A 3-PHASE Z-SOURCE INVERTER DRIVEN BY A NOVEL HYBRID SWITCHING ALGORITHM

Submitted for the degree of

MASTERS

IN ELECTRICAL ENGINEERING

BY

Jean-Claude Malengret

Electrical Engineering
University of Cape Town
Cape Town

JUNE 2007

DECLARATION

I hereby declare that I am the sole author of this thesis. I authorize the University of Cape to lend this thesis to other institutions or individuals for the purpose of scholarly research.

ACKNOWLEDGMENTS

Many thanks to supervisor Martin Braae and my father Michel Malengret for there advise and guidance. Many thanks to MLT Drives for funding all equipment used to build and test this inverter. Thank you to all the engineers at MLT drives for their advice and experience.

Abstract

A 3-phase Z-source inverter has been researched, designed, simulated, built and tested. The purpose of the inverter is to deliver 3-phase 400 VAC from a DC supply that can vary over a range of 20 to 70 Vdc. This is done with a Z-source inverter topology which is a single conversion method with no additional DC to DC boost converter. A novel DSP control algorithm allows the inverter to achieve the following:

- Run Space Vector Pulse Width Modulation (SV-PWM) for maximum DC bus voltage utilization while boosting the DC bus during zero space vector states using shoot through.
- Seamless transition between modulation control and modulation / shoot through control.
- Optimised efficiency and DC bus utilisation using Hybrid Space Vector Boost Pulse Width Modulation (HSVB PWM) which is unique to this dissertation.

Such a system is particularly suited to fuel cell and particularly wind turbine applications where the DC bus voltage is varies over a wide range resulting in the need for a DC to DC buck/boost to regulate the DC bus to maintain a steady 3-phase sinusoidal output. A further application could be for general purpose 3-phase inverter capable of operating on different DC standard bus voltages (e.g. 24, 36, 48 VDC).

The benefits of a Z-source topology for the above purposes are a reduction in high power semi-conductor components (e.g. power MOSFET). There is also a reduction in switching losses and inherent shoot through protection. Furthermore, the inverter is more robust in the sense that it is not vulnerable to

spurious shoot through, which could be disastrous in the case of a traditional voltage fed inverter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATIONS/ABBREVIATIONS

PWM - Pulse Width Modulation.

SVPWM – Space Vector Pulse Width Modulation.

HSVB PWM – Hybrid Space Vector Boost Pulse Width Modulation.

IGBT – Insulated-Gate Bipolar Transistor

MOSFET - Metal–Oxide–Semiconductor Field-Effect Transistor

Chapter 1
INTRODUCTION

1.1 Basic System overview

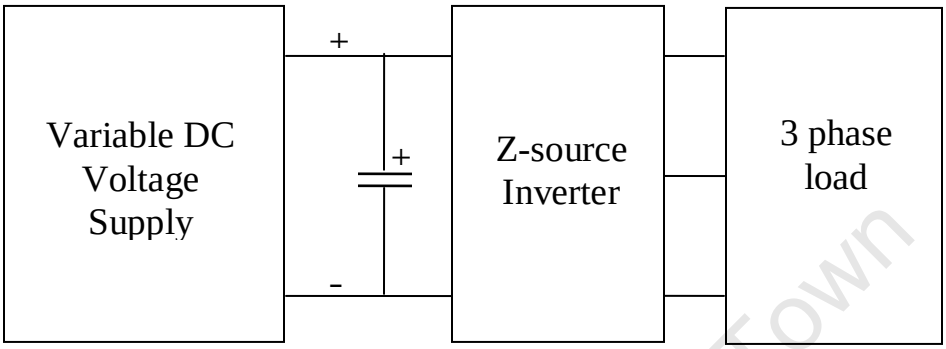


Fig. 1.1: Basic System overview

A varying voltage, which is decoupled by a capacitor bank, is used as the DC input to a z-source inverter. The z-source inverter uses its capability to boost the input voltage while producing 50 Hz, 3-phase AC at constant or controllable amplitude.

1.2 The Inverter

The inverter is capable of the following:

- Run Space Vector Pulse Width Modulation (SV PWM) for maximum DC bus voltage utilization while boosting the DC bus during zero space vector states using shoot through.
- Seamless swap over between full modulation control and modulation / shoot through control.
- Optimised efficiency and DC bus utilisation using Hybrid Space Vector Boost Pulse Width Modulation (HSVB PWM) which is unique to this dissertation.

Conventional 3-phase inverters can do the last two of these but cannot boost the DC bus. They require an additional DC to DC buck/boost converter to regulate the DC bus as shown in fig 1.5.

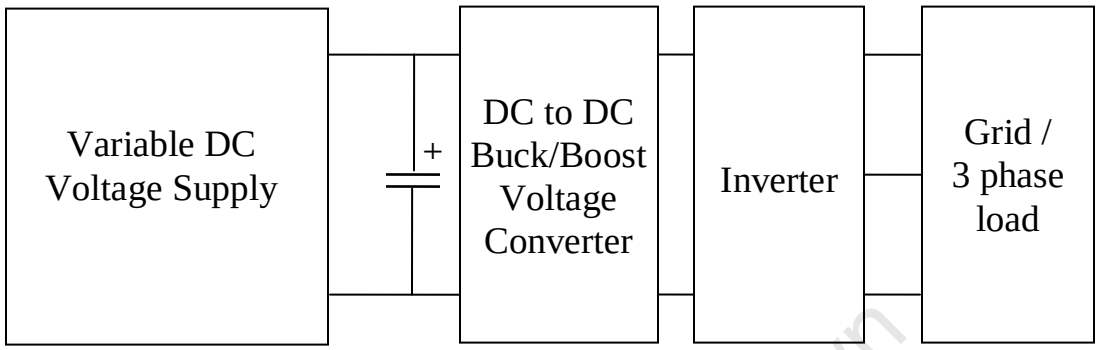


Fig. 1.2: Basic System overview with DC to DC buck/boost design

The problem with this system is that it needs a 2 stage energy conversion process. A 2 stage conversion process means more parts, less efficiency and hence a greater cost [3].

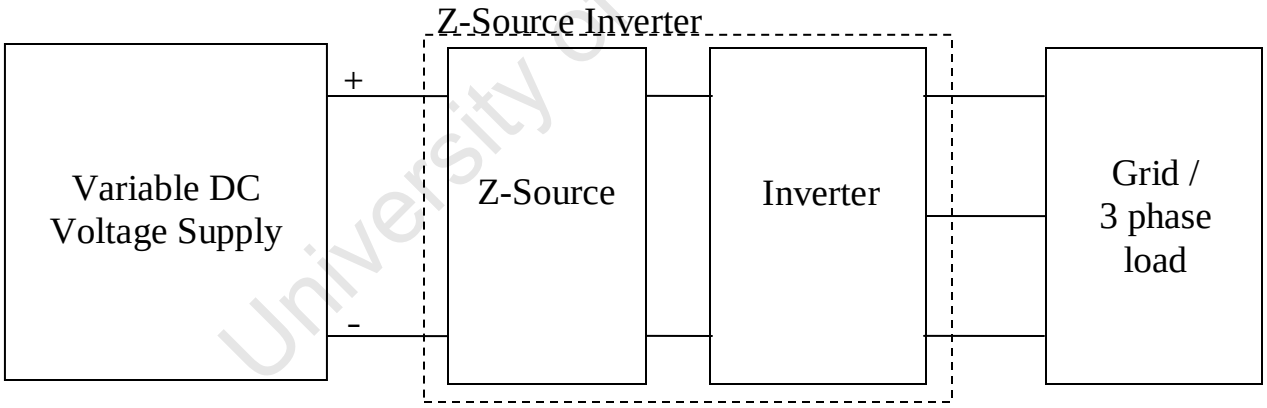


Fig. 1.3: Basic System overview with Z-source inverter design

Fig. 1.3 shows how the Z-source inverter is essentially a standard 3 phase inverter with an impedance source known as the Z-source.

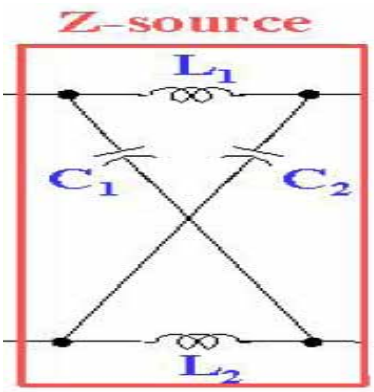


Fig. 1.4: Z-Source impedance network topology

The impedance network is used in conjunction with a modified switching scheme, hence allowing a voltage boost without an additional DC to DC converter. There are only passive components in this DC boost topology and therefore there are no significant additional switching losses from the Z-source side. Although the system can be broken down into a Z-source component and a 3 phase inverter they cannot function independently with the proposed switching algorithms. The combination is thought of as a single conversion process.

1.4 **Structure of Thesis**

A novel DSP control algorithm is to be developed and implemented that allows a Z-Source 3-phase inverter to achieve the following:

- Run Space Vector Pulse Width Modulation (SV PWM) for maximum DC bus voltage utilization while boosting the DC bus during zero space vector states using shoot through.
- Seamless swap over between full modulation control and modulation / shoot through control.
- Optimised efficiency and DC bus utilisation using Hybrid Space Vector Boost Pulse Width Modulation (HSVB PWM) which is unique to this dissertation.

To begin with, the basic 3-phase inverter and its various switching algorithms are introduced and discussed. These are detailed in appendix A (Sine triangular PWM) and appendix B (Space Vector PWM).

Once SVPWM has been achieved in simulation (Appendix D) a modified SVPWM scheme is developed and coded for simulation (Appendix E). Here the full boost capabilities of the Z-source inverter are achieved in simulation. This is discussed in Chapter 3.

Next a working prototype is designed, built and tested. The prototype consists of a physical topology, control instruments and DSP using embedded C code. First standard SVPWM is achieved, then modified SVPWM and finally a voltage locking hybrid algorithm is introduced and implemented. Results are obtained and discussed in Chapter 4.

1.5 Review of Related Literature

Literature on Z-source inverters started with its realization by Fang Zheng Peng in April 2003 with his paper [3]. He was the first to propose the addition of the Z-source impedance network on the DC side of a 3 phase inverter bridge. This impedance source combined with a modified switching scheme allowed for a single stage DC boost and DC to AC conversion. He then went on to produce in partnership another paper comparing traditional voltage source inverters with the z-source inverter [2]. This paper focused on applications for fuel cell powered vehicles specifically.

Jin-Woo Jung, Min Dai, Ali Keyhani in [13] and [1] discuss the characteristics and go on to further approve its use in fuel cell applications. [13] and [1] also explains application using modified space vector PWM (SVPWM) switching techniques. Applications using space vector algorithms have become widely used with the advance in low cost microprocessors capable of running these algorithms at the necessary speeds. The benefits of these SVPWM techniques are presented in [4]. A detailed implementation of traditional SVPWM is investigated in [10].

This master's thesis will combine the knowledge from the various sources to design, simulate and build a 3-phase z-source inverter using a novel HSVB PWM switching scheme. There is currently no literature documenting the specifics of control switching for a Z-source inverter. The basic theory behind the addition of shoot through is documented in [1] and [13]. There is no literature covering the idea or implementation of the proposed Hybrid Space Vector Boost Pulse Width Modulation (HSVB PWM) Technique. The idea is developed, explained, simulated and implemented in a fully functional prototype and all details are discussed and presented.

Chapter 2

Proposed modification for SVPWM algorithm to implement variable DC regulation capabilities on a 3 phase bridge

In this chapter the proposed modified SVPWM algorithm is introduced and discussed. In existing literature there no open source code on the implementation SVPWM with controlled shoot through. To arrive at the proposed algorithm a thorough understanding of the evolution of 3-phase PWM is necessary. For this reason a very relevant overview of related theory investigated and documented by the author is detailed in Appendix A and B.

2.1 Overview and Background of 3-phase inverter topology and operation using sine triangular PWM generation

An investigation into the fundamentals of traditional 3-phase inverter topology and operation is detailed and discussed in Appendix A. Section 1 discusses the topology and circuit operation of the 3-phase bridge. Section 2 and 3 investigates the limitations of the traditional voltage and current source 3-phase inverters. Section 4 describes in detail the conventional switching algorithm that generates sine-triangular carrier based PWM generation and its resultant output.

2.2 Overview and Background of the SVPWM switching algorithm

With the advance in microprocessors more complex and processor intensive algorithms became viable for PWM switching techniques. Space vector PWM (SVPWM) is one such algorithm. SVPWM offers less harmonics distortion on the output voltages and a more efficient use of the supply voltage [1]. Appendix B outlines a detailed investigation into SVPWM implementation using a 3-phase bridge. This chapter will add to the foundation of modern SVPWM theory detailed in Appendix B and is crucial to the understanding of the proposed modification.

2.3 The Z-source inverter topology

Fig. 2.1 shows the total system configuration of a Z-Source inverter connected to a 3-phase load. It has the same configuration as the traditional inverter with an impedance network on the input known as the Z-source. The impedance network consists of two capacitors and two inductors where $L_1 = L_2 = L$ and $C_1 = C_2 = C$. As a result the impedance network is symmetrical and therefore [2]:

$$v_{L1} = v_{L2} = v_L$$
$$V_{C1} = V_{C2} = V_C \qquad (10)$$

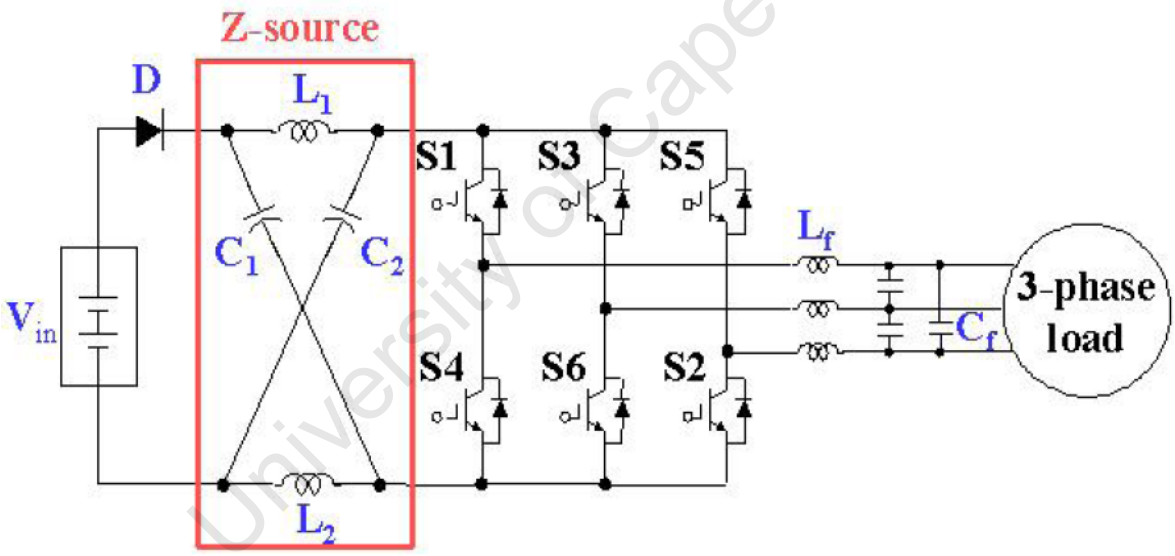


Fig. 2.1: Total system configuration with Z-Source inverter [1].

2.4 Z-source Equivalent Circuits and Equations

In traditional switching schemes for traditional inverters a shoot through condition is avoided at all costs. If both switches in one phase leg are on simultaneously a shoot through occurs as the DC bus is shorted through the two switches. This will in most cases result in significant damage to the inverter switches and possibly other components such as switch drivers and

controllers. Shoot through in a Z-Source inverter is not avoided but used to one's advantage, so as to boost the average DC input voltage to the 3-phase inverter bridge.

Shoot through states can be added to the discussed SVPWM switching scheme to boost the DC input voltage by a desired factor as well as generating the necessary PWM output for generation of a sinusoid [1], [2], [3]. Fig. 2.2a and 2.2b show the equivalent circuit during a shoot through state and a non-shoot through state respectively. During a shoot through interval we have:

$$\begin{aligned} v_{L1} &= V_{C1} \text{ Û } v_L = V_C \\ v_f &= 2V_{C1} = 2V_C \\ v_i &= 0 \end{aligned} \qquad (11)$$

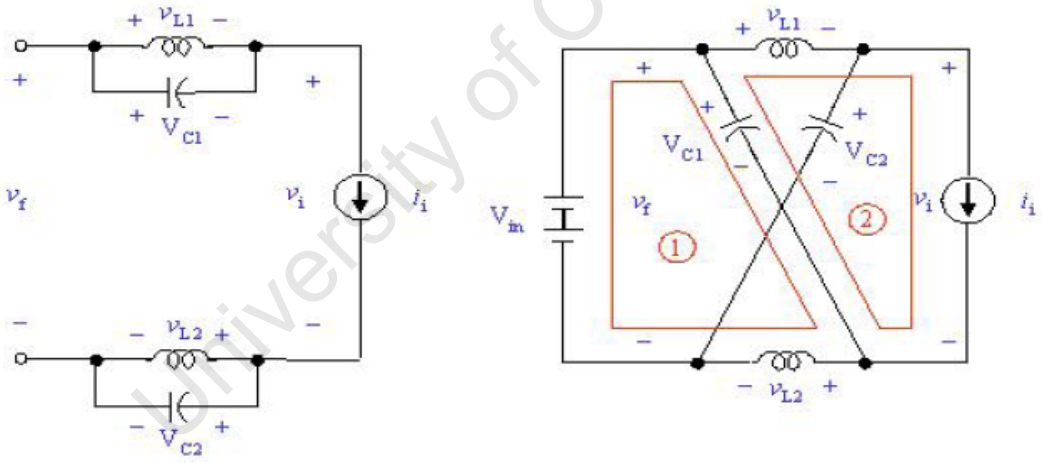


Fig. 2.2: Equivalent circuit of the Z-Source converter. (a) In the shoot through state. (b) In a non-shoot through state [1].

During the non-shoot through interval shown in fig 2.2b we have [1], [2]:

$$\begin{aligned} v_L &= V_{in} - V_C \\ v_i &= V_C - V_L = 2V_C - V_{in} \end{aligned} \qquad (12)$$

V_{in} is the variable input DC voltage. V_i is the voltage seen on the input of the 3-phase bridge, which has two different states (shoot through and non-shoot through). The switching interval T_z is composed of $T_a + T_b = T_z$. Ta is the total shoot through interval within T_z and Tb is the total non-shoot through interval within T_z. The average voltage over the inductors is zero over one switching cycle [1], [2], [7]. From this and equations (11) and (12) we can derive the following equations:

$$V_{L1} = \overline{v_{L1}} = \frac{1}{T_z}\int_0^{T_z} v_{L1}\,dt = \frac{T_a \times V_C + T_b \times (V_{in} - V_C)}{T_z} = 0$$

$$\Rightarrow V_C = \frac{T_b}{T_b - T_a} V_{in} = \frac{1 - \frac{T_a}{T_z}}{1 - 2 \times \frac{T_a}{T_z}} V_{in} \qquad (13)$$

Similarly an equation for V_i, the average DC-link voltage (input voltage across the 3-phase bridge) can be derived as follows [1], [2]:

$$V_i = \overline{v_i} = \frac{1}{T_z}\int_0^{T_z} v_i\,dt = \frac{T_a \times 0 + T_b \times (2V_C - V_{in})}{T_z} = \frac{T_b}{T_b - T_a} V_{in} = V_C \qquad (14)$$

As can be seen from (14), the average DC-link voltage is equal to the capacitor voltage $(V_i = V_C)$. This means we can use the capacitor voltage measurement to regulate the average DC-link voltage. Now using (12) and (13) we can derive an expression for $\hat{v}_i$ (the peak DC-link voltage) [1], [2], [13]:

$$\hat{v}_i = 2V_C - V_{in} = \frac{T_z}{T_b - T_a} V_{in} = B \times V_{in} \qquad (15)$$

B is the boost factor and can be expressed as [1], [2], [13]:

$$B = \frac{T_z}{T_b - T_a} = \frac{1}{1 - 2 \times \frac{T_a}{T_z}} \; ^3 \; 1 \qquad (16)$$

B is always greater or equal to 1. As the shoot through time T_a increases the boost factor B increases and the voltage boost the Z-source impedance network offers increases. Using (15), $\hat{v}_{AC}$ (a peak AC phase voltage of the inverter output) can be expressed as follows [1], [2], [13]:

$$\hat{v}_{AC} = M \times \frac{\hat{v}_i}{2} \qquad (17)$$

M is the modulation index as discussed in the traditional 3-phase V-source inverter. Using (15) and (17) we can then write:

$$\hat{v}_{AC} = M \times B \times V_{in} = B_B \times V_{in} \qquad (18)$$

From (17) it can be seen that the peak phase voltage $\hat{v}_{AC}$ of the inverter output depends on both the boost factor B and the modulation index M. A buck-boost factor B_B from $0 \; ® \; ¥$ can then be chosen to either step up or step down the output voltage where

$$B_B = M \times B$$

It should be noted that the shoot through time portion $T_a \; \mu \; B$ has a maximum. The available shoot through time is limited by the SVPWM zero-state time discussed in Appendix B [2]. Using more than just the available zero state time will affect the active state vectors of the SVPWM switching cycle and work against the benefits that SVPWM brings. In fact most PWM schemes will not be affected by the added shoot through states as the modified switching scheme still delivers the equivalent zero voltage to the load terminals [7]. The available zero state vector time is determined by the modulation index M [1], [13].

Finally using (10), (13) and (16) we can express the capacitor voltage in terms of the fuel cell output voltage as [2]:

$$V_{C1} = V_{C2} = V_C = \frac{1 - \frac{T_a}{T_z}}{1 - 2 \times \frac{T_a}{T_z}} V_{in} \qquad (19)$$

2.5 Z-source Modified SVPWM Switching Scheme

As stated previously, the shoot through modification can be used with most PWM switching schemes using various methods [1], [14]. In this dissertation only the modification on SVPWM is analyzed. SVPWM was chosen because of its various advantages such as less harmonic distortion and more efficient use of the supply voltage [1].

In Appendix B the generation of a SVPWM switching scheme for a 3-phase inverter was discussed. Using the derived formulas time quantities T_1, T_2 and T_0 were calculated. T_1 and T_2 were the quantities of time spent in two adjacent active vector states while T_0 was the total time spent in one of the zero vector states (111 and 000). In this section the addition of shoot through states will be discussed. T_a as previously discussed is the total shoot through time. A new shoot though period $T = T_a/3$ is calculated. The value T will be the time duration for one of 3 shoot through intervals within a switching period T_z.

In fig. 2.3 we can see the total switching pattern result for all 6 sectors of the modified SVPWM switching scheme. Notice how there are three time intervals of length T in which both the upper and lower switch in one leg is on causing a shoot through. For example, in sector 1 we have the 1st T length shoot through interval by turning on switch S1 T seconds early thereby allowing upper S1 and lower S4 to be on simultaneously for a time period T. The 2nd T length shoot through arises from lower S6 be held on an additional T seconds resulting in a T second shoot

through when upper S3 turns on. The 3rd last T second shoot through is created by holding lower S2 on an additional 2T seconds to cause a T second overlap with upper S5. The following T_z switching cycle is has the mirror configuration of the last T_z switching cycle. The switching patterns for the next 5 sectors are different because the 3 reference signals are being swapped between comparators as shown in Appendix B, table B2. In sector 2 for example the S1, S4 leg is driven by the same signals used to drive the S3, S6 leg in sector 1.

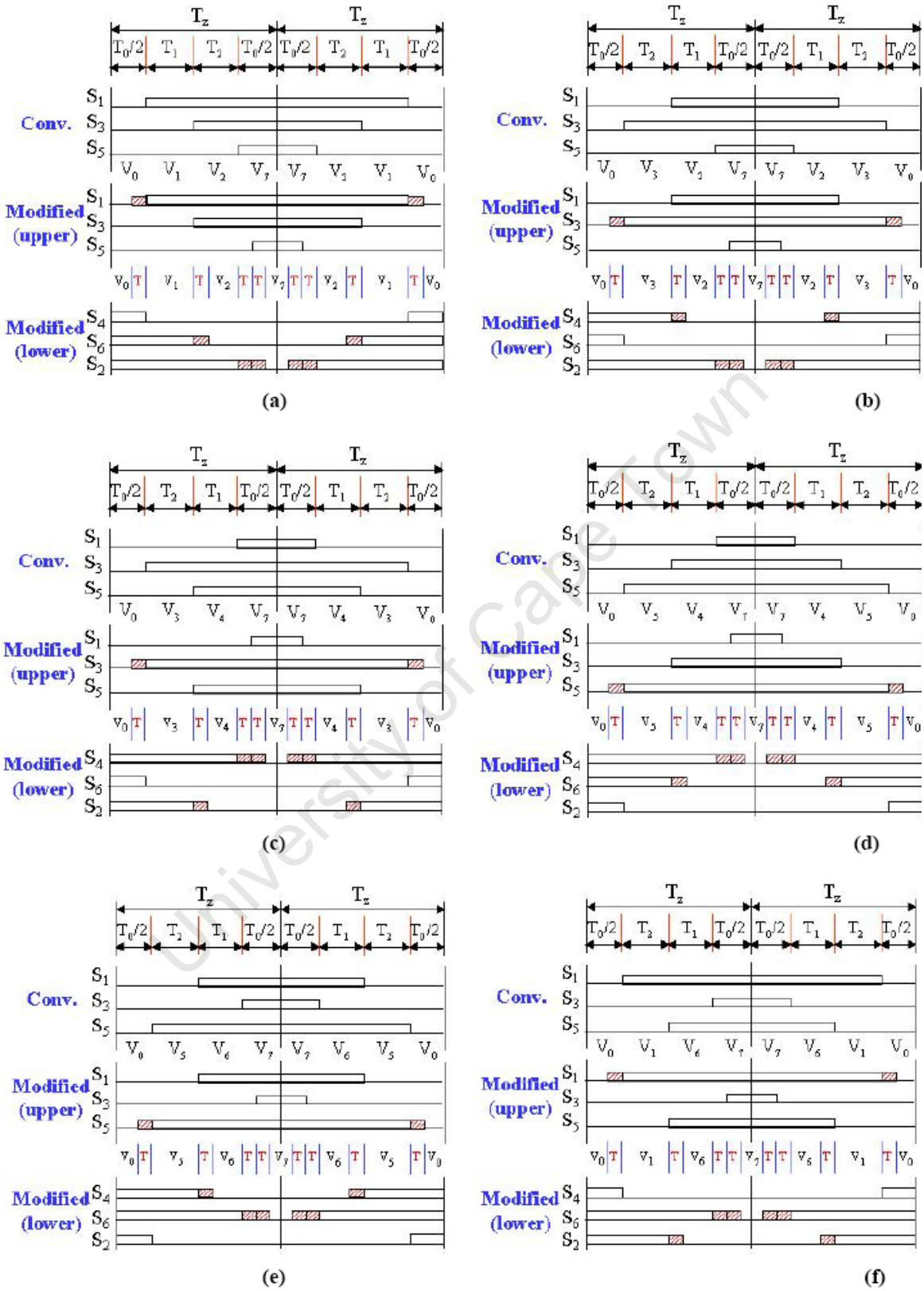


Fig. 2.3. Modified SVPWM implementation. (a) Sector 1. (b) Sector 2. (c) Sector 3. (d) Sector 4. (e) Sector 5. (f) Sector 6. [1]

Sector	Upper (S1, S3, S5)	Lower (S4, S6, S2)
1	S1 = T1 + T2 + T0 /2 + T S3 = T2 + T0 /2 S5 = T0 /2 - T	S4 = T0 /2 S6 = T1 + T0 /2 + T S2 = T1 + T2 + T0 /2 + 2T
2	S1 = T1 + T0 /2 S3 = T1 + T2 + T0 /2 + T S5 = T0 /2 - T	S4 = T2 + T0 /2 + T S6 = T0 /2 S2 = T1 + T2 + T0 /2 + 2T
3	S1 = T0 /2 - T S3 = T1 + T2 + T0 /2 + T S5 = T2 + T0 /2	S4 = T1 + T2 + T0 /2 + 2T S6 = T0 /2 S2 = T1 + T0 /2 + T
4	S1 = T0 /2 - T S3 = T1 + T0 /2 S5 = T1 + T2 + T0 /2 + T	S4 = T1 + T2 + T0 /2 + 2T S6 = T2 + T0 /2 + T S2 = T0 /2
5	S1 = T2 + T0 /2 S3 = T0 /2 - T S5 = T1 + T2 + T0 /2 + T	S4 = T1 + T0 /2 + T S6 = T1 + T2 + T0 /2 + 2T S2 = T0 /2
6	S1 = T1 + T2 + T0 /2 + T S3 = T0 /2 - T S5 = T1 + T0 /2	S4 = T0 /2 S6 = T1 + T2 + T0 /2 + 2T S2 = T2 + T0 /2 + T

Table 2.1: Switching time duration of each switch in each sector [1],[3].

Table 2.1 shows the switching time duration of each switch in each sector. It is the tabulated form of the switching configuration portrayed in fig. 2.3. Fig 2.4 below is included to help avoid confusion about which vector states time intervals T1 and T2 represent. The reference vector approximated by the SVPWM scheme moves from sector 1 to sector 2, T2 associated with the final state of sector 1 becomes the first state of sector 2. Again when moving into sector 3, T1 again becomes the time interval associated with the starting vector state and T2 associated with the finishing vector state.

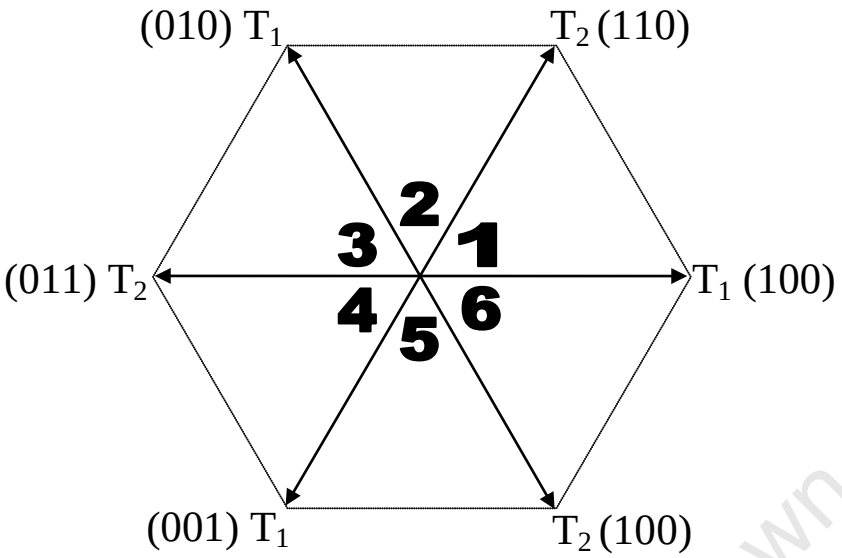


Fig. 2.4: Association of T1 and T2 with the six active vector states.

Fig 2.3 can be used to derive equations for the necessary reference signals that will be fed to the comparators to drive the top and bottom switches. In chapter 4 the conventional SVPWM scheme used 3 reference signals and 3 comparators. The reference signals were swapped between the 3 comparators so that the 3 comparators saw 3 different control signals on their inputs constructed from 6 pieces of the original reference waveforms.

For the Z-source inverter implementation there are now 6 reference waveforms and 6 comparators. 3 reference waveforms are alternately fed to 3 comparators for the top switches and 3 fed to 3 more comparators to control the bottom switches. In the conventional SVPWM switching the top switches were always the compliment of the bottom switches except for the precautionary blanking time intervals between transitions. Now the top and bottom switches are driven independently.

The 6 reference waveforms are calculated as follows:

$$
\begin{aligned}
U_{cTop}(kT_s) &= \frac{T_0}{2} - T \\
U_{bTop}(kT_s) &= \frac{T_0}{2} + T_1 = U_{cTop}(kT_s) + T_2 \\
U_{aTop}(kT_s) &= \frac{T_0}{2} + T_1 + T_2 + T = U_{bTop}(kT_s) + T_1 + T
\end{aligned}
\tag{20}
$$

$$
\begin{aligned}
U_{cBot}(kT_s) &= \frac{T_0}{2} \\
U_{bBot}(kT_s) &= \frac{T_0}{2} + T_1 + T = U_{cBot}(kT_s) + T_1 + T \\
U_{aBot}(kT_s) &= \frac{T_0}{2} + T_1 + T_2 + 2T = U_{bBot}(kT_s) + T_2 + T
\end{aligned}
\tag{21}
$$

T1 and T2 are the time spent in each of the two adjacent active vector states. T1 is the time associated with the starting active state while T2 is associated with the finishing active state. In sector 1, the starting active state associated with T1 would be (100) and the finishing active state (110), would be associated with T2. T0 is the total time spent in the zero vector state in one switching cycle Tz. T = Ta/3 where Ta is the total shoot through time Tz.

Using (7) in Appendix B, the value of the 3 reference signals for SVPWM were calculated using T_1, T_2 and T_0. Using (20), the value of the 3 reference signals fed to the 3 comparators whose outputs drive the top switches can be calculated. Similarly using (21) reference signals associated with the bottom switches can be calculated. Note that if T = T_a/3 = 0 the modified switching scheme becomes the conventional SVPWM switching scheme.

Inductor and capacitor requirements for the Z-source are discussed in Appendix G

Chapter 3

SIMULATION IN SIMPLORER 7:

3.1. Simplorer 7

“Simplorer® is multi-domain, system simulation software for the design of high-performance electromechanical systems commonly found in the automotive, aerospace/defense, and industrial automation industries. With a wide range of modeling techniques, statistical analysis capability and adherence to IEEE standards,

Simplorer along with MATLAB simulink are the most widely used packages for power electronic simulations. Simplorer was used in simulations because of its VHDL support. VHDL in short is faster. The simulations use VHDL to compute switching algorithms on the fly. Some relevant technical issues are detailed in Appendix C.

3.2. Simplorer 7 simulation: Open Loop 3-phase SVPWM for standalone 3-phase bridge (detailed in Appendix D)

AIM:

Using the given circuit libraries and VHDL language a working model for accurate simulation is built for implementing unmodified SVPWM. The aim of this simulation is to verify both existing theory and the effectiveness of the simulator for high frequency complex PWM generation using an amalgamation of the component models and the integrated VHDL language. The VHDL language will be used to program the SVPWM algorithm while the circuit simulation uses the standard component libraries. Specific expected outcomes are:

- Develop and test an algorithm to simulate the correct SVPWM generation of switching signals for each phase.

- Realise increased efficiency in DC bus utilisation compared to sine triangular SVPWM.

METHOD SUMMARY:

Appendix D details all work done to achieve the above aims and derive conclusions. In brief the following was done:

- The following topology was realised from traditional 3 phase inverter theory

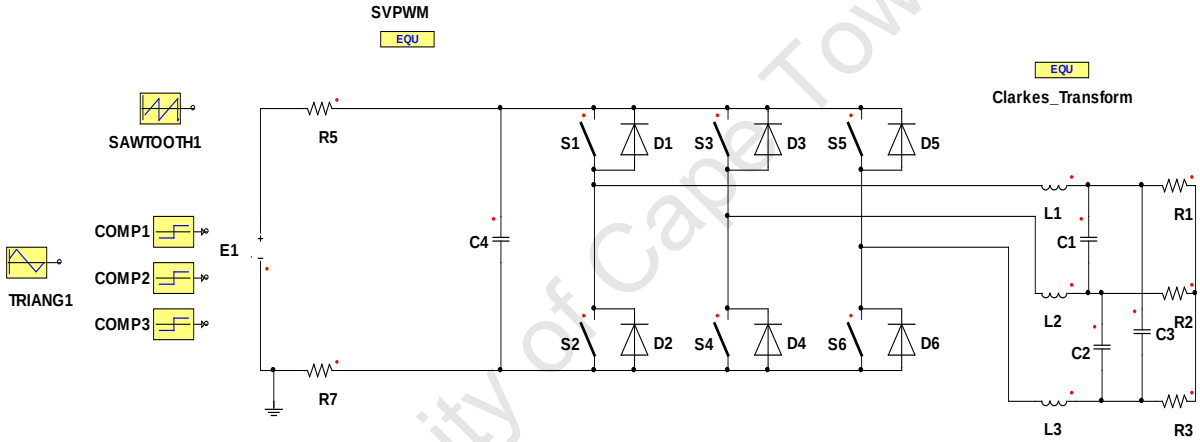


Fig. D1: A 3 phase inverter with filter and load

- VHDL code was written to simulate the driving signals of the 6 switches. The code is detailed and explained in Appendix D. The outputs of the VHDL algorithm and their function in the system can be summarised in Fig D2.

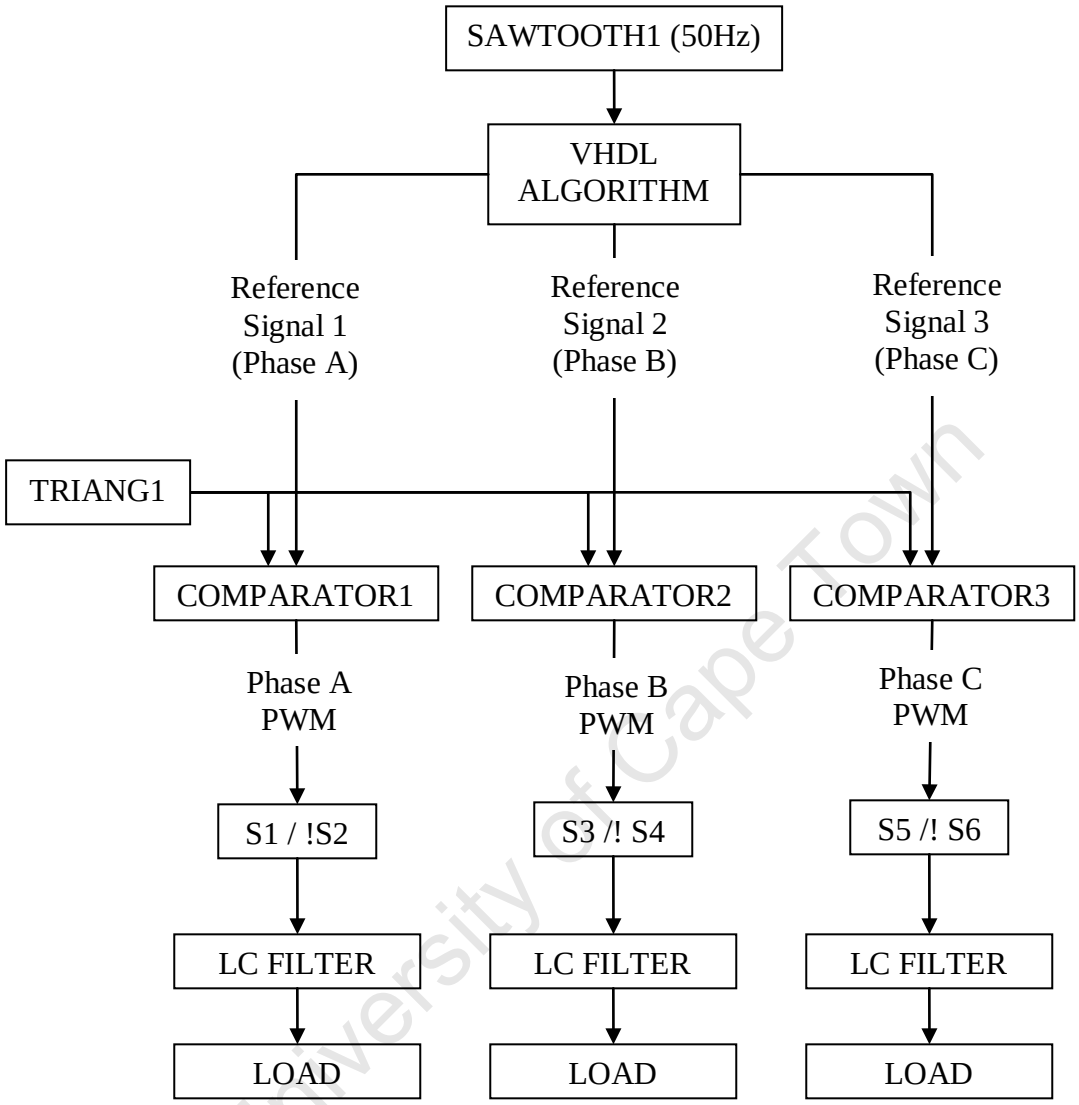
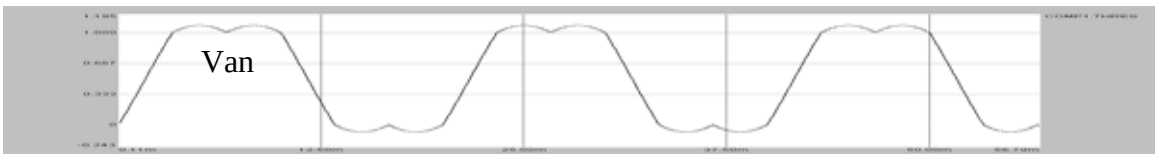


Fig. D2: Simulation System Process Flow Diagram

RESULTS SUMMARY:

The following control signals were produced from the VHDL code:



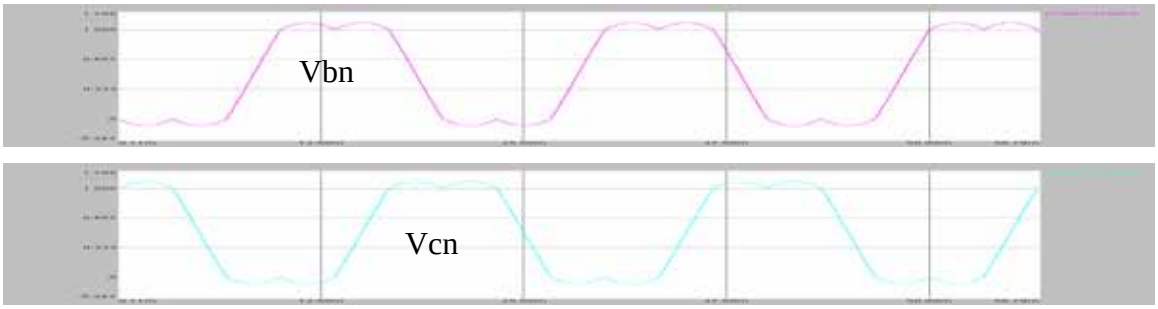


Fig. D4: The 3 control signals that are seen on each of the comparator inputs

Switching signals generated from comparator drivers using the above control signals versus the 5 kHz triangular reference signal are shown in fig. D6. Switching signals shown in fig. D6 resulted in the following 3 phase voltage output over the 3 phase load:

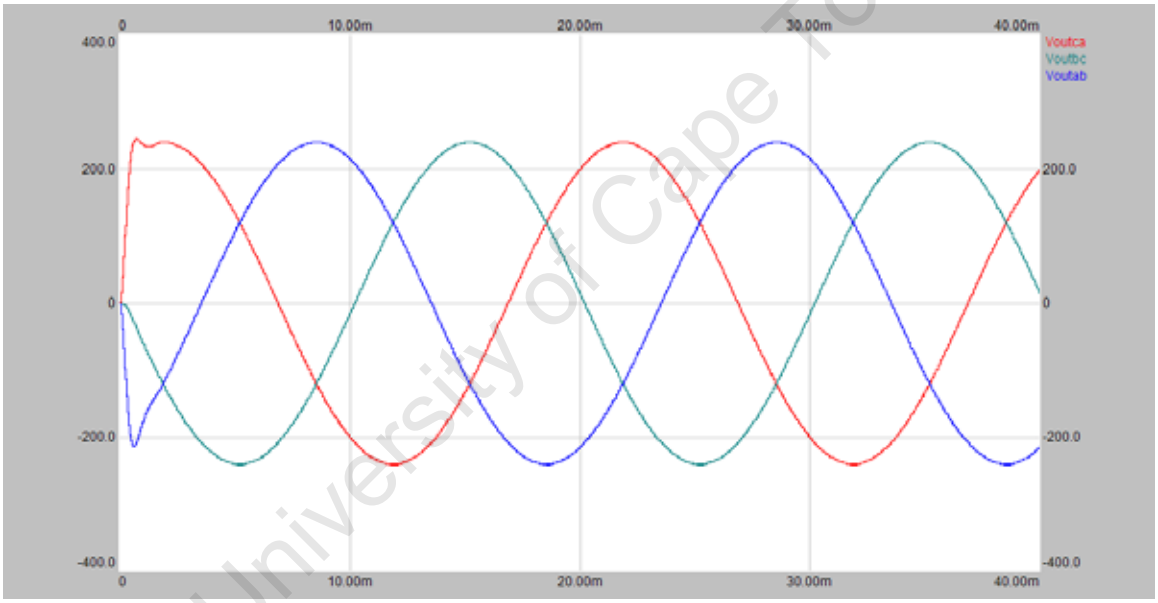


Fig. D8: LC filter output of 3 sinusoids 120˚ out of phase (0.8 modulation)

CONCLUSION:

- Control signals were correctly generated as discussed in the SVPWM theory investigation details in Appendix B.
- The PWM signals are filtered through the first order LC filter and the result is three sinusoids 120 ˚ out of phase as shown in fig D8.The initial distortion

is caused by the capacitive and inductive elements of the system reaching steady state.

- The source was a steady 300V DC. With a modulation ratio of 0.8, the peak of the line to line sinusoidal output is 240V which is 80% of the DC bus. This is shown in fig D8. With a modulation ratio of 1, the peak of the line to line sinusoidal output is 300V which is 100% of the DC bus.
- Conventional Sine-Triangle PWM generation utilizes ½ the DC bus for the line to neutral output yielding $\sqrt{3}/2$ the DC bus for the line to line peak voltage. This is 86.6%. SVPWM allows us to use 100% (+15%) of the DC bus utilization at full modulation. The theory presented in Appendix B has been verified by simulation.

3.4. Simplorer 7 simulation: Open Loop 3-phase Modified SVPWM for the Z-source inverter (detailed in Appendix E)

AIM:

The following simulation uses a modified SVPWM technique to take advantage of the z-source impedance network and boost the DC bus voltage while generating a sinusoidal output with all the benefits and characteristics of conventional SVPWM. Primary aims are:

- Add z-source to traditional topology
- Develop and implement a modified SVPWM switching algorithm in VHDL to drive the 6 switches.
- Boost the DC bus using the added shoot though intervals in the switching scheme.

METHOD SUMMARY:

Appendix E details all work done to achieve the above aims and derive conclusions. In brief the following was done:

The following topology was realised from the chapter 2 theory investigation.

