Space Vector Modulation of a 4-Leg Matrix Converter

Nicholas J. Mason, BEng. MSc.

Thesis submitted to
The University of Nottingham
for the degree of Doctor of Philosophy

November 2011

I'd like to dedicate this work to my father, David Mason, who passed away before he was able to see its completion, and whose memory is, and will always be, an inspiration to me.

Abstract

THIS thesis covers the investigation into the use of Space Vector Modulation for the control of a 4-leg matrix converter, which is capable of providing a 3-phase plus neutral supply from a standard balanced 3-phase source. Traditional 3x3 matrix converters have limited use in this application as they are only capable of supplying a balanced three-phase load. It would be desirable to be able to power unbalanced and non-linear loads, requiring that the converter provides a neutral connection. As with voltage source inverters, this goal can be achieved by extending the number of output legs in the matrix converter to four. In this thesis, a new Space Vector Modulation technique is proposed for this 4-leg, or 3x4, matrix converter. This technique is an extension of the method currently in use on 3x3 matrix converters, and so it allows the de-coupled control of both the input and output. The thesis then goes onto cover the build of a demonstration converter, looking at the different aspects which make up a converter, to finally go on to prove the theory, and a set of results are presented to validate this.

Acknowledgements

I would like to express my utmost thanks for my supervisors, Prof. P. Wheeler and Prof. J. Clare, for both of their guidance and support, and most importantly, for their utmost patience in me. I am sure there were times when they, like me, gave up hope that this work would ever be completed, but their willingness to help always meant that I could get going going again.

I'd do also like to thank all of the staff within the PEMC group at the University of Nottingham who have helped me over the now many years that this work has been in progress. Alongside this I would like to specifically thank Dr Lee Empringham and Dr Alan Watson for there help at times with the build of the converter, who, along with the workshop technicians were always happy to help when I needed it and whose help is much appreciated.

I'd then like to thank my family and friends, they have been such a immense support to me, more so than I think many of them know. They have often picked me up and brushed me off when times were bad, and so to them I owe a debt of gratitude. In particular though I would like to say so many thanks to my wonderful partner Elizabeth Heslop, without her help and support over the months and years that this has taken, through the sad times and the happy, I am sure it would never have been finished, and so for that I am eternally thankful. I'd also like to say thank you to my mother, who's always been a support to me, and has really helped so many times.

And finally like to thank my sponsor, the Engineering and Physical Science Research Council(EPSRC), without whose support I would not have been able to embark on this course of study, and I am sure they, as much as anyone will be happy for me to complete.

Contents

List of Figures

List of Tables

Acronyms

DSP digital signal processor.

FCC force-commutated cycloconverter.

FET field effect transistor.

FFT fast Fourier transform.

FIFO First-In First-Out.

FPGA field programmable gate array.

GPU ground power supply unit.

GTO gate turn-off thyristor.

GUI graphical user interface.

HPI Host Port Interface.

IGBT insulated gate bipolar transistor.

MEX Matlab executable.

NCC naturally commutated cycloconverter.

PCB printed circuit board.

PEMC Power Electronic and Machine Control.

PLL phase-locked loop.

PWM pulse-width modulation.

SCR silicon controlled rectifier.

SVM space vector modulation.

UPS uninterruptible power supply.

USB Universal Serial Bus.

CHAPTER 1

Introduction

Over recent years there has been an increased interest in static power converters, and this has led to a rapid improvement in their technology. The use of such converters in various AC power supply applications has also grown, and still continues to do so, with static power converters now being a common part in many different systems, from uninterruptible power supplies (UPSs), power converters in submarines through to ground power supply units (GPUs) for aircraft. One other typical application is for mobile power generation, and such a system is shown in Figure 1.1. This is an outline of a complete standalone system for providing a three-phase power supply wherever it is needed using a variable speed generator. The particular setup shown, uses the variable speed generator connected to a power converter, which allows the engine driving the generator to operate over a wide range of engines speeds and loads, with the power converter then providing a fixed frequency fixed voltage supply[1].

The power converter used in the scheme shown in Figure 1.1 could be a standard 3-phase rectifier/inverter setup, although this is a good setup where a matrix converter could be used. There is a problem with both of these types of converter, and it is that, while capable of generating a balanced three-phase supply, they are not able to operate with unbalanced, non-linear, or single-phase loads. To enable the system to generate an AC supply capable of supplying single-phase or unbalanced three-phase loads the converter needs to be able to deal with the unbalanced neutral,or zero-sequence, current. One method for doing this with an inverter is to increase the number of converter output legs to include a neutral connection[2]. Extending this in the case of the matrix converter would result in a 3x4, or four-leg, matrix converter as shown in Figure 1.2. This shows a simplified version of Figure 1.1 with a four-leg matrix converter attached to the generator.

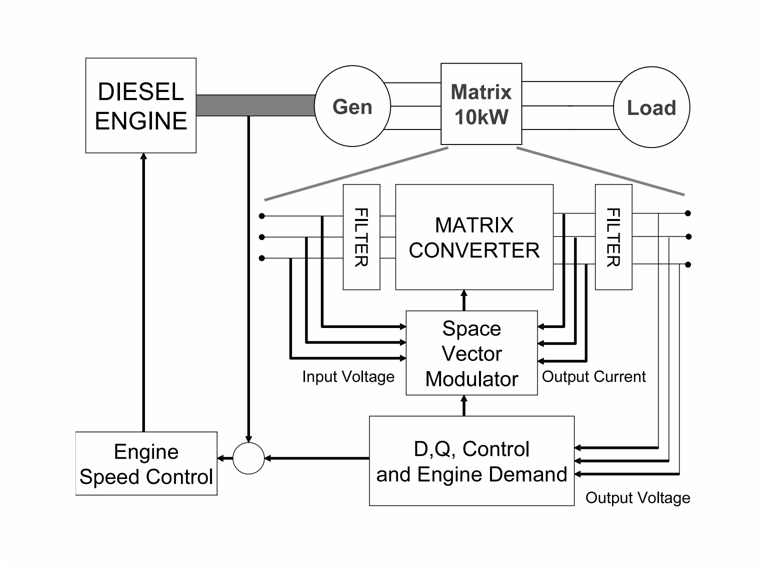


Figure 1.1: An Example of a Standalone Power Supply System using a variable speed generator

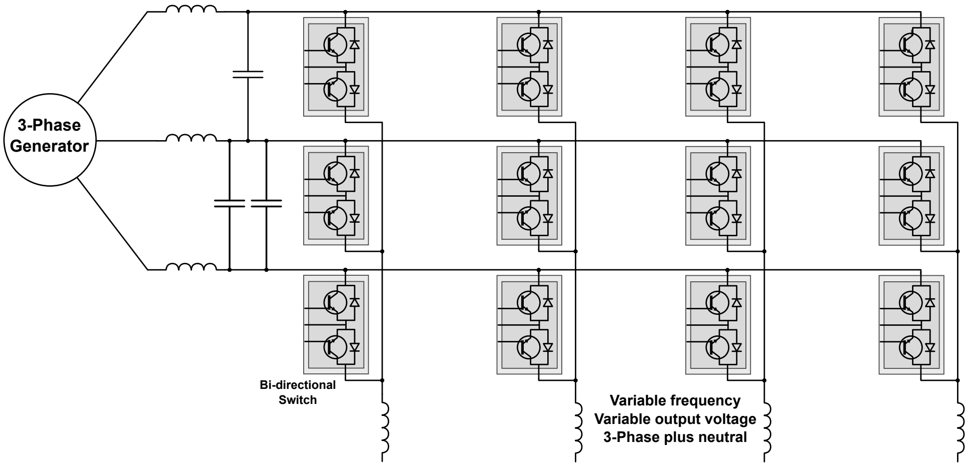


Figure 1.2: A 4-Leg Matrix Converter as part of a Standalone Power Generation System

With the introduction of the fourth leg, this adds another output variable that needs to be controlled, and so the modulation techniques why apply to the 3x3 matrix converter[3–6] are no longer valid, and so a new modulation strategy needs to be formulated for this new arrangement. After the work by Huber and Borojevic [7–10] and later Casadei[3, 4] on adapting space vector modulation (SVM) for use with the 3x3 matrix converter, and then Ryan and De Doncker [11, 12] along with Zhang at al[2, 13, 14] for the work on 3-dimensional SVM for the 4-leg inverter, these both proved as useful pointers towards how a successful technique could be derived for the 4-leg Matrix Converter. Therefore, this work sets out to explore the design of a Space Vector Modulation technique that is capable of controlling a 4-leg matrix converter, and then goes on to detail the building and testing of such a converter.

The basis of the space vector modulation strategy for the 3x3 matrix converter[3, 7] was derived from the techniques for both the controlled rectifier, and the inverter. This new technique will mirror this, and so be derived from the modulation method used for a 3-phase rectifier and a four-leg inverter[12, 14]. This will use a space vector representation in a three-dimensional $\alpha\beta\gamma$ space to describe the output voltage and switching states[11, 12, 14]. This modulation technique will enable the inverter, and thus the matrix converter, to provide any set of output line-to-neutral voltages, balanced or unbalanced, that may be required. This three-dimensional approach will then be combined with the standard two-dimensional SVM method for the input side of the matrix converter, in a similar way to that performed for the 3x3 matrix converter[3, 4]. The resulting SVM method will be a way of controlling the modulation of the four-leg matrix converter so as to produce any set of line-to-neutral voltages whilst also independently controlling the phase of the input current waveforms.

1.1 The Matrix Converter

Matrix converters are a class of converters which perform direct AC-AC power conversion and themselves are part of the larger class of converter which are called cycloconverters. The process is called direct conversion to differentiate them from the more well known rectifier and inverter pairing as there is no intermediate DC link, and no energy storage devices within the converter. Cycloconverters first appeared in the 1920s and 1930s using mercury arc rectifiers[15–17] and are still in use today[17, 18], although now using devices such as the silicon controlled rectifier (SCR) and gate turn-off thyristor (GTO), in very high power industrial applications such a rolling mills.[19]. An example of 3-phase cycloconverter is shown in Figure 1.3.

The cycloconverters themselves fall into two main groups:-

- those which use the supply line voltage to perform the current commutation between the devices, which are known as naturally commutated cycloconverters (NCCs)
- those in which the current commutation is forced between devices,

and are known as force-commutated cycloconverters (FCCs)

The naturally commutated cycloconverters use devices such as thyristors and SCRs to be able to control the point at which the device starts conducting, thus, by choosing the correct phases from the input voltage, an output voltage can be synthesised. However, due to the nature of the devices they require the current in the device to reach zero before the device can be turned off, so while it is possible to control the turn on of each device, the turn off happens as the current naturally falls within the device due to the changes in input voltage. Being able to control only the device turn on means that the range output frequency is limited to being below the input frequency[15–17].

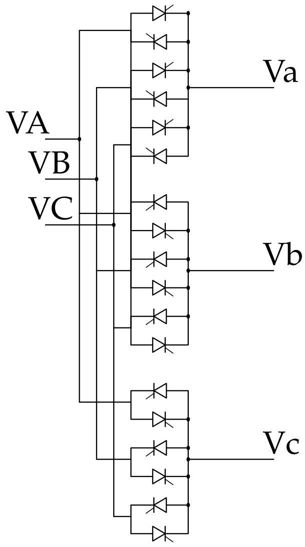


Figure 1.3: A 3-phase Cycloconverter

With the advance in the capability of switching devices, most notably insulated gate bipolar transistors (IGBTs), and due to its relative complexity, which can be seen in Figure 1.3, this type of cycloconverter now tends to only be used in high power applications, above the 1MW level, where the low frequency output is able to be used effectively, for example, like driving induction motors in large industrial processes, or for thrusters for large navel vessels[17, 18].

The force-commutated cycloconverters on the other hand, use devices which are capable of being switched off while current is still flowing through them, and so can be turned off at any point in the conduction cycle, instead of having to wait for the current through the device to drop to zero. This allows the force-commutated cycloconverter to overcome the largest drawback of the naturally commutated one, and be able to produce outputs at any frequency required, regardless of the input frequency[5].

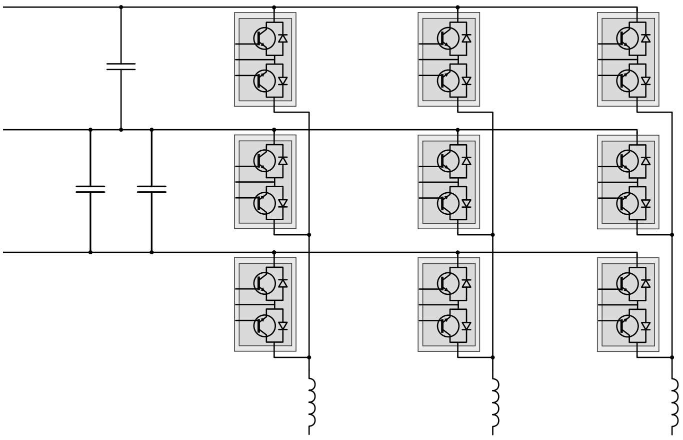


Figure 1.4: A 3-Leg Matrix Converter

A 3x3 matrix converter is shown in Figure 1.4, and due to using switches that have the ability to be turned of, this allows the circuit to be simpler. Although the concept of matrix converters had been around for a number of years it was not until Alesina and Venturini published a paper at the beginning of the 1980s[5] that there become a way of controlling them which allowed both the input current and the output voltage to be

The initial Alesina and Venturini method had a disadvantage though, that the output voltage was restricted to be a maximum of only 0.5 of the input. However, towards the end of the 1980s they published a number of papers which first showed that the theoretically maximum output voltage was $\frac{\sqrt{3}}{2}$[20] and then went on to show how this was possible with their optimum method[6]. Lipo produced a similar result in 1989, while also succeeding in building a working converter[21] to demonstrate this.

At around the same time, a number of papers by Huber and Borojevic were published, which presented a new space vector modulation technique for the matrix converter [7–10]. This like the Alesina\Venturini Optimum method, was able to produce the theoretical maximum output voltage. Space Vector Modulation is a well known technique for controlling inverters, and it uses a technique which transforms the input and output voltages and currents into a 2-dimensional plane where they can then be easily manipulated. This is as opposed to the method Alesina and Venturini used which is based around the use and theory of sine waves.

Later on the space vector modulation technique was refined further by Casadei[3, 4], and while more recent matrix converter configurations and modulation strategies have been proposed[22, 23], it is the technique proposed by Casadei which this project seeks to extend onto the 4-Leg Matrix Converter.

1.2 The 4-leg Inverter

The four-leg inverter first appeared in the early 1990's[24] when a solution was sought to allow the standard 3-leg inverter to drive unbalanced and non-linear loads[24–27]. Due to the nature of these loads they cause an imbalance in the current drawn from each phase of the supply, which then requires an extra neutral connection to deal with the zero sequence current which then results. While the use of a split capacitors to provide this neutral point was investigated[26, 28], this arrangement only works well if the imbalance, or non-linearity, is small. Instead of the split capacitors, the 4-legged inverter uses an extra inverter leg to provide this connection, and Figure 1.5 shows an example.

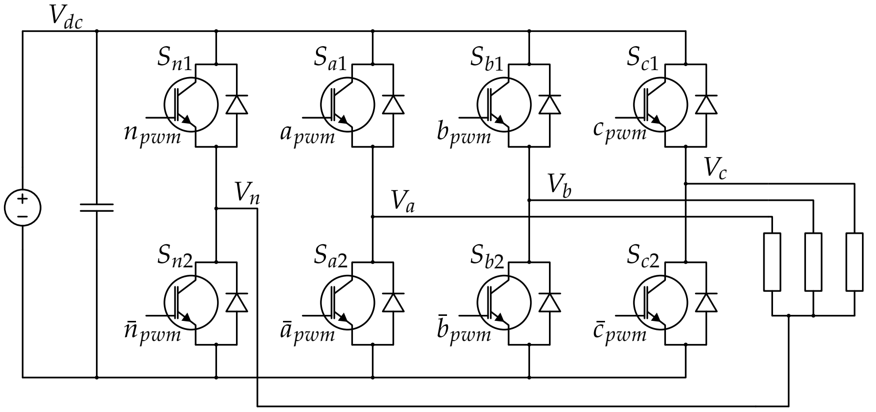


Figure 1.5: A 4-Leg Inverter

The initial attempts at controlling the 4-leg inverter were based around controlling the fourth leg separately, and using it to reduce any deviation in the neutral voltage[24, 26]. It was then in the late 1990's that Space Vector Modulation techniques for these inverters were introduced[25]. This was around the same time that Zhang et al[2, 13] introduced a new 3-Dimensional Space Vector Modulation technique which allowed the independent control of each of the 3 main output phases with respect to the neutral. The concepts which were introduced by Zhang were then later expanded upon by Ryan, Lorenz and De Doncker[11, 12] who produced a mathematical grounding for this new SVM technique.

Later in 2002, Zhang et al presented a fuller version of their original approach[14], and it is this work that this project seeks to extend in conjunction with the SVM technique for matrix converters which was proposed by Casadei[4].

1.3 The 4-leg Matrix Converter

The 4-leg Matrix Converter was, at the start of this project, a novel circuit which had yet to see any work published about it, although this may not be surprising with the majority of research effort in the matrix converter field being concentrated on the practical side to converter implementation, and the relative lack of appreciable applications.

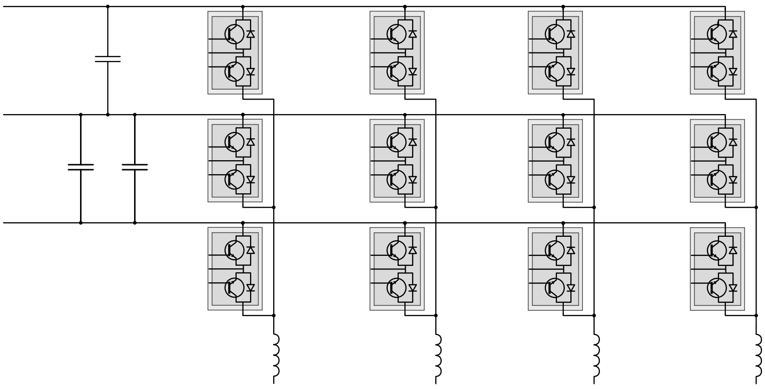


Figure 1.6: A 4-Leg Matrix Converter

It is, quite simply a 3x3 matrix converter with a fourth output leg added to act as the neutral leg, with Figure 1.6 showing the basic layout. The idea being that as with the 4-leg inverters, the extra output leg would enable the converter to be able to power non-linear and unbalanced loads.

At the beginning of this work, although presumed possible, it was unknown as to the ability to control one of these converters by using either the Space Vector Modulation principles, or the Alesina\Venturini methods. The Space Vector Modulation SVM method was chosen to be the basis of the work as it looked to be the most promising due to the existence of similar methods for both the matrix converter and the 4-leg inverter.

1.4 Applications and Limitations

One major application for this type of converter is as part of a field power supply, which is the type of system briefly described above and shown in

Figure 1.1, and which has since entered development by the University of Nottingham[29, 30]. In this application it is able to demonstrate its main advantage over other types of converter in that it is able to supply power for not only balanced 3-phase sources, but also unbalanced and single phase loads. So, along with the advantages that comes from being a matrix converter, so it being compact and able to achieve high power densities due to not requiring a DC link capacitor, means that it is highly suited to this task.

Having the converter in this configuration, running directly from a generator also helps mitigate the biggest drawback of this converter, and that is the quality of the input current when drawing an unbalanced load. As will be demonstrated, once the load becomes unbalanced, due to the direct nature of the converter with it having no energy storage elements unlike a normal inverter, there is no way to buffer the supply from the irregular power drawn, which, if the load is highly unbalanced can lead to distortion of the supply voltage. However, with the standalone setup above these harmonic currents will be drawn from the generator, and by careful speed control of the engine the worst effects of this imbalance can be moderated.

One characteristic that this converter has, is the ability to individually alter its output phase voltages, and frequencies, independently of one another. While this might not immediately appear to have an application, it might be of some use when dealing with fault situations on the load, where the converter would be able to balance the load power per phase individually in order to not affect the supply, however such work fell outside the scope of this project and so will need to be investigated separately.

1.5 Objectives

The objective of this project was simple, to investigate the use of Space Vector Modulation to enable the control a 4-Leg Matrix converter. To this end it would have a number of outcomes:

Firstly, as this is a novel implementation of SVM on a new converter, it would involve the understanding and analysis of the two similar types of space vector modulation that previously existed, that of the 4-leg inverter and the matrix converter, and only from this point would it be possible to extent both methods to cover the 4-leg matrix converter. Then once the theory behind the operation of those converters was understood, it would then be possible to understand the rules which would govern the behaviour of the 4-leg matrix converter. At this point it would then be possible to derive the equations which would govern its operation.

The next objective would be to then produce a set of simulations which would be able to accurately model the behaviour of the 4-leg matrix converter using the derived equations. This would lead to a set of Matlab programs which however this would first require the implementation of the two other converters to prove that the

The final objective in this project is to build and run a working demonstration model of the 4-leg matrix converter. This would involve the evolution of a pre-existing matrix converter design to add the extra output leg, and then require a new set of software to be able to operate and control the converter.

1.6 Thesis Overview

In Chapter 1 the concept of the matrix converter is discussed along with a brief history of the circuit, followed by a short discussion on the 4-leg inverter. This will then follow on to an introduction of the concept of the 4-Leg Matrix converter, followed by short look at a possible implementation, and also a look at its possible drawback.

Chapter 2 then looks into the detail of how a number of different modulation schemes work for the 4-leg inverter, most importantly, it sets out the basis for and the equations which govern the use of space vector modulation with the circuit.

Chapter 3 looks at the other important relative of the 4-leg matrix converter, the normal matrix converter and goes on to examine the two major different methods which enable its control, finishing on the implementation of space vector modulation for the converter.

Chapter 4 then takes a brief look into the different possible commutation strategies which are employed in switched power converters, and matrix converters specifically.

In Chapter 5 the basis behind Space Vector Modulation of a 4-Leg Matrix converter is derived. Starting by looking at both the input and outputs vector spaces, this chapter goes on to see how they are linked together, demonstrating the symmetry between the two and produces the calculations which will select the required switching states and calculate the duty cycles.

Chapter 6 then builds on the work of the previous chapter, taking the derived equations and then initially simulates them mathematically within Matlab, before moving on to perform a number of circuit based simulations within the Saber simulation software. These simulations will be used to validate the derivation, while also becoming a good base for the code which will be required to build an actual converter.

Chapter 7 describes the build of a demonstration converter based on an existing 3x3 matrix converter design. This chapter briefly looks at each of the main parts that goes into building a converter, describing its operation and going into detail where it is required.

Chapter 8 gives a summary of the results from testing the converter. It shows how the output from the converter is very well matched with that expected after the simulations, but then goes on to describe the issues which the converter had during testing, which ultimately meant that the converter failed and stopped testing short of the ideal, but still with enough information to demonstrate that the concept worked well.

Chapter 9 then follows with the conclusions on the work presented here.

Chapter 2

Four-Leg Inverter Modulation

The four-leg inverter is an evolution of the standard three-leg inverter, and was bought about by the need to run non-linear and unbalanced three-phase loads, a fourth-wire as the neutral connection to the load. This was initially configured so that the fourth wire was held at the mid-point of a pair of capacitors[28], with Figure 2.1 showing an example of this layout.

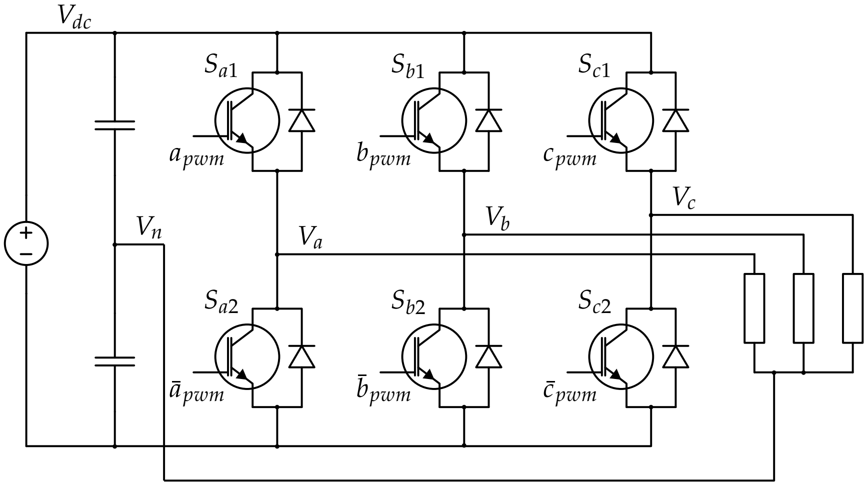
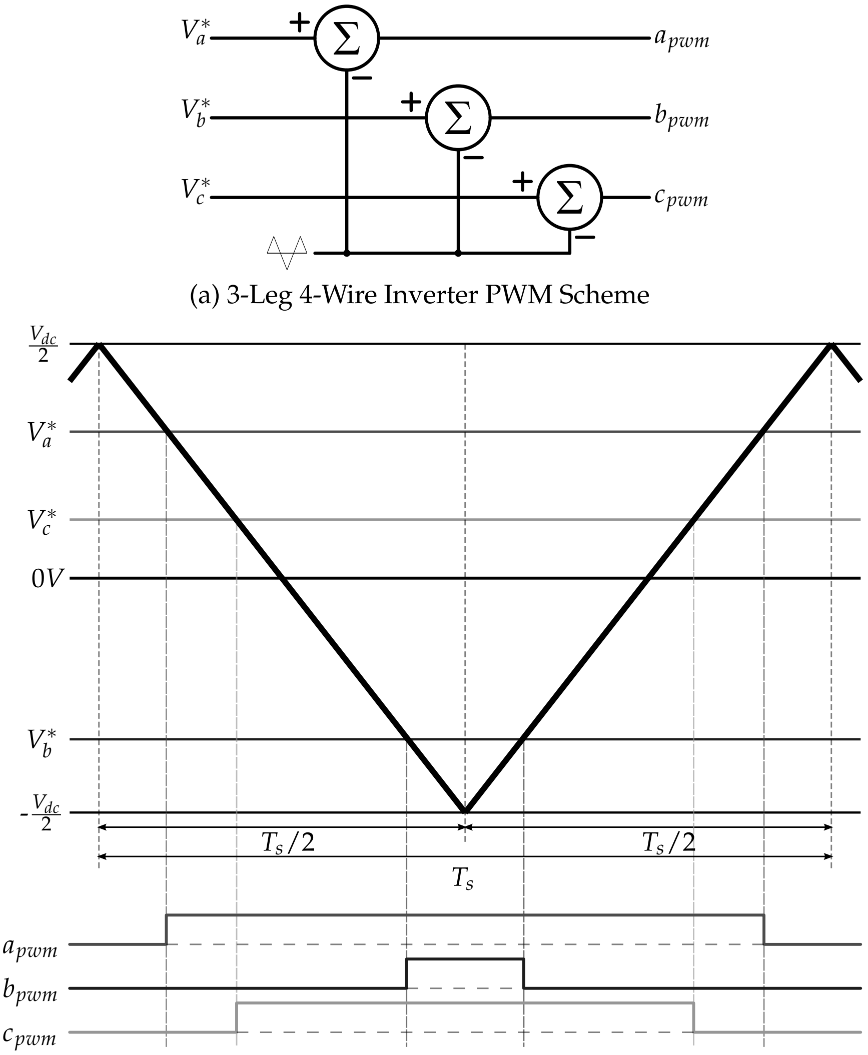


Figure 2.1: 3-Leg 4-Wire Inverter with a Split dc Neutral Connection

As can be seen the pair of capacitors, when charged, hold the voltage of the fourth wire at the midpoint of the DC voltage, with any neutral current being passed to one or the other of these. This arrangement works well when the level of unbalance is low, and so neutral currents are relatively low, as the technique relies upon the capacitors holding the neutral point voltage steady at the mid-point of the DC-link. But as loads become more non-linear or unbalanced then the neutral current can become large, with this current needing to flow through the capacitors and so requiring large value capacitors capable of handling high ripple currents to avoid shifting the neutral point, which if allowed to occur will distort the output voltages.

This is actually a relatively simple circuit to control however, with each of the three switched legs effectively operating as a stand-alone half-bridge single phase inverter, switching its output about the clamped DC mid-point. This means that a simple carrier based pulse-width modulation (PWM) technique can be employed as shown below in Figure 2.10. In this technique, each of the three output phases is driven by a separate voltage demand signal and these are then compared to a triangular carrier waveform. The results of these comparisons are then fed into the respective gate drive circuits, which in turn drive the IGBTs.



(a) 3-Leg 4-Wire Inverter PWM Scheme

(b) 3-Leg 4-Wire Inverter PWM Timing

Figure 2.2: 3-Leg 4-Wire Inverter PWM

However, using such a simple technique shows another drawback of this topology, in that it only allows a maximum voltage transfer ratio of

$0.5V_{dc}$ per phase. Techniques which are used in 3-phase 3-wire inverters to increase the maximum voltage transfer ratio, such as Third Harmonic Injection, are now not really applicable. For example, with third harmonic injection, the injected harmonics present in the output phases will force a shift in the neutral point voltage, at the third harmonic frequency, and so generate a respective neutral current through the capacitors.

2.1 Space Vector Modulation of the 4-Leg 4-Wire Inverter

To overcome the limitations with the above circuit, instead of generating the neutral voltage using a split DC link the neutral voltage can be generated by using a fourth inverter leg[24, 25, 27], leading to the actual 4-leg inverter as shown in Figure 2.3.

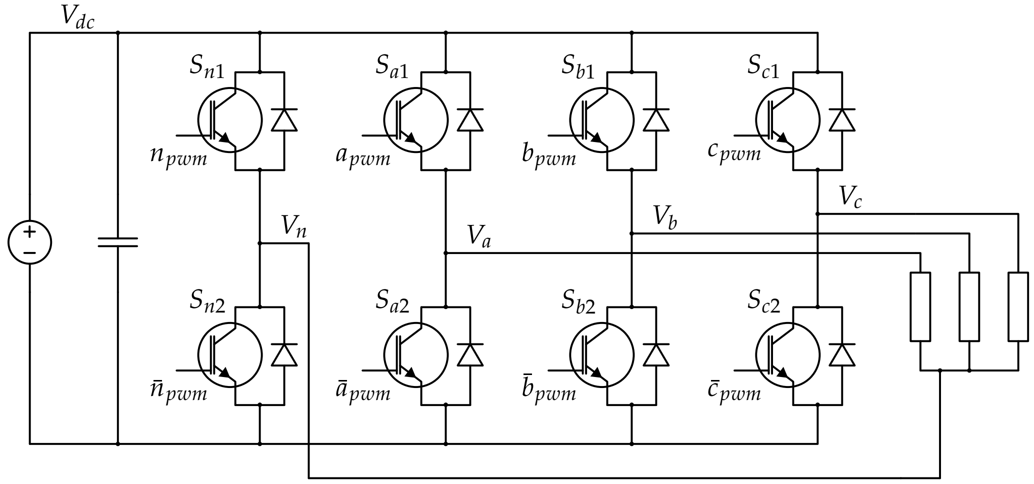


Figure 2.3: 4-Leg 4-Wire Inverter

Now while it is possible to independently control the neutral voltage to mimic that generated by the split capacitor circuit above, this does not make full use of the ability of the extra leg to control the neutral point to any voltage up to the DC link voltage. However, once the neutral voltage is shifted from its mid point the simple PWM control of the 3-Leg 4-wire inverter is no longer enough, as and shift in the neutral voltage needs to be reflected in changes to all 3 phase voltages. As such, the Space Vector Modulation technique which is commonly used in 3-wire inverters was extended by Zhang et al in 1997[2, 13] to allow the control of the 4-Leg inverter. This method is briefly described below, after a short introduction to how Space Vector Modulation on a more normal 3-leg inverter.

So, consider first a standard 3-leg inverter where the relationship between the three phase voltages V_a, V_b and V_c defined by

$$V_a + V_b + V_c = 0 \tag{2.1.1}$$

This constraint means that only 2 of the phase voltage variables are actually independent. Using this means that the three phase voltages can be transformed into a 2-dimensional space, in this case the $\alpha\beta$ space using (2.1.2)

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{2.1.2}$$

With a 3-leg inverter, each leg has only got two possible switching states during operation, that is to switch that output leg to either the positive or the negative of the DC link. With 3 legs, each with 2 possible switching states, this gives 8 possible switching states that can be generated and these are shown in Table 2.1 below.

Table 2.1: 3-Leg Inverter Switching Combinations

	ppp	nnn	pnn	ppn	npn	npp	nnp	pnp
V_a	$\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$
V_b	$\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$
V_c	$\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$-\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$	$\frac{1}{2}V_{dc}$

Transforming the information in Table 2.1 into the 2-dimensional $\alpha\beta$ space using (2.1.2) then gives the results shown in Table 2.2

Table 2.2: 3-Leg Inverter Switching Combinations in $\alpha\beta$ Space

	ppp	nnn	pnn	ppn	npn	npp	nnp	pnp
V_α	0	0	$\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$
V_β	0	0	0	$\frac{1}{\sqrt{3}}V_{dc}$	$\frac{1}{\sqrt{3}}V_{dc}$	0	$-\frac{1}{\sqrt{3}}V_{dc}$	$-\frac{1}{\sqrt{3}}V_{dc}$

The vectors resulting from these 8 switching states can then be plotted onto an Argand diagram, giving the well known result shown in Figure 2.4. This plot shows 6 space vectors placed equidistant from each other around the origin, and these are created by the non-zero switching states. The remaining two vectors, *ppp* and *nnn*, switch the same voltage to each of the three output legs, and as such they are known as the zero voltage states and do not produce a vector within the $\alpha\beta$ space.

Next, consider the demand voltages which are required as an output from the inverter, now because this is a 3-leg inverter this will be a 3-phase set of voltages which meet the requirement set out in equation (2.1.1). Converting this demand voltage into $\alpha\beta$ space using (2.1.2) produces a vector, v_d, as can be seen in Figure 2.4.

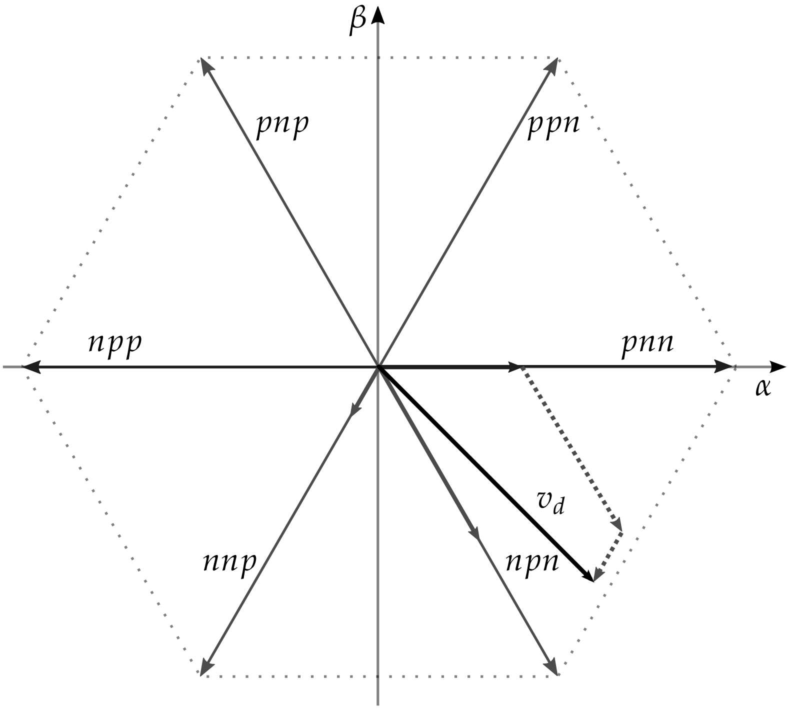


Figure 2.4: 3-Leg Inverter Demand Vector

The basic assumption behind Space Vector Modulation is that this demand vector, v_d, within $\alpha\beta$ space can be synthesised by using the two adjacent space vectors created by the converter switching states. So, in the example shown in Figure 2.4, the demand vector, v_d can be generated using the space vector 1 and 6, which correspond to the switching states pnn and npn.

Now, consider an inverter with an added fourth leg, as shown in Figure 2.3. By adding this extra leg it means that the equation (2.1.1) above is no longer necessarily valid as

$$V_a + V_b + V_c + V_n = 0 \tag{2.1.3}$$

This means that all three phase voltages can now be considered to be truly independent of each other, and so the 2-dimensional $\alpha\beta$ space does not have enough degrees of freedom to be able to describe them fully. To cater for the extra independent variable another degree of freedom needs to be added, and so this is instead transformed into a 3-dimensional space. For this case it is the $\alpha\beta\gamma$ space using (2.1.4)

$$\begin{bmatrix} V_\alpha \\ V_\beta \\ V_\gamma \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \\ V_n \end{bmatrix} \tag{2.1.4}$$

Now, due to inclusion of the extra neutral leg, instead of the 8 possible switching combinations shown in Table 2.1 above for the 3-leg inverter, there are now 16 different ones, which are shown in Table 2.3 below

Table 2.3: 4-Leg Inverter Switching Combinations

	pppp	nnnp	pnnp	ppnp	npnp	nppp	nnpp	pnpp
V_{an}	0	$-V_{dc}$	0	0	$-V_{dc}$	$-V_{dc}$	$-V_{dc}$	0
V_{bn}	0	$-V_{dc}$	$-V_{dc}$	0	0	0	$-V_{dc}$	$-V_{dc}$
V_{cn}	0	$-V_{dc}$	$-V_{dc}$	$-V_{dc}$	$-V_{dc}$	0	0	0
	pppn	nnnn	pnnn	ppnn	npnn	nppn	nnpn	pnpn
V_{an}	V_{dc}	0	V_{dc}	V_{dc}	0	0	0	V_{dc}
V_{bn}	V_{dc}	0	0	V_{dc}	V_{dc}	V_{dc}	0	0
V_{cn}	V_{dc}	0	0	0	0	V_{dc}	V_{dc}	V_{dc}

Transforming this into the $\alpha\beta\gamma$ space using (2.1.4) then gives the results shown in Table 2.4.

Table 2.4: 4-Leg Inverter Switching Combinations in $\alpha\beta\gamma$ Space

	pppp	nnnp	pnnp	ppnp	npnp	nppp	nnpp	pnpp
V_α	0	0	$\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$
V_β	0	0	0	$\frac{1}{\sqrt{3}}V_{dc}$	$\frac{1}{\sqrt{3}}V_{dc}$	0	$-\frac{1}{\sqrt{3}}V_{dc}$	$-\frac{1}{\sqrt{3}}V_{dc}$
V_γ	0	$-V_{dc}$	$-\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$
	pppn	nnnn	pnnn	ppnn	npnn	nppn	nnpn	pnpn
V_α	0	0	$\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$
V_β	0	0	0	$\frac{1}{\sqrt{3}}V_{dc}$	$\frac{1}{\sqrt{3}}V_{dc}$	0	$-\frac{1}{\sqrt{3}}V_{dc}$	$-\frac{1}{\sqrt{3}}V_{dc}$
V_γ	V_{dc}	0	$\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$

The vectors resulting from the 14 non-zero switching states can then be plotted in the $\alpha\beta\gamma$ space as shown in Figure 2.5. The remaining 2 vectors, *pppp* and *nnnn*, switch the same voltage to each of the four output legs, and so they are known as the zero vectors as they do not produce a voltage across the output. As with the vectors for the 3-leg inverter shown in Figure 2.4, the vectors for the 4-leg inverter are all positioned equidistantly from each other within the $\alpha\beta\gamma$ space, and the shape they describe is a type of icositetrahedron (24-faced polyhedron).

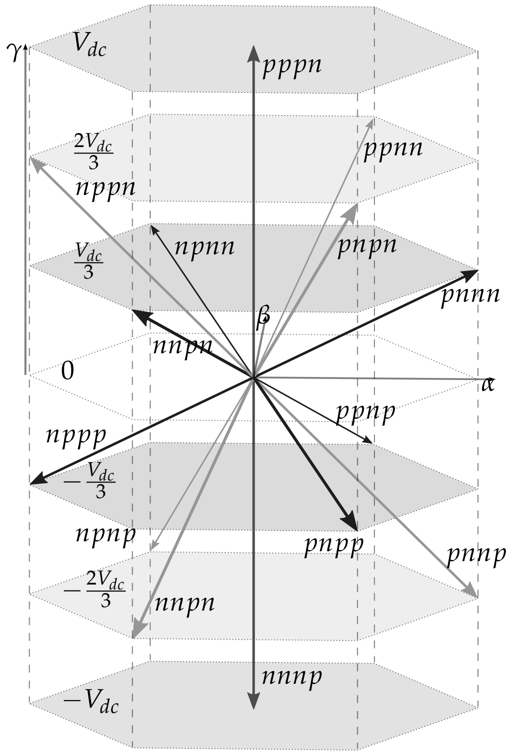


Figure 2.5: 4-Leg Inverter Switching Vectors

Next, consider a set of demand voltages which are required as the output from the inverter. Now, because this is a 4-leg inverter this can be any set of voltages which meet the requirement set out in equation (2.1.3) however, taking the example for the 3-leg inverter above, and applying (2.1.3) now gives

$$\begin{aligned} V_a &= 240.00 \\ V_b &= -327.85 \\ V_c &= 87.84 \\ V_n &= 0 \end{aligned} \tag{2.1.5}$$

Converting this demand voltage into $\alpha\beta\gamma$ space using (2.1.4) produces a vector, v_d,

$$\begin{aligned} V_\alpha &= 240.00 \\ V_\beta &= -240.00 \\ V_\gamma &= 0 \end{aligned} \tag{2.1.6}$$

This vector can be seen in Figure 2.6 labelled as the demand vector, v_d, along with the 3 space vectors required to generate this demand.

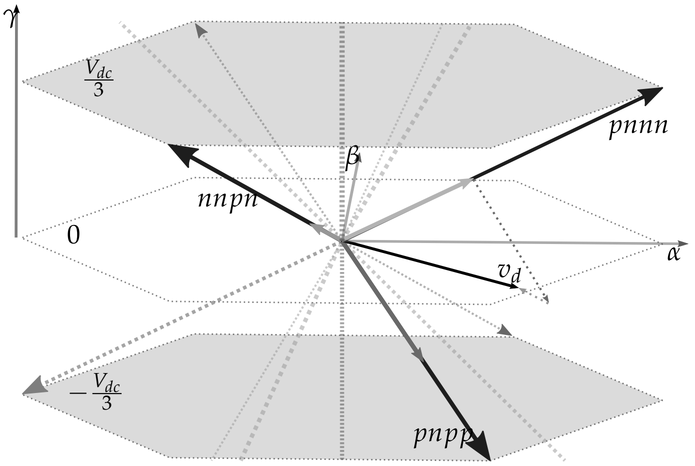


Figure 2.6: 4-Leg Inverter Demand Vector

At any one time the demand vector, v_d, is bound in a space by 3 of the switching vectors, in the same way that the demand vector for the 2-dimensional Space Vector Modulation is bounded by 2. These 3 switching vectors form the vertices of a tetrahedron, the base of which forms one face of the icositetrahedron in the $\alpha\beta\gamma$ switching space. Just as the 3-leg inverter was able to synthesise the demand voltage by switching between the 2 adjacent switching vectors, the 4-leg inverter is able to synthesise its demand voltage by using these 3 switching vectors, which can be seen in Figure 2.7.

