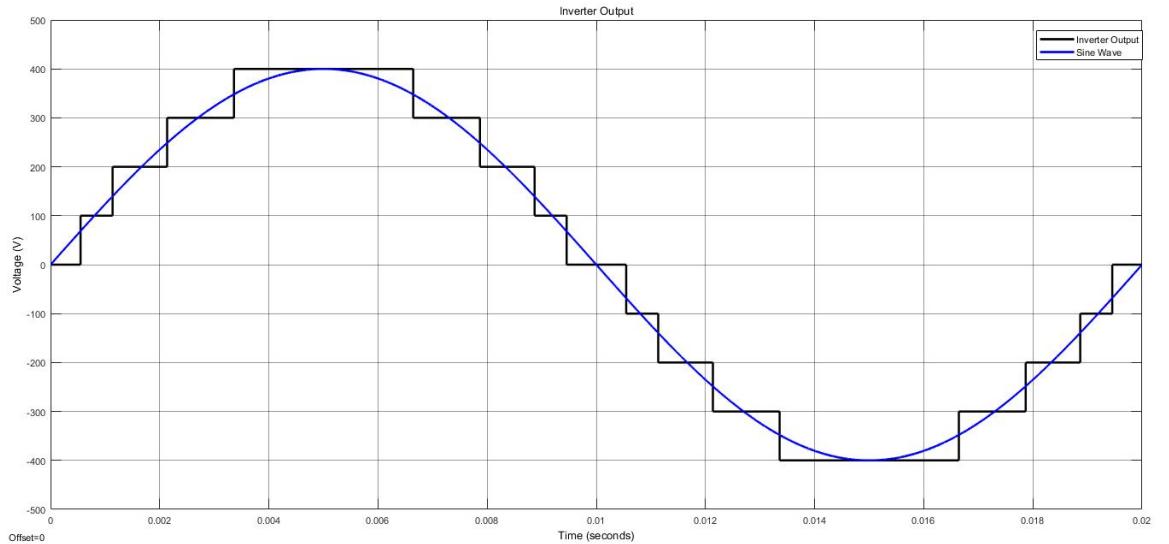


FIGURE 5.23: Output Signal (SHE PWM)

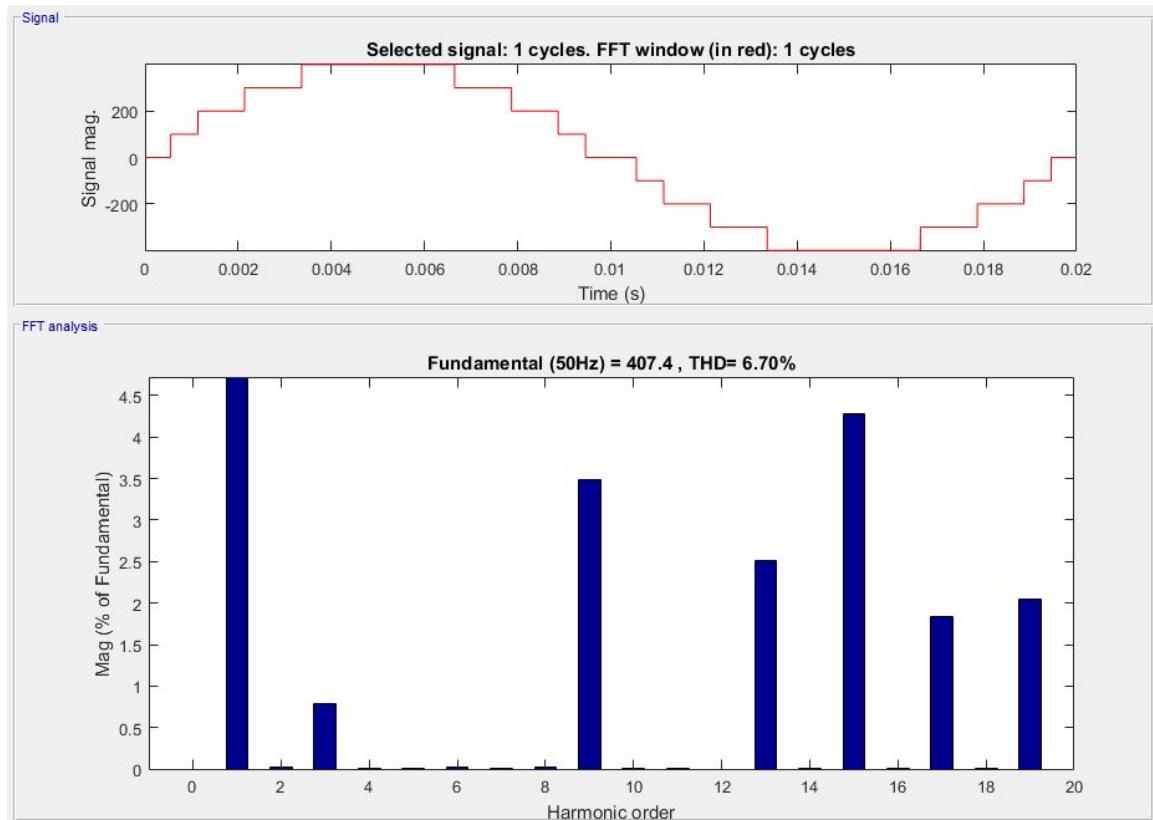


FIGURE 5.24: FFT of Output Signal (SHE PWM)

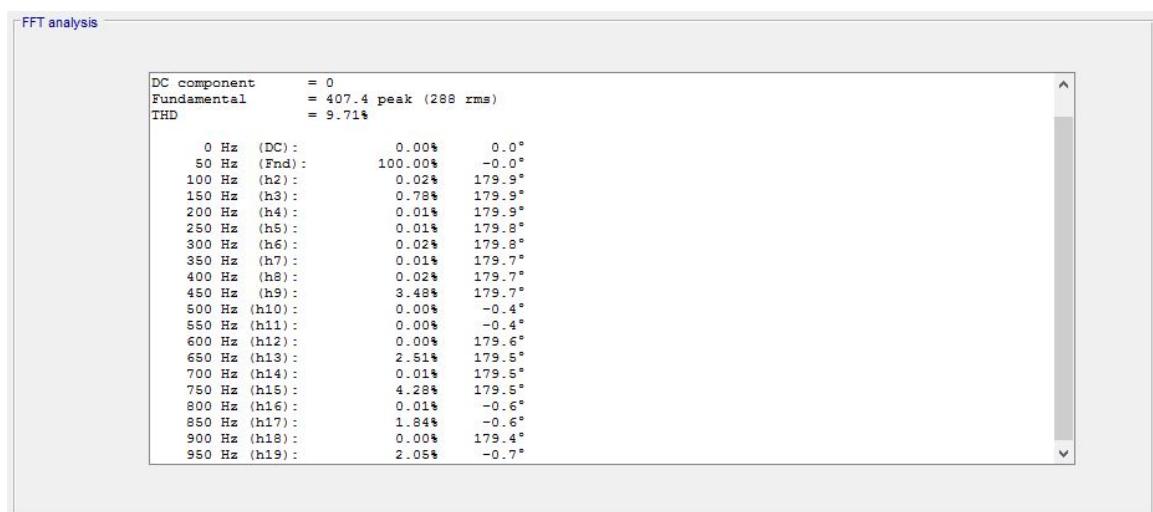


FIGURE 5.25: FFT Record Output Signal (SHE PWM)

5.11 Voltage Control

We can control the value of RMS output voltage by taking AND operation of gating signals with a high frequency PWM signal. Then by changing the duty of PWM signal we can change the value of RMS output voltage. Below is the relation between duty and V_{rms} :

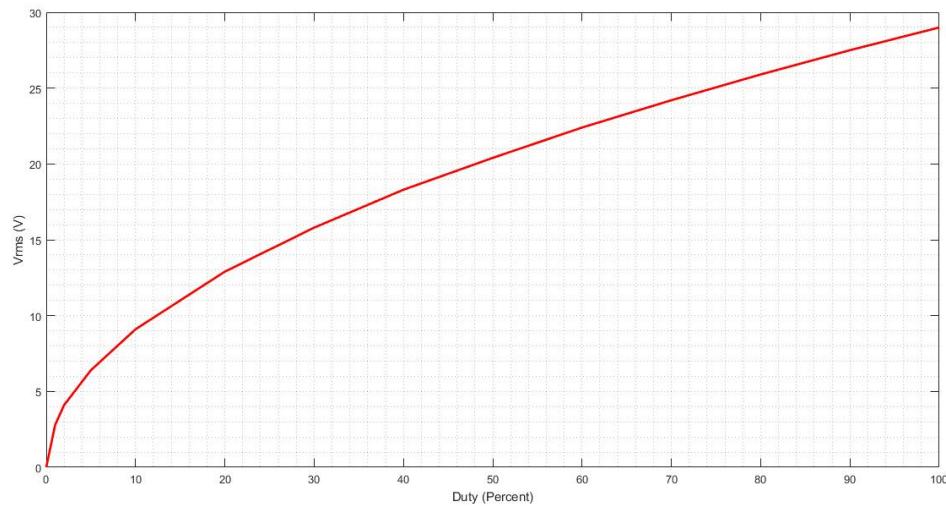


FIGURE 5.26: Relation between Duty Cycle of PWM and V_{rms}

Chapter 6

Hardware Results

6.1 Output Voltage

The output of Hardware circuit is shown below, the frequency can be verified that is 50Hz.