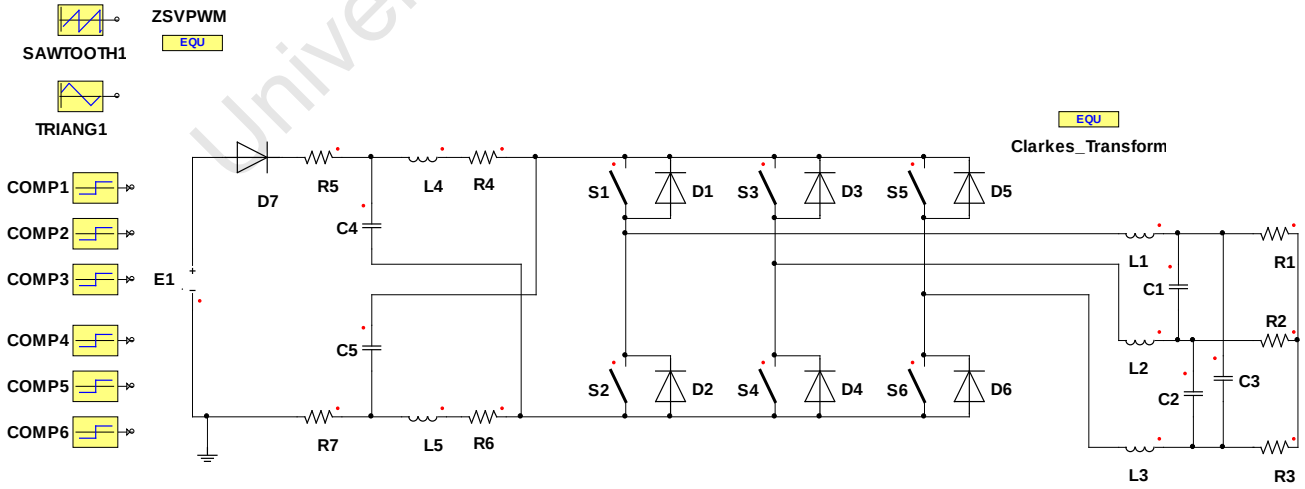


Fig. E1: A 3 phase z-source inverter with filter and load

· VHDL code was written to simulate the driving signals of the 6 switches. The code is detailed and explained in Appendix E. The outputs of the VHDL algorithm and their function in the system can be summarised in Fig E2.

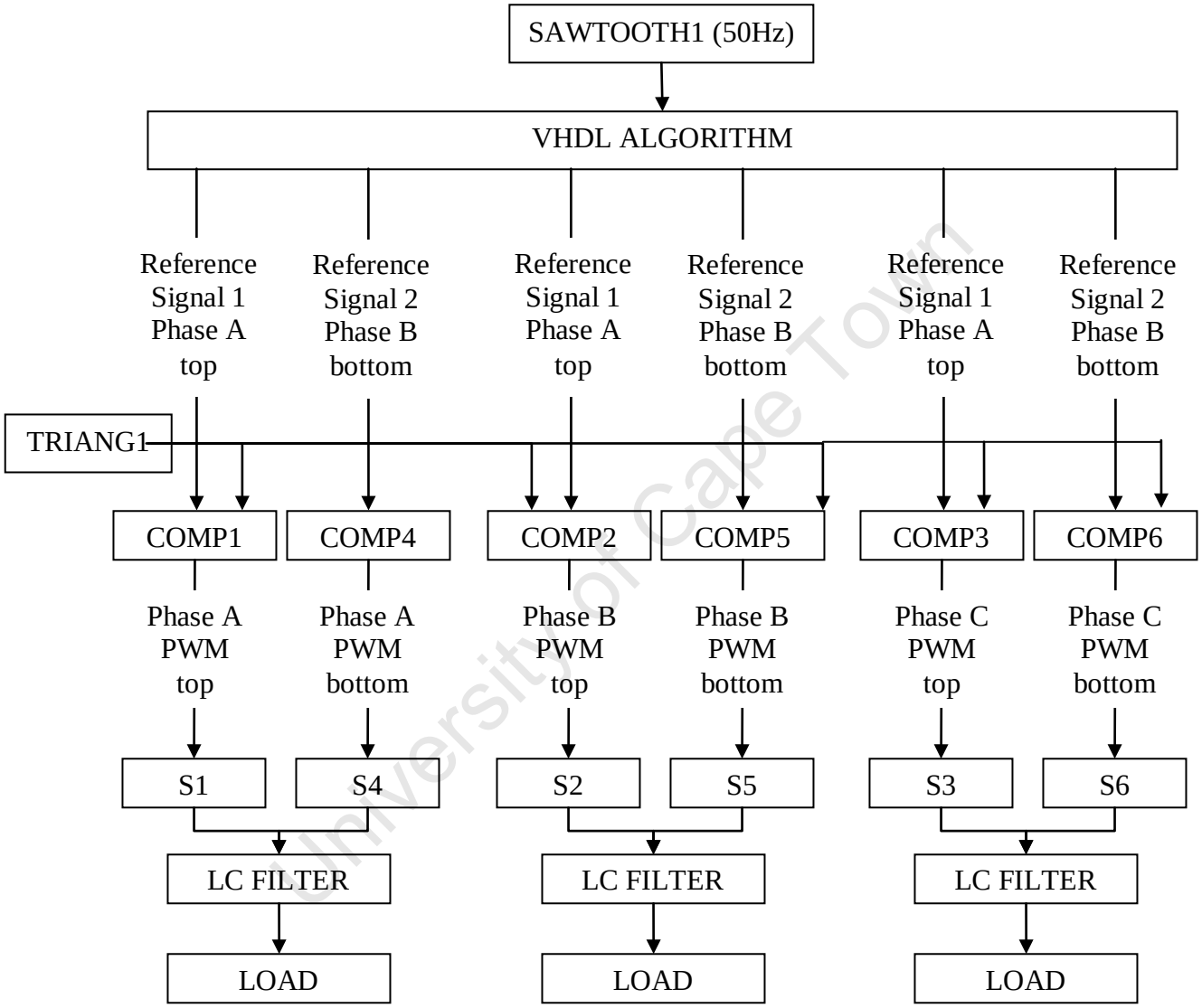


Fig. E2: Simulation System Process Flow Diagram of z-source simulation

RESULTS SUMMARY:

The control signals for the 6 comparators produced from the VHDL algorithm are detailed in fig E7 in Appendix E. Fig E8 shows all control signals on one plot.

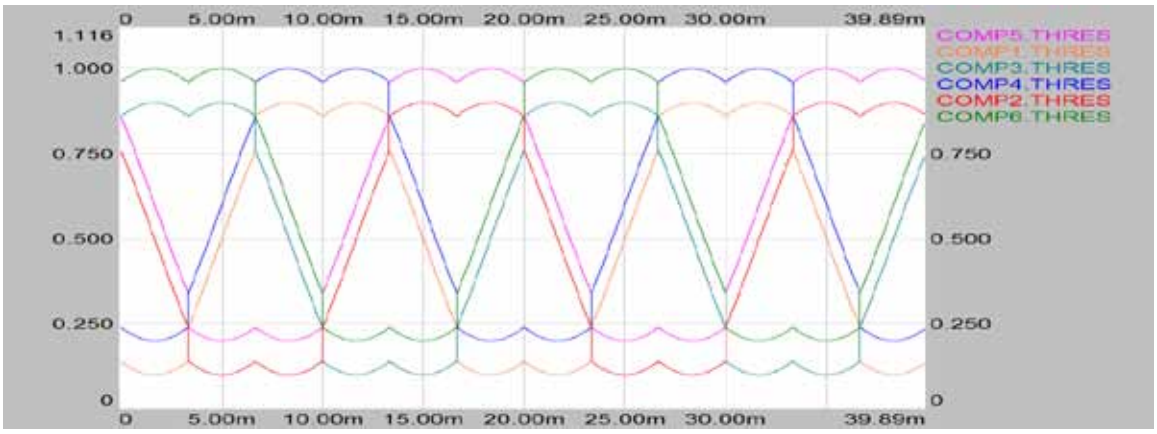


Fig. E8: The 6 generated control signals seen on each comparator input.

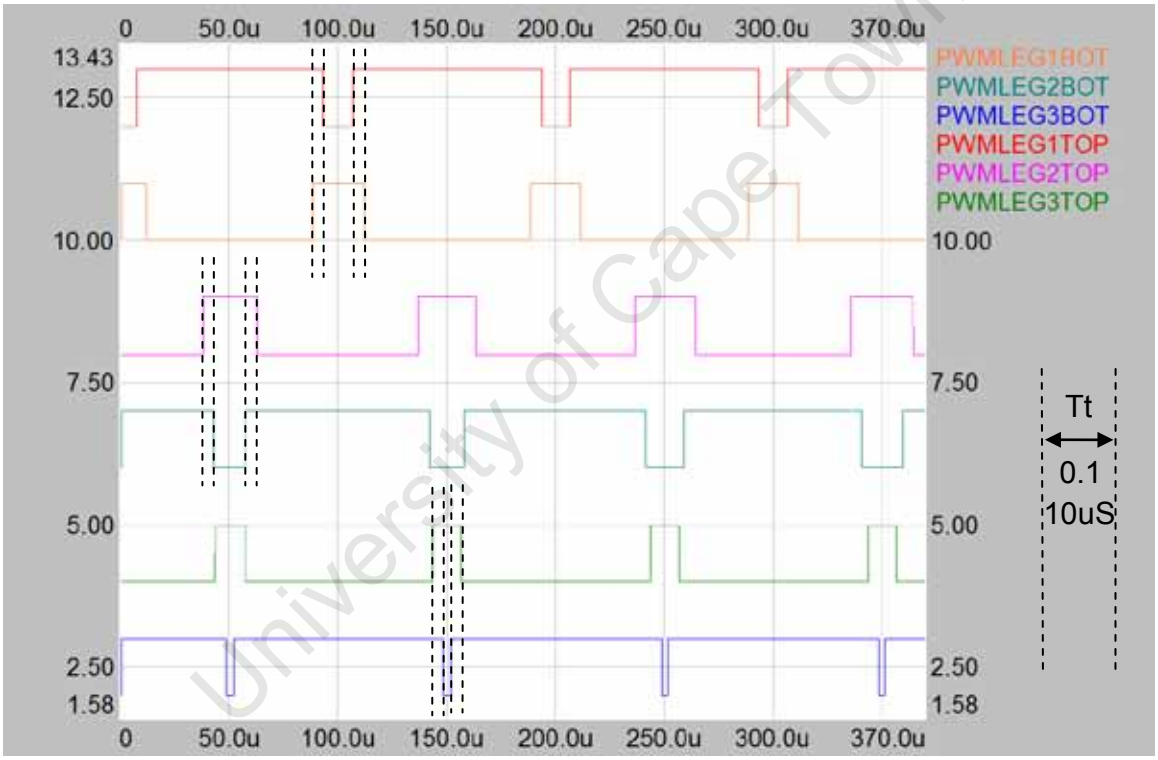


Fig. E10: The 6 PWM signals produced for sector 1.

The resultant switching signals for a shoot through value of 30% or 0.3 are show in Fig E10 where Tt = Ta/3 = 0.1 or 10%. Overlaps have been indicated resulting in 6 shoot through periods of 10% of the switching cycle each. The DC bus seen by the 3 phase bridge is shown in fig E13. The filtered 3 phase output of the 3 phase bridge is shown in fig E11.

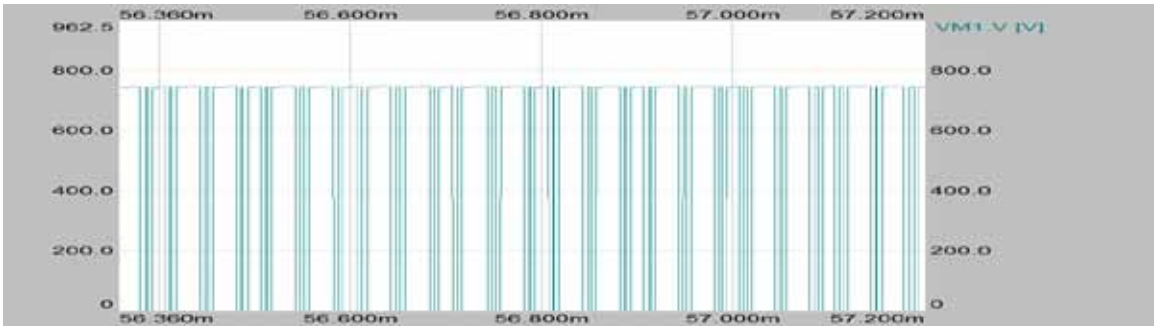


Fig. E13: High time resolution slim section of $\hat{v}_i$, steady state voltage waveform seen by 3 Phase Bridge. Peak voltage = 750V.

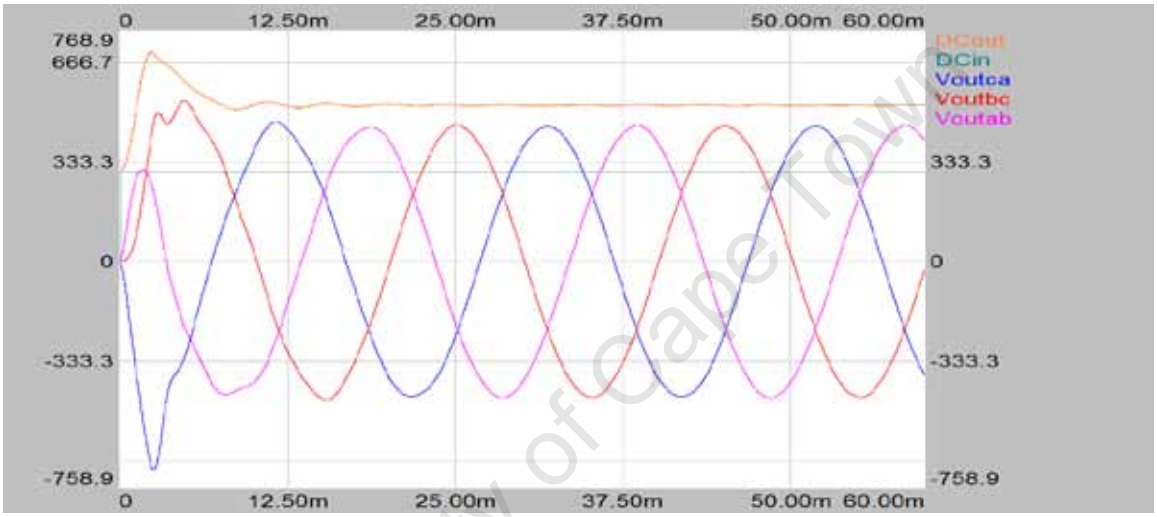


Fig. E11: LC filter output of 3 sinusoids 120˚ out of phase (Ta=0.3)

Conclusion (detailed discussion in Appendix E)

Using a 300V DC input a 3 phase sinusoidal output was generated. The sinusoids have a peak voltage of 455V using a shoot through ratio of Ta=0.3. With no shoot through and full modulation (M=1) a maximum peak voltage of 260V can be achieved. In this simulation we have boosted the output peak voltage by 57% above the best case scenario using conventional SVPWM. Simulation results are consistent with the presented theory in chapter 2. Single stage DC boost and sinusoid generation has been achieved.

3.5. Simulation: Investigation into theoretical voltage boost characteristics (detailed in Appendix F)

AIM:

The following simulation focuses on the relationship between the shoot through period length, the input and output voltage over the impedance source and the final 3 phase sinusoidal output.

RESULTS SUMMARY:

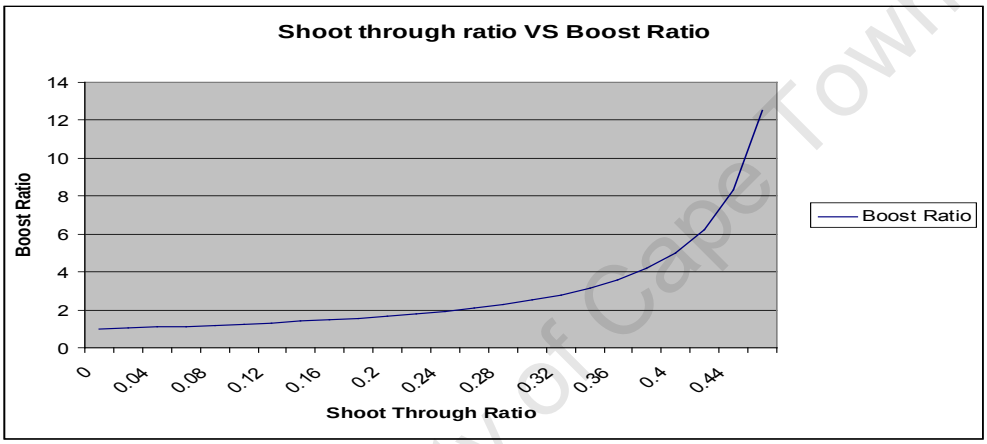


Fig. F1: Plot of Shoot through ratio VS Boost Ratio. Ta = 0:0.46.

Fig F1 along with fig F2 in Appendix F shows the boost characteristics exponential nature. Fig F4 shows how the output 3 phase line to line voltage also has an exponential despite the diminishing modulation ratio to compensate for shoot through time. Refer to Appendix F for detailed discussion.

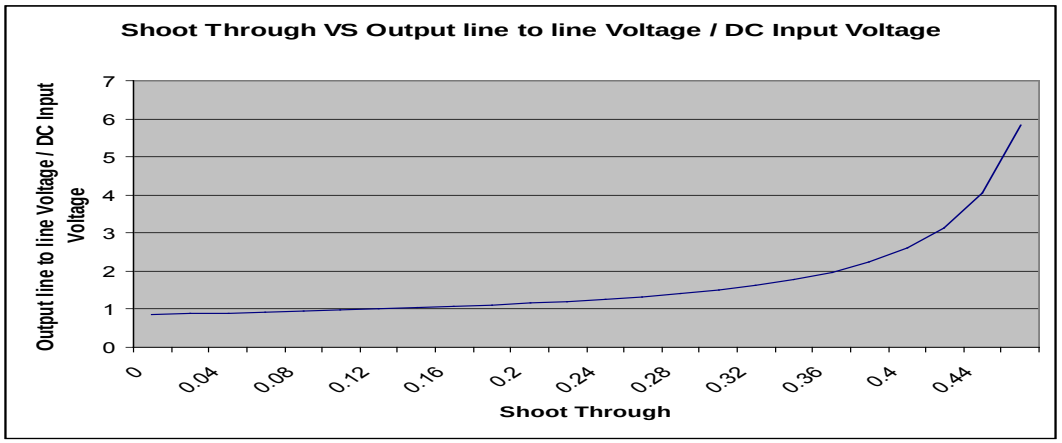


Fig. F4: Plot of Shoot through ratio VS $\hat{v}_{ACLL} : DCin$. Ta = 0:0.46.

Chapter 4

PROTOTYPE DEVELOPMENT:

4.1. Background

Prototypes were built by the author at MLT Drives CC in Kenilworth Cape Town. They were built over a period of 2 years using parts sponsored by MLT Drives. The prototypes were built to validate simulation results in the real world to ensure the subtle yet vital complexities of a real world implementation had been considered. A working prototype is the chief endeavour of this dissertation.

4.2. Step One: Building a standard 3-phase inverter with SVPWM

4.2.1. Prototype topology

Figure 4.1 is an overview of the various components of the standard 3-phase inverter prototype system built at MLT Drives. Figure 4.2 is a photographic overview of the actual components used. The system process can be summarised as follows:

- A PC with custom C# software written for control and monitoring of the prototype communicates with a DSP. The PC issues commands and takes readings from the DSP.
- The DSP takes commands and sends readings to the PC. Depending on control variables set by the PC software, the DSP produces 6 independent PWM outputs which are outputted 3 separate driver boards via an IO buffer and splitter board.
- The driver boards receive 2 PWM signals each and drive the upper and lower mosfets on their mosfet block.

· Each mosfet block comprises of 3 bus bars, positive, negative and mid. The upper set of mosfets which switch in parallel short the positive to the mid. The lower set short mid to negative.

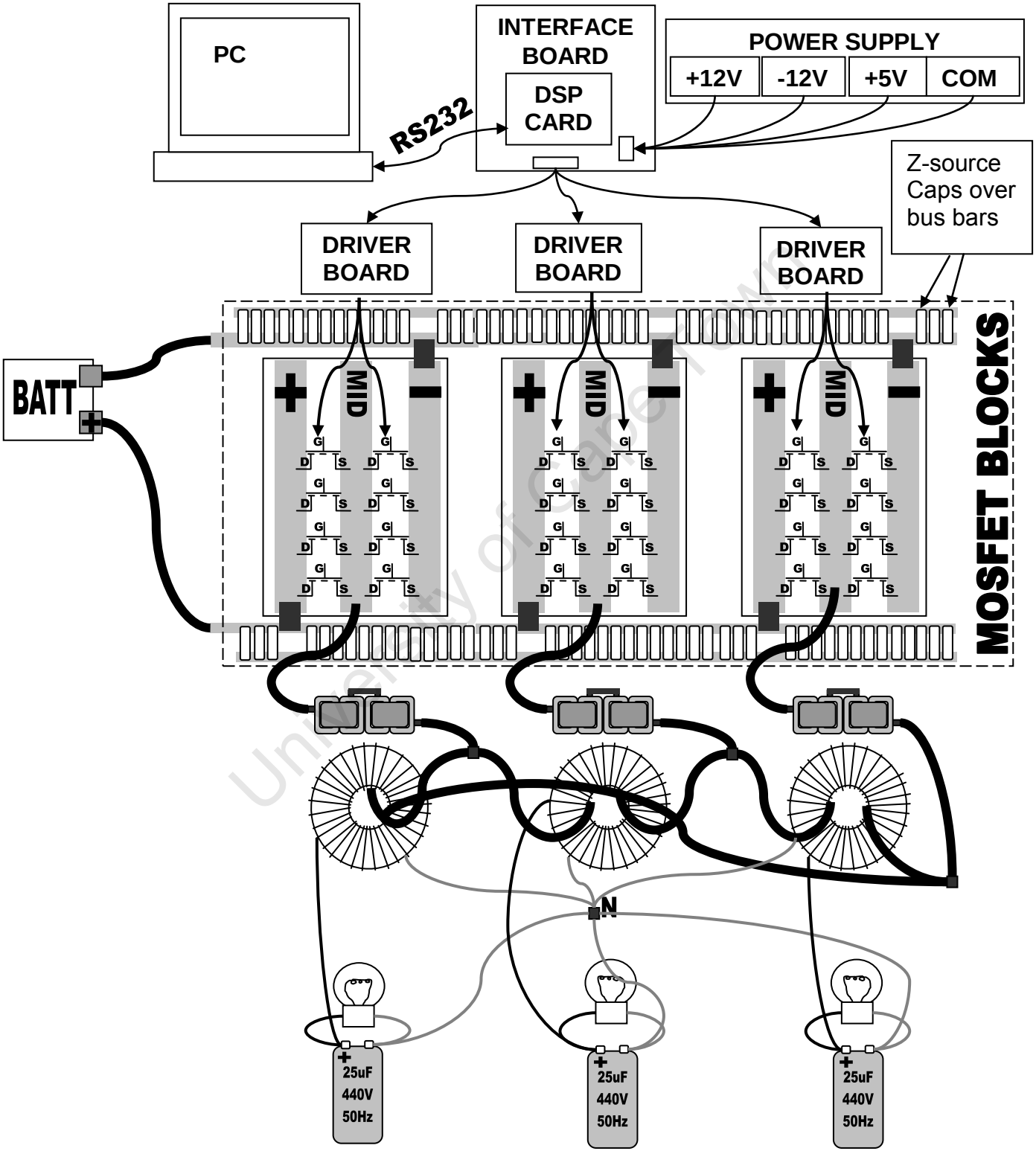


Fig. 4.1: Layout of standard 3-phase inverter prototype.

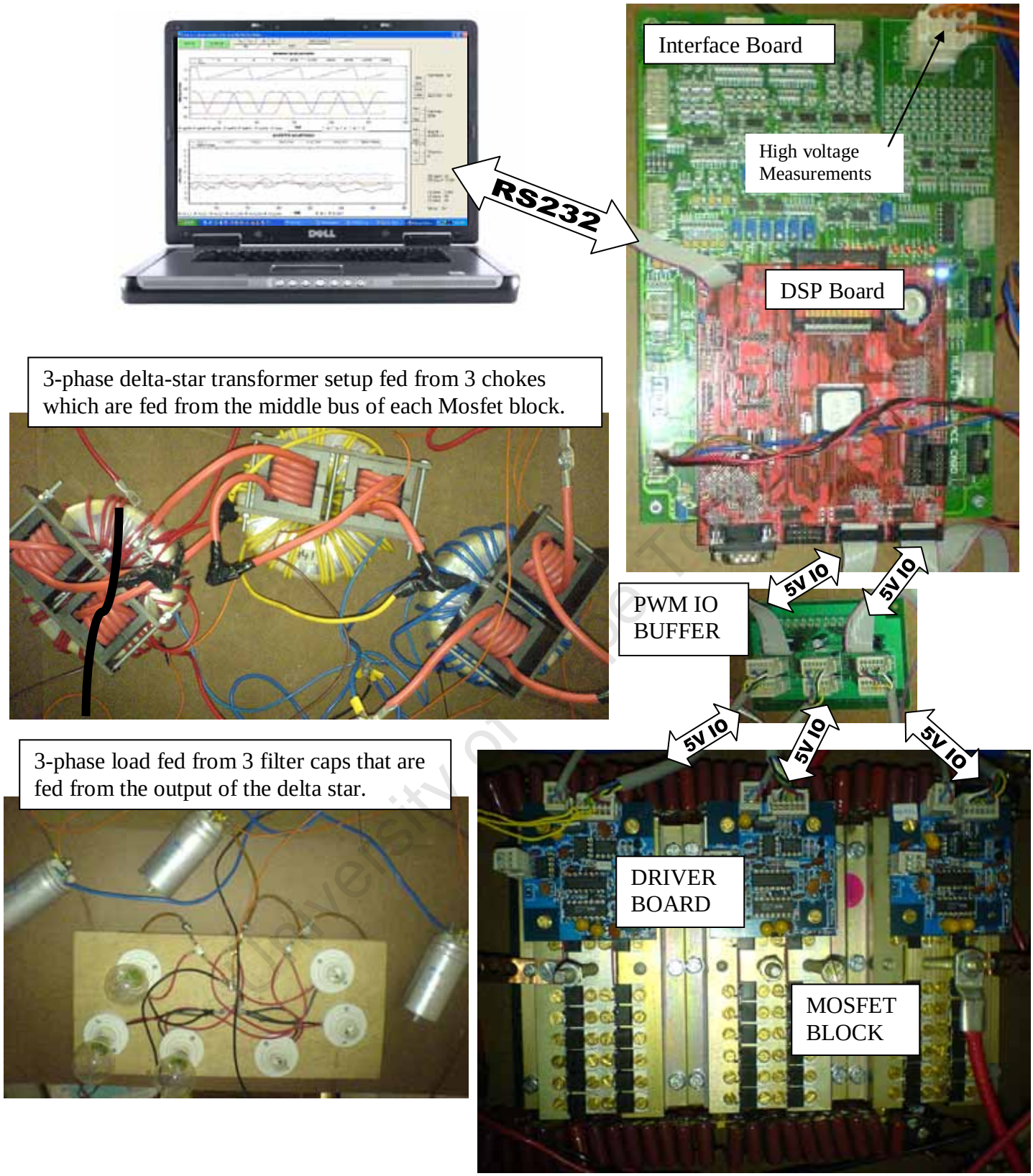


Fig. 4.2: Photo's of a standard 3-phase inverter prototype

- The mid bus will show the effective voltage as a result of the mosfet switching. The 3 PWM signals on the mid bus bars switching between positive and negative are fed to the primary side of the 3 phase delta star configuration via 3 chokes.
- The output on the secondary star is 3 sinusoidal voltages phase shifted by 120º and a neutral. The 3 600W loads are connected between each phase and neutral with AC caps in parallel. The chokes on the primary with the caps on the secondary form a low pass filter and extract the 50Hz fundamental while filtering out the switching frequency and generated harmonics.

4.2.2. Testing and Monitoring Equipment

As mentioned in the system setup in 4.2.1 control and monitoring was done through custom PC software written in C#. Design and development of this program was done at MLT drives over a period of 2 months. The design, the code and functionality is detailed in Appendix H. Fig. 4.3 shows the interface used to monitor and control the system.

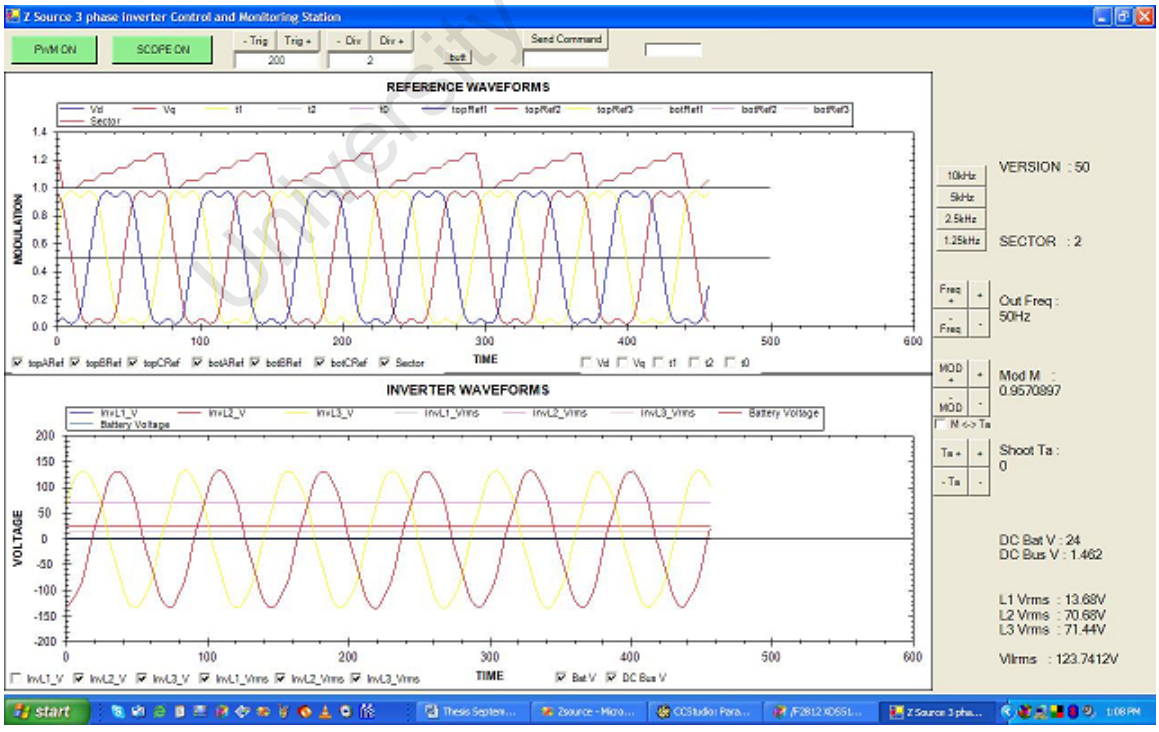


Fig. 4.3: PC control and monitoring interface written in C#

For the traditional inverter the PC software sets the following variables:

- PWM Switching frequency (2kHz – 20kHz)
- Output frequency (0-300Hz), was always run at 50Hz but was slowed down for detailed analysis of PWM.
- M, the modulation ratio ranging from 0 to 100% modulation
- Ta, blanking time between PWM transitions for shoot through protection.
- Scope settings for triggering and waveform resolutions.

The PC software monitors the following outputs:

- Sector number (1-6) of the hexagonal space vector plane
- The 3 reference wave forms produced from the DSP algorithm given a phase angle and modulation.
- Variables associated with space vector algorithm (Vd, Vq, t1, t2, t0).
- Battery Voltage.
- DS bus voltage.
- LN voltages for each line.
- rms LN voltages for each line.
- PWM outputs in batch downloads because of resolution limits

Scopes were used in conjunction with the PC software for higher frequency non-periodic outputs which the PC program could not manage on the fly. These outputs were:

- PWM outputs on the drain source of the mosfets and associated noise
- Real time PWM from the driver boards (as apposed to batch downloads).

4.2.3. SVPWM implementation

The algorithms used were written in C and compiled for the Texas instruments TMS2812 DSP at MLT drives CC by the author over a period of 2 months. A sample of the code is detailed in Appendix I. Appendix M details the decision to use this processor.

4.2.4 Results

Fig. 4.4 shows the reference signals generated by the DSP code when executed and displayed by the PC based software. Following the simulated work detailed in Appendix D, the reference wave forms are compared against an up down counter at a desired switching frequency to produce the 3 independent PWM for the top mosfets of each leg. Three other PWM signals mirroring the top mosfet signals are generated for the bottom mosfets. An instant of the PWM for sector 1 generated from the comparison of figure 4.4 reference waveforms with a 10kHz triangular waveform are shown in figure 4.5.

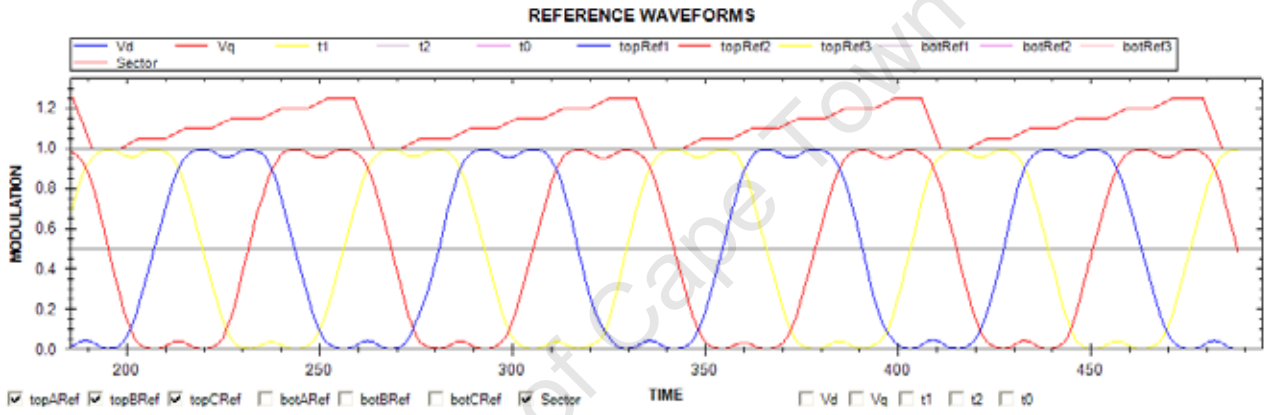


Fig. 4.4: PC monitoring of reference signals for standard SVPWM

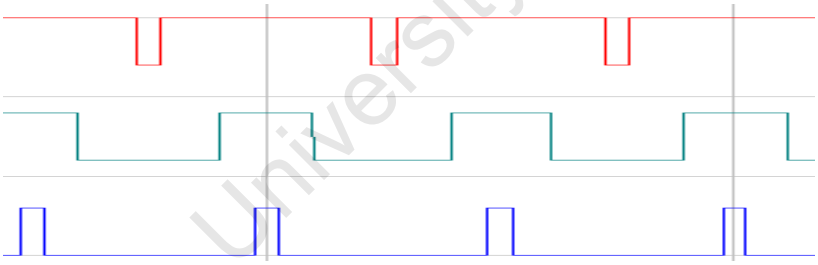


Fig. 4.5: Instant of sector 1 PWM for top switches taken from batch download.

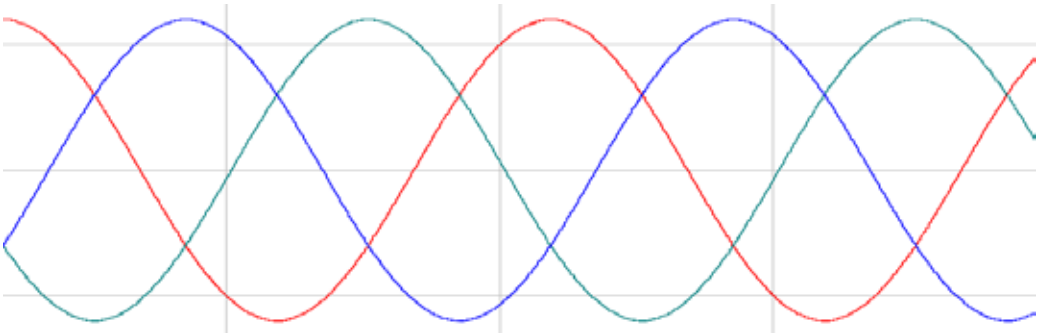


Fig. 4.6: 3 phase output voltage seen by the load taken from batch download.

Figure 4.6 shows the 3 phase output seen from the load.

4.2.4 Conclusion

Based on the input and output values the conclusion is that a 3 phase inverter using SVPWM switching has successfully been built and monitored with accordance to Appendix B theory and Appendix D simulations.

4.3. Step Two: Building a 3-phase Z-source inverter with modified SVPWM

4.3.1. Prototype configuration modifications

The Z-source inverter is essentially a modified standard 3-phase inverter so only the modifications relevant to the section 4.2 will be presented. The following modifications were made:

- An impedance network (z-source) was added on the DC bus side to utilise the shoot through created by the new modified switching scheme.
- A high frequency power diode was added between the battery positive and z-source positive input. The diode stops current from flowing back into the battery when the Z-source boosts the DC bus voltage up to 400% of the supplied battery voltage.
- A snubber cap was added between battery positive and battery negative directly onto the diode connection and negative Z-source input to cancel voltage spikes caused by the high frequency shoot through combined with the inductance of the length of wire from the battery. Failure to quell the voltage spike causes the mosfets to exceed their break down voltage and the results is unpredictable switching.
- Two inductors in conjunction with the 2 banks of caps were used to create the z-source impedance (values discussed in appendix G).
- The capacitor banks are soldered directly onto the bus bars to minimise stray inductance to avoid the same problem mentioned in the third point.

- The 2 inductor sets each consists of a large low frequency iron core inductor in series with a small high frequency ferried inductor. The iron core gives the inductive bulk while the ferried placed before the iron core suppresses the high frequency component that would cause high losses in the low frequency material iron core inductance. There is a significant high frequency component. As mentioned in Chapter 2 theory the shoot though is 6 times the switching frequency as there are 6 shoot through periods per cycle.
- The available driver boards had circuitry that detects when top and bottom mosfets of one leg are on at the same time or within a certain limit. If these thresholds are broken the top mosfets are prevented from turning on to prevent shoot through. This feature was disabled by modifying each board. The circuit diagrams for the driver boards can be found in Appendix J.

The schematic topology and related theory for the Z-source inverter is discussed and detailed in chapter 2. Figure 4.7 shows the modified system configuration.

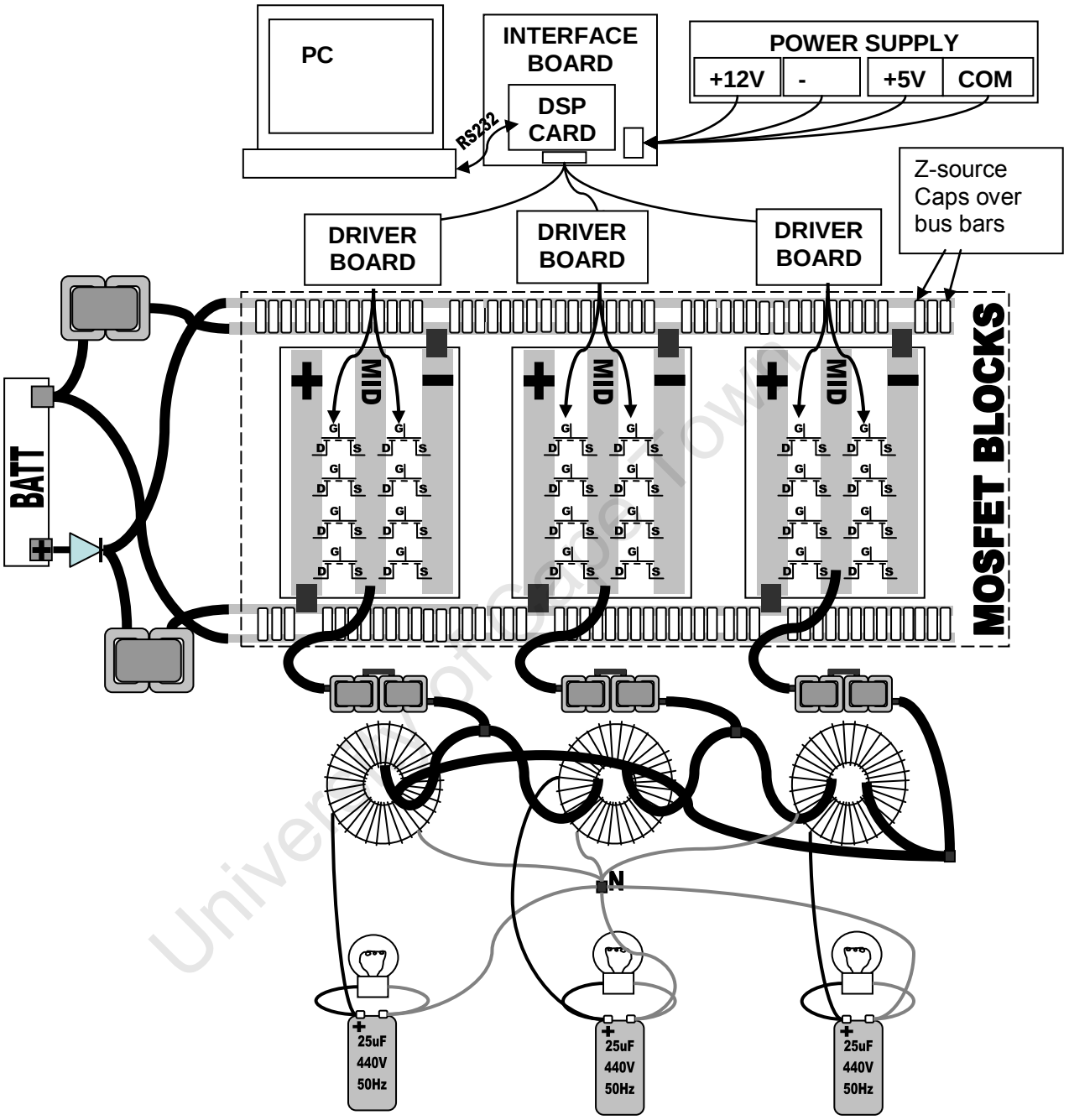


Fig. 4.7: Layout of 3-phase Z-source inverter prototype.

4.3.2. Testing and Monitoring Equipment

The same PC based software was used to control and monitor the modified system. Additional control and monitoring is added:

- One additional control variable is added to the software. A shoot through period Ta can now be varied on the fly.
- An option to lock Ta and M to maximise utilisation of the full switching cycle for shoot through and active space vector states. Refer to theory of chapter 2 and simulations of Appendix D for detailed explanation.
- Also 3 more reference waveforms need to be monitored making a total of 6 modified independent SVPWM reference waveforms.

4.3.3. Modified SVPWM implementation

The modified algorithms used were written in C and compiled for the Texas instruments TMS2812 DSP at MLT drives CC by the author over a period of 5 weeks. A sample of the code is detailed in Appendix I.

4.3.4 Results

Fig. 4.8 shows the reference signals generated by the DSP code when executed and displayed by the PC based software. These reference signals are generated with Ta=0.1 and Ma = 0.9.

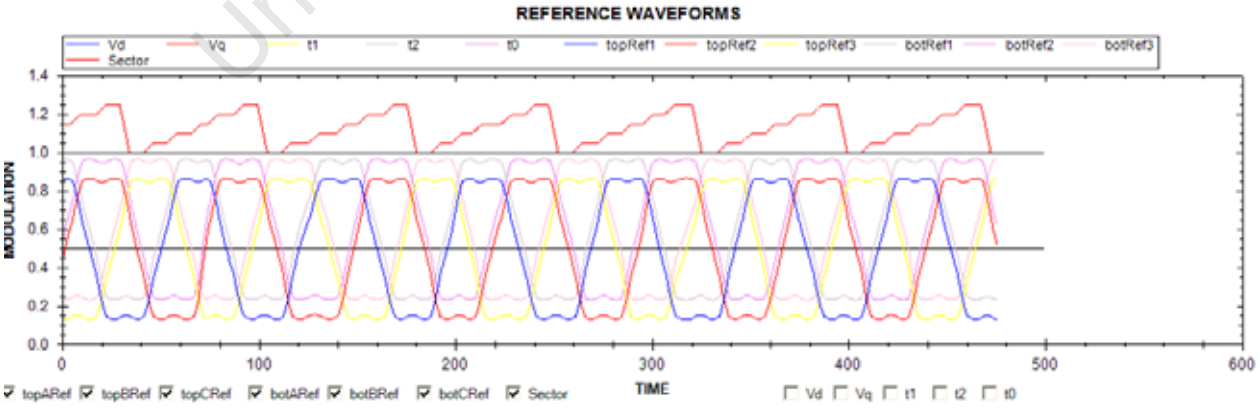


Fig. 4.8: PC monitoring of 6 independent reference signals for modified SVPWM.

Following the simulated work detailed in Appendix E, the reference wave forms are compared against an up down counter at a desired switching frequency. The result is 6 independent PWM for each of the 6 mosfets of the 3-phase bridge. An instant of the PWM for sector 1 phase A generated from the comparison of figure 4.8 reference waveforms with a 10 kHz up down counter (triangular waveform) are shown in figure 4.9. The top and bottom PWM signals are the top and bottom switches respectively. The middle PWM waveform is the addition of the two. The result is a '1' on overlap (shoot through) or '-1' when neither switch is on (dead time). The first image is with 0.5% dead time (no shoot through at all).

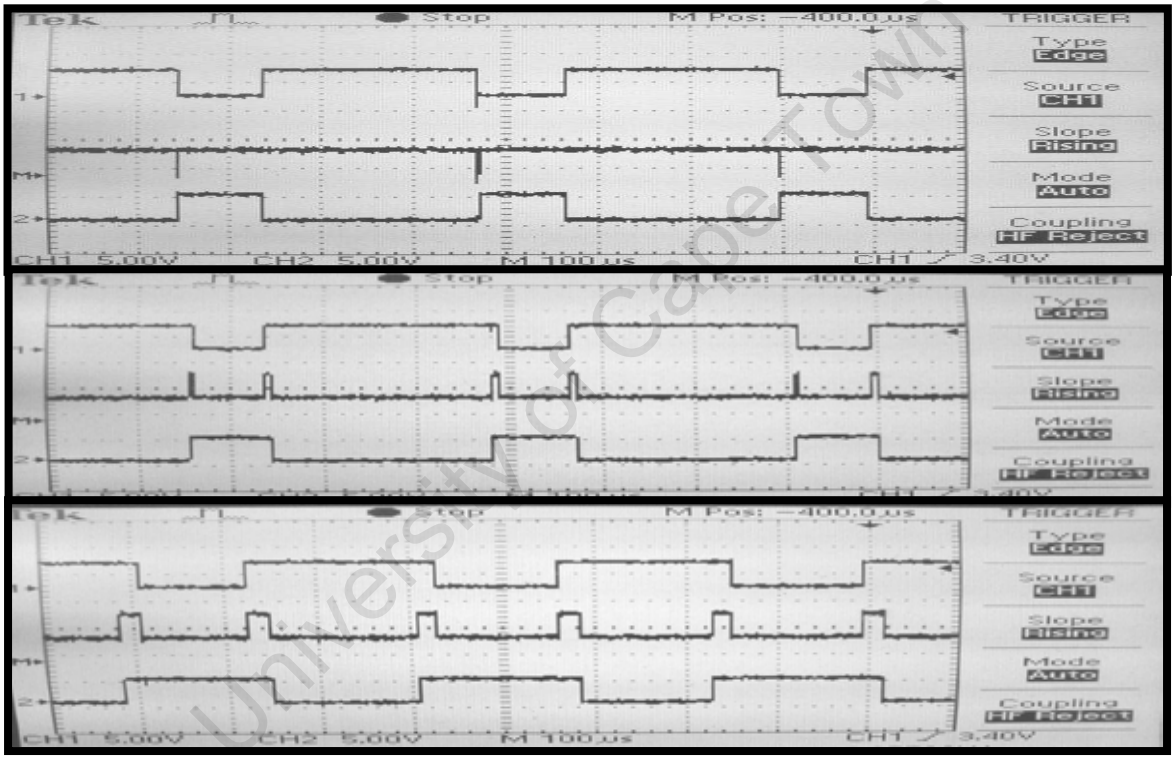


Fig. 4.9: Instant of sector 1 phase A PWM for top switches for Ta= -.5%, 10%, 40%

The second image is a result of 10% shoot through and the third 40% shoot through. The shoot though is divided equally over 3 phases so only a 3rd of the shoot though is seen on phase A. Fig 4.10 shows the DC bus at 30% shoot through. In the image the DC bus is pulled low by the short for 30% of a switching cycle. The ripple after each rising edge is a result of inductance between components, namely the larger electrolytic DC caps and bus bars. To reduce this specialised bus plates separated by a few mm could me manufactured into which

the Z-source caps are fixed directly onto one plate and through the top plate to the lower plate.