Now, while it is a relatively straightforward task to then calculate the duty cycles for those switching states for the 3-leg inverter, it becomes somewhat harder for the 4-leg inverter due to the number of different possible tetrahedrons. The solution proposed by Zhang is to use a look-up table which would define a matrix[2, 13] that could be used to directly calculate the duty cycles, so for the example shown in Figure 2.6, the demand vector would be in Prism IV and Tetrahedron 1, and the duty cycles would be calculated as

$$\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \frac{1}{V_{dc}} \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \\ \frac{1}{2} & -\frac{\sqrt{3}}{2} & -1 \\ \frac{3}{2} & \frac{\sqrt{3}}{2} & 0 \end{bmatrix} \begin{bmatrix} V_\alpha(240.00) \\ V_\beta(-240.00) \\ V_\gamma(0.00) \end{bmatrix} \tag{2.1.7}$$

$$d_z = 1 - d_1 - d_2 - d_3 \tag{2.1.8}$$

Once the duty cycles are known, it is a simple case to produce the re-

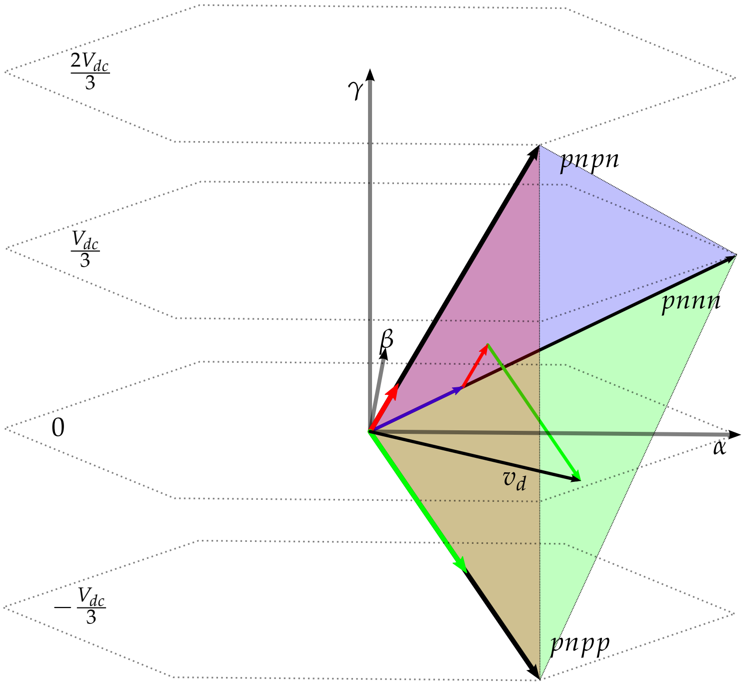


Figure 2.7: 4-Leg Inverter Demand Vector Tetrahedron

spective PWM signals as shown in Figure 2.8 where each duty cycle is for a respective switching state.

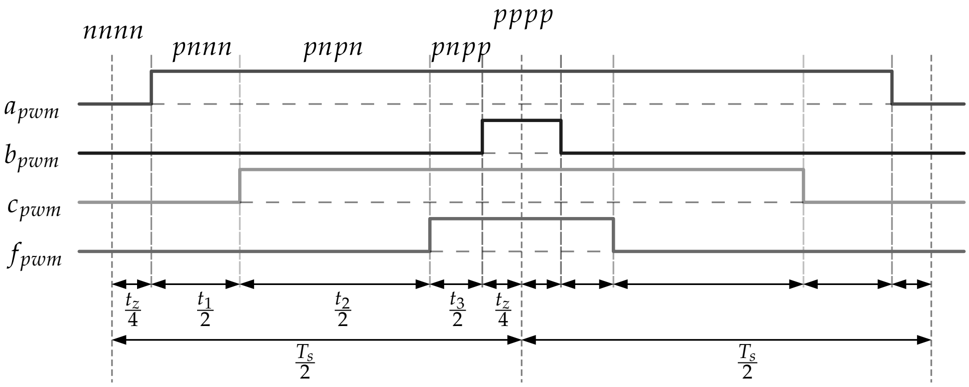


Figure 2.8: 4-Leg Inverter Space Vector Modulation PWM Output

2.2 Carrier Based Pule-Width Modulation of the 4-Leg 4-Wire Inverter

Alongside the Space Vector Modulation technique shown above, a complete carrier based implementation of the 4-leg inverter has been proposed by Kim and Sul[31] in 2004. This technique uses an ′offset voltage′

concept to be able to generate the three independent voltages V_{af}, V_{bf} and V_{cf}.

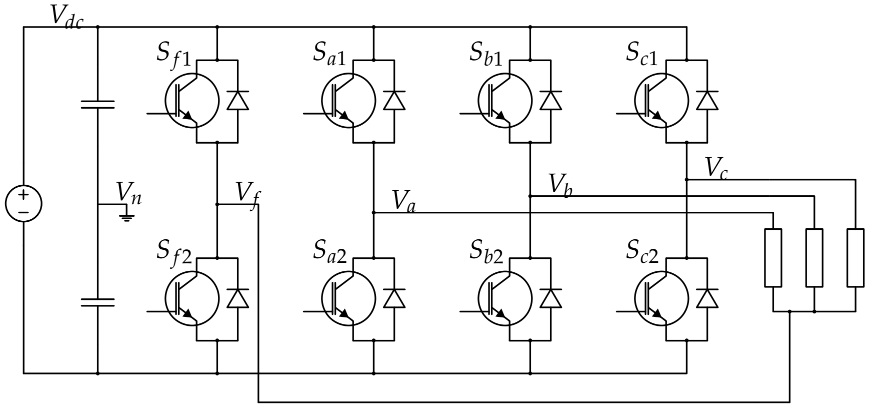


Figure 2.9: 4-Leg 4-Wire Inverter

Using the circuit of the 4-leg inverter as shown in Figure 2.9 the method states that the output line to neutral voltages are constrained by the following equation

$$-V_{dc} \leq V_{af}, V_{bf}, V_{cf} \leq V_{dc} \tag{2.2.1}$$

where each of the output voltages V_{af}, V_{bf} and V_{cf} are defined by

$$\begin{aligned} V_{af} &= V_{an} - V_{fn} \\ V_{bf} &= V_{bn} - V_{fn} \\ V_{cf} &= V_{cn} - V_{fn} \end{aligned} \tag{2.2.2}$$

This allows the use of V_{fn}, the fourth inverter output leg, as an offset voltage which can be manipulated to control the other 3 output leg voltages where

$$-\frac{V_{dc}}{2} \leq V_{an}, V_{bn}, V_{cn} \leq \frac{V_{dc}}{2} \tag{2.2.3}$$

and

$$-\frac{V_{dc}}{2} \leq V_{fn} \leq \frac{V_{dc}}{2} \tag{2.2.4}$$

It can then be shown that while V_{fn} is required to meet the constraints shown in (2.2.4), in fact V_{fn} rarely falls outside the following limits

$$-\frac{V_{dc}}{2} - V_{min} \leq V_{fn} \leq \frac{V_{dc}}{2} - V_{max} \tag{2.2.5}$$

where

$$V_{min} = \mathbf{min}(V_{af}, V_{bf}, V_{cf}) \tag{2.2.6}$$
$$V_{mid} = \mathbf{mid}(V_{af}, V_{bf}, V_{cf}) \tag{2.2.7}$$
$$V_{max} = \mathbf{max}(V_{af}, V_{bf}, V_{cf}) \tag{2.2.8}$$

and this should be limited to

$$V_{max} - V_{min} < V_{dc} \tag{2.2.9}$$

as this would limit the possible offset voltages if the difference between the maximum and minimum output voltages is equal to or greater than the DC link voltage.

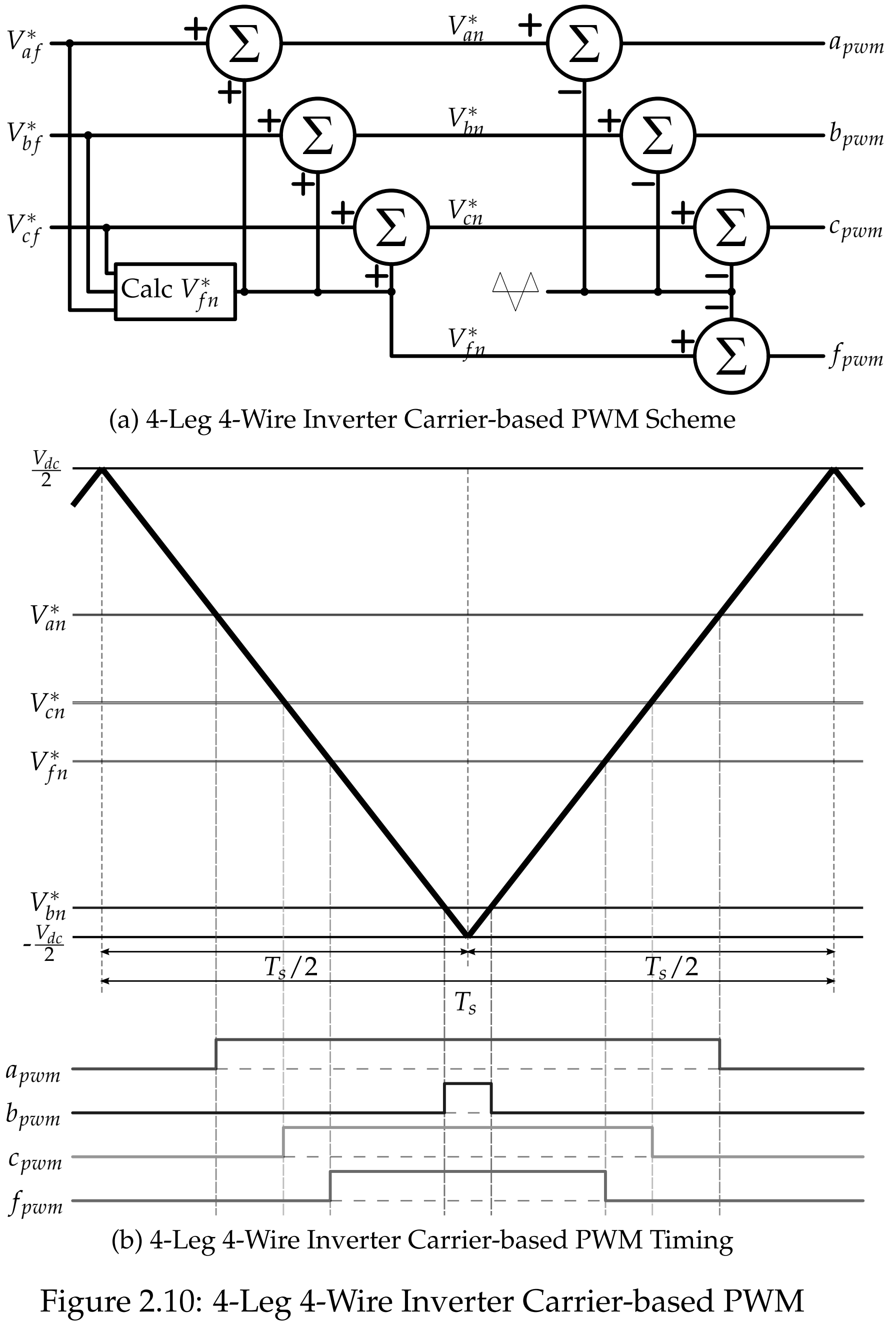
The optimum level for V_{fn} is then be calculated as

$$V_{fn} = \mathbf{mid}\left(-\frac{V_{max}}{2}, -\frac{V_{min}}{2}, -\frac{V_{max}+V_{min}}{2}\right) \tag{2.2.10}$$

and the switching times for each leg calculated as

$$\begin{aligned} T_a &= \frac{T_s}{2} + \frac{V_{an}}{V_{dc}} T_s \\ T_b &= \frac{T_s}{2} + \frac{V_{bn}}{V_{dc}} T_s \\ T_c &= \frac{T_s}{2} + \frac{V_{cn}}{V_{dc}} T_s \\ T_f &= \frac{T_s}{2} + \frac{V_{fn}}{V_{dc}} T_s && (2.2.11) \\ & && (2.2.12) \end{aligned}$$

As demonstrated by Kim and Sul[31], this method gives equivalent performance to the 3-dimensional Space Vector Modulation method as described above.



(a) 4-Leg 4-Wire Inverter Carrier-based PWM Scheme

(b) 4-Leg 4-Wire Inverter Carrier-based PWM Timing

Figure 2.10: 4-Leg 4-Wire Inverter Carrier-based PWM

CHAPTER 3

Matrix Converter Modulation

The basic premise behind the matrix converter is that you have a set of nxp bi-directional switches arranged to connect n input phases to p output legs so that it is possible to connect any output leg to input phase. The most common configuration for this is the 3-phase to 3-phase matrix converter which is shown in Figure 1.4. Due to the inherent bi-directional nature of the converter, with power able to flow in both directions, the labelling of the input and output is essentially arbitrary. However, due to the general modes of operation of the converter it has become the normal convention that the voltage stiff port is designated the input and the current stiff port the output. This designation does however impose a fundamental restriction on how the converter can be switched. This restriction is that no two input phases can be shorted together due to their voltage stiff characteristic, and no output legs can be left open circuit due to their current stiff behaviour.

Prior to the start of this work, the 3x3 matrix converter was the largest matrix converter for which a solution had been proposed, and this chapter will investigate the different modulation strategies which were common at that time, and which were used as the basis of this investigation.

Since the initial publication of the basic derivation on the 4-leg matrix converter [32] there have since been a number of different modulation methods for the 4-leg matrix converter proposed. Since these were not available at the time the work on this started they have not been included in this chapter, however a summary of the different methods will be discussed at the end of the Chapter 5.

3.1 Basic theory

If we define the input and output voltages of the converter as the vectors (3.1.1) and (3.1.2) respectively

$$\mathbf{V_i}(t) = \begin{bmatrix} V_A(t) \\ V_B(t) \\ V_C(t) \end{bmatrix} \tag{3.1.1}$$

$$\mathbf{V_o}(t) = \begin{bmatrix} V_a(t) \\ V_b(t) \\ V_c(t) \end{bmatrix} \tag{3.1.2}$$

then a switching function $\mathbf{S}(t)$ can be defined as

$$\mathbf{S}(t) = \begin{bmatrix} s_{Aa}(t) & s_{Ba}(t) & s_{Ca}(t) \\ s_{Ab}(t) & s_{Bb}(t) & s_{Cb}(t) \\ s_{Ac}(t) & s_{Bc}(t) & s_{Cc}(t) \end{bmatrix} \tag{3.1.3}$$

where a switch s_{np} in its on-state is represented by a 1 and in its off-state by a 0. This relates the output voltages $\mathbf{V_o}(t)$ to the input voltages $\mathbf{V_i}(t)$ so that

$$\mathbf{V_o}(t) = \mathbf{S}(t) \cdot \mathbf{V_i}(t) \tag{3.1.4}$$

To satisfy the restriction that no inputs can be shorted together and no outputs can be open circuit the following must always be true

$$\sum_{n=A,B,C} s_{na}(t) = \sum_{n=A,B,C} s_{nb}(t) = \sum_{n=A,B,C} s_{nc}(t) = 1 \tag{3.1.5}$$

In a similar way the relationship for the input current is given by

$$\mathbf{I_i}(t) = \mathbf{S^T}(t) \cdot \mathbf{I_o}(t) \tag{3.1.6}$$

where the matrix $\mathbf{S}^T$ is the transpose of the matrix $\mathbf{S}$.

The switching matrix $\mathbf{S}$ gives us the switching function at any particular instant but as the switches are operated in sequence over a set sampling period, a more useful value is a modulation index. This states that for each sampling period each input is switched to each output for a time t_{np}, so that

$$m_{np} = \frac{t_{np}}{T_{samp}} \tag{3.1.7}$$

Taking this set of modulation indexes the modulation matrix $\mathbf{M}(t)$ can be constructed such that

$$\mathbf{V_o}(t) = \mathbf{M}(t) \cdot \mathbf{V_i}(t) \tag{3.1.8}$$

and

$$\mathbf{I_i}(t) = \mathbf{M^T}(t) \cdot \mathbf{I_o}(t) \tag{3.1.9}$$

which expand to give

$$\begin{bmatrix} V_a(t) \\ V_b(t) \\ V_c(t) \end{bmatrix} = \begin{bmatrix} m_{Aa}(t) & m_{Ba}(t) & m_{Ca}(t) \\ m_{Ab}(t) & m_{Bb}(t) & m_{Cb}(t) \\ m_{Ac}(t) & m_{Bc}(t) & m_{Cc}(t) \end{bmatrix} \begin{bmatrix} V_A(t) \\ V_B(t) \\ V_C(t) \end{bmatrix} \tag{3.1.10}$$

and

$$\begin{bmatrix} I_A(t) \\ I_B(t) \\ I_C(t) \end{bmatrix} = \begin{bmatrix} m_{Aa}(t) & m_{Ab}(t) & m_{Ac}(t) \\ m_{Ba}(t) & m_{Bb}(t) & m_{Bc}(t) \\ m_{Ca}(t) & m_{Cb}(t) & m_{Cc}(t) \end{bmatrix} \begin{bmatrix} I_a(t) \\ I_b(t) \\ I_c(t) \end{bmatrix} \tag{3.1.11}$$

Once again, to satisfy the requirement that no input phase is short circuited and no output leg is open circuit

$$\sum_{n=A,B,C} m_{na}(t) = \sum_{n=A,B,C} m_{nb}(t) = \sum_{n=A,B,C} m_{nc}(t) = 1 \tag{3.1.12}$$

where $0 \leq m_{np} \leq 1$.

These are the basic starting assumptions on which the different modulation strategies as based.

3.2 Alesina/Venturini Method

This scheme is was proposed by Alesina and Venturini in 1981[5], and it is a modulation method which allows direct control over the output voltage and input power factor.

It states that with a given set of sinusoidal input voltages $\mathbf{V}_i(t)$ i, and assuming that the output currents are also sinusoidal and given by $\mathbf{I}_o(t)$

$$V_i(t) = \begin{bmatrix} v_i \cos(\omega_i t) \\ v_i \cos(\omega_i t + \frac{2}{3}\pi) \\ v_i \cos(\omega_i t + \frac{4}{3}\pi) \end{bmatrix} \tag{3.2.1}$$

$$I_o(t) = \begin{bmatrix} i_o \cos(\omega_o t + \phi_o) \\ i_o \cos(\omega_o t + \frac{2}{3}\pi + \phi_o) \\ i_o \cos(\omega_o t + \frac{4}{3}\pi + \phi_o) \end{bmatrix} \tag{3.2.2}$$

From this a suitable modulation matrix $\mathbf{M}(t)$ can be determined such that the required set of input currents $\mathbf{I}_i(t)$ and output voltages $\mathbf{V}_o(t)$ are both sinusoidal, while also being constrained by the modulation restriction stated in (3.1.12).

$$V_o(t) = \begin{bmatrix} v_o \cos(\omega_o t) \\ v_o \cos(\omega_o t + \frac{2}{3}\pi) \\ v_o \cos(\omega_o t + \frac{4}{3}\pi) \end{bmatrix} \tag{3.2.3}$$

$$I_i(t) = \begin{bmatrix} i_i \cos(\omega_i t + \phi_i) \\ i_i \cos(\omega_i t + \frac{2}{3}\pi + \phi_i) \\ i_i \cos(\omega_i t + \frac{4}{3}\pi + \phi_i) \end{bmatrix} \tag{3.2.4}$$

The solution to this the generalised equation for the modulation function $\mathbf{M}(t)$ as

$$\begin{aligned} \mathbf{M}(t) = \quad & \tfrac{1}{3}\alpha_1 \left\{ \begin{matrix} 1+2qCS(0) & 1+2qCS(-\frac{2}{3}\pi) & 1+2qCS(-\frac{4}{3}\pi) \\ 1+2qCS(-\frac{4}{3}\pi) & 1+2qCS(0) & 1+2qCS(-\frac{2}{3}\pi) \\ 1+2qCS(-\frac{2}{3}\pi) & 1+2qCS(-\frac{4}{3}\pi) & 1+2qCS(0) \end{matrix} \right\} \\ & +\tfrac{1}{3}\alpha_2 \left\{ \begin{matrix} 1+2qCA(0) & 1+2qCS(-\frac{2}{3}\pi) & 1+2qCS(-\frac{4}{3}\pi) \\ 1+2qCS(-\frac{2}{3}\pi) & 1+2qCS(-\frac{4}{3}\pi) & 1+2qCS(0) \\ 1+2qCS(-\frac{4}{3}\pi) & 1+2qCS(0) & 1+2qCS(-\frac{2}{3}\pi) \end{matrix} \right\} \end{aligned} \tag{3.2.5}$$

where

$$\begin{aligned}
&CS(x) = \cos\left[\omega_M t + x\right] \\
&CA(x) = \cos\left[-(\omega_M + 2\omega_i)t + x\right] \\
&\omega_M = \omega_o - \omega_i \\
&\alpha_1 = \tfrac{1}{2}\left[1 + \tan(\phi_i) \cdot \cot(\phi_o)\right] \\
&\alpha_2 = 1 - \alpha_1 = \tfrac{1}{2}\left[1 - \tan(\phi_i) \cdot \cot(\phi_o)\right] \\
&q = \tfrac{v_o}{v_i},
\end{aligned}$$

with

$$\begin{aligned}
&\alpha_1 \geq 0 \\
&\alpha_2 \geq 0 \\
&0 \leq q \leq \tfrac{1}{2},
\end{aligned}$$

Solving the above equation gives the required modulation indices for each of the switches, and so can be used to directly drive the converter. There is however one problem with this method, which is shown in the last equation, and that is the maximum voltage transfer ratio q is only 0.5, meaning that the output voltage can only be a maximum of half the input.

3.3 Alesina/Venturini Optimum Method

To overcome the low voltage transfer ratio found in the original Alesina/Venturini method it was proposed that the transfer ratio could be increased by adding a proportion of the third harmonic into the output frequency, and due to the balanced nature of the loads being driven, this third harmonic would boost the output voltage yet not feed back into the supply side current. This method was formalised in 1989 by Alesina and Venturini which became known as the Optimum Method[6] and produces a voltage transfer ratio, q, of $\frac{\sqrt{3}}{2}$ or 0.866.

By using the same input voltage (3.2.1), input current (3.2.4) and output current (3.2.2) as in the earlier method, and now requiring the output voltage to be

$$V_o(t) = \begin{bmatrix} v_o\cos(\omega_o t) + \frac{1}{4}v_i\cos(3\omega_i t) - \frac{1}{6}v_o\cos(3\omega_o t) \\ v_o\cos(\omega_o t + \frac{2}{3}\pi) + \frac{1}{4}v_i\cos(3\omega_i t) - \frac{1}{6}v_o\cos(3\omega_o t) \\ v_o\cos(\omega_o t + \frac{4}{3}\pi) + \frac{1}{4}v_i\cos(3\omega_i t) - \frac{1}{6}v_o\cos(3\omega_o t) \end{bmatrix} \tag{3.3.1}$$

Solving for the modulation matrix $\mathbf{M}(t)$ then gives the following from[6]: Let $Z_\alpha^\beta(\gamma)$ be a function of time defined as

$$Z_\alpha^\beta(\gamma)(t) = \cos\left((\alpha\omega_o + \beta_i)t + \gamma\frac{\pi}{3}\right) \tag{3.3.2}$$

and

$$\begin{aligned} m(x_1, x_2, x_3, x_4, x_5, x_6) =& \frac{1}{3}\Bigg\{1 + \frac{\sqrt{3}}{2}p\Bigg[Z_1^1(x_1) + Z_1^{-1}(x_2) \\ & - \frac{1}{6}Z_3^1(x_3) - \frac{1}{6}Z_3^{-}1(x_4) \\ & + sgn(p)\left(-\frac{1}{6\sqrt{3}}Z_0^4(x_5) + \frac{7}{6\sqrt{3}}Z_0^2(x_6)\right)\Bigg] \\ & + a_1 Z_1^1(x_1) + a_2 Z_1^{-}1(x_2)\Bigg\} \end{aligned} \tag{3.3.3}$$

where

$$\begin{aligned} &\Theta = \tfrac{\tan(v_i)}{\tan(v_0)} \\ &a = 2|\Theta|\tfrac{v_o}{v_i} \\ &p = \tfrac{\left(2\frac{v_o}{v_i} - a\right)}{\sqrt{3}} \\ &a_1 = a \text{ and } a_2 = 0 \text{ if } \Theta < 0 \\ &a_1 = 0 \text{ and } a_2 = a \text{ if } \Theta > 0 \\ &a_1 = a_2 = 0 \text{ if } \Theta = 0, \end{aligned}$$

then

$$\mathbf{M}(t) = \begin{bmatrix} m(0,0,0,0,0,0) & m(2,4,2,4,2,4) & m(4,2,4,2,4,2) \\ m(2,2,0,0,0,0) & m(4,0,2,4,2,4) & m(0,4,4,2,4,2) \\ m(4,4,0,0,0,0) & m(0,2,2,4,2,4) & m(2,0,4,2,4,2) \end{bmatrix} \tag{3.3.4}$$

Both this set of equations and the ones from previous Alesina/Venturini method look complicated, as would be their direct implementation, however both are simplified by making the assumption that having a unity input power is beneficial. But, due to the complexities in this approach other methods were sought out with the Space Vector Modulation method being chief amongst them.

3.4 Space Vector Modulation

While the Space Vector representation had previously been used for inverter control it was not until Huber and Borojevic in 1989[7–10] where this method was proposed for use with matrix converters. This was then extended with a paper by Casadei at al in 2002[3, 4], giving a full Space Vector Modulation strategy that controlled both the output voltage and input power factor.

This space vector modulation approach to matrix converter modulation is all based on the space vector representation of the input and output voltages and currents at any one instant. For a 3-phase set of line-to-neutral voltages this is given by

$$\mathbf{V}(t) = \frac{2}{3}\left(v_a + av_b + a^2 v_c\right) \tag{3.4.1}$$

where

$$a = e^{j\frac{2\pi}{3}} \tag{3.4.2}$$

Plotting the three vectors that make up $\mathbf{V}(t)$ on an Argand diagram gives a set of 3 vectors spaced evenly 120° apart due to the a and a^2 terms. Plotting $\mathbf{V}(t)$ on the same diagram then produces a vector of constant length, v_o rotating about the origin at the output frequency ω_o and with a phase angle of α_o, as shown in Figure 3.1.

The basis of the Space Vector Modulation technique is that the output voltage $\mathbf{V}_o(t)$, which is expressed in the above space vector form (3.4.1), can be generated by switching between adjacent space vectors and producing a time averaged value over a switching period that is equal to the required output space vector. This is the same technique as used in conventional inverters but by adding in the control of the input current it allows complete control over the operation of a matrix converter. However, instead of the 8 possible switching states of the 3-phase inverter,

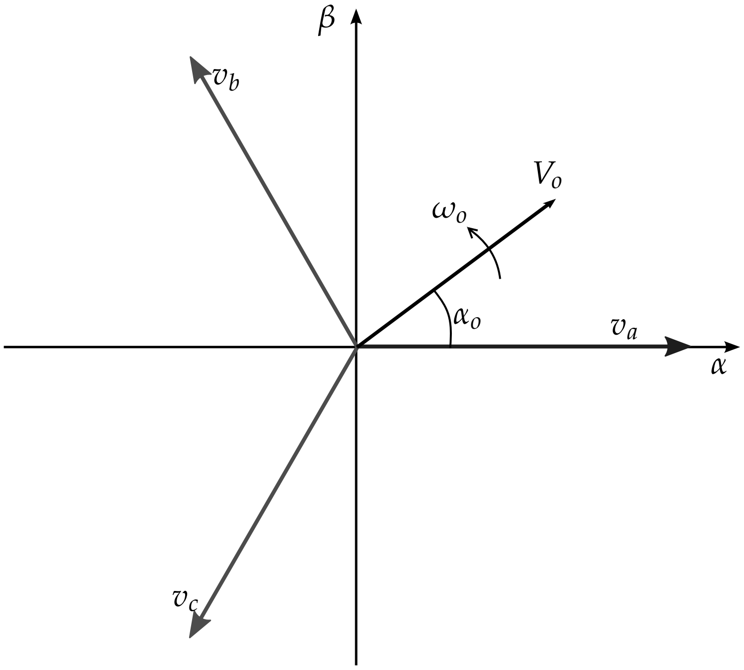


Figure 3.1: Base set of SVM Vectors for a Balanced 3-Phase Set plotted in the $\alpha\beta$ plane

there are now 27 possible switching states. This is because instead of just switching the DC link onto the outputs, the converter has the three input phases to use. Now, unlike with the inverter, not all of the switching states are usable and they fall in the following three groups

- Group I : Where all 3 input phases are switched to the output, thus producing a constant amplitude vector which rotates about the origin at the input frequency.
- Group II : Where any 2 input phases are switched to the output, meaning that two output legs are connected to the same phase. This gives a vector of a fixed angular displacement but with varying amplitude.
- Group III : Where only a single phase is switched to all 3 outputs, meaning that there is no difference in voltage between the output legs. This gives a vector of zero length placed at the origin.

Of these three groups, only Groups II and III, which are shown in Table 3.1 are required in this method, with the rotating Group I switching states not being used as they cannot readily be employed to generate the required output voltages.

Table 3.1: 3x3 Matrix Converter Stationary and Zero Switching Combinations

Vector No.	Leg **a**	Leg **b**	Leg **c**	V_{ab}	V_{bc}	V_{ca}
$+1$	A	B	B	V_{AB}	0	$-V_{AB}$
-1	B	A	A	$-V_{AB}$	0	V_{AB}
$+2$	B	C	C	V_{BC}	0	$-V_{BC}$
-2	C	B	B	$-V_{BC}$	0	V_{BC}
$+3$	C	A	A	V_{CA}	0	$-V_{CA}$
-3	A	C	C	$-V_{CA}$	0	V_{CA}
$+4$	B	A	B	$-V_{AB}$	V_{AB}	0
-4	A	B	A	V_{AB}	$-V_{AB}$	0
$+5$	C	B	C	$-V_{BC}$	V_{BC}	0
-5	B	C	B	V_{BC}	$-V_{BC}$	0
$+6$	A	C	A	$-V_{CA}$	V_{CA}	0
-6	C	A	C	V_{CA}	$-V_{CA}$	0
$+7$	B	B	A	0	$-V_{AB}$	V_{AB}
-7	A	A	B	0	V_{AB}	$-V_{AB}$
$+8$	C	C	B	0	$-V_{BC}$	V_{BC}
-8	B	B	C	0	V_{BC}	$-V_{BC}$
$+9$	A	A	C	0	$-V_{CA}$	V_{CA}
-9	C	C	A	0	V_{CA}	$-V_{CA}$
0_1	A	A	A	0	0	0
0_2	B	B	B	0	0	0
0_3	C	C	C	0	0	0

In the same way as was done for the switching states for the 4-leg inverter, the switching states in Table 3.1 are then transformed into the $\alpha\beta$ plane using the transform

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{3.4.3}$$

to give the results shown in Table 3.2.

If these vectors are now plotted in the $\alpha\beta$ planes, then the familiar hexagonal shape from the standard SVM used within inverters can be seen. Figure 3.2 and 3.3 show these plots.

As stated above, this method works by identifying a set of 2 output space vectors that need to be generated which sit either side of the required output voltage vector. For each of these space vectors there are then

Table 3.2: 3x3 Matrix Converter Stationary and Zero Switching Combinations transformed in the $\alpha\beta$ Plane

Vector No.	v_o	α_o	i_i	β_i
$+1$	$\frac{2}{3}V_{AB}$	0	$\frac{2}{\sqrt{3}}i_a$	$-\frac{\pi}{6}$
-1	$-\frac{2}{3}V_{AB}$	0	$-\frac{2}{\sqrt{3}}i_a$	$-\frac{\pi}{6}$
$+2$	$\frac{2}{3}V_{BC}$	0	$\frac{2}{\sqrt{3}}i_a$	$\frac{\pi}{2}$
-2	$-\frac{2}{3}V_{BC}$	0	$-\frac{2}{\sqrt{3}}i_a$	$\frac{\pi}{2}$
$+3$	$\frac{2}{3}V_{CA}$	0	$\frac{2}{\sqrt{3}}i_a$	$\frac{7\pi}{6}$
-3	$-\frac{2}{3}V_{CA}$	0	$-\frac{2}{\sqrt{3}}i_a$	$\frac{7\pi}{6}$
$+4$	$\frac{2}{3}V_{AB}$	$\frac{2\pi}{3}$	$\frac{2}{\sqrt{3}}i_b$	$-\frac{\pi}{6}$
-4	$-\frac{2}{3}V_{AB}$	$\frac{2\pi}{3}$	$-\frac{2}{\sqrt{3}}i_b$	$-\frac{\pi}{6}$
$+5$	$\frac{2}{3}V_{BC}$	$\frac{2\pi}{3}$	$\frac{2}{\sqrt{3}}i_b$	$\frac{\pi}{2}$
-5	$-\frac{2}{3}V_{BC}$	$\frac{2\pi}{3}$	$-\frac{2}{\sqrt{3}}i_b$	$\frac{\pi}{2}$
$+6$	$\frac{2}{3}V_{CA}$	$\frac{2\pi}{3}$	$\frac{2}{\sqrt{3}}i_b$	$\frac{7\pi}{6}$
-6	$-\frac{2}{3}V_{CA}$	$\frac{2\pi}{3}$	$-\frac{2}{\sqrt{3}}i_b$	$\frac{7\pi}{6}$
$+7$	$\frac{2}{3}V_{AB}$	$\frac{4\pi}{3}$	$\frac{2}{\sqrt{3}}i_c$	$-\frac{\pi}{6}$
-7	$-\frac{2}{3}V_{AB}$	$\frac{4\pi}{3}$	$-\frac{2}{\sqrt{3}}i_c$	$-\frac{\pi}{6}$
$+8$	$\frac{2}{3}V_{BC}$	$\frac{4\pi}{3}$	$\frac{2}{\sqrt{3}}i_c$	$\frac{\pi}{2}$
-8	$-\frac{2}{3}V_{BC}$	$\frac{4\pi}{3}$	$-\frac{2}{\sqrt{3}}i_c$	$\frac{\pi}{2}$
$+9$	$\frac{2}{3}V_{CA}$	$\frac{4\pi}{3}$	$\frac{2}{\sqrt{3}}i_c$	$\frac{7\pi}{6}$
-9	$-\frac{2}{3}V_{CA}$	$\frac{4\pi}{3}$	$-\frac{2}{\sqrt{3}}i_c$	$\frac{7\pi}{6}$
0_1	0	0	0	0
0_2	0	0	0	0
0_3	0	0	0	0

6 switching states that can generate this output voltage vector depending on the sign of the input voltage. However, each of these switching states generates a different input current vector, as shown in Figures 3.2 and 3.3, and it is this link between the input current and the output voltage that allows the selection of the switching states, and thus control over the converter. Due to the number of possible switching states it is always possible to find a set of 4 states that can generate the output voltage required while also controlling the phase of the input current. This method is set out by Casadei [3, 4] and is generated as follows.

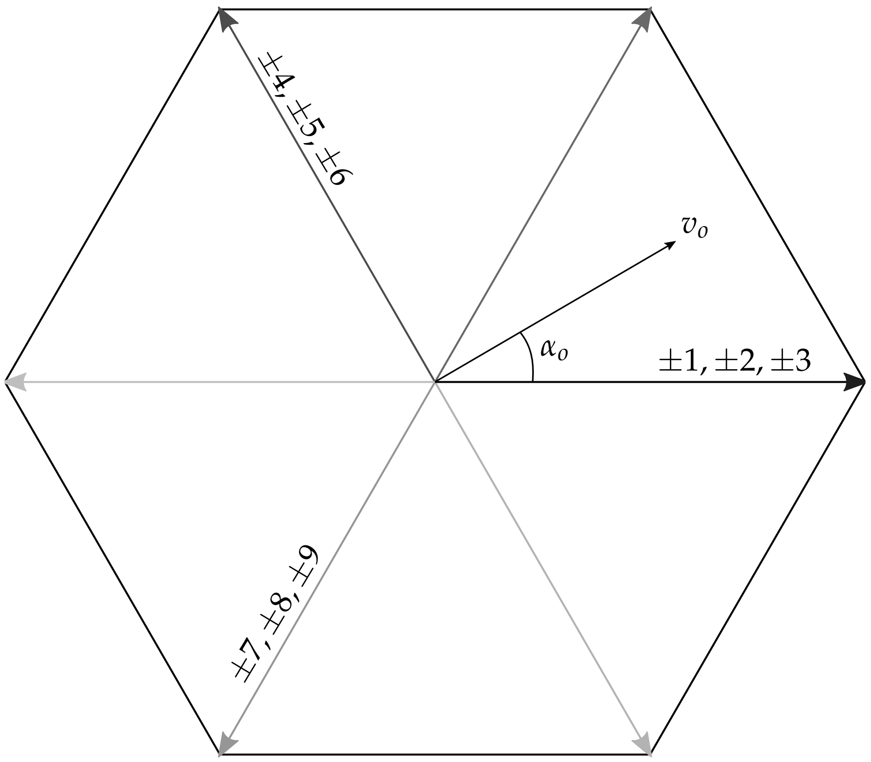


Figure 3.2: Output Voltage Vectors plotted in the $\alpha\beta$ plane

Figure 3.4 shows the input current and output voltage sectors which contain their respective vectors, each with its pair of bounding space vectors, and from this it is possible to write down the equations for the lengths of the output voltage space vectors as

$$\begin{aligned}
\vec{v_o'} &= \vec{v_o^I}\delta^I + \vec{v_o^{II}}\delta^{II} \\
&= \frac{2}{\sqrt{3}} v_o \cos\left(\hat{\alpha_o} - \frac{\pi}{3}\right) e^{j\left[(K_v-1)\frac{\pi}{3}+\frac{\pi}{3}\right]} \qquad (3.4.4) \\
\vec{v_o''} &= \vec{v_o^{III}}\delta^{III} + \vec{v_o^{IV}}\delta^{IV} \\
&= \frac{2}{\sqrt{3}} v_o \cos\left(\hat{\alpha_o} + \frac{\pi}{3}\right) e^{j\left[(K_v-1)\frac{\pi}{3}\right]} \qquad (3.4.5)
\end{aligned}$$

where δ^I to δ^{IV} are the duty cycles for the 4 switching states.

Now, from the current sector plot in Figure 3.4, the equations governing the behaviour of the input current can be written down as

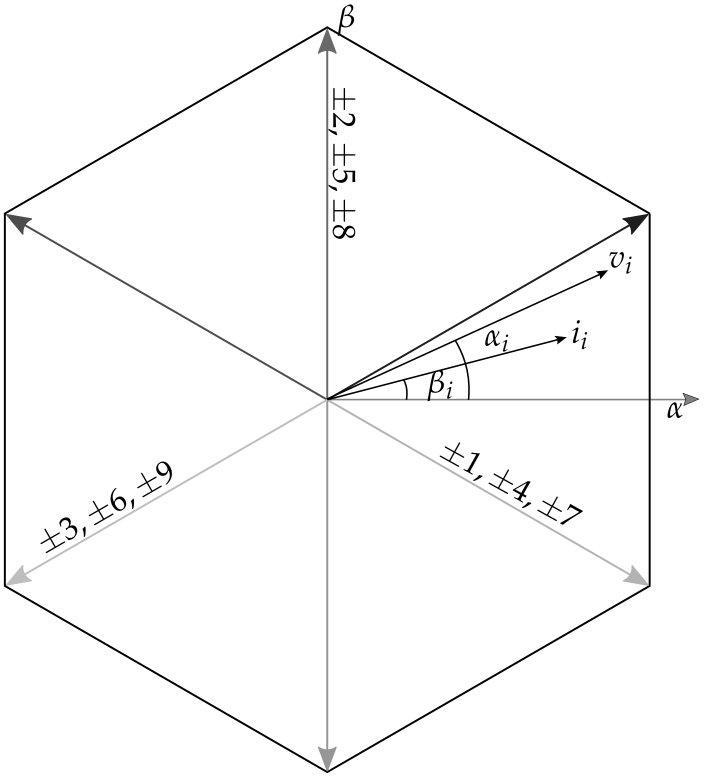


Figure 3.3: Input Current Vectors plotted in the $\alpha\beta$ plane

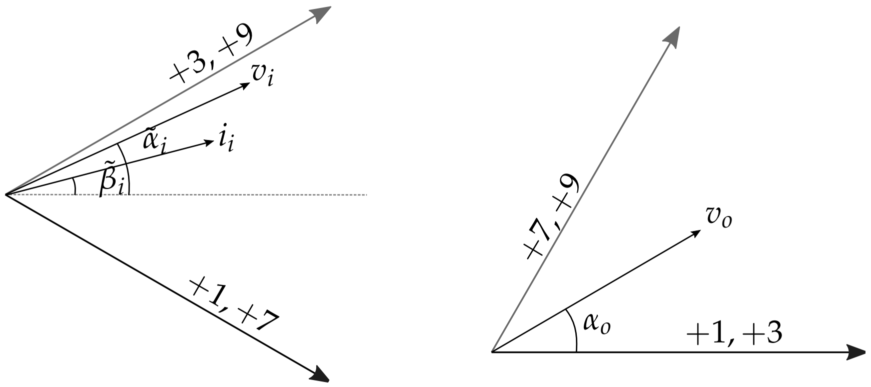


Figure 3.4: Input Current and Output Voltage Sectors, showing the required switching states

$$\left(i_i^{\vec{I}}\delta^{I} + i_i^{\vec{I}I}\delta^{II}\right)\cdot je^{j\beta_i}e^{j(K_i-1)\frac{\pi}{3}} = 0 \tag{3.4.6}$$

$$\left(i_i^{\vec{I}II}\delta^{III} + i_i^{\vec{I}V}\delta^{IV}\right)\cdot je^{j\beta_i}e^{j(K_i-1)\frac{\pi}{3}} = 0 \tag{3.4.7}$$

Solving equations (3.4.4) to (3.4.7) with respect to the duty cycles then gives us

$$\delta^{I} = (-1)^{K_v+K_i+1}\frac{2}{\sqrt{3}}q\frac{\cos\left(\tilde{\alpha}_o - \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)}{\cos\phi_i} \tag{3.4.8}$$

$$\delta^{II} = (-1)^{K_v+K_i}\frac{2}{\sqrt{3}}q\frac{\cos\left(\tilde{\alpha}_o - \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)}{\cos\phi_i} \tag{3.4.9}$$

$$\delta^{III} = (-1)^{K_v+K_i}\frac{2}{\sqrt{3}}q\frac{\cos\left(\tilde{\alpha}_o + \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)}{\cos\phi_i} \tag{3.4.10}$$

$$\delta^{IV} = (-1)^{K_v+K_i+1}\frac{2}{\sqrt{3}}q\frac{\cos\left(\tilde{\alpha}_o + \frac{\pi}{3}\right)\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)}{\cos\phi_i} \tag{3.4.11}$$

Taking the sum of these duty cycles then gives

$$|\delta^{I}| + |\delta^{II}| + |\delta^{III}| + |\delta^{IV}| \leq 1 \tag{3.4.12}$$

and from here it is possible to the calculate the maximum possible transfer ratio, q, as

$$q \leq \frac{\sqrt{3}}{2}\frac{|\cos\phi_i|}{\cos\tilde{\beta}_i\cos\tilde{\alpha}_o} \tag{3.4.13}$$

These equations, with those described in Section 2.1, are going to form the basis of the derivation of space vector modulation of the 4-leg matrix converter.

CHAPTER 4

Commutation Methods

As has been stated several times, there are two fundamental rules for the operation of a matrix converter that arise from the voltage-stiff input and current-stiff output characteristics. The first is that no two input phases can be shorted together and the second is that no output phase can go open circuit. Because of these restrictions, and due to the bi-directional nature of the switches that are needed for the converter to operate, it means that there is a problem with the two most commonly used commutation strategies for the inverter, Overlap and Dead-time.

This chapter will first look at the Overlap and Dead-time strategies, and explain why neither of these is really suitable for the needs of the matrix converter, and will then go on to look at a number of other types which do not have the drawbacks of the first two.