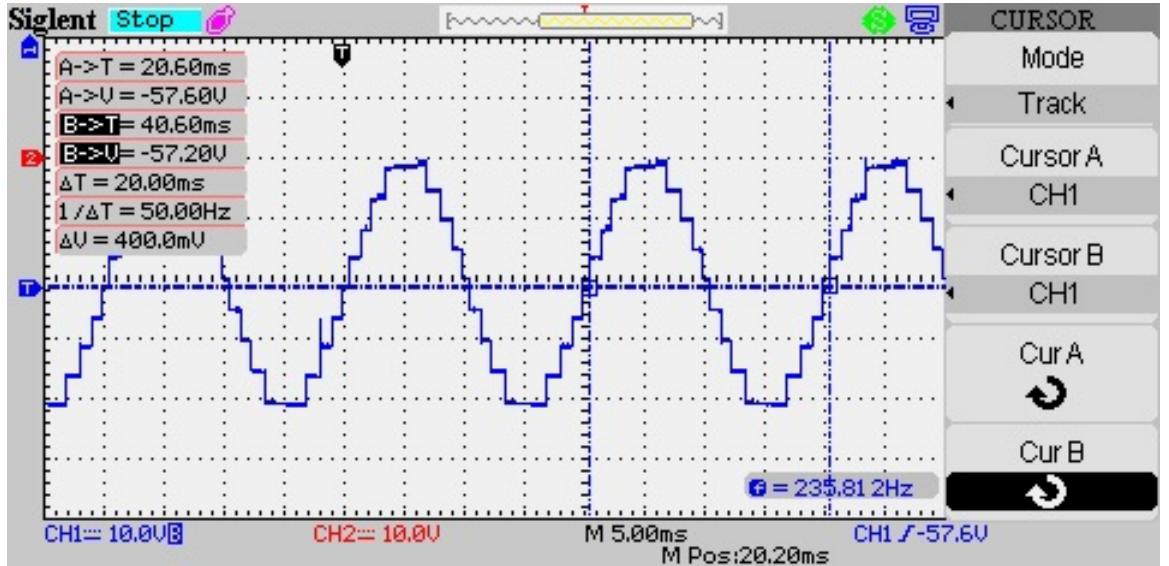


FIGURE 6.1: Output Voltage at 50Hz

The 9 levels in output waveform can be verified. The Value of voltage applied to H-Bridge 1 is 10V and the voltage applied to H-Bridge 2 is 30V i.e 3 times 10. The Switching sequence applied to all switches is shown in previous chapter. The maximum voltage is 40V and minimum voltage is -40V making it sinusoidal. The complete load parameters are shown in below figure.

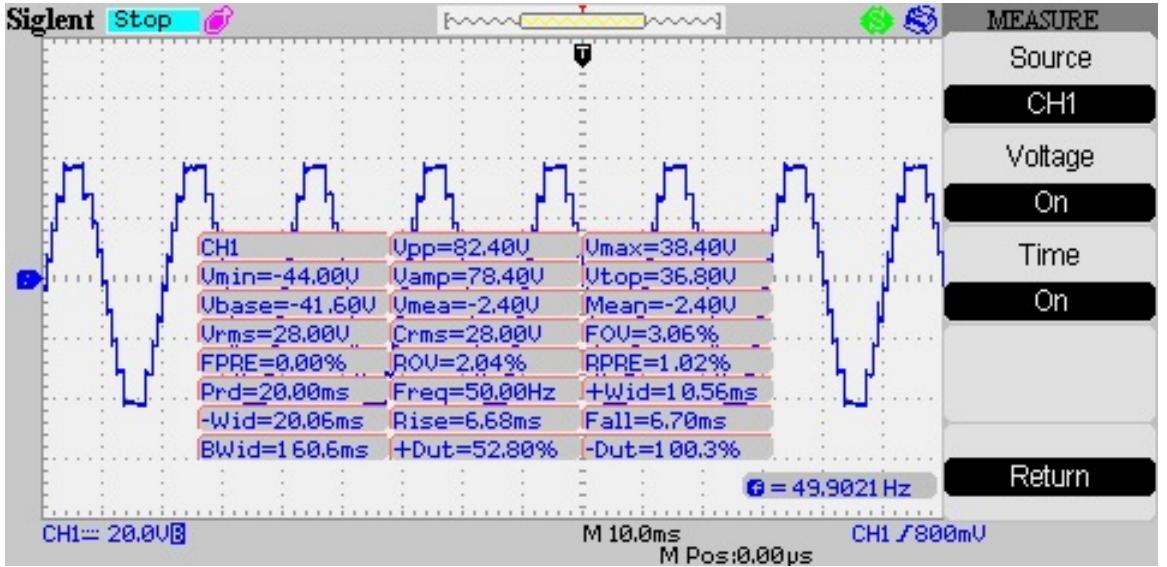


FIGURE 6.2: Output Voltage with all Load Parameters

6.2 Verification of Conduction Angles

The conduction angles that we need to verify are given in the below table:

Conduction Angles	
Angle	Value in Degrees
α_1	6°
α_2	22°
α_3	38°
α_4	60°

6.2.1 Verification of α_1

To verify α_1 we need to calculate the time delay first step and divide it to total time period and then multiply it with 360. From the figure shown below:

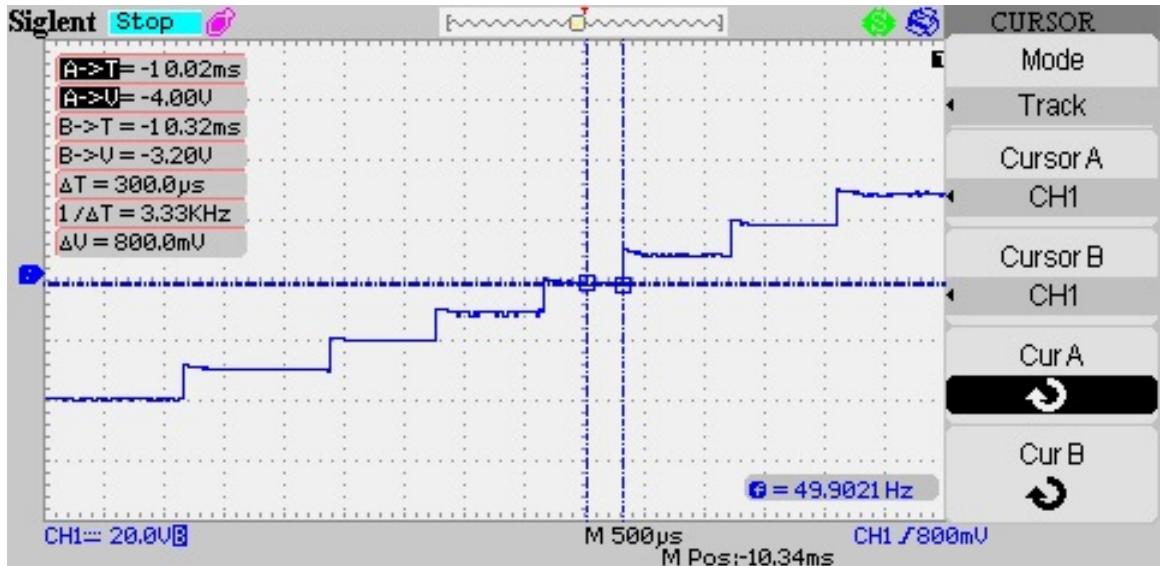
$$\Delta T = 0.3ms \quad (6.1)$$

$$T = 20ms \quad (6.2)$$

$$\alpha_1 = 360 * \frac{\Delta T}{T} \quad (6.3)$$

$$\alpha_1 = 360 * \frac{0.3ms}{20ms} \quad (6.4)$$

$$\alpha_1 = 5.4^\circ \quad (6.5)$$

FIGURE 6.3: Verification of $\alpha_1 = 6^\circ$

6.2.2 Verification of α_2

To verify α_2 we need to calculate the time delay of second step and divide it to total time period and than multiply it with 360. From the figure shown below:

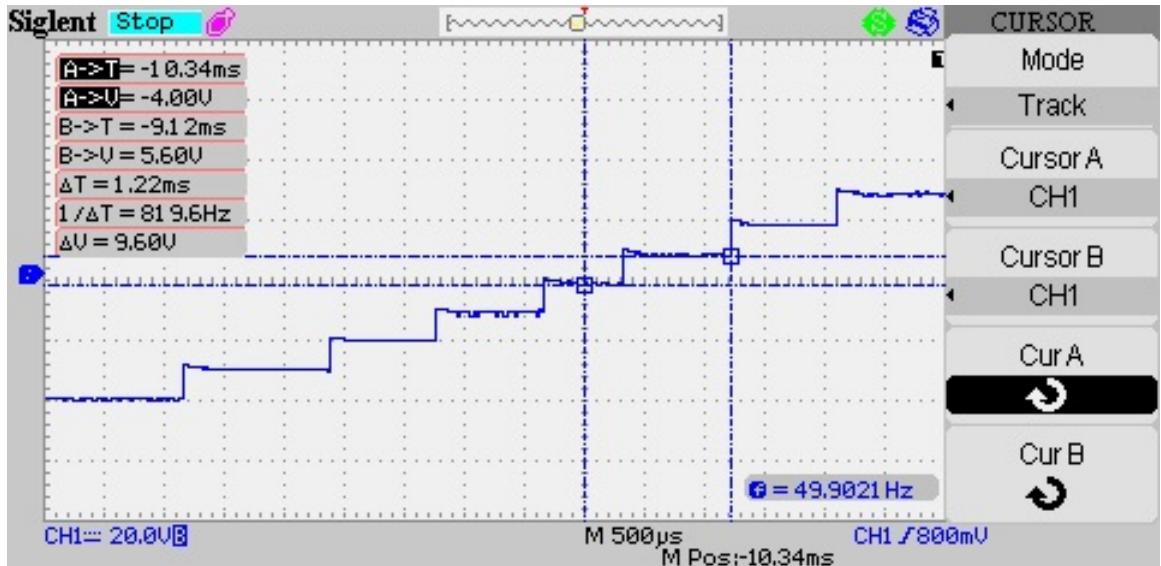
$$\Delta T = 1.22ms \quad (6.6)$$

$$T = 20ms \quad (6.7)$$

$$\alpha_1 = 360 * \frac{\Delta T}{T} \quad (6.8)$$

$$\alpha_1 = 360 * \frac{1.22ms}{20ms} \quad (6.9)$$

$$\alpha_1 = 21.96^\circ \quad (6.10)$$

FIGURE 6.4: Verification of $\alpha_2 = 22^\circ$

6.2.3 Verification of α_3

To verify α_3 we need to calculate the time delay of third step and divide it to total time period and than multiply it with 360. From the figure shown below:

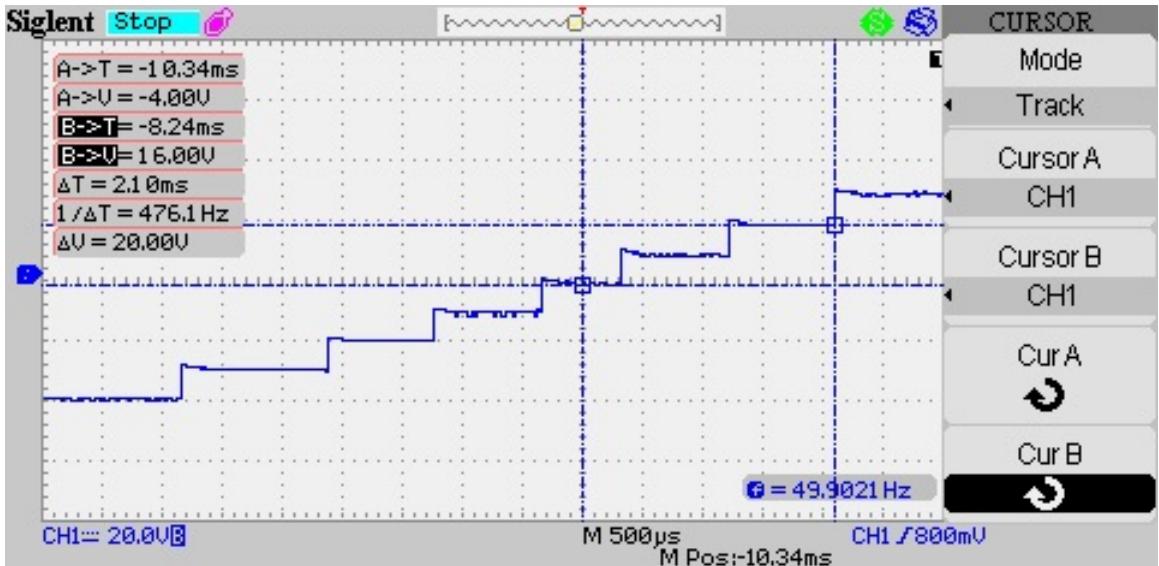
$$\Delta T = 2.10ms \quad (6.11)$$

$$T = 20ms \quad (6.12)$$

$$\alpha_1 = 360 * \frac{\Delta T}{T} \quad (6.13)$$

$$\alpha_1 = 360 * \frac{2.10ms}{20ms} \quad (6.14)$$

$$\alpha_1 = 37.8^\circ \quad (6.15)$$

FIGURE 6.5: Verification of $\alpha_3 = 38^\circ$

6.2.4 Verification of α_4

To verify α_3 we need to calculate the time delay of fourth step and divide it to total time period and than multiply it with 360. From the figure shown below:

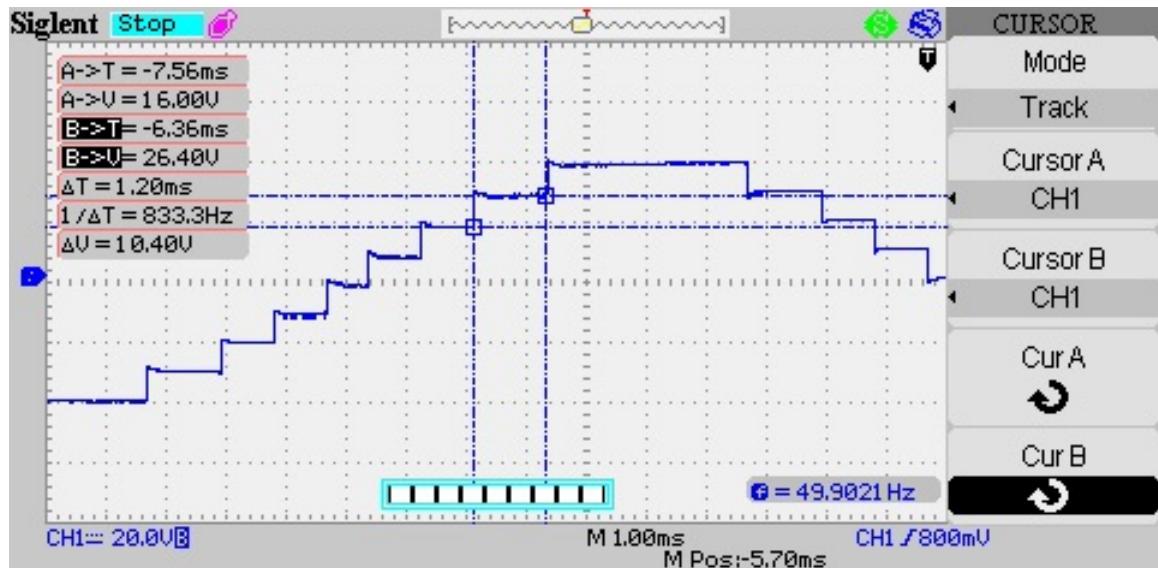
$$\Delta T = (2.10 + 1.2)ms \quad (6.16)$$

$$T = 20ms \quad (6.17)$$

$$\alpha_1 = 360 * \frac{\Delta T}{T} \quad (6.18)$$

$$\alpha_1 = 360 * \frac{(2.10 + 1.2)ms}{20ms} \quad (6.19)$$

$$\alpha_1 = 59.4^\circ \quad (6.20)$$

FIGURE 6.6: Verification of $\alpha_4 = 60^\circ$

6.3 AND Operation with 10KHz PWM

Below is the results of AND Operation with 10KHz PWM.

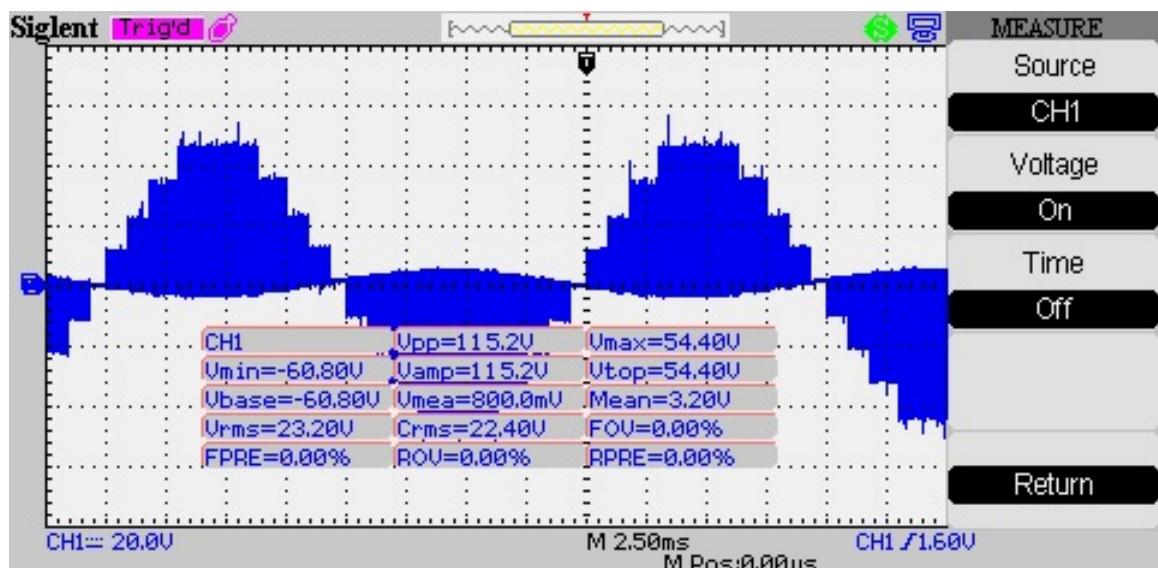


FIGURE 6.7: AND Operation with 10KHz PWM

6.4 PWM Operation

Below is the results when PWM Operation at 10KHz is applied.

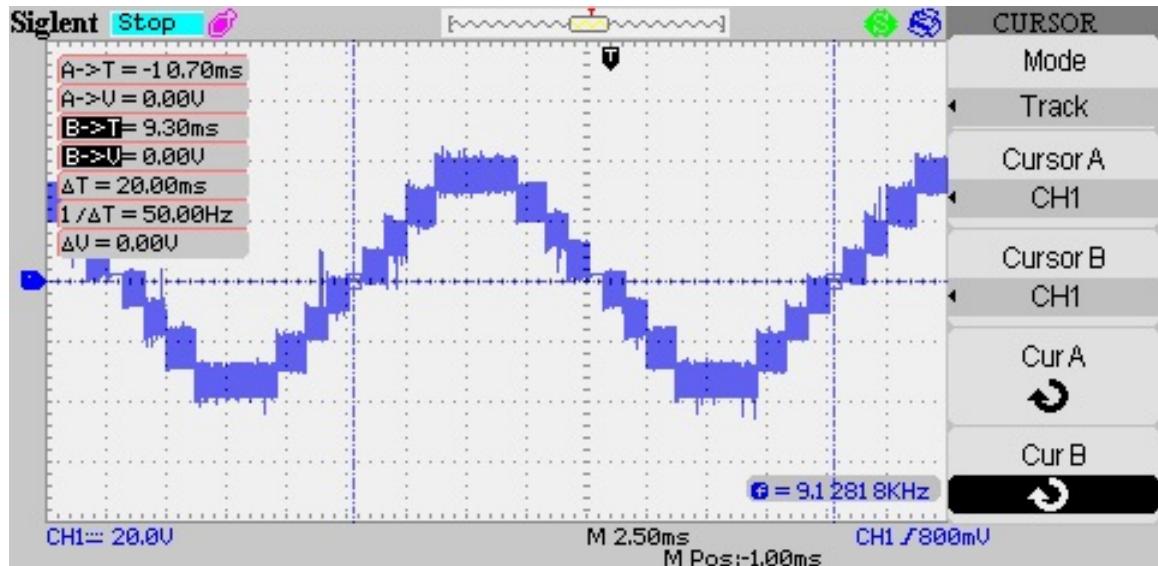


FIGURE 6.8: PWM Operation at 10KHz

6.5 Frequency Control

The frequency can be easily change by programming. Below is the result when the output frequency is 200Hz.