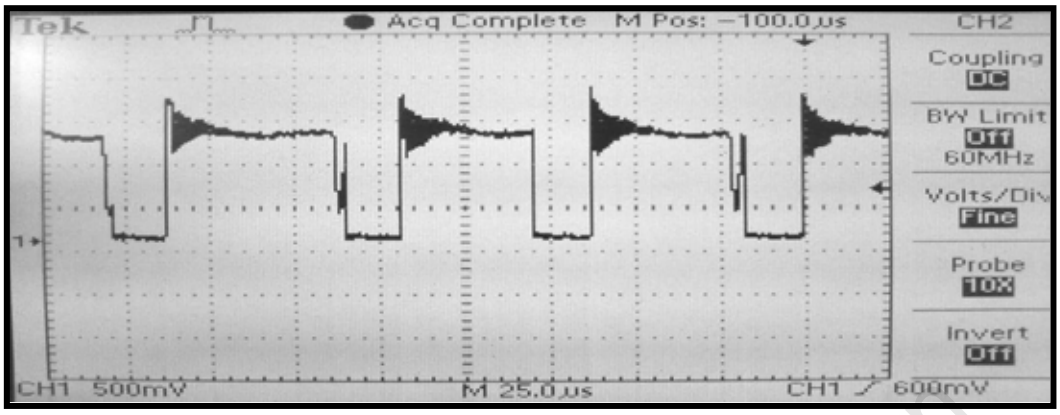


Fig. 4.10: DC bus over 3-phase Bridge at 30% shoots through

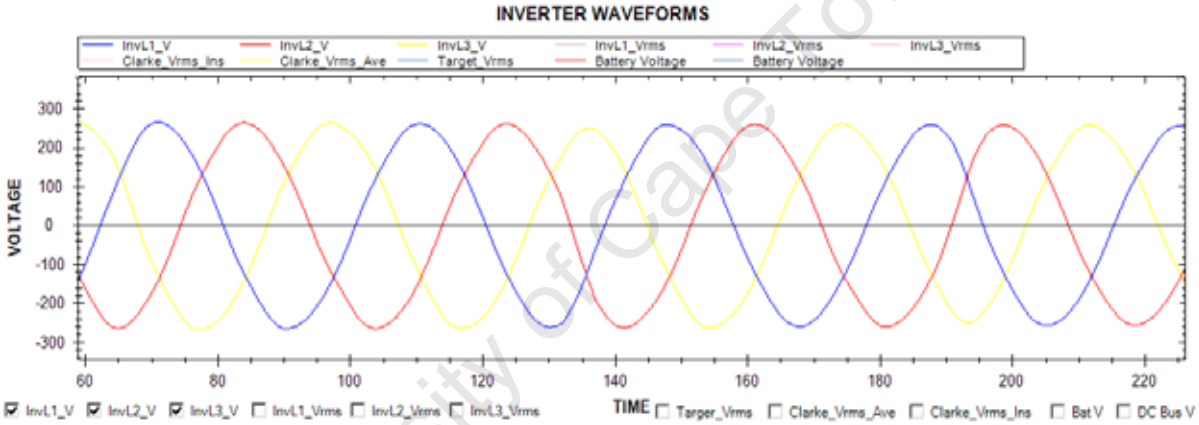


Fig 4.11: 230Vrms 3-phase output from 24V battery through 1:10 transformer.

Fig 4.11 shows a 230Vrms 3-phase output from 24V battery through 1:10 transformer. This image was taken via the custom PC monitoring and control program. Table 4.1 below details the setting and results.

	THEORETICAL	ACTUAL
Ta/Tz (% shoot through)	0.33	0.33
Tb/Tz = M	0.56	0.56
Vin	24	24
Average DC boost	1.970588235	1.970588235
C2 average Voltage	47.29411765	45
Peak DC link voltage	70.58823529	64
KTR (transformer step up)	10	10
V LNrms	242.0672216	230

Table 4.1: Setting and results for attaining 230Vrms LN from 24V DC 1

These results are inline with the theoretical expectations of the Z-source capabilities detailed in Chapter 2. Based on Chapter 2 equations and Appendix E simulations we can expect 235Vrms LN. As a result of losses in

- the DC bus diode array
- 3-phase bridge switching losses
- Magnetizing losses for all inductive components

The final actual output of 230Vrms LN is 95% of the expected theoretical output.

4.3.5 Conclusion

Based on the input and output values the conclusion is that a 3 phase Z-source inverter with modified SVPWM has successfully been built and monitored with accordance to Chapter 2 theory and Appendix E simulations. The inverter achieved a 230Vrms LN voltage running from a battery voltage as low as 24V. This is an unachievable result using conventional SVPWM but has been accomplished with the proposed topology and modified SVPWM algorithm.

4.4. Step THREE: Implementing a Hybrid modified SV-PWM / SVZS-PWM algorithm with seamless transition between switching modes.

4.4.1. Goal of prototype experiment

There is no detailed literature on how to implement what was achieved in Chapter 4.3 and there is none on the proposed algorithm to be demonstrated in the following experiment. The goal is to again modify the SVPWM algorithm so that is can seamlessly swap over between SVPWM and SVZS-PWM. In other words the inverter can lock on to voltages within and out of range of SVPWM capabilities by swapping seamlessly to SVPWM with shoot through. Using shoot through increases losses. There are:

- I^2 R losses from current surges through semiconductors switches [2].
- Delay on reverse clamping of the diode array when for a short period of time the diode is not fully on or off [2].

- A high frequency current ripple over the Z-Source inductors causes losses in all inductive windings.

It would be ideal to run an inverter to its modulation limits without using shoot through. Running in over modulation mode introduces unwanted harmonics which also cause losses in all inductive components [6] and affect the output waveform quality. Once the inverter has reached full modulation but has not attained the desired output voltage it should make a smooth transition into a shoot through algorithm allowing the voltage to be boosted far beyond traditional limits without the addition of over modulation harmonics. Shoot through mode itself introduces fewer harmonics than a traditional inverter in over modulation [7]. For this reason a hybrid algorithm is detailed, implemented and discussed in this section.

4.4.2. Hybrid SV-PWM / SVZS-PWM algorithm objective

Fig 4.12 shows the result of a prototype voltage locking test using the proposed Hybrid SV-PWM / SVZS-PWM. The upper image shows the 6 independent reference waveforms used to generate the switching patterns. The lower image shows:

- In Blue: the modulation M
- In Red: the Shoot through
- In Grey: Vrms lock on voltage
- In Yellow: Vrms calculated from the 3 instantaneous output sinusoid values using the Clarke Transform.

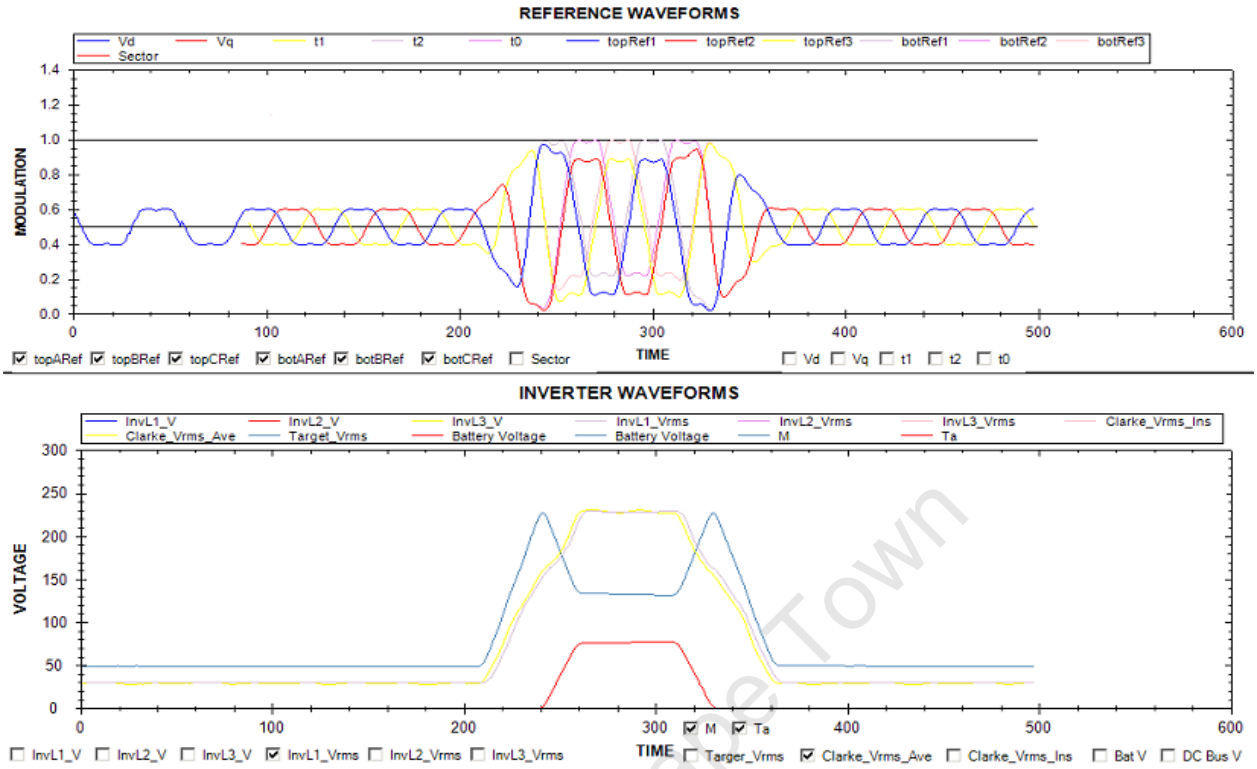


Fig 4.12: Voltage locking using Hybrid SV-PWM / SVZS-PWM from 30V-230Vrms

Using the PC monitoring and control software written by the author, the lock on Vrms LN voltage was varied from 30V to 230V. The DSP algorithm chases the lock on Vrms LN voltage specified by the user. As the inverter modulation breaches the over modulation threshold the inverter makes the transition into shoot through mode. On detection of over modulation without reaching its target voltage it brings the modulation ratio down and adds in shoot through intervals. This allows the inverter to use the entire switching period for either an active vector state or shoot through state sitting on the over modulation threshold. Notice how in fig 4.12 no reference waveforms ever breach 1(100%) of the modulation boundary. Notice how Vrms LN rises linearly during SVPWM and exponentially in SVZS-PWM mode which is consistent with the theory of chapter 2 and simulations of Appendix E and F. As the lock on voltage is varied back down to 30V so the algorithm chases through another seamless transition from SVZS-PWM to SV-PWM. The next section discusses how these results were achieved.

4.4.3. Hybrid SV-PWM / SVZS-PWM implementation

Vrms computation

To track the voltage a high speed computation of the RMS line to line voltage is required. Using the Clarke transform will allow the voltage locking algorithm to update using instantaneous values as apposed to calculating the RMS value from the root mean square of a number of values over a period of time. On the DSP the following computation are made to arrive at an RMS value from the instantaneous values of the 3 output LN voltages:

- First the real and imaginary parts of the three instantaneous values of the 3 phases are calculated using (22)

$$\begin{pmatrix} V_a \\ V_b \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}\begin{pmatrix} v_a \\ v_b \\ v_c \end{pmatrix} \qquad (22)$$

- Then the RMS of the magnitude of the real and imaginary resultant vector is calculated using (23). The root of the sum of the squares gives the space vector. The multiplication by $\sqrt{2}/3$ gives the RMS for 1 phase. See SVPWM theory in Appendix B.

$$Vrms\_LN = \frac{\sqrt{2}}{3}\sqrt{V_a^{\,2} + V_b^{\,2}} \qquad (23)$$

On the DSP the following code implements this calculation at 312Hz.

```
//////////////////////////////////////////////////////////////////
//Clarke Transform
//Real:=Voutab - 0.5*Voutbc -0.5*Voutca
Vd=(s32)(InvL1_V<<1)-InvL2_V-InvL3_V;
Vd>>=1;
//Imag:=Voutbc *sqrt(3)/2 - Voutca *sqrt(3)/2
Vq=( ((s32)(InvL3_V-InvL2_V)) * 56753 )>>16;
instantRMS= Vd*Vd + Vq*Vq; //Magnitude squared
instantRMS=(s32)qsqrt((u32)instantRMS); //Magnitude
instantRMS=(instantRMS*30894)>>16; //RMS
//////////////////////////////////////////////////////////////////
```

The locking algorithm will use this instantaneous value of Vrms LN for fast voltage feedback.

Voltage locking

The flow diagram in Fig 4.14 depicts an overview of the voltage locking control algorithm implemented on the DSP using Hybrid SV-PWM / SVZS-PWM for the Z-source inverter prototype. The algorithm integrates a seamless transition between SV-PWM and SVZS-PWM with a binary search for an effective Modulation value to produce the desired output Vrms LN.

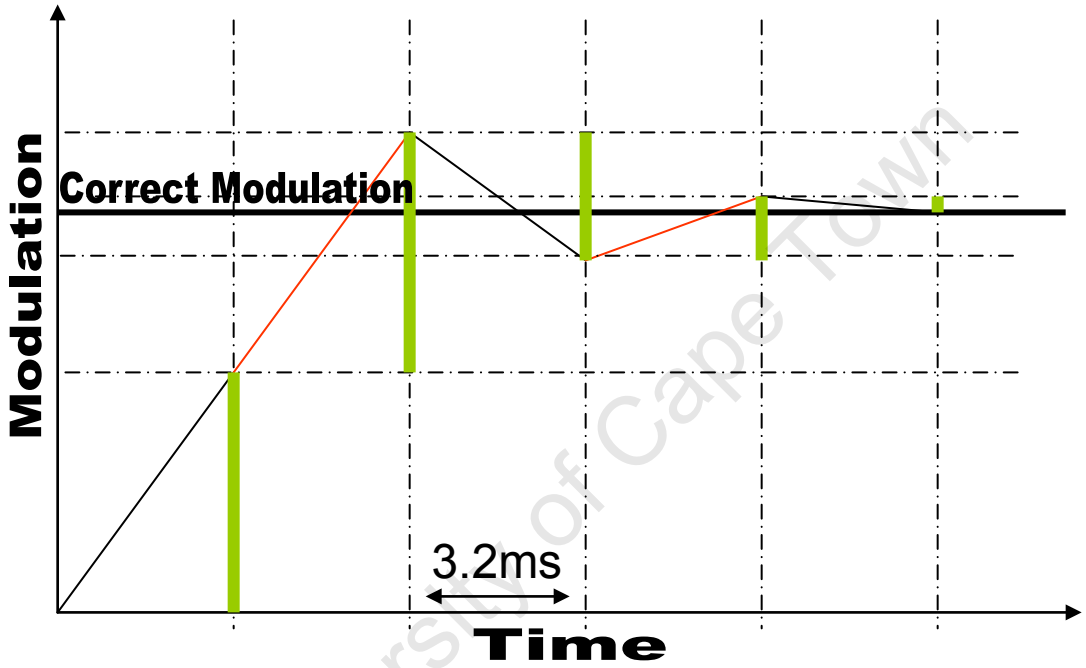


Fig 4.13: Binary Modulation search for correct

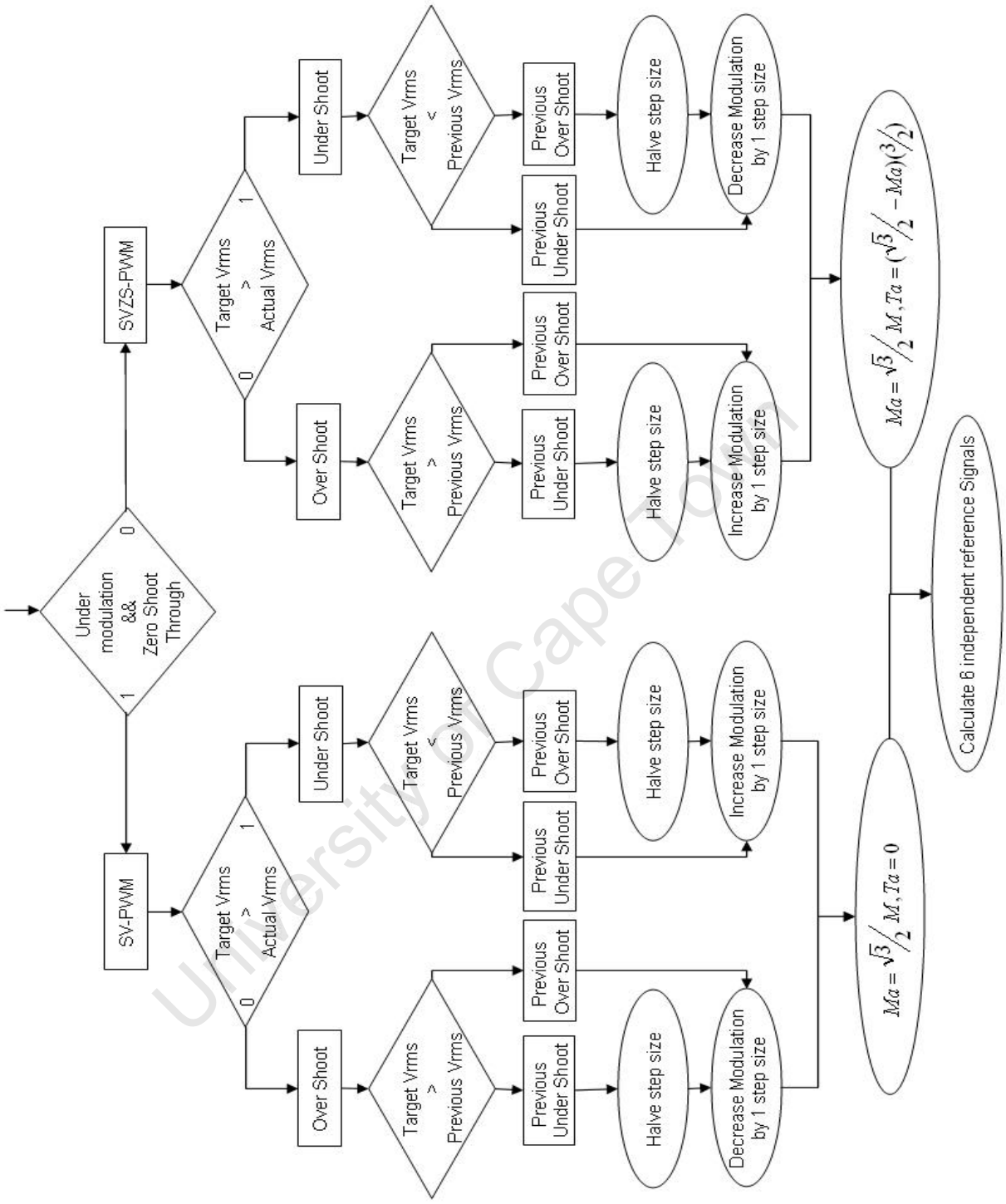


Fig 4:14: Overview of voltage locking control algorithm implemented on DSP using Hybrid SV-PWM / SVZS-PWM for Z-source inverter prototype.

4.4.4. Results analysis and Conclusions

Using the PC monitoring and control program the lock on voltage is set to 230Vrms LN from a 24V battery. In fig 4.15 The inverter increases its modulation (lower image in blue) to 100% but only achieves 158Vrms LN. It then decreases the modulation and fills the increasing zero vector periods with as much shoot through as possible without breaking the modulation boundary. Notice how only the 3 reference signals for the top switches can be seen in SV-PWM mode because the other 3 are identical and used to mirror the top switching signals. As soon as SVZS-PWM is introduced the other 3 reference signals appear with independent values so that 6 independent reference signals can be seen. At 56% Modulation with 33% Shoot through it achieves 230Vrms LN and stabilizes.

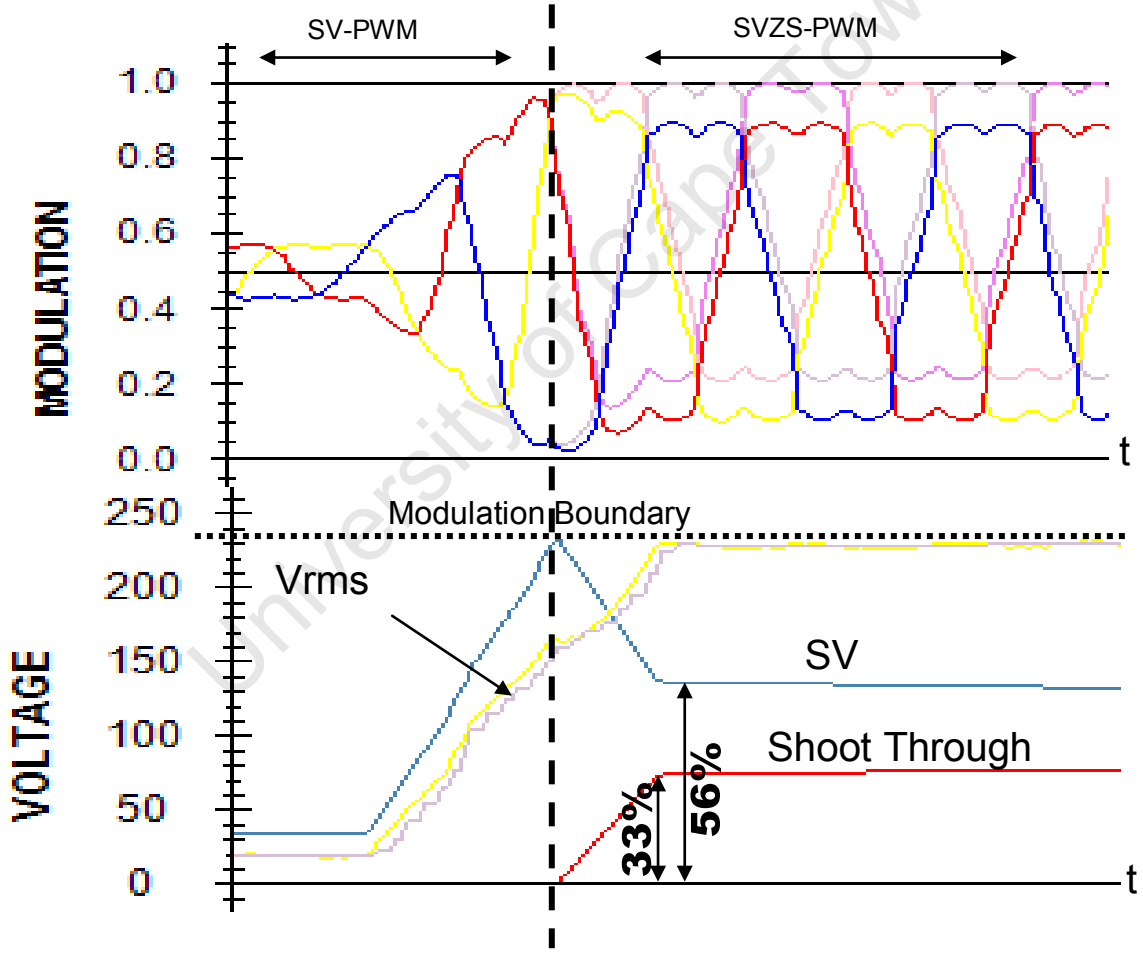


Fig 4:15: Reference signals, Vrms LN, modulation and shoot through relationship.

Based on the results it can be stated that the Z-source inverter prototype with Hybrid SV-PWM / SVZS-PWM control can lock to voltages as high as 230Vrms from a DC supply as low as 24V with a 1:10 transformer step up. This is not achievable using traditional SV-PWM technology.

A control algorithm has been introduced and implemented that can use a DC voltage as low as 24V and seamlessly move between control algorithms to achieve a desired output voltage up to 230Vrms LN on the 3 –phase output.

Fig 4.16: The 4 12V battery cells used to test a 24V, 36V and 48V configuration

As a commercial product without any software setup or hardware options the inverter can use 24V, 36V or 48V battery configurations and deliver a clean 3 phase 2kW 230Vrms supply. This allows distributors to stock one inverter line only without having to manage the supply and demand of 3 different stock items.

4.5 Step FOUR: Implementing a Hybrid modified SV-PWM / SVZS-PWM algorithm with seamless transition between switching modes while varying the input voltage during full load conditions.

4.5.1. Goal of prototype experiment

In this experiment the battery supply was replaced by a variable DC source. A variac was fed into a step down transformer and rectified. The DC supply varies during run time and the inverter should hold a steady 230Vrms LN. The DC bus will vary between 22V and 50V.

4.5.2. Setup

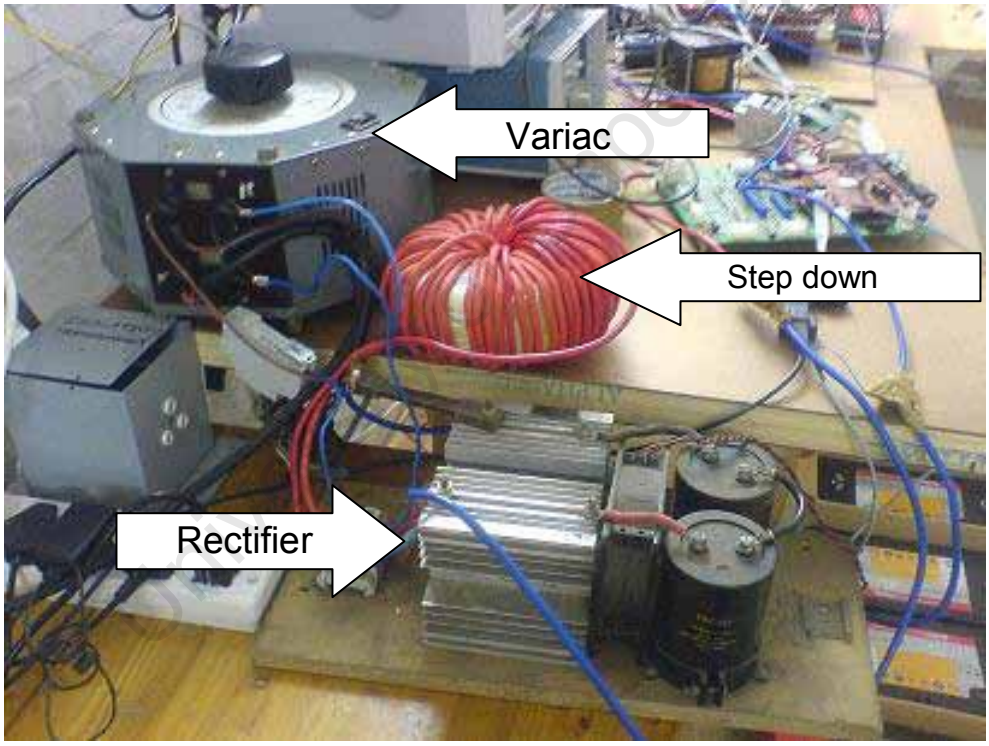


Fig 4.17: Variable DC supply setup with variac, transformer and rectifier.

For a 2 kW system at 24V the variac can have up to 80Amps flowing through it. Even a large variac like the one in fig 4.17 can only handle up to 20A through its windings. By stepping down the voltage from the variac we can allow the variac to

operate at a higher voltage and lower current. The DSP was programmed to limit the shoot through to 38% of a switching cycle to avoid over current conditions.

4.5.3. Results and Conclusions

In fig. 4.18 we see the results of a varying DC supply. The voltage varies between 20V and 50V DC as a fuel cell or wind turbine would do. The upper image shows the 6 independent reference waveforms used to generate the switching patterns.

The lower image shows:

- In Blue: the modulation M
- In Red: the Shoot through
- In Yellow: Vrms calculated from the 3 instantaneous output sinusoid values using the Clarke Transform.
- In Purple: Vrms calculated from the root mean squared of V1rms LN instantaneous values. This is shown because it does not reflect the ripple associated with unbalanced phases when calculating Vrms using the Clarke Transform (in Yellow). The ripple is not from unstable voltage locking but rather from a 1-2V voltage imbalance between phases.
- In Black: The variable DC supply (magnified 500% for clarity)

Notice that when there is zero shoot through (lower image in red) only 3 modulation signals can be seen as the top and bottom switches are mirroring each other and therefore use the same modulation signals. When the input is lower enough that traditional SVPWM cannot achieve 230V the modulation is decreased and the increasing zero vector periods are used for shoot through intervals to maintain 230V. Notice how during these conditions six modulation signals can be seen.

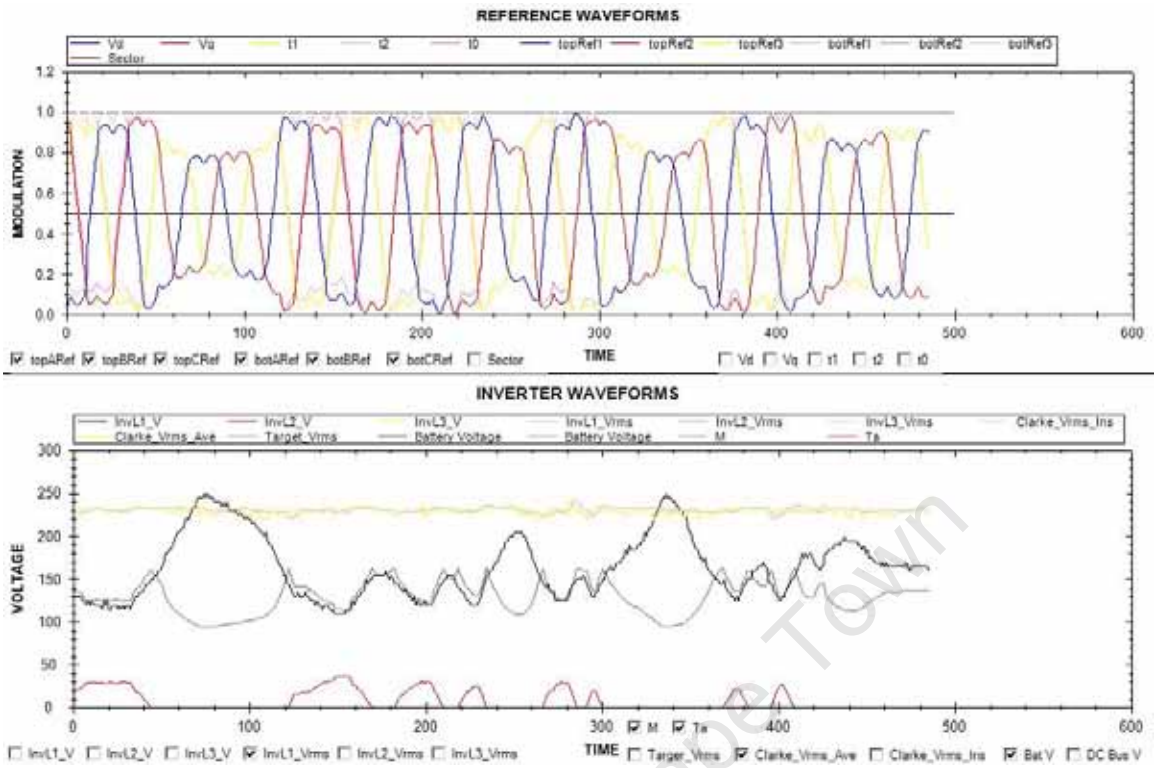


Fig. 4.18: Upper image - 6 independent modulation signals. Lower image - The modulation VS shoot though VS variable DC input VS Vrms LN.

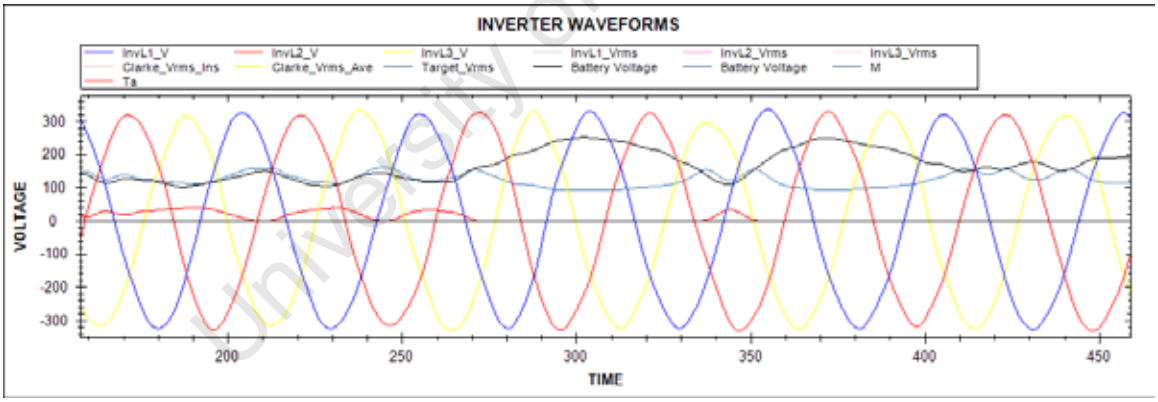


Fig. 4.19: The modulation VS shoot though VS variable DC input VS 3-phase output voltages.

In fig. 4.19 the same achievement is demonstrated except the 3-phase output voltage as apposed to the Clarke transform Vrms is shown. In black we have an unstable DC input with the necessary modulation and shoot through ratios to achieve a stable clean 230Vrms 3-phase sinusoidal output.

4.6 Final Conclusions

4.6.1. Statement of Achievements

- Theory: All relevant theory leading up to a HSVB-PWM has been investigated in detail (Chapter 2, Appendix A, B, C and F).
- Simulations:
  - Chapter 3 with Appendix D detailed the successful simulated of traditional SV-PWM.
  - Chapter 3 with Appendix E detailed the successful simulation of the addition of shoot through using a Z-source topology.
  - Together they consolidated theory mentioned in the last point and presented in detail a practical implementation.
- Prototype:
  - PC software: A completer control and monitoring program was designed and implemented on which most of the waveforms presented were captured. All set points of the prototype could be varied on the fly using the PC program control interface.
  - Hardware: An operational 2 kW 3-phase Z-source inverter has been built running from multiple battery bank configurations or a variable DC input.
  - Embedded software: C code was written that achieved:
    - 5 kHz ADC sampling of the 3-phase output voltages, the DC supply voltage and the DC bridge voltage.
    - PC <-> TMS2812 monitoring and control integration at 115200bps
    - SV-PWM / SVZS-PWM control algorithms running up to 10 kHz.
    - Voltage locking up to 240 Vrms from an on the fly variable supply as low as 20V DC.
    - Seamless SV-PWM / SVST-PWM transition during voltage locking (HSVB-PWM).

- 6 independent PWM outputs with a 20 kHz switching cycle.
- Shoot through current protection during supply under voltage.

4.7. Future development

An energy company in Ireland has expressed interest in this inverter for wind turbine applications. Using the voltage locking capabilities a wind turbine can be connected through a rectifier as a volatile DC supply. The inverter could maintain a stable 230Vrms supply given the volatile DC input conditions. Additional control needed:

- Maximum power point tracking (MPPT) to utilize the wind turbine at maximum power output given a specific wind speed.
- Voltage locking must be extended to grid phase locking using the control algorithms briefly investigated in Appendix X.
- The inverter must be rebuilt using specially designed overlapping bus plates as discussed in chapter 4 to minimize inductive spikes.
- An investigation into the diode choice used on the Z-source input should be done to minimize diode losses during voltage clamping at shoot through frequencies up to 60 kHz.
- A cost analysis between increasing the output transformer size and Z-source component size should be done to find the optimum balance of the 2 when trying to operate at 230Vrms from low voltage supplies.

END

APPENDIX A

Traditional 3-phase inverter Topology and Operation

A1. Basic Topology

The traditional voltage source inverter and the current source inverter are shown in Fig. A1 and Fig. A2 respectively. They are either converters or inverters depending on the power flow direction. When power flow is to a DC load they are converters and when to an AC load they are inverters.

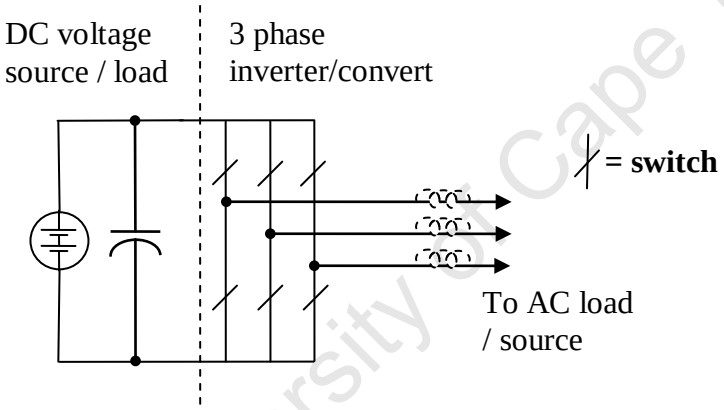


Fig A1 Traditional Voltage source inverter

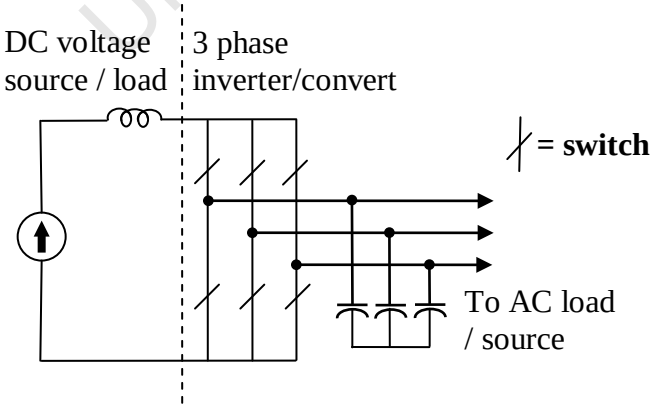


Fig A2 Traditional Current source inverter

A2. Comparison between voltage source and current source inverter

A2.1. The Current source inverter

The DC current source can be a battery or fuel cell with a relatively large inductor in series to provide the current source to the 3 phase Inverter Bridge. The current source inverter is a step-up or boost inverter. This means that the AC output voltage can never be less than the DC bus voltage [3].

A2.2. The Voltage source inverter

The DC voltage source can be a battery or fuel cell normally with a relatively large capacitor in parallel. This DC voltage source feeds the 3 phase inverter bridge. The voltage source inverter is a step-down or buck inverter. This means that the obtainable AC output is less and can never exceed the DC bus voltage [3].

A3. Downfall of these traditional inverter types

- Voltage range: The fact that the voltage source inverter can only step-down the voltage and the current source inverter can only step up the voltage is a problem. In the proposed system the aim is to use a fuel cell stack as a DC source. The fuel cell stack voltage as previously discussed is highly unstable depending on the current drawn. What is needed is an inverter capable of stepping up and stepping down the voltage of the fuel cell output. Neither of the traditional inverters can achieve this.

- EMI susceptibility: The top switches and bottom switches in each leg always compliment each other in that they are always in opposite states in both inverter types switching schemes. If the upper and lower switch

in one leg were to close at the same time or overlap on at any point the DC bus would be shorted through that leg blowing one or both switches and seriously damaging the inverter. To prevent this there is normally a blanking time added between switches state transitions to counteract any timing error that could result in an overlap. The overlap though effective adds additional complications in harmonic distortion of the output sinusoids. The switches toggle at rates up to 20 kHz in some applications. This makes driving these switches in a timely manner crucial. A microprocessor computing the switching patterns based on the programmed algorithm will output signals to a driver board that drives the switches on or off. These systems are extremely susceptible to electro magnetic interference (EMI) which can cause major interference in the driving signals to these switches resulting in an overlap even with a large blanking time between switching states. This is one of the major challenges in inverter reliability [2-4].

A4. Traditional Carrier Based PWM

The voltage source inverter is the more commonly used traditional inverter type. We will therefore use this inverter type in this chapter to describe the traditional switching scheme in detail before we move onto the more advanced switching algorithms.

Fig. A3 shows a voltage source inverter with its six switches and anti-parallel diodes forming the 3 phase inverter bridge. The switching scheme that needs to be generated in each leg is identical except 120 degrees out of phase if one switching cycle can be considered a 360 degree interval. This is needed to generate three identical sinusoidal PWM signals 120 degrees out of phase that will in turn produce 3 identical sinusoidal voltages through three identical LC filters.

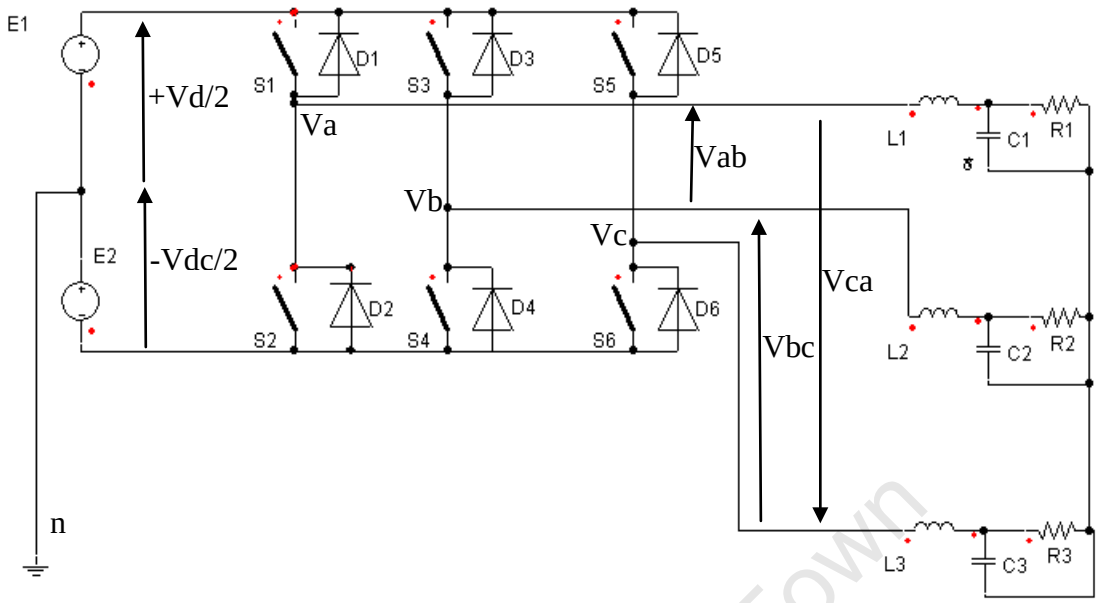


Fig. A3 Three phase inverter topology

In Fig A3:

Vab = Van-Vbn
Vbc = Vbn-Vcn
Vca = Vcn-Van

The inverter bridge is made up of 3 legs each with 2 switches, one leg per phase.

By operating the switches in the first leg for example Van = Va – n can be controlled.

S1	S2	Van
OFF	OFF	Vdc
OFF	ON	-Vdc/2
ON	OFF	+Vdc/2
ON	ON	0 (short)

Table A1 Van at the different switching states on the first leg of the inverter.

Note that when both switches are on we have a short or “shoot through” which would ordinarily blow the switches in this inverter. Previously it was mentioned

that a blanking or dead time was used between switching transitions to avoid this but for now in this discussion we will assume the top and bottom switches are perfect compliments at all times and exclude any dead time in the switching scheme.

A5. Sine-Triangle Pulse Width Modulation Implementation

The switching signals are usually generated by a microprocessor executing the appropriate algorithms. The switching signals can and were often generated using comparators and waveform generators. Waveforms were compared through a comparator and the output '1' or '0' signal from the comparators would trigger the driver circuits which would drive the switches. Microprocessors work in much the same way by comparing counters and look up table values.

The most common carrier based PWM method is Sine-Triangle Pulse Width Modulation (STPWM). The 2 signals compared are the carrier signal and the control signal also known as the modulation signal. The carrier signal is a triangular wave. In software an interrupt driven up-down counter would be used to generate the appropriate values. The modulation signal in STPWM is a sine wave. Fig A4.a shows the carrier triangular waveform against the three sinusoids used as the control signals. By comparing the control signals with the carrier signals switching signals are generated as follows with reference to Fig. A3.

Values Compared	Switch State Transitions
Vcontrol (phase-a) > Vtriangle	S1 ON / S2 OFF
Vcontrol (phase-a) < Vtriangle	S2 ON / S1 OFF
Vcontrol (phase-b) > Vtriangle	S3 / S4 OFF
Vcontrol (phase-b) < Vtriangle	S4 / S3 OFF
Vcontrol (phase-c) > Vtriangle	S5 / S6 OFF
Vcontrol (phase-c) < Vtriangle	S6 / S5 OFF

Table A2: Switch states depending on the comparison of Vcontrol [4]

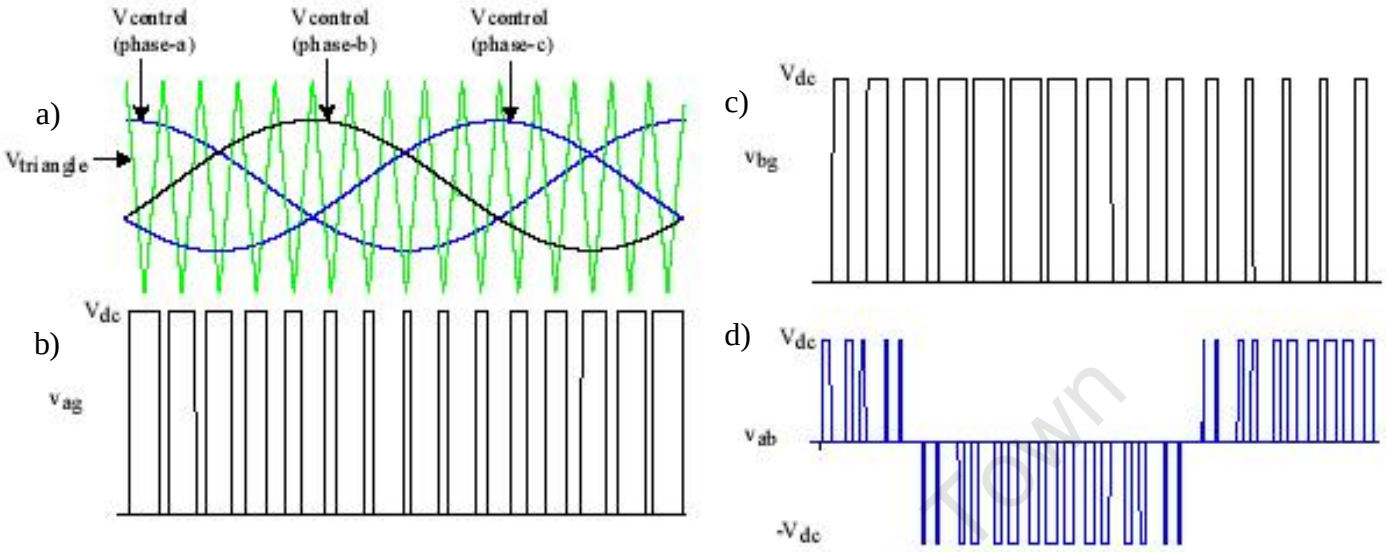


Fig. A4.a: Triangular waveform carrier with the 3 sine wave modulation signals.
Fig. A4.b: PWM signal on phase-a leg with reference to ground.
Fig. A4.c: PWM signal on phase-b leg with reference to ground.
Fig. A4.d: PWM voltage signal between phase-a and phase-b

A6. Modulation Boundary Limitations

The amplitude modulation ratio is referred to as m_a where:

$$m_a = \frac{Vcontrol}{Vtriangle} = \frac{Voutput}{Vdc/2} \qquad (2)$$

Fig A5 shows the output sign wave peak amplitude as m_a is increased. The input DC voltage in this case is 350V. The graph shows that the line to line voltage increases linearly for $m_a < 1$ to a peak value of about 300V. After that the over modulation range is entered and the line to line voltage continues to increase as we approach square wave generation.