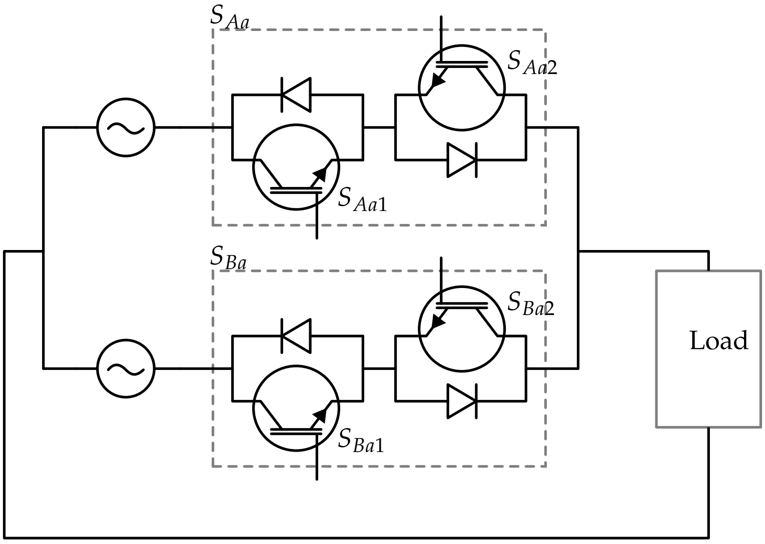
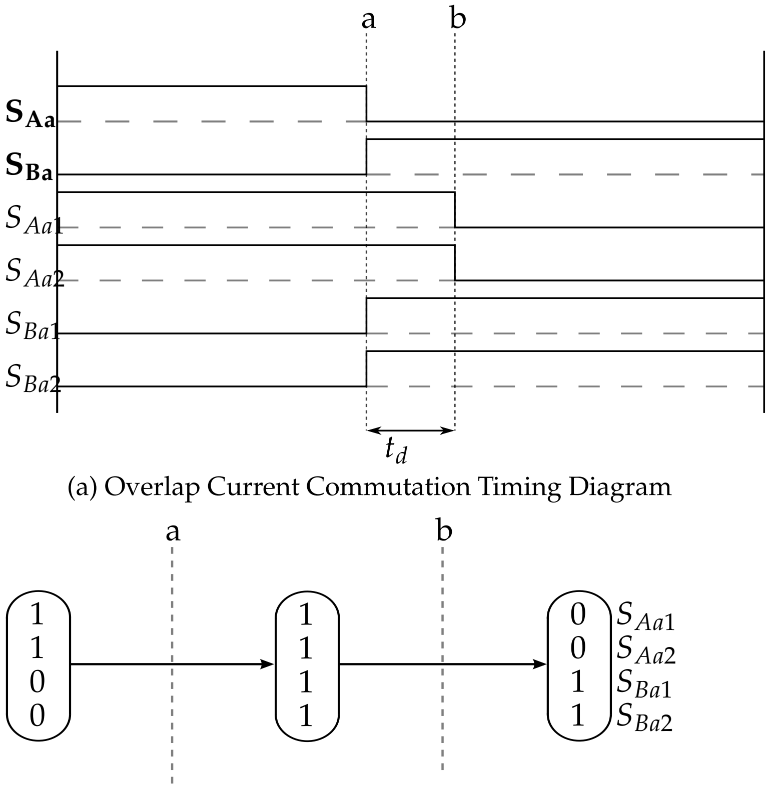


Figure 4.1: Segment from a Matrix Converter Circuit

Figure 4.1 shows a small portion of a matrix converter circuit which will be used to demonstrate the various different types of commutation technique. The circuit contains two bi-directional switches, $\mathbf{S_{Aa}}$ and $\mathbf{S_{Ba}}$, each made up of a pair of IGBTs labelled 1 and 2, which connect two AC voltage source to a load which contains some inductance. For each commutation strategy there is a timing diagram showing how the IGBTs are being switched, with a high level turning that device on, and alongside the timing diagram there is a state representation of the switching signals. These both aim to give an easy understanding of some of the different types of commutation that are possible.

4.1 Overlap Commutation

Overlap commutation is a relatively simple method where the incoming devices are switched on before the outgoing devices are switched off, and is demonstrated in Figure 4.2.



(a) Overlap Current Commutation Timing Diagram

(b) Overlap Current Commutation State Diagram

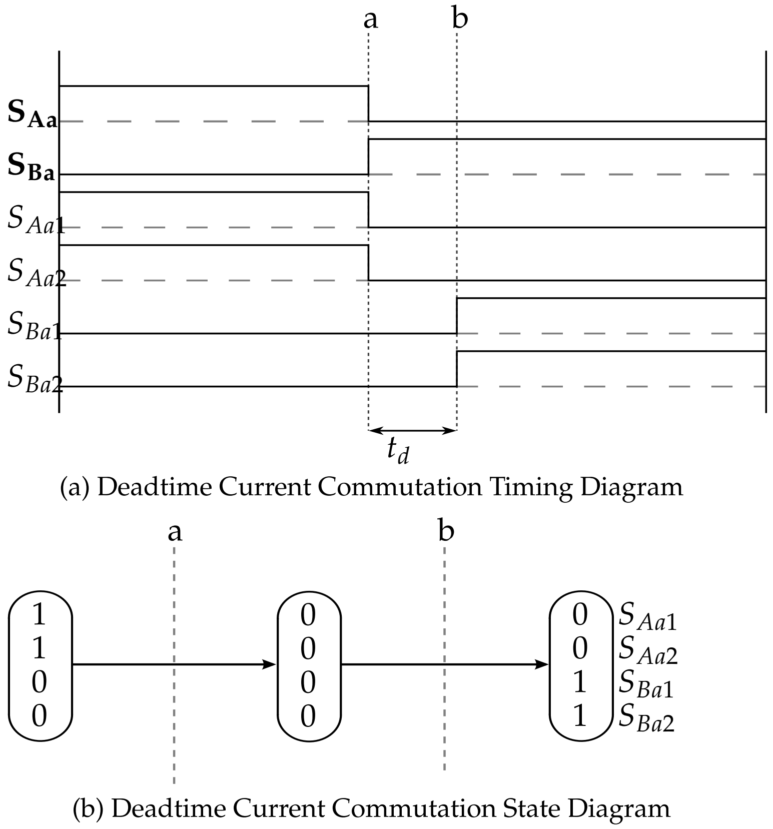
Figure 4.2: Overlap Current Commutation

As can be seen this method ensures that the output legs are always connected, but it also means that two of the input phases are shorted together, thus breaking the first rule specified above. While this is acceptable for use in an inverter, where all legs are connected to the same

DC link voltage, this is not acceptable for the matrix converter. While it is possible to overcome the problem caused by shorting the input phases by the inclusion of large inductors on the input lines to limit the rise in input currents from shorting the phases together, it is not a favoured solution as these inductors will not only need to be large, they will also be expensive and heavy, and so remove one of the advantages of matrix converters, that of being compact and an all silicon solution.

4.2 Dead-time Commutation

Dead-time commutation is also relatively simple, and can almost be seen as the opposite of overlap commutation, in that instead of switching the incoming device on before turning the outgoing device off, this method turns off the outgoing device before turning on the incoming one, thus giving a dead-time between conducting devices.



(a) Deadtime Current Commutation Timing Diagram

(b) Deadtime Current Commutation State Diagram

Figure 4.3: Deadtime Current Commutation

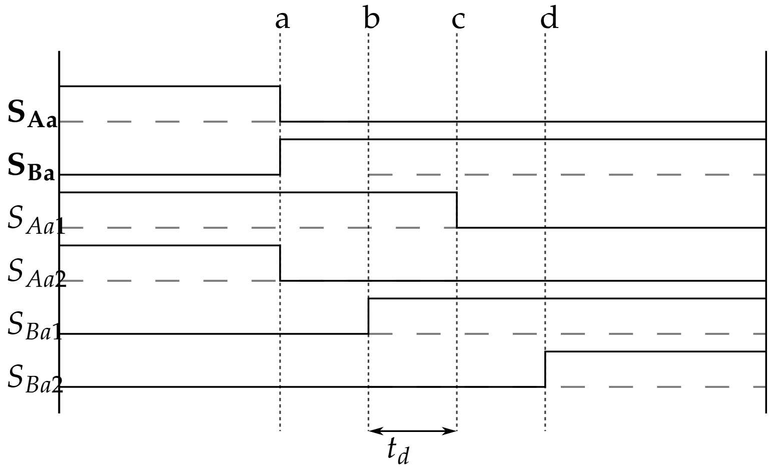
This is shown in Figure 4.3, and as can be seen means that when switching, unlike with Overlap commutation the input phases are never shorted together, however the output legs are now allowed to go open circuit, breaking the second of the rules for matrix converter operation as specified above. As before, there is a way to mitigate the problems associated with this method, and that is to use snubber circuits around each switch to allow the current to continue flowing in the output leg while the switching is occurring. However, as for the Overlap commutation before, one of the advantages of the matrix converter are that they are compact and all silicon, and this would be compromised by adding high capacity snubber networks using capacitors, and so this method is not used.

4.3 4-Step Commutation

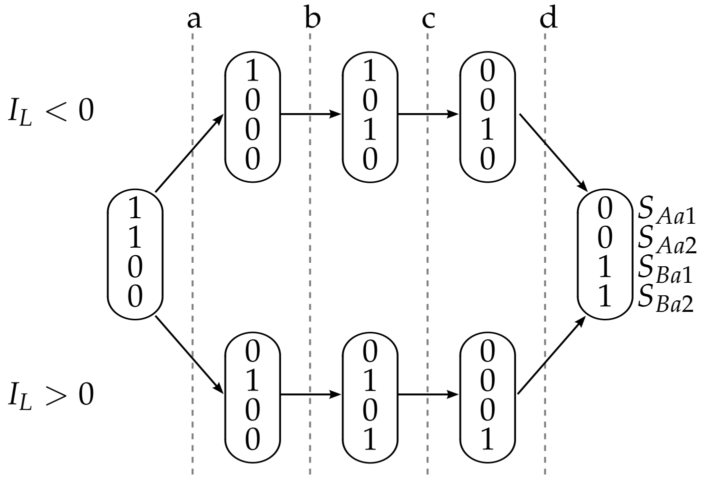
The 4-Step commutation process was one of the first methods proposed that allowed the matrix converter to be successfully switched between states without violating either of the two rules defined above for their operation. However, for this approach to work it requires a bi-directional switch which can independently control the conduction path in each direction, this is the example that is shown in Figure 4.1 above. As its name suggests, unlike the previous two methods this method uses four individual steps and this sequence is demonstrated in Figure 4.4

The first step is for the non-conducting device in the outgoing switch to be turned off(a). Then in the incoming switch, the device which will be conducting is switched on(b). This can happen without creating a short circuit between the two input phases because both switches are only being turned on in one direction only, and both in the same direction as the current flow, so the short circuit is not possible due to the series diodes in the bi-directional switch. At this point it is then possible to turn off the conducting device in the outgoing switch(c), turning off that bi-directional switch entirely. The final step, once the outgoing device is completely turned off, is then for the non-conducting device in the incoming switch to be turned on(d).

Unlike the 2-step commutation processes above, which only need to know which switches are involved, this technique also needs to know the current direction within the switch. This is important as it is the current direction within the bi-directional switch that defines which order the devices are switched, and this is simple to achieve at higher currents using hall effect sensors on the output legs. However there is a problem using the hall effect sensors at very low currents, but at these current levels, while not ideal, it would be possible to switch to a dead-time



(a) Four-Step Current Commutation Timing Diagram



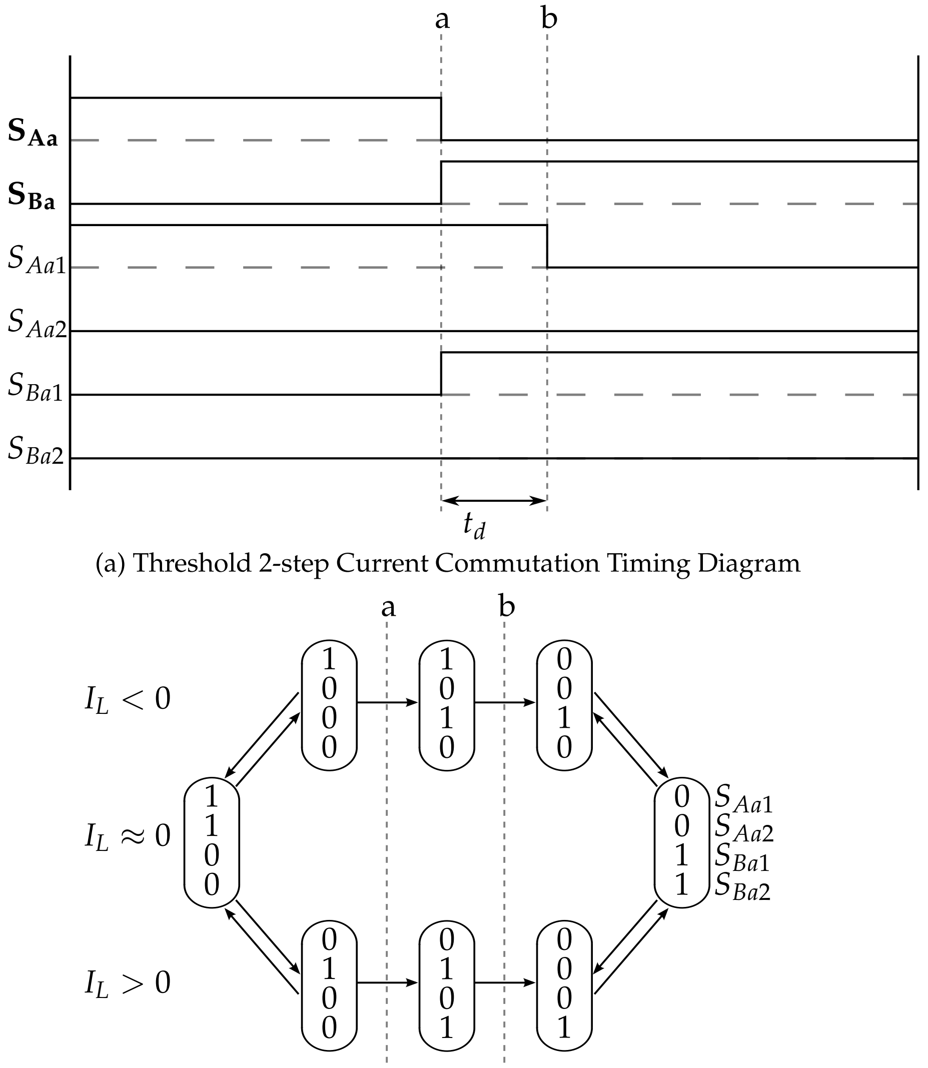
(b) Four-Step Current Commutation State Diagram

Figure 4.4: Four-Step Current Commutation

commutation arrangement as long as an adequate voltage clamp circuit was placed on the outputs. This circuit would have to be far smaller than that needed if performing dead-time commutation all the time due to this only being used at low levels of current.

4.4 Threshold 2-Step Current Commutation

As can be seen in the four-step commutation process above, at the required switching instant the first and last steps do not really appear to achieve much as neither of the devices are conducting, and they just slow down the overall switching process as the incoming device only gets turned on t_d after the switching process has started. To allow for this the threshold 2-step commutation process was derived. It is essentially as the four-step process except that it only ever turns on the conducting elements in the switch, keeping the non-conducting elements turned off until the current direction changes. This means that to commutate between input phases the method only needs to turn on the incoming switches conducting element followed by turning off the outgoing switch and the method is shown in Figure 4.5.



(a) Threshold 2-step Current Commutation Timing Diagram

(b) Threshold 2-step Current Commutation State Diagram

Figure 4.5: Threshold 2-Step Current Commutation

While this appears to have an advantage over the 4-step method in its simplicity with only having to turn one device on and turn off another, just as in the overlap method, it does mean however that the gate drive circuits themselves need to be aware at all times what the current direction is, and be able to turn on the other device if the current direction changes. As with the four-step process, the current detection is simple at high currents using hall effect sensors, but once again there is a problem at lower currents.

With the inability to accurately detect low current levels, this leads to two problems. The first is the same as that seen in the four-step technique, in that for commutation to occur correctly, the current direction needs to be known. As with the four-step commutation, this can be overcome by using dead-time commutation for the range of currents where the hall effect sensor is inaccurate, as shown in Figure 4.6, along with placing a voltage clamp circuit on the output.

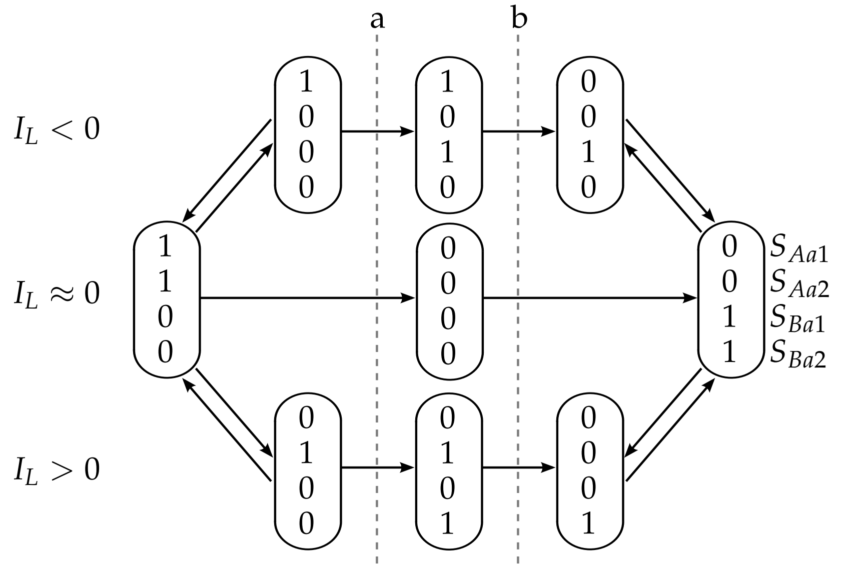


Figure 4.6: Threshold 2-Step with Low Current Deadtime Commutation

The second problem is that outside of any commutation between input phases, the gate drive needs to ensure that the correct device is turned on within the bi-directional switch so that normal current can flow. This problem is overcome by setting a threshold level, below which the gate drive turns on both devices within the switch, thus ensuring that the switch is able to conduct no matter which direction the current flow is in. Once the current level has risen once again to a level where the hall effect sensors are accurate, then the gate drive circuit will switch off the non-conducting device within the switch.

With these extra requirements over that of the four-step commutation process, it means that the gate drive circuit needs to be more complex.

4.5 Voltage Sensed Current Commutation

As a solution to the current detection problem with the four-step, and its derived 2-step, technique, a method was developed at Nottingham University[33] that allowed the detection of the input current direction even at very low levels. This was achieved by measuring, and comparing, the voltage drop across different parts of the switch, as shown in Figure 4.7.

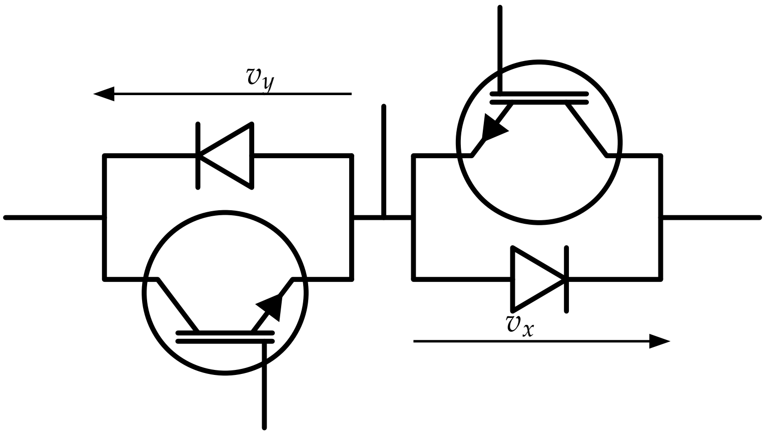
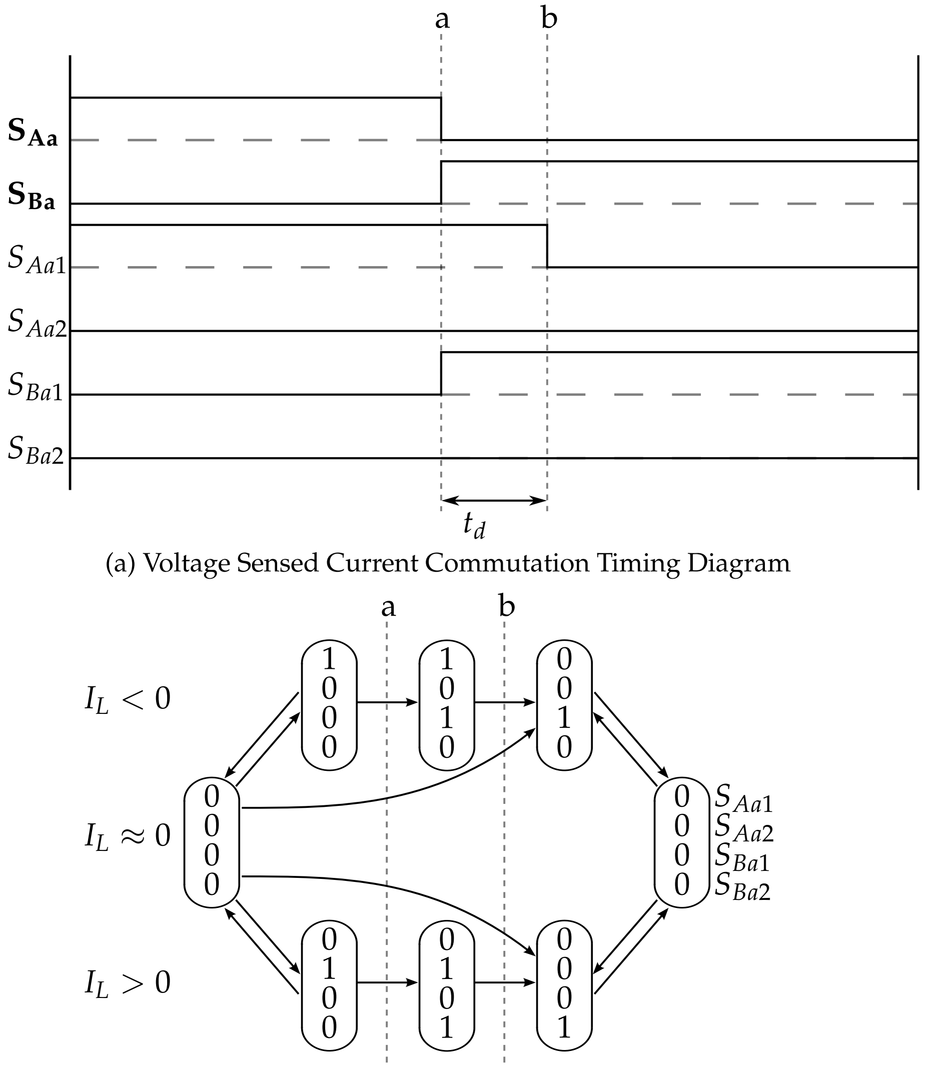


Figure 4.7: Current Direction Detection using Voltage drops in a bi-directional Switch

If the switch is conducting in one direction, where the current is flowing from the left to the right, then the voltage drop across the conducting IGBT, v_y, will be positive with respect to the mid-point and the voltage across the conducting diode, v_x will be negative with respect to the mid-point. If the current direction changes, then so will the polarity of the voltages and this is easily detected by a comparator circuit and this information used for the commutation. By using this technique it is possible to accurately detect the current direction down to very small currents.

Because of the highly accurate nature of the current direction measurement, this allowed the derivation of this commutation method from the 2-step process above, in which there is only ever a maximum of a single device turned on within a bi-directional switch at one time, as shown in Figure4.8



(a) Voltage Sensed Current Commutation Timing Diagram

(b) Voltage Sensed Commutation State Diagram

Figure 4.8: Voltage Sensed Current Commutation

CHAPTER 5

Derivation of the Space Vector Modulation technique for the 4-leg Matrix Converter

As has been shown previously, all Space Vector Modulation SVM techniques are based around the idea that you can represent the input and output voltages of a converter by a set of vectors within specific input and output coordinate systems. By then defining the relationship between these two sets of vectors, this is used to define a set of switching states for the converter that will allow both the output voltage and input current requirements to be satisfied. At any one point in time these vectors are defined completely by the instantaneous voltages at both the inputs and outputs of the converter.

It has previously been shown in Sections 2.1 and 3.4 how this has been achieved for both the 4-leg inverter and the 3x3 matrix converter respectively, and the derivation of the technique for the 4-leg matrix converter draws from both of these techniques and so develops along similar lines.

The technique will Initially define the input voltage, followed by the 3-dimensional output voltage space. From this it will lead on to the specification of the different switching states, which in turn allows the definition of the generated space vectors. Once these space vectors have been defined, it will then be possible to show the link between the input and output spaces, which will show how the converter will be able to operate. The derivation of the duty cycles for each of the required switching states will then be performed, before finally looking at the possible voltage transfer ratio for this modulation strategy.

5.1 The Input Voltage Plane

Looking at the diagram in Figure 1.6, it shows the basic layout of the 4-leg matrix converter. From this it can be seen that while there are 4 output legs to the converter, there are only 3 input legs, so while the output is capable of generating an unbalanced set of phase voltages, the input must form a balanced three phase set as defined by

$$V_A + V_B + V_C = 0 \tag{5.1.1}$$

where each of the input phase voltages is defined by

$$V_A = V_i \cos\left(\alpha_i\right) \tag{5.1.2}$$
$$V_B = V_i \cos\left(\alpha_i - \frac{2\pi}{3}\right) \tag{5.1.3}$$
$$V_C = V_i \cos\left(\alpha_i - \frac{4\pi}{3}\right) \tag{5.1.4}$$

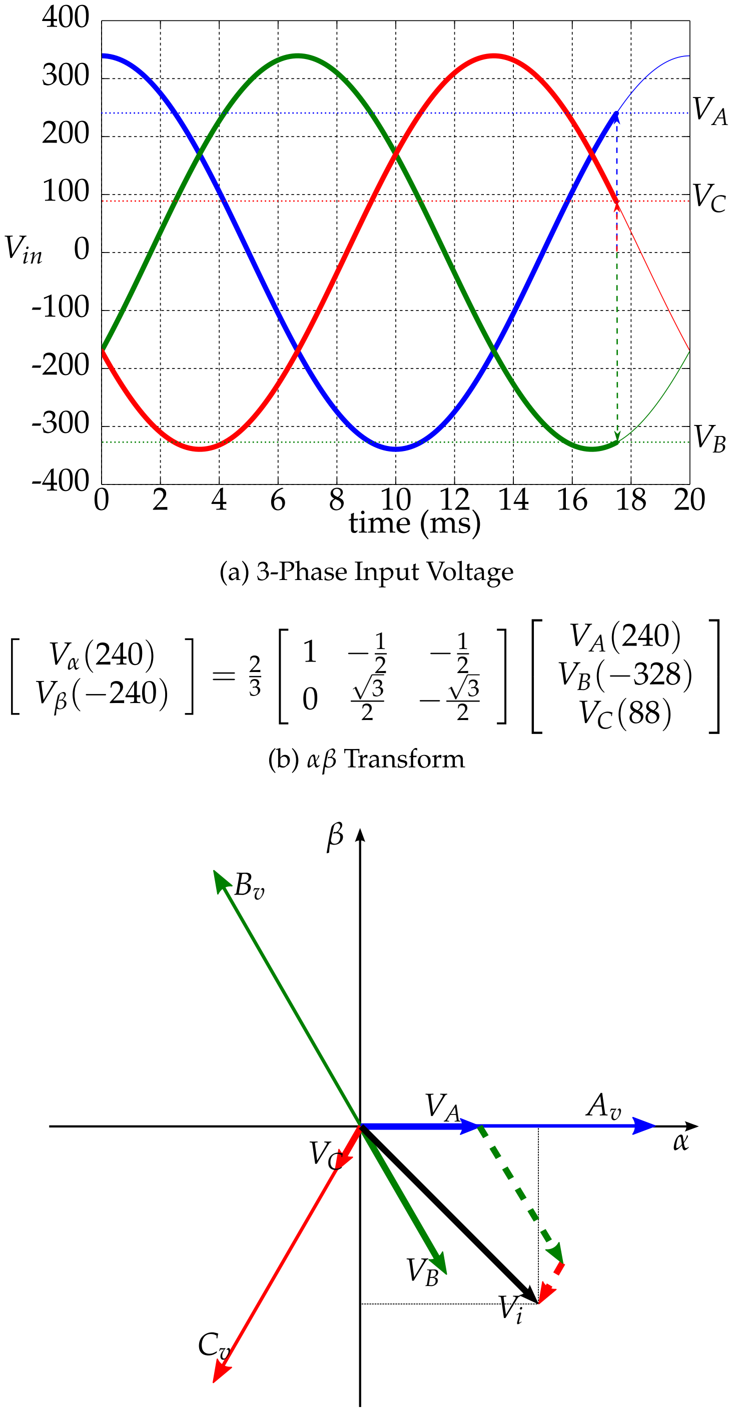
where V_i is the phase input voltage, and α_i is the phase of the input voltage at a particular instant in time. An example set of input voltages are also shown in Figure 5.1(a).

In the same way as for the 3x3 matrix converter, this set of input voltages can be represented as a space vector by using the following well known $\alpha\beta$-transform

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} \tag{5.1.5}$$

This equation (5.1.5) takes the three input phase voltages and calculates the values V_α and V_β, which can be plotted as a vector $\vec{v}_i$ in the $\alpha\beta$ plane. Figure 5.1(c) shows an example where $\vec{v}_i$ has been plotted for the input voltages shown in Figure 5.1(b).

As the input is a balanced 3-phase set this vector will be of a constant amplitude, $|\vec{v}_i|$, and rotate around the origin at the input frequency, ω_i. Using equations (5.1.1) and (5.1.5) it is possible to write down three primary voltage space vectors which relate to the three input phase voltages, and there are labelled as $\mathbf{A_v}, \mathbf{B_v}, \mathbf{C_v}$, such that



(a) 3-Phase Input Voltage

$$\begin{bmatrix} V_\alpha(240) \\ V_\beta(-240) \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_A(240) \\ V_B(-328) \\ V_C(88) \end{bmatrix}$$

(b) $\alpha\beta$ Transform

(c) $\alpha\beta$ Input Space showing the example from (b)

Figure 5.1: Input $\alpha\beta$ space showing the 3 primary input voltage vectors

$$\mathbf{A_v} = \tfrac{2}{3}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{B_v} = \tfrac{2}{3}\begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \quad \mathbf{C_v} = \tfrac{2}{3}\begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{5.1.6}$$

These three vectors can be seen plotted in Figure 5.1(c), and are the three vectors spaced evenly 120° about the origin.

5.2 The Input Current Plane

In exactly the same way as for the input voltage plane shown above, the input currents, I_A, I_B and I_C, can also be transformed into the $\alpha\beta$ space using the same transform as the input voltage, so

$$\begin{bmatrix} I_\alpha \\ I_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} I_A \\ I_B \\ I_C \end{bmatrix} \tag{5.2.1}$$

As for the voltages above, the values I_α and I_β can be plotted to produce an input current space vector, $\vec{i_i}$, in the $\alpha\beta$ space whose magnitude, $|\vec{i_i}|$, and phase, α_i, are determined by the operation of the converter and by the load. However, the converter is being designed to achieve a unity input displacement factor which means that the fundamental frequency of the input current should be in phase with the fundamental frequency of the input voltage. In this case the phase angle of the input current space vector will always match that of the input voltage vector, and so will rotate around the origin at the input voltage frequency, α_i.

Therefore, as the input voltage vector was formed by a balanced 3-phase set, and the input current vector rotates in phase with the resultant voltage vector, then then at any one instant the input currents must also form a balanced 3-phase set such that

$$I_A + I_B + I_C = 0 \tag{5.2.2}$$

From equation (5.2.2) and using equation (5.2.1), the three primary current space vectors, $\mathbf{A_i}, \mathbf{B_i}, \mathbf{C_i}$ relating to the current in each input phase can be written down as

$$\mathbf{A_i} = \tfrac{2}{3}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{B_i} = \tfrac{2}{3}\begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \quad \mathbf{C_i} = \tfrac{2}{3}\begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{5.2.3}$$

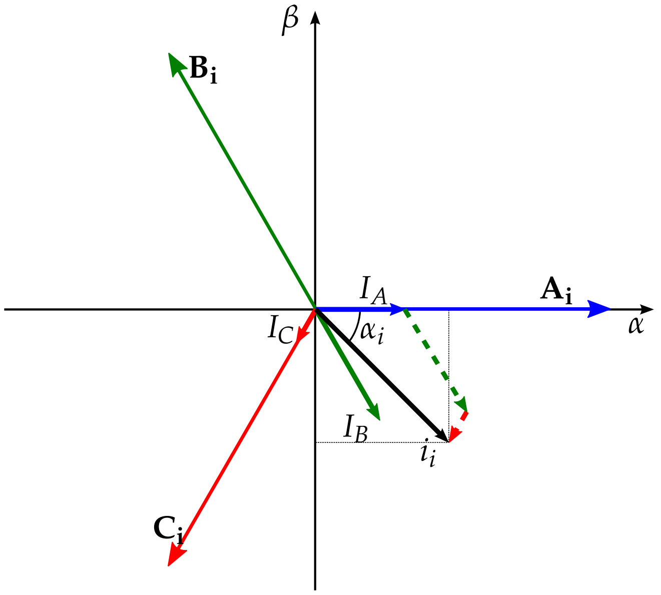


Figure 5.2: Input $\alpha\beta$ space showing the 3 primary input current vectors

Figure 5.2 shows the $\alpha\beta$ input current plane, with these three primary current vectors plotted, these are the three vectors spaced evenly 120° about the origin, and are the same as for the input voltage. Also shown is the direction of the desired input current vector $\vec{i_i}$ for the example shown in Figure 5.1.

5.3 The Output Voltage Space

The 3x3 matrix converter produces 3 different output voltages, but because these voltages are all referenced to each another, and form a balanced set, they are constrained by the following equation

$$V_a + V_b + V_c = 0 \tag{5.3.1}$$

So while there are three different variables, V_a, V_b and V_c, they are all related to one another and so this actually only gives 2 degrees of freedom. This means that it is only possible to independently set 2 out of 3 of the output voltages, with the third being set by the solution to equation 5.3.1.

With the 4-leg matrix converter, which is able to produce 4 different output voltages, once again all the voltages are referenced to one another, but are now constrained by

$$V_a + V_b + V_c + V_n = 0 \tag{5.3.2}$$

Because there are now four different variables, V_a, V_b, V_c and V_n, in the same way as before, this now gives 3 degrees of freedom. So this means it is possible to independently set each of the 3 phase voltages, V_a, V_b and V_c, while the fourth, V_n, will be set by the solution to equation (5.3.2).

So, unlike with the 3x3 matrix converter, the 4-leg matrix converter is able to independently control each of the 3 phase outputs with respect to the voltage generated by the neutral leg, and so allow an unbalanced set of 3-phase voltages to be demanded. This means that the three output phase voltages can have any value, and are not constrained by equation (5.3.1), so

$$V_a + V_b + V_c \neq 0 \tag{5.3.3}$$

Due to this extra degree of freedom required by equation (5.3.2), the two dimensional $\alpha\beta$ space used for the input voltage is just not able to define a unique space vector resulting from all 4 output legs and so a three dimensional output space is required. This is defined as the $\alpha\beta\gamma$ space[12], and is the same as that described for the 4-leg inverter in Section 2.1. The output voltage space vector is then generated using

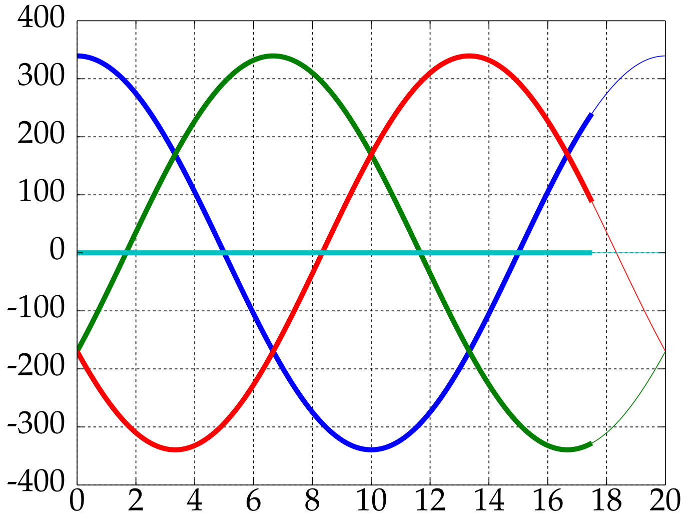
$$\begin{bmatrix} X_\alpha \\ X_\beta \\ X_\gamma \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{-3}{2\sqrt{2}} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \\ V_n \end{bmatrix} \tag{5.3.4}$$

This transform takes the required instantaneous output voltages, V_a V_b, V_c and V_n, and generates a single space vector that can be plotted in the $\alpha\beta\gamma$ space as shown in Figure 5.3(c). An example transformation is shown in Figure 5.3(b).

In a similar way to the input voltage space above, if the four output space vectors relating to the four output legs are labelled $\vec{\mathbf{a}}, \vec{\mathbf{b}}, \vec{\mathbf{c}}, \vec{\mathbf{n}}$, then using equation (5.3.2) the following equation can be written down

$$\vec{\mathbf{a}} + \vec{\mathbf{b}} + \vec{\mathbf{c}} + \vec{\mathbf{n}} = 0 \tag{5.3.5}$$

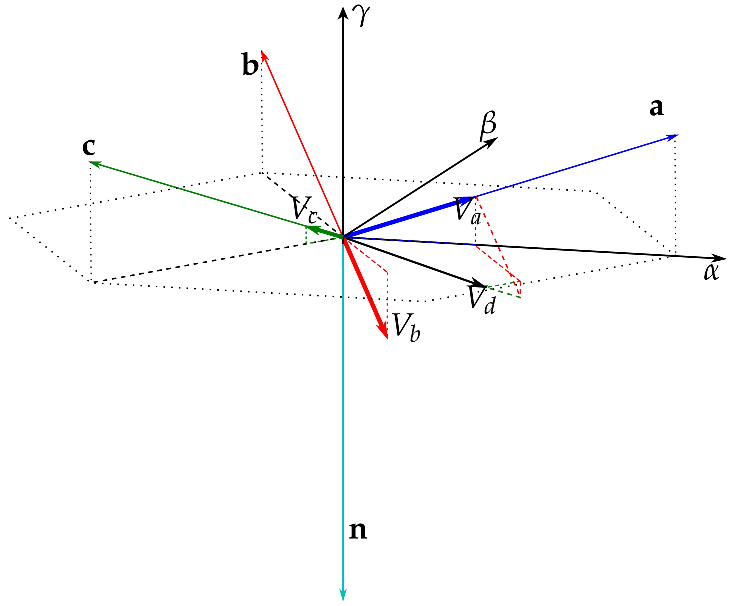
Using the $\alpha\beta\gamma$ transform in equation (5.3.4) the four space vectors repre-



(a) 4-Phase Output Voltage

$$\begin{bmatrix} X_\alpha(120) \\ X_\beta(-120) \\ X_\gamma(0) \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{3}{2\sqrt{2}} \end{bmatrix} \begin{bmatrix} V_a(120) \\ V_b(-164) \\ V_c(44) \\ V_n(0) \end{bmatrix}$$

(b) $\alpha\beta\gamma$ Transform



(c) $\alpha\beta\gamma$ Output Space showing the example from (b)

Figure 5.3: Output $\alpha\beta\gamma$ space showing the 4 primary output voltage vectors

senting the output legs are

$$\mathbf{a} = \frac{2}{3}\begin{bmatrix} 1 \\ 0 \\ \frac{1}{2\sqrt{2}} \end{bmatrix} \quad \mathbf{b} = \frac{2}{3}\begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ \frac{1}{2\sqrt{2}} \end{bmatrix} \quad \mathbf{c} = \frac{2}{3}\begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \\ \frac{1}{2\sqrt{2}} \end{bmatrix} \quad \mathbf{n} = \frac{2}{3}\begin{bmatrix} 0 \\ 0 \\ -\frac{3}{2\sqrt{2}} \end{bmatrix} \tag{5.3.6}$$

These are the four primary output vectors, and are also plotted on Figure 5.3(c) where it can be seen that, in a similar way to the input voltage space vectors before, these are all equidistant from one another within the $\alpha\beta\gamma$ space.

It should be noted that between the two papers which were used as the source for the 3-dimensional space vector modulation that is used in this derivation, there are a couple of differences with this transform matrix. The first being that Ryan, Lorenz and De Doncker[12] use the T_{dqo} notation more commonly used for a rotating frame of reference, while Zhang[14] uses the static $\alpha\beta\gamma$ notation. The $\alpha\beta\gamma$ notation which was chosen for this derivation as the vector space does not move, and the static frame of reference was thought to be preferable.

The second difference being the values used for the γ row in the transform matrix, with Ryan, Lorenz and De Doncker[12] using $1/(2\sqrt{2})$ and Zhang[14] using $1/2$. The value used in this row determines the range of possible values that γ can take, and for this derivation of the 4-leg matrix converter, the value of $1/(2\sqrt{2})$ was chosen because this means that the 4 primary vectors, $\vec{\mathbf{a}}$, $\vec{\mathbf{b}}$, $\vec{\mathbf{c}}$ and $\vec{\mathbf{n}}$, are all the same length. This is not the case with the values used by Zhang. While this is not a problem with the forms of the solution proposed by Zhang for the 4-leg inverter, it was found that having the primary vectors of the same length was advantageous for the 4-leg matrix converter.

5.4 The Switching States

Now that the input and output spaces have been initially set out, and the primary space vectors in each area defined, the next stage in the derivation of the 4-leg matrix converter is to ascertain the complete set of switching states that are possible. These switching states are those which obey the fundamental rules for any matrix converter, that two input phases cannot be shorted together, and no output leg can be open circuit. For the standard 3x3 matrix converter it has been shown that there are 27 (3^3) possible switching states[4]. However, the addition of the fourth output leg increases the total number of combinations for the 3x4 matrix converter to 81 (3^4). All of these switching states are shown in Tables 5.1 and 5.2.

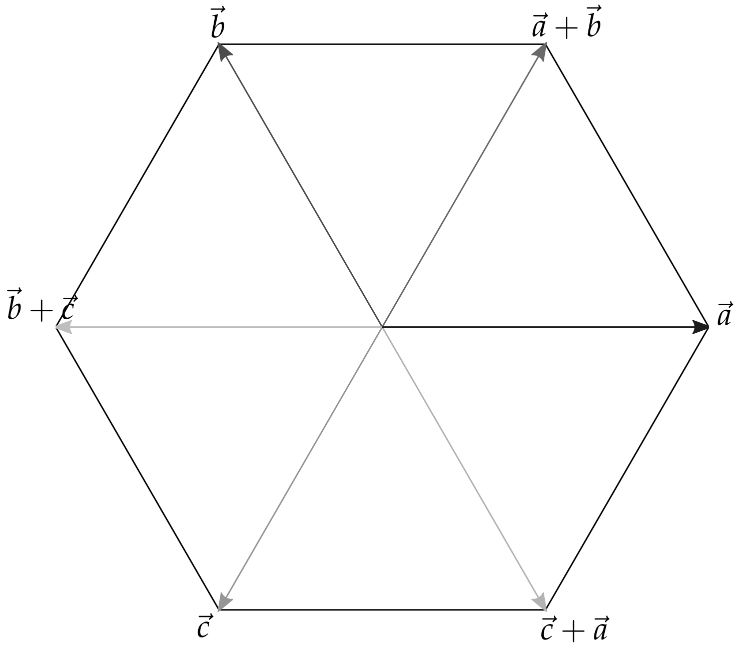


Figure 5.4: 3x3 Matrix Converter stationary vectors

As with the 3x3 converter [4], not all of these possible switching states are useful for Space Vector Modulation, and so only those that produce a vector with a constant direction within the output space are to be used, and these are called the stationary vectors. The switching states of any 3-input matrix converter can be broken down into three distinct groups

- Group I : Those that connect all three input phases to the output. These switching states, which are all are shown in Table 5.1, are those states that connect to all three of the input phases. As all of the output legs have to be switched to an input phase this means that two different line-to-line voltages must be present on the output legs. These line-to-line voltages that are generated change independently of each other meaning that the resultant vector created at the output legs is not only changing in magnitude, but will also changing direction as well. These are the set of rotating vectors which, as with the standard 3x3 matrix converter space vector modulation, are not used within this method.

- Group II : Those that connect any two input phases to the output.