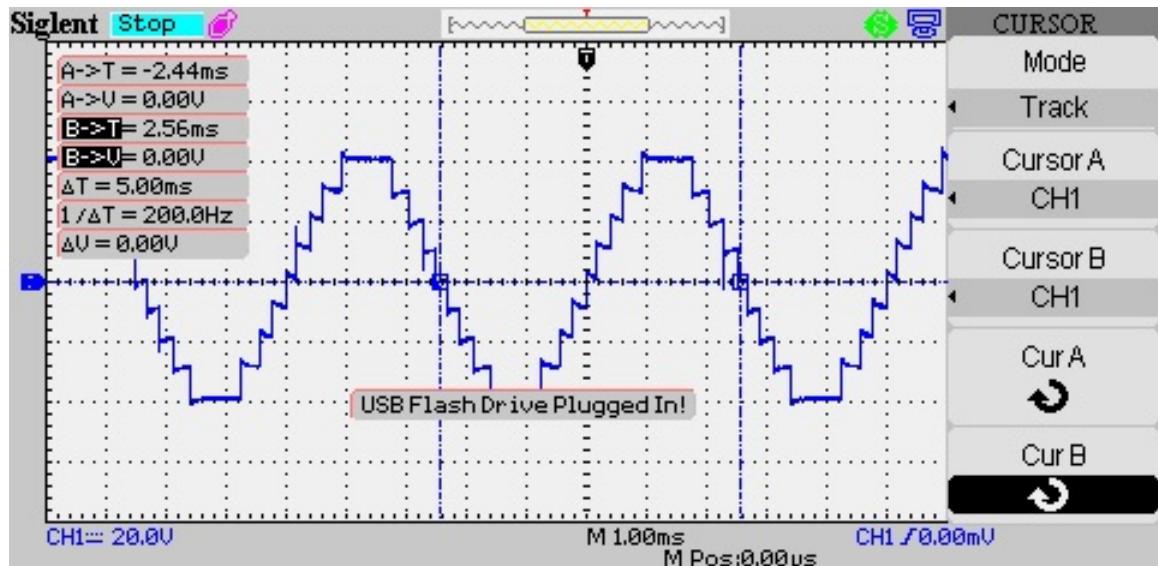


FIGURE 6.9: Frequency = 200Hz

6.6 Voltage Control

The output voltage can be controlled by taking AND Operation with a PWM signal at 10KHz frequency. Now by changing the value of duty cycle we can change the output voltage. Below are the figures showing output voltage at different duty cycles.