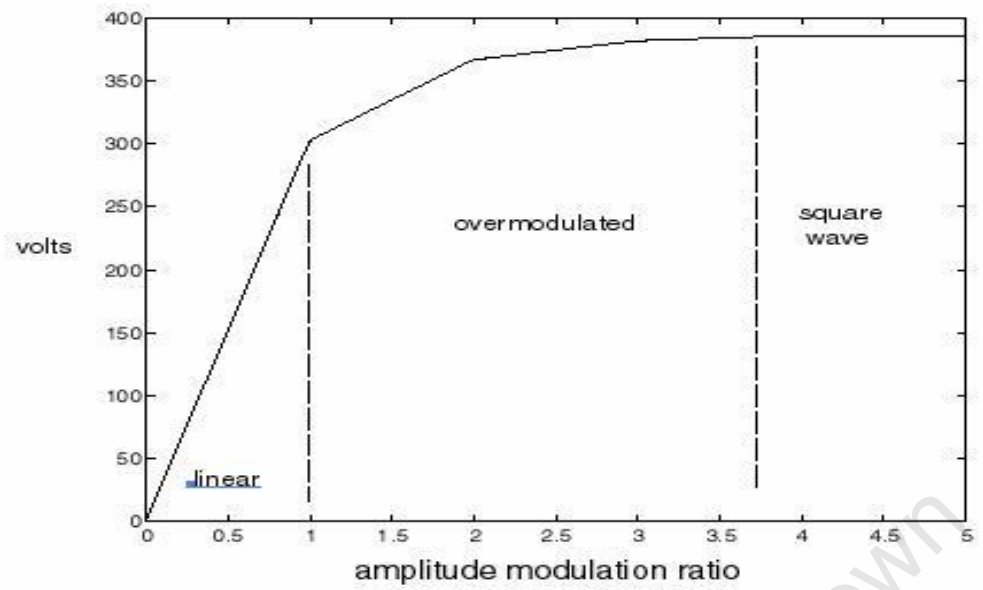


Fig A5: Output sign wave peak amplitude versus m_a [5].

As m_a increases over 1 the quality of the sign wave deteriorates as more harmonics join the fundamental until the wave form becomes a square wave. So over modulation can be used to bolster the output voltage at the cost of harmonic distortion[8]. In appendix B the modulation boundary is increased for better voltage utilization of the DC bus through SVPWM switching techniques.

APPENDIX B

Space Vector Pulse Width Modulation for a 3-phase bridge

With the advance in microprocessors more complex and processor intensive algorithms became viable for PWM switching techniques. Space vector PWM (SVPWM) is one such algorithm. SVPWM offers less harmonics distortion on the output voltages and a more efficient use of the supply voltage [1].

B1 The Space vectors

SVPWM treats each switching state as a vector. Fig 4.1 shows the 2^3 or 8 possible switching states S0 to S7. Note that the bottom switch in one leg is still always the compliment of the top switch. Each switching state is associated with a vector U_0 to U_7 respectively. Switching to states S0 and S7 gives the zero vectors U_0 and U_7 respectively. The six none zero vectors form the axis of a hexagonal plane with the zero vectors at the origin as shown in Fig 4.2 [1-4].

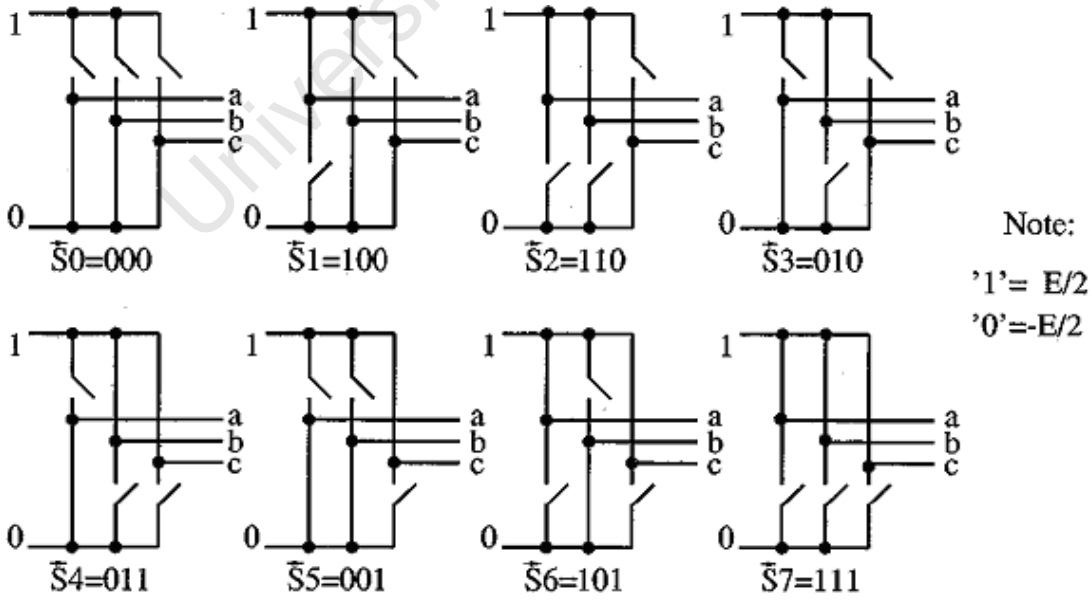


Fig. B1: Possible switching [4].

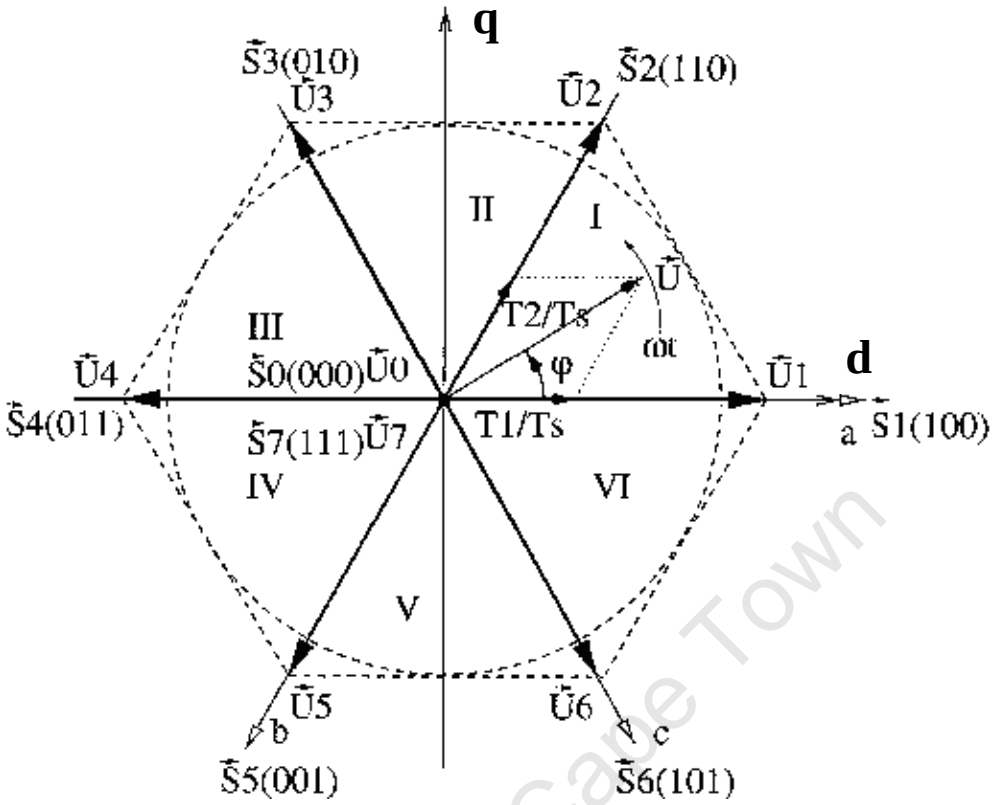


Fig. B2: Hexagonal plane formed by the 6 none zero vectors and the zero vectors at the origin [4].

The vectors are 60 apart with a magnitude of 1. The hexagonal vector composition sits on the dq plane with vector U_1 on the positive d or real axis. Essentially the idea is to jump between two adjacent vectors and a zero vector to replicate the desired reference vector angle and magnitude. This reference vector spinning in the qd plane is what we ultimately use to build are sinusoidal output. The ratio of time spent between say U_1 and U_2 will determine the angle of the resultant vector U in sector 1. If an equal amount of time is spent in either state the resultant vector would be at 30 equidistant between the U_1 and U_2 60 apart. How much time is spent in a zero vector state will affect the magnitude of the reference vector U [1, 2, and 4].

Notice that the switching states of adjacent vectors are only one switch difference away. Which means toggling between these states only requires one switch to change. This reduces switching losses to a minimum.

B2. Approximating the Reference Vector U

In sector 1 the reference vector U in one sampling interval can be written as:

$$\overset{\vee}{U} = \frac{T_1}{T_s}\overset{\vee}{U}_1 + \frac{T_2}{T_s}\overset{\vee}{U}_2 + \frac{T_7}{T_s}\overset{\vee}{U}_7 + \frac{T_0}{T_s}\overset{\vee}{U}_0 \qquad (3)$$

T_1, T_2, T_7 and T_0 are the duration of time spent in states U1, U2, U7 and U0 respectively. T_s is the total time or the time of one sampling interval.

$$T_s = T_1 + T_2 + T_7 + T_0 \qquad (4)$$

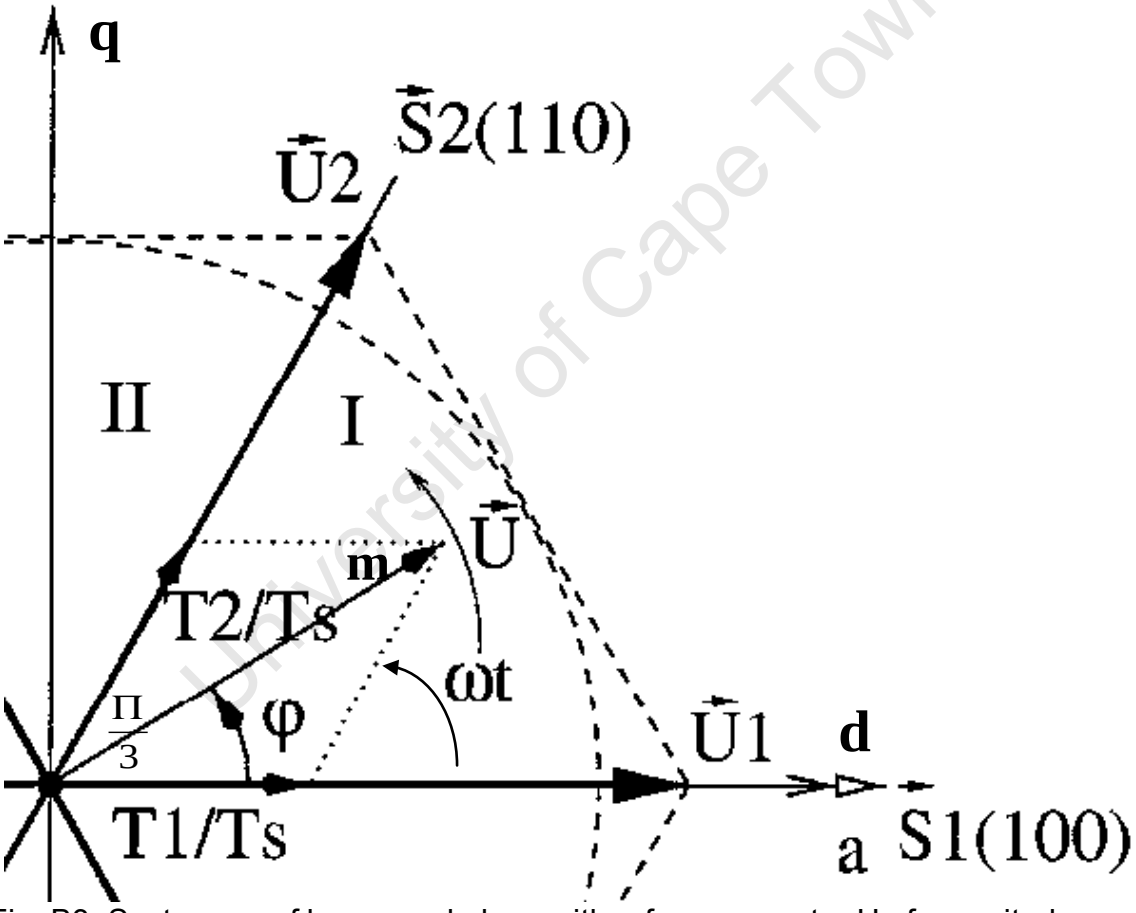


Fig. B3: Sector one of hexagonal plane with reference vector U of magnitude m at an angle φ [4].

We can find the appropriate T_1 and T_2 given the desired magnitude and angle of the reference vector. In fig. B3 we have the hexagonal plane once again with a

desired reference vector of length m and angle of φ. From fig. B3 we can see that the following relationship between $\vec{U}_1$, $\vec{U}_2$ and U exists.

$$\frac{\left|\vec{U}\right|}{\sin(2\Pi/3)} = \frac{T_1}{T_s}\frac{\left|\vec{U}_1\right|}{\sin(\Pi/3 - \vartheta)} = \frac{T_2}{T_s}\frac{\left|\vec{U}_2\right|}{\sin(\varphi)} \qquad (5)$$

$\vec{U}_1$ and $\vec{U}_2$ are unit vectors(magnitude of 1) and are scaled by the fractions $\frac{T_1}{T_s}$ and $\frac{T_2}{T_s}$. From (5) we can write s $\frac{T_1}{T_s}$ and $\frac{T_2}{T_s}$ as follows:

$$\frac{T_1}{T_s} = \frac{2}{\sqrt{3}}\left|\vec{U}\right|\sin(\Pi/3 - \varphi) = \frac{2}{\sqrt{3}}\left|\vec{U}\right|\cos(\varphi + \Pi/6)$$

$$\frac{T_2}{T_s} = \frac{2}{\sqrt{3}}\left|\vec{U}\right|\sin(\varphi) = \frac{2}{\sqrt{3}}\left|\vec{U}\right|\cos(\varphi + 3\Pi/2)$$

$T_0 + T_7 = T_s - T_1 - T_2$ (6)

Table B1 shows the formulae for calculating the time duration values for the two appropriate adjacent active vectors in each of the six sectors. It also shows in each case similarly how the total time spent in the zero vector state can be calculated by subtracting the total time spent in the active vector states from the switching period duration T_s [1, 2, and 4].

Sector 1 $0 \le wt = \varphi \le \frac{\Pi}{3}$	**Sector 2** $\frac{\Pi}{3} \le wt = \varphi \le \frac{2\Pi}{3}$	**Sector 3** $\frac{2\Pi}{3} \le wt = \varphi \le \Pi$
$T_1 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + \Pi/6)$ $T_2 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 3\Pi/2)$ $T_0 + T_7 = T_s - T_1 - T_2$	$T_2 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 11\Pi/6)$ $T_3 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 7\Pi/6)$ $T_0 + T_7 = T_s - T_2 - T_3$	$T_3 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 3\Pi/2)$ $T_4 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 5\Pi/6)$ $T_0 + T_7 = T_s - T_3 - T_4$
Sector 4 $\Pi \le wt = \varphi \le \frac{4\Pi}{3}$	**Sector 5** $\frac{4\Pi}{3} \le wt = \varphi \le \frac{5\Pi}{3}$	**Sector 6** $\frac{5\Pi}{3} \le wt = \varphi \le \Pi$
$T_4 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 7\Pi/6)$ $T_5 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + \Pi/2)$ $T_0 + T_7 = T_s - T_4 - T_5$	$T_5 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 5\Pi/6)$ $T_6 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + \Pi/6)$ $T_0 + T_7 = T_s - T_5 - T_6$	$T_6 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + \Pi/2)$ $T_1 = \frac{2}{\sqrt{3}}\lvert\vec{U}\rvert T_s \cos(\varphi + 11\Pi/6)$ $T_0 + T_7 = T_s - T_6 - T_1$

Table B1: The formulae for calculating the time duration values for the two appropriate adjacent active vectors and the zero vectors in each of the 6 sectors [4].

B3. Generating the reference signals

Once an appropriate T_1 and T_2 have been calculated three reference signals are generated as follows:

$$\begin{aligned} U_c(kT_s) &= \frac{T_0 + T_7}{2} \\ U_b(kT_s) &= \frac{T_0 + T_7}{2} + T_1 = U_c(kT_s) + T_1 \\ U_a(kT_s) &= \frac{T_0 + T_7}{2} + T_1 + T_2 = U_b(kT_s) + T_2 \end{aligned} \tag{7}$$

Fig. B4 shows the three reference signals $U_a(kT_s), U_b(kT_s), U_c(kT_s)$ generated in a Simplorer 6.0 SVPWM simulation. These three reference signals are used to build the three control signals in SVPWM

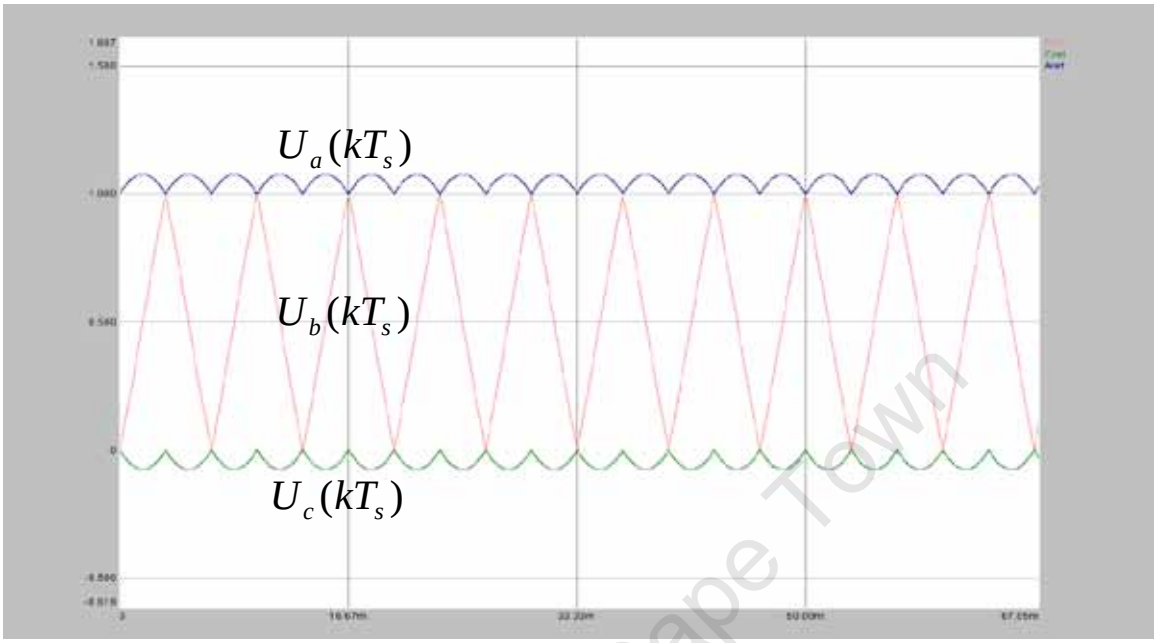


Fig. B4: Reference signals $U_a(kT_s)$, $U_b(kT_s)$, $U_c(kT_s)$.

B4. Generating the control signals

The control signals are composed of a piecewise combination of reference signals. Then these control signals are compared against the carrier triangular wave signal through comparators. The input into each of the three comparators is switched between the three reference signals $U_a(kT_s)$, $U_b(kT_s)$, $U_c(kT_s)$. This happens as we change from one sector to the next. This is the same as inputting a new signal (control signal) built from a 6 piece combination of the three reference signals.

Table B2 shows which reference signals are used at each comparator input for the 6 sectors. Using the configuration for each sector shown in table B2 the waveforms shown in fig B5 are generated from the three reference waveforms shown in fig. B4.

Sector	Comparator 1 input	Comparator 2 input	Comparator 3 input
1	U**b**(kT_S)	U**c**(kT_S)	U**a**(kT_S)
2	U**a**(kT_S)	U**c**(kT_S)	U**b**(kT_S)
3	U**a**(kT_S)	U**b**(kT_S)	U**c**(kT_S)
4	U**b**(kT_S)	U**a**(kT_S)	U**c**(kT_S)
5	U**c**(kT_S)	U**a**(kT_S)	U**b**(kT_S)
6	U**c**(kT_S)	U**b**(kT_S)	U**a**(kT_S)

Table B2: The appropriate configuration of reference signals fed to the 3 comparators for each sector.

Note that unlike in sine triangular wave PWM we do not have sinusoidal control signals. The line to neutral voltage signals will have the form of these control signals and not sinusoids. Note that they are 120 apart in phase.

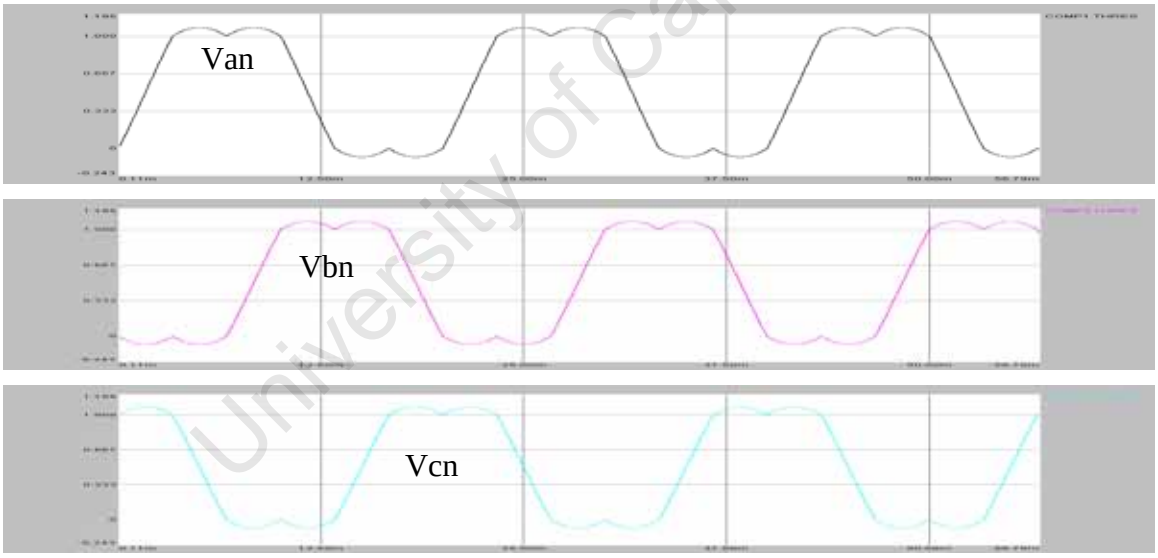


Fig. B5: The 3 control signals that are seen on each of the comparator inputs

B5. Switching patterns and PWM output

The control signals are the same as the desired output frequency of the 3-phase inverter, in this case 50Hz. The triangular wave runs at a much higher 10 kHz. Fig. B6 shows how the three reference signals are compared with the carrier triangular wave to give the necessary switching pattern over 1 period in sector 1 [1], [3], [4]. Notice how each of the six switches need only turn on and off once per switching cycle. The same applies to the PWM output in all other sectors as can be seen in figure B7. The swapping of reference signals to each of the three comparator inputs can be seen in the changing patterns from one sector to the next.

To avoid confusion note that in figure B7 the switching period is divided into T_z where $2T_z = T_s$. Also T_0 represents the entire zero vector state now not just state S1 but S1 and S7. Also T_1 and T_2 now represent half the time spent in any two adjacent vectors and not just those in sector 1. Notice how the order of T_1 and T_2 swap as we move from one sector to the next showing how the last vector state of the last sector is the first vector state of the next sector. I.e. U_1 and U_2 have T1 and T2 respectively in sector 1 and then U_2 and U_3 have T2 and T1 respectively in sector 2.

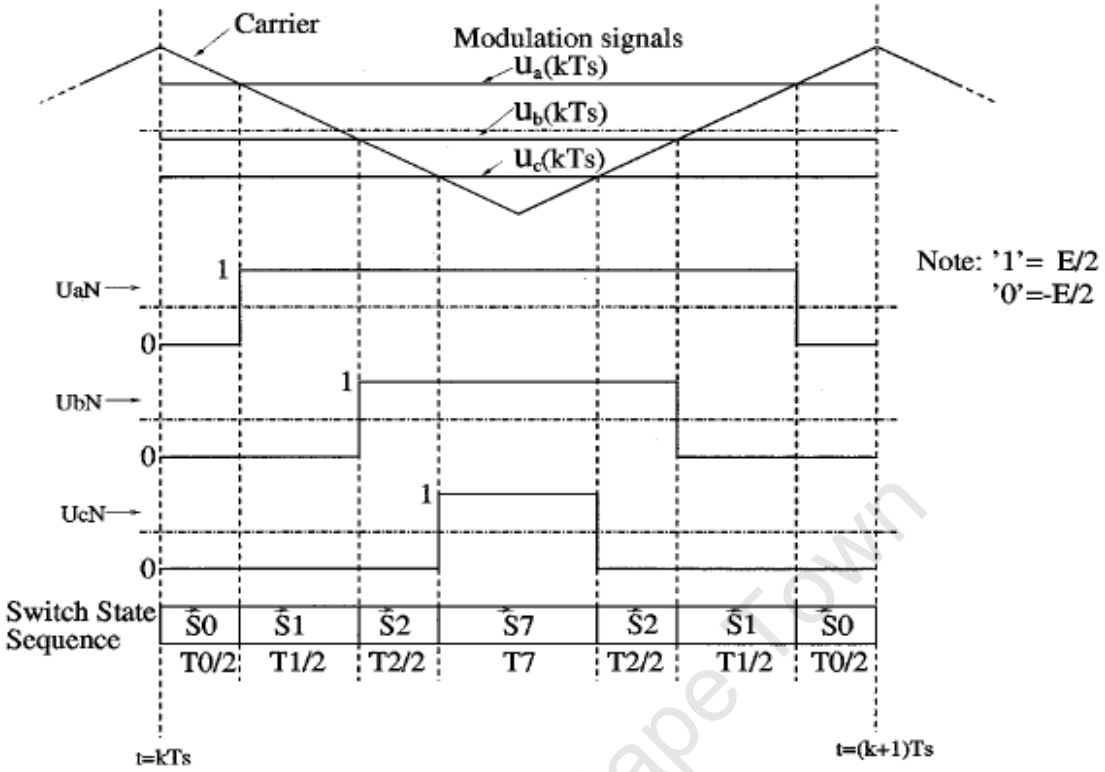


Fig. B6: SVPWM switching signals over one switching period in sector 1

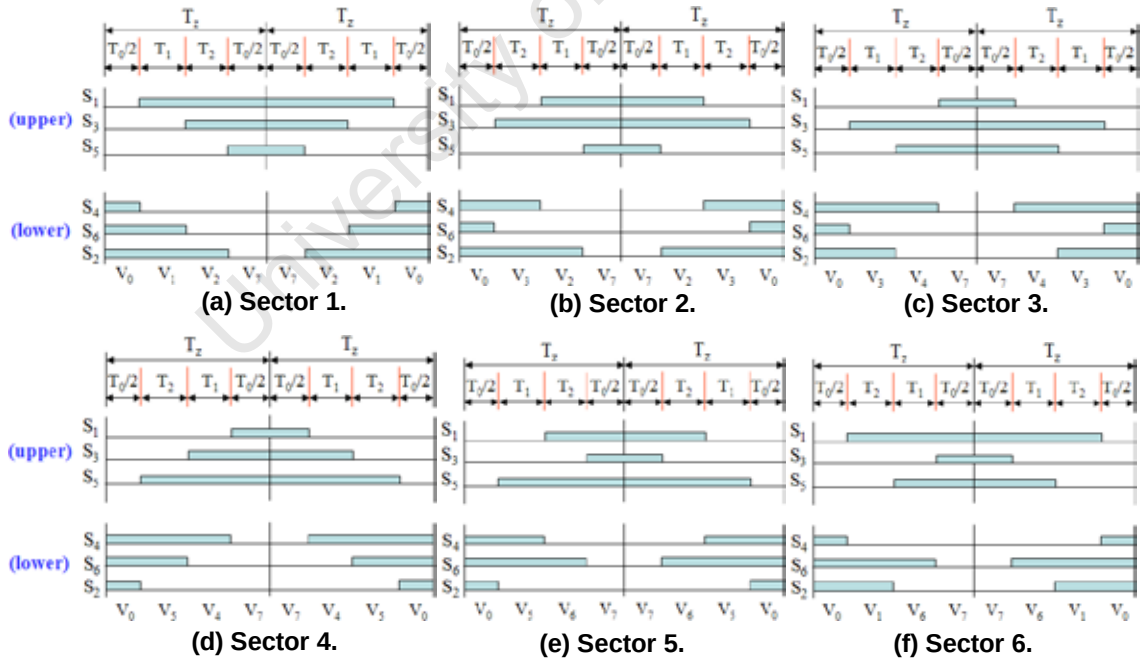


Fig. B7: SVPWM switching patterns at each sector [12]

B6. Inverter Output Form

The final line to neutral output voltage signal after the LC filter will take the form of the modulating signals shown in fig. B5. The signals generated from each leg after each LC filter will also be 120° out of phase from each other. The line to line voltage however is a sine wave. Figure B8 shows the final output of the inverter on all three phases.

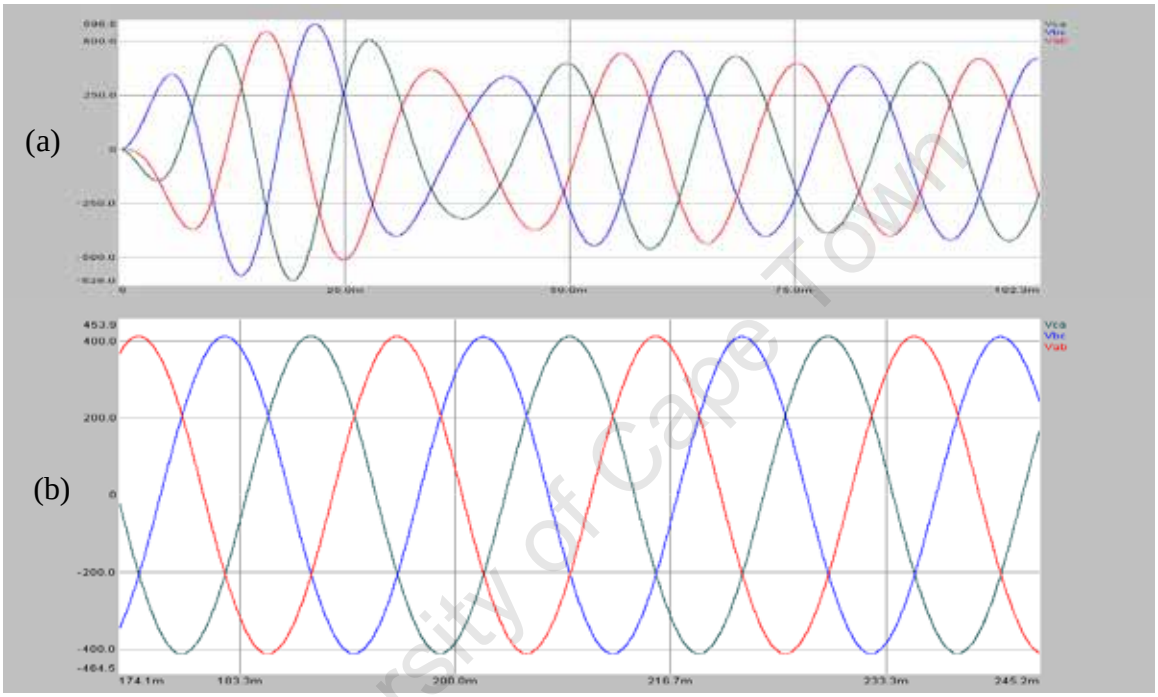


Fig. B8: (a) Initial 3-phase output voltage signals starting from zero state.
(b) Steady state 3-phase output voltage signals

B7. The linear modulation boundary

In sector 1 if we were to have $T_0 + T_7 = 0$ we would have that $T_1 + T_2 = T_s$. T_1 and T_2 would take up the entire switching cycle. This condition places the vector on the boundary between linear modulation and over modulation.

Under these conditions using equation 6 yields:

$$\left|\overset{v}{U}\right| = \frac{\sin(\Pi/3)}{\sin(2\Pi/3 - \varphi)} \qquad (8)$$

Therefore, the boundary between the linear modulation range and the over modulation range is the hexagon [9], [11] in Fig. B4 with the linear modulation range lying within the hexagonal boundary.

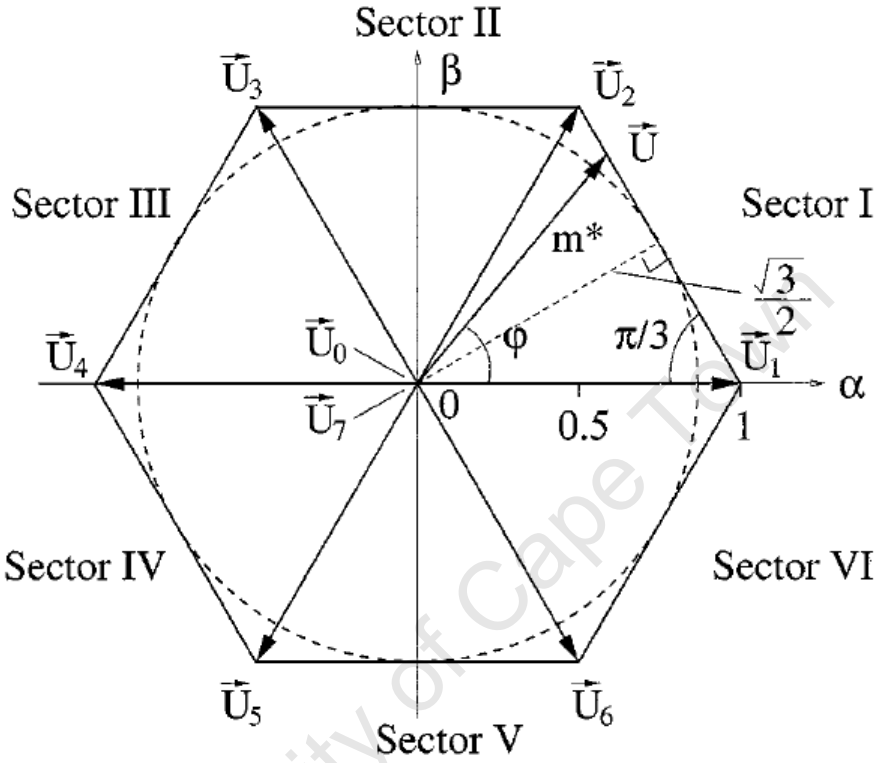


Fig. B4: The modulation range of space vector pulse width modulation

The inscribed circle within the hexagon is the maximum constant magnitude that can be maintained to generate a sinusoidal output. If we place $\vec{U}$ equidistant in angle from $\vec{U}_1$ and $\vec{U}_2$ at an angle of $\varphi = \Pi/3$ using (8) we get the following for the radius of the inscribed circle:

$$\left|\vec{U}\right| = \frac{\sin(\Pi/3)}{\sin(2\Pi/3 - \Pi/3)} = \frac{\sqrt{3}}{2} \qquad (9)$$

Therefore $\frac{\sqrt{3}}{2}V_{dc}$ is the maximum output line to neutral peak voltage.

APPENDIX C

Simplorer 7 Setup and Simulator Technical issues

"Simplorer® is multi-domain, system simulation software for the design of high-performance electromechanical systems commonly found in the automotive, aerospace/defense, and industrial automation industries. With a wide range of modeling techniques, statistical analysis capability and adherence to IEEE standards, Simplorer greatly reduces engineering time and prototype iterations while improving design performance of electrical, mechatronic, power-electronic, and electromechanical systems" – www.ansoft.com.

All simulations were done using Simplorer v6. Simplorer v6 simulations are forward compatible with Simplorer v7. With regards to my simulations the only difference between v6 and v7 is that v7 has an upgraded VHDL compiler. All other necessary libraries and components are available in v6. VHDL code in v6 often incorrectly compiles nested and compound "IF" statements and causes unpredictable errors at compilation and/or runtime.

When using v6 three level "IF" statements with trailing else statements should be avoided. Also all variables compared in an "IF" statement must be fully resolved. I.e. there can be no computations besides the comparison of previously calculated variables. E.g. IF(abs(x)<tan(y)) will not compile or not compile correctly. These small details are mentioned to aid and spare tedious debugging for anyone who wishes to reproduce these simulations.

Simplorer along with MATLAB simulink are the most widely used packages for power electronic simulations. Simplorer was used in simulations because of its VHDL support. VHDL in short is faster. The simulations use VHDL to quickly compute switching algorithms on the fly at switching speeds up to 20 kHz.

APPENDIX D

Open Loop 3-phase SVPWM simulation for standalone 3-phase bridge with 3 phase load.

AIM:

Using the given circuit libraries and VHDL language a working model for accurate simulation is built for implementing unmodified SVPWM. The aim of this simulation is to verify both existing theory and the effectiveness of the simulator for high frequency complex PWM generation using an amalgamation of the component models and the integrated VHDL language. The VHDL language will be used to program the SVPWM algorithm while the circuit simulation uses the standard component libraries. Specific expected outcomes are:

- Develop and test an algorithm to simulate the correct SVPWM generation of switching signals for each phase.
- Realise increased efficiency in DC bus utilisation compared to sine triangular SVPWM.

D1. Topology of 3-phase inverter used in simulation

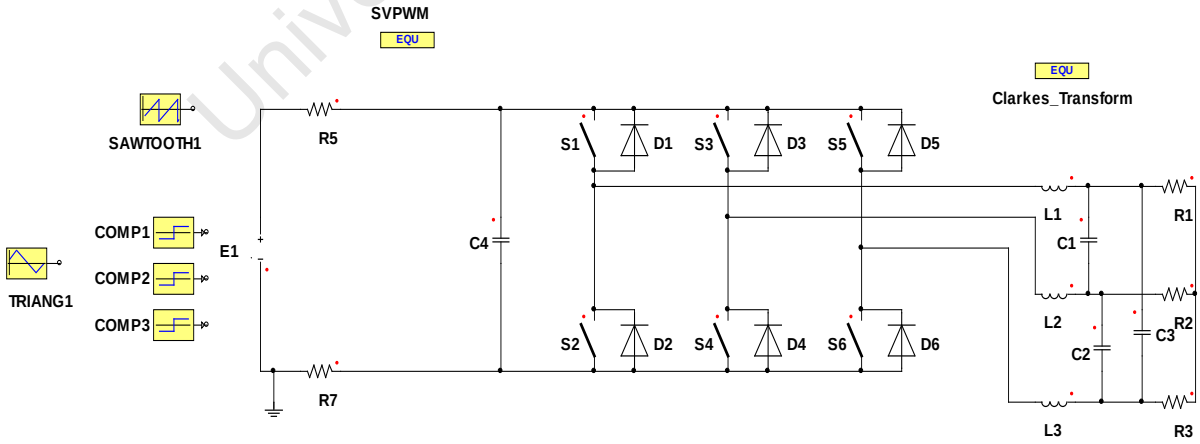


Fig. D1: A 3 phase inverter with filter and load

Simulation component summary:

- E1: DC voltage source (300V).
- R5 and R7 (1uΩ): Simulator calculations do not converge unless some small impedance is added to the system.
- C4: Used to steady the DC bus
- S1 to S6 and D1 to D6: These 6 ideal switches with anti-parallel diodes form the 3 phase bridge.
- L1, L2 and L3 along with C1, C2 and C3 form a first order 3-phase filter for the 3 output phases.
- R1, R2 and R3 load the system.
- COMP1, COMP2 and COMP3 compare the signal from TRIANG1 with the output of ZSVPWM which generates the reference signals using VHDL coded algorithms. The Clarkes_Transform block is used for analysis.
- SAWTOOTH1 is a ramp function running from 0 to 360 at 50Hz. The variable SAWTOOTH1.VAL is used as an input to give the correct frequency and angle of the generated space vector.

D2. Simulation Process Overview

Figure D2 gives an overview of the simulation algorithm programmed in VHDL.

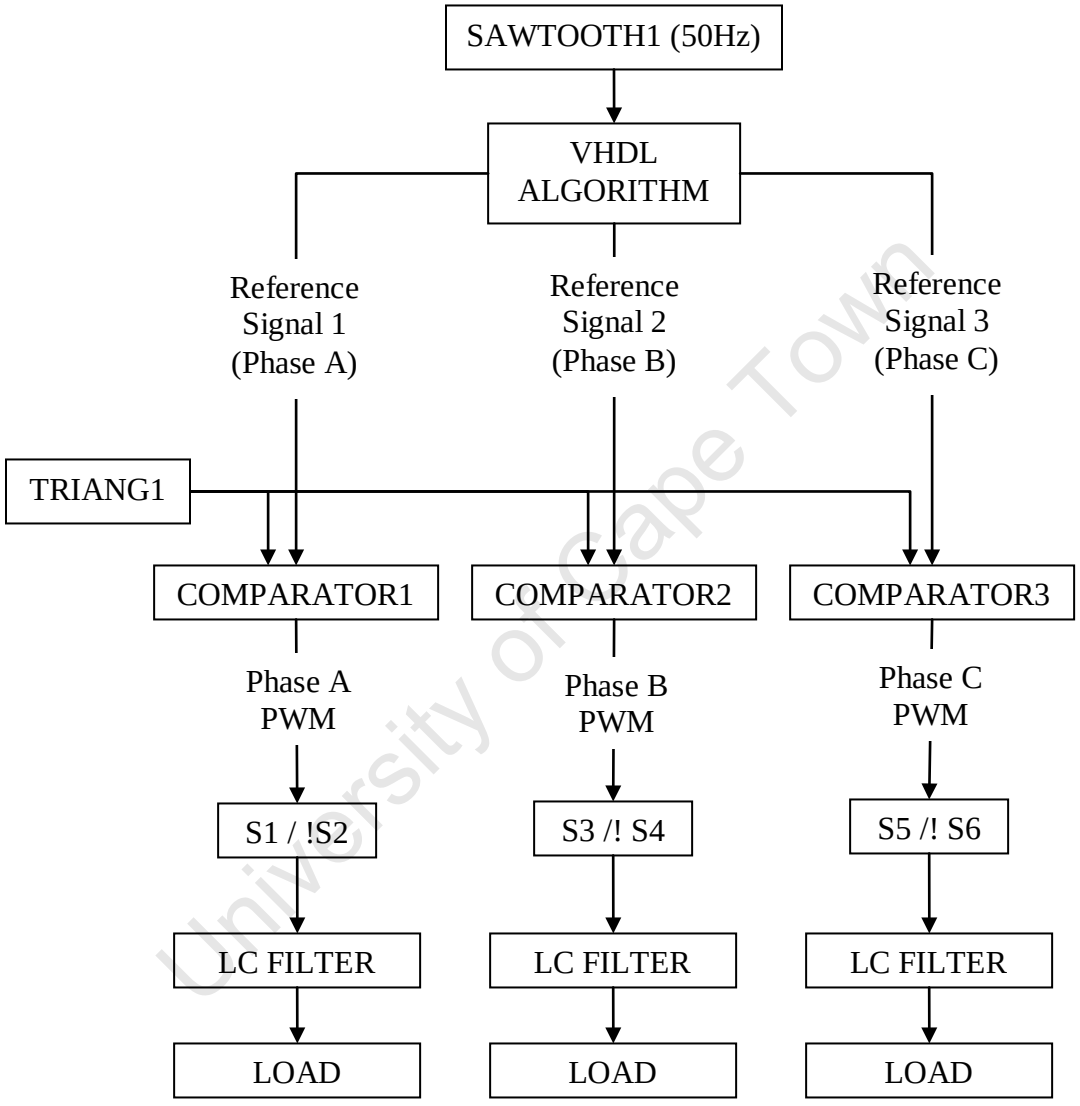


Fig. D2: Simulation System Process Flow Diagram

The VHDL algorithm has SAWTOOTH1 as an input to drive the algorithm with respect to the output reference signals frequency and phase. The VHDL code is

now presented to explain its implementation in this simulation environment. The code was written using the theory presented in chapter 4.