Table 5.1: Rotating Switching States for the 4-Leg Matrix Converter

	Output Leg Switching				V_{vector}	L-N Output Voltage			I_{vector}	Input Current		
	Leg a	Leg b	Leg c	Leg n		V_{an}	V_{bn}	V_{cn}		I_A	I_B	I_C
Rotating Vectors	A	A	B	C		$-V_{CA}$	$-V_{CA}$	V_{BC}		I_a+I_b	I_c	$-I_a-I_b-I_c$
	A	B	A	C		$-V_{CA}$	V_{BC}	$-V_{CA}$		I_a+I_c	I_b	$-I_a-I_b-I_c$
	B	A	A	C		V_{BC}	$-V_{CA}$	$-V_{CA}$		I_b+I_c	I_a	$-I_a-I_b-I_c$
	B	B	A	C		V_{BC}	V_{BC}	$-V_{CA}$		I_c	I_a+I_b	$-I_a-I_b-I_c$
	B	A	B	C		V_{BC}	$-V_{CA}$	V_{BC}		I_b	I_a+I_c	$-I_a-I_b-I_c$
	A	B	B	C		$-V_{CA}$	V_{BC}	V_{BC}		I_a	I_b+I_c	$-I_a-I_b-I_c$
	A	A	C	B		V_{AB}	V_{AB}	$-V_{BC}$		I_a+I_b	$-I_a-I_b-I_c$	I_c
	A	C	A	B		V_{AB}	$-V_{BC}$	V_{AB}		I_a+I_c	$-I_a-I_b-I_c$	I_b
	C	A	A	B		$-V_{BC}$	V_{AB}	V_{AB}		I_b+I_c	$-I_a-I_b-I_c$	I_a
	C	C	A	B		$-V_{BC}$	$-V_{BC}$	V_{AB}		I_c	$-I_a-I_b-I_c$	I_a+I_b
	C	A	C	B		$-V_{BC}$	V_{AB}	$-V_{BC}$		I_b	$-I_a-I_b-I_c$	I_a+I_c
	A	C	C	B		V_{AB}	$-V_{BC}$	$-V_{BC}$		I_a	$-I_a-I_b-I_c$	I_b+I_c
	B	B	C	A		$-V_{AB}$	$-V_{AB}$	V_{CA}		$-I_a-I_b-I_c$	I_a+I_b	I_c
	B	C	B	A		$-V_{AB}$	V_{CA}	$-V_{AB}$		$-I_a-I_b-I_c$	I_a+I_c	I_b
	C	B	B	A		V_{CA}	$-V_{AB}$	$-V_{AB}$		$-I_a-I_b-I_c$	I_b+I_c	I_a
	C	C	B	A		V_{CA}	V_{CA}	$-V_{AB}$		$-I_a-I_b-I_c$	I_c	I_a+I_b
	C	B	C	A		V_{CA}	$-V_{AB}$	V_{CA}		$-I_a-I_b-I_c$	I_b	I_a+I_c
	B	C	C	A		$-V_{AB}$	V_{CA}	V_{CA}		$-I_a-I_b-I_c$	I_a	I_b+I_c
	A	B	C	A		0	$-V_{AB}$	V_{CA}		$-I_b-I_c$	I_b	I_c
	B	A	C	A		$-V_{AB}$	0	V_{CA}		$-I_a-I_c$	I_a	I_c
	B	C	A	A		$-V_{AB}$	V_{CA}	0		$-I_a-I_b$	I_a	I_b
	C	B	A	A		V_{CA}	$-V_{AB}$	0		$-I_a-I_b$	I_b	I_a
	C	A	B	A		V_{CA}	0	$-V_{AB}$		$-I_a-I_c$	I_c	I_a
	A	C	B	A		0	V_{CA}	$-V_{AB}$		$-I_b-I_c$	I_c	I_b
	A	B	C	B		V_{AB}	0	$-V_{BC}$		I_a	$-I_a-I_c$	I_c
	B	A	C	B		0	V_{AB}	$-V_{BC}$		I_b	$-I_b-I_c$	I_c
	B	C	A	B		0	$-V_{BC}$	V_{AB}		I_c	$-I_b-I_c$	I_b
	C	B	A	B		$-V_{BC}$	0	V_{AB}		I_c	$-I_a-I_c$	I_a
	C	A	B	B		$-V_{BC}$	V_{AB}	0		I_b	$-I_a-I_b$	I_a
	A	C	B	B		V_{AB}	$-V_{BC}$	0		I_a	$-I_a-I_b$	I_b
	A	B	C	C		$-V_{CA}$	V_{BC}	0		I_a	I_b	$-I_a-I_b$
	B	A	C	C		V_{BC}	$-V_{CA}$	0		I_b	I_a	$-I_a-I_b$
	B	C	A	C		V_{BC}	0	$-V_{CA}$		I_c	I_a	$-I_a-I_c$
	C	B	A	C		0	V_{BC}	$-V_{CA}$		I_c	I_b	$-I_b-I_c$
	C	A	B	C		0	$-V_{CA}$	V_{BC}		I_b	I_c	$-I_b-I_c$
	A	C	B	C		$-V_{CA}$	0	V_{BC}		I_a	I_c	$-I_a-I_c$

These states, which connect any two of the input phases to the outputs are the main ones that are of interest as these are the ones that generate the stationary vectors. With only two of the input phases are connected through to the output legs at any one time, and with all the output legs needing to be connected to an input phase, the output legs must be connected to one or other of these two phases. From this it can be seen that the only output voltages that can be generated on the outputs are either zero, or the line-to-line voltage between the two connected phases. Even though this line-to-line voltage can be generated on more than one output leg, because all of the voltages are the same, the resultant vector, even if changing in magnitude, will always have a fixed angle within the output space. This is true no matter which input phases are switched. This was also true for the 3x3 matrix converter, where this generates the set of 6 vectors evenly spaced around the zero point, giving the well known hexagonal plot. In this case, for the 4-leg matrix converter, this generates a set of constant vectors within a three dimensional $\alpha\beta\gamma$ output space as shown in Figure 5.5. These vectors can be seen in Table 5.2 marked as the Stationary Vectors.

- Group III : Those that connect a single input phase to the outputs. These are the states which connect to all four output legs to a single input phase can only ever produce a zero vector at the outputs. With only a single input phase connected through to the output legs, and once again, with all the output legs needing to be connected to an input phase, the output legs can only be connected to that single phase. As they are all now connected to the same input phase, the voltage between them will therefore be zero. These vectors are important as they allow a zero state to be applied to the output while also maintaining the validity of equation ref4 phase balanced output voltage sum, and without using these states it would not be possible to otherwise generate a zero state without disconnecting the output legs from the input phases. These vectors can be seen at the top of Table 5.2 marked as the Zero Vectors.

Table 5.2: Stationary and Zero Switching States for the 4-Leg Matrix Converter

	Switching	Output Leg Switching				V_{vector}	L-N Output Voltage			I_{vector}	Input Current		
	State	Leg a	Leg b	Leg c	Leg n		V_{an}	V_{bn}	V_{cn}		I_A	I_B	I_C
Zero Vectors	0_A	A	A	A	A	ZA	0	0	0		0	0	0
	0_B	B	B	B	B	ZB	0	0	0		0	0	0
	0_C	C	C	C	C	ZC	0	0	0		0	0	0
Stationary Vectors	+1	A	B	B	B	V8	V_{AB}	0	0	I6	I_a	$-I_a$	0
	-1	B	A	A	A	V7	$-V_{AB}$	0	0	I3	$-I_a$	I_a	0
	+2	B	C	C	C	V8	V_{BC}	0	0	I2	0	I_a	$-I_a$
	-2	C	B	B	B	V7	$-V_{BC}$	0	0	I5	0	$-I_a$	I_a
	+3	C	A	A	A	V8	V_{CA}	0	0	I4	$-I_a$	0	I_a
	-3	A	C	C	C	V7	$-V_{CA}$	0	0	I1	I_a	0	$-I_a$
	+4	B	A	B	B	V4	0	V_{AB}	0	I6	I_b	$-I_b$	0
	-4	A	B	A	A	V11	0	$-V_{AB}$	0	I3	$-I_b$	I_b	0
	+5	C	B	C	C	V4	0	V_{BC}	0	I2	0	I_b	$-I_b$
	-5	B	C	B	B	V11	0	$-V_{BC}$	0	I5	0	$-I_b$	I_b
	+6	A	C	A	A	V4	0	V_{CA}	0	I4	$-I_b$	0	I_b
	-6	C	A	C	C	V11	0	$-V_{CA}$	0	I1	I_b	0	$-I_b$
	+7	B	B	A	B	V2	0	0	V_{AB}	I6	I_c	$-I_c$	0
	-7	A	A	B	A	V13	0	0	$-V_{AB}$	I3	$-I_c$	I_c	0
	+8	C	C	B	C	V2	0	0	V_{BC}	I2	0	I_c	$-I_c$
	-8	B	B	C	B	V13	0	0	$-V_{BC}$	I5	0	$-I_c$	I_c
	+9	A	A	C	A	V2	0	0	V_{CA}	I4	$-I_c$	0	I_c
	-9	C	C	A	C	V13	0	0	$-V_{CA}$	I1	I_c	0	$-I_c$
	+10	A	A	B	B	V12	V_{AB}	V_{AB}	0	I6	I_a+I_b	$-I_a-I_b$	0
	-10	B	B	A	A	V3	$-V_{AB}$	$-V_{AB}$	0	I3	$-I_a-I_b$	I_a+I_b	0
	+11	B	B	C	C	V12	V_{BC}	V_{BC}	0	I2	0	I_a+I_b	$-I_a-I_b$
	-11	C	C	B	B	V3	$-V_{BC}$	$-V_{BC}$	0	I5	0	$-I_a-I_b$	I_a+I_b
	+12	C	C	A	A	V12	V_{CA}	V_{CA}	0	I4	$-I_a-I_b$	0	I_a+I_b
	-12	A	A	C	C	V3	$-V_{CA}$	$-V_{CA}$	0	I1	I_a+I_b	0	$-I_a-I_b$
	+13	B	A	A	B	V6	0	V_{AB}	V_{AB}	I6	I_b+I_c	$-I_b-I_c$	0
	-13	A	B	B	A	V9	0	$-V_{AB}$	$-V_{AB}$	I3	$-I_b-I_c$	I_b+I_c	0
	+14	C	B	B	C	V6	0	V_{BC}	V_{BC}	I2	0	I_b+I_c	$-I_b-I_c$
	-14	B	C	C	B	V9	0	$-V_{BC}$	$-V_{BC}$	I5	0	$-I_b-I_c$	I_b+I_c
	+15	A	C	C	A	V6	0	V_{CA}	V_{CA}	I4	$-I_b-I_c$	0	I_b+I_c
	-15	C	A	A	C	V9	0	$-V_{CA}$	$-V_{CA}$	I1	I_b+I_c	0	$-I_b-I_c$
	+16	A	B	A	B	V10	V_{AB}	0	V_{AB}	I6	I_a+I_c	$-I_a-I_c$	0
	-16	B	A	B	A	V5	$-V_{AB}$	0	$-V_{AB}$	I3	$-I_a-I_c$	I_a+I_c	0
	+17	B	C	B	C	V10	V_{BC}	0	V_{BC}	I2	0	I_a+I_c	$-I_a-I_c$
	-17	C	B	C	B	V5	$-V_{BC}$	0	$-V_{BC}$	I5	0	$-I_a-I_c$	I_a+I_c
	+18	C	A	C	A	V10	V_{CA}	0	V_{CA}	I4	$-I_a-I_c$	0	I_a+I_c
	-18	A	C	A	C	V5	$-V_{CA}$	0	$-V_{CA}$	I1	I_a+I_c	0	$-I_a-I_c$
	+19	A	A	A	B	V14	V_{AB}	V_{AB}	V_{AB}	I6	$I_a+I_b+I_c$	$-I_a-I_b-I_c$	0
	-19	B	B	B	A	V1	$-V_{AB}$	$-V_{AB}$	$-V_{AB}$	I3	$-I_a-I_b-I_c$	$I_a+I_b+I_c$	0
	+20	B	B	B	C	V14	V_{BC}	V_{BC}	V_{BC}	I2	0	$I_a+I_b+I_c$	$-I_a-I_b-I_c$
	-20	C	C	C	B	V1	$-V_{BC}$	$-V_{BC}$	$-V_{BC}$	I5	0	$-I_a-I_b-I_c$	$I_a+I_b+I_c$
	+21	C	C	C	A	V14	V_{CA}	V_{CA}	V_{CA}	I4	$-I_a-I_b-I_c$	0	$I_a+I_b+I_c$
	-21	A	A	A	C	V1	$-V_{CA}$	$-V_{CA}$	$-V_{CA}$	I1	$I_a+I_b+I_c$	0	$-I_a-I_b-I_c$

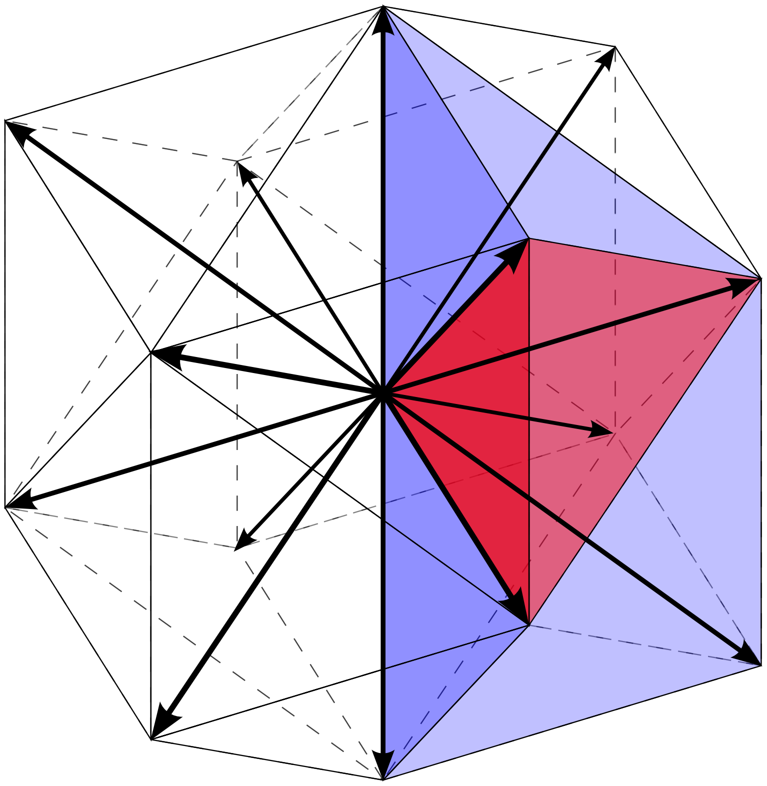


Figure 5.5: Stationary vectors plotted in $\alpha\beta\gamma$ space

5.4.1 Output Voltage and Switching States

Now, each of the switching states within Table 5.2 defines a particular space vector within the $\alpha\beta\gamma$ space, and are shown plotted in Figure 5.5. So, if switching state $+\mathbf{1}$ from Table 5.2 is examined, it can be seen that input phase **A** is switched to output leg **a**, while input phase **B** is switched to the remaining output legs **bcn**. Looking at the output voltages it is easy to see that this gives

$$\begin{aligned} V_a &= V_A \\ V_b &= V_B \\ V_c &= V_B \\ V_n &= V_B \end{aligned} \tag{5.4.1}$$

Transforming this into the $\alpha\beta\gamma$ space using (5.3.4) gives

$$\vec{V}_o = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{-3}{2\sqrt{2}} \end{bmatrix}\begin{bmatrix} V_A \\ V_B \\ V_B \\ V_B \end{bmatrix} = \frac{2}{3}V_{AB}\begin{bmatrix} 1 \\ 0 \\ \frac{1}{2\sqrt{2}} \end{bmatrix} \tag{5.4.2}$$

This same result can also be achieved using the primary voltage vectors as set out in equation (5.3.6), where $\vec{\mathbf{a}}$ is

$$\vec{\mathbf{a}} = \frac{2}{3}V_A\begin{bmatrix} 1 \\ 0 \\ \frac{1}{2\sqrt{2}} \end{bmatrix} \tag{5.4.3}$$

and then $\vec{\mathbf{b}} + \vec{\mathbf{c}} + \vec{\mathbf{n}}$ gives

$$\vec{\mathbf{b}} + \vec{\mathbf{c}} + \vec{\mathbf{n}} = \tfrac{2}{3}V_B\begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \\ \frac{1}{2\sqrt{2}} \end{bmatrix} + \tfrac{2}{3}V_B\begin{bmatrix} -\frac{1}{2} \\ \frac{-\sqrt{3}}{2} \\ \frac{1}{2\sqrt{2}} \end{bmatrix} + \tfrac{2}{3}V_B\begin{bmatrix} 0 \\ 0 \\ \frac{-3}{2\sqrt{2}} \end{bmatrix} = \tfrac{2}{3}V_B\begin{bmatrix} -1 \\ 0 \\ \frac{-1}{2\sqrt{2}} \end{bmatrix} \tag{5.4.4}$$

As can be seen, this is a vector in the opposite direction to $\vec{\mathbf{a}}$ with magnitude V_B. The resultant space vector for this switching state is the sum of the two vectors which is

$$\tfrac{2}{3}V_A\begin{bmatrix} 1 \\ 0 \\ \frac{1}{2\sqrt{2}} \end{bmatrix} + \tfrac{2}{3}V_B\begin{bmatrix} -1 \\ 0 \\ -\frac{1}{2\sqrt{2}} \end{bmatrix} = \tfrac{2}{3}V_{AB}\begin{bmatrix} 1 \\ 0 \\ \frac{1}{2\sqrt{2}} \end{bmatrix} \tag{5.4.5}$$

and gives a vector in the direction of $\vec{\mathbf{a}}$ with a magnitude of V_{AB}, and is the same result as shown in equation (5.4.2).

Now, if the values $8, 4, 2, 1$ are assigned to the primary space vectors defining the output legs $(\vec{\mathbf{a}}, \vec{\mathbf{b}}, \vec{\mathbf{c}}, \vec{\mathbf{n}})$, the above gives

$$\begin{aligned}\vec{\mathbf{a}} &= 8 \\ \vec{\mathbf{b}} + \vec{\mathbf{c}} + \vec{\mathbf{n}} &= 4 + 2 + 1 = 7\end{aligned} \tag{5.4.6}$$

and, as noted above, the vector generated by $\vec{\mathbf{b}} + \vec{\mathbf{c}} + \vec{\mathbf{n}}$ is the opposite of $\vec{\mathbf{a}}$ and so can be considered to be part of a pair, one vector being the inverse of the other. If the values assigned the two vectors in this pair are then added together, the result equals 15.

In the same way, each of the other space vectors can be generated by the addition of 2 or more of the 4 primary vectors. If this is done for all combinations of the four primary space vectors, that give stationary vectors from Table 5.2, this results in a total of 10 extra vectors, giving a total of 14 space vectors. Each vector has a unique value of 1 to 14 generated by adding together the values assigned to the primary vectors, and each vector having an inverse that it is paired with. Just as with the example above, if the two values associated with the vector pair are added together, the result is always 15. This lead to the vectors being labelled with their assigned numbers, so for example $\vec{\mathbf{a}}$ is $\vec{V}8$ and $\vec{\mathbf{b}} + \vec{\mathbf{c}} + \vec{\mathbf{n}}$ is $\vec{V}7$. Figure 5.6 shows these 14 vectors plotted in $\alpha\beta\gamma$ space.

Looking once again at Table 5.2, it can be seen that the switching states that produce these vectors have all been labelled as positive/negative pairs(eg. ± 1), one being the inverse of the other. This is because each of these pairs of switching states generates a corresponding pair of space vectors that, like the switching states, are the inverse of each other. For example, switching state $+1$ generates the voltage space vector $\vec{V}8$, while switching state -1 generates the opposite space vector $\vec{V}7$.

It should also be noted that this single pair of switching states is not unique in generating this pair of voltage space vectors. Looking at Table 5.2 shows that for each pair of output voltage space vectors there are three pairs of switching states that can be used to generate it. Each of these different switching state pairs produces the space vector using a different pair of input phases. It is this the symmetry between the switching states and the output space vectors which means that at any given point in the input voltage cycle, there will be a set of switching states that can create any required output voltage space vector.

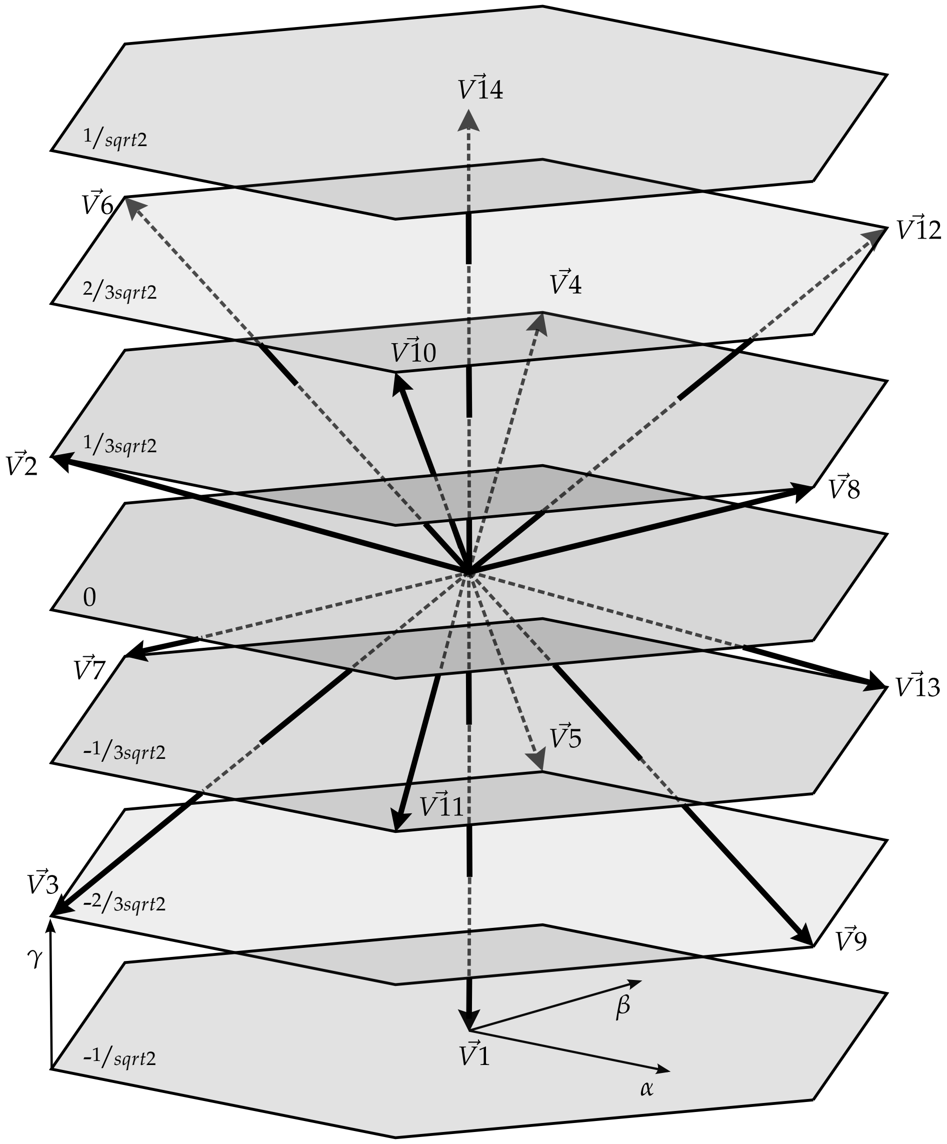


Figure 5.6: Stationary vectors plotted in $\alpha\beta\gamma$ space

5.4.2 Input Current and Switching States

Now looking at the input current, and in a similar way to above, if switching state $+\mathbf{1}$ is once again examined it can be seen that only two input phases are used, V_A and V_B, so no current flows in the third phase, V_C. Also, as only a single output leg is switched to V_A, with the other output legs switched to V_B the only current flowing is that from output leg **a**, also taking into account equation (5.2.2) then

$$\vec{I6} = I_a.\vec{\mathbf{A_i}} + (-I_a).\vec{\mathbf{B_i}} \qquad (5.4.7)$$

substituting in from (5.2.3) gives

$$\begin{aligned}\vec{I6} =& \frac{2}{3}I_a \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{2}{3}I_a \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \\ \vec{I6} =& I_a \begin{bmatrix} 1 \\ \frac{-1}{\sqrt{3}} \end{bmatrix} \end{aligned} \tag{5.4.8}$$

which defines a space vector with magnitude $\frac{2}{\sqrt{3}}I_a$ and a fixed phase displacement of $-\frac{\pi}{6}$. If this process is repeated for each of the switching states, as seen in the right-hand columns in Table 5.2, it can be seen that the input current only has six space vectors, labelled $\vec{I1}$ to $\vec{I6}$, and that each can be generated using any of seven switching states. These vectors can be seen plotted in the $\alpha\beta$ space in Figure 5.7, and it can be seen that although the hexagonal shape made by the vectors is the same, with an angle of $\frac{\pi}{3}$ between each vector, the overall space has been rotated by $\frac{\pi}{6}$ to that seen in Figure 5.4.

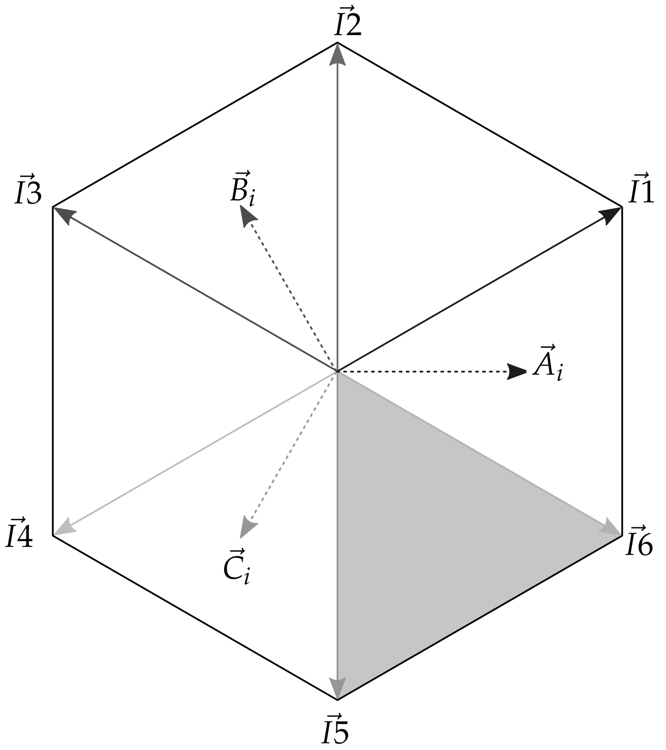


Figure 5.7: Stationary Current vectors plotted in $\alpha\beta$ space

As with the voltage vectors above, this means that there will always be a set of switching states that can be used to generate any particular input current space vector irrespective of the required output space vector. It is this ability to decouple the input vectors from the output vectors that allows the independent control of both output voltage and input current phase[4].

5.5 Determining the output sector

With all the input and output vectors now defined in terms of the available switching states, the next stage is to identify which of these vectors is required generate the demanded output voltage. This is initially done by calculating which input and output sectors the input current and output voltage vectors reside within.

Looking at the output first, which as previously stated has four output legs and is transformed using equation (5.3.4) into a three-dimensional ($\alpha\beta\gamma$) space as shown in Figure 5.6. From Table 5.2, and as shown in Section 5.4.1, there are fourteen fixed space vectors describing this volume[11, 12, 14]. On closer inspection it can be seen that this volume is actually a superset of the two-dimensional $\alpha\beta$ plane, which can be seen in Figure 5.8 where the $\alpha\beta\gamma$ space is projected on the $\alpha\beta$ plane[11] by looking along the γ axis.

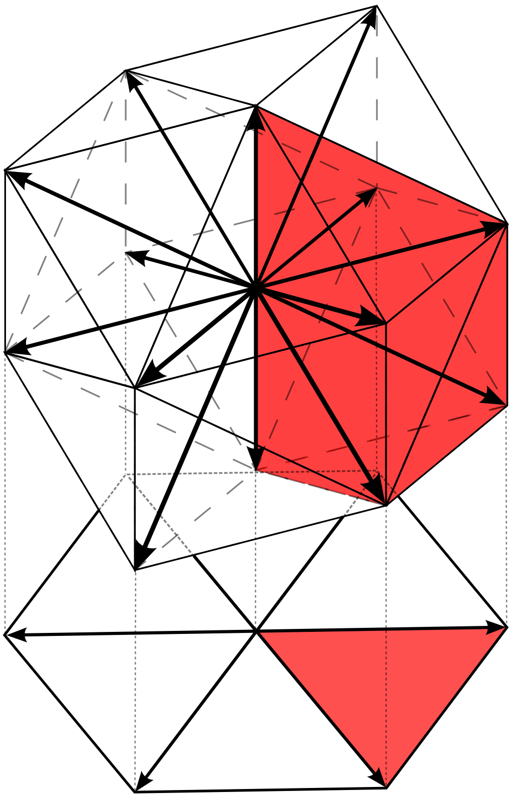


Figure 5.8: Projection of the $\alpha\beta\gamma$ Vector Space onto the $\alpha\beta$ plane

In Figure 5.8 there can be seen two areas which are shaded red. These show a prism taken from the switching volume in $\alpha\beta\gamma$ space and made up from two rows of adjacent switching vectors along its boundaries, and as can be seen, this prism corresponds to a sector in the $\alpha\beta$ plane. Now looking back at Figure 5.5, the prism is shown as the lighter blue shaded area, while the darker red shaded area in this figure shows the smallest possible bounded volume for the output voltage space. This is a tetrahedron bounded by three of the stationary space vectors at its vertices, and this tetrahedron is one of four stacked on top of one another that makes up a single prism. Figure 5.9 shows how the tetrahedrons fit together to form a single prism.

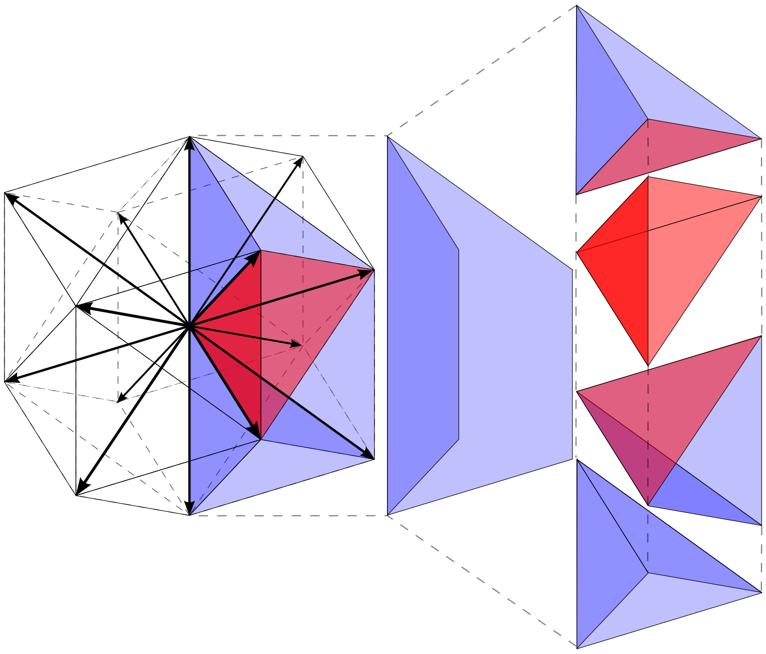


Figure 5.9: Exploded Diagram of how the Tetrahedrons make up a Prism

The way that four tetrahedrons fit together to form a single prism, and then project as a single sector into the $\alpha\beta$ plane means that it is relatively simple to calculate which prism the output voltage vector sits within, the harder part however, is to calculate which one of the 4 tetrahedrons it lies within, although this is been simplified greatly by using the method found by Zhangl[14].

So, if we take a set of output demand voltages, V_d, and using the example given in Figure 5.3(b), it can be defined as

$$V_d = \begin{bmatrix} V_a(120) \\ V_b(-164) \\ V_c(44) \\ V_n(0) \end{bmatrix} \tag{5.5.1}$$

and transform this set into the three-dimensional $\alpha\beta\gamma$ space by using (5.3.4), giving the following result

$$\begin{bmatrix} X_\alpha(120) \\ X_\beta(-120) \\ X_\gamma(0) \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \end{bmatrix} \begin{bmatrix} V_a(120) \\ V_b(-164) \\ V_c(44) \\ V_n(0) \end{bmatrix} \tag{5.5.2}$$

As has been stated previously in this section, and shown in Figure 5.8, by looking along the γ-axis this three-dimensional space can be seen projected on the $\alpha\beta$ plane. This projection is in the same form as the input voltage plane, and as such can be seen to be made up of six sectors. By comparing (5.1.5) and (5.3.4), and ignoring the third dimension in (5.3.4), the transforms are then identical. This allows the output voltage vector to be easily converted into the 2-dimensional $\alpha\beta$ plane by ignoring the γ axis, and from here it is trivial to calculate the phase angle in this plane by using equation (5.5.3) below, which specifies which prism the output voltage vector lies within.

$$Prism_o = \begin{cases} 1, & \text{for } 0 \le \theta_{\alpha\beta} < \frac{\pi}{3} \\ 2, & \text{for } \frac{\pi}{3} \le \theta_{\alpha\beta} < \frac{2\pi}{3} \\ 3, & \text{for } \frac{2\pi}{3} \le \theta_{\alpha\beta} < \pi \\ 4, & \text{for } \pi \le \theta_{\alpha\beta} < \frac{4\pi}{3} \\ 5, & \text{for } \frac{4\pi}{3} \le \theta_{\alpha\beta} < \frac{5\pi}{3} \\ 6, & \text{for } \frac{5\pi}{3} \le \theta_{\alpha\beta} < 0 \end{cases} \tag{5.5.3}$$

where $\theta_{\alpha\beta}$ is the phase angle of the output voltage in the $\alpha\beta$ plane.

So, using the values defined in equation (5.5.1) and ignoring γ, the phase of the demand vector in the $\alpha\beta$ plane can be calculated by

$$\theta_{\alpha\beta} = \arctan\left(\frac{X_\beta}{X_\alpha}\right) = \arctan\left(\frac{-140}{140}\right) = \frac{-\pi}{4} = \frac{7\pi}{4} \qquad (5.5.4)$$

Taking this result and comparing it with (5.5.3), it can be seen that this output demand voltage lies within Prism 6.

Once the prism has been determined, the exact tetrahedron containing the demand vector needs to be found. This is achieved by using a method first described by Zhang[14], where the tetrahedron is found using a lookup table, and the one required for the 4-leg matrix converter is shown in Table 5.3. Along with specifying which tetrahedron the demand vector lies within, this table also gives the three output space vectors which make up the vertices of that tetrahedron, and these are the space vectors required to be able to create any demand vector within that tetrahedron.

Table 5.3: Tetrahedron Lookup Table

		Tetrahedron			
		1	2	3	4
Prism	1	Vectors $V1$ $V9$ $V13$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ -$	Vectors $V8$ $V9$ $V13$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ -\ -$	Vectors $V8$ $V12$ $V13$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ -$	Vectors $V8$ $V12$ $V14$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ +$
	2	Vectors $V1$ $V5$ $V13$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ -$	Vectors $V4$ $V5$ $V13$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ +\ -$	Vectors $V4$ $V12$ $V13$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ -$	Vectors $V4$ $V12$ $V14$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ +$
	3	Vectors $V1$ $V5$ $V7$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ -$	Vectors $V4$ $V5$ $V7$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ +\ -$	Vectors $V4$ $V6$ $V7$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ +\ +$	Vectors $V4$ $V6$ $V14$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ +$
	4	Vectors $V1$ $V3$ $V7$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ -$	Vectors $V2$ $V3$ $V7$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ +$	Vectors $V2$ $V6$ $V7$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ +\ +$	Vectors $V2$ $V6$ $V14$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ +$
	5	Vectors $V1$ $V3$ $V11$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ -$	Vectors $V2$ $V3$ $V11$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ +$	Vectors $V2$ $V10$ $V11$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ -\ +$	Vectors $V2$ $V10$ $V14$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ +$
	6	Vectors $V1$ $V9$ $V11$ $V_{an}\ V_{bn}\ V_{cn}$ $-\ -\ -$	Vectors $V8$ $V9$ $V11$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ -\ -$	Vectors $V8$ $V10$ $V11$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ -\ +$	Vectors $V8$ $V10$ $V14$ $V_{an}\ V_{bn}\ V_{cn}$ $+\ +\ +$

Zhang found that if you know the prism that the demand vector lies within, then it is possible to find the correct tetrahedron by matching the signs of the three line-neutral voltages V_{an}, V_{bn}, V_{cn} for the demand with the polarities of each of the switching vectors which make up the vertices of the four tetrahedrons within a single prism. So, from the example above, where it has already been shown that the output demand voltage is within Prism 6, the phase-neutral demand voltages are

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \begin{bmatrix} 120 \\ -164 \\ 44 \end{bmatrix} = \begin{bmatrix} + \\ - \\ + \end{bmatrix} \tag{5.5.5}$$

Now if these are compared with the signs of the tetrahedrons which go to make up Prism 6 in Table 5.3, this gives the result that the output demand voltage is contained within tetrahedron 3 of that prism, and the table then gives the three bounding space vectors as being $\vec{V}8$, $\vec{V}10$ and $\vec{V}11$.

5.6 Determining the input sector

Moving onto the input current, and once again looking at Figure 5.7 it can be seen that one area is shaded, and this shows the smallest bounded area for the input current space. Any space vector within this area can be generated by combining the two stationary space vectors that bound either side, and this can be seen in Figure 5.10.

As before, these bounded areas are called sectors and are numbered as shown in Figure 5.10.

For the matrix converter where the primary concern is controlling the output voltage, while also controlling ensuring that the converter has a unity input displacement factor, it is then not the input current magnitude that is directly controlled, as this is dependant on both the output voltage and the characteristics of the load, instead it is the phase of the input current which needs be controlled for a unity displacement factor. Hence the input current vector will be maintained in phase with the input voltage vector. This phase relationship allows a simple determination of which of the input current sectors the demand vector lies within for any set of input voltages[4], as it will have the same phase angle as the input voltage vector, and where the input current sectors are defined as

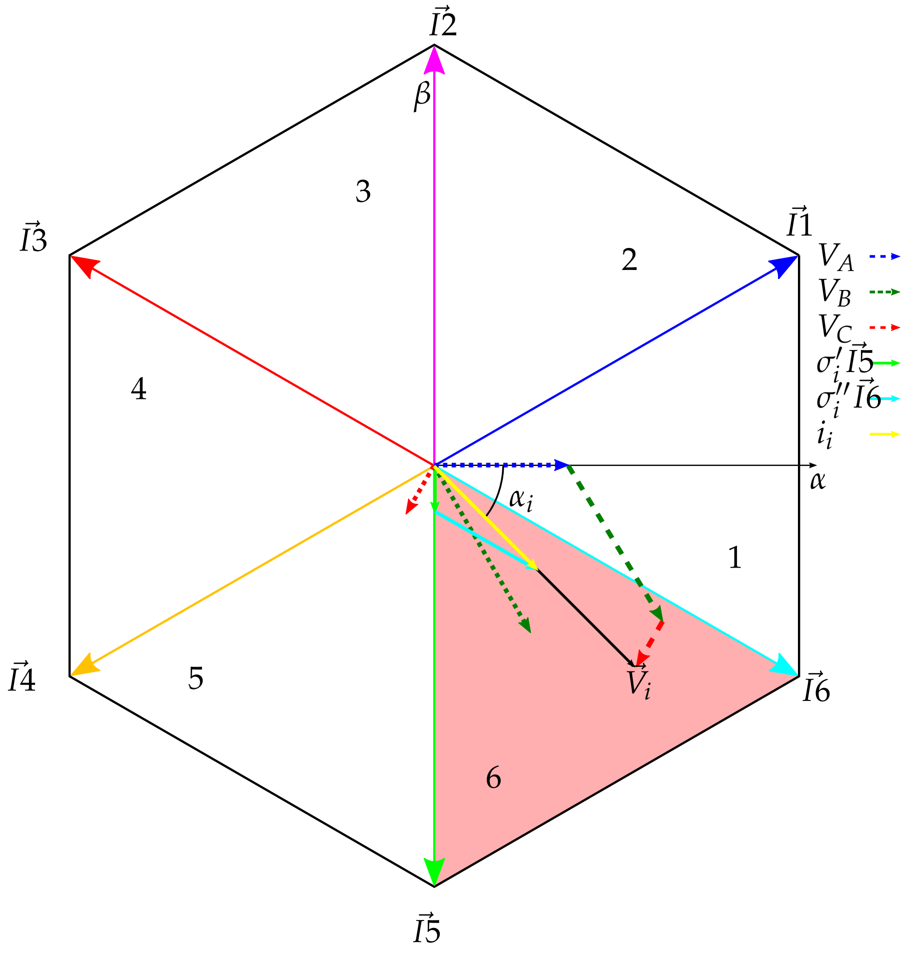


Figure 5.10: Input Current Plane showing input current vector i_i

$$\text{Sector} = \begin{cases} 1, & \text{for } \frac{11\pi}{6} \leq \phi_{\alpha\beta} < \frac{\pi}{6} \\ 2, & \text{for } \frac{\pi}{6} \leq \phi_{\alpha\beta} < \frac{\pi}{2} \\ 3, & \text{for } \frac{\pi}{2} \leq \phi_{\alpha\beta} < \frac{5\pi}{6} \\ 4, & \text{for } \frac{5\pi}{6} \leq \phi_{\alpha\beta} < \frac{7\pi}{6} \\ 5, & \text{for } \frac{7\pi}{6} \leq \phi_{\alpha\beta} < \frac{3\pi}{2} \\ 6, & \text{for } \frac{3\pi}{2} \leq \phi_{\alpha\beta} < \frac{11\pi}{6} \end{cases} \tag{5.6.1}$$

with $\phi_{\alpha\beta}$ being the phase angle of the input voltage vector within the $\alpha\beta$ plane. This can be seen in Figure 5.10 where the input voltage phasor is within the shaded area, which is defined as current sector 6 and the required input space vectors are $\vec{I}5$ and $\vec{I}6$.

5.7 Selecting the Switching States

Now that both the input and output sectors are known for the required output voltage and input current phase, these are then used to define the set of switching states that are needed. However, due to the extra output leg this is more complex than with the 3x3 matrix converter. The switching states are found by comparing the two sets of possible switching states that can be used to generate the input current vector and output voltage vector, and by using the measured input line-to-line voltages to select the correct set.

For the output, as has been described above in Section 5.4.1, the entire output voltage space is defined by fourteen vectors and using the symmetry between the vectors themselves and the switching states used to create them, it is easy to see that for each output space vector there are three ($\frac{42}{14}$) associated switching state pairs, so as there are three output space vectors, this then gives nine possible switching state pairs that can be used within any one tetrahedron.

In a similar manner for the input, the input space is defined by six space vectors and using the inherent symmetry between the vectors themselves, and switching states, as set out in Sections 5.4.1 and 5.4.2, it can be seen that for each input current space vector there are seven ($\frac{42}{6}$) associated switching state pairs. So, as there are two input current space vectors for each input current sector, this gives fourteen possible switching state pairs that can be used for any one stationary space vector.

Continuing to the examples from above, where the input current is in sector 6, so requiring input vectors $\vec{I}5$ and $\vec{I}6$ and the output voltage in tetrahedron 3 of prism 6, requiring output vectors $\vec{V}8$, $\vec{V}10$ and $\vec{V}11$. Then, referring to Table 5.2 it is possible to select all of the possible switching states that are able to generate the required vectors, remembering that the vectors themselves come in positive/negative pairs and so the switching states associated with both of these should be used here.

So for the example above the Table 5.4 shows all the possible switching states that can be used to generate the different input and output vectors. However, only those switching states which correspond to both the input current and output voltage vectors can possibly be used with this combination of input current sector and output tetrahedron, and these are shown as the Matching States in Table 5.4.

Looking at these six matching switching state pairs in Table 5.2, where the required output vectors, $V8$, $V10$ and $V11$, can be created by these

Table 5.4: Switching state selection example

	Input	Output
Space Vectors Required	Current Sector 1 $\begin{cases} I1 \\ I6 \end{cases}$	Voltage Sector 6:3 $\begin{cases} V8 \\ V10 \\ V11 \end{cases}$
Possible Switching States	$I5: \pm 2, \pm 5, \pm 8, \pm 11, \pm 14, \pm 17, \pm 20$ $I6: \pm 1, \pm 4, \pm 7, \pm 10, \pm 13, \pm 16, \pm 19$	$V8: \pm 1, \pm 2, \pm 3$ $V10: \pm 16, \pm 17, \pm 18$ $V11: \pm 4, \pm 5, \pm 6$
Matching Switching States	$\pm 1, \pm 2, \pm 4, \pm 5, \pm 16, \pm 17$	
Actual Switching States used	$+1, -2, -4, +5, +16, -17$	

matching switching states, however the value of the switched line-to-line voltage will dictate which of these 12 switching states are actually used. Once again using the example above, it is known that the input current vector is within sector 6, where as can be seen in Figure 5.11, V_{AB} will always be positive and V_{BC} will always be negative. Looking further at Figure 5.11 it can be seen that for every input current sector, there is always a single line voltage which is always positive, and is always the most positive line voltage. There is also a single line voltage which is also always negative, and this is always the most negative line voltage. It is the point where the third line voltage either become the most positive, or the most negative, when the input current sector changes.