Duty Cycle = 100%

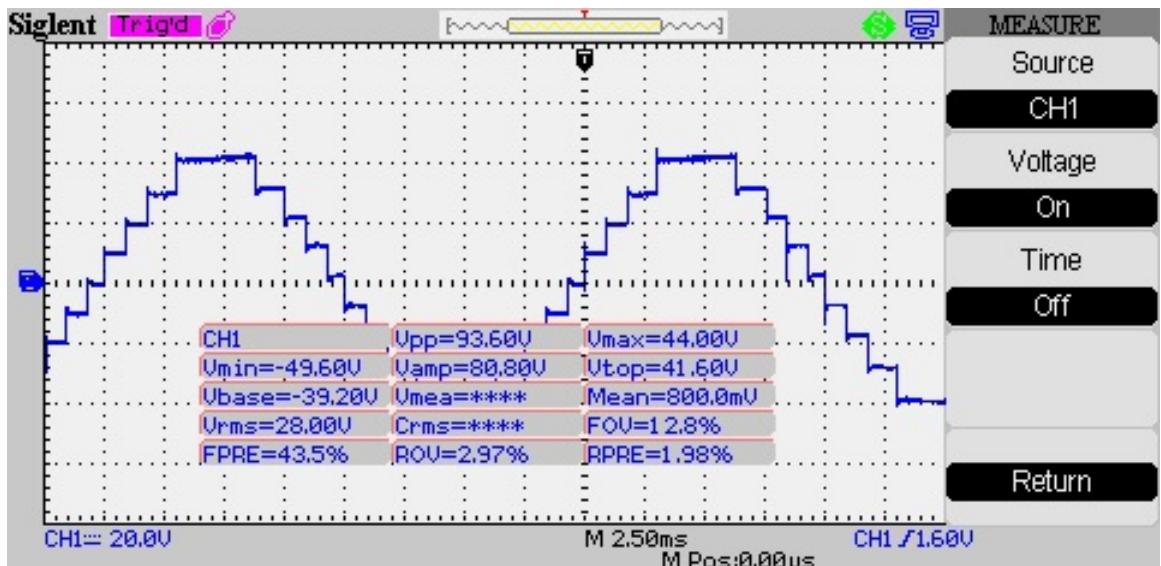


FIGURE 6.10: Duty Cycle = 100%

Duty Cycle = 90%

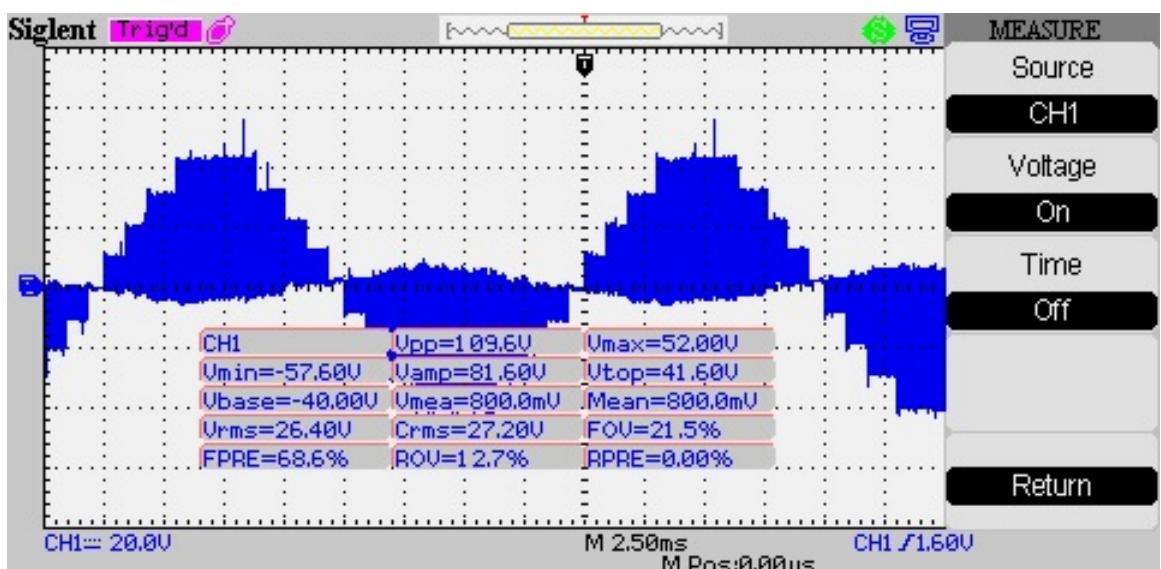


FIGURE 6.11: Duty Cycle = 90%

Duty Cycle = 80%

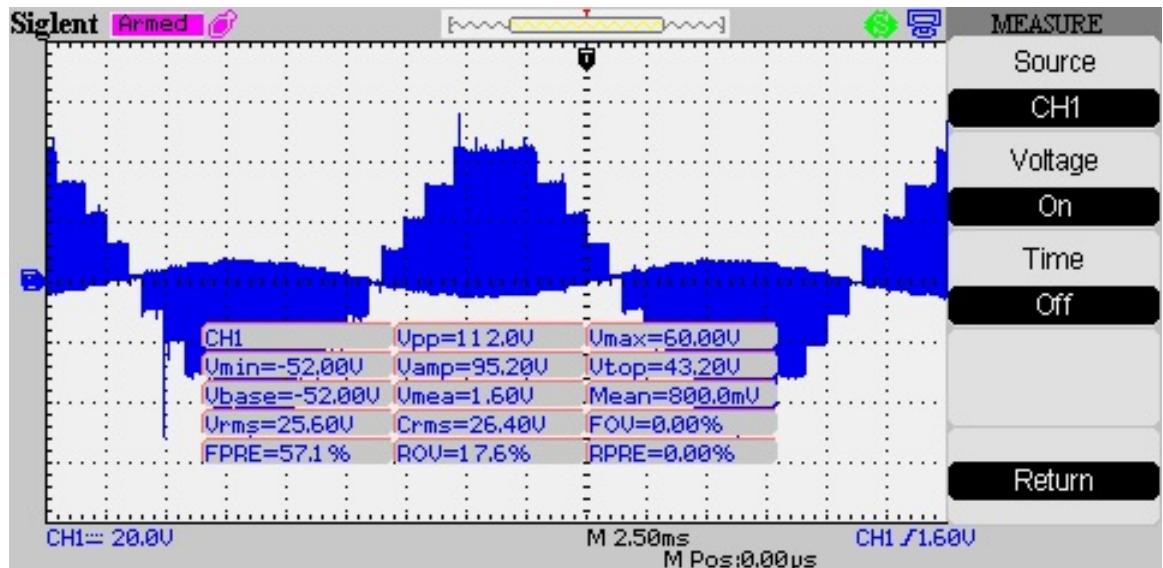


FIGURE 6.12: Duty Cycle = 80%

Duty Cycle = 70%

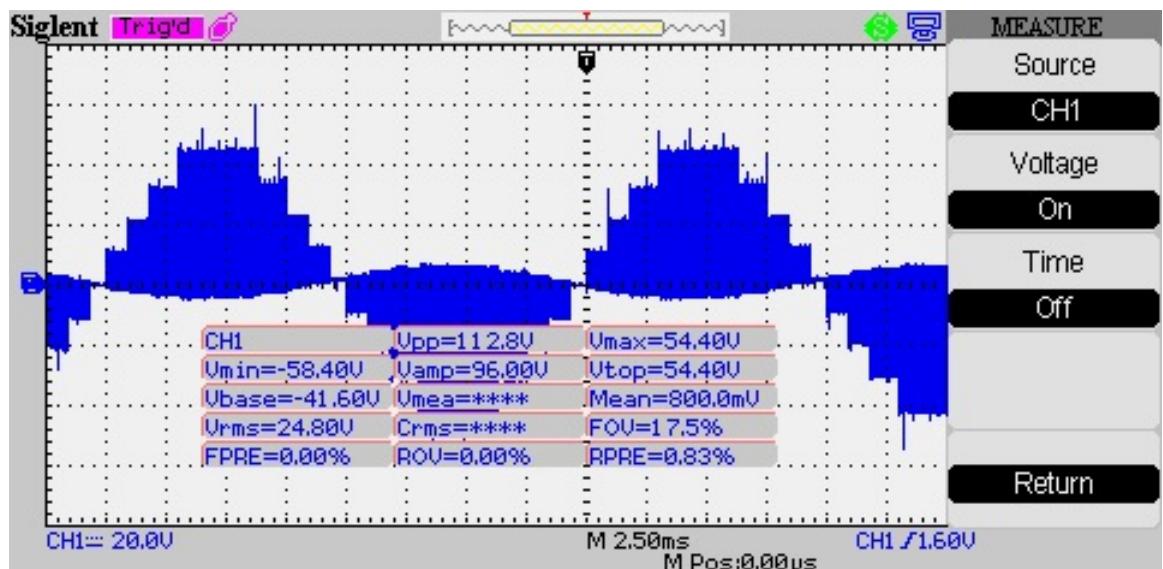


FIGURE 6.13: Duty Cycle = 70%

Duty Cycle = 60%

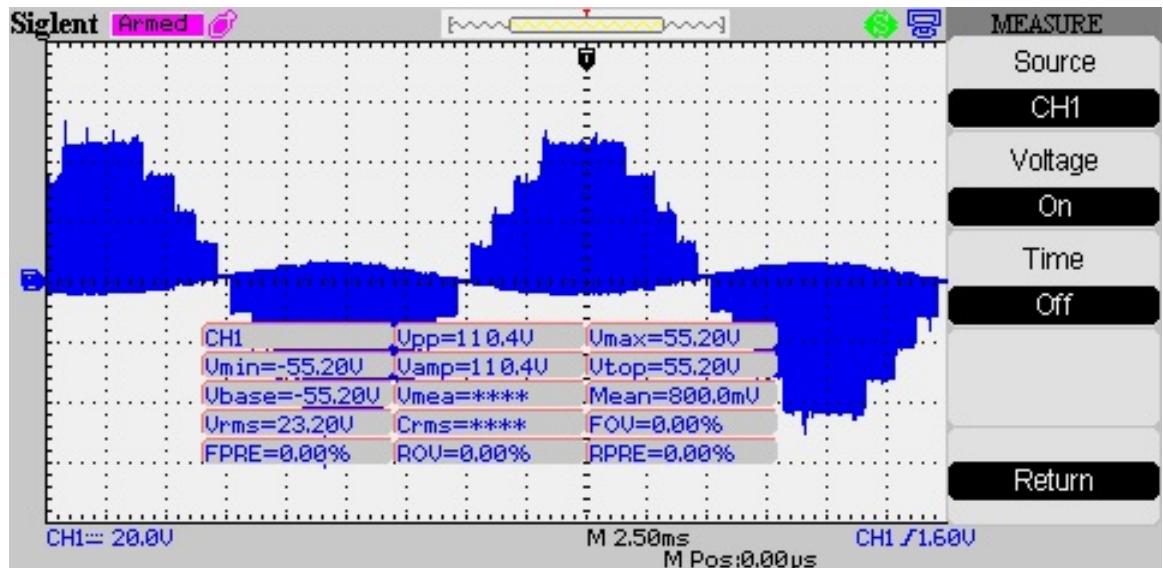


FIGURE 6.14: Duty Cycle = 60%

Duty Cycle = 50%

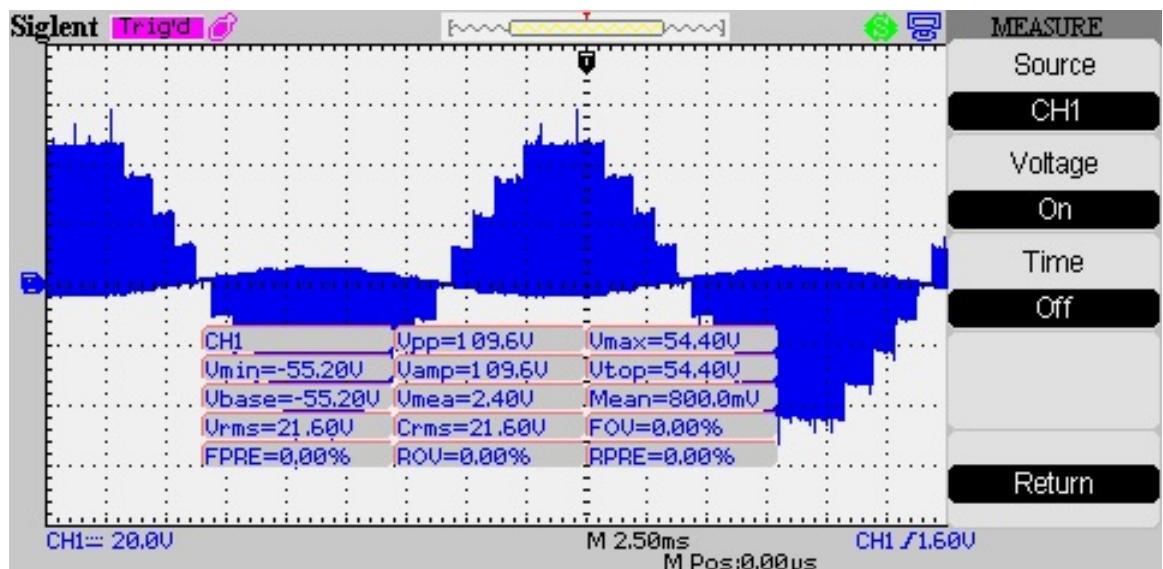


FIGURE 6.15: Duty Cycle = 50%

Duty Cycle = 40%

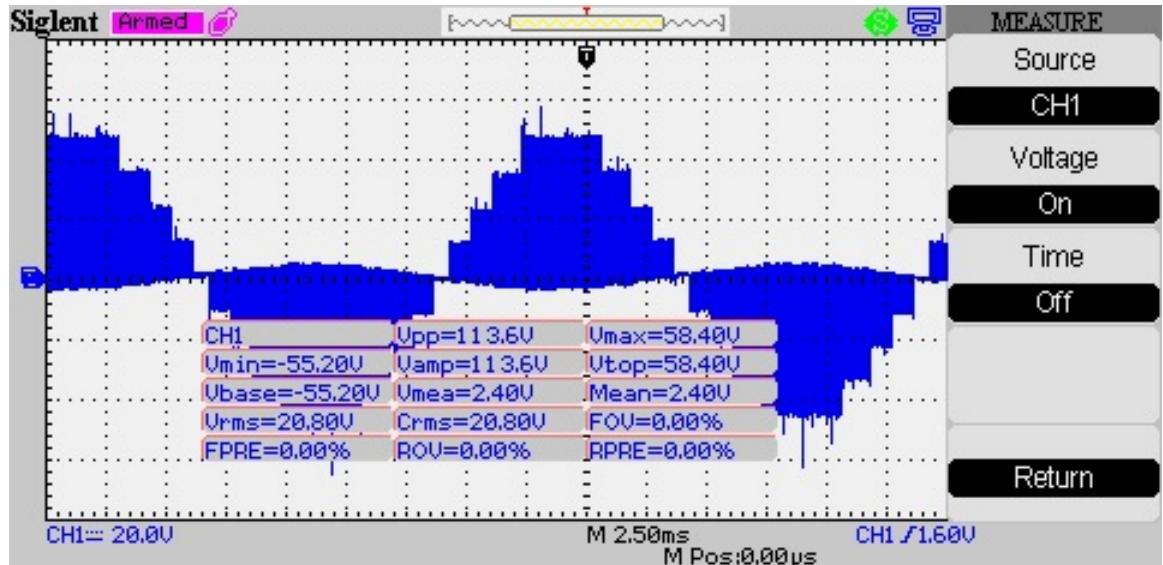


FIGURE 6.16: Duty Cycle = 40%

Duty Cycle = 30%

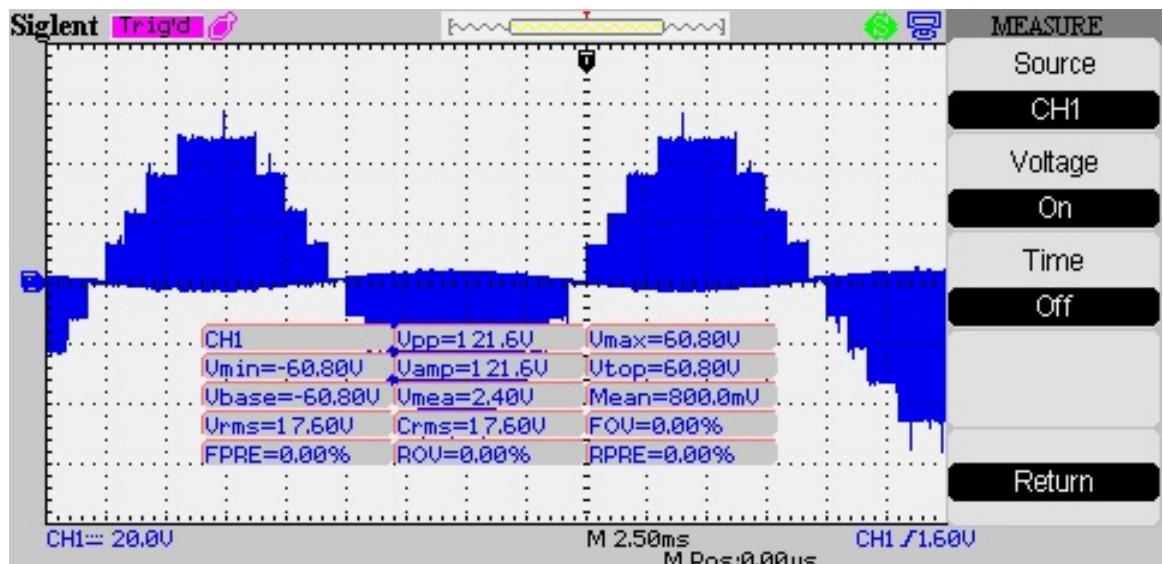


FIGURE 6.17: Duty Cycle = 30%

Duty Cycle = 20%

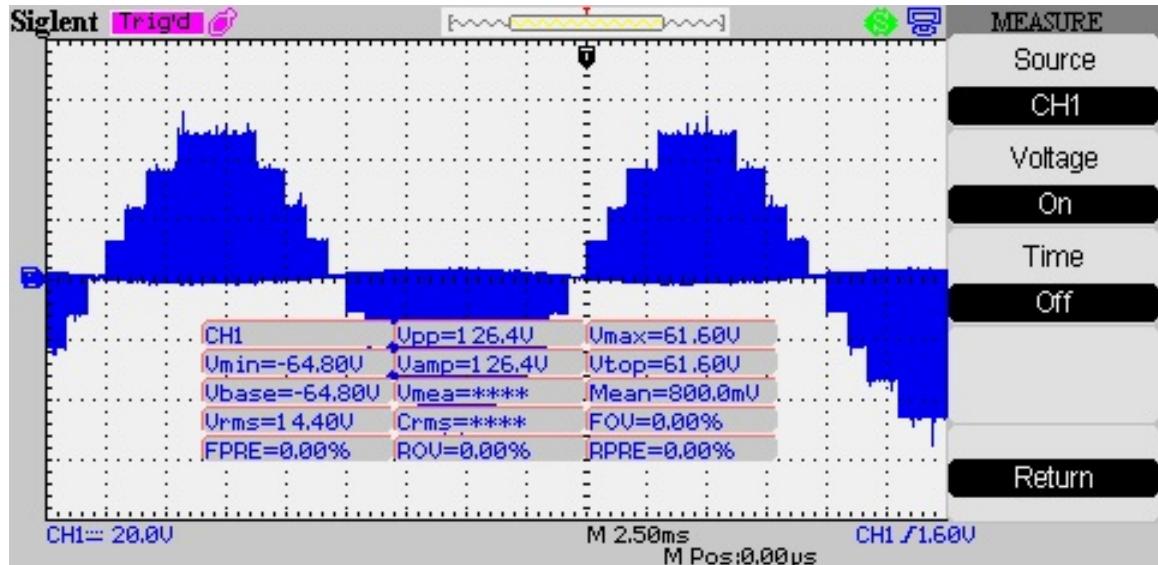


FIGURE 6.18: Duty Cycle = 20%

Duty Cycle = 10%

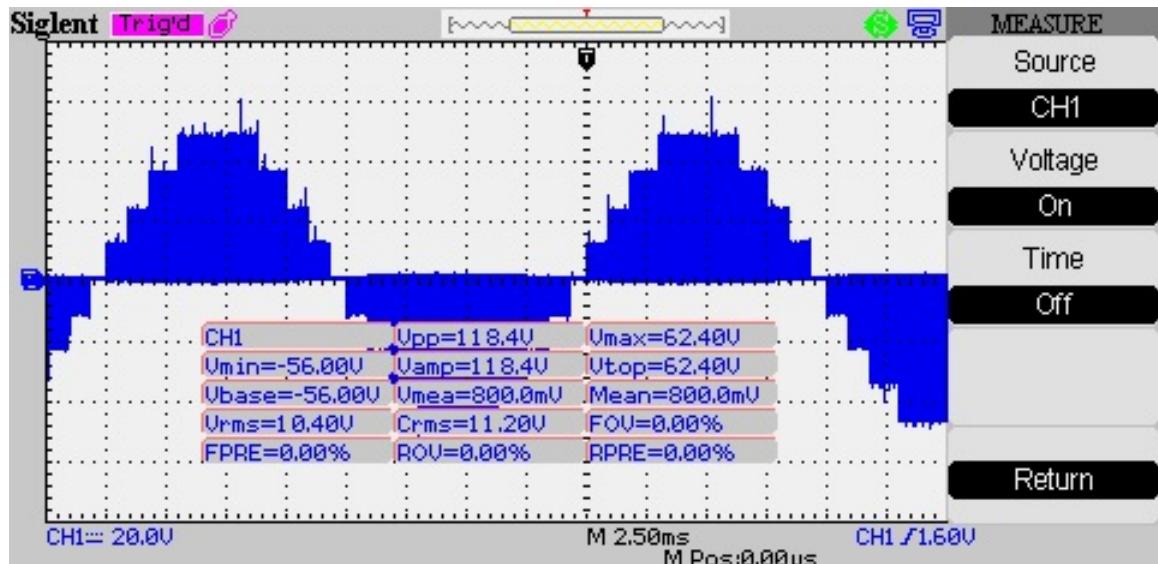


FIGURE 6.19: Duty Cycle = 10%

Duty Cycle = 5%

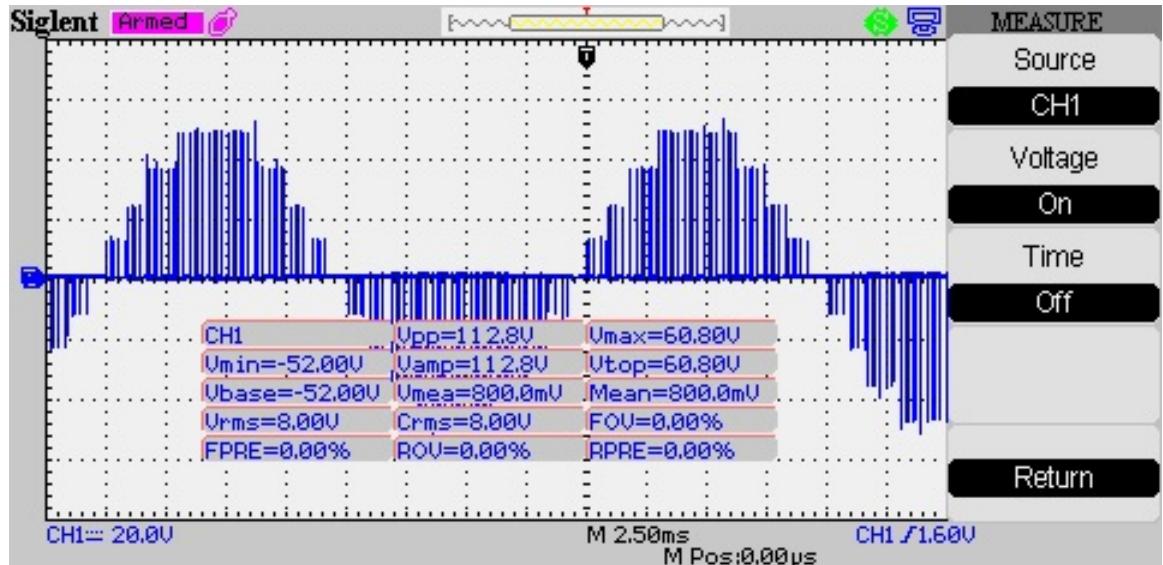