First w is calculated as a ratio of 2π.

```
w:=2*pi*(SAWTOOTH1.VAL/360)
```

M, the modulation index is set to 0.8 where the maximum is 1 or 100% Modulation.

```
M:=0.8
```

Ma is calculated as M of the square root of 3 divided by two. Root 3 over two is the real maximum modulation has discussed in the section on SVPWM and the boundaries of the hexagonal d-q plane.

```
Ma:=M*(root3/2)
```

The reference vector to be generated is composed of a real part and an imaginary part. Both are sinusoidal, 90° out of phase and their composition creates the rotating reference vector on the d-q plane discussed in Chapter 4: SVPWM. Vd and Vq phase and magnitude are built from a sine and cosine waveform being 90° out of phase and scaled by the modulation index. Note that this reference vector composed of Vd and Vq must stay within the modulation boundaries discussed in Chapter 4: SVPWM. In this simulation the resultant reference vector magnitude will be 80% of the maximum magnitude designated by the inscribed circle of the hexagonal plane formed by the 6 space vectors.

```
Vd:=Ma*cos(w);
Vq:=Ma*sin(w);
```

The sector number is calculated by simply dividing the current position in degrees of the reference vector by six, rounding down and adding 1 to give us a number from 1 to 6.

```
Sector := ROUNDDOWN(SAWTOOTH1.VAL/6) + 1;
```

Once the sector number is known 4 constants a, b, c and d and set. These constants are used to simplify the computation represented in Table 4.1 and discussed in Chapter 4: SVPWM.

```
IF(sector=1)
{
    a:=1;
    b:=-1/root3;
    c:=0;
    d:=2/root3;
}
ELSE IF(sector=2)
{
    a:=-1;
    b:=1/root3;
    c:=1;
    d:=1/root3;
}
ELSE IF(sector=3)
{
    a:=0;
    b:=2/root3;
    c:=-1;
    d:=-1/root3;
}
ELSE IF(sector=4)
{
    a:=0;
    b:=-2/root3;
    c:=-1;
    d:=1/root3;
}
ELSE IF(sector=5)
{
    a:=-1;
    b:=-1/root3;
    c:=1;
    d:=-1/root3;
}
ELSE IF(sector=6)
{
    a:=1;
    b:=1/root3;
    c:=0;
    d:=-2/root3;
}
```

Sector 1	**Sector 2**	**Sector 3**
$0 \le wt = \varphi \le \frac{\Pi}{3}$	$\frac{\Pi}{3} \le wt = \varphi \le \frac{2\Pi}{3}$	$\frac{2\Pi}{3} \le wt = \varphi \le \Pi$
$T_1 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + \Pi/6)$ $T_2 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 3\Pi/2)$ $T_0 + T_7 = T_s - T_1 - T_2$	$T_2 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 11\Pi/6)$ $T_3 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 7\Pi/6)$ $T_0 + T_7 = T_s - T_2 - T_3$	$T_3 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 3\Pi/2)$ $T_4 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 5\Pi/6)$ $T_0 + T_7 = T_s - T_3 - T_4$
Sector 4	**Sector 5**	**Sector 6**
$\Pi \le wt = \varphi \le \frac{4\Pi}{3}$	$\frac{4\Pi}{3} \le wt = \varphi \le \frac{5\Pi}{3}$	$\frac{5\Pi}{3} \le wt = \varphi \le \Pi$
$T_4 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 7\Pi/6)$ $T_5 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + \Pi/2)$ $T_0 + T_7 = T_s - T_4 - T_5$	$T_5 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 5\Pi/6)$ $T_6 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + \Pi/6)$ $T_0 + T_7 = T_s - T_5 - T_6$	$T_6 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + \Pi/2)$ $T_1 = \frac{2}{\sqrt{3}}\lvert\overset{v}{U}\rvert T_s \cos(\varphi + 11\Pi/6)$ $T_0 + T_7 = T_s - T_6 - T_1$

Table D1: The formulae for calculating the time duration values for the two appropriate adjacent active vectors and the zero vectors in each of the 6 sectors [4].

Once the appropriate constants are chosen for a given sector table 3.1 can be computed with the following code. Note that no cosine functions are used to calculate t1 and t2. Using preset constants a, b, c and d, table 4.1 is simplified maximizing computation speed.

```
t1:=(Vd*a+Vq*b);
t2:=(Vd*c+Vq*d);
```

Now that we have t1 and t2 we can find t0 which is the remaining sum of time spent at zero states 111 and 000. Because t1 and t2 have been calculated as ratios with total time being 1 we simply subtract their values from 1 to get the resultant ratio of time spent in zero states.

```
t0:=1-(t1+t2);
```

Next 3 reference signals are generated using the following code. These signals waveforms are shown in Fig 4.4.

```
Aref:=t0/2+t1+t2;
Bref:=t0/2+t1;
Cref:=t0/2;
```

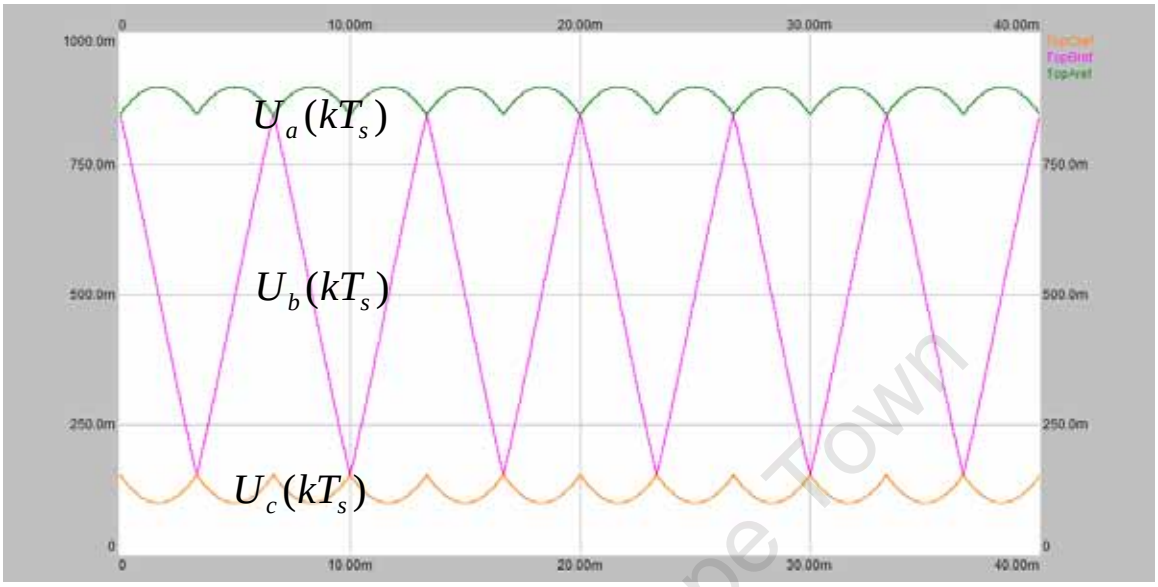


Fig. D3: Reference signals $U_a(kT_s)$, $U_b(kT_s)$, $U_c(kT_s)$. M=0.8

These 3 reference signals or control signals are compared against a high frequency triangular wave at a given switching frequency. In this simulation a 10 kHz triangular wave is fed to the comparators to be compared with a reference signal and a resultant PWM is outputted to drive the associated switches. Once this PWM has been filtered a waveform of the same shape as the control signal will be seen on the output of the filter. The 3 reference signals we would like are not those 3 reference signals shown in Fig 3.3 but rather 3 reference signals made up of a composition of $U_a(kT_s)$, $U_b(kT_s)$, $U_c(kT_s)$. For this reason we need to switch or alternate the reference signals on the inputs of the 3 comparators so that they see a combination composition of $U_a(kT_s)$, $U_b(kT_s)$, $U_c(kT_s)$ on each of their inputs. This will give us new control signals, new PWM outputs and a waveform with the shape desired once filtered.

Fig D4 shows the resultant 3 waveforms seen on the comparators inputs if the inputs to the 3 comparators are alternated as shown in the following code. Note for

each sector we have a new input for each comparators threshold value which is compared to the 10 kHz triangular wave.

```
IF(sector=1)
{
COMP1.THRES:=TopCref;
COMP2.THRES:=TopBref;
COMP3.THRES:=TopAref;
}
ELSE IF(sector=2)
{
COMP1.THRES:=TopBref;
COMP2.THRES:=TopCref;
COMP3.THRES:=TopAref;
}
ELSE IF(sector=3)
{
COMP1.THRES:=TopAref;
COMP2.THRES:=TopCref;
COMP3.THRES:=TopBref;
}
ELSE IF(sector=4)
{
COMP1.THRES:=TopAref;
COMP2.THRES:=TopBref;
COMP3.THRES:=TopCref;
}
ELSE IF(sector=5)
{
COMP1.THRES:=TopBref;
COMP2.THRES:=TopAref;
COMP3.THRES:=TopCref;
}
ELSE IF(sector=6)
{
COMP1.THRES:=TopCref;
COMP2.THRES:=TopAref;
COMP3.THRES:=TopBref;
}
```

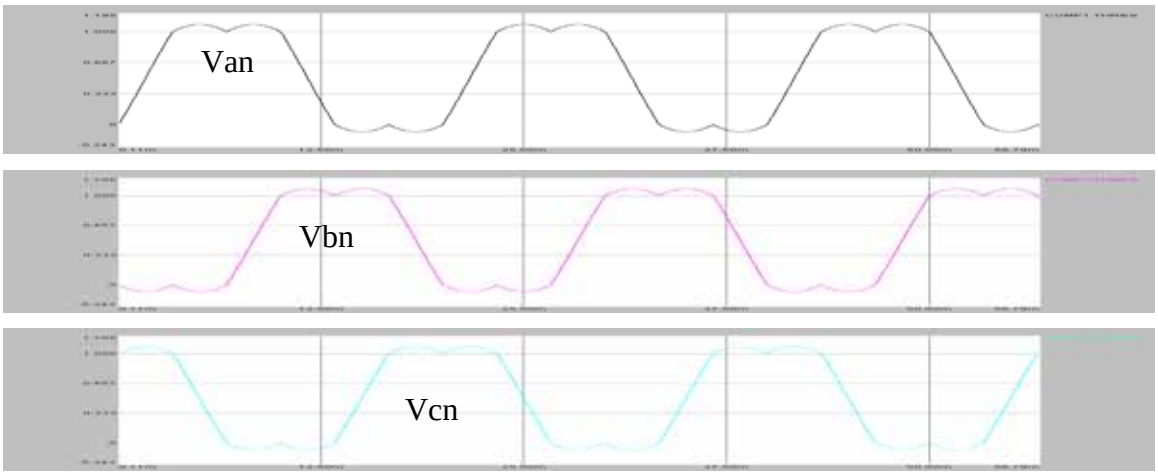


Fig. D4: The 3 control signals that are seen on each of the comparator inputs

On the filter outputs we can see Van, Vbn and Vcn line to neutral voltage waveforms which will be replicas of the control or reference signals applied to the comparators. These are obviously not sinusoidal but their differences are. As a preliminary check the difference of the control signals are computed and plotted to ensure these are the correct control signals for sinusoid generation. The difference signals Vab, Vbc and Vca are shown in fig D5.

```
Vab:=COMP1.THRES -COMP2.THRES;
Vbc:=COMP2.THRES -COMP3.THRES;
Vca:=COMP3.THRES -COMP1.THRES;
```

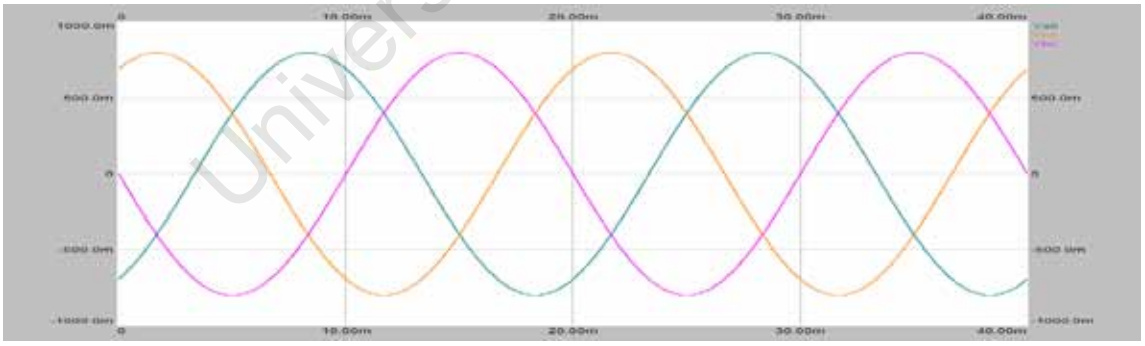


Fig. D5: Sinusoidal difference signals Vab, Vbc and Vca

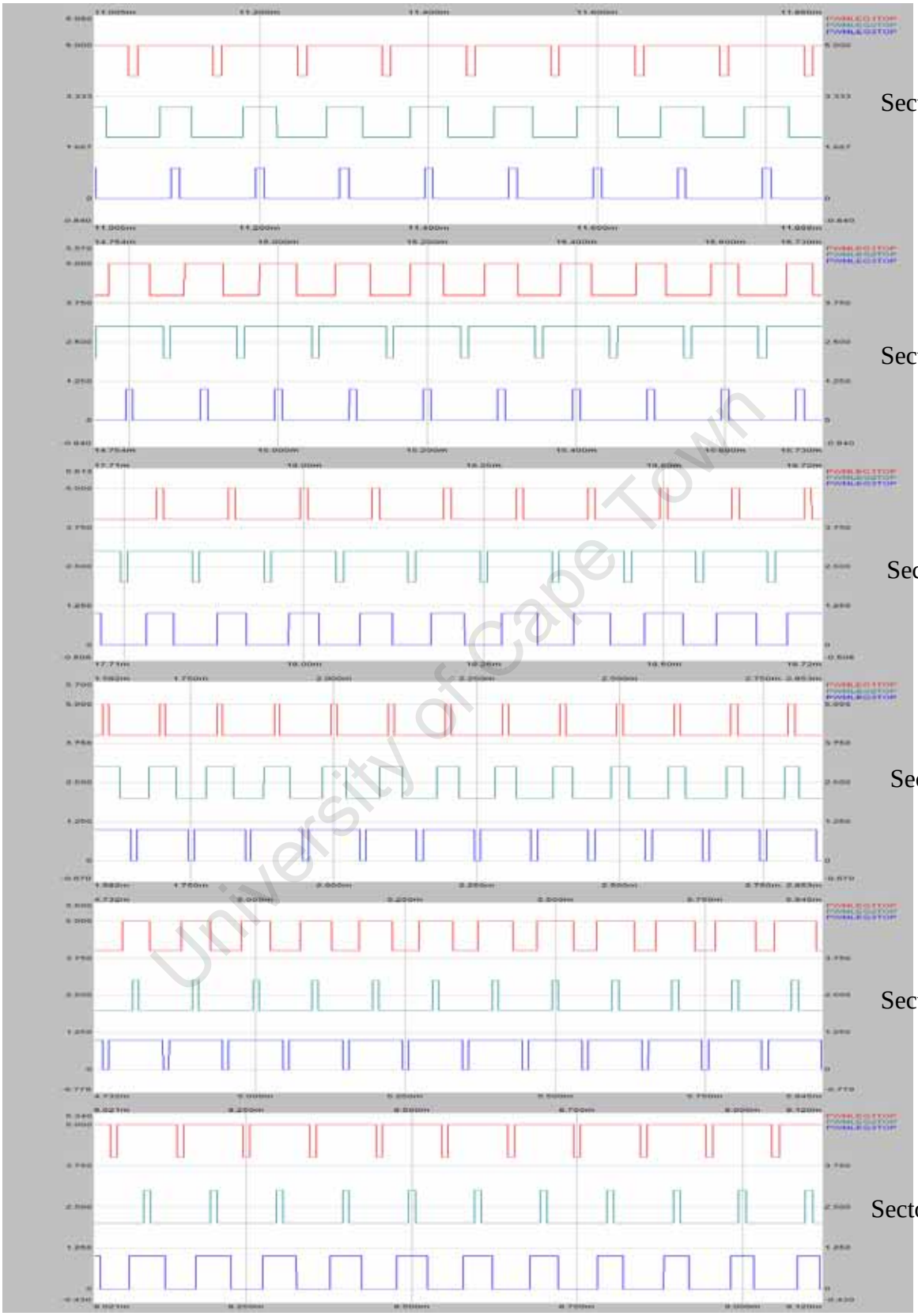


Fig. D6: Switching Patterns for sectors 1 to 6, Red-S1, Green-S3, Blue-S5

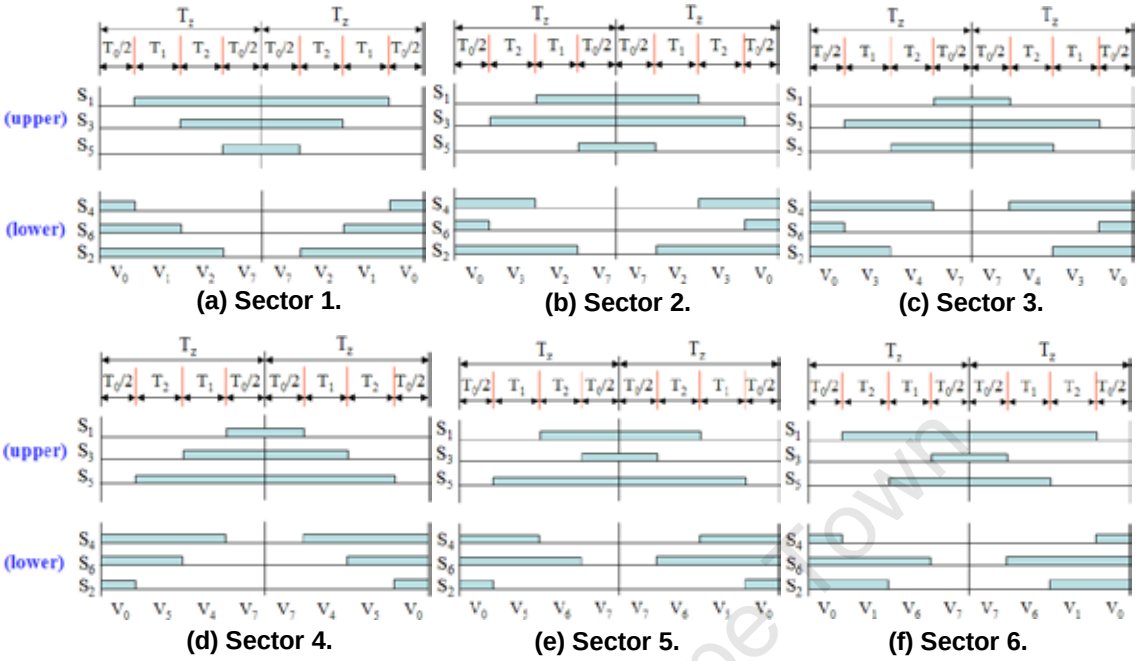


Fig. D7: SVPWM switching patterns at each sector [12]

Comparing Fig D6 with Fig D7 we can see that the simulation has generated the correct switching patterns as stated in conventional SVPWM theory. Only the upper switching pattern has been shown in fig D6 with the lower switching pattern being the compliment.

In conclusion the code presented generates the correct control signals and switching patterns necessary for SVPWM sinusoid generation.

D3. Sinusoidal Output

The PWM signals are filtered through the first order LC filter and the result is three sinusoids 120 ˚ out of phase as shown in fig DE The initial distortion is caused by the capacitive and inductive elements of the system reaching steady state.

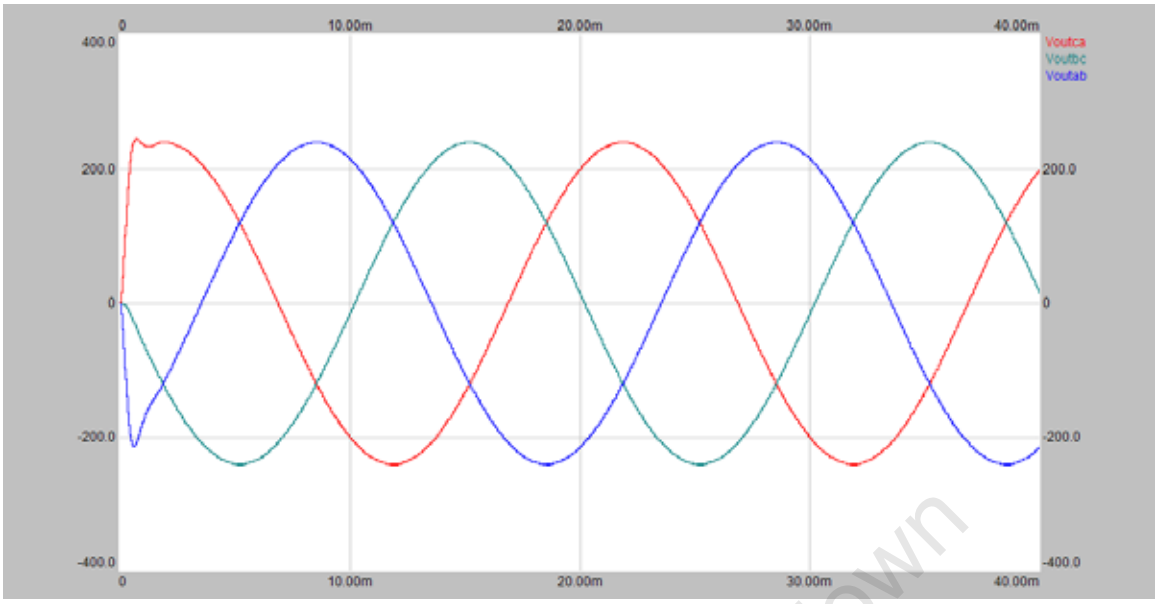


Fig. D8: LC filter output of 3 sinusoids 120˚ out of phase (0.8 modulation)

The source was a steady 300V DC. With a modulation ratio of 0.8, the peak of the line to line sinusoidal output is 240V which is 80% of the DC bus. This is shown in fig DE With a modulation ratio of 1, the peak of the line to line sinusoidal output is 300V which is 100% of the DC bus.

D4. Conclusions

Conventional Sine-Triangle PWM generation utilizes ½ the DC bus for the line to neutral output yielding $\sqrt{3}/2$ the DC bus for the line to line peak voltage. This is 86.6%. SVPWM allows us to use 100% (+15%) of the DC bus utilization at full modulation. The theory presented in Appendix B has been verified by simulation.

D5. Simulator Sampling Settings

Because of the high switching frequency of 10 kHz the simulator default sample rate settings need to be changed to sample sufficiently. The filter characteristics i.e. the characteristics of the capacitors and inductors need a number of samples between taken between switches. As a result we cannot simply sample at twice 10 kHz. In these simulations the optimal sampling configuration was found to be those

setting shown in fig D9. Failure to sample sufficiently will result in distorted filter outputs such as those shown in fig D10.

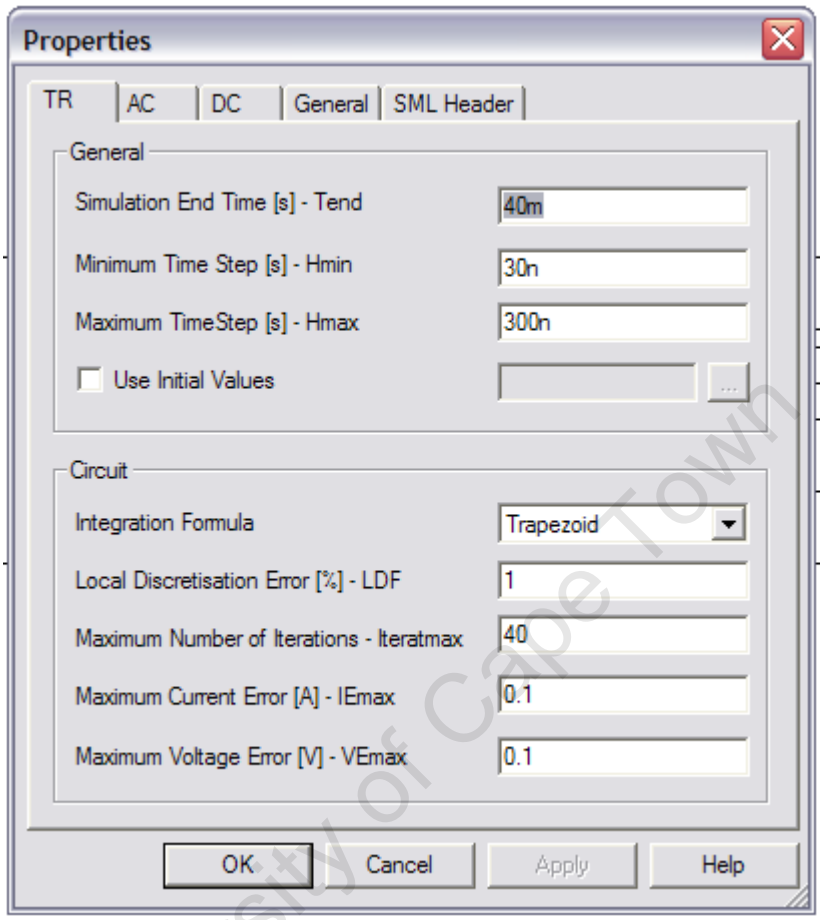


Fig. D9: Simulation Parameter Settings

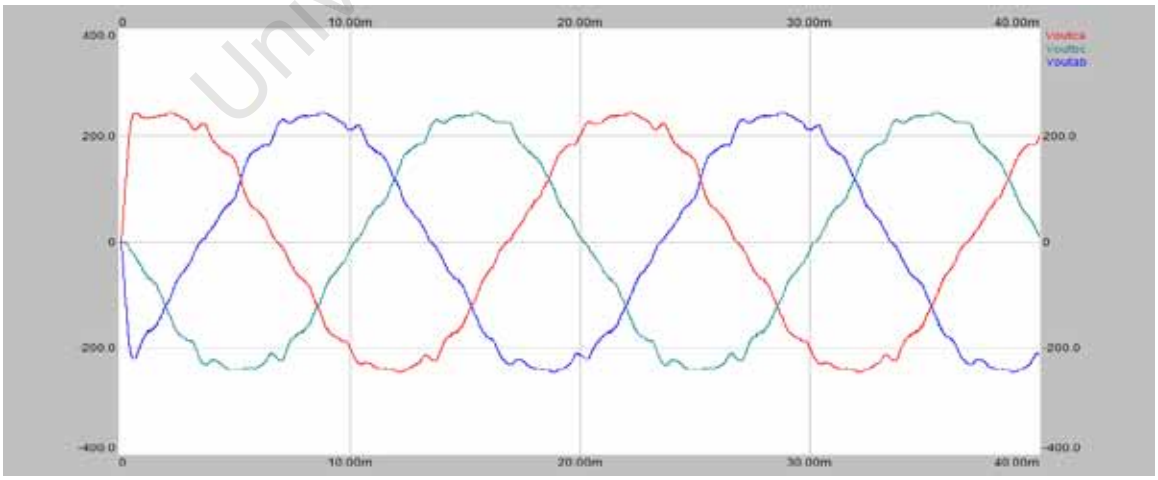


Fig. D10: LC filter output of 3 distorted sinusoids 120˚ out of phase as a result of insufficient sampling by the simulator.

APPENDIX E

Simplorer 7 simulation: Open Loop 3-phase Modified SVPWM for the Z-source inverter

AIM:

The following simulation uses a modified SVPWM technique to take advantage of the z-source impedance network and boost the DC bus voltage while generating a sinusoidal output with all the benefits and characteristics of conventional SVPWM. Primary aims are:

- Add z-source to traditional topology
- Develop and implement a modified SVPWM switching algorithm in VHDL to drive the 6 switches.
- Boost the DC bus using the added shoot though intervals in the switching scheme.

E1. Topology

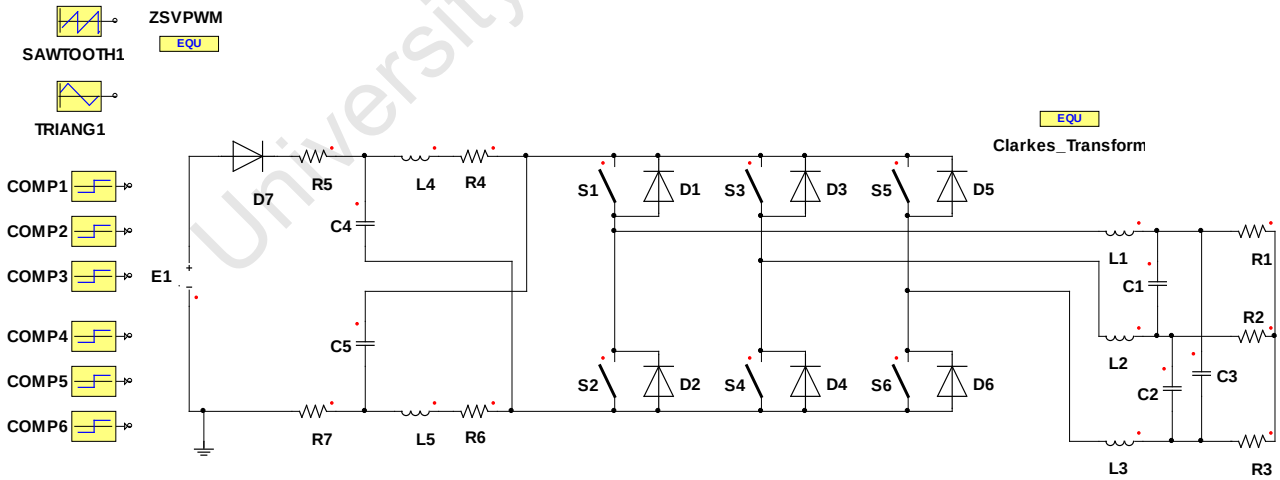


Fig. E1: A 3 phase z-source inverter with filter and load

Simulation component summary:

- E1: DC voltage source (300V)

- D7: Since voltage will be boosted the diode stops reverse current flow back to the source.
- R4, R5, R6 and R7 (1uΩ): Simulator calculations do not converge unless some small impedance is added to the system.
- C4, C5: Capacitive components of z-source impedance network.
- L4, L5: Inductive components of z-source impedance network.
- S1 to S6 and D1 to D6: These 6 ideal switches with anti-parallel diodes form the 3 phase bridge.
- L1, L2 and L3 along with C1, C2 and C3 form a first order 3-phase filter for the 3 output phases.
- R1, R2 and R3 load the system.
- COMP1, COMP2, COMP3, COMP4, COMP5 and COMP6 compare the signal from TRIANG1 with the output of ZSVPWM which generates 6 reference signals using VHDL coded algorithms implementing modified SVPWM.
- The Clarkes_Transform block is used for analysis.
- SAWTOOTH1 is a ramp function running from 0 to 360 at 50Hz. The variable SAWTOOTH1.VAL is used as an input to give the correct frequency and angle of the generated space vector.

E2. Simulation Process Overview

Figure E2 gives an overview of the simulation system. In this simulation there are 6 different reference signals outputted by the VHDL block. These 6 reference signals are compared against a 10 kHz triangular wave to get the 6 different PWM outputs to drive the six switches of the three phase bridge. Unlike traditional SVPWM these PWM outputs will result in overlaps as the top and bottom switches operate independently and not as compliments. When both top and bottom switches are closed at the same time a shoot through occurs charging the z-source impedance network and boosting the DC bus. The boosted DC bus is used to generate a

sinusoid of a larger magnitude possible than that of a traditional voltage source inverter. The results will be presented and discussed and conclusions made.

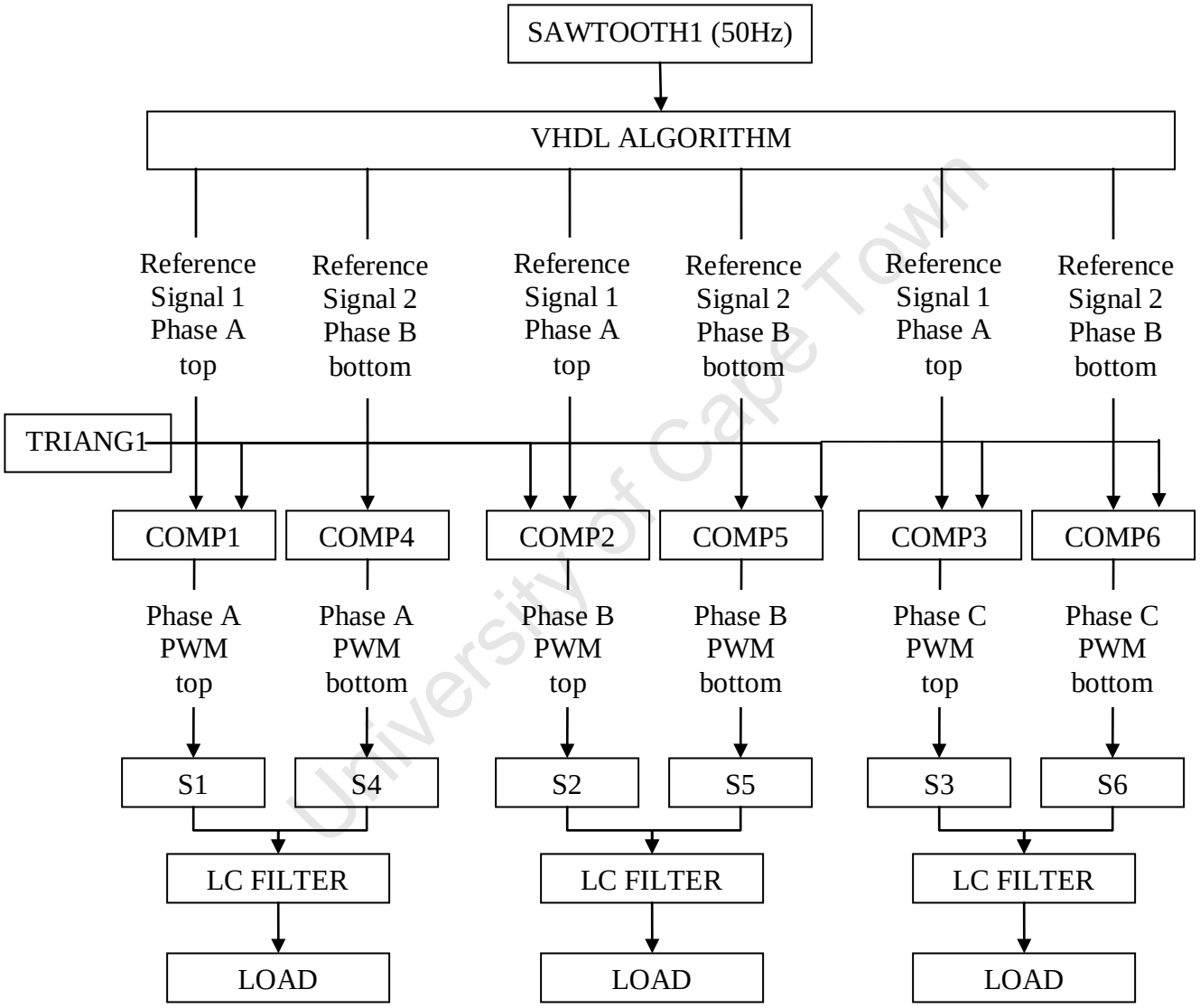


Fig. E2: Simulation System Process Flow Diagram of z-source simulation

E3. The VHDL Algorithm

As show in figure E2 the VHDL algorithm has SAWTOOTH1 as an input to drive the algorithm with respect to the output reference signals frequency and phase. The VHDL code is now presented to explain its implementation in this simulation environment. The code was written using the theory presented in chapter 2.

Most of the code is very similar to that of the last simulation in Appendix D. Any code not discussed in this chapter will have been discussed in Appendix D.

```
w:=2*pi*(SAWTOOTH1.VAL/360)
```

Ta is the total amount of shoot through time. In the second line of code of this segment M is calculated as a ratio of 1 by subtracting 4 thirds of Ta from 1. The ratio 4/3 will be explained later with reference to fig E3.

```
Ta:=0.3;
M:=1-Ta*(4/3);
```

The code below is identical to that of the previous simulation in Appendix D.

```
Ma:=M*(root3/2)
Vd:=Ma*cos(w);
Vq:=Ma*sin(w);
Sector := ROUNDDOWN(SAWTOOTH1.VAL/6) + 1;

IF(sector=1)
{
      a:=1;
      b:=-1/root3;
      c:=0;
      d:=2/root3;
}
ELSE IF(sector=2)
{
      a:=-1;
      b:=1/root3;
      c:=1;
      d:=1/root3;
}
…
…
…
…
}
ELSE IF(sector=6)
{
      a:=1;
      b:=1/root3;
```

```
    c:=0;
    d:=-2/root3;
}

t1:=(Vd*a+Vq*b);
t2:=(Vd*c+Vq*d);
t0:=1-(t1+t2);
```

T1, t2 and t0 are calculated as they would be in the traditional SVPWM algorithm. Now 6 reference signals are to be generated by the next segment of code. It can be seen that a values of Tt=Ta/3 are added to the reference signals to create waveforms that will result in shoot through conditions while not effecting the effectiveness of SVPWM.

```
Tt:=Ta/3;
TopAref:=t0/2+t1+t2+Tt;
TopBref:=t0/2+t1;
TopCref:=t0/2-Tt;
BotAref:=t0/2+t1+t2+2*Tt;
BotBref:=t0/2+t1+Tt;
BotCref:=t0/2;
```

The shoot through periods will be of length Tt. There will be 3 shoot through periods per half switching period and so Tt is divided into 3 Ta's. The addition and subtraction of Ta values from the traditional SVPWM reference signals result in the following changes to the traditional SVPWM switching patterns. Fig E3 reflects the changes made by the last segment of VHDL code. For one sector the PWM for the conventional SVPWM is shown for S1, S2 and S3. Below the PWM for the top and bottom switches are shown for the modified SVPWM.

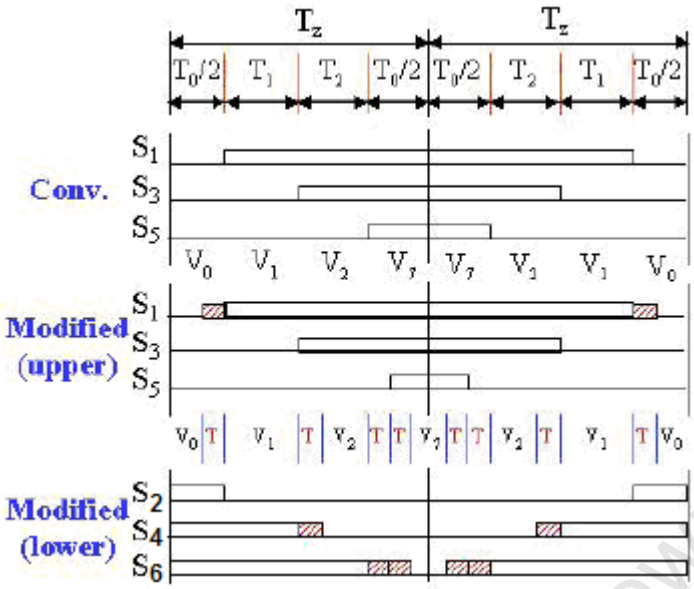


Fig. E3: Modified SVPWM switching patterns for sector 1 [1].

For this sector the reference signals used for each switch are as follows:

S1: TopCref S2: BotCref

S3: TopBref S4: BotBref

S5: TopAref S6: BotAref

Note that for the comparators comparing the bottom switch reference signals when the reference signals is greater than the compared switching triangular wave the output is low. The comparators for the top switches output are high when the reference signals are greater than the compared triangular wave. As a result a top switch reference signal shifted higher will result in less ‘on’ time while the reverse effect is true for a bottom switch reference signal.

From the code and the switching pattern shown in E3 we can see the following:

- SHOOT THROUGH 1:

  Waveform TopCref has been shifted lower by the subtraction of constant Tt. This results in S1 coming on for 2Tt longer over the full switching period. Waveform BotCref has been left as it would be in conventional SVPWM. This results in an overlap for S1 and S2 causing a shoot through of Tt length on the first leg per half period.

- SHOOT THROUGH 2:

Waveform TopBref has been left as it would be in conventional SVPWM. Waveform BotBref has been shifted higher by the addition of constant Tt resulting in S4 staying high for 2Tt longer in a full period. This results in an overlap for S3 and S4 causing a shoot through of Tt length on the second leg per half period.

- SHOOT THROUGH 3:

  Waveform TopAref has been shifted higher by the addition of constant Tt. This results in S5 coming on Tt late and for 2Tt longer over the full switching period. Waveform BotAref has been shifted higher by the addition of 2Tt. This results in S6 staying high for 2Tt longer on a half period or 4Tt over a full period. This results in an overlap for S5 and S6 causing a shoot through of Tt length on the first leg per half period.

Not that there are 2 shoot through times of length Tt for each leg for each period of the switching period. To the DC bus this is seen as six shoot through periods per cycle. As a result the DC bus in this simulation sees 60 kHz switching (6 times more than the 10 kHz triangular wave). As discussed in chapter 5 this reduces the inductor and capacitor requirements of the impedance source. This will be discussed further in later simulations.

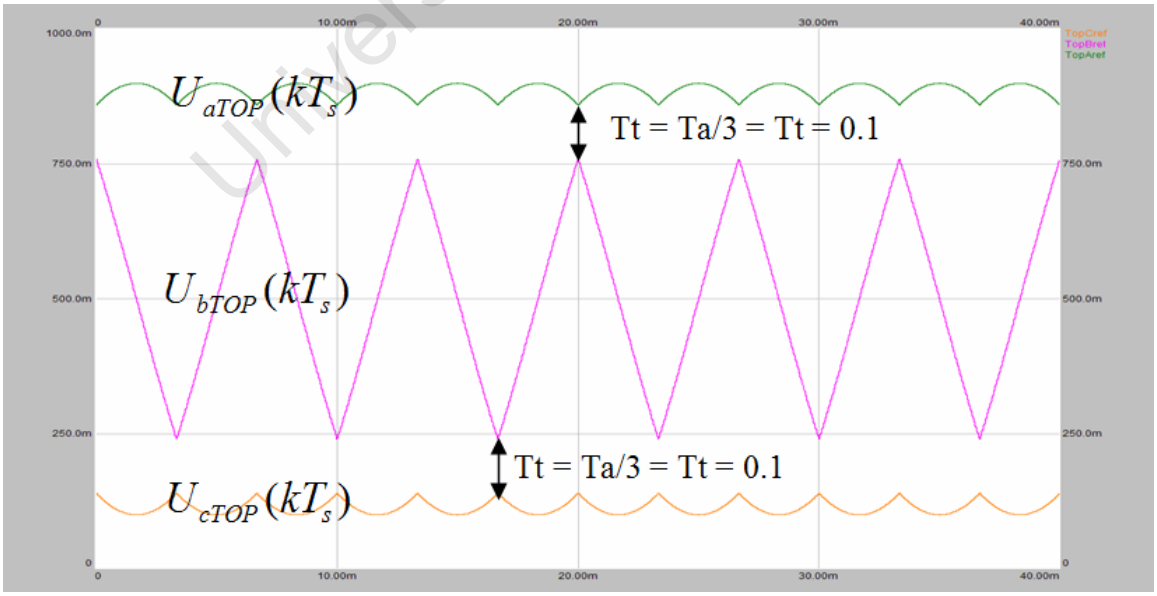


Fig. E4: Reference signals for top switches with Ta=0.3

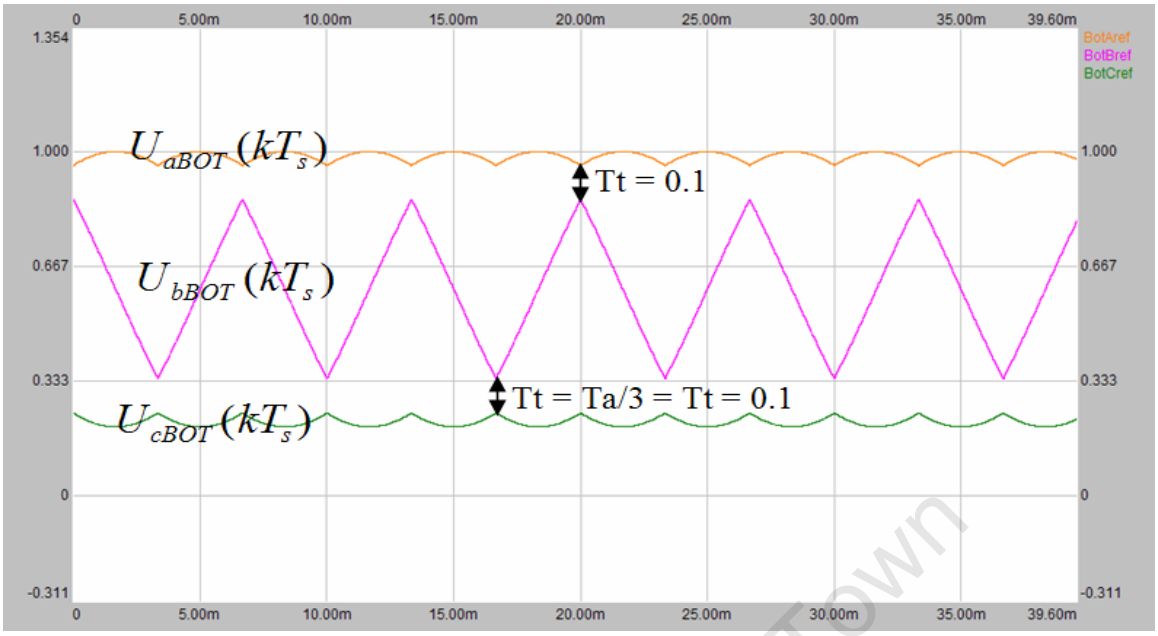


Fig. E5: Reference signals for bottom switches with Ta=0.3

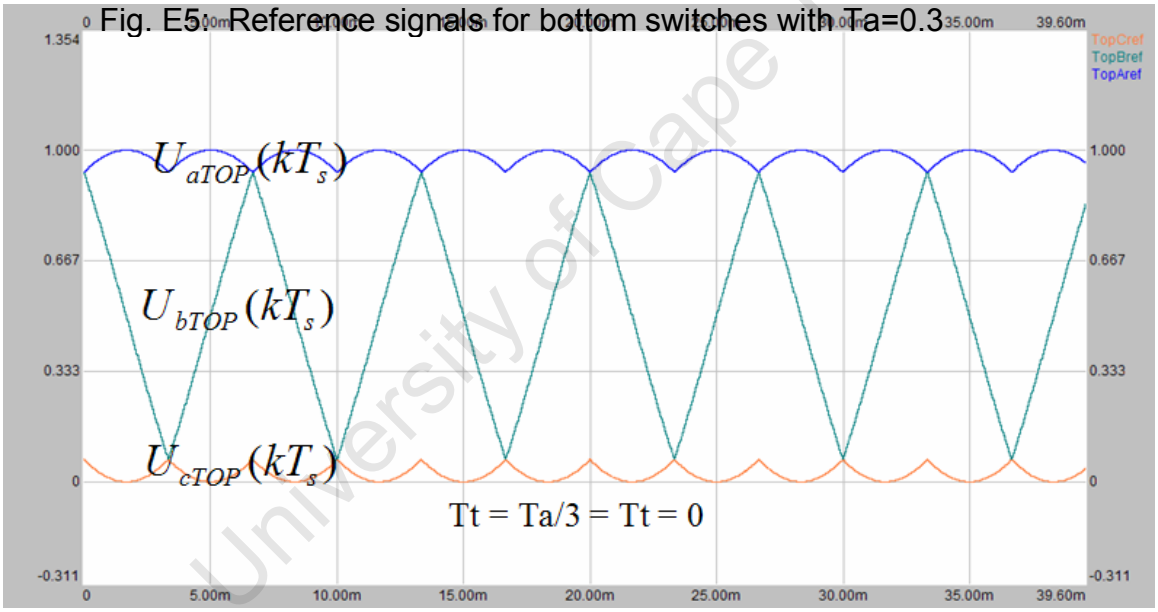


Fig. E6: Reference signals for top or bottom switches with Ta=0

Fig E6 shows the reference signals for the top switches if Ta=0. These will be identical for the bottom switches. This will result in the top switches always being the compliment of the bottom switches as the comparators for the bottom switches output high for the same conditions the comparators for the top switches output low. So Ta=0 will result in conventional SVPWM. Fig E5 shows the reference signals for the bottom switches.