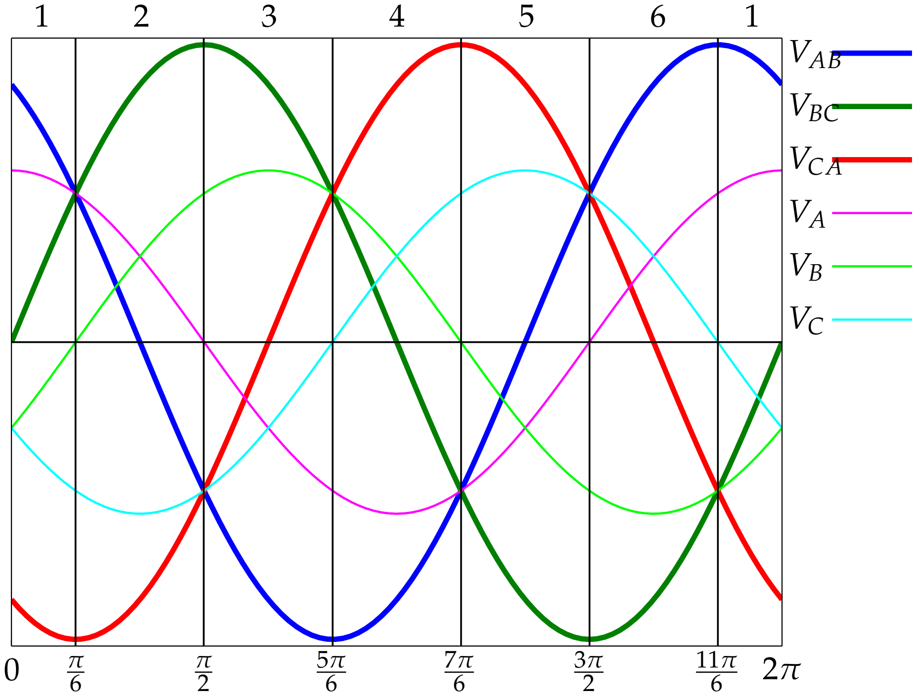


Figure 5.11: Line and Phase Voltage plots showing current sector

Now looking at the switching state pair ±2 that will be used to generate the vector $\vec{V}8$. This switching state uses the input phases V_B and V_C and switches them to the outputs, so, knowing this and using the fact that with V_{BC} will always be negative in this input current sector, it can be seen from Table 5.2 that switching state -2 is needed. Using the symmetry of the vectors, switching state $+2$ would normally create $V8$ if is V_{BC} was positive, but as V_{BC} is negative in this case, switching state $+2$ would actually create vector $\vec{V}7$. The opposite is true for switching state -2, which is why it is selected in this case.

However, if the same exercise is performed for the switching state pair ± 1, which is also being used to generate vector $V8$ except this time it is using V_{AB}, which in this example is positive, the switching state $+1$ will need to be used. Completing this for each of the switching state pairs listed gives the six switching states which are required for the example, and these are shown at the bottom of Table 5.4 labelled Actual Switching Stated used.

5.8 Combining the Input and Output Spaces

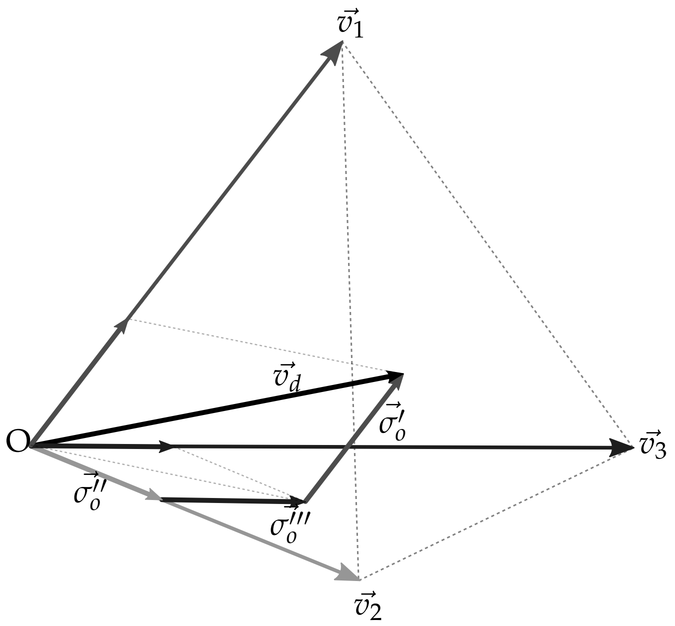


Figure 5.12: Demand Vector generation from the 3 Space Vectors

From the vector diagram given in Figure 5.12 the following equation can be written down for the demand voltage

$$\vec{v}_d = \vec{\sigma}_o' + \vec{\sigma}_o'' + \vec{\sigma}_o''' \tag{5.8.1}$$

Table 5.5: The Base Vectors for the stationary space vectors

	$\vec{V}0$	$\vec{V}1$	$\vec{V}2$	$\vec{V}3$	$\vec{V}4$	$\vec{V}5$	$\vec{V}6$	$\vec{V}7$
	0	$\vec{\mathbf{n}}$	$\vec{\mathbf{c}}$	$\vec{\mathbf{c}}+\vec{\mathbf{n}}$	$\vec{\mathbf{b}}$	$\vec{\mathbf{b}}+\vec{\mathbf{n}}$	$\vec{\mathbf{b}}+\vec{\mathbf{c}}$	$\vec{\mathbf{b}}+\vec{\mathbf{c}}+\vec{\mathbf{n}}$
	$nnnn$	$nnnp$	$nnpn$	$nnpp$	$npnn$	$npnp$	$nppn$	$nppp$
V_α	0	0	$-\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$	$-\frac{2}{3}$
V_β	0	0	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	0	0
V_γ	0	$-\frac{1}{\sqrt{2}}$	$\frac{1}{3\sqrt{2}}$	$-\frac{2}{3\sqrt{2}}$	$\frac{1}{3\sqrt{2}}$	$-\frac{2}{3\sqrt{2}}$	$\frac{2}{3\sqrt{2}}$	$-\frac{1}{3\sqrt{2}}$

	$\vec{V}8$	$\vec{V}9$	$\vec{V}10$	$\vec{V}11$	$\vec{V}12$	$\vec{V}13$	$\vec{V}14$	$\vec{V}15$
	$\vec{\mathbf{a}}$	$\vec{\mathbf{a}}+\vec{\mathbf{n}}$	$\vec{\mathbf{a}}+\vec{\mathbf{c}}$	$\vec{\mathbf{a}}+\vec{\mathbf{c}}+\vec{\mathbf{n}}$	$\vec{\mathbf{a}}+\vec{\mathbf{b}}$	$\vec{\mathbf{a}}+\vec{\mathbf{b}}+\vec{\mathbf{n}}$	$\vec{\mathbf{a}}+\vec{\mathbf{b}}+\vec{\mathbf{c}}+\vec{\mathbf{n}}$	0
	$pnnn$	$pnnp$	$pnpn$	$pnpp$	$ppnn$	$ppnp$	$pppn$	$pppp$
V_α	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0
V_β	0	0	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	0	0
V_γ	$\frac{1}{3\sqrt{2}}$	$-\frac{2}{3\sqrt{2}}$	$\frac{2}{3\sqrt{2}}$	$-\frac{1}{3\sqrt{2}}$	$\frac{2}{3\sqrt{2}}$	$-\frac{1}{3\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0

where $\vec{\sigma_o'}$, $\vec{\sigma_o''}$ and $\vec{\sigma_o'''}$ are defined as the 3 vectors that when combined make up the required output demand voltage vector in $\alpha\beta\gamma$ space and

$$\vec{\sigma_o'} = v_o'\vec{v}_1 \tag{5.8.2}$$

$$\vec{\sigma_o''} = v_o''\vec{v}_2 \tag{5.8.3}$$

$$\vec{\sigma_o'''} = v_o'''\vec{v}_3 \tag{5.8.4}$$

where v_o', v_o'' and v_o''' are the magnitudes, and $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$ are the base vectors in the directions of the three output space vectors. These base vectors are the direction vectors for the 14 stationary output switching vectors, along with the 2 zero state vectors, and these are all generated from the primary vectors (5.3.6) as shown in Section 5.4.1 and listed in Table 5.5.

So, it is known that the smallest bound volume in the output space is the tetrahedron, and that it requires 3 stationary voltage vectors to define the vertices of this tetrahedron. From Section 5.7 it is known that for each of the 3 stationary vectors are themselves created by using two switching states. This therefore means that for each of the vectors $\vec{\sigma_o'}$, $\vec{\sigma_o''}$ and $\vec{\sigma_o'''}$, there are two switching states associated with them, so therefore

$$\vec{\sigma_o'} = v_o'\vec{v}_1 = v_o^I\vec{v}_1\delta^I + v_o^{VI}\vec{v}_1\delta^{VI} \tag{5.8.5}$$

$$\vec{\sigma_o''} = v_o''\vec{v}_2 = v_o^{II}\vec{v}_2\delta^{II} + v_o^{V}\vec{v}_2\delta^{V} \tag{5.8.6}$$

$$\vec{\sigma_o'''} = v_o'''\vec{v}_3 = v_o^{III}\vec{v}_3\delta^{III} + v_o^{IV}\vec{v}_3\delta^{IV} \tag{5.8.7}$$

where δ^I to δ^{VI} are the duty cycles for the six switching states, and v_o^I to v_o^{VI} are the instantaneous voltages generated at the output legs.

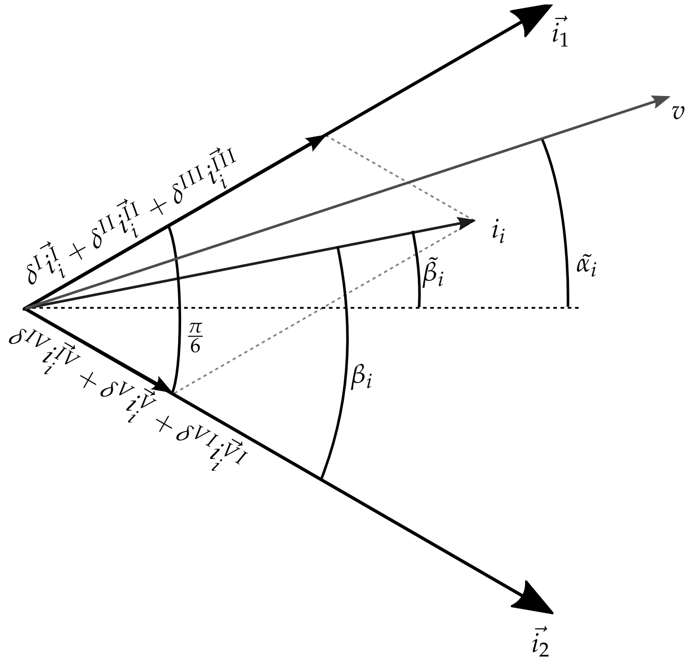

$\vec{i}_1$
v_i
$\delta^I\vec{i}_i^I + \delta^{II}\vec{i}_i^{II} + \delta^{III}\vec{i}_i^{III}$
i_i
$\tilde{\beta}_i$
$\tilde{\alpha}_i$
$\frac{\pi}{6}$
$\delta^{IV}\vec{i}_i^{IV} + \delta^{V}\vec{i}_i^{V} + \delta^{VI}\vec{i}_i^{VI}$
β_i
$\vec{i}_2$

Figure 5.13: Input Current Vector generation from the 2 Space Vectors

For in the input current vector, the situation is slightly different. As stated above, the converter is not directly controlling the magnitude of the input current vector, instead it is controlling the phase. As this is done to give the converter a unity displacement factor this means that the phase angle of the input current should match the phase angle of the input voltage, $\phi_{\alpha\beta}$. So, as the input current vector is required to follow the input voltage vector and looking at Figure 5.13, the following equations, (5.8.8) to (5.8.10), can be written down. These equations force a zero value onto the perpendicular component of the required current vector with respect to the input voltage vector. This perpendicular component is signified by the $je^{j\beta_i}$ term. This is done in the same way as for the 3x3 matrix converter[4] and so gives

$$\left(\vec{i_i^I}\delta^I + \vec{i_i^{VI}}\delta^{VI}\right) \cdot je^{j\tilde{\beta}_i}e^{j(K_i-1)\frac{\pi}{3}} = 0 \qquad (5.8.8)$$

$$\left(\vec{i_i^{II}}\delta^{II} + \vec{i_i^{V}}\delta^{V}\right) \cdot je^{j\tilde{\beta}_i}e^{j(K_i-1)\frac{\pi}{3}} = 0 \qquad (5.8.9)$$

$$\left(\vec{i_i^{III}}\delta^{III} + \vec{i_i^{IV}}\delta^{IV}\right) \cdot je^{j\tilde{\beta}_i}e^{j(K_i-1)\frac{\pi}{3}} = 0 \qquad (5.8.10)$$

where K_i is the input current sector and $\tilde{\beta}_i$ is the input current phase angle measured with respect to a line bisecting the input current sector, and is limited by

$$-\frac{\pi}{6} \leq \tilde{\beta}_i \leq \frac{\pi}{6} \qquad (5.8.11)$$

What this means is that a particular output voltage vector is created by the sum of the two vectors created by the associated pair of switching states (5.8.5), however these two switching states create different vectors within the input current plane and so by balancing the ratio of the duty cycles of this pair of switching states, it is possible to create a resultant input current vector which is directly in phase with the input voltage vector, or in other words, has a zero length component in the direction perpendicular to the input voltage vector, and this is what equation 5.8.8 states.

So, looking at Figure 5.13, and taking into account the restriction being forced in equation (5.8.8) which forces the components of each vector in the direction perpendicular to the input voltage vector to equal one another, so as the two current space , it is then possible to write down the following ratio

$$|\vec{i_i^I}|\delta^I \cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right) = |\vec{i_i^{VI}}|\delta^{VI} \cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right) \qquad (5.8.12)$$

then due to the current stiff nature of the output of the matrix converter, and that the two switching states being used to create a single output voltage space always switch the same current, as shown in Table 5.2, this means that

$$|\vec{i_i^I}| = |\vec{i_i^{VI}}| \qquad (5.8.13)$$

therefore

$$\frac{\delta^{I}}{\delta^{VI}} = \frac{\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)}{\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)} = \frac{\delta^{II}}{\delta^{V}} = \frac{\delta^{III}}{\delta^{IV}} \tag{5.8.14}$$

Substituting (5.8.14) into (5.8.5) then gives

$$v_o' = v_o^I \delta^I + v_o^{VI} \delta^I . \frac{\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)}{\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)} \tag{5.8.15}$$

As has been shown, as a consequence of the symmetry between the input and output vectors it can be seen that the pair of switching states used to generate a particular output voltage vectors are used to generate different input current vectors. This means that by changing the duty ratios of the two switching states with respect to one another, it is possible to alter either the input current phase, or output voltage vector, independently. So, for a particular output voltage vector, the input current phase can be any value within the limits of the sector, and likewise, for any particular input current phase, the output voltage vector can have any value within the limits of the tetrahedron.

This result shows that for the 4-leg matrix converter, the input current phase and output voltage vector can be completely uncoupled from one another, and this therefore allows complete control over the output voltage and input current phase.

5.9 Calculating the Duty Cycles

The final stage of this derivation is to determine the required duty ratios for each of the 6 switching states.

Initially, the length of each of the three vectors $\vec{\sigma_o'}$, $\vec{\sigma_o''}$ and $\vec{\sigma_o'''}$ which produce the demand voltage is required. One solution to this problem is given by Zhang[14] by using a lookup table, however, while this works well for the 4-leg matrix converter, it was not thought to be general enough for use with the 4-leg matrix converter. So, a more general approach to solving this problem was sought, initially using trigonometry in a similar fashion to that employed by Casadei[4] for the 3x3 matrix converter. This initial method was unsuccessful as, while it was possible to perform the calculation, the addition of the third dimension complicated things greatly, and on a practical level the approach did not appear to be simple, or suitable, for implementation in a DSP.

After further investigation a solution was found using the vector nota-

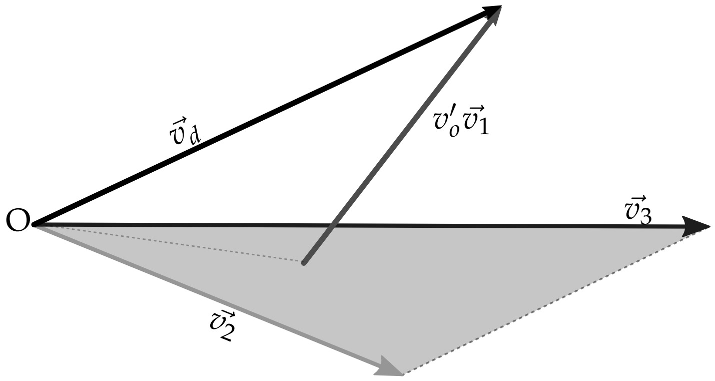


Figure 5.14: Distance of the vector $\vec{v}_d$ from the plane Ov_2v_3

tion already used to describe the vectors. Figure 5.14 shows how this method works, where a plane is defined which contains the Origin and the two vectors $\vec{v}_2$ and $\vec{v}_3$. As the plane contains two of the three switching vectors which define one of the output tetrahedrons, any deviation of the demand vector from this plane can only be due to the third vector $\vec{v}_1$. The distance of the demand vector from the plane in the direction of $\vec{v}_1$ can then easily be calculated. Taking the general vector equation of a plane using the dot product as

$$\vec{n_{23}} \cdot (P - P_0) = 0 \tag{5.9.1}$$

where P_0 is a point in the plane and $\vec{n_{23}}$ is the normal to the plane, and is defined as

$$\vec{n_{23}} = \vec{v}_2 \times \vec{v}_3 \tag{5.9.2}$$

In this case, and that of every plane that is of interest, the plane always passes through the Origin, so the equation then becomes

$$\vec{n_{23}} \cdot (P) = 0 \tag{5.9.3}$$

Now, the equation of a line is defined as

$$P = P_1 - u(P_2 - P_1) \tag{5.9.4}$$

where P_1 and P_2 are two points on the line and $(P_2 - P_1)$ defines the direction, and so for every value of u there is a point P that lays on the line. In this case we want to know the distance, in the direction of vector $\vec{v}_1$, from the tip of the demand vector $\vec{v}_d$ to the plane formed by the vectors $\vec{v_2}$ and $\vec{v_3}$. Therefore, in this case

$$\begin{aligned} P_1 &= \vec{v_d} \\ P_2 - P_1 &= \vec{v_1} \end{aligned}$$

therefore

$$P = \vec{v_d} - v_o'\vec{v_1} \tag{5.9.5}$$

Substituting (5.9.5) into (5.9.3) then gives

$$\vec{n_{23}} \cdot (\vec{v_d} - v_o'\vec{v_1}) = 0 \tag{5.9.6}$$

and rearranging then gives

$$v_o' = \frac{\vec{n_{23}} \cdot \vec{v_d}}{\vec{n_{23}} \cdot \vec{v_1}} \tag{5.9.7}$$

Now, the equation for the dot product is given by

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos\theta \tag{5.9.8}$$

where θ is the angle between the two vectors $\vec{a}$ and $\vec{b}$. So, expanding equation (5.9.7) now gives

$$v_o' = \frac{|\vec{n_{23}}||\vec{v_d}| \cos\omega_o'}{|\vec{n_{23}}||\vec{v_1}| \cos\mu_1} \tag{5.9.9}$$

where μ_1 is the angle between the vector $\vec{v_1}$ and the normal vector $\vec{n_{23}}$, and ω_o' is the angle between the demand vector $\vec{v_d}$ and the normal vector $\vec{n_{23}}$.

As vector $\vec{v_1}$ is a fixed space vector, it is therefore fixed in relation to the plane Ov_2v_3 and likewise, it is fixed in relation to the normal of this plane. Therefore $|\vec{v_1}| \cos\mu_1$ is a constant when working in this tetrahedron, and it can easily be shown that, due to the symmetry between the space vectors, it is in fact constant for all tetrahedra within this space, and for all stationary space vectors, and it is equal to

$$|\vec{v_x}| \cos\mu_x = \frac{1}{\sqrt{3}} \tag{5.9.10}$$

Now substituting (5.9.10) into (5.9.9) and cancelling $|\vec{n_{23}}|$ leaves

$$v_o' = \sqrt{3}|\vec{v_d}| \cos \omega_o' \tag{5.9.11}$$

Now, substituting this into (5.8.15) gives

$$\sqrt{3}|\vec{v_d}| \cos \omega_o' = v_o^I \delta^I + v_o^{VI} \delta^I . \frac{\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)}{\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)} \tag{5.9.12}$$

Re-arranging for the duty cycle δ^I gives

$$\delta^I = \sqrt{3}|\vec{v_d}| \frac{\cos \omega_o' \cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)}{v_o^I \cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right) + v_o^{VI} \cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)} \tag{5.9.13}$$

Now, once again using the example from above, where the three output space vectors being used are $\vec{V8}$, $\vec{V10}$ and $\vec{V11}$, if it is taken that $\vec{v_1}$ corresponds to $\vec{V8}$ then the two switching states that are being used to generate this vector are $+1$ and -2 as shown in Table 5.2.

For switching state -2, this produces input space vector $\vec{I5}$, and as this would correspond to the lower of the two vectors when looking at the sector as in Figure 5.13, referring back to (5.9.13) it can be seen that the switching state is therefore controlled by duty cycle δ^{VI}. When this switching state is selected, the input phases V_B and V_C are switched onto the output, so the voltage, v_o^{VI}, generated across the output legs which is defined by

$$V_{BC} = V_B - V_C \tag{5.9.14}$$

substituting in from (5.1.3) and (5.1.4) gives

$$V_{BC} = V_i \cos\left(\alpha_i - \frac{2\pi}{3}\right) - V_i \cos\left(\alpha_i - \frac{4\pi}{3}\right) \tag{5.9.15}$$

where α_i is the input voltage phase angle. Then using the cosine sum-to-product identity gives

$$\begin{aligned}
V_{BC} &= -2.V_i\left[\sin\left(\alpha_i - \frac{\pi}{3}\right).\sin\left(\frac{4\pi}{3}\right)\right] \\
V_{BC} &= \sqrt{3}.V_i\sin\left(\alpha_i - \frac{\pi}{3}\right) \\
V_{BC} &= \sqrt{3}.V_i\cos\left(\alpha_i - \frac{5\pi}{6}\right) \qquad (5.9.16)
\end{aligned}$$

However as the equations which define the input current duty cycles (5.8.8) to (5.8.10) use a phase angle related to a line which bisects the current sector, the input voltage phase angle α_i needs to be referred to the same line, therefore

$$\tilde{\alpha_i} = \alpha_i - \frac{\pi}{3}\left(Sector_i - 1\right) \qquad (5.9.17)$$

where $Sector_i$ is in the input current sector, and so as the input voltage vector is within sector 6 equation (5.9.16) becomes

$$\begin{aligned}
V_{BC} &= \sqrt{3}.V_i\cos\left(\tilde{\alpha_i} + \frac{5\pi}{3} - \frac{5\pi}{6}\right) \\
V_{BC} &= \sqrt{3}.V_i\cos\left(\tilde{\alpha_i} + \frac{5\pi}{3}\right) \qquad (5.9.18)
\end{aligned}$$

In the example the input voltage vector is sat within input current sector 6, and as can be seen in Figure 5.11, this means that the voltage V_{BC} is negative. However switching state -2 is being used and this switches $-V_BC$ onto the output, so

$$\begin{aligned}
-V_{BC} &= -\sqrt{3}.V_i\cos\left(\tilde{\alpha_i} + \frac{5\pi}{6}\right) \\
V_{BC} &= \sqrt{3}.V_i\cos\left(\alpha_i - \frac{\pi}{2}\right) \qquad (5.9.19)
\end{aligned}$$

which means that the voltage produced alon

$$v_o^{VI} = \sqrt{3}.V_i\cos\left(\alpha_i - \frac{\pi}{2}\right) \qquad (5.9.20)$$

In a similar manner, when switching state $+1$ is used, it produces input vector $\vec{I6}$, which corresponds to the upper of the two vectors when looking at the sector as in Figure 5.13. Once again, referring back to (5.9.13) it can be seen that this switching state is therefore controlled by duty cycle δ^{VI}. So, during this input sector the voltage, v_o^I, produced at the output is defined by

$$v_o^I = \sqrt{3}.V_i \cos\left(\tilde{\alpha}_i + \frac{\pi}{6}\right) \tag{5.9.21}$$

Now, substituting (5.9.20) and (5.9.21) into equation (5.9.13) gives

$$\delta^I = \frac{|\vec{v_d}|}{V_i} \frac{\cos\omega_o' \cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right)}{\left[\cos\left(\tilde{\alpha}_i + \frac{\pi}{6}\right)\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right) + \cos\left(\tilde{\alpha}_i - \frac{\pi}{2}\right)\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)\right]} \tag{5.9.22}$$

Looking at the denominator, and now using the cosine product-to-sum rule this becomes

$$\begin{aligned}
&\cos\left(\tilde{\alpha}_i + \frac{\pi}{6}\right)\cos\left(\tilde{\beta}_i + \frac{\pi}{3}\right) + \cos\left(\tilde{\alpha}_i - \frac{\pi}{2}\right)\cos\left(\tilde{\beta}_i - \frac{\pi}{3}\right)\\
=&\frac{1}{2}\Big[\cos\left(\tilde{\alpha}_i + \tilde{\beta}_i + \frac{\pi}{2}\right) + \cos\left(\tilde{\alpha}_i - \tilde{\beta}_i - \frac{\pi}{6}\right)\\
&+ \cos\left(\tilde{\alpha}_i + \tilde{\beta}_i - \frac{\pi}{2}\right) + \cos\left(\tilde{\alpha}_i - \tilde{\beta}_i + \frac{\pi}{6}\right)\Big]
\end{aligned} \tag{5.9.23}$$

Collecting alike terms

$$\begin{aligned}
&= \frac{1}{2}\left[\cos\left(\tilde{\alpha}_i + \tilde{\beta}_i + \frac{\pi}{2}\right) + \cos\left(\tilde{\alpha}_i + \tilde{\beta}_i - \frac{\pi}{2}\right)\right]\\
&+ \Big[\cos\left(\tilde{\alpha}_i - \tilde{\beta}_i - \frac{\pi}{6}\right) + \cos\left(\tilde{\alpha}_i - \tilde{\beta}_i + \frac{\pi}{6}\right)\Big]
\end{aligned} \tag{5.9.24}$$

Using the identity $\cos(\theta + \pi/2) = -\cos(\theta)$, and then cancelling gives

$$\begin{aligned}
&= \frac{1}{2}\left[\cos\left(\tilde{\alpha}_i + \tilde{\beta}_i + \frac{\pi}{2}\right) - \cos\left(\tilde{\alpha}_i + \tilde{\beta}_i + \frac{\pi}{2}\right)\right]\\
&+ \left[\cos\left(\tilde{\alpha}_i - \tilde{\beta}_i - \frac{\pi}{6}\right) + \cos\left(\tilde{\alpha}_i - \tilde{\beta}_i + \frac{\pi}{6}\right)\right]\\
&= \frac{1}{2}\left[\cos\left(\tilde{\alpha}_i - \tilde{\beta}_i - \frac{\pi}{6}\right) + \cos\left(\tilde{\alpha}_i - \tilde{\beta}_i + \frac{\pi}{6}\right)\right]
\end{aligned} \tag{5.9.25}$$

This leaves two cosine terms, both using $\tilde{\alpha}_i - \tilde{\beta}_i$ which is the defined as ϕ, the phase lag between the input voltage and input current. Substituting this into (5.9.25) and simplifying using the cosine sum-to-product rule once again

$$\begin{aligned}
&= \frac{1}{2}\left[\cos\left(\phi - \frac{\pi}{6}\right) + \cos\left(\phi + \frac{\pi}{6}\right)\right] \\
&= \frac{1}{2}\left[2\cos(\phi)\cos\left(\frac{\pi}{6}\right)\right] \\
&= \frac{\sqrt{3}}{2}\cos(\phi) \qquad (5.9.26)
\end{aligned}$$

Substituting this back into (5.9.22) gives

$$\delta^{I} = \frac{2}{\sqrt{3}}\frac{|v_d|}{V_i}\frac{\cos\omega_o'\cos(\beta_i + \frac{\pi}{3})}{\cos\phi} \qquad (5.9.27)$$

In a similar way the equations for the other five duty cycles can be calculated as

$$\begin{aligned}
\delta^{II} &= \frac{2}{\sqrt{3}}\frac{|v_d|}{V_i}\frac{\cos\omega_o''\cos(\beta_i + \frac{\pi}{3})}{\cos\phi} \qquad (5.9.28) \\
\delta^{III} &= \frac{2}{\sqrt{3}}\frac{|v_d|}{V_i}\frac{\cos\omega_o'''\cos(\beta_i + \frac{\pi}{3})}{\cos\phi} \qquad (5.9.29) \\
\delta^{IV} &= \frac{2}{\sqrt{3}}\frac{|v_d|}{V_i}\frac{\cos\omega_o'''\cos(\beta_i - \frac{\pi}{3})}{\cos\phi} \qquad (5.9.30) \\
\delta^{V} &= \frac{2}{\sqrt{3}}\frac{|v_d|}{V_i}\frac{\cos\omega_o''\cos(\beta_i - \frac{\pi}{3})}{\cos\phi} \qquad (5.9.31) \\
\delta^{VI} &= \frac{2}{\sqrt{3}}\frac{|v_d|}{V_i}\frac{\cos\omega_o'\cos(\beta_i - \frac{\pi}{3})}{\cos\phi} \qquad (5.9.32)
\end{aligned}$$

It should be noted that $|v_d|$ in these equations is the the length of the demand vector in the $\alpha\beta\gamma$ space, and is not the peak voltage of the demand. However, if the demand voltages do form a balanced 3-phase set then the length of $|v_d|$ equates to the peak phase voltage.

5.10 Voltage Transfer Ratio

A normal procedure at this point would be to derive the voltage transfer function for the converter, however by adding the extra output leg, it now becomes possible to produce a wide range of possible waveform outputs, ranging from single phase outputs, up to the more normal balanced set of voltages, and on to having the converter produce different voltage and frequencies on each output leg. As the range of outputs that the converter is capable of producing is so large, it becomes very difficult to write a generic equation that would give a voltage transfer function in every possible case. But, as a general rule the instantaneous peak output voltage between any of the output phases, including the neutral leg, must always be less than the smallest difference between the most positive and most negative phase voltages. Figure 5.15 shows how the output voltages sit within the envelope of the input voltages.

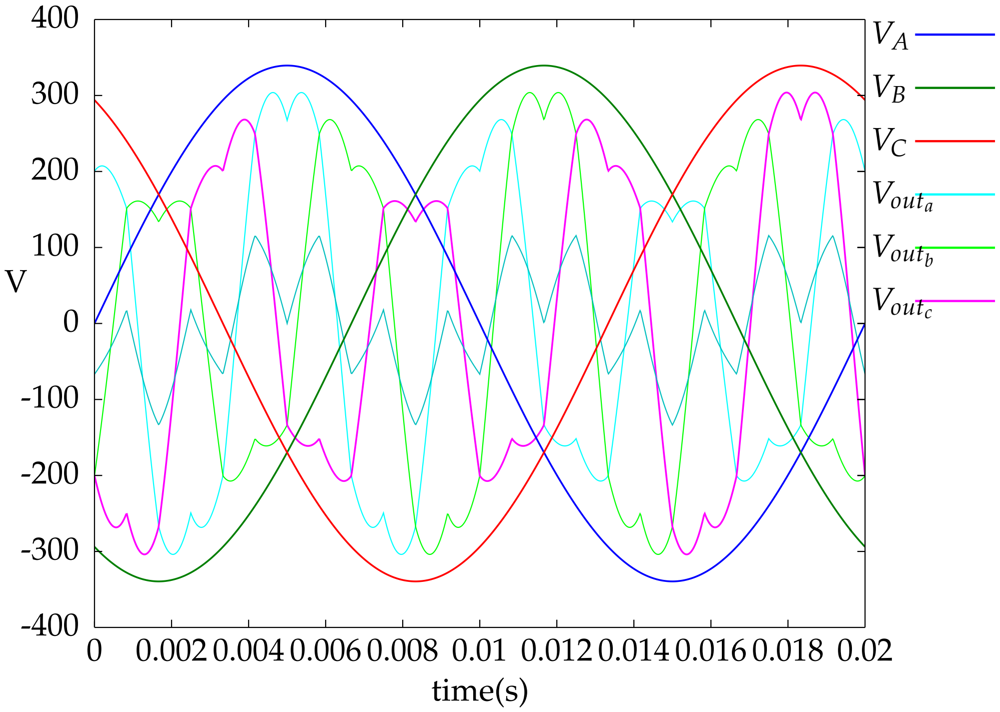


Figure 5.15: Input Phase Voltage waveforms with the Output Voltage Waveforms contained with the envelope

So, if a general, balanced or unbalanced, set of voltages is defined such that

$$\begin{aligned} V_a &= V_a \sin(\alpha) \\ V_b &= V_b \sin\left(\alpha - \frac{2\pi}{3}\right) \\ V_c &= V_c \sin\left(\alpha - \frac{4\pi}{3}\right) \end{aligned} \tag{5.10.1}$$

the difference between these phases are the respective line voltages, which in this case are

$$\begin{aligned} V_{ab} = V_a - V_b \quad &= V_a \sin(\alpha) - V_b \sin\left(\alpha - \frac{2\pi}{3}\right) \\ V_{bc} = V_b - V_c \quad &= V_b \sin\left(\alpha - \frac{2\pi}{3}\right) - V_c \sin\left(\alpha - \frac{4\pi}{3}\right) \qquad (5.10.2) \\ V_{ca} = V_c - V_a \quad &= V_c \sin\left(\alpha - \frac{4\pi}{3}\right) - V_a \sin(\alpha) \end{aligned}$$

(5.10.3)

Now, taking the exponential form of sin such that

$$\sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j} \qquad (5.10.4)$$

and substituting this into the equation for V_{ab} in (5.10.2) above gives

$$V_{ab} = V_a.\frac{e^{j\alpha} - e^{-j\alpha}}{2j} - V_b.\frac{e^{j\left(\alpha - \frac{2\pi}{3}\right)} - e^{-j\left(\alpha - \frac{2\pi}{3}\right)}}{2j} \qquad (5.10.5)$$

Re-arranging the V_b part, this then becomes

$$V_{ab} = V_a.\frac{e^{j\alpha} - e^{-j\alpha}}{2j} - V_b.e^{j\frac{2\pi}{3}}.\frac{e^{j\alpha} - e^{-j\alpha}}{2j} \qquad (5.10.6)$$

and using the exponential form of sin once more from (5.10.4)

$$V_{ab} = \left(V_a - V_b.e^{j\frac{2\pi}{3}}\right).\sin\alpha \qquad (5.10.7)$$

Now, as

$$e^{j\theta} = \cos\theta + j\sin\theta \qquad (5.10.8)$$

and solving for $\theta = \frac{2\pi}{3}$ the equation in (5.10.7) now becomes

$$V_{ab} = \left(\left(V_a + \frac{V_b}{2}\right) - jV_{bi}.\frac{\sqrt{3}}{2}\right).\sin\alpha \qquad (5.10.9)$$

This voltage, when plotted on an Argand diagram produces a vector of length $|V_{ab}|$ and phase α_{ab}, where

$$|V_{ab}| = \sqrt{\left(V_a + \frac{V_b}{2}\right)^2 + \left(V_b.\frac{sqrt3}{2}\right)^2} \tag{5.10.10}$$

and

$$\alpha_{ab} = \arctan\left(\frac{\left(V_b.\frac{sqrt3}{2}\right)}{\left(V_a + \frac{V_b}{2}\right)}\right) \tag{5.10.11}$$

It is only the magnitude of the voltage which is of interest at this point, and so re-arranging (5.10.10) then gives

$$|V_{ab}| = \sqrt{V_a^2 + V_a.V_b + V_b^2} \tag{5.10.12}$$

The same exercise can be performed for each of the other two line-line voltages giving

$$\begin{aligned} |V_{bc}| &= \sqrt{V_b^2 + V_b.V_c + V_c^2} \\ |V_{ca}| &= \sqrt{V_c^2 + V_c.V_a + V_a^2} \end{aligned} \tag{5.10.13}$$

Taking these equations, it can be easily seen that if the magnitudes of the phase voltages V_a, V_b and V_c are the same, V_x, then the magnitude of the resultant line-line voltages will all be the same, and become the well known result

$$|V_{ab}| = |V_{bc}| = |V_{ca}| = \sqrt{3}V_x \tag{5.10.14}$$

Looking at the input voltages first, and plotting the resultant line-line voltages in Figure 5.16, it can be seen that the difference between the most positive and most negative input phase voltages is the envelope of the rectified input voltage waveforms.

The entire area under the rectified waveform could be used to generate the output voltage, however, in order to guarantee that it is always possible to produce the desired output, the output line-line voltage must always be less than the minimum line-line voltage. This is because at any instant in the input voltage cycle the converter must be able to generate any point in the output voltage cycle, so the peak output line voltage

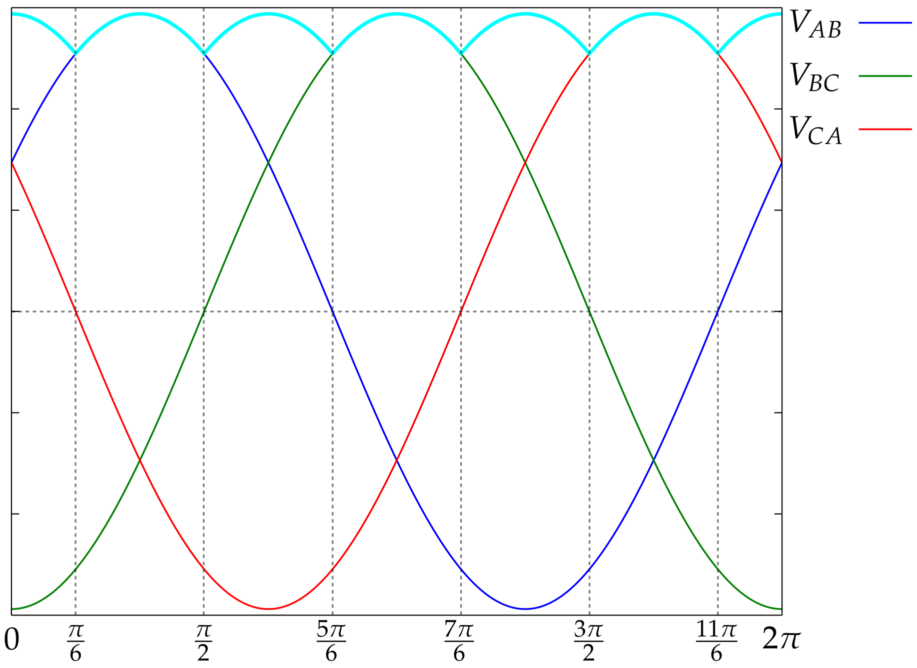


Figure 5.16: Line-Line Input Voltage waveforms, showing the envelope of the rectified waveforms

must always be less than the minimum input line voltage.

As can be seen, the minimum occurs when the input angle, α_i, equals $\frac{\pi}{3}$, therefore the peak line-line voltage, $V_{o(line)}$, which can appear between any of the output phases is defined as

$$\begin{aligned} V_{o(line)} &\leq \sqrt{3}V_i \sin\left(\frac{\pi}{3}\right) \\ &\leq \sqrt{3}V_i\frac{\sqrt{3}}{2} \\ &\leq \frac{3}{2}V_i \end{aligned} \tag{5.10.15}$$

and as

$$V_{o(line)} = \sqrt{3}V_{o(phase)} \tag{5.10.16}$$

Combining these two equations gives the well known voltage transfer function for space vector modulation

$$q \leq \frac{\sqrt{3}}{2} \tag{5.10.17}$$

Exactly the same procedure can be used with an unbalanced set of output voltages as well, however in this case there will be a different magnitudes for each of the output line voltages. This will therefore mean that there are three different results for the voltage transfer ratio, and so some care must be taken to ensure that the correct ratio is chosen.

If an unbalanced set of output voltages are demanded such that

$$\begin{aligned} V_a &= \frac{V_o}{3} \\ V_b &= \frac{2V_o}{3} \\ V_c &= V_o \end{aligned} \tag{5.10.18}$$

with V_o being the largest of the three output voltage values. Then using the equations laid out in (5.10.12) and (5.10.13) the following results are obtained

$$\begin{aligned} |V_{ab}| &= \sqrt{\left(\frac{V_o}{3}\right)^2 + \frac{V_o}{3}.\frac{2V_o}{3} + \left(\frac{2V_o}{3}\right)^2} &&= \sqrt{\left(\frac{7V_o}{9}\right)^2} = \frac{\sqrt{7}}{3}V_o \\ |V_{bc}| &= \sqrt{\left(\frac{2V_o}{3}\right)^2 + \frac{2V_o}{3}.V_o + V_o^2} &&= \sqrt{\left(\frac{19V_o}{9}\right)^2} = \frac{\sqrt{19}}{3}V_o \\ |V_{ca}| &= \sqrt{V_o^2 + V_o.\frac{V_o}{3} + \left(\frac{V_o}{3}\right)^2} &&= \sqrt{\left(\frac{13V_o}{9}\right)^2} = \frac{\sqrt{13}}{3}V_o \end{aligned} \tag{5.10.19}$$

and combining these results with the result in equation 5.10.16 then gives

$$\begin{aligned} q_{ab} &\leq \frac{3}{2}\frac{3}{\sqrt{7}} && \leq 1.70 \\ q_{bc} &\leq \frac{3}{2}\frac{3}{\sqrt{19}} && \leq 1.03 \\ q_{ca} &\leq \frac{3}{2}\frac{3}{\sqrt{13}} && \leq 1.25 \end{aligned} \tag{5.10.20}$$

and therefore

$$q \leq \mathbf{min}\left(q_{ab}, q_{bc}, q_{ca}\right) \leq 1.03 \tag{5.10.21}$$

It can be seen that with with this arrangement of unbalanced output voltages, that the maximum voltage transfer ratio here is actually greater than 1. The limit of this is when producing only a single output voltage, with the other two output phases set to zero. In this case the maximum voltage transfer ratio rises to become 1.5. This is because the converter is able to control the voltage on the neutral leg to be negative, at the same time as the output leg is being being positive, up to the maximum allowed in equation 5.10.15, which is $1.5V_i$. On further investigation, it turns out that the maximum voltage transfer ratio always tracks that calculated from the two largest phase voltages, which in the above example are V_b and V_c.

It needs to be noted here that the above derivation is only valid for unbalanced outputs if all the three controlled outputs are all at the same frequency, and have an equal phase displacement from each other. While it would be possible to re-do the above equations for a different set of phase relationships quite simply, it was found to be a significantly more complex task to produce a similar set of equations for outputs with different frequencies. As such a slightly different approach was looked at.

So, irrespective of whether the output is balanced or unbalanced, at different or the same frequency, it is also possible to calculate the instantaneous voltage transfer ratio from the duty cycles calculated earlier, and this has been laid out below.

Having previously derived all of the duty cycles, δ^{I} to δ^{VI}, it is possible to calculate the voltage transfer ratio for the converter.

$$\delta^{I} + \delta^{II} + \delta^{III} + \delta^{IV} + \delta^{V} + \delta^{VI} \leq 1 \tag{5.10.22}$$

substituting in equations (5.9.27) to (5.9.32) gives

$$\begin{aligned}\frac{2}{\sqrt{3}}\frac{v_d}{V_i}\frac{1}{\cos\phi}\Big[&\cos\omega_o'\left(\cos(\beta_i+\frac{\pi}{3})+\cos(\beta_i-\frac{\pi}{3})\right)+\\ &\cos\omega_o''\left(\cos(\beta_i+\frac{\pi}{3})+\cos(\beta_i-\frac{\pi}{3})\right)+\\ &\cos\omega_o'''\left(\cos(\beta_i+\frac{\pi}{3})+\cos(\beta_i-\frac{\pi}{3})\right)\Big]\leq 1\end{aligned} \tag{5.10.23}$$

Given that

$$\cos\left(\beta_i+\frac{\pi}{3}\right)+\cos\left(\beta_i-\frac{\pi}{3}\right)=\cos\beta_i \tag{5.10.24}$$

this can be substituted into (5.10.23) and re-arranged which therefore gives

$$q\leq\frac{\sqrt{3}}{2}\frac{\cos\phi}{\cos\beta_i}\frac{1}{(\cos\omega_o'+\cos\omega_o''+\cos\omega_o''')} \tag{5.10.25}$$

where q is the ratio of the input and output voltages.