FIGURE 6.20: Duty Cycle = 5%

Duty Cycle = 2%

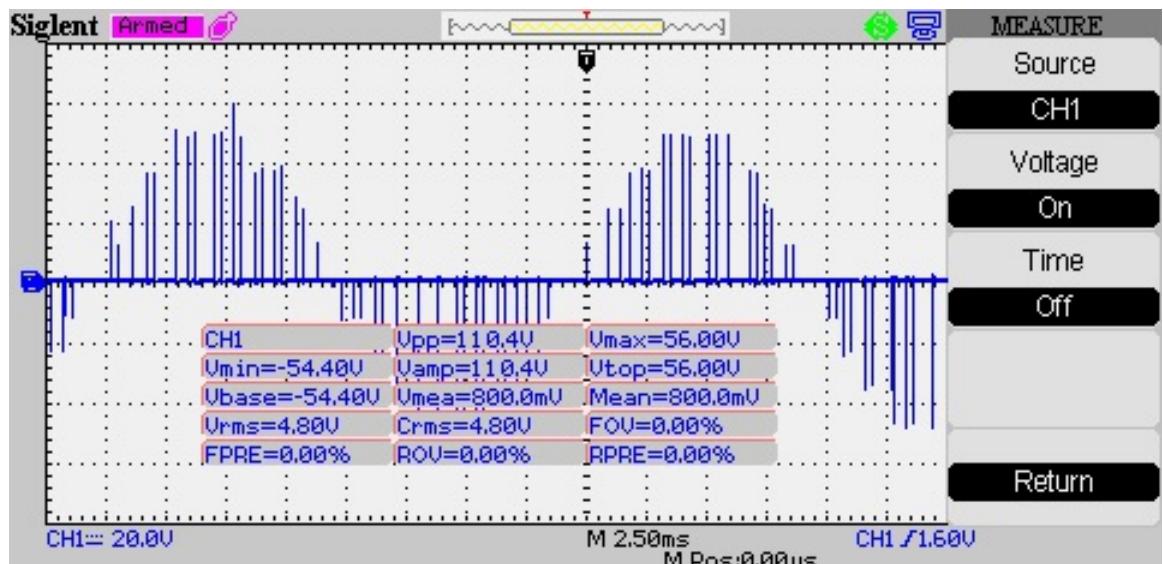


FIGURE 6.21: Duty Cycle = 2%

Duty Cycle = 1%

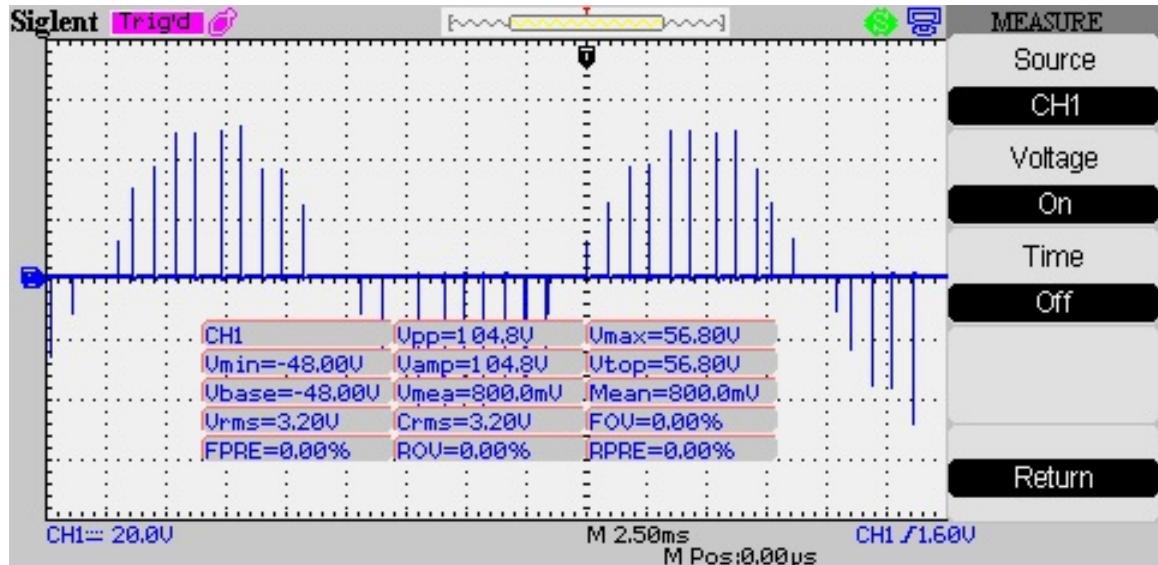


FIGURE 6.22: Duty Cycle = 1%

Duty Cycle = 0%

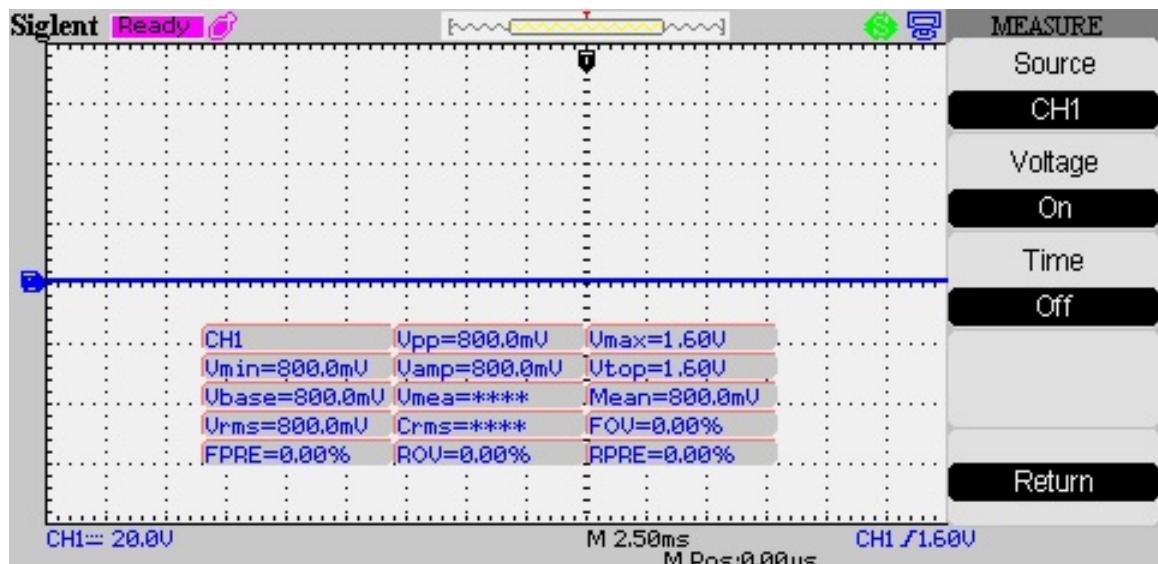


FIGURE 6.23: Duty Cycle = 0%

6.6.1 Table

Variation of V_{output} (RMS) w.r.t Duty cycle	
Duty Cycle (%)	V_{output} (RMS)
100	28
90	26.4
80	25.6
70	24.8
60	23.20
50	21.6
40	20.8
30	17.6
20	14.4
10	10.4
5	8
2	4.8
1	3.2
0	0.8

6.6.2 Relation Between Duty Cycle and V_{rms}

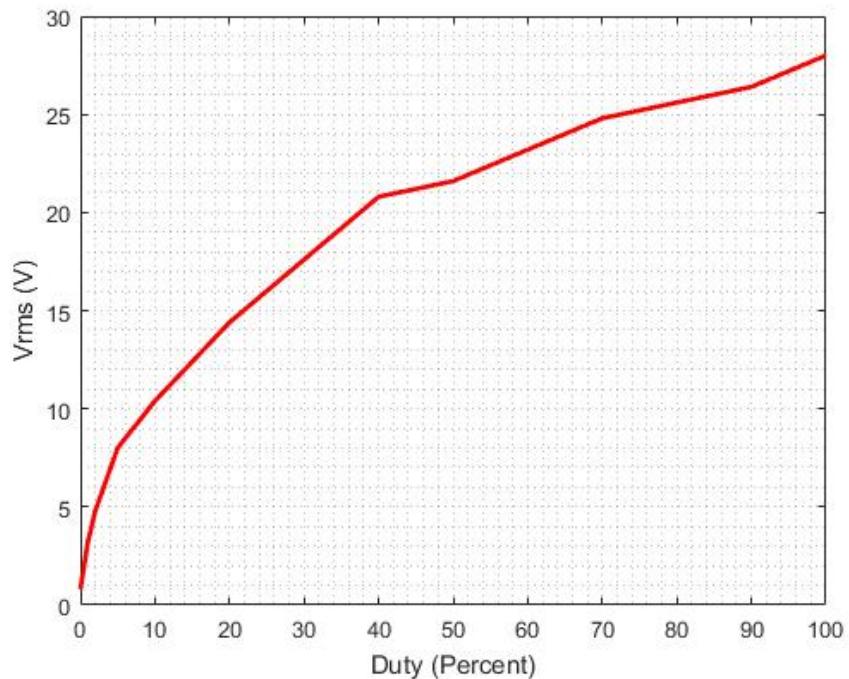


FIGURE 6.24: Relation Between Duty Cycle and V_{rms}

6.7 Voltage and Current Parameters

Below is the load parameters when 100W load is connected at the output terminals.

$$V_{H-Bridge-1} = 50V \quad (6.21)$$

$$V_{H-Bridge-2} = 150V \quad (6.22)$$

$$V_{out rms} = 150V \quad (6.23)$$

$$I_{out rms} = 2.205A \quad (6.24)$$

Load Parameters		
Component	Voltage	Current
Isolated Supplies for Gate Driver	15V	14mA
TLP250 Input (Gate Driver)	3.3V	15mA
TLP250 Output (Gate Driver)	15V	14mA
IRF450 (H-Bridge-1)	$50V_{rms}$	0.7352A
IRF450 (H-Bridge-2)	$150V_{rms}$	2.205A
STM32F4 (GPIO Pins)	3.3V	15mA

Chapter 7

Conclusion

In this paper, a three-phase 9-level cascaded multilevel inverter with output voltage and frequency control has been presented, achieving output signals with high quality and very low THD. The phase shift and disposition orders of modulating signals are another important point of design to reduce THD of line current and voltages. The DC level of H-bridges has been intended to construct output levels and dual DC supplies at $V_{dc}:3V_{dc}$ ratio have been constituted 9-level output voltage. The proposed model has been tested and compared to its precedent conventional models for THD rates and switching bandwidth. The measurement results have presented perfect outcomes on THD analysis. It is also seen that the switching frequency is directly effective on THD.