Earlier in the VHDL code the modulation M was calculated as M=1-(4/3)*Ta. It can be seen that $U_{aBOT}(kT_s)$ has a peak of 1. This is on the border of over modulation and if any higher would result in distorted sinusoid generation. For this reason the modulation needs to be calculated in such a way that after the conventional SVPWM reference signals have been modified by shifts up and down the entire band of reference signals needs to be scaled down. In the case of the code it is scaled appropriately first and then shifted. In fig E4, $U_{aBOT}(kT_s)$ is the result of shifting a pre-scaled $U_{aTOP}(kT_s)$ in fig E6 up by 2Tt. t1 and t2 are used to calculate the $U_{aTOP}(kT_s)$ and they depend on M=1-(4/3)*Ta. The simplest way to find this factor of Tt to be subtracted from M is to look at the PWM pattern. In fig E3 S6 is driven by comparator 6 using $U_{aTOP}(kT_s)$ as an input. For maximum boost control t0 (time spend in zero vector state) should come to zero at t1+t2+2Tt peak to maximize utilization of active states and shoot through states. I.e. all zero state time should be used as much as possible. Using more than the available zero state is over modulation. S6 is held on for 2Tt longer in a half period and 4Tt over a full period. If the addition of 2Tt goes over the half period threshold it will be impossible to add the necessary 2Tt of the second half period as there will not be enough zero state left. For this reason 4Tt needs to be subtracted from M to find the appropriate M to give the correct t1, t2 and t0 values. This will ensure that the sum of the time slots in fig E3 (t1+t2+t0+2Tt) never exceeds a half period. M therefore should be calculated as M=1-4*Tt where Tt =Ta/3. Therefore the VHDL code is written as M=1-(4/3)*Ta.

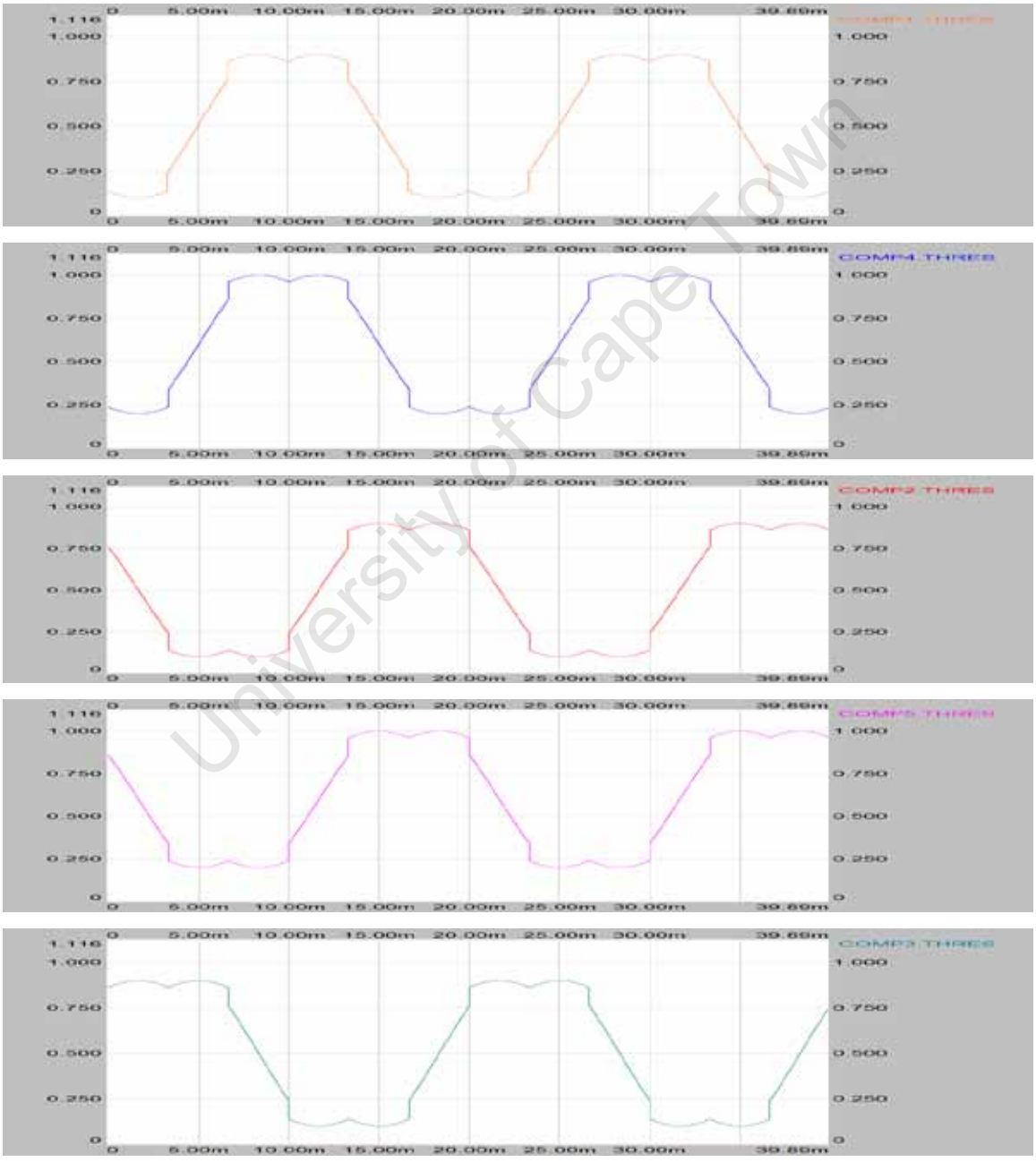
The next segment of code swaps the 6 generated reference signals between the 6 comparator inputs in the same way shown for conventional SVPWM.

```
IF(sector=1)
{
COMP1.THRES:=TopCref;
COMP2.THRES:=TopBref;
COMP3.THRES:=TopAref;

COMP4.THRES:=BotCref;
COMP5.THRES:=BotBref;
```

```
COMP6.THRES:=BotAref;
}
ELSE IF(sector=2)
{
COMP1.THRES:=TopBref;
COMP2.THRES:=TopCref;
COMP3.THRES:=TopAref;

COMP4.THRES:=BotBref;
COMP5.THRES:=BotCref;
COMP6.THRES:=BotAref;
}
ELSE IF(sector=3)
{
COMP1.THRES:=TopAref;
COMP2.THRES:=TopCref;
COMP3.THRES:=TopBref;

COMP4.THRES:=BotAref;
COMP5.THRES:=BotCref;
COMP6.THRES:=BotBref;
}
ELSE IF(sector=4)
{
COMP1.THRES:=TopAref;
COMP2.THRES:=TopBref;
COMP3.THRES:=TopCref;

COMP4.THRES:=BotAref;
COMP5.THRES:=BotBref;
COMP6.THRES:=BotCref;
}
ELSE IF(sector=5)
{
COMP1.THRES:=TopBref;
COMP2.THRES:=TopAref;
COMP3.THRES:=TopCref;

COMP4.THRES:=BotBref;
COMP5.THRES:=BotAref;
COMP6.THRES:=BotCref;
}
ELSE IF(sector=6)
{
COMP1.THRES:=TopCref;
COMP2.THRES:=TopAref;
COMP3.THRES:=TopBref;

COMP4.THRES:=BotCref;
COMP5.THRES:=BotAref;
COMP6.THRES:=BotBref;
}
```


E4. Control Signal Output

This results in the following 6 composite reference signals shown in fig E7. These are the reference signals seen by the comparators and compared with the 10 kHz triangular waveform to give the appropriate PWM. The resultant composite reference signals are shown in fig E7. Note: COMP1.THRES, COMP2.THRES, and COMP3.THRES are control signals for the top switches. COMP4.THRES, COMP5.THRES, and COMP6.THRES are control signals for the bottom switches.



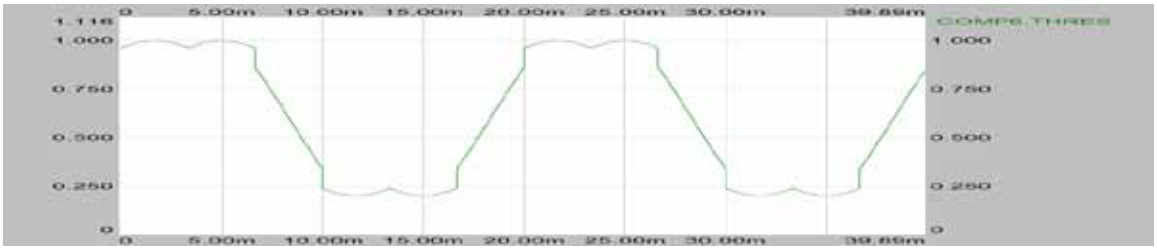


Fig. E7: The 6 generated control signals seen on each comparator input.

If we compare these reference signals to those shown in chapter 7 where conventional SVPWM was implemented we notice that the heart shaped tops and bottoms of the waveform appear after discontinuous jumps. This is best represented in fig E7. Fig E8 shows all control signals together. Note how all discontinuous jumps are of equal length (Tt=0.1=Ta/3).

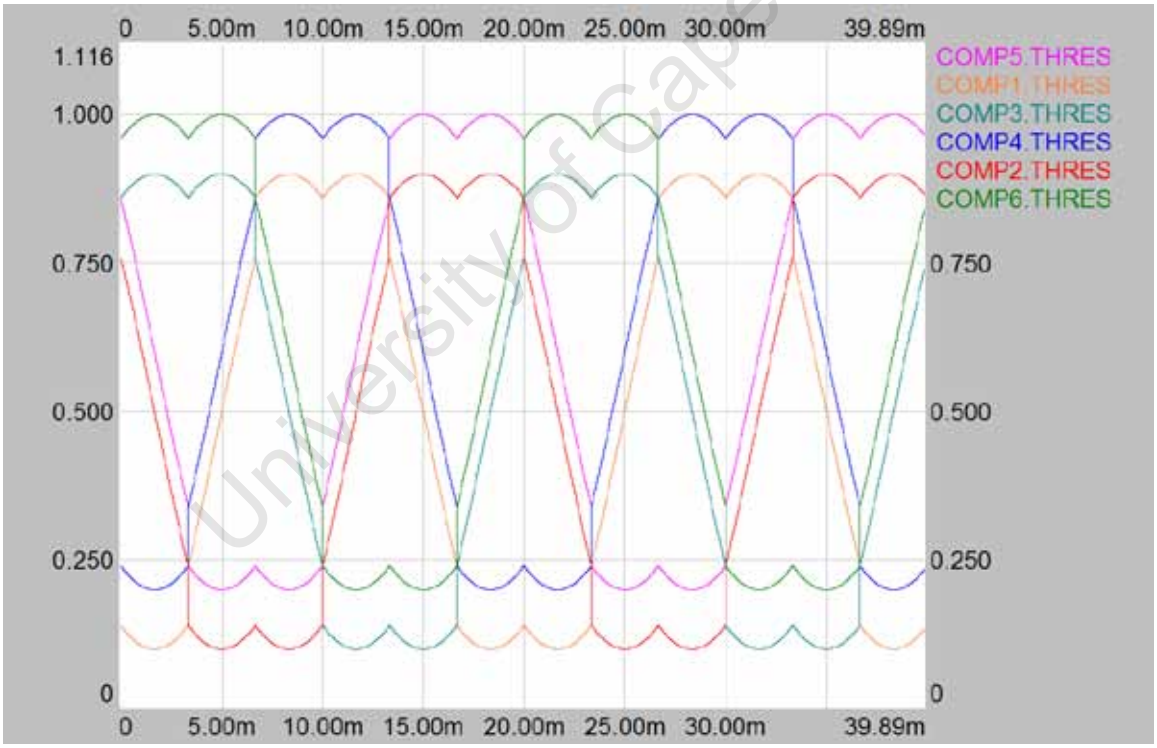


Fig. E8: The 6 generated control signals seen on each comparator input.

In the previous conventional SVPWM simulation the difference between two control signals gave us the shape of our line to line voltage waveform after filtering the PWM. If we plot the difference between 2 control signals in this simulation for different legs we get a waveform like the one shown in fig E9.

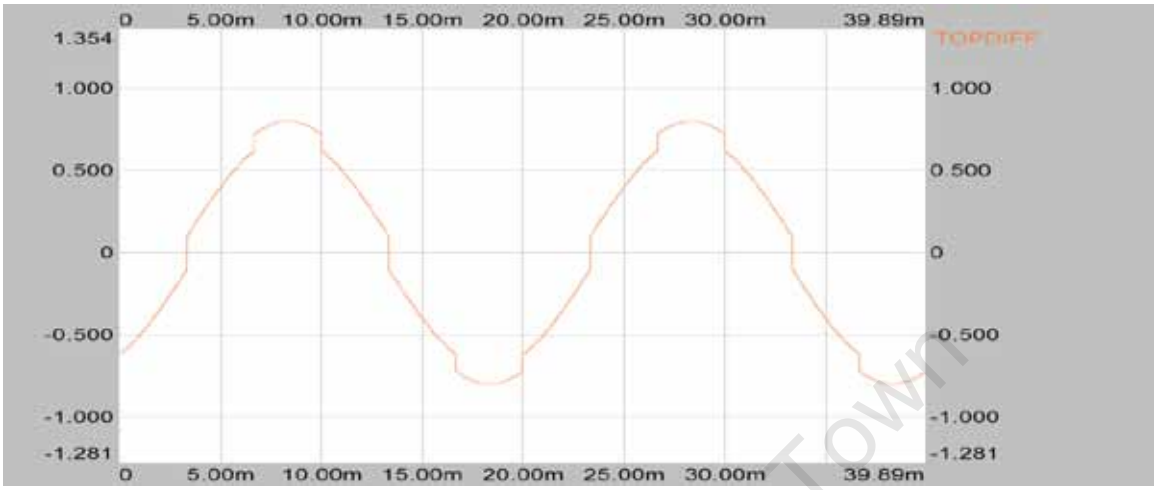


Fig. E9: Difference between S1 and S3 control signal

Fig E9 shows the difference between the control signals used to drive top S1 and S3 of leg 1 and 2 respectively. If the discontinuities were taken out the waveform would be a continuous sinusoid. This does not affect the sinusoidal output after the filter which will be discussed in section E6 of this chapter.

E5. PWM output

The PWM signals in fig E9 represent the switching pattern along with the resultant shoot through from the overlap between top and bottom switches. Offsets have been added to each PWM signal to space them out and present them in the same fashion as the switching pattern shown in fig E3. In fig E9 the switching frequency of the triangular waveform used as the input to the comparators and compared to the control signals runs at 10 kHz. The switching period is then 100uS. Note that in a 100uS period we have 6 shoot through periods. Ta was set as 0.3 for the simulation resulting in a Tt = Ta/3 = 0.1 shoot through ration. 10% (0.1) of 100uS is 10uS. We can see all overlaps 10uS long. Comparing the theoretically expected

results shown in fig E3 with those in fig E9 we can conclude that the VHDL algorithm produces the correct switching pattern.

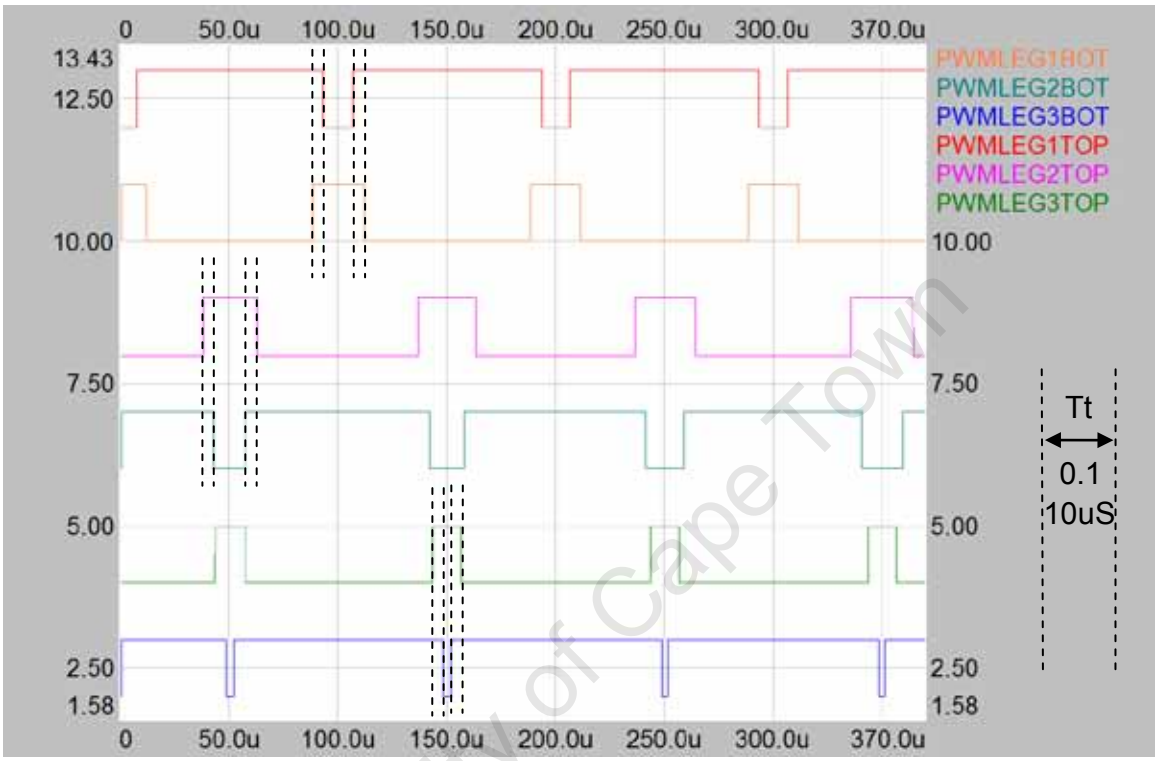


Fig. E10: The 6 PWM signals produced for sector 1.

E6. Sinusoidal Output and DC boost

The PWM signals are filtered through the first order LC filter and the result is three sinusoids 120˚ out of phase as shown in fig E11. The initial distortion is caused by the capacitive and inductive elements of the system reaching steady state.

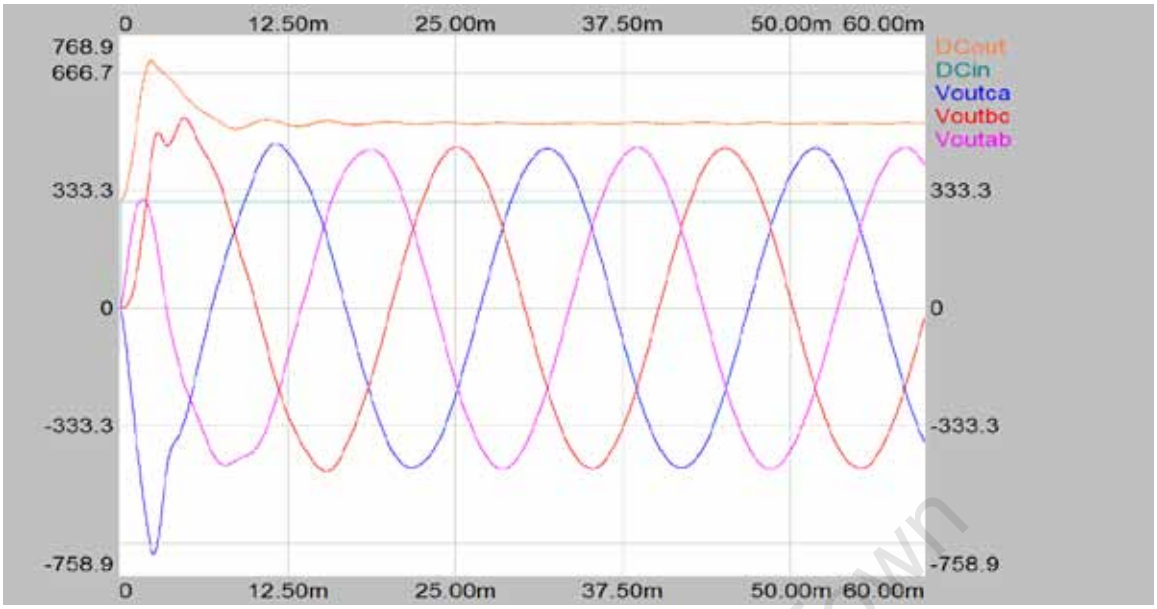


Fig. E11: LC filter output of 3 sinusoids 120˚ out of phase (Ta=0.3)

Also shown in fig E11 are the DC input voltage to the impedance source and the average output voltage seen on C4. We now discuss the values pertaining to fig E11 and how they relate to the theoretical expectations discussed in chapter 2.

We start with a 300V DC voltage source which feeds the impedance network. Equation 14 states that the steady sate voltage on capacitor C4 is calculated as:

$$\frac{T_b}{T_b - T_a} V_{in} = V_C \qquad (14)$$

Substituting simulation values we get:

$$\frac{0.3}{0.7 - 0.3} 300V = 525V$$

Fig E11 shows steady state voltage on C4 to be 525V.

$\hat{v}_i$, the peak DC-link voltage (actual voltage utilized by 3 phase bridge), can be calculated as follows:

$$\hat{v}_i = \frac{T_z}{T_b - T_a} V_{in} \qquad (15)$$

$$\hat{v}_i = \frac{1}{0.7 - 0.3} 300V = 750V$$

This can be seen in fig E11 which shows the voltage waveform on the input to the 3 phase bridge. Fig E13 shows a smaller time section of the wave form to show the sudden transition between zero and the peak voltage as the 3 phase bridge switches in and out of shoot through states.

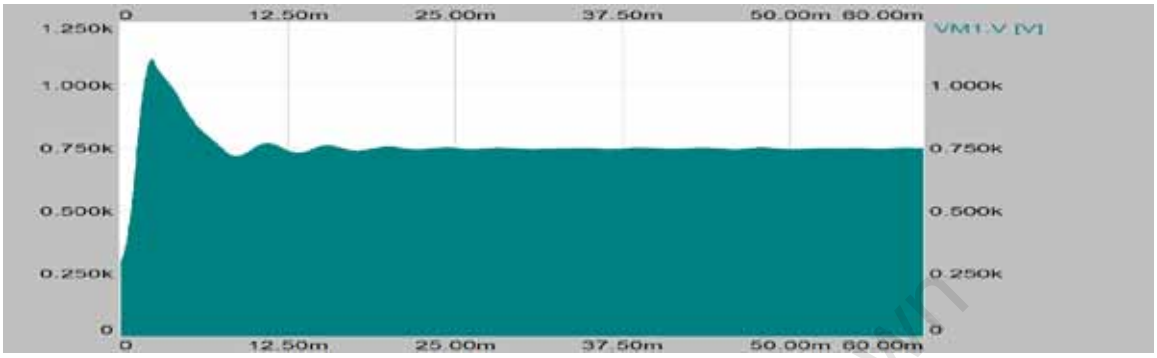


Fig. E12: $\hat{v}_i$, Voltage waveform seen by 3 phase bridge.

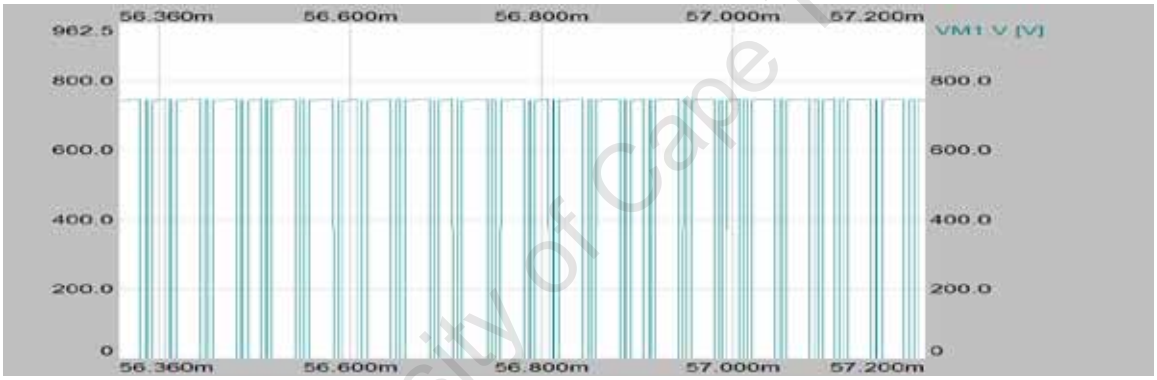


Fig. E13: High time resolution slim section of $\hat{v}_i$, steady state voltage waveform seen by 3 Phase Bridge. Peak voltage = 750V.

$\hat{v}_{AC}$, the peak line to neutral voltage can be calculated as follows:

$$\hat{v}_{AC} = M \cdot \frac{\hat{v}_i}{2} \qquad (17)$$

$$\hat{v}_{AC\ln} = 0.7 \cdot \frac{750V}{2} = 262.5V$$

$$\therefore \hat{v}_{ACll} = 262.5V \cdot \sqrt{3} = 454.66V$$

$\hat{v}_{ACll}$ or DCOUT in fig E11 has a steady state peak of 455V as expected by theory.

E6. Conclusion

Using a 300V DC input a 3 phase sinusoidal output was generated. The sinusoids have a peak voltage of 455V using a shoot through ratio of Ta=0.3. With no shoot through and full modulation (M=1) a maximum peak voltage of 260V can be achieved. In this simulation we have boosted the output peak voltage by 57% above the best case scenario using conventional SVPWM. Simulation results are consistent with the presented theory in chapter 2. Single stage DC boost and sinusoid generation has been achieved using the discussed simulation topology and VHDL algorithms.

APPENDIX F

Simulation: Investigation into voltage boost characteristics

The following simulation focuses on the relationship between the shoot through period length, the input and output voltage over the impedance source and the final 3 phase sinusoidal output.

F1. Shoot through and Voltage Boost

Using the formulae that relate the input DC voltage to the average DC output on C4 in chapter 2 we plot the expected characteristics shown in fig F1 and fig F2. Fig F1 ranges from 0 shoot through to 0.46 and fig F2 ranges from 0.46 to 0.498. The plot is separated because of the exponential nature of the plot. Equation 16 was used to generate the plot.

$$B = \frac{T_z}{T_b - T_a} = \frac{1}{1 - 2 \cdot \frac{T_a}{T_z}} \geq 1 \qquad (16)$$

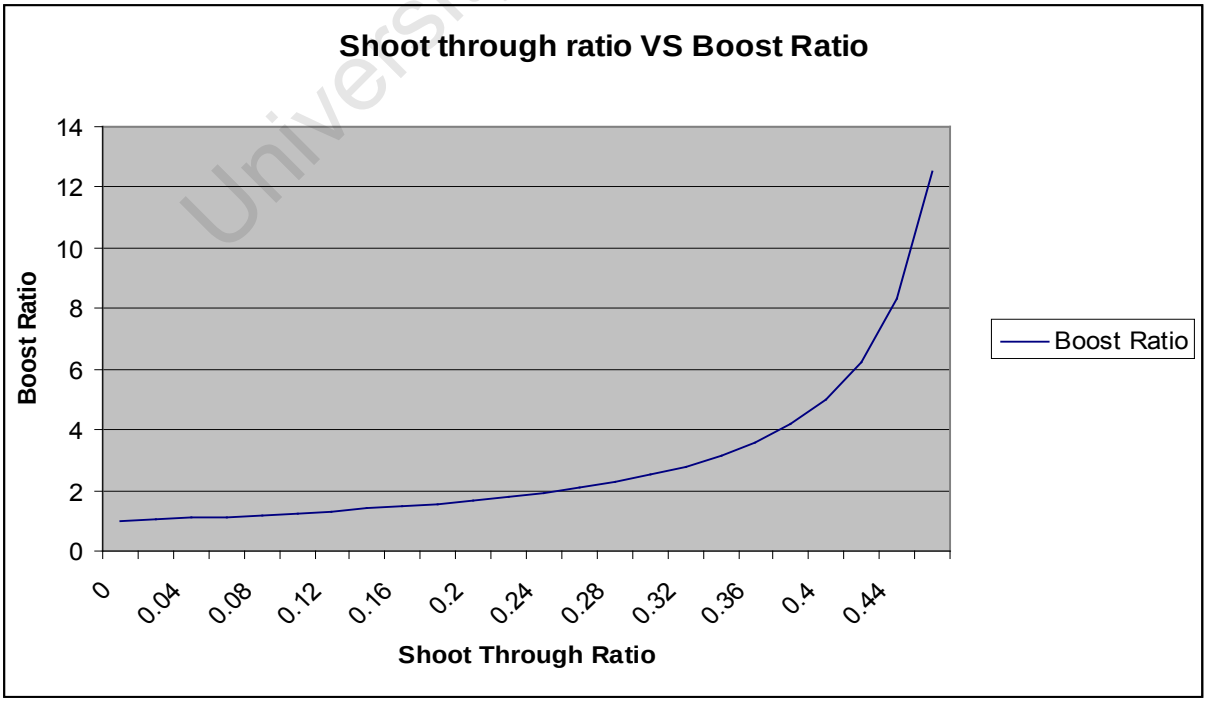


Fig. F1: Plot of Shoot through ratio VS Boost Ratio. Ta = 0:0.46.

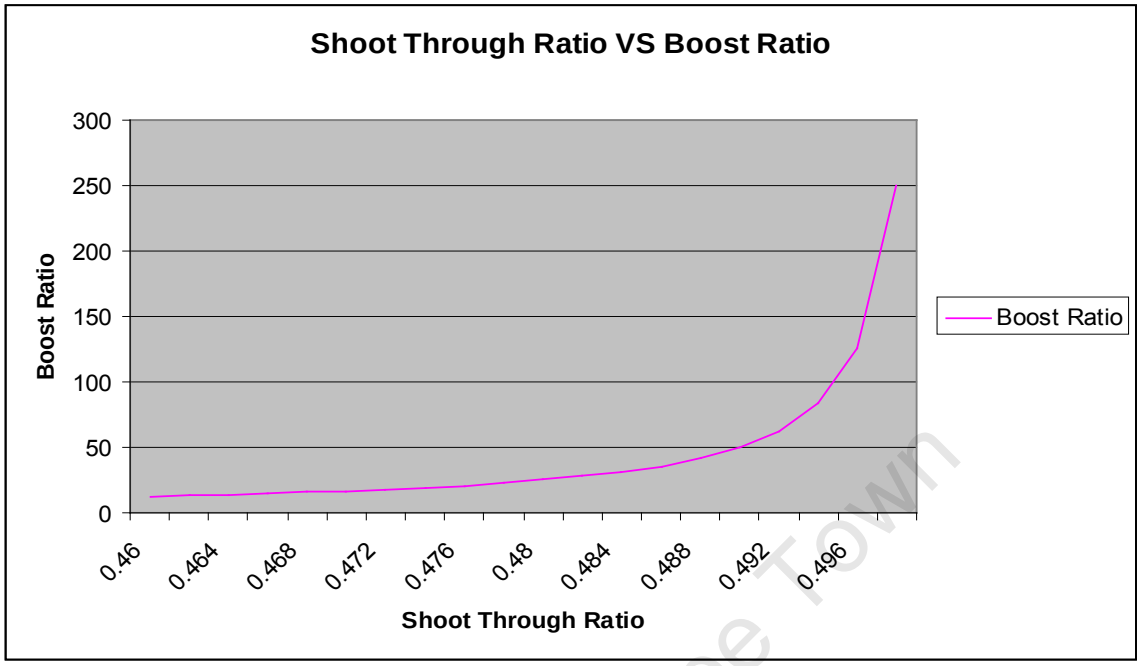
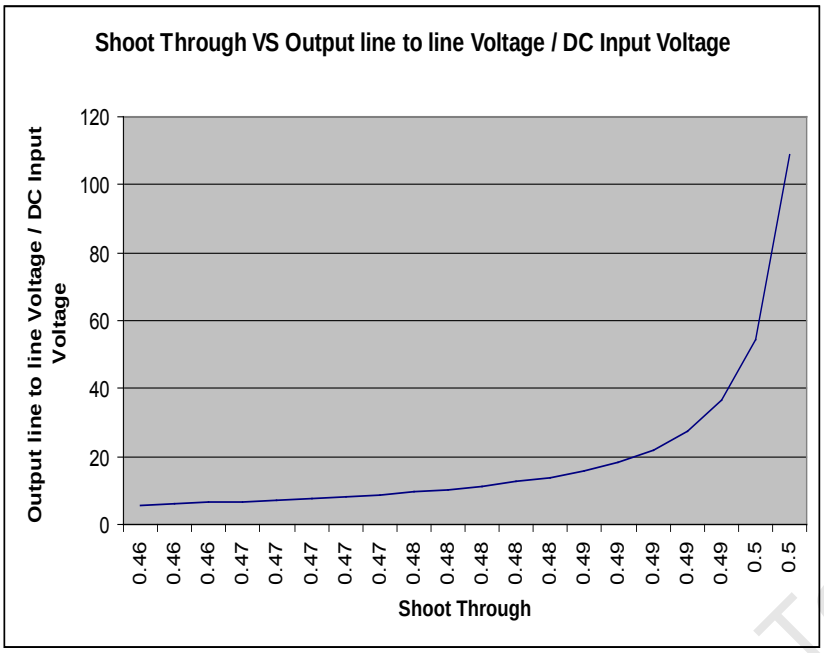


Fig. F2: Plot of Shoot through ratio VS Boost Ratio. Ta = 0.46:0.498.

From equation 16 we can see that as Ta approaches 0.5 of the switching period the DC boost theoretically approaches infinity. At the same time Tb (non-shoot through time) diminishes and affects the resultant modulation ratio. Tb represents M, the modulation ratio. M scales the output sine wave linearly as it would in conventional SVPWM. Tb ranges from 0.5 to 1 as Ta cannot be greater than 0.5 given equation 16. A final output equation relating Ta, Tb, the input DC bus and the output line to line sine wave on any phase can be written as follows.

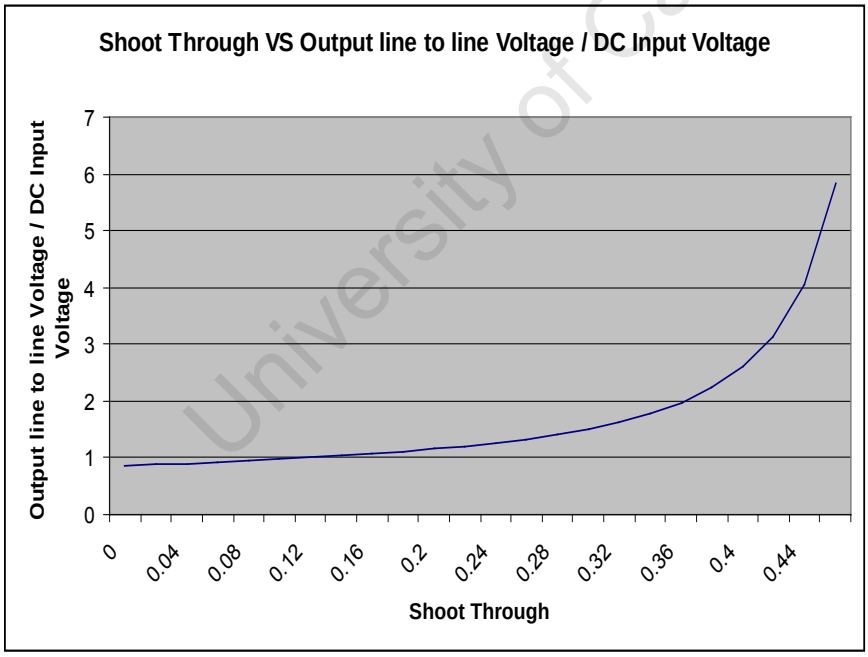
$$\hat{v}_{ACLL} = (1 - T_a)\frac{\sqrt{3}}{2 - 4T_a}V_{in} = M\frac{\sqrt{3}}{2 - 4T_a}V_{in} \qquad (22)$$

Equation 22 yields the plots shown in fig. E3 and E4.



0.46	6.130548253
0.462	6.447078006
0.464	6.800846553
0.466	7.198836169
0.468	7.649891067
0.47	8.165382379
0.472	8.760180046
0.474	9.454110658
0.476	10.27421047
0.478	11.25833025
0.48	12.46114331
0.482	13.96465964
0.484	15.89775206
0.486	18.47520861
0.488	22.0836478
0.49	27.49630657
0.492	36.51740453
0.494	54.55960044
0.496	108.6861882
0.498	

Fig. F3: Plot of Shoot through ratio VS $\hat{v}_{ACLL} : DCin$. Ta = 0.46:0.498.



0	0.866025404
0.02	0.8840676
0.04	0.903678682
0.06	0.92507259
0.08	0.948504014
0.1	0.974278579
0.12	1.002766257
0.14	1.034419232
0.16	1.069796087
0.18	1.109595049
0.2	1.154700538
0.22	1.20624967
0.24	1.265729436
0.26	1.335122498
0.28	1.417132479
0.3	1.515544457
0.32	1.635825763
0.34	1.786177395
0.36	1.979486637
0.38	2.237232293
0.4	2.598076211
0.42	3.139342089
0.44	4.041451884
0.46	5.845671476

Fig. F4: Plot of Shoot through ratio VS $\hat{v}_{ACLL} : DCin$.
Ta = 0:0.46.

APPENDIX G

Z-Source Inductor and Capacitor requirement calculations

The Z-source consists of 2 capacitors and two inductors as shown in fig. 2.1. The combined impedance source topology essentially forms a second order filter which is effective in suppressing voltage and current ripples on the DC bus [7].

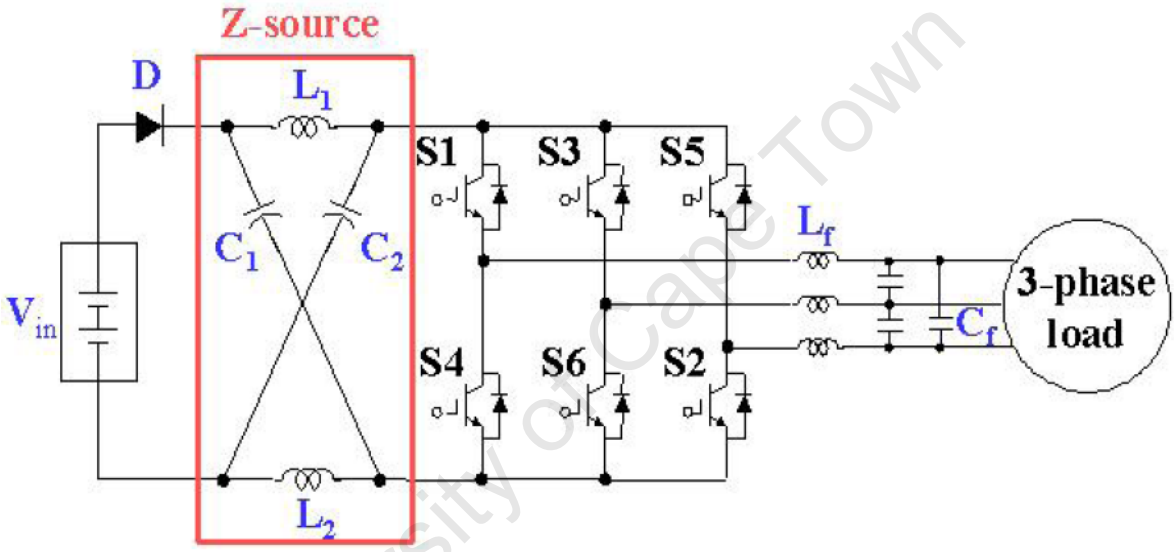


Fig.G1: Total system configuration with Z-Source inverter [1].

Looking at figure 2.1 one should note that the only difference between this z-source inverter configuration is the added inductance. Traditional inverters often use capacitors on the DC bus to stabilize the voltage. As L1 and L2 approach zero the impedance source network becomes the two capacitors (C1 and C2) in parallel. The topology is then essentially that of a traditional voltage source inverter.

From this we can deduct that the capacitor requirements for the z-source impedance network are at worst those required for a traditional voltage source inverter. The fact that the added inductance provides additional

filtering and energy storage means the capacitor requirements will be less than that of a traditional voltage source inverter. In the same way as C1 and C2 approach zero the system resolves into a traditional current source inverter. By the same logic the inductor requirements of the impedance network will be less than that of a traditional current source inverter [7].

Inductor Parameters

The goal on choosing a specific inductance value is to minimize the current ripple seen on the 3-phase bridge. The ripple will distort output sinusoid generation and must therefore be kept to a minimum. The ripple is dependant on the shoot through period. According to the theory associated with voltage boost and shoot through the following maximum shoot though period can be expected.

	THEORETICAL
Tz (uS)	185.1851852
Ta	61.11111111
Tb	103.7037037
Tt	20.37037037
Ta/Tz (% shoot through)	0.33
Tb/Tz = M	0.56
Ma	0.426025404
a	0.42
Vin	24
Average DC boost	1.970588235
C2 average Voltage	47.29411765
K	2.941176471
Peak DC link voltage	70.58823529
MaxSV	61.13120497
LNpeak	34.23347478
INV V LLpeak	59.29411765
KTR (transformer step up)	10
OUT LLpeak	592.9411765
V LNrms	242.0672216
V LLrms	419.2727267

Table G1: Theoretical estimates of shoot through requirements.

Calculations related to shoot through in Chapter 2.4 give the following theoretical data on shoot through requirements for the scope of this project. According to theory we will need 33% shoot through in a period to achieve 240Vrms LN from a 24V dc supply using a KTR (transformer ration) of 1:10. In practice we will only need 230Vrms LN so calculations for 240Vrms LN will give room for voltage losses in a real prototype implementation. The inverter switching cycle runs at 10 kHz. The maximum expected shoot through period is 33% which gives us 33uS. We would like to keep the inductor current ripple to 5% at maximum power output. We wish to deliver a maximum of 2 kW from our prototype at 230Vrms LN supplied from a 24V battery. Maximum current is around 80bAmps so we would want a maximum current ripple of 4 Amps. Given the conditions above and equation G-1:

$$L = \frac{Vdc * Ta}{Iripple} \qquad \text{(G-1)}$$

$$L = \frac{24 * 33 * 10^{-6}}{4} = 198\mu H$$

Two identical inductors are called for by theory. We therefore use a cross coupled dual inductance. In practice it is cheaper to manufacture as it uses the same core. It is also more compact and lighter. Iron core low frequency inductors are used to get the 198uH of inductive bulk but high frequency ferried inductors were put in series to absorb the high frequency component before it causes inefficiencies in the low frequency iron cores. The topology is discussed in Chapter 4.3.

Capacitor Parameters

To minimise the DC voltage ripple seen by the 3-phase bridge suitable capacitor sizes need to be calculated for the Z-Source. Based on table G1 theoretical estimates the average capacitor voltage during table G1 conditions will be 50V. If we wish to keep the voltage ripple within 1% G-2 will give us:

$$C = \frac{I * Ta}{Vripple} \qquad \text{(G-2)}$$

$$C = \frac{80 * 33 * 10^{-6}}{0.5} = 5280 \mu F$$

To get the bulk of the capacitance electrolytic DC caps were used. High frequency non-electrolytic caps were used is parallel to handle the high frequency ripple. Together in series the combination can minimise the magnitude and the high frequency of the ripple voltage within 1%.

APPENDIX H

For the traditional inverter the PC software sets the following variables:

- PWM Switching frequency (2kHz – 20kHz)
- Output frequency (0-300Hz), was always run at 50Hz but was slowed down for detailed analysis of PWM.
- M, the modulation ratio ranging from 0 to 100% modulation
- Ta, blanking time between PWM transitions for shoot through protection.
- Scope settings for triggering and waveform resolutions.

The PC software monitors the following outputs:

- Sector number (1-6) of the hexagonal space vector plane
- The 3 reference wave forms produced from the DSP algorithm given a phase angle and modulation.
- Variables associated with space vector algorithm (Vd, Vq, t1, t2, t0).
- Battery Voltage.
- DS bus voltage.
- LN voltages for each line.
- rms LN voltages for each line.
- PWM outputs in batch downloads because of resolution limits

Code and Code Commentary is on the attached CD. The interface design can be opened using Visual Studio 6 of higher. A video demonstration of the control and monitoring software is also on the attached CD.