This equation represents the maximum voltage transfer ratio for the converter at any one instant when the output phase voltages are balanced. If this equation is compared to that of the 3x3 matrix converter (3.4.13) then it can be seen that this is of a very similar form, with both equations being dependant on the input current phase angle and its displacement angle to the input voltage, plus a the output voltage phase angle. Now, whereas for (3.4.13) this is simply down to the phase angle of the output voltage, in (5.10.25) it can be seen to be more complex, and is actually dependant on the angles of the output voltage vector with respect to the normal vectors for each of the three planes $(\cos\omega_o'+\cos\omega_o''+\cos\omega_o''')$.

Just as for the 3x3 matrix converter, the maximum voltage transfer ratio for a whole cycle occurs when the denominator of (5.10.25) is at its minimum, and this occurs when $\cos\beta_i$ and $(\cos\omega_o'+\cos\omega_o''+\cos\omega_o''')$ both equal one, and so this then gives

$$q\leq\frac{\sqrt{3}}{2}\cos\phi \tag{5.10.26}$$

which if unity displacement factor is assumed leads to the maximum voltage transfer ratio being equal to 0.866, which is the well known value seen in space vector modulation when used for other converters.

5.11 Other Four-Leg Matrix Converter Modulation Methods

At the time when this work started, the four-leg matrix converter was a novel circuit for which there was no available prior work, and as such the work here is entirely based on the extension of the methods used for modulation in the 3x3 matrix converter and the four-leg inverter. However, after the first publication of the basic derivation [32] there have since been two other approaches to the problem of modulating the four-leg matrix, both of which have originated from the Power Electronic and Machine Control (PEMC) group at the University of Nottingham.

The first approach is an extension of the Alesina/Venturini optimum method [30], as it was perceived that the SVM based method described within the original work was too computationally intensive to be able to implement with any reasonably fast switching frequency. While this was correct with the equations given in the original paper [32], the simplification to the calculation of the duty cycles described in this chapter, and the further simplifications which will be described in Chapter 6 mean that it is believed that this is no longer the case.

With the difference in computational effort between the extended Alesina /Venturini optimum method and the method described in this work now being negligible, there is thought to be very little difference in performance between them, with the choice of which one to use being mostly down to personal preference.

The second, and possibly most interesting, of the alternative modulation methods is the one developed by Yue Fan [34, 35]. For this method it separates the two functional halves of the matrix converter, the rectifier and inverter, and deals with each separately, joining them by a 'fictitious DC link'. Figure 5.17 below shows a diagram of a 4-leg matrix converter using this technique, and as can be seen there is a significant difference to the circuit layout for this type of matrix converter as it looks a lot closer to a normal rectifier/inverter circuit. However, as can be seen, there is no DC link capacitor and so this is a type of direct converter just as the more standard 4-leg matrix converter used in this work, although due to the two-stage nature this is sometimes called an indirect converter.

While this looks very much like a standard rectifier/inverter setup, the link between the two is not actually a DC link as it varies depending on the switching of the rectifier stage, and this is where the 'fictitious DC link' name comes from. The main job of the rectifier stage in this circuit

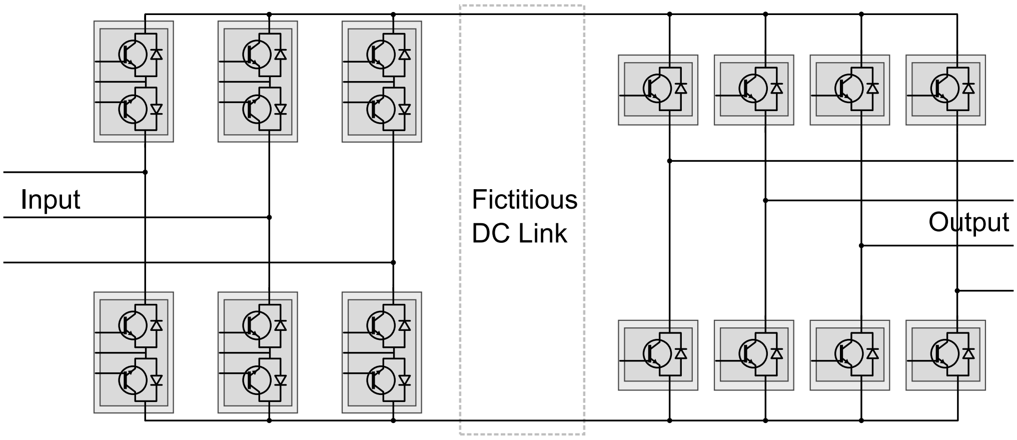


Figure 5.17: Circuit Diagram of the 4-Leg Matrix Converter using the 'Fictitious DC Link'

is to ensure that there is a positive voltage across the link, and to draw unity displacement input current.

The inverter in this circuit works just as a normal inverter does, but has the complexity of having to operate without a steady DC link voltage, and so this complicates the operation. However, by knowing the input voltages, and controlling the rectification stage the voltage across the link can be known at all times, and so the inverter switching adjusted accordingly.

This circuit has a number of advantages over the standard four-leg matrix converter, firstly the obvious one is that the number of switches in this circuit is reduced from the 24 required in the standard converter, to 20 in this one with it's related cost savings. The second advantage is that due to some further advances in inverter and rectifier modulation [] which allow the use of the demand voltages to define the duty cycles, it is possible to create a very simple method for calculating the duty cycles.

Apart from the different computational requirements between the three techniques discussed here, it is thought that, based on their 3-phase equivalents, all three techniques would give an equivalent output performance.

CHAPTER 6

Simulation of the 4-leg Matrix Converter

While the concept and theory of the equations for the 4-leg matrix converter derived in Chapter 5 appears to hold together in a coherent manner, with the resulting equations being relatively simple and holding a similar form to those found in the space vector modulation method for the 3x3 matrix converter, the equations still require validation to demonstrate that they are correct. This will be achieved via simulation.

It was decided that the simulation would proceed in several stages.

The initial stages were performed using Matlab in order to verify both the method of simulation and the basic equations and concepts of the 4-leg matrix converter. Initially the method of simulation was tested by implementing a standard 3x3 matrix converter. This was then followed by the implementation of a 4-leg inverter to prove the 3-dimensional output space vector, which, once working was then extended to be the full 4-leg matrix converter.

In order to verify the derivation of the Space Vector Modulation technique shown in Chapter 5 above, it was necessary to preform a series of simulations. The first simulations were performed using Matlab as this allowed a relatively quick way to check that the fundamental equations outlining the output behaviour of the converter were sound, and to show that part of the derivation to be correct. The limitation in the model that was used in for these simulations in Matlab was that it only looked at the output voltage and was also time-averaged, so giving the average voltage over a switching period at each of the output legs.

Once it had been shown that the basic equations governing the behaviour of the output voltage of the 4-leg matrix converter were sound, the simulation moved onto a more advanced stage using the Saber modelling and simulation software. This is a powerful software package that allowed the converter to be defined as a set of bi-directional switches all controlled by a software block implementing the equations for the 4-leg matrix converter. The Saber package allows for very accurate simulations of both the voltage and the current within the circuit, which allowed an examination of both the input and output, voltage and current.

6.1 Simulation using Matlab

The initial simulation work was done using the Matlab software package, as this would be a relatively quick and easy way to implement the equations derived in Chapter 5, allowing them to be checked that they were valid and that the conclusions were correct.

6.1.1 4-Leg Inverter

The initial stages of the simulation work was taken up by proving that the chosen simulation method was correct, where the output voltage is time averaged over a switching period. This was done by simulating a 3x3 matrix converter in a number of different methods, and checking that the results for this chosen method were consistent. As the 3x3 matrix converter has been extensively simulated the results for this work will not be included here, and so the first simulation covered here is an implementation of a 4-leg inverter. This would allow the basics of the 3-dimensional SVM method as defined in Section 2.1 above to be tested. Checking this would be especially important as it would ensure that the code which is used for selecting the required output tetrahedron is correct, and that the method of calculating the duty cycles for the space vectors is also correct. These need to be shown to work correctly are they are both required for the simulations of the 4-leg matrix converter.

There are several methods which can be used for calculating the duty cycles for the 4-leg inverter, Zhang[14] uses a simple lookup table approach, however due to the more complex nature of the calculations required for the 4-leg matrix converter, it was felt that the more general approach derived in Chapter 5 should ultimately be used here. However, the look-up table approach would also be implemented to allow the general method to be proved prior to using it within the 4-leg matrix converter simulation.

The implementation for the simulation using the both of these methods is a relatively simple task within Matlab, with the equations being easily written out into a series of Matlab m-files, with Figure 6.1 showing the major steps required.

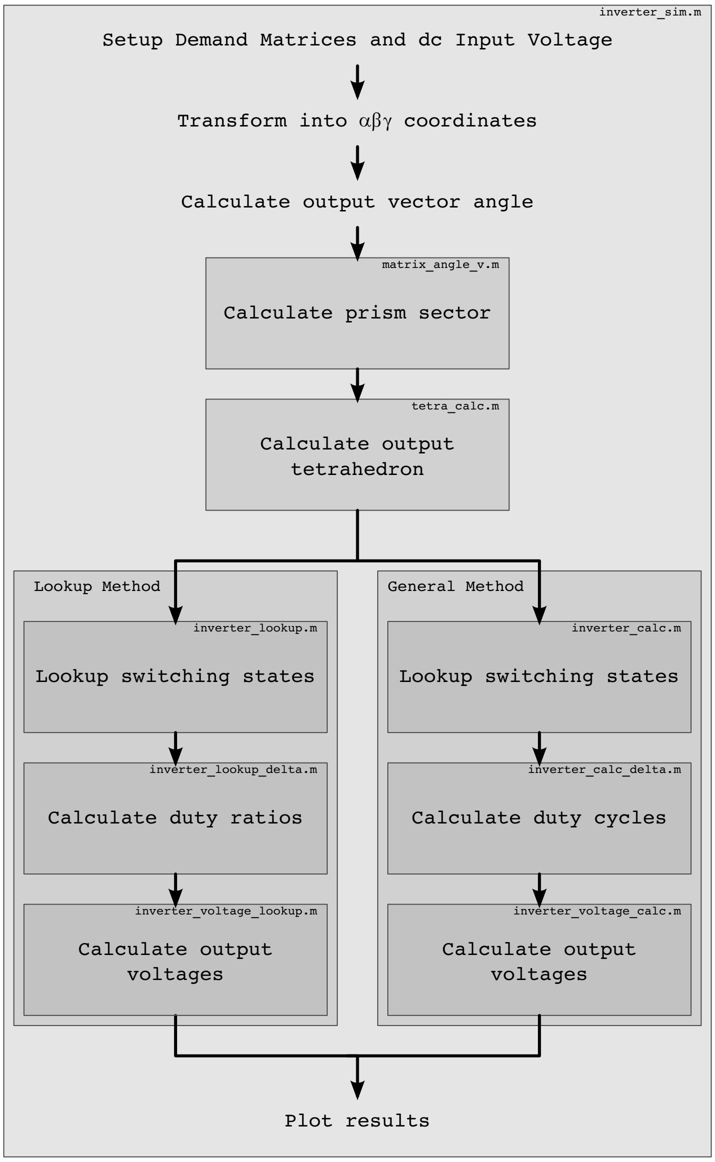


Figure 6.1: Block Diagram for the Matlab simulation for the 4-Leg Inverter

This being a converter with 4 output legs, instead of the 3 for the 3x3 matrix, it might be expected that the simulation setup of the demand voltages would require 4 different demand voltages. However, in practice it is either the output line voltage demands, V_{ab}, V_{bc} and V_{ca}, or the phase voltage demands, V_a, V_b and V_c, that are used. The value of V_n in these situations is generally not specified as this is not generally of interest when setting the demands, with it’s value being defined by the other three voltages.

So, the output voltage demand for the 4-leg inverter simulation can be defined as

$$V_{dem_{line}} = \begin{bmatrix} V_{dem_{ab(0)}} & V_{dem_{ab(1)}} & \cdots & V_{dem_{ab(t-1)}} & V_{dem_{ab(t)}} \\ V_{dem_{bc(0)}} & V_{dem_{bc(1)}} & \cdots & V_{dem_{bc(t-1)}} & V_{dem_{bc(t)}} \\ V_{dem_{ca(0)}} & V_{dem_{ca(1)}} & \cdots & V_{dem_{ca(t-1)}} & V_{dem_{ca(t)}} \end{bmatrix} \tag{6.1.1}$$

such that

$$\begin{aligned} V_{dem_{ab(t)}} &= \sqrt{2}V_{dem}\cos(2\pi f_{dem}t) \\ V_{dem_{bc(t)}} &= \sqrt{2}V_{dem}\cos(2\pi f_{dem}t - \frac{2\pi}{3}) \\ V_{dem_{ca(t)}} &= \sqrt{2}V_{dem}\cos(2\pi f_{dem}t - \frac{4\pi}{3}) \end{aligned} \tag{6.1.2}$$

where V_{dem} is the line-line voltage and f_{dem} is the frequency for the desired output. These line-neutral demand voltages are then converted to being the relevant phase voltages as

$$V_{dem_{phase}} = \begin{bmatrix} V_{dem_{a(0)}} & V_{dem_{a(1)}} & \cdots & V_{dem_{a(t-1)}} & V_{dem_{a(t)}} \\ V_{dem_{b(0)}} & V_{dem_{b(1)}} & \cdots & V_{dem_{b(t-1)}} & V_{dem_{b(t)}} \\ V_{dem_{c(0)}} & V_{dem_{c(1)}} & \cdots & V_{dem_{c(t-1)}} & V_{dem_{c(t)}} \end{bmatrix} \tag{6.1.3}$$

using

$$\begin{aligned} V_{dem_{a(t)}} &= \frac{1}{3}(2V_{dem_{ab(t)}} + V_{dem_{bc}}) \\ V_{dem_{b(t)}} &= V_{dem_{a(t)}} - V_{dem_{ab(t)}} \\ V_{dem_{c(t)}} &= -V_{dem_{a(t)}} - V_{dem_{b(t)}} \end{aligned} \tag{6.1.4}$$

Alternatively, the phase voltages can be defined directly as

$$V_{dem_{phase}} = \begin{bmatrix} V_{dem_{a(0)}} & V_{dem_{a(1)}} & \cdots & V_{dem_{a(t-1)}} & V_{dem_{a(t)}} \\ V_{dem_{b(0)}} & V_{dem_{b(1)}} & \cdots & V_{dem_{b(t-1)}} & V_{dem_{b(t)}} \\ V_{dem_{c(0)}} & V_{dem_{c(1)}} & \cdots & V_{dem_{c(t-1)}} & V_{dem_{c(t)}} \end{bmatrix} \tag{6.1.5}$$

using

$$\begin{aligned}
V_{dem_{a(t)}} &= \sqrt{2}V_{dem}\cos(2\pi f_{dem}t) \\
V_{dem_{b(t)}} &= \sqrt{2}V_{dem}\cos(2\pi f_{dem}t - \frac{2\pi}{3}) \\
V_{dem_{c(t)}} &= \sqrt{2}V_{dem}\cos(2\pi f_{dem}t - \frac{4\pi}{3})
\end{aligned} \tag{6.1.6}$$

where in this case, V_{dem} is the RMS phase voltage.

The demand vector is then transformed into the $\alpha\beta\gamma$ space using

$$\begin{bmatrix} V_{\alpha(t)} \\ V_{\beta(t)} \\ V_{\gamma(t)} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \end{bmatrix}\begin{bmatrix} V_{dem_{a(t)}} \\ V_{dem_{b(t)}} \\ V_{dem_{c(t)}} \end{bmatrix} \tag{6.1.7}$$

The next step is to identify which sector, or in this case as the sector is a volume in $\alpha\beta\gamma$ space, tetrahedron, the output demand vector lies in, and the method for doing this is as described in Section 2.1. It by first calculating the angle that the demand vector would project on the $\alpha\beta$ plane using

$$\theta = \arctan\left(\frac{V_\beta}{V_\alpha}\right) \tag{6.1.8}$$

and then selecting the Prism using

$$Prism_o = \begin{cases} 1 & \text{if} \quad 0 \leq \theta_o < \frac{\pi}{3} \\ 2 & \text{if} \quad \frac{\pi}{3} \leq \theta_o < \frac{2\pi}{3} \\ 3 & \text{if} \quad \frac{2\pi}{3} \leq \theta_o < \pi \\ 4 & \text{if} \quad \pi \leq \theta_o < \frac{4\pi}{3} \\ 5 & \text{if} \quad \frac{4\pi}{3} \leq \theta_o < \frac{5\pi}{3} \\ 6 & \text{if} \quad \frac{5\pi}{3} \leq \theta_o < 2\pi \end{cases} \tag{6.1.9}$$

The sector in $\alpha\beta$ space identifies which of the 6 prisms in $\alpha\beta\gamma$ space contains the demand vector. The difficulty now lies in algorithmically finding a way to tell which of the 4 tetrahedrons, which go to make up each prism, is the one containing the demand vector. Fortunately there is a simple technique found by Zhang[14] where the correct tetrahedron is selected by matching the signs of the output line-neutral voltages to the signs of the three vectors which make up the vertices of the tetrahedron in the γ direction. Figure 6.2 below shows an exploded view of one of the six prisms which make up the output voltage space. Looking at the vertices for each of these tetrahedrons it can be seen that each one has a different distribution of signs in the γ direction, and a similar distribution carries through on each of the other 5 prisms.

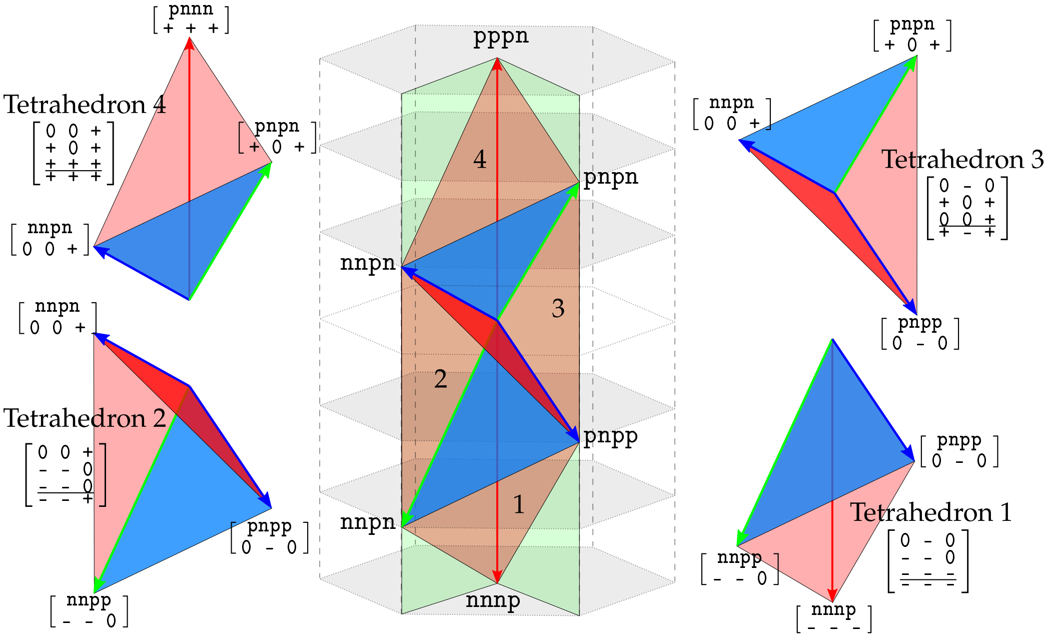


Figure 6.2: Exploded view of a single prism, showing the 4 tetrahedrons

Taking the example in equation 5.3 above where we know that this demand vector falls within prism number 6, now, looking at the signs of the demand vector, there are 2 positive and 1 negative demand line-neutral voltages. Comparing that to the prisms shown in Figure 6.2 it can be seen that the tetrahedron 2 matches the signs, and so this is the tetrahedron which the demand vector falls within, with Figure6.3 showing this demand vector plotted into the output voltage space, and it can be seen that it does indeed fall within tetrahedron 2, showing that it this is correct.

The simplest way to perform this in the simulation is to count how many of the phase voltages which make up the demand voltage are positive using

```
sign = 1;

if (Vdemand(1) >= 0)
    sign = sign + 1;
end

if (Vdemand(2) >= 0)
    sign = sign + 1;
```

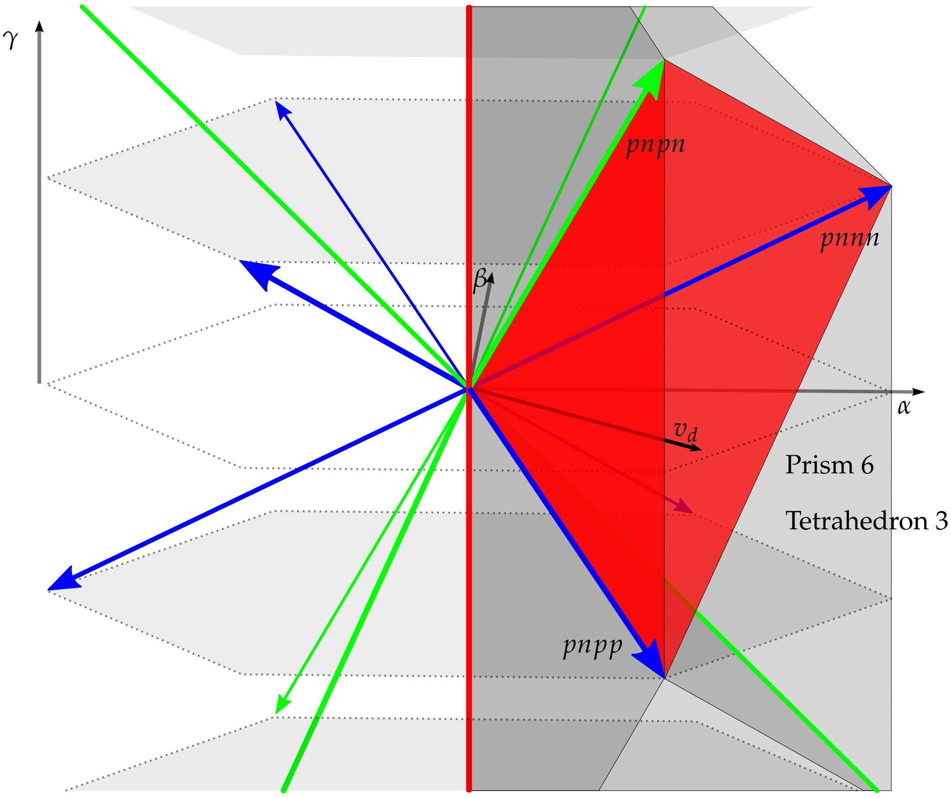


Figure 6.3: Closeup view of a single prism, showing the Demand Vector

```
end

if (Vdemand(3) >= 0)
    sign = sign + 1;
end
```

This is the point where the more general method that is going to be used for the calculation of the duty cycles in the 4-leg matrix converter differs from this method, in that the information that it gathers from the lookup table does not include the duty cycle matrix. For the more general method this matrix is calculated on the fly during the simulation.

6.1.1.1 Lookup Table Method for Calculating Duty Cycles

The output from the code above gives us the tetrahedron number which can then be used, alongside the prism number, to look up the required switching states, and the information used to calculate the duty cycles, using Table 6.1.

Taking the matrix defined within the table, the duty cycles are then worked

Table 6.1: Switching state and Duty-Cycle selection for the 4-Leg Inverter

		Output Voltage Sector			
		Tetrahedron 1	Tetrahedron 2	Tetrahedron 3	Tetrahedron 4
		I,II,III	I,II,III	I,II,III	I,II,III
Prism	1	11,6,7 $\begin{bmatrix} 3/2 & -\sqrt{3}/2 & 0 \\ 0 & \sqrt{3} & 0 \\ -1 & 0 & -1 \end{bmatrix}$	1,11,6 $\begin{bmatrix} 1 & 0 & 1 \\ 1/2 & -\sqrt{3}/2 & -1 \\ 0 & \sqrt{3} & 0 \end{bmatrix}$	1,9,6 $\begin{bmatrix} 3/2 & -\sqrt{3}/2 & 0 \\ -1/2 & \sqrt{3}/2 & 1 \\ 1/2 & \sqrt{3}/2 & -1 \end{bmatrix}$	1,9,8 $\begin{bmatrix} 3/2 & -\sqrt{3}/2 & 0 \\ 0 & \sqrt{3} & 0 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix}$
	2	6,14,7 $\begin{bmatrix} 3/2 & \sqrt{3}/2 & 0 \\ -3/2 & \sqrt{3}/2 & 0 \\ 1/2 & -\sqrt{3}/2 & -1 \end{bmatrix}$	9,6,3 $\begin{bmatrix} 1 & 0 & 1 \\ 1/2 & \sqrt{3}/2 & -1 \\ -3/2 & \sqrt{3}/2 & 0 \end{bmatrix}$	6,3,14 $\begin{bmatrix} 3/2 & \sqrt{3}/2 & 0 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1 & 0 & -1 \end{bmatrix}$	9,3,8 $\begin{bmatrix} 3/2 & \sqrt{3}/2 & 0 \\ -3/2 & \sqrt{3}/2 & 0 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix}$
	3	14,2,7 $\begin{bmatrix} 0 & \sqrt{3} & 0 \\ -3/2 & -\sqrt{3}/2 & 0 \\ 1/2 & -\sqrt{3}/2 & -1 \end{bmatrix}$	3,14,2 $\begin{bmatrix} -1/2 & \sqrt{3}/2 & 1 \\ 1/2 & \sqrt{3}/2 & -1 \\ -3/2 & -\sqrt{3}/2 & 0 \end{bmatrix}$	3,12,2 $\begin{bmatrix} 0 & \sqrt{3} & 0 \\ -1/2 & -\sqrt{3}/2 & 1 \\ -1 & 0 & -1 \end{bmatrix}$	3,12,8 $\begin{bmatrix} 0 & \sqrt{3} & 0 \\ -3/2 & -\sqrt{3}/2 & 0 \\ 1 & 0 & 1 \end{bmatrix}$
	4	2,10,7 $\begin{bmatrix} -3/2 & \sqrt{3}/2 & 0 \\ 0 & -\sqrt{3} & 0 \\ 1/2 & \sqrt{3}/2 & -1 \end{bmatrix}$	12,2,5 $\begin{bmatrix} -1/2 & \sqrt{3}/2 & 1 \\ -1 & 0 & -1 \\ 0 & -\sqrt{3} & 0 \end{bmatrix}$	2,5,10 $\begin{bmatrix} -3/2 & \sqrt{3}/2 & 0 \\ -1/2 & -\sqrt{3}/2 & 1 \\ 1/2 & -\sqrt{3}/2 & -1 \end{bmatrix}$	12,5,8 $\begin{bmatrix} -3/2 & \sqrt{3}/2 & 0 \\ 0 & -\sqrt{3} & 0 \\ 1 & 0 & 1 \end{bmatrix}$
	5	10,4,7 $\begin{bmatrix} -3/2 & -\sqrt{3}/2 & 0 \\ 3/2 & -\sqrt{3}/2 & 0 \\ 1/2 & \sqrt{3}/2 & -1 \end{bmatrix}$	5,10,4 $\begin{bmatrix} -1/2 & -\sqrt{3}/2 & 1 \\ -1 & 0 & -1 \\ 3/2 & -\sqrt{3}/2 & 0 \end{bmatrix}$	5,13,4 $\begin{bmatrix} -3/2 & -\sqrt{3}/2 & 0 \\ 1 & 0 & 1 \\ 1/2 & -\sqrt{3}/2 & -1 \end{bmatrix}$	5,13,8 $\begin{bmatrix} -3/2 & -\sqrt{3}/2 & 0 \\ 3/2 & -\sqrt{3}/2 & 0 \\ -1/2 & \sqrt{3}/2 & 1 \end{bmatrix}$
	6	4,11,7 $\begin{bmatrix} 0 & -\sqrt{3} & 0 \\ 3/2 & \sqrt{3}/2 & 0 \\ -1 & 0 & -1 \end{bmatrix}$	13,4,1 $\begin{bmatrix} -1/2 & -\sqrt{3}/2 & 1 \\ 1/2 & -\sqrt{3}/2 & -1 \\ 3/2 & -\sqrt{3}/2 & 0 \end{bmatrix}$	4,1,11 $\begin{bmatrix} 0 & -\sqrt{3} & 0 \\ 1 & 0 & 1 \\ 1/2 & \sqrt{3}/2 & -1 \end{bmatrix}$	13,1,8 $\begin{bmatrix} 0 & -\sqrt{3} & 0 \\ 3/2 & \sqrt{3}/2 & 0 \\ -1/2 & \sqrt{3}/2 & 1 \end{bmatrix}$

out directly using

$$\begin{bmatrix} \delta^{I}_{(t)} \\ \delta^{II}_{(t)} \\ \delta^{III}_{(t)} \end{bmatrix} = \frac{1}{V_{dc}} \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \\ \frac{1}{2} & -\frac{\sqrt{3}}{2} & -1 \\ \frac{3}{2} & \frac{\sqrt{3}}{2} & 0 \end{bmatrix} \begin{bmatrix} V_{\alpha(t)} \\ V_{\beta(t)} \\ V_{\gamma(t)} \end{bmatrix} \tag{6.1.10}$$

Now that both the duty cycles and switching states are known, the output voltages at each time point can be calculated using

$$V_o = \begin{bmatrix} V_{a(t)} \\ V_{b(t)} \\ V_{c(t)} \\ V_{n(t)} \end{bmatrix} = \mathbf{V_{I(t)}}.\delta^{I}_{(t)} + \mathbf{V_{II(t)}}.\delta^{II}_{(t)} + \mathbf{V_{III(t)}}.\delta^{III}_{(t)} \tag{6.1.11}$$

where $\mathbf{V_x}$ are the voltages at the output legs for the respective switching states taken from Table 6.2.

Table 6.2: Switching States for a 4-Leg Inverter

	Switching	Output Leg Switching				L-N Output Voltage		
	State	Leg a	Leg b	Leg c	Leg n	V$_{an}$	V$_{bn}$	V$_{cn}$
Zero Vectors	0_A	1	1	1	1	0	0	0
	0_C	0	0	0	0	0	0	0
Vectors	1	1	0	0	0	V$_{dc}$	0	0
	2	0	1	1	1	-V$_{dc}$	0	0
	3	0	1	0	0	0	V$_{dc}$	0
	4	1	0	1	1	0	-V$_{dc}$	0
	5	0	0	1	0	0	0	V$_{dc}$
	6	1	1	0	1	0	0	-V$_{dc}$
	7	0	0	0	1	-V$_{dc}$	-V$_{dc}$	-V$_{dc}$
	8	1	1	1	0	V$_{dc}$	V$_{dc}$	V$_{dc}$
	9	1	1	0	0	V$_{dc}$	V$_{dc}$	0
	10	0	0	1	1	-V$_{dc}$	-V$_{dc}$	0
	11	1	0	0	1	0	-V$_{dc}$	-V$_{dc}$
	12	0	1	1	0	0	V$_{dc}$	V$_{dc}$
	13	1	0	1	0	V$_{dc}$	0	V$_{dc}$
	14	0	1	0	1	-V$_{dc}$	0	-V$_{dc}$

The results from this calculation being saved into the matrix shown below

$$V_{out_{phase}} = \begin{bmatrix} V_{out_{an(0)}} & V_{out_{an(1)}} & .. & V_{out_{an(t-1)}} & V_{out_{an(t)}} \\ V_{out_{bn(0)}} & V_{out_{bn(1)}} & .. & V_{out_{bn(t-1)}} & V_{out_{bn(t)}} \\ V_{out_{cn(0)}} & V_{out_{cn(1)}} & .. & V_{out_{cn(t-1)}} & V_{out_{cn(t)}} \end{bmatrix} \tag{6.1.12}$$

where

$$\begin{aligned} V_{out_{an(t)}} &= V_{out_{a(t)}} - V_{out_{n(t)}} \\ V_{out_{bn(t)}} &= V_{out_{b(t)}} - V_{out_{n(t)}} \\ V_{out_{cn(t)}} &= V_{out_{c(t)}} - V_{out_{n(t)}} \end{aligned} \tag{6.1.13}$$

If required, the inverter line output voltages can be calculated using

$$V_{out_{line}} = \begin{bmatrix} V_{out_{ab(0)}} & V_{out_{ab(1)}} & .. & V_{out_{ab(t-1)}} & V_{out_{ab(t)}} \\ V_{out_{bc(0)}} & V_{out_{bc(1)}} & .. & V_{out_{bc(t-1)}} & V_{out_{bc(t)}} \\ V_{out_{ca(0)}} & V_{out_{ca(1)}} & .. & V_{out_{ca(t-1)}} & V_{out_{ca(t)}} \end{bmatrix} \tag{6.1.14}$$

where

$$\begin{aligned} V_{out_{ab(t)}} &= V_{out_{an(t)}} - V_{out_{bn(t)}} \\ V_{out_{bc(t)}} &= V_{out_{bn(t)}} - V_{out_{cn(t)}} \\ V_{out_{ca(t)}} &= V_{out_{cn(t)}} - V_{out_{an(t)}} \end{aligned} \tag{6.1.15}$$

The calculated output phase voltage matrix can then be easily plotted against time, and compared to the expected values defined by $V_{dem_{a(t)}}$, $V_{dem_{b(t)}}$ and $V_{dem_{c(t)}}$ from equation 6.1.2. The results from a simulation run can be seen in Figure 6.4, where the calculated output from the converter is using the + marker, and the expected output is shown with the o marker. From this it is simple to see that the two sets of outputs are matching.

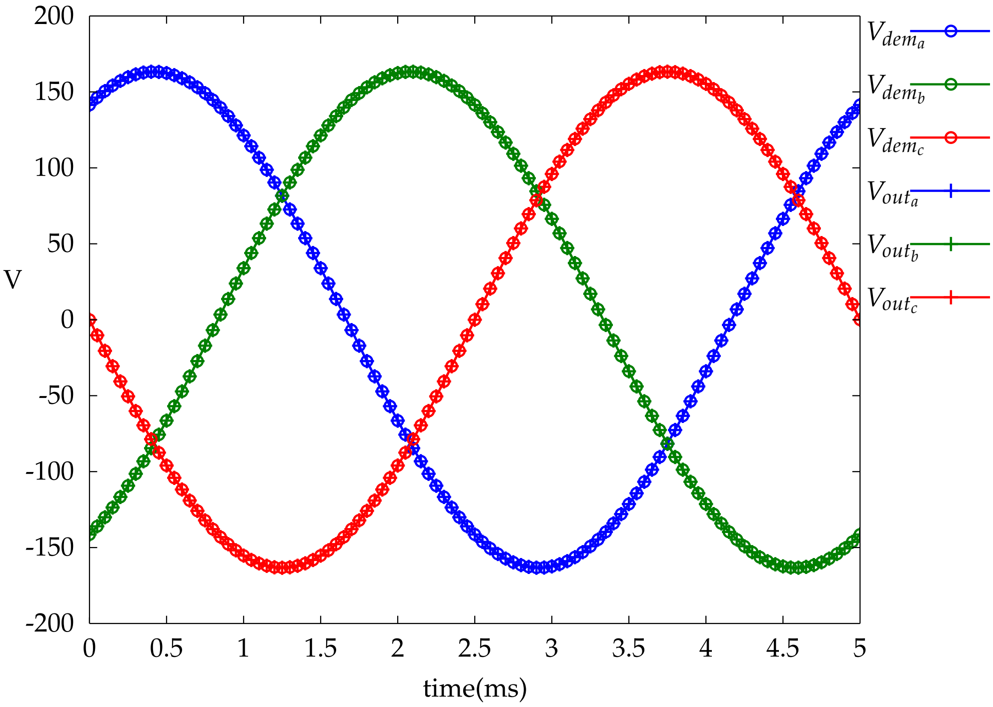


Figure 6.4: Output from the 4-Leg Inverter Simulation, showing the Calculated and Demand Output Voltages

In Figure 6.5 it shows how the selected tetrahedron and prism change with vector angle.

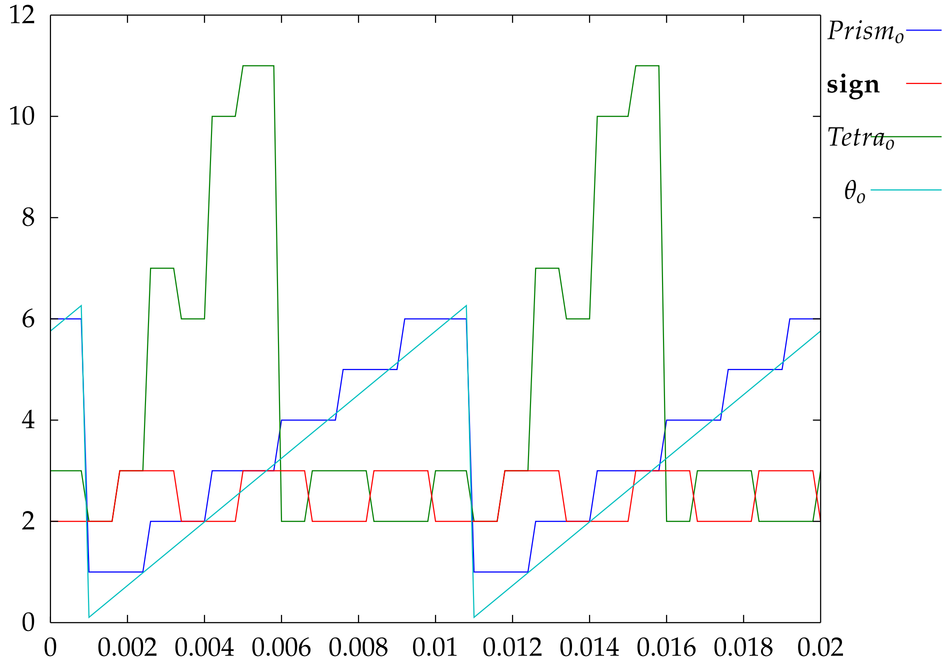


Figure 6.5: Output Voltage Vector angle, Tetrahedron and Prism for the 4-Leg Inverter Simulation

The results demonstrated in these figures show that the method used to implement the 3-dimensional space vector modulation for the 4-leg inverter is correct. The next step is to implement the more general method of calculating the duty cycles, which will be used with the 4-leg matrix converter, and ensure that the outputs from both methods match.

6.1.1.2 General Method for Calculating Duty Cycles

With the prism and tetrahedron identified, the next step in calculating the duty cycles via this method is to calculate the length of the space vectors at the vertices. This is examined in Section 5.9, and while the derivation of the lengths of the space vectors $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$ shown is correct, and this same method is shared with the inverter, it does not lend itself to straightforward calculation. This means that a modification is required for the program to be as efficient as possible, and this is described below.

As the directions of all three output vectors is known, we can define a plane which contains any two of them and the origin Ov_2v_3 such that

Table 6.3: Switching state and Duty-Cycle selection for the 4-Leg Inverter

		Output Voltage Sector			
		Tetrahedron 1	Tetrahedron 2	Tetrahedron 3	Tetrahedron 4
		I,II,III	I,II,III	I,II,III	I,II,III
Prism	1	1 9 13	8 9 13	8 12 13	8 12 14
	2	1 5 13	4 5 13	4 12 13	4 12 14
	3	1 5 7	4 5 7	4 6 7	4 6 14
	4	1 3 7	2 3 7	2 6 7	2 6 14
	5	1 3 11	2 3 11	2 10 11	2 10 14
	6	1 9 11	8 9 11	8 10 11	8 10 14]

$$\vec{n_{23}} \cdot (P) = 0 \tag{6.1.16}$$

where $\vec{n_{23}}$ is the normal vector to the plane Ov_2v_3 and P is any point on the plane.

Then, using the same method as shown in Section 5.9, v_o', is then the length of the vector which intersects both the demand voltage $\vec{v_d}$ and the plane Ov_2v_3 in the direction $\vec{v_1}$. This then gives the equation

$$\vec{n_{23}} \cdot (\vec{v_d} - v_o'\vec{v_1}) = 0 \tag{6.1.17}$$

and re-arranging then gives

$$v_o' \left(\vec{n_{23}} \cdot \vec{v_1}\right) = \vec{n_{23}} \cdot \vec{v_d} \tag{6.1.18}$$

Using the equation for the dot product (5.9.8) on the left hand side then gives

$$v_o'|\vec{n_{23}}||\vec{v_1}|\cos\mu_1 = \vec{n_{23}} \cdot \vec{v_d} \tag{6.1.19}$$

And as stated earlier that

$$|\vec{v_1}|\cos\mu_1 = \frac{1}{\sqrt{3}} \tag{6.1.20}$$

Substituting that back into equation (6.1.19) and re-arranging gives

$$v_o'.|\vec{n_{23}}| = \sqrt{3}\vec{n_{23}} \cdot \vec{v_d} \tag{6.1.21}$$

Now, the vector $\vec{n_{23}}$ is the normal vector to the plane Ov_2v_3 and is defined by

$$\vec{n_{23}} = \vec{v_2} \times \vec{v_3} \tag{6.1.22}$$

which in turn is defined by

$$\vec{v_2} \times \vec{v_3} = \begin{bmatrix} \left(v_{2_\beta}v_{3_\gamma} - v_{2_\gamma}v_{3_\beta}\right) \\ \left(v_{1_\alpha}v_{3_\gamma} - v_{1_\gamma}v_{3_\alpha}\right) \\ \left(v_{1_\alpha}v_{2_\beta} - v_{1_\beta}v_{2_\alpha}\right) \end{bmatrix} \tag{6.1.23}$$

Now, for any single tetrahedron, there will be 3 space vectors which make up its vertices, using the example from Chapter 5 once again these would be $\vec{V8}$, $\vec{V10}$ and $\vec{V11}$, and these are defined as

$$\vec{V8} = \begin{bmatrix} \frac{2}{3} \\ 0 \\ \frac{1}{3\sqrt{2}} \end{bmatrix} \quad \vec{V10} = \begin{bmatrix} \frac{1}{3} \\ -\frac{1}{\sqrt{3}} \\ \frac{2}{3\sqrt{2}} \end{bmatrix} \quad \vec{V11} = \begin{bmatrix} \frac{1}{3} \\ -\frac{1}{\sqrt{3}} \\ -\frac{1}{3\sqrt{2}} \end{bmatrix} \tag{6.1.24}$$

Calculating the vector product (6.1.23) on the three combinations of these vectors which are used in this method, and then calculating the length of each vector gives the result that

$$|\vec{n_{xy}}| = \frac{\sqrt{2}}{3} \tag{6.1.25}$$

So, the length of the normal vector to the plane, irrespective of which 2 of the 3 vectors are used to define it, is always the same length. Looking at sim: tetrahedron vectors it can be seen that of the 3 vectors, $\vec{V8}$ and $\vec{11}$ are the same length$\left(\frac{1}{sqrt2}\right)$, while $\vec{10}$ is longer$\left(\sqrt{\frac{2}{3}}\right)$. It can then be shown that unlike the 6 space vectors which describe the $\alpha\beta$ plane in 2-dimensional SVM, the 14 space vectors used in 3-dimensional SVM are not all equidistant from each other, and in fact this must be true for the result in (6.1.25) to stand. It is in fact the case that there are two different length groups of vectors, 8 shorter vectors and 6 longer vectors, and that every tetrahedron is defined by 2 short and 1 long vector, and so the result shown in (6.1.25) holds true at all times.

So, substituting (6.1.25) into (6.1.28) now gives

$$v'_o = \frac{3\sqrt{3}}{\sqrt{2}} \vec{n_{23}} \cdot \vec{v_d} \tag{6.1.26}$$

As both the vectors $\vec{n_{23}}$ and $\vec{v_d}$ are known, then the value of their dot product can be easily calculated using

$$\vec{n_{23}} \cdot \vec{v_d} = n_{23_\alpha}.v_{d_\alpha} + n_{23_\beta}.v_{d_\beta} + n_{23_\gamma}.v_{d_\gamma} \tag{6.1.27}$$

which is a straightforward calculation.