Appendix A

Code for Project

A.1 STM32F4 Code

```
/* Includes */
#include "stm32f4xx.h"
#include "stm32f401_discovery.h"
#include "math.h"
/*
PD0 -> S11
PD1 -> S13
PD2 -> S21
PD3 -> S23
PD4 -> S12
PD5 -> S14
PD6 -> S22
PD7 -> S24
*/
const int sine_table[50] = { 2089, 1052, 0, 1052, 2088, 3092, 4046, 4937, 5750,
    6472, 7092, 7600, 7988, 8251, 8383, 8383, 8251, 7988, 7600, 7092, 6472,
    5750, 4937, 4046, 3092, 2089, 1052, 0, 1052, 2088, 3092, 4046, 4937,
    5750, 6472, 7092, 7600, 7988, 8251, 8383, 8383, 8251, 7988, 7600, 7092,
    6472, 5750, 4937, 4046, 3092 };

const int hex_values[17] = { 0xF0, 0xF0, 0xE1, 0x96, 0x87, 0xA5, 0x87, 0x96,
    0xE1, 0xF0, 0xD2, 0x69, 0x4B, 0x5A, 0x4B, 0x69, 0xD2 };
uint16_t arr_values[17] = { 0 };
uint8_t s11, s13, s21, s23, ss;
uint8_t index0 = 1, sin_index = 0;
uint16_t duty = 100;
float m_a = 1.0f;

void timer3_Init() {
```

```
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);
TIM_TimeBaseInitTypeDef tim;
tim.TIM_Period = arr_values[0] - 1;
tim.TIM_Prescaler = 9;
tim.TIM_ClockDivision = TIM_CKD_DIV1;
tim.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM3, &tim);

NVIC_InitTypeDef nvic;
nvic.NVIC IRQChannel = TIM3 IRQn;
nvic.NVIC IRQChannelCmd = ENABLE;
NVIC_Init(&nvic);

TIM_ITConfig(TIM3, TIM_IT_Update, ENABLE);

TIM_ARRPreloadConfig(TIM3, ENABLE);

TIM_Cmd(TIM3, ENABLE);
}

void timer2_Init() {
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
TIM_TimeBaseInitTypeDef tim;
tim.TIM_Period = 33599;
tim.TIM_Prescaler = 0;
tim.TIM_ClockDivision = TIM_CKD_DIV1;
tim.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM2, &tim);

NVIC_InitTypeDef nvic;
nvic.NVIC IRQChannel = TIM2 IRQn;
nvic.NVIC IRQChannelCmd = ENABLE;
NVIC_Init(&nvic);
TIM_ITConfig(TIM2, TIM_IT_Update, ENABLE);

TIM_ARRPreloadConfig(TIM2, ENABLE);
TIM_Cmd(TIM2, ENABLE);
}

void init_GPIOs() {
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

GPIO_InitTypeDef gpio;

gpio.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 | GPIO_Pin_3
| GPIO_Pin_4 | GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7;
```

```
    gpio.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_Init(GPIOD, &gpio);

    gpio.GPIO_Pin = GPIO_Pin_0;
    gpio.GPIO_Mode = GPIO_Mode_IN;
    GPIO_Init(GPIOA, &gpio);
}

void PWM_Init() {

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);

    GPIO_InitTypeDef gpio;
    gpio.GPIO_Pin = GPIO_Pin_12;
    gpio.GPIO_Mode = GPIO_Mode_AF;
    gpio.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_Init(GPIOD, &gpio);

    GPIO_PinAFConfig(GPIOD, GPIO_PinSource12, GPIO_AF_TIM4);

    TIM_TimeBaseInitTypeDef tim;
    tim.TIM_Period = 8399;
    tim.TIM_Prescaler = 0;
    tim.TIM_ClockDivision = TIM_CKD_DIV1;
    tim.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM4, &tim);

    TIM_OCInitTypeDef tim_oc;
    tim_oc.TIM_OCMode = TIM_OCMode_PWM1;
    tim_oc.TIM_OCPolarity = TIM_OCPolarity_High;
    tim_oc.TIM_OutputState = TIM_OutputState_Enable;
    tim_oc.TIM_Pulse = 4199;
    TIM_OC1Init(TIM4, &tim_oc);
    TIM_OC1PreloadConfig(TIM4, TIM_OCPreload_Enable);
    TIM_Cmd(TIM4, ENABLE);

}

void init_ARR() {
    double p = 20; // 20ms, 50Hz
    double ca[4] = { 6, 22, 38, 60 }; //{ 6, 22, 38, 60 }; // {9.8409, 20.3828,
    // 38.4054, 60.4164}
    double delay_time[17] = { 0 };
    double sum = 0;
    int i = 0;
    for (i = 0; i < 17; i++) {
```

```
if (i < 4) {
    delay_time[i] = (ca[i] / 360) * p - sum;
    sum += delay_time[i];
} else if (i == 4) {
    delay_time[i] = (p / 2 - (ca[3] / 360) * p) - sum;
} else if (i > 4 && i < 8) {
    delay_time[i] = delay_time[8 - i];
} else if (i == 8) {
    delay_time[i] = delay_time[0] * 2;
} else if (i > 8 && i < 16) {
    delay_time[i] = delay_time[i - 8];
} else if (i == 16) {
    delay_time[16] = delay_time[0];
}
}

for (i = 0; i < 17; i++) {
    arr_values[i] = lround((double) (delay_time[i] * 8400));
}
}

int main(void) {
    ss = 1;
    int index1 = 0;
    uint16_t CCR_value = 0;
    uint8_t AND = 2;
    uint8_t spwm = 0;
    init_GPIOs();
    init_ARR();
    PWM_Init();
    timer3_Init();
    if (spwm) {
        timer2_Init();
    }
    while (1) {
        if (!spwm) {
            CCR_value = (uint16_t) (((float) (duty / 100.0)) * 8400);
            TIM4->CCR1 = CCR_value;
        }
        index1 = index0 - 1;
        if (index1 < 0) {
            index1 = 16;
        }
        ss = GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0);

        if (AND == 2) {
            GPIO_Write(GPIOD, (hex_values[index1]));
        }
    }
}
```

```
if (AND == 1) {
    GPIO_Write(GPIOB, (hex_values[index1] * ss));
} else if (AND == 0) {

    GPIO_Write(GPIOB, (hex_values[index1]));

    if (index1 == 6 || index1 == 7 || index1 == 14 || index1 == 15) {
        if (ss == 0) {
            GPIO_Write(GPIOB, (hex_values[index1 + 1]));
        }
    }
    if (index1 == 2 || index1 == 8 || index1 == 10 || index1 == 16) {
        GPIO_Write(GPIOB, (hex_values[index1] * ss));
    }
    if (index1 == 3 || index1 == 4 || index1 == 5 || index1 == 11
    || index1 == 12 || index1 == 13) {
        if (ss == 0) {
            GPIO_Write(GPIOB, (hex_values[index1 - 1]));
        }
    }
}
/* s11 = GPIO_ReadOutputDataBit(GPIOB, GPIO_Pin_0);
s13 = GPIO_ReadOutputDataBit(GPIOB, GPIO_Pin_1);
s21 = GPIO_ReadOutputDataBit(GPIOB, GPIO_Pin_2);
s23 = GPIO_ReadOutputDataBit(GPIOB, GPIO_Pin_3);*/
}

void TIM3_IRQHandler() {
    if (TIM_GetITStatus(TIM3, TIM_IT_Update)) {
        TIM_ClearITPendingBit(TIM3, TIM_IT_Update);

        TIM3->ARR = arr_values[index0] - 1;
        index0 = (index0 + 1) % 17;

    }
}

void TIM2_IRQHandler() {
    if (TIM_GetITStatus(TIM2, TIM_IT_Update)) {
        TIM_ClearITPendingBit(TIM2, TIM_IT_Update);
        TIM4->CCR1 = (uint16_t) (m_a * sine_table[sin_index]);
        sin_index = (sin_index + 1) % 50;
    }
}

/*
```

```

* Callback used by stm32f401_discovery_audio_codec.c.
* Refer to stm32f401_discovery_audio_codec.h for more info.
*/
void EVAL_AUDIO_TransferComplete_CallBack(uint32_t pBuffer, uint32_t Size) {
    /* TODO, implement your code here */
    return;
}

/*
* Callback used by stm32f401_discovery_audio_codec.c.
* Refer to stm32f401_discovery_audio_codec.h for more info.
*/
uint16_t EVAL_AUDIO_GetSampleCallBack(void) {
    /* TODO, implement your code here */
    return -1;
}

```

A.2 MATLAB CODE

A.2.1 Switching Sequence Generator:

```

clear;clc;
%% Sequence Generator
% This script generates 4 switching sequences o11, 013, o21, o23 for
% switches Q11, Q13, Q21, Q23 respectively. o12, o14, o22 and o24 can be
% obtain by inverting o11, 013, o21, o23 respectively
% -----
%% Input
options.Interpreter = 'tex';
% Include the desired Default answer
options.Default = 'No PWM';
% Use the TeX interpreter in the question
qstring = 'Please Select Operation Mode?';
choice = questdlg(qstring,'Operation Mode',...
'No PWM','PWM Operation','SPWM Operation',options);

% Defining Variables
d = 100; % Duty
ma = 1.0; % Modulation Index

% Handle response
switch choice
case 'No PWM'
option = 0;
case 'PWM Operation'
option = 1;
answer = inputdlg("Enter Duty Value: ");

```

```

d = str2double(answer{1}); % Store Duty Value
case 'SPWM Operation'
option = 2;
answer = inputdlg("Enter Value of Modulation Index: ");
ma = str2double(answer{1}); % Store Value of Modulation Index
end
%% Initialization

Fs = 10000; % Samples per Period
p = 20e-3; % Time Period. (Variable Frequency can be obtained by changing period
            value)
t = linspace(0, p, Fs); % Time axis
duty = d; % From User Input
sw_freq = 10e3; % Switching Frequency for PWM Operation
if (option == 2) % Sine PWM
pwm = SPWM(sw_freq, t, ma);
end
if (option == 1) % PWM
pwm = (1 + square(2*pi*sw_freq*t, duty))/2;
end
if (option == 0) % No PWM
pwm = ones(1, Fs);
end
o11 = zeros(1, Fs);
o13 = zeros(1, Fs);
o21 = zeros(1, Fs);
o23 = zeros(1, Fs);
%% Data

% Conduction Angles or Switching Angles in Degrees
cond_angles = [6 22 38 60]; %(for SHE PWM)[9.8409 20.3828 38.4054 60.4164];

% Switching Sequence with minimum transitions
s11 = [0 1 0 1 1 1 0 1 0 0 1 1 0 1 1 0 0];
s13 = [0 0 1 1 0 1 1 0 0 1 0 1 1 1 0 1 0];
s21 = [0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0];
s23 = [0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0];

%% Calculations
% Calculating Time Axis values of all steps based on conduction angles
time_value = [
(cond_angles(1)/360)*p
((cond_angles(2)/360)*p)
((cond_angles(3)/360)*p)
((cond_angles(4)/360)*p)
((p/2 - (cond_angles(4)/360)*p))
((p/2 - (cond_angles(3)/360)*p))
```

```

((p/2 - (cond_angles(2)/360)*p))
((p/2 - (cond_angles(1)/360)*p))
((p/2 + (cond_angles(1)/360)*p))
((p/2 + (cond_angles(2)/360)*p))
((p/2 + (cond_angles(3)/360)*p))
((p/2 + (cond_angles(4)/360)*p))
((p - (cond_angles(4)/360)*p))
((p - (cond_angles(3)/360)*p))
((p - (cond_angles(2)/360)*p))
((p - (cond_angles(1)/360)*p))

p
];
%% Output Generation
% Based on conduction angles generating sequences for Q11, Q13, Q21 and
% Q23
j = 1;
for i = 1:Fs
o11(i) = s11(j)*pwm(i);
o13(i) = s13(j)*pwm(i);
o21(i) = s21(j)*pwm(i);
o23(i) = s23(j)*pwm(i);

if t(i) > time_value(j)
j = j + 1;
end
end

```

A.2.2 Sine PWM Generator

```

function y = SPWM(f, t, ma)
counter = 1;
y = zeros(1, length(t));
table = zeros(1, 200);
for angle = 0:0.9:179
table(counter) = ma*100*sind(angle);
counter = counter + 1;
end
sin_table = [table table];
del = length(t)/length(sin_table);
lower = 1;
upper = del;
angle = 1;
for counter = 1:del:length(t)
y(lower:upper) = (1 + square(2*pi*f*t(lower:upper), sin_table(angle)))/2;
lower = upper + 1;
upper = upper + del;

```

```
angle = angle + 1;
end
```

A.2.3 SHE PWM

```
clear; clc;
% x = [9.8409    20.3828    38.4054    60.4164]
x0 = [6 22 38 60];
fun = @set_of_equations;
options = optimoptions('fsolve', 'MaxFunctionEvaluations', 20000,
    'OptimalityTolerance', 1e-6, 'StepTolerance', 1e-6, 'MaxIterations', 2000,
    'PlotFcn', @optimplotfval);
x = fsolve(fun, x0, options);
```

A.2.4 Equations

```
function F = set_of_equations(x)
F = zeros(1,4);
F(1) = cosd(x(1)) + cosd(x(2)) + cosd(x(3)) + cosd(x(4)) - 3.2;
F(2) = cosd(5*x(1)) + cosd(5*x(2)) + cosd(5*x(3)) + cosd(5*x(4));
F(3) = cosd(7*x(1)) + cosd(7*x(2)) + cosd(7*x(3)) + cosd(7*x(4));
F(4) = cosd(11*x(1)) + cosd(11*x(2)) + cosd(11*x(3)) + cosd(11*x(4));
```