APPENDIX I

C code for the Texas Instruments TMS2812

```
if(lockVrmsFlag) //Voltage locking
{
        if( ((prev_Batt_VPC-Batt_VPC) > 5) || ((Batt_VPC-prev_Batt_VPC) > 5) )
                adjustM=64;
        //Clarke Transform
        //Real:=Voutab - 0.5*Voutbc -0.5*Voutca
        Vd=(s32)(InvL1_V<<1)-InvL2_V-InvL3_V;
        Vd>>=1;
        //Imag:=Voutbc *sqrt(3)/2 - Voutca *sqrt(3)/2
        Vq=( ((s32)(InvL3_V-InvL2_V)) * 56753 )>>16;
        instantRMS= Vd*Vd + Vq*Vq; //Magnitude squared
        instantRMS=(s32)qsqrt((u32)instantRMS); //Magnitude
        instantRMS=(instantRMS*30894)>>16; //RMS

        sumRMS+=instantRMS;
        averageRMSCounter++;

        if(averageRMSCounter>=32)
        {averageRMSCounter=0;
                averageRMS=sumRMS>>5;
                sumRMS=0;
////////////////////////////////////////////////////////////////////////////////
                if((M<=65535)&&(Ta<=0))//If Standard modulation control
                {
                        scope[35]=(s16)(100);
                        if(targetVrms>averageRMS) //under shoot
                        {
                                if(prevOverShoot && (adjustM>1)) //if undershoot and prev overshoot
                                        adjustM>>=1;

                                prevOverShoot=0;
                                M+=adjustM;
                        }
                        else                    //overshoot
                        {
                                if(!prevOverShoot && (adjustM>1)) //if overshoot and prev undershoot
```

```
                    adjustM>>=1;
                prevOverShoot=1;
                M-=adjustM;
        }
        //Ma = M*root3/2; root3/2
        Ma = ((M*28378) >> 15);
}

else // standard modulation plus shoot through
{
scope[35]=(s16)(0);
        if(targetVrms>averageRMS) //under shoot
        {
                if(prevOverShoot && (adjustM>1)) //if undershoot and prev overshoot
                        adjustM>>=1;
                prevOverShoot=0;
                M-=adjustM;
        }
        else                    //overshoot
        {
                if(!prevOverShoot && (adjustM>1)) //if overshoot and prev undershoot
                        adjustM>>=1;

                prevOverShoot=1;
                M+=adjustM;
        }
        if(M<35000)
                M=35000;
        //Ma = M*root3/2; root3/2
        Ma = ((M*28378) >> 15); // Ma = ((M*56753) >> 16);
        Ta = ((56754-Ma)*225)>>8; //((56754-Ma)*3)>>2; 3)>>2

        if(Ta>500000)
                Ta = 0;
        if(Ta>30000)
                Ta = 30000;
        //Tt=Ta/3;
        Tt = ((Ta*21845) >> 16);

        if(M>65535)//put back into normal Mod control
        {
                M=65535;
                Ta=0;
```

```
                                        }
                                }
////////////////////////////////////////////////////////////////////////////////
                        }//end if average ready

        }//end if lock

        if(Ma>56754)
                Ma=56754;
        if(Tt<0)
                Tt=0;
        //work out preRef signal values for a given angle
        spaceVectorAngle += (s16)angleIncrement;

        // qsinlt takes a s16 (num from -32768 to 32767 : -180 deg to 179.99 deg)
        Vd = (s32)( (s32)(Ma >> 3)*(s32)(qcoslt(spaceVectorAngle)) );
        Vq = (s32)( (s32)(Ma >> 3)*(s32)(qsinlt(spaceVectorAngle)) );

        Vd >>= 12;
        Vq >>= 12;


        //Finds Sector using Vd and Vq
        FindSectorSV();

        if(sectorSV==1)
        {
                t1 = Vd-((Vq*37837) >> 16);
                t2 = ((Vq*37837) >> 15);
                //a=1;
                //b=-1/root3;
                //c=0;
                //d=2/root3;
        }
        else if(sectorSV==2)
        {
                t1 = -Vd+((Vq*37837) >> 16);
                t2 = Vd+((Vq*37837) >> 16);
                //a=-1;
                //b=1/root3;
                //c=1;
                //d=1/root3;
        }
```

```
else if(sectorSV==3)
{
		t1 = ((Vq*37837) >> 15);
		t2 = -Vd-((Vq*37837) >> 16);
		//a=0;
		//b=2/root3;
		//c=-1;
		//d=-1/root3;
}
else if(sectorSV==4)
{
		t1 = -((Vq*37837) >> 15);
		t2 = -Vd+((Vq*37837) >> 16);
		//a=0;
		//b=-2/root3;
		//c=-1;
		//d=1/root3;
}
else if(sectorSV==5)
{
		t1 = -Vd-((Vq*37837) >> 16);
		t2 = Vd-((Vq*37837) >> 16);
		//a=-1;
		//b=-1/root3;
		//c=1;
		//d=-1/root3;
}
else if(sectorSV==6)
{
		t1 = Vd+((Vq*37837) >> 16);
		t2 = -((Vq*37837) >> 15);
		//a=1;
		//b=1/root3;
		//c=0;
		//d=-2/root3;

}

//Calculate reference signals
t0=0xFFFF-(t1+t2);//0xFFFF-(t1+t2);
//jc3
if(t0<0)
		t0=0;
```

```
TopAref=t0/2+t1+t2+Tt;
TopBref=t0/2+t1;
TopCref=t0/2-Tt;
BotAref=t0/2+t1+t2+2*Tt;
BotBref=t0/2+t1+Tt;
BotCref=t0/2;


TopAref >>= shift;
TopBref >>= shift;
TopCref >>= shift;
BotAref >>= shift;
BotBref >>= shift;
BotCref >>= shift;

/*PWM configuration
```

	CMPR2			CMPR3		CMPR1			
PWM to driver board		PWM2	PWM1			PWM4	PWM3		
	PWM6	PWM5							
Pin	on WAGO connector	Yel4	NULL	Yel4	NULL			Yel4	
NULL									
Inverter PWM switch		TopA	NULL			TopB	Null		
	TopC	Null							
ACTRA									

						CMPR4			
	CMPR5			CMPR6					
PWM to driver board		PWM8	PWM7			PWM10	PWM9		
	PWM12	PWM11							
Pin	on WAGO connector	Gre4	Yel3			Gre4	Yel3		
	Gre4	Yel3							
Inverter PWM switch		BotA	NULL			BotB	Null		
	BotC	Null							

```
*/
if(sectorSV==1)
{
EvaRegs.CMPR1=(u16)TopCref;
EvaRegs.CMPR2=(u16)TopBref;
EvaRegs.CMPR3=(u16)TopAref;

EvbRegs.CMPR4=(u16)BotCref;
EvbRegs.CMPR5=(u16)BotBref;
```

```
EvbRegs.CMPR6=(u16)BotAref;
}
else if(sectorSV==2)
{
EvaRegs.CMPR1=(u16)TopBref;
EvaRegs.CMPR2=(u16)TopCref;
EvaRegs.CMPR3=(u16)TopAref;

EvbRegs.CMPR4=(u16)BotBref;
EvbRegs.CMPR5=(u16)BotCref;
EvbRegs.CMPR6=(u16)BotAref;
}
else if(sectorSV==3)
{
EvaRegs.CMPR1=(u16)TopAref;
EvaRegs.CMPR2=(u16)TopCref;
EvaRegs.CMPR3=(u16)TopBref;

EvbRegs.CMPR4=(u16)BotAref;
EvbRegs.CMPR5=(u16)BotCref;
EvbRegs.CMPR6=(u16)BotBref;
}
else if(sectorSV==4)
{
EvaRegs.CMPR1=(u16)TopAref;
EvaRegs.CMPR2=(u16)TopBref;
EvaRegs.CMPR3=(u16)TopCref;

EvbRegs.CMPR4=(u16)BotAref;
EvbRegs.CMPR5=(u16)BotBref;
EvbRegs.CMPR6=(u16)BotCref;
}
else if(sectorSV==5)
{
EvaRegs.CMPR1=(u16)TopBref;
EvaRegs.CMPR2=(u16)TopAref;
EvaRegs.CMPR3=(u16)TopCref;

EvbRegs.CMPR4=(u16)BotBref;
EvbRegs.CMPR5=(u16)BotAref;
EvbRegs.CMPR6=(u16)BotCref;

TRIGGER=TRUE;
```

```
}
else if(sectorSV==6)
{
EvaRegs.CMPR1=(u16)TopCref;
EvaRegs.CMPR2=(u16)TopAref;
EvaRegs.CMPR3=(u16)TopBref;

EvbRegs.CMPR4=(u16)BotCref;
EvbRegs.CMPR5=(u16)BotAref;
EvbRegs.CMPR6=(u16)BotBref;



                        //Trigger
if(TRIGGER)//first calc of sector 6, runs every 20ms
{TRIGGER=FALSE;

            scopeCount=0; //reset for new period set

            scopeSpeedDivideCounter++;
            if(scopeSpeedDivideCounter>=scopeDivide)//runs at 20ms X scopeDivide
            {scopeSpeedDivideCounter=0;

                        nextValueCounter++;

                        if(nextValueCounter>resetCount)
                        {
                        nextValueCounter=0;
                        }
            }
}

}//end sector 6 jc4



/*
code runs at 10khz = 0.0001s
wave cycle at 50hz = 0.02s
In one cycle of 50Hz there are 10kHz/50Hz = 200 sampling points
We want to take one of these samples every cycle.
This means it will take 200 cycles to get one cycle of data complete.
This means it take 20ms X 200 = 4 seconds per period
Display samples at 10Hz which gives us 40 points per period
```

```
|----cycle 20ms-----|----cycle 20ms-----|----cycle 20ms-----|----cycle 20ms-----|
*/

if(scopeCount==nextValueCounter)
{

scope[0]=82;
scope[1]=0;
scope[2]=(s16)(M>>2);
scope[3]=(s16)(Ta>>2);
scope[4]=(s16)outputFrequency;
scope[5]=(s16)scopeCount;
scope[6]=(s16)nextValueCounter;
scope[7]=(s16)resetCount;
scope[8]=(s16)scopeDivide;
scope[9]=0;

scope[10]=(s16)EvaRegs.CMPR1;
scope[11]=(s16)EvaRegs.CMPR2;
scope[12]=(s16)EvaRegs.CMPR3;
scope[13]=(s16)EvbRegs.CMPR4;
scope[14]=(s16)EvbRegs.CMPR5;
scope[15]=(s16)EvbRegs.CMPR6;
scope[16]=(s16)sectorSV;
scope[17]=(s16)timer1_per;

scope[20]=InvL1_V;
scope[21]=InvL2_V;
scope[22]=InvL3_V;
scope[23]=InvL1_VRMS;
scope[24]=InvL2_VRMS;
scope[25]=InvL3_VRMS;
scope[26]=Batt_VPC;

scope[27]=(s16)(Vd>>2);
scope[28]=(s16)(Vq>>2);
scope[29]=(s16)(t1>>2);
scope[30]=(s16)(t2>>2);
scope[31]=(s16)(t0>>2);

scope[32]=(s16)(instantRMS);
scope[33]=(s16)(averageRMS);
scope[34]=(s16)(targetVrms);
```

```
		//scope[35]=(s16)();
		scope[36]=(s16)(targetVrms);
		scope[37]=(s16)(targetVrms);
		scope[38]=(s16)(targetVrms);
		scope[39]=(s16)(targetVrms);

		}


			scopeCount++;

////////////////////////////////////////////////////////////////////////////////


		return 0;
}


// this function can be called from an interrupt only, call PWM_output(ena) from routine
//-------------------------------------------------------------------------------

void PWM_output_I(u8 ena){



		//if(((ena == ON) && (pwm_output_ena == OFF)) || ((ena == RESUME) && (pwm_output_ena == ON)))

		if(pwmEnableFlag)
		{//jc5

			pwm_output_ena = ON;

				if(pwmFrequencyUpdateFlag)
				{	pwmFrequencyUpdateFlag = 0;


					//Update Switching Frequency

					DINT;					//disable all interrupts as they will interfere
with the setup
					EALLOW;
```

```
                timer1_per          = 0xFFFF >> shift ;//(u16)(25000000/(u32)Sys_Sw_freq);
                timer3_per = timer1_per;

                // *T1CON                       = 0x0842;      // configure T1CON register
                //EvaRegs.T1CON.bit.SET1PR = 0;          // use own period register
                EvaRegs.T1CON.bit.TECMPR = 1;          // enable timer compare operation
                EvaRegs.T1CON.bit.TCLD10 = 0;          // timer compare reload when 0
                EvaRegs.T1CON.bit.TCLKS10 = 0;         // internal clock is the source
                EvaRegs.T1CON.bit.TENABLE = 1;         // enable timer operation
                //EvaRegs.T1CON.bit.T1SWT1 = 0;        // use own timer enable pin

                EvaRegs.T1CON.bit.TPS = switchingFrequencyPreScaler;// input clock prescaler is 128

                EvaRegs.T1CON.bit.TMODE = 1;           // continuous up-down count mode
                EvaRegs.T1CON.bit.FREE = 1;            // operation is not affected by emulation
suspend
                //EvaRegs.T1CON.bit.SOFT = 0;          // operation is not affected by emulation
suspend

                // Timer 1:
            EvaRegs.T1CNT           = 0x0000;                   // clear timer counter
            EvaRegs.T1PR            = timer1_per;               // set timer period
                //-----------------------------------------------------------------------------

                // configure T3CON register:
                //EvaRegs.T3CON.bit.SET3PR = 0;          // use own period register
                EvbRegs.T3CON.bit.TECMPR = 1;          // enable timer compare operation
                EvbRegs.T3CON.bit.TCLD10 = 0;          // timer compare reload when 0
                EvbRegs.T3CON.bit.TCLKS10 = 0;         // internal clock is the source
                EvbRegs.T3CON.bit.TENABLE = 1;         // enable timer operation
                //EvbRegs.T3CON.bit.T4SWT3 = 0;        // use own timer enable pin
                EvbRegs.T3CON.bit.TPS = switchingFrequencyPreScaler;            // input
clock prescaler is 0
                EvbRegs.T3CON.bit.TMODE = 1;           // continuous up-down count mode
                EvbRegs.T3CON.bit.FREE = 1;            // operation is not affected by emulation
suspend
                //EvbRegs.T3CON.bit.SOFT = 0;          // operation is not affected by emulation
suspend

            // Timer 3:
            EvbRegs.T3CNT           = 0x0000;          // clear timer counter
            EvbRegs.T3PR            = timer3_per;               // set timer period
                //-----------------------------------------------------------------------------

                PieCtrlRegs.PIEIER2.bit.INTx4 = 1;     // EVA TIMER 1
```

```
                                PieCtrlRegs.PIEIER4.bit.INTx4 = 1;          // EVB TIMER 3
                                IER |= 0x000A;

                                EDIS;
                                EINT;
                        }//end frequency update

                EvaRegs.ACTRA.all = 0x0888; // 0000 0100 0100 0100          active low for pwm 2,4,6        old =
0x0666;
                EvbRegs.ACTRB.all = 0x0444; // 0000 1000 1000 1000          active high for pwm 8,10,12     old =
0x0666;



                /* enable compare operation        */

                EvaRegs.COMCONA.bit.CENABLE = ON;

                EvbRegs.COMCONB.bit.CENABLE = ON;

                /* enable PWM pins */

                EvaRegs.COMCONA.bit.FCOMPOE = ON;

                EvbRegs.COMCONB.bit.FCOMPOE = ON;

                /* Top drive is enabled */

                PWM_A_TOP_DIS = ENABLE;

                PWM_B_TOP_DIS = ENABLE;

        } // end pwm enable


        else //if ((ena == OFF) && (pwm_output_ena == ON))

        {
                        pwmDisabledFlag=1;

                        pwm_output_ena = OFF;

                        // disable compare operation

                        EvaRegs.COMCONA.bit.CENABLE = OFF;

                        EvbRegs.COMCONB.bit.CENABLE = OFF;
```

```
                    // disable PWM pins
                    EvaRegs.COMCONA.bit.FCOMPOE = OFF;
                    EvbRegs.COMCONB.bit.FCOMPOE = OFF;
                    EvaRegs.ACTRA.all = 0x0000;
                    EvbRegs.ACTRB.all = 0x0000;
                    // Top drive is disabled
                    PWM_A_TOP_DIS = DISABLE;
                    PWM_B_TOP_DIS = DISABLE;
            }//end else

}//end of pwm_output
```

APPENDIX J

Circuit diagrams

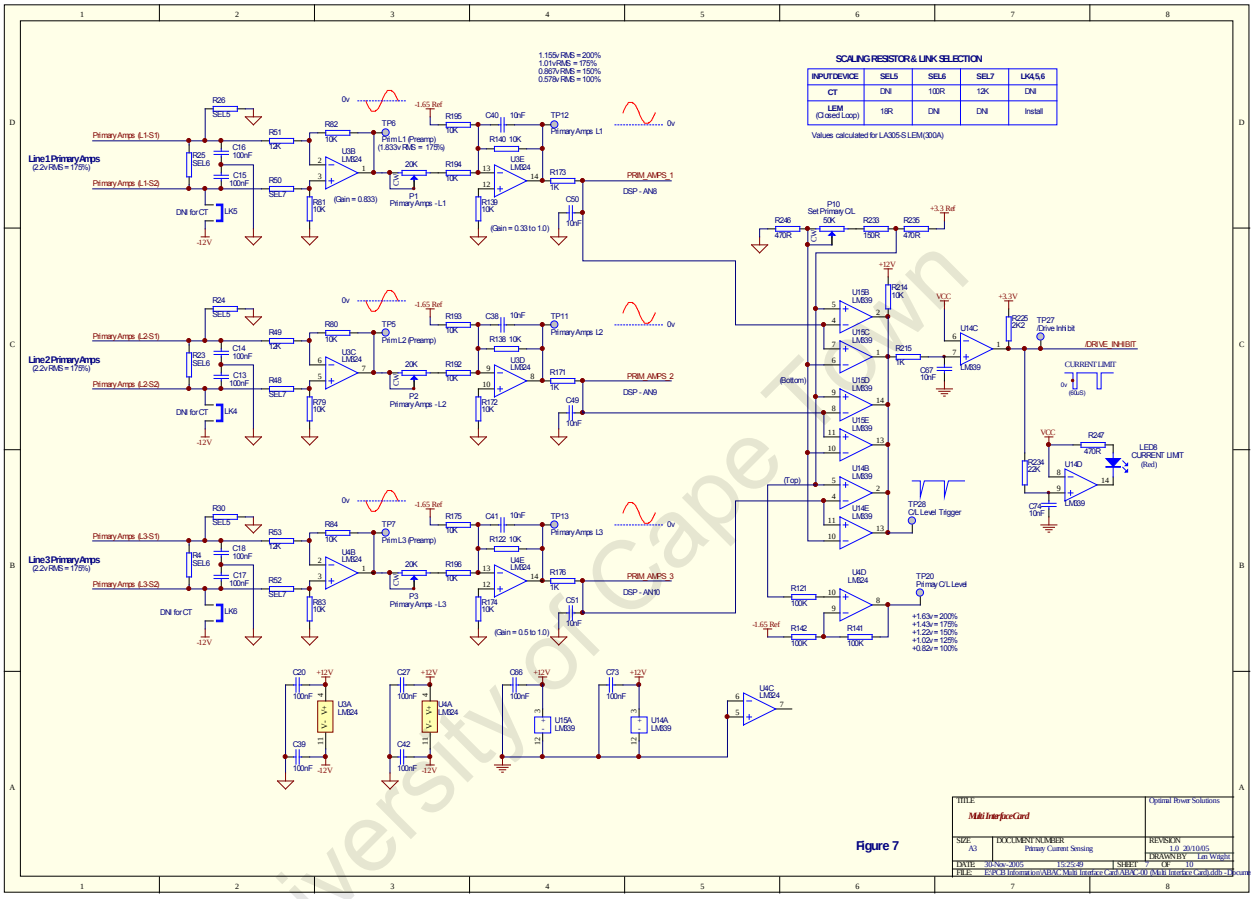


Fig J1: Protection and measurement circuitry

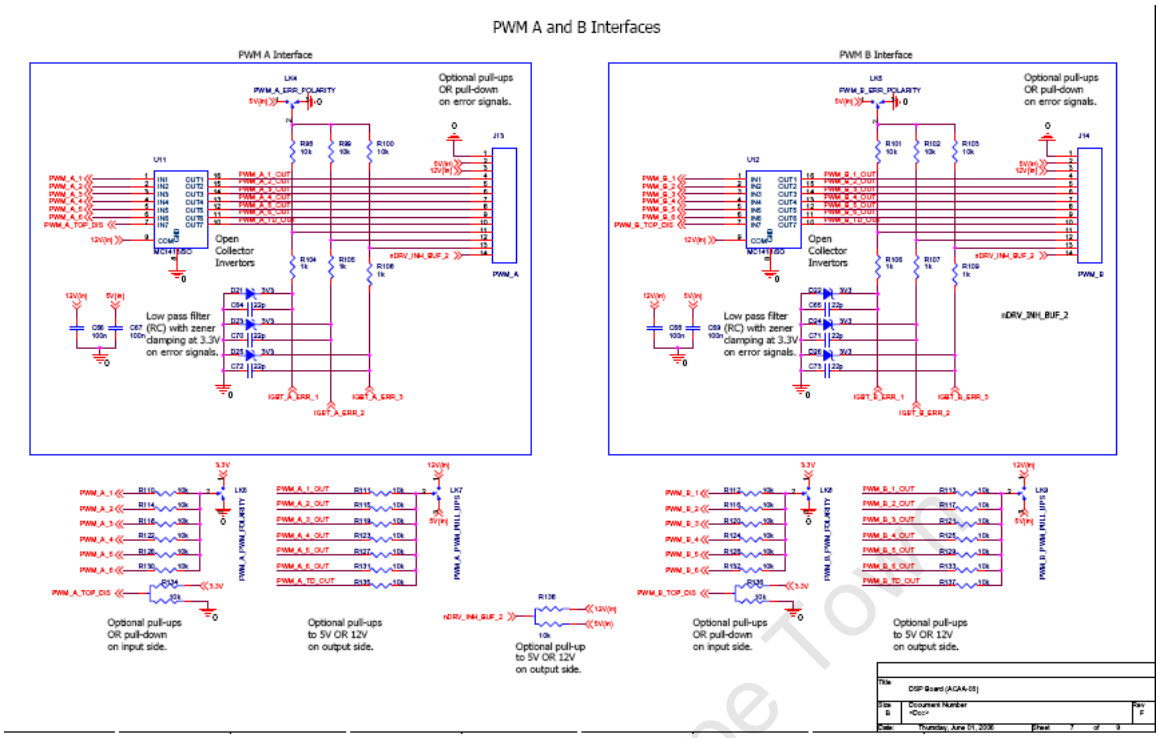


Fig J2: PWM A and B interfaces

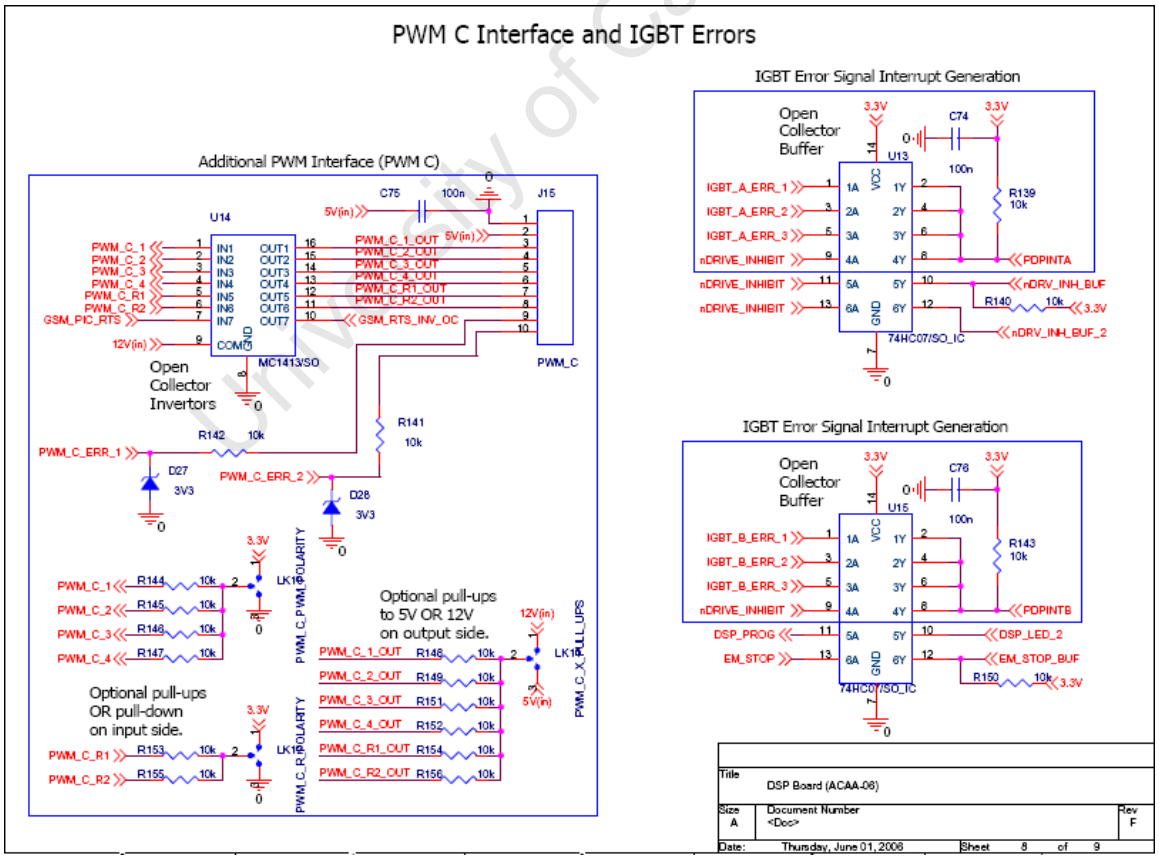


Fig J3: PWM C interface circuitry

APPENDIX K

Phase Locked Loop Controller for 3 phase Grid Locking

Kauro and Blasko, [20] consider the following model shown in Fig X1 for grid locking to the 3 phase mains. This PLL can be implemented entirely through DSP control without the use of any filters.

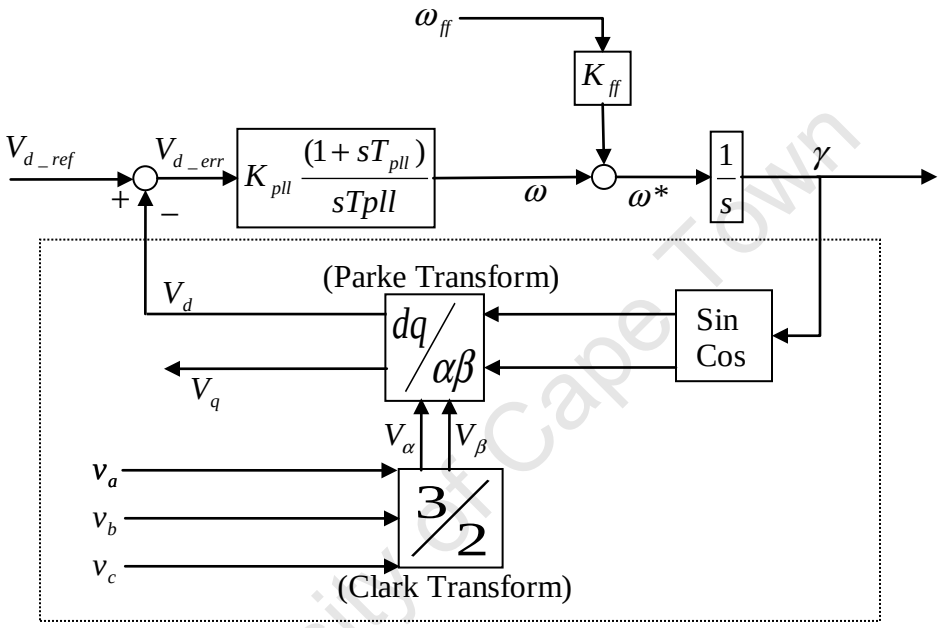


Fig K1: Block diagram of PLL PI-based controller [20].

In fig K1:

In fig K1 the 3 phase grid voltages v_a, v_b & v_c via the Positive sequence Clark Transform produce V_α & V_β. The transformation is as follows:

$$\begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}\begin{pmatrix} v_a \\ v_b \\ v_c \end{pmatrix} \qquad \text{(G-1)}$$

V_α & V_β are orthogonal sinusoids where V_α is real and V_β imaginary. V_β lags V_α by 90° in the case of this positive sequence transform. V_α & V_β on the stationary reference plane are then transformed to V_d & V_q on the rotating reference of the dq plane [21].

Once again the two are orthogonal with V_d & V_q real and imaginary respectively. The Parke transform is as follows:

$$\begin{pmatrix} V_d \\ V_q \end{pmatrix} = \begin{pmatrix} Cos\gamma & Sin\gamma \\ -Sin\gamma & Cos\gamma \end{pmatrix} \begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} \qquad \text{(G-2)}$$

Relative to the dq reference frame rotating at grid frequency, V_d & V_q are stationary. In this rotating reference frame an error signal V_{d_err} can be taken from the difference between the inverter V_{d_ref} and the grid V_d. V_{d_err} The 3 phase inverter phase is angle is γ and is the output of the PI controller given V_{d_err} as an input.

Simplified PLL Equation Derivation

With:

$$\begin{pmatrix} v_a \\ v_b \\ v_c \end{pmatrix} = \begin{pmatrix} V \cdot Sin\theta \\ V \cdot Sin(\theta - \frac{2\pi}{3}) \\ V \cdot Sin(\theta - \frac{4\pi}{3}) \end{pmatrix} \qquad \text{(G-3)}$$

From (G-1), the positive sequence Clarke Transform we have V_α and V_β can be written as follows:

$$\begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \frac{2}{3} \begin{pmatrix} v_a & -\frac{1}{2}v_b & -\frac{1}{2}v_c \\ 0 & -\frac{\sqrt{3}}{2}v_b & \frac{\sqrt{3}}{2}v_c \end{pmatrix} \qquad \text{(G-4.1)}$$

$$V_\alpha = \frac{2}{3}\left(v_a \quad -\frac{1}{2}v_b \quad -\frac{1}{2}v_c \right) \qquad \text{(G-4.2)}$$

$$V_\beta = \frac{2}{3}\left(0 \quad -\frac{\sqrt{3}}{2}v_b \quad \frac{\sqrt{3}}{2}v_c \right) \qquad \text{(G-4.3)}$$

In [21] it is discussed that when the coefficient of the Clarke Transform is $\frac{2}{3}$, V_α would be equal to v_a in magnitude thus (G-4.1) can be written as:

$$\begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \begin{pmatrix} v_a \\ \frac{v_b - v_c}{\sqrt{3}} \end{pmatrix} \qquad \text{(G-5)}$$

Substituting (G-5) into (G-2) we have:

$$\begin{pmatrix} V_d \\ V_q \end{pmatrix} = \begin{pmatrix} Cos\gamma \cdot v_a + Sin\gamma \cdot \dfrac{v_b - v_c}{\sqrt{3}} \\ -Sin\gamma \cdot v_a + Cos\gamma \cdot \dfrac{v_b - v_c}{\sqrt{3}} \end{pmatrix} \qquad \text{(G-6)}$$

Substituting (G-3) we have:

$$\begin{pmatrix} V_d \\ V_q \end{pmatrix} = \begin{pmatrix} Cos\gamma \cdot vSin\theta + \dfrac{v}{\sqrt{3}} Sin\gamma \left(Sin(\theta - \dfrac{2\pi}{3}) - Sin(\theta - \dfrac{4\pi}{3}) \right) \\ -Sin\gamma \cdot vSin\theta + \dfrac{v}{\sqrt{3}} Cos\gamma \left(Sin(\theta - \dfrac{2\pi}{3}) - Sin(\theta - \dfrac{4\pi}{3}) \right) \end{pmatrix} \qquad \text{(G-7)}$$

From [22]:

$$Sin(\varphi - \beta) = Sin\varphi Cos\beta + Cos\varphi Sin\beta \quad \text{(G-8)}$$

Substituting (G-8) into (G-7) we have:

$$\begin{pmatrix} V_d \\ V_q \end{pmatrix} = \begin{pmatrix} Cos\gamma \cdot VSin\theta - \dfrac{V}{\sqrt{3}} Sin\gamma Cos\theta \cdot \sqrt{3} \\ -Sin\gamma \cdot VSin\theta - \dfrac{V}{\sqrt{3}} Cos\gamma Cos\theta \cdot \sqrt{3} \end{pmatrix} \qquad \text{(G-9)}$$

$$\begin{pmatrix} V_d \\ V_q \end{pmatrix} = -V \begin{pmatrix} Sin\gamma Cos\theta - Cos\gamma Sin\theta \\ Cos\gamma Cos\theta + Sin\gamma Sin\theta \end{pmatrix} \qquad \text{(G-10)}$$

Again from [22]:

$$Cos(\varphi - \beta) = Cos\varphi Cos\beta + Sin\varphi Sin\beta \qquad \text{(G-11)}$$

Therefore:

$$\begin{pmatrix} V_d \\ V_q \end{pmatrix} = -V \begin{pmatrix} Sin(\gamma - \theta) \\ Cos(\gamma - \theta) \end{pmatrix} \qquad \text{(G-12)}$$

HSVB-PWM FOR A 3-PHASE Z-SOURCE INVERTER

Jean-Claude Malengret
Department of Electrical Engineering
University of Cape Town, South Africa

Abstract: A 3-phase Z-source inverter has been researched, designed, simulated, built and tested. The purpose of the inverter is to deliver 3-phase 400 VAC from a DC supply that can vary over a range of 20 to 70 Vdc. This is done with a Z-source inverter topology which is a single conversion method with no additional DC to DC boost converter. A novel DSP control algorithm allows the inverter to achieve the following: 1) Run Space Vector Pulse Width Modulation (SV-PWM) for maximum DC bus voltage utilization while boosting the DC bus during zero space vector states using shoot through. 2) Seamless transition between modulation control and modulation / shoot through control. 3) Optimised efficiency and DC bus utilisation using Hybrid Space Vector Boost Pulse Width Modulation (HSVB PWM).

Key Words. Z-source, SVPWM, DC boost, voltage locking.

1. INTRODUCTION

Literature on Z-source inverters started with its realization by Fang Zheng Peng in April 2003 with his paper [3]. He was the first to propose the addition of the Z-source impedance network on the DC side of a 3 phase inverter bridge. This impedance source combined with a modified switching scheme allowed for a single stage DC boost and DC to AC conversion. He then went on to produce in partnership another paper comparing traditional voltage source inverters with the z-source inverter [2]. This paper focused on applications for fuel cell powered vehicles specifically. Jin-Woo Jung, Min Dai, Ali Keyhani in [13] and [1] discuss the characteristics and go on to further approve its use in fuel cell applications. [13] and [1] also explains application using modified space vector PWM (SVPWM) switching techniques. Applications using space vector algorithms have become widely used with the advance in low cost microprocessors capable of running these algorithms at the necessary speeds. The benefits of these SVPWM techniques are presented in [4] and a detailed implementation of traditional SVPWM is investigated in [10]. A 3-phase Z-source inverter has been researched, designed, simulated, built and tested. The purpose of the inverter is to deliver 3-phase 400 VAC from a DC supply that can vary over a range of 20 to 70 Vdc. This is done with a Z-source inverter topology which is a single conversion method with no additional DC to DC boost converter. A novel DSP control algorithm allows the inverter to achieve the following:

- Run Space Vector Pulse Width Modulation (SV-PWM) for maximum DC bus voltage utilization while boosting the DC bus during zero space vector states using shoot through.
- Seamless transition between modulation control and modulation / shoot through control.
- Optimised efficiency and DC bus utilisation using Hybrid Space Vector Boost Pulse Width Modulation (HSVB PWM) which is unique to this paper.

2. Z-SOURCE DC BOOST THEORY

2.1. Topology

Fig.1 shows the z-source topology and its integration into a traditional 3-phase bridge together with output filter and load. The z-source impedance network consists of two capacitors and two inductors where $L_1 = L_2 = L$ and $C_1 = C_2 = C$. As a result the impedance network is symmetrical and therefore [2]:

$$v_{L1} = v_{L2} = v_L$$
$$V_{C1} = V_{C2} = V_C \quad (1)$$

2.2. Z-source Equivalent Circuits and Equations

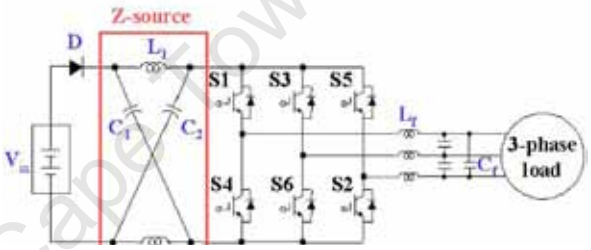


Fig. 1: Z-source impedance network addition to 3-phase voltage source inverter topology [2].

In traditional switching schemes for traditional inverters a shoot through condition is avoided at all costs. If both switches in one phase leg are on simultaneously a shoot through occurs as the DC bus is shorted through the two switches. This will in most cases result in significant damage to the inverter switches and possibly other components such as switch drivers and controllers. Shoot through in a Z-Source inverter is not avoided but used to one's advantage, so as to boost the average DC input voltage to the 3-phase inverter bridge. Shoot through states can be added to the discussed SVPWM switching scheme to boost the DC input voltage by a desired factor as well as generating the necessary PWM output for generation of a sinusoid [1], [2], [3]. Fig. 2a and 2b show the equivalent circuit during a shoot through state and a non-shoot through state respectively. During a shoot through interval we have:

$$v_{L1} = V_{C1} \Leftrightarrow v_L = V_C$$
$$v_f = 2V_{C1} = 2V_C \quad (2)$$
$$v_i = 0$$

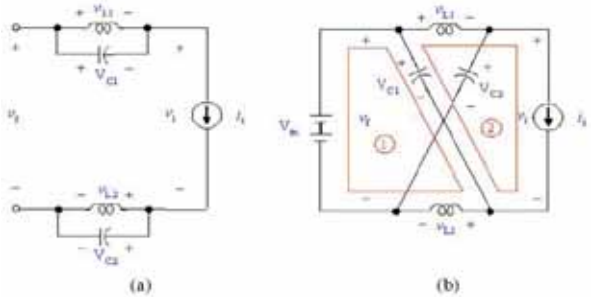


Fig. 2: Equivalent circuit of the Z-Source converter. (a) In the shoot-through state (b) In a non-shoot through state [1].

During the non-shoot through interval shown in fig 2b we have [1], [2]:

$$v_L = V_{in} - V_C$$
$$v_i = V_C - V_L = 2V_C - V_{in} \quad (3)$$

V_{in} is the variable input DC voltage. V_i is the voltage seen on the input of the 3-phase bridge, which has two different states (shoot through and non-shoot through). The switching interval T_z is composed of $T_a + T_b = T_z$. Ta is the total shoot through interval within T_z and Tb is the total non-shoot through interval within T_z. The average voltage over the inductors is zero over one switching cycle [1], [2], [7]. From this and equations (2) and (3) we can derive the following equations:

$$V_{L1} = \overline{v_{L1}} = \int_0^{T_z} v_{L1} dt = \frac{T_a \cdot V_C + T_b \cdot (V_{in} - V_C)}{T_z} = 0$$

$$\Rightarrow V_C = \frac{T_b}{T_b - T_a} V_{in} = \frac{1 - \frac{T_a}{T_z}}{1 - 2 \cdot \frac{T_a}{T_z}} V_{in} \quad (4)$$

Similarly an equation for V_i, the average DC-link voltage (input voltage across the 3-phase bridge) can be derived as follows [1], [2]:

$$V_i = \overline{v_i} = \int_0^{T_z} v_i dt = \frac{T_a \cdot 0 + T_b \cdot (2V_C - V_{in})}{T_z}$$

$$= \frac{T_b}{T_b - T_a} V_{in} = V_C \quad (5)$$

As can be seen from (5), the average DC-link voltage is equal to the capacitor voltage $(V_i = V_C)$. This means we can use the capacitor voltage measurement to regulate the average DC-link voltage. Now using (3) and (4) we can derive an expression for $\hat{v}_i$ (the peak DC-link voltage) [1], [2], [13]:

$$\hat{v}_i = 2V_C - V_{in} = \frac{T_z}{T_b - T_a} V_{in} = B \cdot V_{in} \quad (6)$$

B is the boost factor and can be expressed as [1], [2], [13]:

$$B = \frac{T_z}{T_b - T_a} = \frac{1}{1 - 2 \cdot \frac{T_a}{T_z}} \geq 1 \quad (7)$$

B is always greater or equal to 1. As the shoot through time T_a increases the boost factor B increases and the voltage boost the Z-source impedance network offers increases. Using (6), $\hat{v}_{AC}$ (a peak AC phase voltage of the inverter output) can be expressed as follows [1], [2], [13]:

$$\hat{v}_{AC} = M \cdot \frac{\hat{v}_i}{2} \quad (8)$$

M is the modulation index as discussed in the traditional 3-phase V-source inverter. Using (15) and (17) we can then write:

$$\hat{v}_{AC} = M \cdot B \cdot V_{in} = B_B \cdot V_{in} \quad (9)$$

From (8) it can be seen that the peak phase voltage $\hat{v}_{AC}$ of the inverter output depends on both the boost factor B and the modulation index M. A buck-boost factor B_B from $0 \rightarrow \infty$ can then be chosen to either step up or step down the output voltage where $B_B = M \cdot B$

It should be noted that the shoot through time portion $T_a \propto B$ has a maximum. The available shoot through time is limited by the SVPWM zero-state time discussed in Appendix B [2]. Using more than just the available zero state time will affect the active state vectors of the SVPWM switching cycle and work against the benefits that SVPWM brings. In fact most PWM schemes will not be affected by the added shoot through states as the modified switching scheme still delivers the equivalent zero voltage to the load terminals [7]. The available zero state vector time is determined by the modulation index M [1], [13]. Finally using (1), (4) and (8) we can express the capacitor voltage in terms of the input voltage as [2]:

$$V_{C1} = V_{C2} = V_C = \frac{1 - \frac{T_a}{T_z}}{1 - 2 \cdot \frac{T_a}{T_z}} V_{in} \quad (10)$$

2.3. Z-source modified SV-PWM

The switching algorithm SV-PWM was chosen because of its various advantages such as less harmonic distortion and more efficient use of the supply voltage [1]. Using the derived formulas time quantities T_1, T_2 and T_0 were calculated. T_1 and T_2 were the quantities of time spent in two adjacent active vector states while T_0 was the total time spent in one of the zero vector states (111 and 000). T_a is the total shoot through time. A new shoot though period $T = T_a/3$ is calculated. The value T will be the time duration for one of 3 shoot through intervals within a switching period T_z. In fig. 3 we can see the total switching pattern result for all 6 sectors of the modified SVPWM switching scheme. Notice how there are three time intervals of length T in which both the upper and lower switch in one leg is on causing a shoot through.

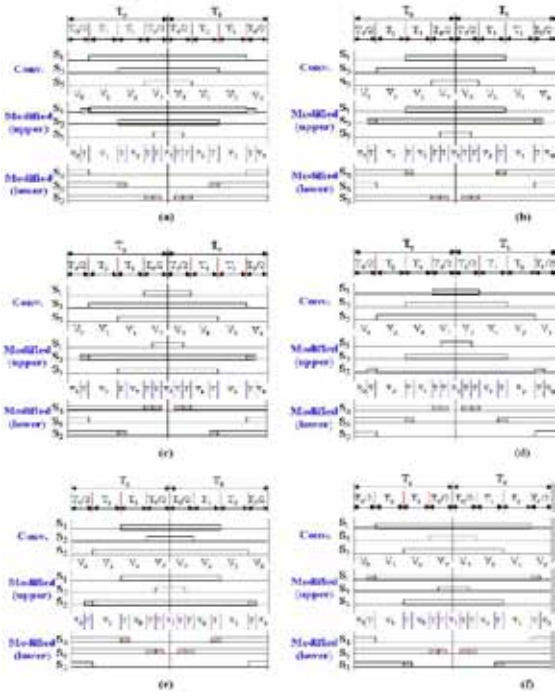


Fig. 3: Modified SVPWM implementation. (a) Sector 1. (b) Sector 2. (c) Sector 3. (d) Sector 4. (e) Sector 5. (f)

Sector	Upper (S1, S3, S5)	Lower (S4, S6, S2)
1	S1 = T1 + T2 + T0 /2 + T S3 = T2 + T0 /2 S5 = T0 /2 - T	S4 = T0 /2 S6 = T1 + T0 /2 + T S2 = T1 + T2 + T0 /2 + 2T
2	S1 = T1 + T0 /2 S3 = T1 + T2 + T0 /2 + T S5 = T0 /2 - T	S4 = T2 + T0 /2 + T S6 = T0 /2 S2 = T1 + T2 + T0 /2 + 2T
3	S1 = T0 /2 - T S3 = T1 + T2 + T0 /2 + T S5 = T2 + T0 /2	S4 = T1 + T2 + T0 /2 + 2T S6 = T0 /2 S2 = T1 + T0 /2 + T
4	S1 = T0 /2 - T S3 = T1 + T0 /2 S5 = T1 + T2 + T0 /2 + T	S4 = T1 + T2 + T0 /2 + 2T S6 = T2 + T0 /2 + T S2 = T0 /2
5	S1 = T2 + T0 /2 S3 = T0 /2 - T S5 = T1 + T2 + T0 /2 + T	S4 = T1 + T0 /2 + T S6 = T1 + T2 + T0 /2 + 2T S2 = T0 /2
6	S1 = T1 + T2 + T0 /2 + T S3 = T0 /2 - T S5 = T1 + T0 /2	S4 = T0 /2 S6 = T1 + T2 + T0 /2 + 2T S2 = T2 + T0 /2 + T

Table 1: Switching time duration of each switch in each sector [1],[3].

For example, in sector 1 we have the 1st T length shoot through interval by turning on switch S1 T seconds early thereby allowing upper S1 and lower S4 to be on simultaneously for a time period T. The 2nd T length shoot through arises from lower S6 be held on an additional T seconds resulting in a T second shoot through when upper S3 turns on. The 3rd last T second shoot through is created by holding lower S2 on an additional 2T seconds to cause a T second overlap with upper S5. The following T_z switching cycle has the mirror configuration of the last. Table 1 shows the switching time duration of each switch in each sector. It is the tabulated form of the switching configuration portrayed in fig. 3. Fig 3 can be used to derive equations for the necessary reference signals that will be fed to the comparators to drive the top and bottom switches. The conventional SVPWM scheme used 3 reference signals and 3 comparators. The reference signals were swapped between the 3 comparators so that the 3 comparators see 3 different control signals on their inputs constructed from 6 pieces of the original reference waveforms. For the Z-source inverter implementation there are now 6 reference waveforms and 6 comparators. Three reference waveforms are alternately fed to 3 comparators for the top switches and 3 fed to 3 more comparators to control the bottom switches. In the conventional SVPWM switching the top switches were always the compliment of the bottom switches except for the precautionary blanking time intervals between transitions. Now the top and bottom switches are driven independently. The 6 reference waveforms are calculated as follows:

$$U_{cTop}(kT_s) = \frac{T_0}{2} - T$$

$$U_{bTop}(kT_s) = \frac{T_0}{2} + T_1 = U_{cTop}(kT_s) + T_2 \qquad (11)$$

$$U_{aTop}(kT_s) = \frac{T_0}{2} + T_1 + T_2 + T = U_{bTop}(kT_s) + T_1 + T$$

$$U_{cBot}(kT_s) = \frac{T_0}{2}$$

$$U_{bBot}(kT_s) = \frac{T_0}{2} + T_1 + T = U_{cBot}(kT_s) + T_1 + T \qquad (12)$$

$$U_{aBot}(kT_s) = \frac{T_0}{2} + T_1 + T_2 + 2T = U_{bBot}(kT_s) + T_2 + T$$

T1 and T2 are the time spent in each of the two adjacent active vector states. T1 is the time associated with the starting active state while T2 is associated with the finishing active state. Note that if $T = T_a/3 = 0$ the modified switching scheme becomes the conventional SVPWM switching scheme.

3. PROTOYTPE DEVELOPEMENT

3.1 Hardware setup

The hardware is a modified 3-phase inverter with control and monitoring instrumentation using custom C# PC software communicating with a Texas Instruments TS2812. An impedance network (z-source) was added on the DC bus side to utilise the shoot through created by the new modified switching scheme. A high frequency power diode was added between the battery positive and z-source positive input. The diode stops current from flowing back into the battery when the Z-source boosts the DC bus voltage up to 400% of the supplied battery voltage. A snubber cap was added between battery positive and battery negative directly onto the diode connection and negative Z-source input to cancel voltage spikes caused by the high frequency shoot through combined with the inductance of the length of wire from the battery. Failure to quell the voltage spike causes the MOSFET semiconductors to exceed their break down voltage and the results is unpredictable switching. Two inductors in conjunction with the 2 banks of caps were used to create the z-source impedance. The capacitor banks are soldered directly onto the bus bars to minimise stray inductance to avoid the same problem mentioned in the third point. The 2 inductor sets each consists of a large low frequency iron core inductor in series with a small high frequency ferrite inductor. The iron core gives

the inductive bulk while the ferrite placed before the iron core suppresses the high frequency component that would cause high losses in the low frequency material iron core inductance.