The duty cycle for the switching state which then produces that vector is calculated by dividing the result by the input voltage, in this case the dc link voltage, giving the overall equation as

$$\delta' = \frac{3\sqrt{3}}{v_{dc}\sqrt{2}} \vec{n_{23}} \cdot \vec{v_d} \tag{6.1.28}$$

with the other duty cycles defined as

$$\delta'' = \frac{3\sqrt{3}}{v_{dc}\sqrt{2}} \vec{n_{13}} \cdot \vec{v_d} \tag{6.1.29}$$

$$\delta''' = \frac{3\sqrt{3}}{v_{dc}\sqrt{2}} \vec{n_{12}} \cdot \vec{v_d} \tag{6.1.30}$$

While it would be perfectly possible, and correct, to calculate the value of v'_o, and then δ' directly from (6.1.18), by dividing the two dot products, the simplification was performed with a thought towards how the calculations would be implemented within a DSP, in which case the number of divisions should be kept to a minimum. With the equations in (6.1.28) and (6.1.29), only a single division needs to be performed, to calculate the value of $\frac{1}{v_{dc}}$, which is then used in each calculation.

Now, with all the duty cycles now defined, all that is required is to perform a similar set of calculations as for the lookup method, and to produce the output phase voltage matrix

$$V'_{out_{phase}} = \begin{bmatrix} V'_{out_{an(0)}} & V'_{out_{an(1)}} & .. & V'_{out_{an(t-1)}} & V'_{out_{en(t)}} \\ V'out_{bn(0)} & V'_{out_{bn(1)}} & .. & V'_{out_{bn(t-1)}} & V'_{out_{bn(t)}} \\ V'_{out_{cn(0)}} & V'_{out_{cn(1)}} & .. & V'_{out_{cn(t-1)}} & V'_{out_{cn(t)}} \end{bmatrix} \tag{6.1.31}$$

where

$$\begin{aligned}
V'_{out_{an(t)}} &= V'_{out_{a(t)}} - V'_{out_{n(t)}} \\
V'_{out_{bn(t)}} &= V'_{out_{b(t)}} - V'_{out_{n(t)}} \\
V'_{out_{cn(t)}} &= V'_{out_{c(t)}} - V'_{out_{n(t)}}
\end{aligned} \tag{6.1.32}$$

and the inverter line output voltages can be calculated using

$$V'_{out_{line}} = \begin{bmatrix} V'_{out_{ab(0)}} & V'_{out_{ab(1)}} & .. & V'_{out_{ab(t-1)}} & V'_{out_{ab(t)}} \\ V'_{out_{bc(0)}} & V'_{out_{bc(1)}} & .. & V'_{out_{bc(t-1)}} & V'_{out_{bc(t)}} \\ V'_{out_{ca(0)}} & V'_{out_{ca(1)}} & .. & V'_{out_{ca(t-1)}} & V'_{out_{ca(t)}} \end{bmatrix} \tag{6.1.33}$$

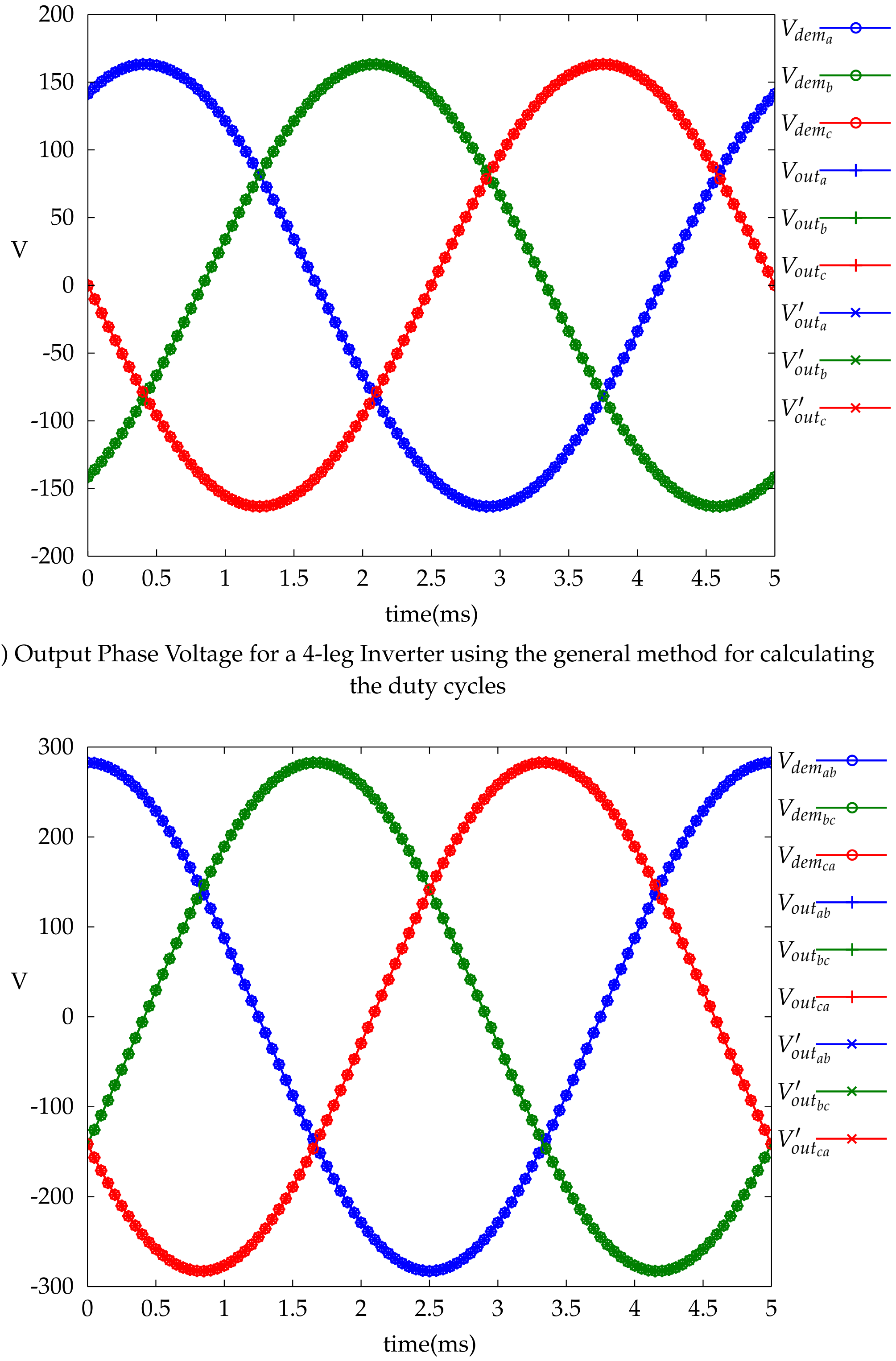
where

$$\begin{aligned}
V'_{out_{ab(t)}} &= V'_{out_{an(t)}} - V'_{out_{bn(t)}} \\
V'_{out_{bc(t)}} &= V'_{out_{bn(t)}} - V'_{out_{cn(t)}} \\
V'_{out_{ca(t)}} &= V'_{out_{cn(t)}} - V'_{out_{an(t)}}
\end{aligned} \tag{6.1.34}$$

6.1.1.3 Matlab 4-Leg Inverter Simulation Results

As before, the results from a simulation, where the demand voltage is 200V and the demand frequency is 200Hz, were plotted to compare the results with the demand voltages, $V_{dem_{an(t)}}$, $V_{dem_{bn(t)}}$ and $V_{dem_{cn(t)}}$, along with the results from the first simulation using the lookup table method, $V_{out_{an(t)}}$, $V_{out_{bn(t)}}$ and $V_{out_{cn(t)}}$, and the results shown in Figure 6.6. As can be seen, the results from both types of simulation match the demand voltages.

However, while the output voltages from both simulation methods were found to be the identical, it was discovered that there was a difference between the two methods when comparing the duty cycles. Plots of the two sets of duty cycles are shown in Figure 6.7, where it can be seen that there are jumps in the duty cycle values when using the lookup method, whereas the duty cycle values always appeared to changed smoothly for the general method. When comparing the actual sets of values for both methods, it was easily seen that both sets of duty cycles included the same three values, but they were just in held in different orders. This

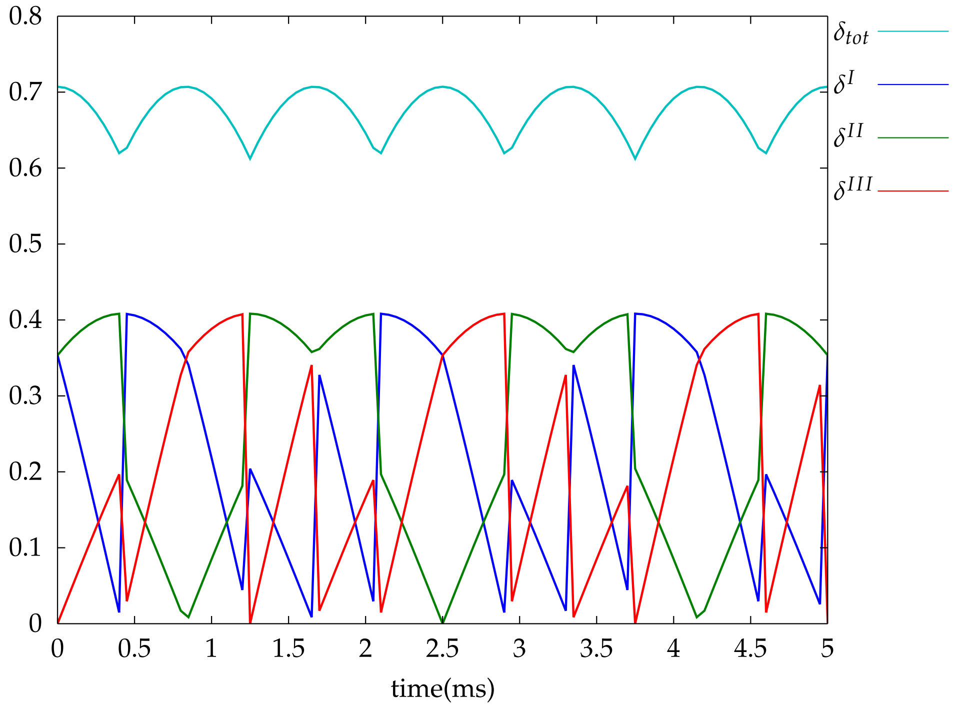


(a) Output Phase Voltage for a 4-leg Inverter using the general method for calculating the duty cycles

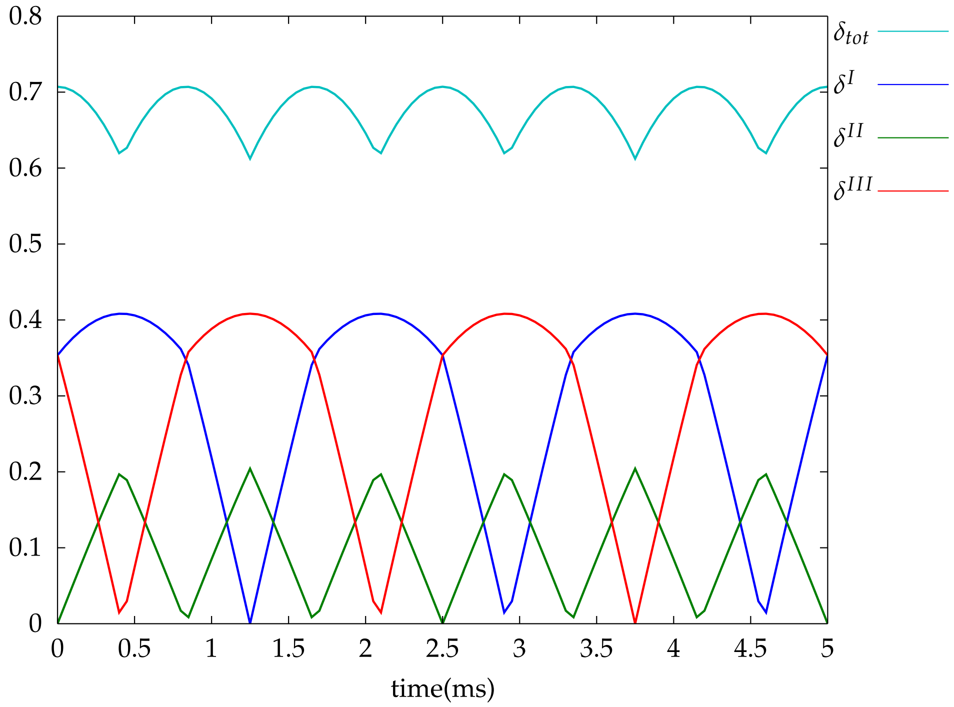
(b) Output Voltage Vector angle, tetrahedron and prism

Figure 6.6: Line and Phase Output Voltages from the 4-Leg Inverter Simulation, showing the results from the Lookup and General Method

is entirely due to the way that the order that the switching states are



(a) Output Phase Voltage for a 4-leg Inverter using the general method for calculating the duty cycles



(b) Duty Cycles for the General method

Figure 6.7: Duty cycle plots for the Lookup and General methods of 4-Leg Inverter Simulation

selected is different between the two methods, and that if these could be matched, the duty cycle output would also be matched.

The results from these simulations demonstrate that the 4-leg inverter is capable of generating a balanced 3-phase set of voltages at the outputs, both when referenced to the neutral leg of the inverter as a phase voltage and also referenced to the other output legs as a line voltage. However, as this is a balanced set of voltages, which means that when the demand voltages are transformed into the $\alpha\beta\gamma$ space, they produce a vector that rotates evenly around the origin in the $\alpha\beta$ plane. Because the demand vector stays in the $\alpha\beta$ plane this means that γ stays at zero for the entire simulation. For a γ value to be generated in the demand vector, the demand needs to be unbalanced. In normal use this would occur when the inverter was supplying a non-linear or unbalanced load when the inverter would need to be able to handle the neutral current, but as the simulation for this inverter is not investigating the current flow, another way needs to be found to introduce γ into the simulations. The easiest way to do this is to create an unbalanced demand vector such that

$$V_{an} + V_{bn} + V_{cn} \neq 0 \tag{6.1.35}$$

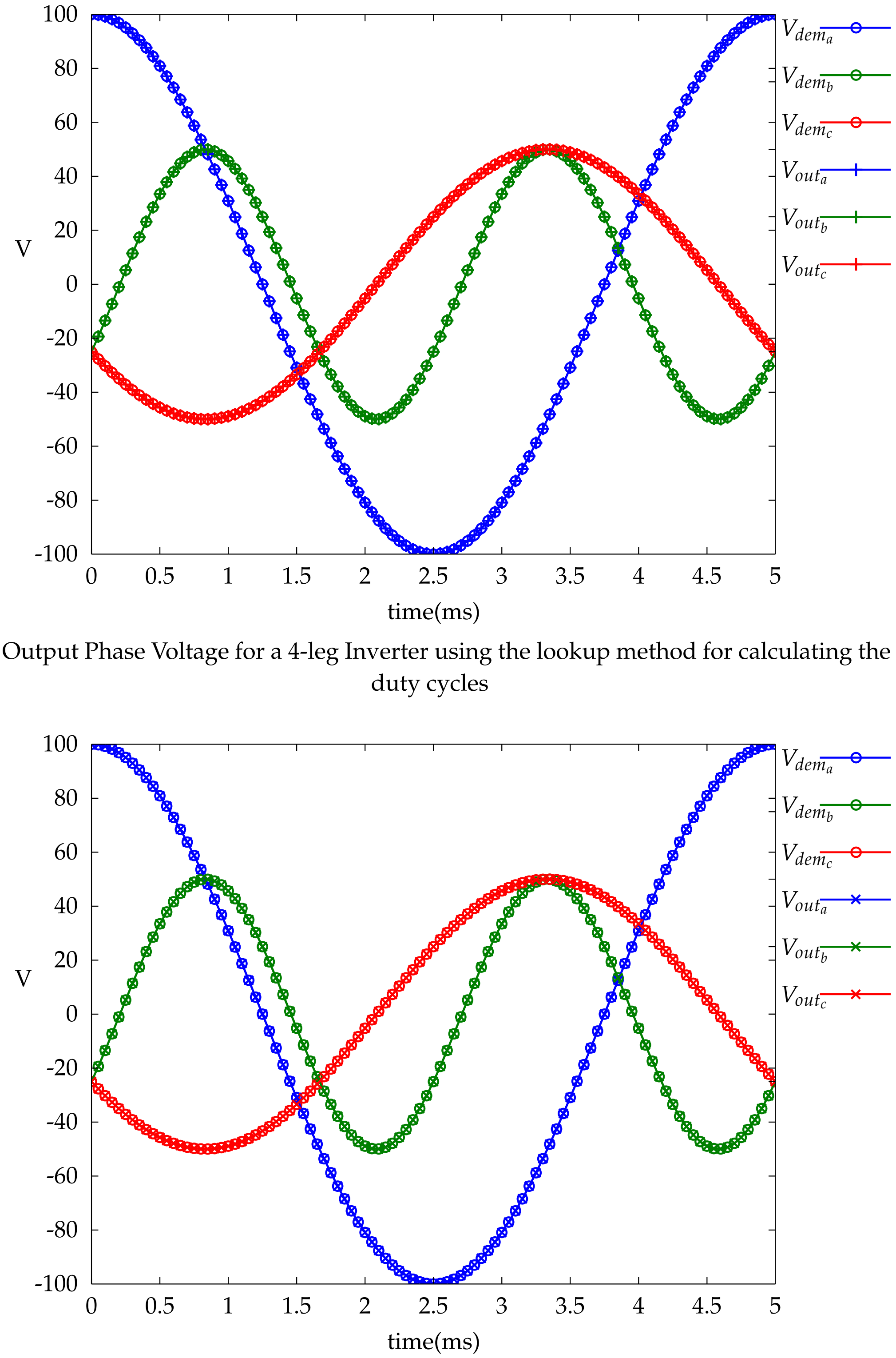
For the simulation this is done by setting one of the demand for one of the output phases, $V_{dem_{a(t)}}$, to have twice the voltage level of the other 2 phases, and then another output phase, $V_{dem_{b(t)}}$, to have a twice the output frequency of the other 2 phases, as shown in equation 6.1.36.

$$\begin{aligned} V_{dem_{a(t)}} &= 2\sqrt{2}V_{dem}\cos(2\pi f_{dem}t) \\ V_{dem_{b(t)}} &= \sqrt{2}V_{dem}\cos(4\pi f_{dem}t - \frac{2\pi}{3}) \\ V_{dem_{c(t)}} &= \sqrt{2}V_{dem}\cos(2\pi f_{dem}t - \frac{4\pi}{3}) \end{aligned} \tag{6.1.36}$$

Unlike the balanced 3-phase set, using these demand voltages gives the demand vector a complex path within the $\alpha\beta\gamma$ space, with the γ value going both positive and negative, instead of the vector being entirely within the $\alpha\beta$ plane as with the balanced set of demand voltages.

The results from this simulation can be seen in Figure 6.8, which show that both the lookup and general method for calculating the duty cycles are capable of producing an output to match the demand voltages.

From these results it is possible to state that the output from the general method for calculating the duty cycles produces the same results on the output legs as for the lookup table method. This shows that the derivation for this general method of calculating the duty cycles is correct, along with also proving that the simulation technique chosen for is correct, backing up the results from the 3x3 matrix converter. This then allows the final stage of the Matlab simulations to proceed, and begin looking at simulating the 4-leg Matrix converter.



(a) Output Phase Voltage for a 4-leg Inverter using the lookup method for calculating the duty cycles

(b) Output Phase Voltage for a 4-leg Inverter using the general method for calculating the duty cycles

Figure 6.8: Output Phase Voltage for a 4-leg Inverter when following a complex set of demand voltages

6.1.2 4-Leg Matrix Converter

With the results from the 4-leg inverter simulations proving that the method for simulating the 3-dimensional space vector modulation was correct, along with the general approach to calculating the duty cycles, the next stage was to pull together the various different parts from the above simulations to produce a simulation model for the complete 4-leg matrix converter. However there were a number of steps that needed to be overcome before this simulation could be made to work.

With the 4-leg inverter, as it only has a single input voltage, it is only possible to create a switching vector in a single way. However with the matrix converter, there are numerous converter switching states which can create any particular switching vector depending on the state of the input voltages at that time. This is a problem shared with the 3x3 matrix converter, however, with the greater number of output phases it increases the possible number of combinations, and makes the selection slightly more complex.

As with the 4-leg inverter, the switching states that would be required at any one instant would need to be vertices in the output voltage tetrahedron, but on top of that, they also need to be vertices in the input sector for the input current. Because any output tetrahedron can be associated with any input sector, and vice versa, then with six input sectors and twenty four output tetrahedrons this gives a possible of 144 different combinations. While it would be perfectly possible to use a large 1:1 lookup table to perform this task, this would lead to a quite large and unwieldy table, and with some experience from using the lookup tables in the previous simulations, one that would be hard to find errors within. As such another method was sought that could make this significantly easier to program, and less likely to error.

Early on in the work it had been noticed that there was a certain amount of symmetry within the input and output spaces, this can also be seen in with the 3x3 matrix converter, and so these would be a good place to start looking to simplify things.

As has been shown in Chapter 5 above, for each output voltage tetrahedron there is a set of possible switching states. Using the example in Table 5.4 above, the tetrahedron $6:3$ is made up of the space vectors $V8$, $V10$ and $V11$, and these can be generated by the following switching states

$$\text{Tetrahedron } 6:3 = \text{Vectors} \begin{bmatrix} V8 \\ V10 \\ V11 \end{bmatrix} = \begin{bmatrix} \pm 1 & \pm 2 & \pm 3 \\ \pm 16 & \pm 17 & \pm 18 \\ \pm 4 & \pm 5 & \pm 6 \end{bmatrix} \quad (6.1.37)$$

By repeating this exercise for each tetrahedron in the output voltage space it can be seen that there is a symmetry between them. With the above example, the opposite tetrahedron would be 3 : 2, which is made up from the space vectors $V4$, $V5$ and $V7$, and these can be generated by the following switching states

$$\text{Tetrahedron } 3:2 = \text{Vectors} \begin{bmatrix} V7 \\ V5 \\ V4 \end{bmatrix} = \begin{bmatrix} \pm 1 & \pm 2 & \pm 3 \\ \pm 16 & \pm 17 & \pm 18 \\ \pm 4 & \pm 5 & \pm 6 \end{bmatrix} \quad (6.1.38)$$

As can be seen, the set of possible switching states that can be used to create the space vectors for the two tetrahedrons opposite to one another are the same set of switching states. This is possible because, unlike the 4-leg inverter, there are always possible negative input voltages available, and switching a negative voltage in a particular switching state produces a vector in the opposite direction.

Now moving onto the input sector, as stated in Section 5.7 above, this selection of switching states can be narrowed depending on which input current sector is required. As one of the requirements for this matrix converter is that the input has a unity displacement factor, it is a simple case of defining the required input current sector as being the sector which the input voltage vector resides. Continuing the example from Table 5.4 above, if the input current sector is 6, this is made up of the space vectors $I6$ and $I5$, and these can be generated by the following switching states

$$\text{Sector } 1 = \text{Vectors} \begin{bmatrix} I6 \\ I5 \end{bmatrix} = \begin{bmatrix} \pm 1, \pm 4, \pm 7, \pm 10, \pm 13, \pm 16, \pm 19 \\ \pm 2, \pm 5, \pm 8, \pm 11, \pm 14, \pm 17, \pm 20 \end{bmatrix} \quad (6.1.39)$$

If this is then also performed for each of the other input current sectors, as for the output voltage tetrahedrons above, it can be seen that once again there is a symmetry between sectors. So, the opposite input current sector to the one shown above is sector 3, and this is made up of the space vectors $I3$ and $I4$, and these can be generated by the following switching states

$$\text{Sector 3} = \text{Vectors}\begin{bmatrix} I3 \\ I2 \end{bmatrix} = \begin{bmatrix} \pm1, \pm4, \pm7, \pm10, \pm13, \pm16, \pm19 \\ \pm2, \pm5, \pm8, \pm11, \pm14, \pm17, \pm20 \end{bmatrix} \tag{6.1.40}$$

Once again it can be seen that the possible switching states that can be used to generate the pair of opposite sectors are the same, and once again this is due to having both positive and negative voltages being generated at the outputs. Using these symmetries it is now possible to reduce the number of combinations of input and output from 144 to 36 such that

$$\text{Tetrahedrons } \frac{3:2}{6:3} \text{ and Sectors } \frac{3}{6} = \ \pm1, \pm2, \pm4, \pm5, \pm16, \pm17 \tag{6.1.41}$$

For each tetrahedron/sector pair there are 6 different pairs of possible switching states, however this needs to be refined further as there are only 6 different individual switching states required at any one point. Once again using the example from above, where the input current vector is in sector 6 and the output voltage demand is in tetrahedron 6 : 3 this gives

$$\text{Tetrahedron } 6:3 \text{ and Sector } 1 = \ \pm1, \pm2, \pm4, \pm5, \pm16, \pm17 \tag{6.1.42}$$

Know that the input current is in sector 6, due to the unity displacement factor this means that the voltage vector must also in input current sector 6, and to be in that sector it means that

$$\text{Sector } 6 = \begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} = \begin{bmatrix} \text{+ve} \\ \text{-ve} \\ \text{+ve} \end{bmatrix} \tag{6.1.43}$$

Looking at Table 5.2 at the 6 switching state pairs it can be seen that for these switching states the converter only switches between the line-line voltages V_{AB} and V_{BC}. So, during this time the only possible input voltages are

$$\text{Sector } 6 = \begin{bmatrix} V_{AB} \\ V_{BC} \end{bmatrix} = \begin{bmatrix} \text{+ve} \\ \text{-ve} \end{bmatrix} \tag{6.1.44}$$

Knowing that to generate the correct space vectors for the output voltage tetrahedron the switching states need to generate the vectors $V8$, $V10$ and $V11$, and once again referring back to Table 5.2, it shows that with a positive V_{AB} and a negative V_{BC} the following switching states are needed

$$\text{Sector } 6 = \begin{bmatrix} +V_{AB} \\ -V_{CA} \end{bmatrix} = \begin{bmatrix} +1 & +5 & +16 \\ -2 & -4 & -17 \end{bmatrix} \tag{6.1.45}$$

Performing this same analysis for the other three tetrahedron/sector pairs which share this set of 6 switching state pairs then gives the results seen in Table 6.4.

Table 6.4: Switching state selection example

		Current Sector	
		Current Sector 3	Current Sector 6
Tetrahedron	3:2	$-1, +2, +4, -5, -16, +17$	$+1, -2, -4, +5, +16, -17$
	6:3	$-1, +2, +4, -5, -16, +17$	$+1, -2, -4, +5, +16, -17$

From these results it is possible to see yet another set of symmetries within this set of sector/tetrahedron pairs, where for the same tetrahedron the two opposite input current sectors lead to sets of switching states with opposite signs, and then for the same input current sector the two opposite output voltage tetrahedrons lead to sets of switching states with opposite signs. If this is considered for a moment then this should be obvious as, for both the pair of input current sectors and pair of output voltage tetrahedrons, they are generated from opposite sets of space vectors, for example $V8$ equals $-V7$ and $I5$ equals $-I2$.

This exercise was then repeated for each of the set of sector/tetrahedron pairs and it was found that they all contain the same symmetry as that seen in Table 6.4. This pointed to how it would be possible to calculate the required switching states for each combination of input current sector and output voltage tetrahedron using the procedure shown below. As will become apparent, most of the information used in this process is held in a number of different lookup tables. While this might not be the most efficient method, it was chosen, instead of having a small separate routine for each tetrahedron/sector pair, because it held all of the similar data together, and so it was far easier to ensure the information was correct, to make changes to the code, and to find any bugs within the simulation files. The loss in speed by performing the simulation in this way is minimal in this instance.

The first step is to calculate the input current sector using the same method as used for the 3x3 matrix converter so

$$Sector_i = \begin{cases} 1 & \text{if} \quad \frac{11\pi}{6} \geq \beta_i < \frac{\pi}{6} \\ 2 & \text{if} \quad \frac{\pi}{6} \leq \beta_i < \frac{\pi}{2} \\ 3 & \text{if} \quad \frac{\pi}{2} \leq \beta_i < \frac{5\pi}{6} \\ 4 & \text{if} \quad \frac{5\pi}{6} \leq \beta_i < \frac{7\pi}{6} \\ 5 & \text{if} \quad \frac{7\pi}{6} \leq \beta_i < \frac{3\pi}{2} \\ 6 & \text{if} \quad \frac{3\pi}{2} \leq \beta_i < \frac{11\pi}{6} \end{cases} \tag{6.1.46}$$

Then calculate the output voltage tetrahedron in the same way as for the 4-leg inverter, by first calculating which prism in $\alpha\beta\gamma$ space the demand vector resides in by

$$Prism_o = \begin{cases} 1 & \text{if} \quad 0 \leq \theta_o < \frac{\pi}{3} \\ 2 & \text{if} \quad \frac{\pi}{3} \leq \theta_o < \frac{2\pi}{3} \\ 3 & \text{if} \quad \frac{2\pi}{3} \leq \theta_o < \pi \\ 4 & \text{if} \quad \pi \leq \theta_o < \frac{4\pi}{3} \\ 5 & \text{if} \quad \frac{4\pi}{3} \leq \theta_o < \frac{5\pi}{3} \\ 6 & \text{if} \quad \frac{5\pi}{3} \leq \theta_o < 2\pi \end{cases} \tag{6.1.47}$$

and then selecting which tetrahedron by counting the number of positive phases in the demand voltages

```
sign = 1;

if (Vdemand(1) >= 0)
    sign = sign + 1;
end

if (Vdemand(2) >= 0)
    sign = sign + 1;
end

if (Vdemand(3) >= 0)
    sign = sign + 1;
end
```

Then, using the values found in $Prism_o$ and **sign**, using the lookup table shown in Table 6.5 to find $Tetra_o$ which determines which tetrahedron the demand vector lies within. Once again it should be noted that due to the symmetry of the space vectors and tetrahedrons, only the 12 tetrahedrons need to be defined, as the as the tetrahedrons opposite each other in $\alpha\beta\gamma$ space are defined by the same set of switching states.

Table 6.5: Tetrahedron Lookup

		sign =			
		1	2	3	4
Prism	1	1	2	3	4
	2	5	6	7	8
	3	9	10	11	12
	4	4	3	2	1
	5	8	7	6	5
	6	12	11	10	9

Next, using the values already found for $Sector_i$ and $Tetra_o$, use another lookup table, shown in Table 6.6 , to find the 3 space vectors, $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$, which form the vertices of the tetrahedron

The order that both the space vectors and the switching states is held is important here because of the relationship between them. As can be seen in Section 5.9 the relationship is that the switching states $\mathbf{ss_I}$ and $\mathbf{ss_{VI}}$ are associated with space vector $\vec{v}_1$, and likewise $\mathbf{ss_2}$ and $\mathbf{ss_5}$ with space vector $\vec{v}_2$ and then $\mathbf{ss_3}$ and $\mathbf{ss_4}$ with $\vec{v}_3$.

With this relationship, it becomes easy to identify the pairs of switching states related to each vector, but some care needs to be taken when identifying which switching state to associate with with duty cycle. Comparing the equations for the duty cycles δ^I (5.9.27) and δ^{VI} (5.9.32), both of which are associated with the same space vector $\vec{v}_1$. As can be seen the only different between the two equations is that one uses $(\tilde{\beta}_i + \frac{\pi}{3}$, and the other $(\tilde{\beta}_i - \frac{\pi}{3}$ in the numerator. By once again using the example throughout Chapter 5, and looking at Figure 5.13, it can be seen that the $(\tilde{\beta}_i + \frac{\pi}{3}$ duty cycle is associated with the lower of the two space vectors, in this case $I5$, while the $(\tilde{\beta}_i - \frac{\pi}{3}$ duty cycle is related to the upper space vector ($I6$).

From Table 5.4, the switching states $+1, -4and + 16$ are associated with space vector $I6$ and so use the $(\beta_i + \frac{\pi}{3}$ duty cycles, and switching states $-2, +5and - 17$ with space vector $I1$ and $(\beta_i - \frac{\pi}{3}$. By performing this analysis for each tetrahedron and input current sector, it was possible

Table 6.6: Switching state selection example

			$Sector_i$	
			1, 3 or 5	2, 4 or 6
			Space Vectors ($\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$)	
Tetrahedron	1	1 : 1 4 : 4	$\vec{V1}$, $\vec{V9}$, $\vec{V13}$	$\vec{V2}$, $\vec{V6}$, $\vec{V14}$
	2	1 : 2 4 : 3	$\vec{V8}$, $\vec{V9}$, $\vec{V13}$	$\vec{V2}$, $\vec{V6}$, $\vec{V7}$
	3	1 : 3 4 : 2	$\vec{V8}$, $\vec{V12}$, $\vec{V13}$	$\vec{V2}$, $\vec{V3}$, $\vec{V7}$
	4	1 : 4 4 : 1	$\vec{V8}$, $\vec{V12}$, $\vec{V14}$	$\vec{V1}$, $\vec{V3}$, $\vec{V7}$
	5	2 : 1 5 : 4	$\vec{V1}$, $\vec{V5}$, $\vec{V13}$	$\vec{V2}$, $\vec{V10}$, $\vec{V14}$
	6	2 : 2 5 : 3	$\vec{V4}$, $\vec{V5}$, $\vec{V13}$	$\vec{V2}$, $\vec{V10}$, $\vec{V11}$
	7	2 : 3 5 : 2	$\vec{V4}$, $\vec{V12}$, $\vec{V13}$	$\vec{V2}$, $\vec{V3}$, $\vec{V11}$
	8	2 : 4 5 : 1	$\vec{V4}$, $\vec{V12}$, $\vec{V14}$	$\vec{V1}$, $\vec{V3}$, $\vec{V11}$
	9	3 : 1 6 : 4	$\vec{V1}$, $\vec{V5}$, $\vec{V7}$	$\vec{V8}$, $\vec{V10}$, $\vec{V14}$
	10	3 : 2 6 : 3	$\vec{V4}$, $\vec{V5}$, $\vec{V7}$	$\vec{V8}$, $\vec{V10}$, $\vec{V11}$
	11	3 : 3 6 : 2	$\vec{V4}$, $\vec{V6}$, $\vec{V7}$	$\vec{V8}$, $\vec{V9}$, $\vec{V11}$
	12	3 : 4 6 : 1	$\vec{V4}$, $\vec{V6}$, $\vec{V14}$	$\vec{V1}$, $\vec{V9}$, $\vec{V11}$

to build up the following sets of lookup tables, which would not only work for the Matlab simulation, but would also be used for the Saber simulations and the build of the actual converter.

So, using the same values found for $Sector_i$ and $Tetra_o$, use the lookup table, shown in Table 6.7 to determine the base set of switching states, $\mathbf{ss_I}$ to $\mathbf{ss_{VI}}$, for that input sector/tetrahedron pair.

The next step is to define which switching state in each of the selected pairs is going to be required, and which duty cycle that switching state is associated with. Once again a lookup table is used, which is shown in Table 6.8.

Table 6.7: Switching state selection example

			$Sector_i$		
			1 or 4	2 or 5	3 or 6
			Switching states ($\mathbf{ss_I}$, $\mathbf{ss_{II}}$, $\mathbf{ss_{III}}$, $\mathbf{ss_{IV}}$, $\mathbf{ss_V}$, $\mathbf{ss_{VI}}$)		
$Tetra_o$	1	1 : 1 4 : 4	19, 13, 7, 9, 15, 21	21, 15, 9, 8, 14, 20	20, 14, 8, 7, 13, 19
	2	1 : 2 4 : 3	1, 13, 7, 9, 15, 3	3, 15, 9, 8, 14, 2	2, 14, 8, 7, 13, 1
	3	1 : 3 4 : 2	1, 10, 7, 9, 12, 3	3, 12, 9, 8, 11, 2	2, 11, 8, 7, 10, 1
	4	1 : 4 4 : 1	1, 10, 19, 21, 12, 3	3, 12, 21, 20, 11, 2	2, 11, 20, 19, 10, 1
	5	2 : 1 5 : 4	19, 16, 7, 9, 18, 21	21, 18, 9, 8, 17, 20	20, 17, 8, 7, 16, 19
	6	2 : 2 5 : 3	4, 16, 7, 9, 18, 6	6, 18, 9, 8, 17, 5	5, 17, 8, 7, 16, 4
	7	2 : 3 5 : 2	4, 10, 7, 9, 12, 6	6, 12, 9, 8, 11, 5	5, 11, 8, 7, 10, 4
	8	2 : 4 5 : 1	4, 10, 19, 21, 12, 6	6, 12, 21, 20, 11, 5	5, 11, 20, 19, 10, 4
	9	3 : 1 6 : 4	19, 16, 1, 3, 18, 21	21, 18, 3, 2, 17, 20	20, 17, 2, 1, 16, 19
	10	3 : 2 6 : 3	4, 16, 1, 3, 18, 6	6, 18, 3, 2, 17, 5	5, 17, 2, 1, 16, 4
	11	3 : 3 6 : 2	4, 13, 1, 3, 15, 6	6, 15, 3, 2, 14, 5	5, 14, 2, 1, 13, 4
	12	3 : 4 6 : 1	4, 13, 19, 21, 15, 6	6, 15, 21, 20, 14, 5	5, 14, 20, 19, 13, 4

At this point everything is known that will allow the duty cycles to be calculated:

- $Sector_i$ states what input current sector the input voltage vector lies within
- $Prism_o$ states which prism in the output voltage space the demand vector sit in
- $Tetra_o$, which is in a particular tetrahedron in the output voltage space the demand vector sits within

Table 6.8: Switching state selection example

		sign	Space Vectors	$Prism_o$	
				1, 2 or 3	4, 5 or 6
$Sector_i$	1, 3 or 5	1	$\vec{v_1} = \vec{v_1}$ $\vec{v_2} = \vec{v_2}$ $\vec{v_3} = \vec{v_3}$	$-\mathbf{ss}_{\mathrm{I}} - \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{III}} + \mathbf{ss}_{\mathrm{IV}} + \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{VI}}$	$+\mathbf{ss}_{\mathrm{III}} + \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{I}} - \mathbf{ss}_{\mathrm{VI}} - \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{IV}}$
		2		$+\mathbf{ss}_{\mathrm{I}} - \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{III}} + \mathbf{ss}_{\mathrm{IV}} + \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{VI}}$	$+\mathbf{ss}_{\mathrm{III}} + \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{I}} + \mathbf{ss}_{\mathrm{VI}} - \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{IV}}$
		3		$+\mathbf{ss}_{\mathrm{I}} + \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{III}} + \mathbf{ss}_{\mathrm{IV}} - \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{VI}}$	$+\mathbf{ss}_{\mathrm{III}} - \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{I}} + \mathbf{ss}_{\mathrm{VI}} + \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{IV}}$
		4		$+\mathbf{ss}_{\mathrm{I}} + \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{III}} - \mathbf{ss}_{\mathrm{IV}} - \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{VI}}$	$-\mathbf{ss}_{\mathrm{III}} - \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{I}} + \mathbf{ss}_{\mathrm{VI}} + \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{IV}}$
	2, 4 or 6	1	$\vec{v_1} = \vec{v_3}$ $\vec{v_2} = \vec{v_2}$ $\vec{v_3} = \vec{v_1}$	$-\mathbf{ss}_{\mathrm{IV}} - \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{VI}} + \mathbf{ss}_{\mathrm{I}} + \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{III}}$	$+\mathbf{ss}_{\mathrm{VI}} + \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{IV}} - \mathbf{ss}_{\mathrm{III}} - \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{I}}$
		2		$-\mathbf{ss}_{\mathrm{IV}} - \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{VI}} - \mathbf{ss}_{\mathrm{I}} + \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{III}}$	$-\mathbf{ss}_{\mathrm{VI}} + \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{IV}} - \mathbf{ss}_{\mathrm{III}} - \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{I}}$
		3		$-\mathbf{ss}_{\mathrm{IV}} + \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{VI}} - \mathbf{ss}_{\mathrm{I}} - \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{III}}$	$-\mathbf{ss}_{\mathrm{VI}} - \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{IV}} - \mathbf{ss}_{\mathrm{III}} + \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{I}}$
		4		$+\mathbf{ss}_{\mathrm{IV}} + \mathbf{ss}_{\mathrm{V}} + \mathbf{ss}_{\mathrm{VI}} - \mathbf{ss}_{\mathrm{I}} - \mathbf{ss}_{\mathrm{II}} - \mathbf{ss}_{\mathrm{III}}$	$-\mathbf{ss}_{\mathrm{VI}} - \mathbf{ss}_{\mathrm{V}} - \mathbf{ss}_{\mathrm{IV}} + \mathbf{ss}_{\mathrm{III}} + \mathbf{ss}_{\mathrm{II}} + \mathbf{ss}_{\mathrm{I}}$

- $\vec{v_1}$, $\vec{v_2}$ and $\vec{v_3}$ hold the 3 space vectors defining the tetrahedron associated with $Tetra_o$
- $\mathbf{ss}_{\mathbf{I}}$ to $\mathbf{ss}_{\mathbf{VI}}$ hold the switching states which create the space vectors associated with $\vec{v_1}$, $\vec{v_2}$ and $\vec{v_3}$

The next step is to define the duty cycles in relation to this information. Using the same method as was used for the 4-leg inverter, and looking at equations 5.9.27-5.9.32 in Section 5.9, they are the set of equations which allow the calculation of the duty cycles for each switching state. These all follow the basic form

$$\delta^x = \frac{2}{\sqrt{3}} \frac{v_d}{v_i} \frac{\cos \omega^x \cos(\tilde{\beta}_i + \frac{\pi}{3})}{\cos \phi} \tag{6.1.48}$$

As previously stated, this converter is designed to have a unity displacement factor, therefore $\cos \phi$ is set to 1. Alongside this, it was shown in (6.1.25) that the length of the normal vector is always constant, no matter which combination of the 3 space vectors which define a tetrahedron are chosen, so the dot product of the normal vector to the plane $\left(\vec{n_{yz}}\right)$ and the demand vector $(\vec{n_d})$ becomes

$$\begin{aligned} \vec{n_{yz}} \cdot \vec{v_d} &= \frac{\sqrt{2}}{3} v_d \cos \omega^x \\ v_d \cos \omega^x &= \frac{3}{\sqrt{2}} \vec{n_{yz}} \cdot \vec{v_d} \end{aligned} \tag{6.1.49}$$

Substituting this back into 6.1.48, the basic form of the equation then becomes

$$\delta^x = \frac{\sqrt{6}}{v_i}\vec{n_{yz}} \cdot \vec{v_d}\cos(\beta_i \pm \frac{\pi}{3}) \tag{6.1.50}$$

As this work on deriving the various equations took place, each part was written into a separate Matlab m-file function. This allowed each stage to be easily checked to ensure that it was correct before then using each of them be used to build up the overall program to simulate the entire converter. Figure 6.9 shows each of these different functions, and how they all fit together to form the overall simulation. As can be seen in the block diagram, the simulation actually performs the calculation of the duty cycles twice, once by using the equations (5.9.27)-(5.9.32) for δ derived in Chapter 5, and the second set is to validate the simplification of the equations which were derived earlier in this section.

6.1.2.1 Matlab 4-Leg Matrix Converter Simulation Results

Once the Matlab functions had each been proved to be functioning, it was then possible to perform a complete simulation of the 4-leg matrix converter, and Figure 6.10 shows the results for the output voltage calculated from the duty cycle equations (5.9.27)-(5.9.32) derived in Chapter 5. In this simulation, the converter was supplied with an input voltage of 240Vrms, and was asked to provide an output with a peak voltage of 200V at 100Hz. The results here are the phase to neutral voltages V_{an}, V_{bn} and V_{cn} plotted alongside the respective voltage demand signal, and as can be seen, the results match perfectly.

The simulation was also setup to calculate the simplified method of calculating the duty cycles, as set out earlier in this chapter, and the results from this method, in the same simulation as before, are shown in Figure 6.11. Once again, the calculated phase-neutral voltages at the outputs match the demand voltages.