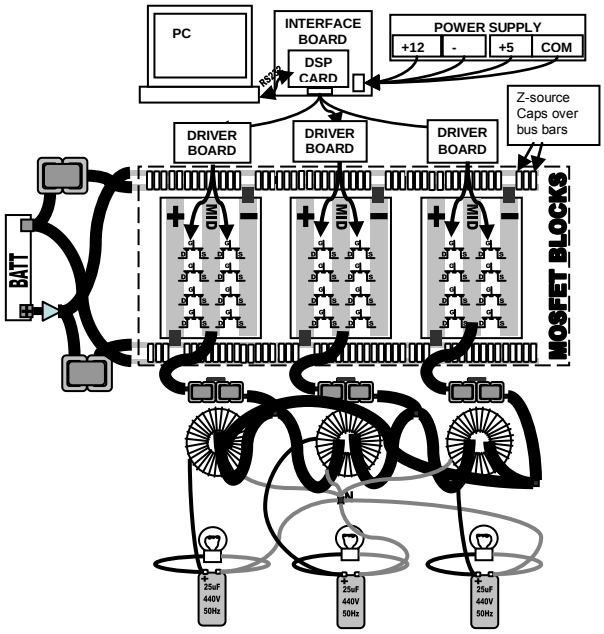


Fig. 4: Layout of 3-phase Z-source inverter

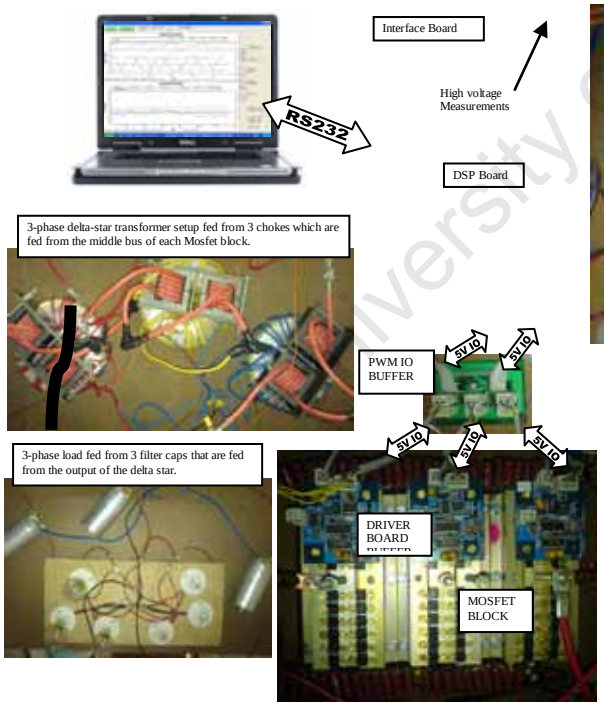


Fig. 5: Photo's of a standard 3-phase

There is a significant high frequency component. The shoot through frequency is 6 times the switching frequency because there are 6 shoot through periods per cycle. The available driver boards had circuitry that detects when top and bottom mosfets of one leg are on at the same time or within a certain limit. If these thresholds are broken the top mosfets are prevented from turning on to prevent shoot through. This protection was disabled to allow overlap necessary for shoot through. Fig 5 is a layout collage of the actual hardware used.

3.2 Voltage locking HSVB-PWM

To track the voltage a high speed computation of the RMS line to line voltage is required. Using the Clarke transform will allow the voltage locking algorithm to update using instantaneous values as apposed to calculating the RMS value from the root mean square of a number of values over a period of time. On the DSP the following computation are made to arrive at an RMS value from the instantaneous values of the 3 output LN voltages: First the real and imaginary parts of the three instantaneous values of the 3 phases are calculated using (13)

$$\begin{pmatrix} V_\alpha \\ V_\beta \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}\begin{pmatrix} v_a \\ v_b \\ v_c \end{pmatrix} \quad (13)$$

Then the RMS of the magnitude of the real and imaginary resultant vector is calculated using (14). The root of the sum of the squares gives the space vector.

$$= \frac{\sqrt{2}}{3}\sqrt{V_\alpha{}^2 + V_\beta{}^2} \quad (14)$$

algorithm uses this instantaneous value for fast voltage feedback in its control ntrol algorithm implemented on the DSP seamless transition between SV-PWM WM to track a given voltage.

of Voltage locking HSVB-PWM while varying the desired AC output.

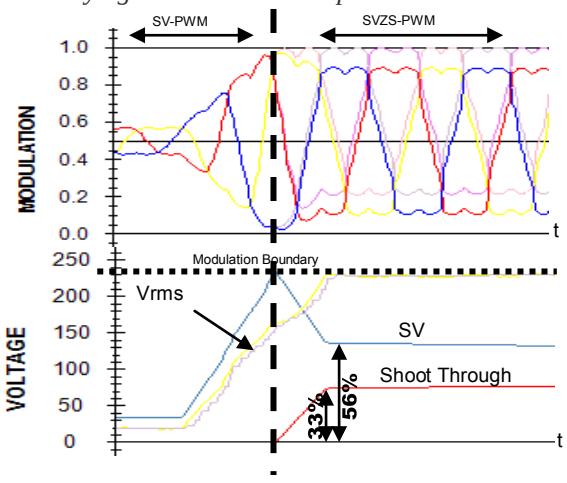


Fig. 6: Reference signals, Vrms LN, modulation and shoot through relationship.

Using the PC monitoring and control program the lock on voltage is set to 230Vrms LN from a 24V battery. In fig 6 the inverter increases its modulation (lower image in blue) to 100% but only achieves

158Vrms LN. It then decreases the modulation and fills the increasing zero vector periods with as much shoot through as possible without breaking the modulation boundary. Notice how only the 3 reference signals for the top switches can be seen in SV-PWM mode because the other 3 are identical and used to mirror the top switching signals. As soon as SVZS-PWM is introduced the other 3 reference signals appear with independent values so that 6 independent reference signals can be seen. At 56% Modulation with 33% Shoot through it achieves 230Vrms LN and stabilizes.

3.4 Results of Voltage locking HSVB-PWM while varying the desired AC output.

In this experiment the battery supply was replaced by a variable DC source. A variac was fed into a step down transformer and rectified. The DC supply varies during run time and the inverter should hold a steady 230Vrms LN. The DC bus was varied between 22V and 50V.

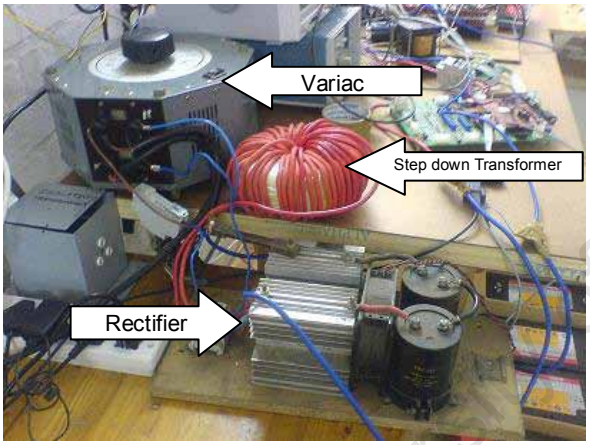


Fig. 7: Variable DC supply setup with variac, transformer and rectifier.

For a 2 kW system at 24V the variac can have up to 80Amps flowing through it. Even a large variac like the one in fig 7 can only handle up to 20A through its windings. By stepping down the voltage from the variac we can allow the variac to operate at a higher voltage and lower current. The DSP was programmed to limit the shoot through to 38% of a switching cycle to avoid over current conditions. In fig. 8 we see the results of a varying DC supply. The voltage varies between 20V and 50V DC as a fuel cell or wind turbine would do. The upper image shows the 6 independent reference waveforms used to generate the switching patterns. The lower image shows: In Blue: the modulation M. In Red: the Shoot through. In Yellow: Vrms calculated from the 3 instantaneous output sinusoid values using the Clarke Transform. In Purple: Vrms calculated from the root mean squared of V1rms LN instantaneous values. This is shown because it does not reflect the ripple associated with unbalanced phases when calculating Vrms using the Clarke Transform (in Yellow). The ripple is not from unstable voltage locking but rather from a 1-2V voltage imbalance between phases. In Black: The variable DC supply (magnified 500% for clarity). Notice that when there is zero shoot through (lower image in red) only 3 modulation signals can be seen as the top and bottom switches are mirroring each other and therefore use the same modulation signals. When the input is lower enough that traditional SVPWM cannot achieve 230V the modulation is decreased and the increasing zero vector periods are used for shoot through intervals to maintain 230V. Notice how during these conditions six modulation signals can be seen.

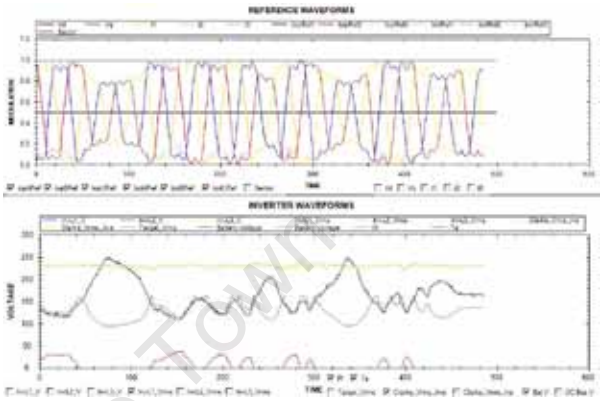


Fig. 8: Upper image - 6 independent modulation signals. Lower image - The modulation VS shoot though VS variable DC input VS Vrms LN.

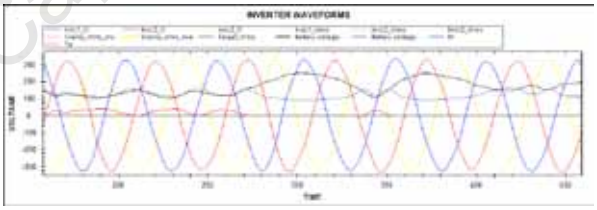


Fig. 9: The modulation VS shoot though VS variable DC input VS 3-phase output voltages.

In fig. 9 the same achievement is demonstrated except the 3-phase output voltage as apposed to the Clarke transform Vrms is shown. In black we have an unstable DC input with the necessary modulation and shoot through ratios to achieve a stable clean 230Vrms 3-phase sinusoidal output.

4. CONCLUSIONS

Statement of Achievements

- PC software: A completer control and monitoring program was designed and implemented on which most of the waveforms presented were captured. All set points of the prototype could be varied on the fly using the PC program control interface.
- Hardware: An operational 2 kW 3-phase Z-source inverter has been built running from multiple battery bank configurations or a variable DC input.
- Embedded software: C code was written that achieved:

- 5 kHz ADC sampling of the 3-phase output voltages, the DC supply voltage and the DC bridge voltage.
- PC <-> TMS2812 monitoring and control integration at 115200bps
- SV-PWM / SVZS-PWM control algorithms running up to 10 kHz.
- Voltage locking up to 240 Vrms from an on the fly variable supply as low as 20V DC.
- Seamless SV-PWM / SVST-PWM transition during voltage locking (HSVB-PWM).
- 6 independent PWM outputs with a 20 kHz switching cycle.
- Shoot through current protection during supply under voltage.

5. FUTURE DEVELOPMENT

Using the voltage locking capabilities a wind turbine can be connected through a rectifier as a volatile DC supply. The inverter could maintain a stable 230Vrms supply given the volatile DC input conditions. Additional control needed:

§ Maximum power point tracking (MPPT) to utilize the wind turbine at maximum power output given a specific wind speed.
§ Voltage locking must be extended to grid phase locking using the control algorithms briefly investigated in Appendix X.
§ The inverter must be rebuilt using specially designed overlapping bus plates as discussed in chapter 4 to minimize inductive spikes.
§ An investigation into the diode choice used on the Z-source input should be done to minimize diode losses during voltage clamping at shoot through frequencies up to 60 kHz.
§ A cost analysis between increasing the output transformer size and Z-source component size should be done to find the optimum balance of the 2 when trying to operate at 230Vrms from low voltage supplies.

REFERENCES

[1] Keyhani Jin-Woo Jung "Distributed Energy Systems", Journal of Iranian Association of Electrical and Electronics Engineers - Vol.1 - No.2 - Summer & Fall 2004.

[2] Miaosen Shen, Alan Joseph, Jin Wang, Fang Z. Peng1, and Donald J. Adams, "Comparison of Traditional Inverters and Z-Source Inverter for Fuel Cell Vehicles", Michigan State University Department of Electrical and Computer Engineering 2120 Engineering Building, East Lansing, MI 48824.

[3] Fang Zheng, Senior Member, IEEE " Z-Source Inverter", Paper IPCSD 02-078 presented at the 2002 Industry Applications Society Meeting, Pittsburgh, PA, October 13-18.

[4] Keliang Zhou and Danwei Wang, Member, IEEE, "Relationship between Space-Vector Modulation and Three-Phase Carrier-Based PWM: A Comprehensive Analysis", Manuscript received November 3, 1999; revised July 6, 2001. Abstract published on the Internet December 5, 2001. Publisher Item Identifier S 0278-0046(02)00933-4.

[5] http://www.ewh.ieee.org/soc/es/Nov1998/08/PWMINV.HTM

[6] S Ekram, Member, Dr B Sarkar, Fellow, "Effects of Harmonics on PWM Inverter fed Induction Machines", S Ekram is with EED, SIG College of Engineering, Mumbai and Dr B Sarkar is with SGS Institute of Technology and Science, Indore 452 003.

[7] B. Justus Rabi and R. Arumugam, "Harmonics Study and Comparison of Z-source Inverter with Traditional Inverters", American Journal of Applied Sciences 2 (10): 1418-1426, 2005 ISSN 1546-9239 © 2005 Science Publications.

[8] Mohan, N., W.P. Robbin, and T. Undealand, 1995, "Power Electronics, Converters, Applications and Design", 2nd Edn. New York,Wiley.

[9] J. Holtz, "Pulsewidth modulation for electronic power conversion,"*Proc. IEEE*, vol. 82, pp. 1194–1214, Aug. 1994.

[10] H. W. v. d. Brocker, H. C. Skudenly, and G. Stanke, "Analysis and realization of a pulse width modulator based on the voltage space vectors,"in *Conf. Rec. IEEE-IAS Annu. Meeting*, Denver, CO, 1986, pp. 244–251.

[11] D. W. Chung, J. S. Kim, and S. K. Sul, "Unified voltage modulation technique for real-time three-phase power conversion," *IEEE Trans. Ind.Applicat.*, vol. 34, pp. 374–380, Mar./Apr. 1998.

[12] Prof. Ali Keyhani, "Pulse-Width Modulation (PWM) Techniques", Department of Electrical and Computer Engineering, The Ohio State University.

[13] Jin-Woo Jung, Min Dai, Ali Keyhani, "Modeling and Control of a Fuel Cell BasedZ-Source Converter", Department of Electrical and Computer Engineering,The Ohio State University Columbus, OH 43220, USA.

APPENDIX M

Control Hardware Decisions

M1: DSP versus FPGA:

The DSP is a specialised microprocessor - typically programmed in C, perhaps with assembly code for performance. It is well suited to extremely complex math-intensive tasks, with conditional processing. It is limited in performance by the clock rate, and the number of useful operations it can do per clock. As an example, a TMS320C6201 has two multipliers and a 200MHz clock – so can achieve 400M multiplies per second. In contrast, an FPGA is an uncommitted "sea of gates". The device is programmed by connecting the gates together to form multipliers, registers, adders and so forth. Their performance is limited by the number of gates they have and the clock rate. Recent FPGAs have included Multipliers especially for performing DSP tasks more efficiently. When sample rates grow above a few Mhz, a DSP has to work very hard to transfer the data without any loss. This is because the processor must use shared resources like memory busses, or even the processor core which can be prevented from taking interrupts for some time. An FPGA on the other hand dedicates logic for receiving the data, so can maintain high rates of I/O. A DSP is optimised for use of external memory, so a large data set can be used in the processing. FPGAs have a limited amount of internal storage so need to operate on smaller data sets. However FPGA modules with external memory can be used to eliminate this restriction. A DSP is designed to offer simple re-use of the processing units, for example a multiplier used for calculating an FIR can be re-used by another routine that calculates FFTs. This is much more difficult to achieve in an FPGA, but in general there will be more multipliers available in the FPGA. If a major context switch is required, the DSP can implement this by branching to a new part of the program. In contrast, an FPGA needs to build dedicated resources for each configuration. If the configurations are small, then several can exist in the FPGA at the same time. Larger configurations mean the FPGA needs to be

reconfigured – a process which can take some time. The DSP can take a standard C program and run it. This C code can have a high level of branching and decision making – for example, the protocol stacks of communications systems. This is difficult to implement within an FPGA. Most signal processing systems start life as a block diagram of some sort. Actually translating the block diagram to the FPGA may well be simpler than converting it to C code for the DSP [24].

M2: Making a choice:

There are a number of elements to the design of most signal processing systems, not least the expertise and background of the engineers working on the project. These all have an impact on the best choice of implementation.

A rough guideline of questions presented by [24] will offer direction:

1) What is the sampling rate of this part of the system? If it is more than a few MHz, FPGA is the natural choice.

ANS: Sampling rate is below 20kHz.

2) Is your system already coded in C? If so, a DSP may implement it directly. It may not be the highest performance solution, but it will be quick to develop.

ANS: Algorithms to be implemented lend themselves easily to straight C coding as the IO and control algorithms are closely woven.

3) What is the data rate of the system? If it is more than perhaps 20-30Mbyte/second, then FPGA will handle it better.

ANS: All data interface options are on a 115200bps uart easily implemented with today's DSP range.

4) How many conditional operations are there? If there are none, FPGA is perfect. If there are many, a software implementation may be better.

ANS: There are many conditional operations as the final algorithm is a slice between existing SV PWM and Shoot though algorithms depending on a number of conditions including user options and ADC readings.

5) Does your system use floating point? If so, this is a factor in favour of the programmable DSP. None of the FPGA cores support floating point today, although you can construct your own.
ANS: Yes, many floating point operations occur when calculating modulation signals from a range of trigonometric formulae.
6) Are libraries available for what you want to do? Both DSP & FPGA offer libraries for basic building blocks like FIRs or FFTs. However, more complex components may not be available, and this could sway your decision to one approach or the other.
ANS: Although I will not be using the direct SV PWM libraries for the DSP they do exist and have a proven track record. I will write my own libraries because the necessary algorithms for HSVB PWM are out of the scope of the basic libraries.

In reality, most systems are made up of many blocks. Some of those blocks are best implemented in FPGA, others in DSP. Lower sampling rates and increased complexity suit the DSP approach; higher sampling rates, especially combined with rigid, repetitive tasks, suit the FPGA [24]. From these questions and answers it is clear a DSP is the favorable option.

M3: Choice of DSP processor:

This prototype at MLT Drives was built using MLT Drives designed and manufactured control instruments with the centre of the control being the TMS2812 DSP from Texas Instruments. The 2812 is the latest and a long line of DSP processors used for power electronic systems where fast processing power coupled with many measurement and PWM output IO ports are available. The 2812 is the new preferred choice for the latest models of high end power electronic devices at MLT Drives. It is for this reason I investigated the TMS2812 for my prototype needs.

M4: The TMS2812 solution to 6 independent PWM channels.

The latest DSP range of 2007 all offer very similar features. The one that was most important to me yet very rare in even today's DSP's was the ability to control 6 PWM channels separately. The TMS2812 is the first in its family to have 12 PWM channels of which 6 can operate independently. HSVB PWM is unlike standard SVPWM where the top switches are dependant on the bottom switches allowing PWM channels to operate in pairs of top and bottom. The ability to generate 6 on the fly independent modulation signals which could be fed to 6 independent comparators to drive 6 independent PWM ports was the concluding factor in my DSP choice.

Figure M1 shows event manager EVA on the TMS2812. There are two such event managers. The other is EVB and is identical in principal although controlled by an entirely separate set of registers and interrupts. Note how EVA like EVB has 6 PWM channels which can operate from control from a number of control registers and control blocks. For example a 3 phase SV PWM inverter can be run using only EVA and the SV block with dead time features. There are only 3 truly independent PWM channels associated with EVA. Although 6 different PWM outputs can be produced they are actually working in pairs with added delays for dead time. By using both EVA, EVB and disregarding the 6 PWM channels that only mirror the others with possible dead time additions we can have 6 independent PWM channels with the computation blocks, timers and interrupts of 2 event managers.

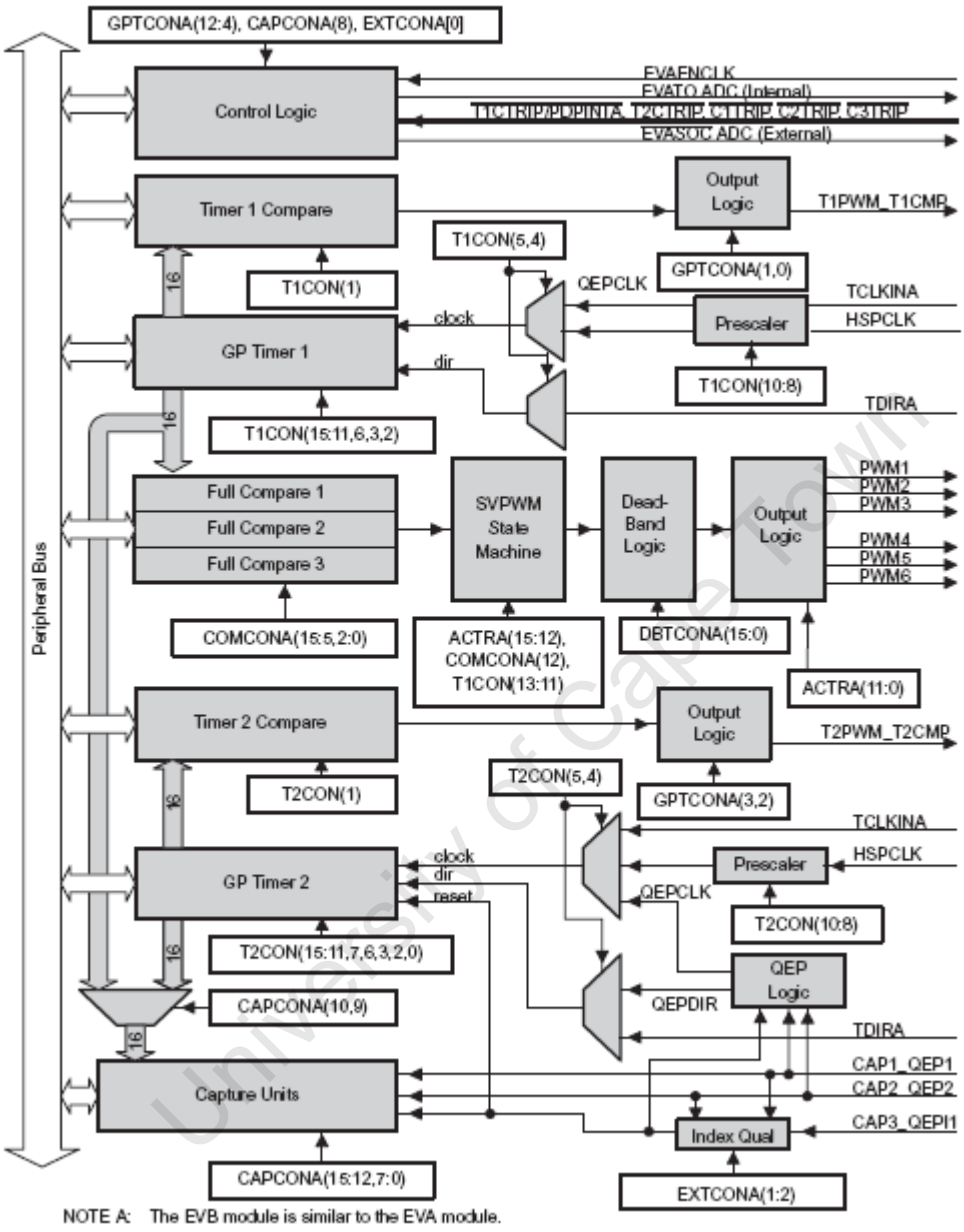


Figure M1: Event Manager A on the TMS2812

APPENDIX N

Comparison of Simulation and prototype results for HSVB PWM

N1: Comparison of Reference Waveforms generated in Simplorer 7 simulation and real time event driven C code on the TMS2812.

The control signals for the 6 comparators produced from the VHDL algorithm are detailed in fig E7 in Appendix E. Fig E8 shows all control signals on one plot.

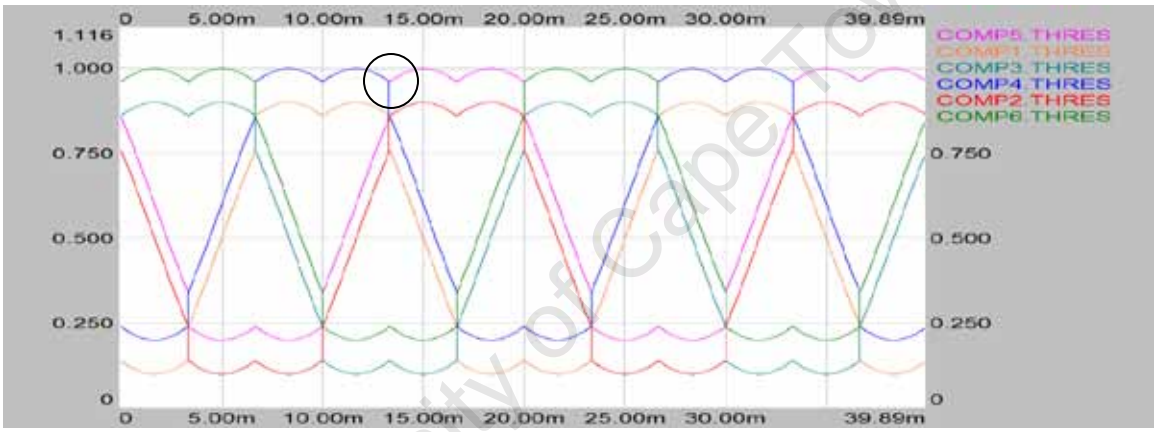


Fig. E8: The 6 generated control signals seen on each comparator input.

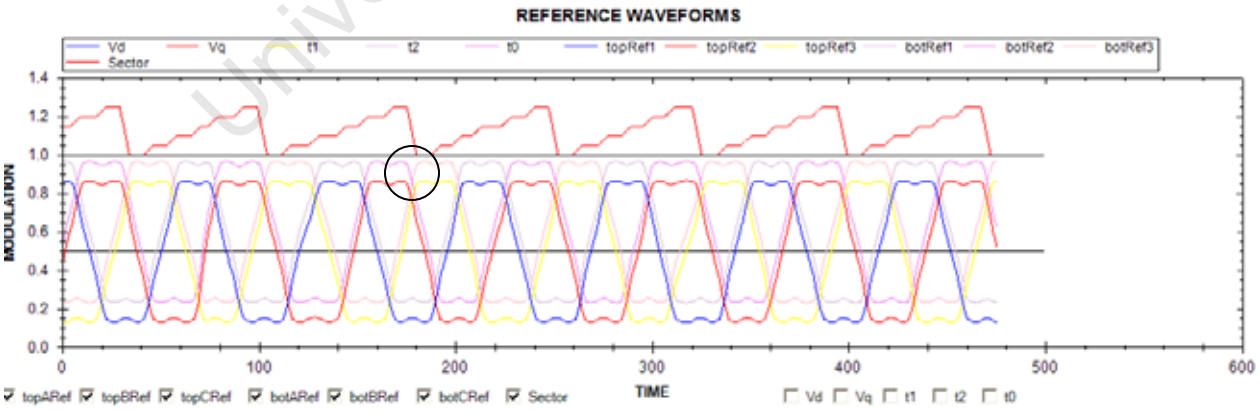


Fig. 4.8: PC monitoring of 6 independent reference signals for modified SVPWM.

Fig. 4.8 shows the reference signals generated by the DSP code when executed and displayed by the PC based software. These reference signals are generated with Ta=0.1 and Ma = 0.9.

Comparing fig E8 and fig 4.8 we can see the TMS2812 is generating reference signals very similar to those generated via Simplorer. The only difference is that the waveforms from the TMS2812 are distorted because of a rolling average function operating on the incoming data to smooth out noise. This is not the actual signal used by the TMS2812 for PWM generation. The actual waveform would be identical to that from the Simplorer simulation results in fig E8.

N2: Comparison of PWM generated in Simplorer 7 simulation and real time event driven C code on the TMS2812.

In the prototype software process the simulated work detailed in Appendix E was followed in deriving c code algorithms, the prototype reference wave forms are compared against an up down counter at a desired switching frequency. The result is 6 independent PWM for each of the 6 mosfets of the 3-phase bridge. An instant of the PWM for sector 1 phase A generated from the comparison of figure 4.8 reference waveforms with a 10 kHz up down counter (triangular waveform) are shown in figure N1. The top and bottom PWM signals are the top and bottom switches respectively. The middle PWM waveform is the addition of the two. The result is a '1' on overlap (shoot through) or '-1' when neither switch is on (dead time). The shoot though is divided equally over 3 phases so only a 3rd of the shoot though is seen on phase A.

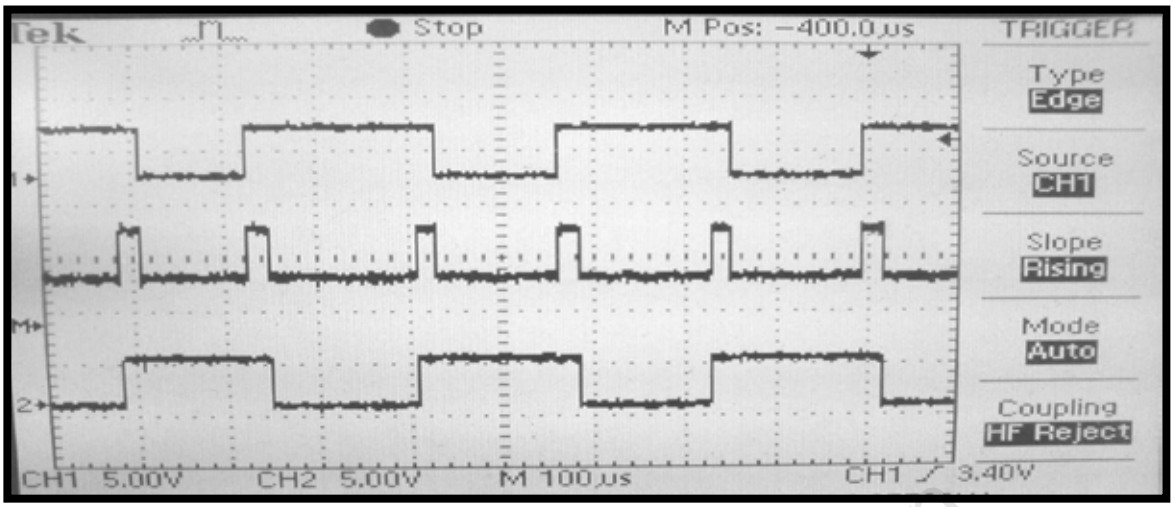


Fig. N1: Instant of sector 1 phase A PWM for top switches for Ta= 30%

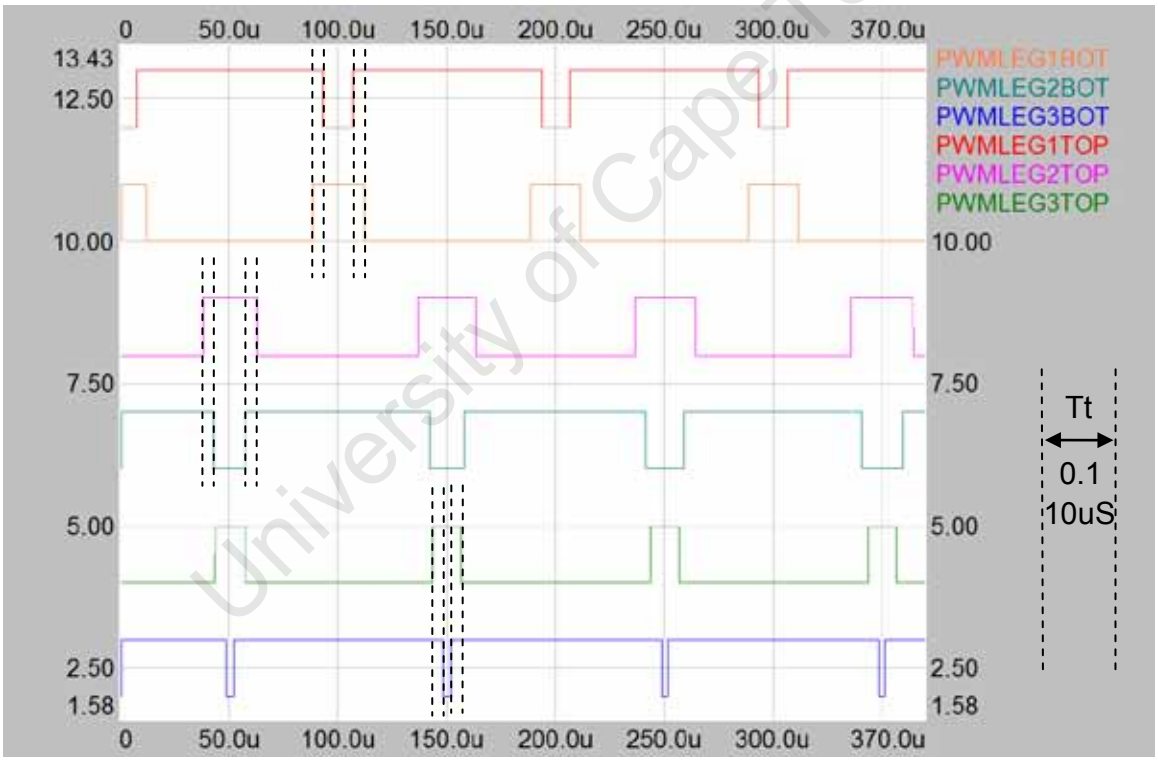


Fig. E10: The 6 PWM signals produced for sector 1 with Ta = 30%.

Fig N1 shows the DC bus at 30% shoot through. In the image the DC bus is pulled low by the short for 30% of a switching cycle. Comparing fig N1 and fig E10 for a value of 30% shoot through we can see that in both cases a 10% overlap was achieved between top and bottom switches. As discussed in chapter 2 theory the shoot through in one half cycle is a 3rd of the total shoot through giving us a 10%

us two 10% shoot though overlaps per leg. This gives us the 6 shoot through periods between three legs over one half cycle. From fig N1 and fig E10 we can see an expected two 10% shoot though overlap periods were achieved in both simulation and TMS2812 PWM generation.

N3: Comparison of Z-source output / 3 Phase Bridge input in Simplorer 7 simulation and real time prototype results.

Comparing fig E13 and Fig N2 we can see the ripple after each rising edge. This is a result of inductance between components, namely the larger electrolytic DC caps and bus bars. To reduce this specialised bus plates separated by a few mm could me manufactured into which the Z-source caps are fixed directly onto one plate and through the top plate to the lower plate.

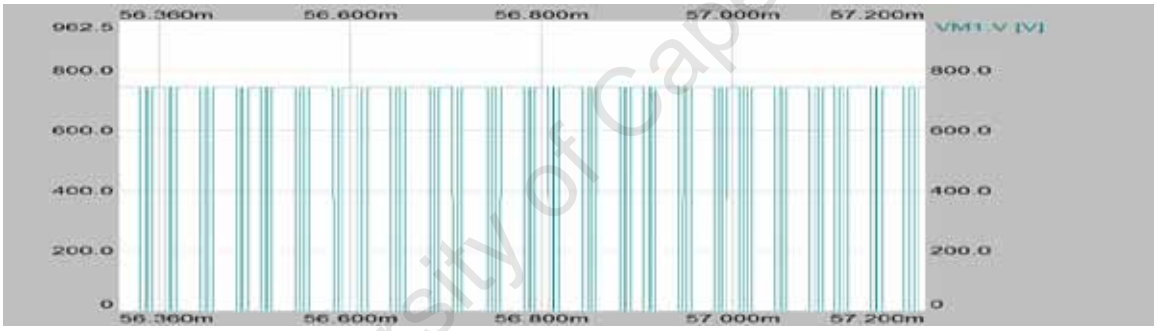
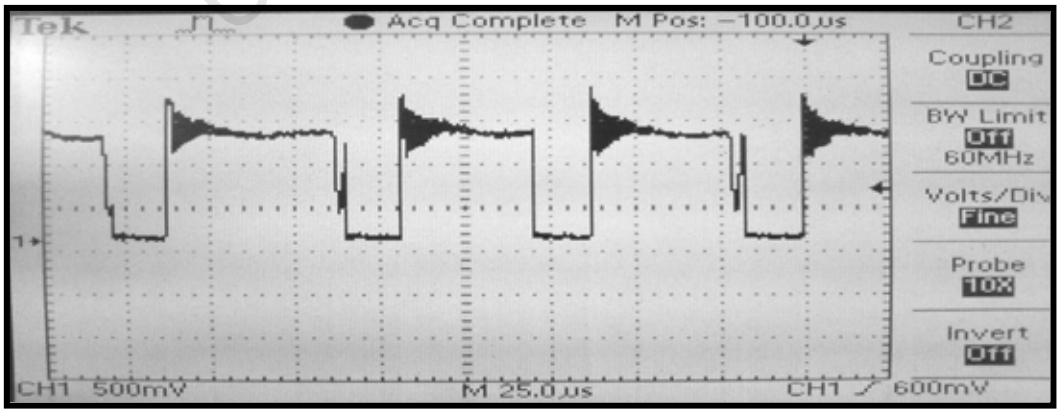


Fig. E13: High time resolution slim section of $\hat{v}_i$, steady state voltage waveform seen by 3 Phase Bridge. Peak voltage = 750V.



N4: Comparison of prototype and Simplorer simulation 3 phase sinusoidal output over load.

The filtered 3 phase output of the 3 phase bridge in Simplorer simulation is shown in fig E11. The final prototype output over the 3 phase load is shown in fig 4.11. Both result in a clean 3 phase sinusoidal output. Table N1 compares results between theory, Simplorer simulation and prototype runtime results.

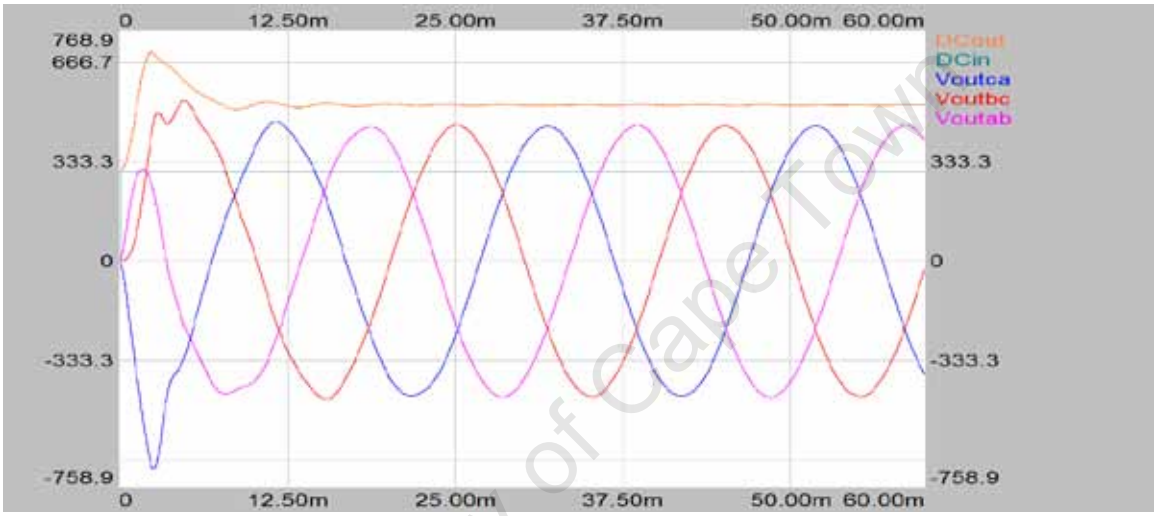


Fig. E11: LC filter output of 3 sinusoids 120˚ out of phase (Ta=0.3)

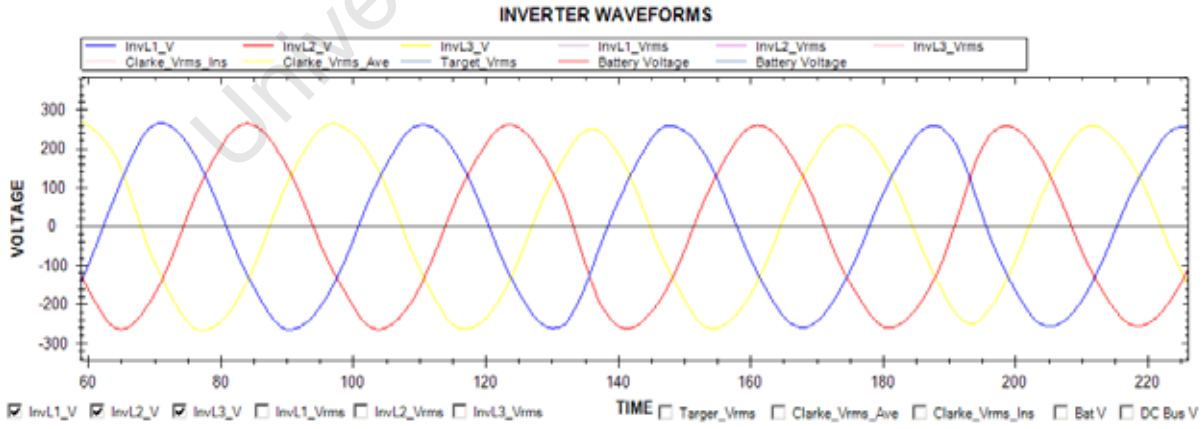


Fig 4.11: 230Vrms 3-phase output from 24V battery through 1:10 transformer.

	THEORETICAL	SIMULATION	Prototype
Ta/Tz (% shoot through)	0.33	0.33	0.33
Tb/Tz = M	0.56	0.56	0.56
Vin	24	24	24
Average DC boost	1.970588235	2	2
C2 average Voltage	47.29411765	47	45
Peak DC link voltage	70.58823529	70	64
KTR (transformer step up)	10	10	10
V LNrms	242.0672216	242	230

Table N1: Setting and results for attaining 230Vrms LN from 24V DC 1

From the table it is clear that the Simplorer simulations that used ideal components gave the expected results with the only discrepancy being the meaningful accuracy of measurement. The prototype results differ and were found to deviate further from the expected simulated or theoretical results as the shoot through was increased. The reason for these discrepancies are as follows:

- The DC bus diode array: The very high switching frequency seen by the diode is 6 times that of the already high 20kHz switching cycle (chapter 2 theory). The diodes function is to block current flowing back towards the DC supply as the voltage is boosted higher on the Z-source bus. Some investigation into a high power diode that can clamp the reverse current at 60kHz needs attention. The prototype diode array allowed the peak DC link voltage to be lag as the shoot through was increased. In table N1 we have a 6 volt discrepancy as a result of failure to block reverse current.
- Magnetizing losses for all inductive components: To add to the current losses on the DC link all large inductive components consume energy that would otherwise be stores in the DC caps which lowers the average voltage on the DC link.
- 3-phase bridge switching losses: As with a standard 3 phase inverter topology the 6 mosfet arrays of the 3 phase bridge have noticeable switching losses resulting in a less than theoretically expected 3 phase rms voltage.

REFERENCES

[1]A. Keyhani Jin-Woo Jung “Distributed Energy Systems”, Journal of Iranian Association of Electrical and Electronics Engineers - Vol.1 - No.2 - Summer & Fall 2004

[2] Miaosen Shen, Alan Joseph, Jin Wang, Fang Z. Peng1, and Donald J. Adams, “Comparison of Traditional Inverters and Z-Source Inverter for Fuel Cell Vehicles”, Michigan State University Department of Electrical and Computer Engineering 2120 Engineering Building, East Lansing, MI 48824

[3] Fang Zheng, Senior Member, IEEE “ Z-Source Inverter”, Paper IPCSD 02-078 presented at the 2002 Industry Applications Society Meeting, Pittsburgh, PA, October 13-18.

[4] Keliang Zhou and Danwei Wang, Member, IEEE, “Relationship between Space-Vector Modulation and Three-Phase Carrier-Based PWM: A Comprehensive Analysis”, Manuscript received November 3, 1999; revised July 6, 2001. Abstract published on the Internet December 5, 2001. Publisher Item Identifier S 0278-0046(02)00933-4.

[5] http://www.ewh.ieee.org/soc/es/Nov1998/08/PWMINV.HTM

[6] S Ekram, Member, Dr B Sarkar, Fellow, “Effects of Harmonics on PWM Inverter fed Induction Machines”, S Ekram is with EED, SIG College of Engineering, Mumbai and Dr B Sarkar is with SGS Institute of Technology and Science, Indore 452 003.

[7] B. Justus Rabi and R. Arumugam, “Harmonics Study and Comparison of Z-source Inverter with Traditional Inverters”, American Journal of Applied Sciences 2 (10): 1418-1426, 2005 ISSN 1546-9239 © 2005 Science Publications.

[8] Mohan, N., W.P. Robbin, and T. Undealand, 1995, “Power Electronics, Converters, Applications and Design”, 2nd Edn. New York,Wiley

[9] J. Holtz, “Pulsewidth modulation for electronic power conversion,” *Proc. IEEE*, vol. 82, pp. 1194–1214, Aug. 1994.

[10] H. W. v. d. Brocker, H. C. Skudenly, and G. Stanke, “Analysis and realization of a pulse width modulator based on the voltage space vectors,” in *Conf. Rec. IEEE-IAS Annu. Meeting*, Denver, CO, 1986, pp. 244–251.

[11] D. W. Chung, J. S. Kim, and S. K. Sul, “Unified voltage modulation technique for real-time three-phase power conversion,” *IEEE Trans. Ind. Applicat.*, vol. 34, pp. 374–380, Mar./Apr. 1998.

[12] Prof. Ali Keyhani, “Pulse-Width Modulation (PWM) Techniques”, Department of Electrical and Computer Engineering, The Ohio State University.

[13] Jin-Woo Jung, Min Dai, Ali Keyhani, “Modeling and Control of a Fuel Cell Based Z-Source Converter”, Department of Electrical and Computer Engineering The Ohio State University Columbus, OH 43220, USA

[14] Poh Chiang Loh, Member,, IEEE, D. Mahinda Vilathgamuwa, Senior Member, IEEE, Yue Sen Lai, Geok Tin Chua and Yunwei Li, Student Member, IEEE, "Pulse-Width Modulation of Z-source Inverter", IEEE Transactioons on power electronics, vol 20, no. 6 November 2005.

[15] www.fueleconomy.gov, fuel image reference. Accessed: June 17th 2007.

[16] Kenneth S. Deffeyes, "Hubbert's Peak: The Impending World Oil Shortage", By Kenneth S. Deffeyes, Published 2001 Princeton University Press, Petroleum industry and trade, ISBN 0691116253.

[17] Karl Kordesch ª *, Josef Gsellmann ª, Martin Cifrain ª, Susanne Voss ª, Victor Hacker ª Robert R. Aronson b, Christoph Fabjan c, Thomas Hejze c, Josef Daniel-Ivad "Intermittent use of a low-cost alkaline fuel cell-hybrid system for electric vehicles",ª *Technical University, Graz, A-8010, Austria*, *b Electric Auto (EAC), Fort, Lauderdale, Fl, USA*, c *Technical University, Vienna, A-1060, Austria, d Battery Technologies, Richmond Hills, ON, Canada LAB 1C3,* Received 10 November 1998; accepted 2 December 1998.

[18] N Brandon and D Hart, "An Introduction to Fuel Cell Technology and Economics", Centre for Energy Policy and Technology (ICCEPT), Imperial College of Science, Technology and Medicine, 48 Prince's Gardens, London SW7 2PE, July 1999.

[19] Nathan Myers, James DeHaan, "Fuel Cells: Will fuel cells be replacing batteries at your facility", Bureau of Reclamation, Denver, CO 80225.

[20] V. Kaura, V. Blasko, "Operations of Phase Locked Loop system under distorted utility conditions" in IEEE Trans. Ind. Applicat., vol. 33, no. 1, Jan/Feb 1997.

[21] Antonio M. M. Cornelio, "Viability study to implement peak load shaving at UCT medical school", Appendix C, master's dissertation, University of Cape Town 2007.

[22] J. Stewart, "Calculus concepts and contexts" McMaster University, ISBN 0534376118.

[23] Antonio M. M. Cornelio, "Viability study to implement peak load shaving at UCT medical school", Appendix D, masters dissertation, University of Cape Town 2007.

[24] http://www.hunteng.co.uk/info/fpga-or-dsp.htm. Accessed January 31st 2008.