The last sets of Matlab simulations to be performed were to compare the output voltage transfer ratio with the sum of the non-zero switching state duty cycles. As discussed in Section 5.10, the voltage transfer ratio is not a simple matter now that the converter has a fourth output leg, and these simulations were performed in order to test the equation (5.10.26), which states that the maximum transfer ratio is defined by

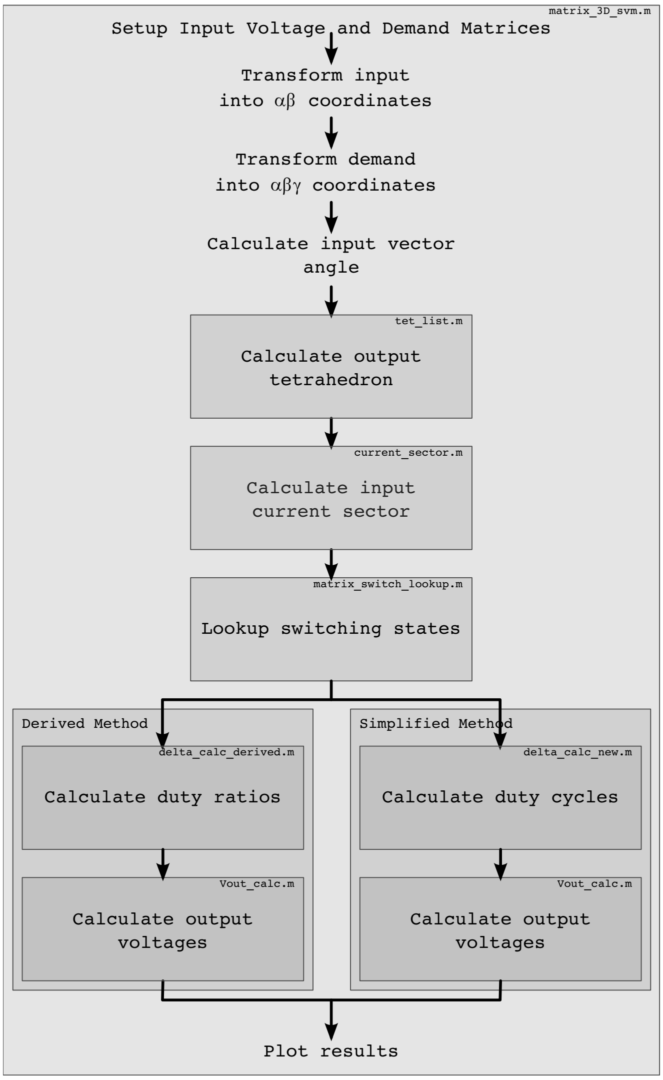

matrix_3D_svm.m
Setup Input Voltage and Demand Matrices
Transform input into αβ coordinates
Transform demand into αβγ coordinates
Calculate input vector angle
tet_list.m
Calculate output tetrahedron
current_sector.m
Calculate input current sector
matrix_switch_lookup.m
Lookup switching states
Derived Method
delta_calc_derived.m
Calculate duty ratios
Vout_calc.m
Calculate output voltages
Simplified Method
delta_calc_new.m
Calculate duty cycles
Vout_calc.m
Calculate output voltages
Plot results

Figure 6.9: Block Diagram of the 4-Leg Matrix Converter Matlab Simulation

$$q_{max} = \frac{\sqrt{3}}{2} \cos \phi \tag{6.1.51}$$

The first simulation used a demand voltage of $\frac{\sqrt{3}}{2} V_{in_{peak}}$ with an output frequency of 100Hz, and the results of the output voltage can be seen in Figure 6.13, and these are as expected, with the output voltage matching the demand voltage. With this demand voltage, being the calculated maximum for the input voltage, the plot of the sum of the duty cycles should reach a peak of 1 at several points during the simulation when the output voltage is at its maximum, and the available input voltage at its minimum. However, looking at the plot of the duty cycles in Figure 6.14a it can be seen that the total of the duty cycles does not actually reach 1,

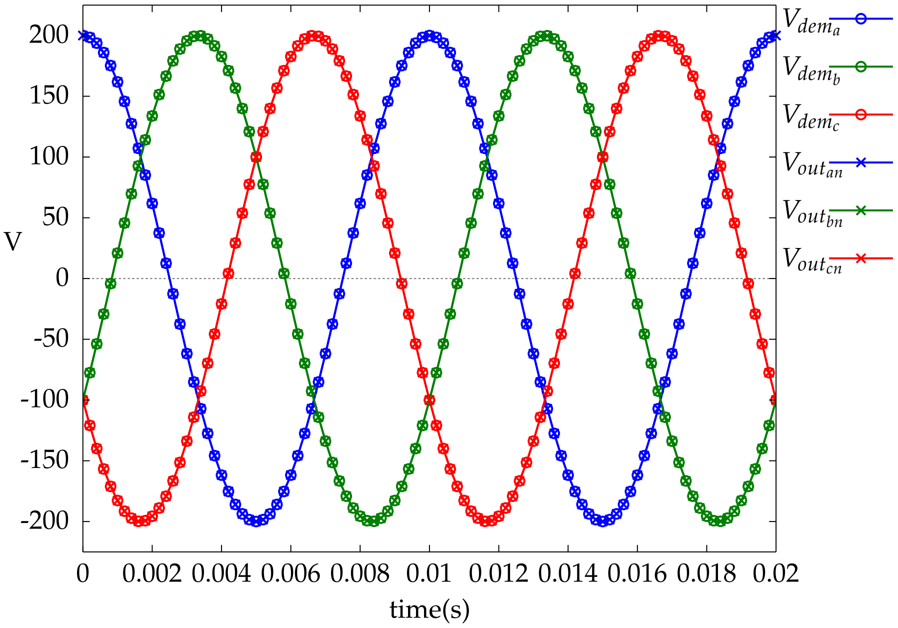


Figure 6.10: 4-Leg Matrix Converter output voltages for the original derived duty cycle equations

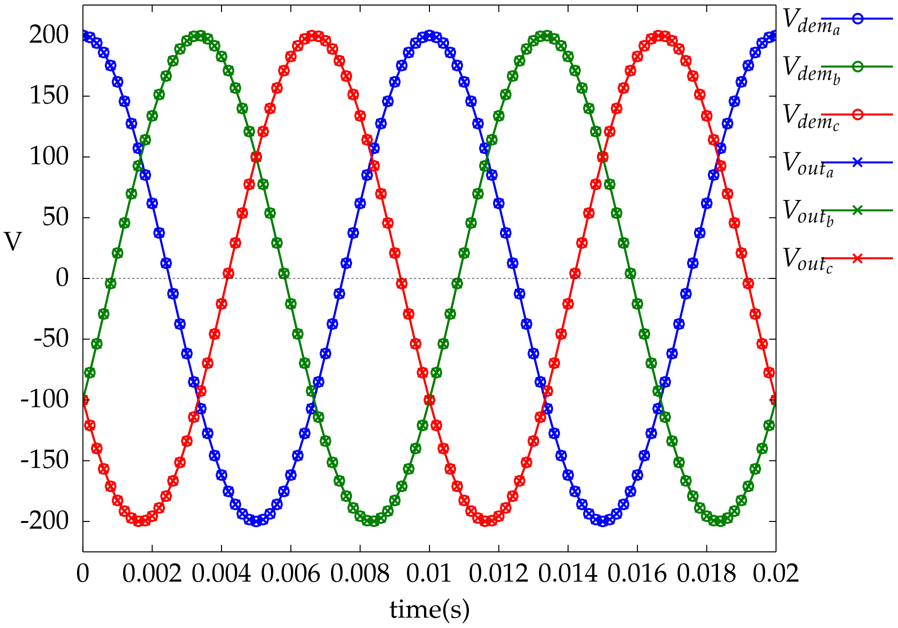
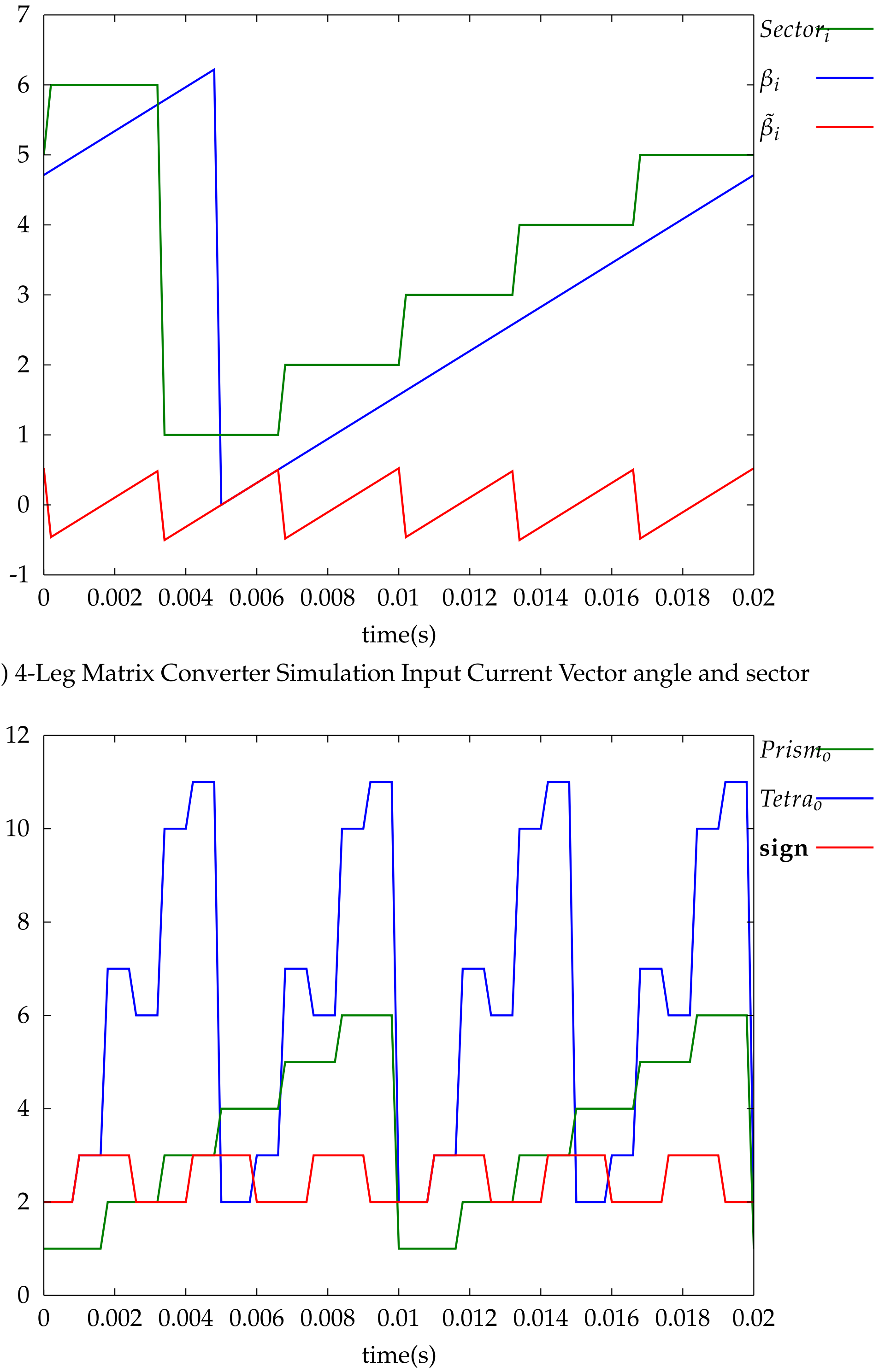


Figure 6.11: 4-Leg Matrix Converter output voltages for the simplified duty cycle equations

it reaches a maximum of approximately 0.96, and so it appears to not be following the inequality found in equation (5.10.26). The reason for this discrepancy is found in Figure 6.14b, which is a plot of the voltage generated at each of the output legs, including the neutral output leg, and all referenced to the supply neutral. This shows that the output



(a) 4-Leg Matrix Converter Simulation Input Current Vector angle and sector

(b) 4-Leg Matrix Converter Simulation Output Voltage Tetrahedron and Sector

Figure 6.12: 4-Leg Matrix Converter Simulation Internal Variables

voltage, being at a frequency which is an integer multiple of the input frequency, is synchronised with the input so that the peak demands in the output voltage match the peaks in the available input voltage, and likewise for the minimums.

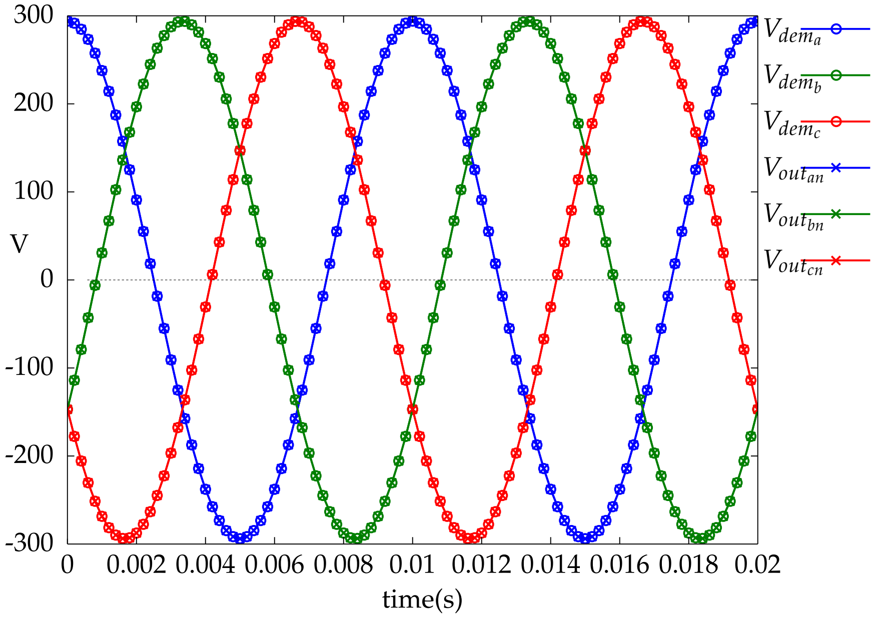
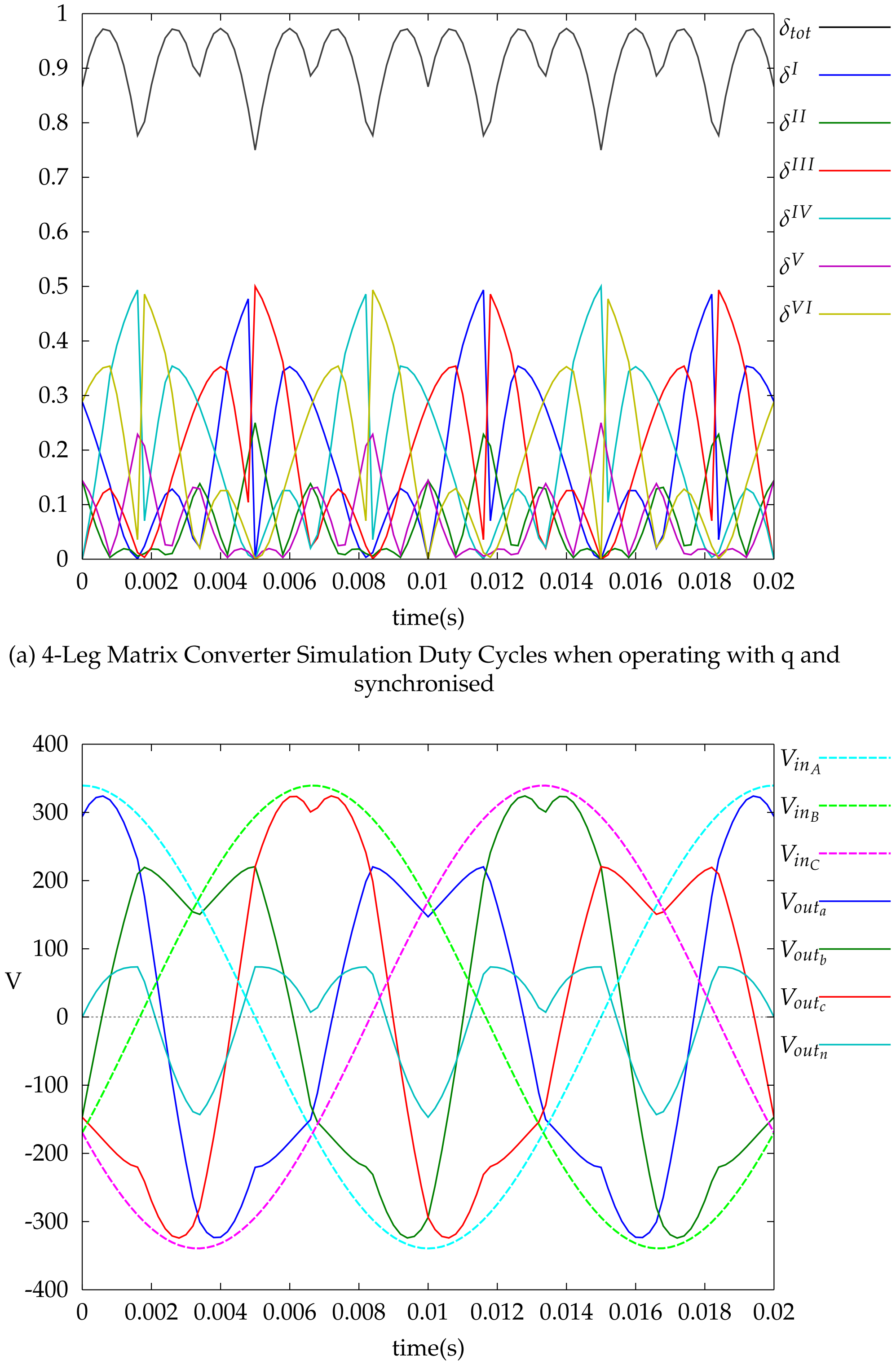


Figure 6.13: 4-Leg Matrix Converter Output Voltages when operating with q

If the frequency of the output voltage is just slightly offset from that of the input frequency this synchronisation is lost and the total of the non-zero duty cycles once again peaks at 1, and this can be seen in Figure 6.15a, where the simulation is identical to the previous one, except that the input frequency has been offset by $\frac{\pi}{12}$. The output phase-neutral voltages remain unchanged from those shown in Figure 6.13, but looking that the individual leg voltages when referenced to the supply neutral, shown in Figure 6.15b, the difference with the previous simulation is easy to see, and that there are several points throughout the cycle where the peak output voltage equals the available input voltage.

So, as can be seen by the results shown above, while limit to the voltage transfer ratio of $0.866V_{in}$ is true in a general case for a balanced 3-phase output, there are once again specific times when this is not the case.

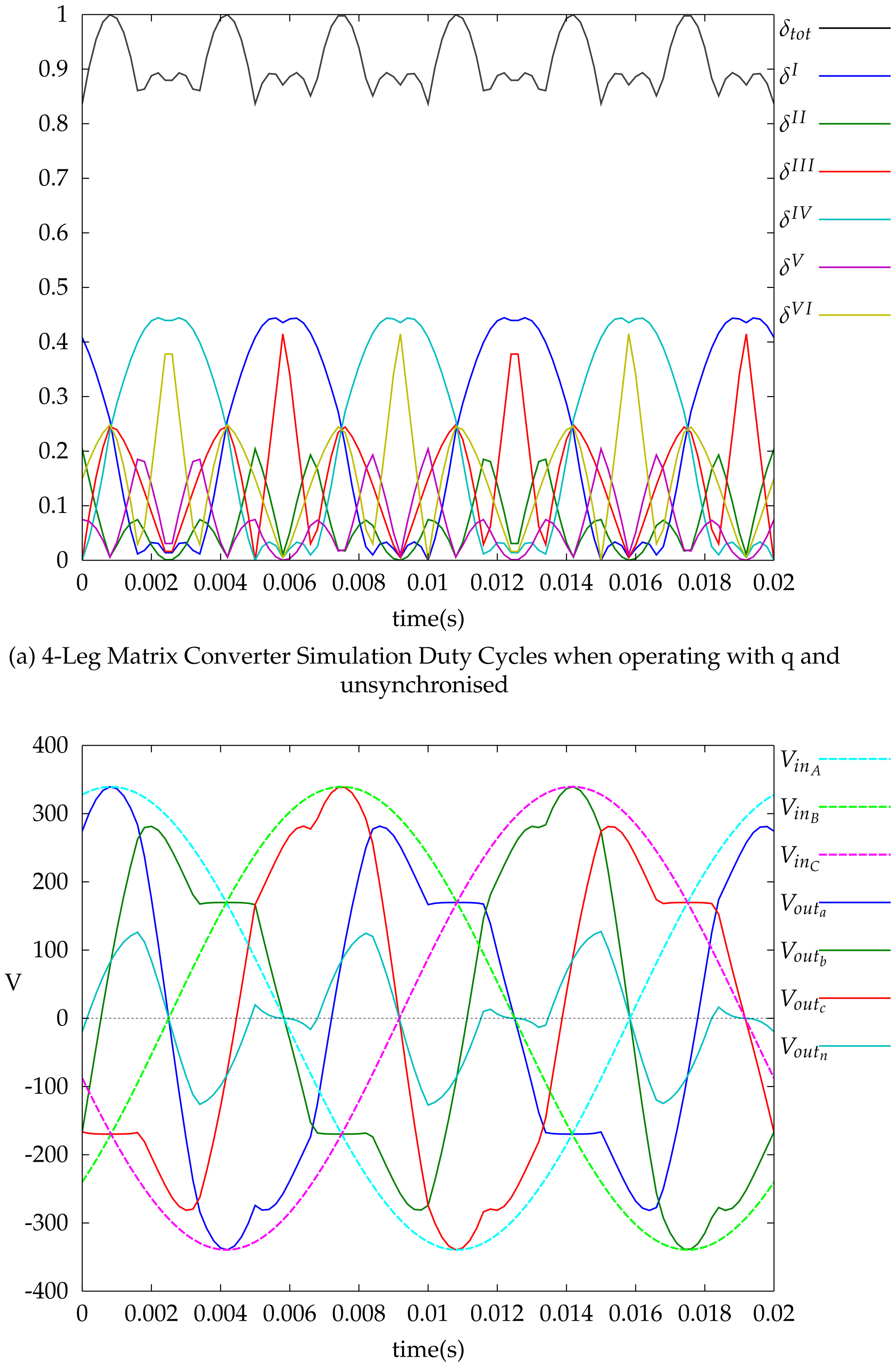
Looking at all the results from the Matlab simulations, they show that the assumptions made, and the derived equations appear to be correct, with the output voltages V_a, V_b and V_c being able to be independently controlled with respect to the neutral leg voltage. Obviously there are limitations to these simulations in that this is only simulating the output voltage, and so the input current control has yet to be tested. So, while this does not fully validate the design, it gives enough confidence in the operation of the circuit at this point to move onto the next stage,



(a) 4-Leg Matrix Converter Simulation Duty Cycles when operating with q and synchronised

(b) 4-Leg Matrix Converter Simulation Input and Output Voltages, referenced to supply neutral, when operating with q and synchronised

Figure 6.14: 4-Leg Matrix Converter Simulation Results when operating with q and voltages synchronised



(a) 4-Leg Matrix Converter Simulation Duty Cycles when operating with q and unsynchronised

(b) 4-Leg Matrix Converter Simulation Input and Output Voltages, referenced to supply neutral, when operating with q and unsynchronised

Figure 6.15: 4-Leg Matrix Converter Simulation Results when operating with q and voltages unsynchronised

simulating the entire circuit using Saber.

6.2 Simulation using Saber

The Matlab simulations had so far proved that the equations which had been derived were providing the expected results at that stage. They show that the output voltage is able to be controlled, and also demonstrated how the internal variables used for selecting the input sector and output tetrahedron were being correctly selected. At this point the simulation was moved on to the next stage by using Saber, which is a simulation software package from Synopsys. This is a powerful circuit simulation tool which can be used to simulate complex circuits, both analogue and digital, and this would allow a more detailed simulation of the 4-leg matrix converter to be performed, this time being able to take into account the load on the converter. This meant that in addition to looking at the output voltage it would also allow the input current waveforms to be investigated as well.

The first step in the simulation process with Saber is to design the circuit which will be simulated. Now, Saber is an incredibly powerful tool, and it is easy to over-complicate a simulation by going into far too much detail in the circuit design, and while this will provide very accurate results it would also be very slow, and the vast amount of the information which it would be able to provide would be unnecessary for this investigation. So, as always with any simulation, there is a trade-off between the speed which the simulation runs and the amount of detail which the results provide.

The circuit for the 4-leg matrix converter is made up of 12 bi-directional switches, the basic layout of which can be seen in Figure 1.6, with an example switch being shown in Figure 6.16. Each switch is made up of 2 diodes and 2 IGBTs, with each of the IGBT's requiring a separate gate drive circuit, and then also each IGBT has a separate gate drive signal to ensure that current commutation occurs correctly, using one of the techniques described in Chapter 4. Now, while a gate drive, with the logic required to commutate the current correctly, might be a simple circuit to build within Saber, having 24 of these will increase the number of elements requiring simulation. Even though this would lead to a very good model of a working converter, the extra information held in the results would be unnecessary as the interest in these simulations is in being able to verify that the derivation and operation of the 4-leg matrix converter is correct, and that the overall switching patterns and overall circuit operation are as expected when attached to a load.

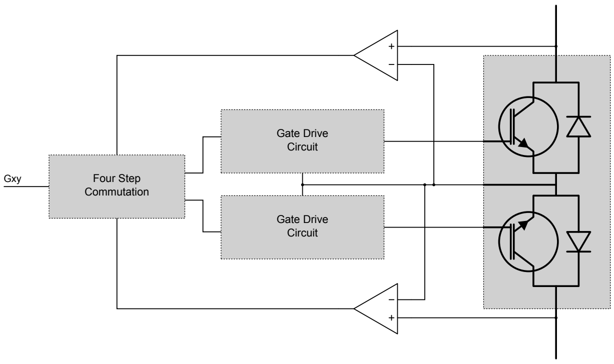


Figure 6.16: Bi-directional Switch Circuit

So, taking this into account, the decision was taken to use a circuit that would be a lot simpler, and therefore quicker, to simulate. In place of the IGBTs with their gate drives and the added complexity of their commutation, ideal switching elements were used. These has a number of advantages, firstly they are driven by a logic level input which removes the need to any extra gate drive circuit, and secondly, due to it being an ideal switch, it has near instantaneous switching times, which removes the need to use any commutation. So, by making this change, and using simple ideal switches instead the IGBTs, a large number of circuit elements were removed, simplifying things greatly, and so speeding up the simulation greatly. The new circuit for the bi-directional switch is shown in Figure 6.17.

The circuit was then built up using a 3 x 4 block of these bi-directional switching elements as the basis of the converter. This circuit is shown in Figure 6.18, and as can be seen, without the need for gate drives and commutation, it is now a relatively simple circuit. The only other required circuit elements being a control block, which will be discussed below, a set of sinusoidal voltage sources which make up the 3-phase source feeding directly into the converter, a set of output inductors on all 4 converter output legs, and finally a resistive load on each of the 3 output phase legs. Using the voltage sources directly onto the inputs gives the voltage stiff characteristic as required at the input to the converter, and likewise, the inductors on each of the 4 output legs ensures that the output is current stiff.

The biggest challenge for this simulation was the formation of the control block, which is shown in Figure 6.19. It is by far the most complex element making up the simulation circuit and its operation will be now

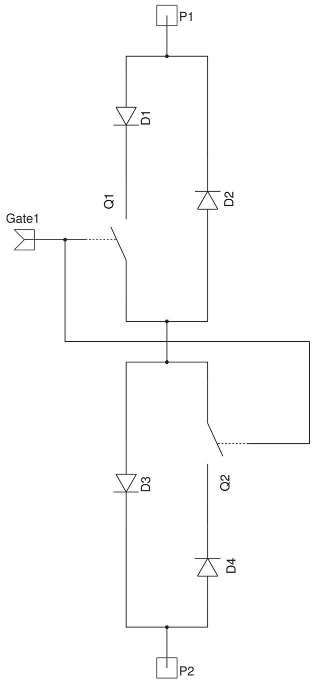


Figure 6.17: Circuit Diagram used for the Bi-Directional Switch block in the Saber simulations

be discussed.

The control block shown, as shown in Figure 6.19, is an element which uses Saber's internal scripting language, MAST, to define its internal operation, and then define how that appears on its outputs. The MAST language is a hardware description language(HDL) which gives a programmatic way of describing both analogue and digital circuits. For this simulation the control block is being used to calculate the duty cycles and then output the respective switch control signals for each of the 12 switching blocks in the converter. These output signals will need to depend on both the input voltage and the demand voltage/frequency. There are also a number of additional outputs labelled *to*0 to *to*7, these are used to allow the internal variables within the block to be monitored.

The design of the MAST language template for the control block started with the Matlab m-files used in Section 6.1.2 above, but as it is not possible to directly run the Matlab simulation files within the simulation package, these had to be modified. Obviously a reasonable amount of the Matlab files themselves were also concerned with the mechanics of the simulation itself, and so needed to be removed anyway, this left behind the core logic required to calculate the switching states and duty cycles for the converter. This core part of program could then be modified to meet the requirements of the MAST language. While the basic

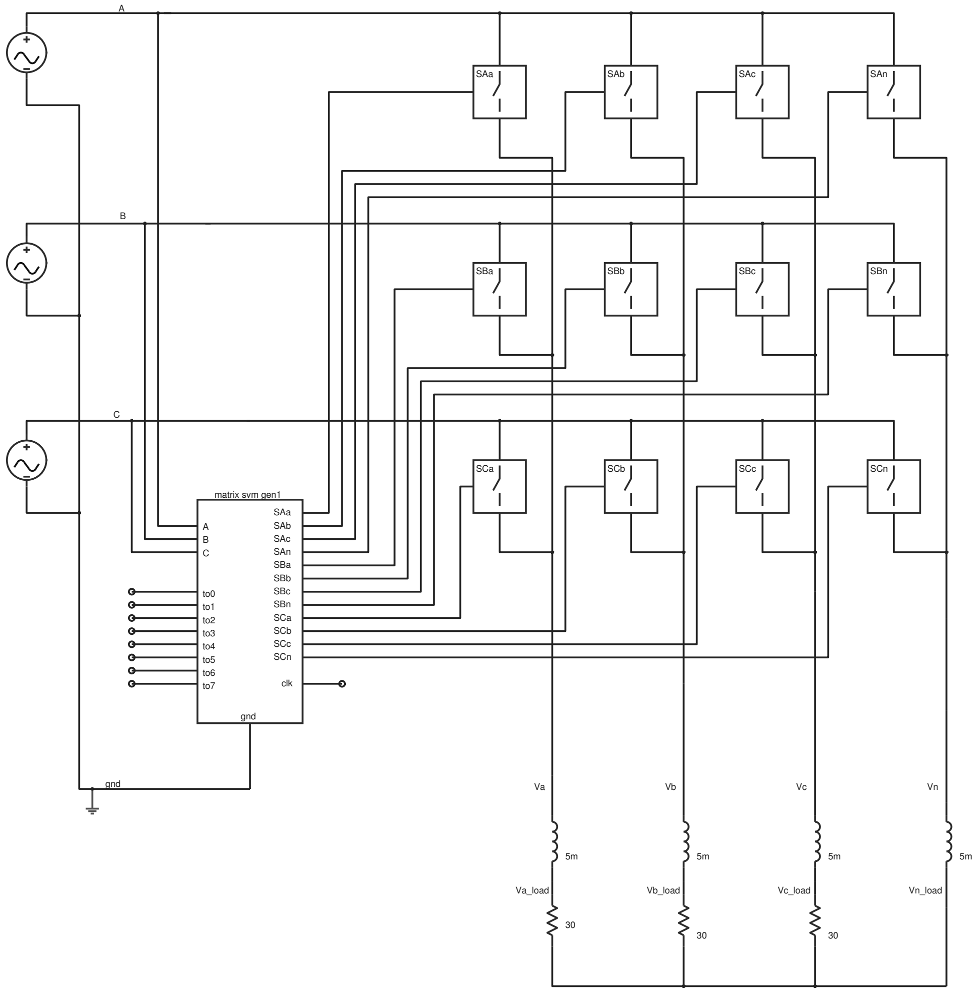


Figure 6.18: Circuit Diagram used for the Saber simulations

steps of the calculations were kept essentially the same as those used within the Matlab simulations, in order to increase the simulation speed a number of the lookup tables were been removed, and this information within the tables was included into the main body of the program. By doing this it increased the speed of the simulation by approximately 10%.

Having modified the core calculations for the converter to more suit the MAST environment, there was still an issue which needed to be solved before the simulation begin. That problem was to do with how the actual switching would take place, in what order, and how that would be generated within the MAST language. Because the actual switching process was not required for the Matlab simulations, as the output voltages were calculated as a time-averaged value directly from the switching states and duty cycles, this switching process had so far remained undefined.

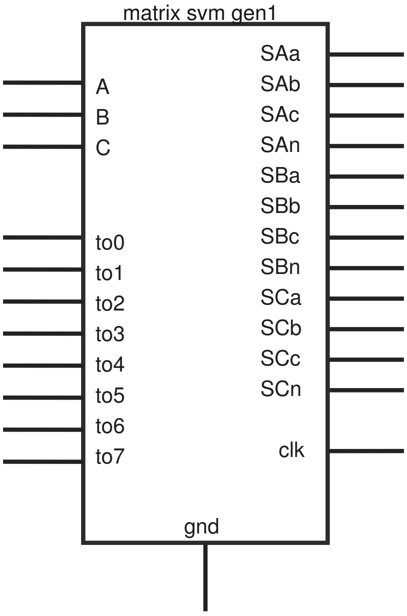


Figure 6.19: Control Block for Saber simulation

Alongside this, when performing the Matlab simulations the only times of interest were those related to the switching states, any unused time within a switching period was simply ignored. These periods are the times when the Zero states are used, and they are those switching states which connect all 4 output legs to the same phase(see Table 5.2), and so the voltage applied across the outputs is 0 for this time. Unlike the Matlab simulations, Saber is a time based simulation package, and it will need to know what to do with the outputs during this Zero state period in order for the simulation to work correctly, just as would be required in a physical converter.

Before choosing the Zero states, and where to place them, it is important to first look at the switching order of the switching states. Referring back at Table 5.2, and once again using the example in used in Chapter 5, it is possible to write down how the input phases are connected through to the outputs for each of the switching states from Table 5.4, which then gives

On examination of the way the phases change between the different switching states when used in this order, then it is easy to see that they are unbalanced, and that output leg **b** does not appear to switch at all for this period, while output leg **a** switches on every change of state. Bearing in mind that a large proportion of the losses seen within a matrix converter will occur during the switching process, when current is forced to flow through a device which is in the process of switching, and

Table 6.9: Switching state output leg input phase

		Output Leg			
		a	b	c	n
Switching State	+1	A	B	B	B
	−2	C	B	B	B
	−4	A	B	A	A
	+5	C	B	C	C
	+16	A	B	A	B
	−17	C	B	C	B

so has a voltage developed across it. So, if the converter is used with the above switching arrangement, the power loss in the devices in output leg **a** will be significantly higher during this period than those in leg **b**. It is therefore an advantage to be able to reduce the number of switching transitions where possible, and especially as in this case as it is as simple as rearranging the order in which the switching states are used.

So, taking the switching order from in Table 6.9 above, these states can then be re-arranged in such a way as to minimise the total number of switching transitions required, and this is shown in Table 6.9.

Table 6.10: Modified switching state order to minimise the number of switching transitions

		Output Leg			
		a	b	c	n
Switching State	+5	C	B	C	C
	−17	C	B	C	B
	−2	C	B	B	B
	+1	A	B	B	B
	+16	A	B	A	B
	−4	A	B	A	A

Alongside the need to reduce the number of switching transitions, to attempt to minimise switching losses in the converter, there is another reason why changing the switching order is advantageous, and that is to do with the harmonic performance of the converter. This is important as the harmonic make-up of both the input current and output voltage will make a large impression on the size and cost of any filters which would be required. As such it is always a good idea to try to keep the amount of harmonics in both the input and output to a minimum. Obviously, with a switching converter like this there will always be large amounts of the switching frequency present in both the input current and output voltage, but due to the relatively high switching frequency, these are easily filtered out. The problem arises with harmonics which are closer to the fundamental frequency of the output, and these get harder to remove the closer they get, so requiring larger capacitors and inductors and increasing both the cost and weight of the converter. So it is a good idea to try to produce an output with as few low frequency harmonics to start with.

In a paper by Zhang[14], which looked at, amongst other things, the harmonic performance of the 4-leg inverter with different switching schemes, it was shown that by using a scheme which had the switch transitions for each output leg symmetrically aligned about the mid point in the switching period, they achieved the best performance of any scheme that they tried. A brief look at the way the 4 legs are being switched in Table 6.10 shows that the switching transitions for each leg are balanced around the mid point, however it is a little more complex in this case as each of the switching states has a different duty cycle. This can be solved by effectively halving the switching period, and mirroring the switching states in the second cycle, and an example of this is shown in Figure 6.20. This does have the effect of doubling the number of switching transitions, although in this case, the trade-off is worth it for the increase in harmonic performance.

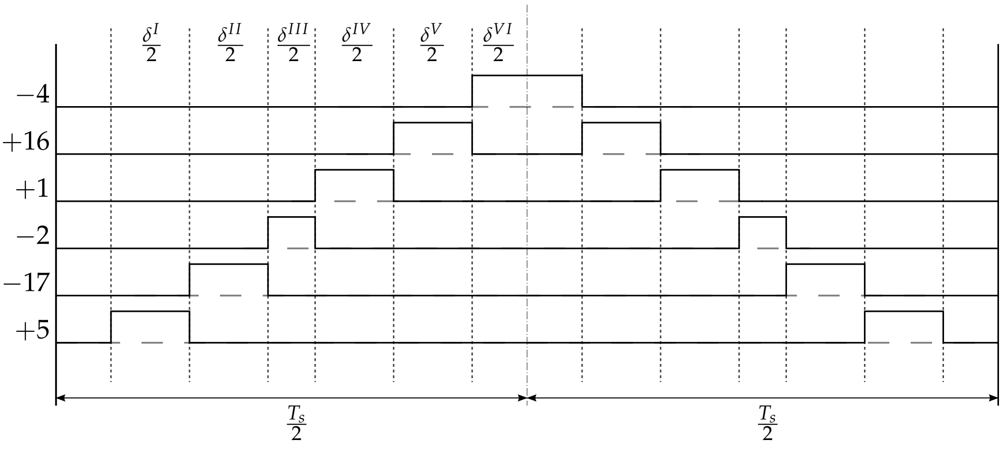


Figure 6.20: A Basic Switching Sequence

Moving to look at the Zero states now, and knowing that at least one Zero state needs to be present to take up any unused time within the period, it can be seen that the duty cycle for the zero state will be defined by

$$\delta_z = 1 - \delta_1 - \delta_2 - \delta_3 - \delta_4 - \delta_5 - \delta_6 \tag{6.2.1}$$

$$t_z = T_s.\delta_z \tag{6.2.2}$$

where T_s is the sampling period. It is entirely possible to place this Zero state at any point within the switching period, but as above, care needs to be taken to ensure that the impact to the harmonic performance, and any increase in switching loss, should be kept to a minimum. Looking back at Zhang[14] once again, it was shown that the best distribution of the Zero states throughout the switching cycle is to have them placed symmetrically around the mid point, and for this converter that can be done in one of two ways.

Looking at the switching sequence in Table 6.10 it can be seen that for the first and last switching states 3 out of 4 of the output legs are all switched to a similar input phase. This neatly allows a Zero state to be placed at each end of the period, with a duty cycle equal to $\left(\frac{\delta_z}{2}\right)$ with only a single switching transition being added each time. In this example state ZC would be used at the beginning of the sequence, and ZA at the end. This now appears to fulfil the criteria in that all the switching edges will be placed symmetrically around the mid point in the cycle, and that there are now two Zero states that will take up the unused time in the switching period. This is called the Two Zero method and is listed in Table 6.11.

Table 6.11: Two Zero Switching State Sequence

		Output Leg			
		a	b	c	n
Switching State	ZC	C	C	C	C
	$+5$	C	B	C	C
	-17	C	B	C	B
	-2	C	B	B	B
	$+1$	A	B	B	B
	$+16$	A	B	A	B
	-4	A	B	A	A
	ZA	A	A	A	A

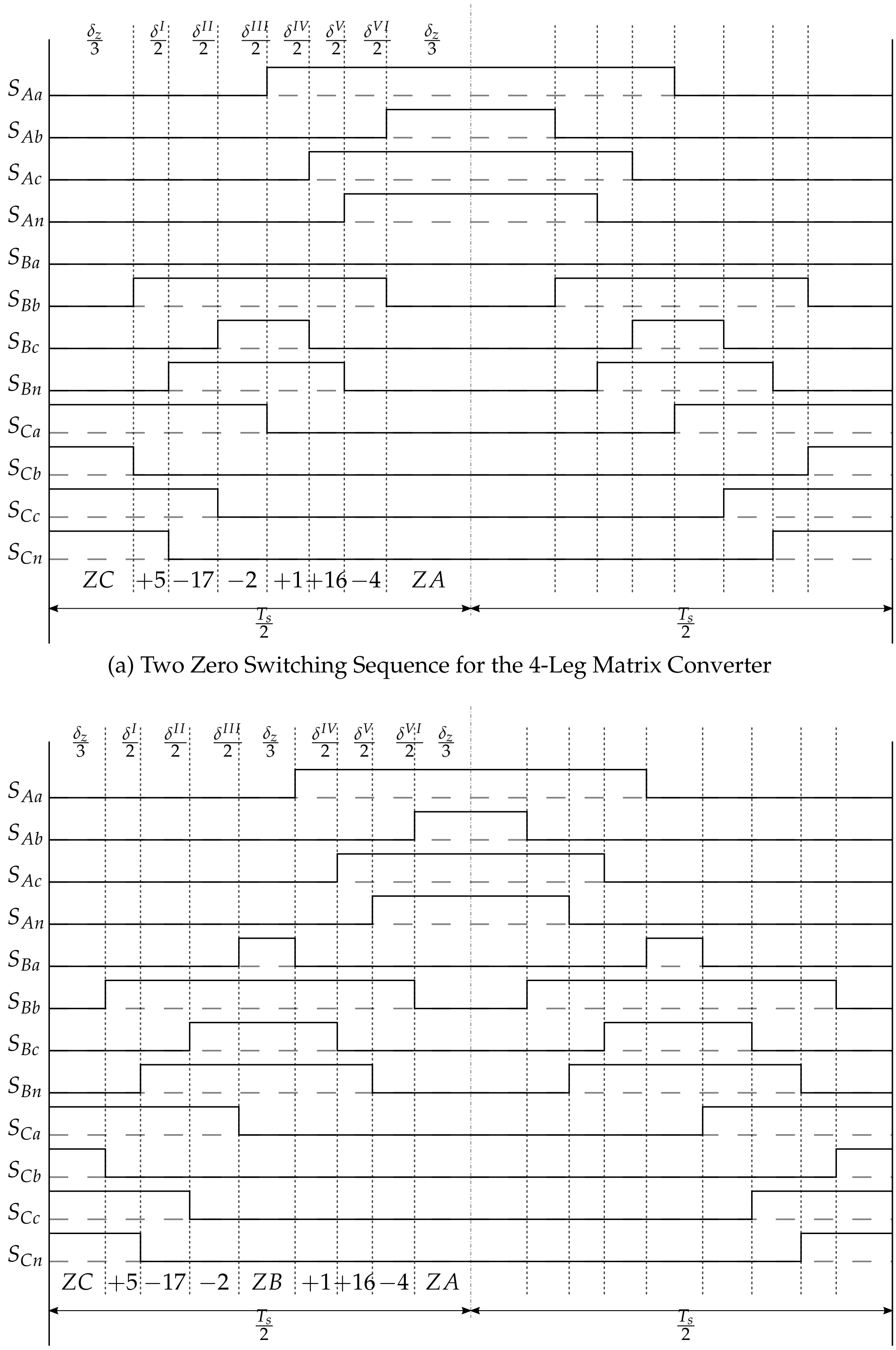
The second method starts off looking at the switching states in exactly the same way, however this time it is also noted that the 3rd ($+1$) and 4th(-2) switching states also have 3 of the 4 outputs switched to a similar input phase, which allows a third zero state to be added into the switching sequence. At first, this seems to not be necessary as everything required is in place for the Two Zero method, however on closer examination of the Two Zero method, if the number of switching transitions per leg is examined, then without this third zero state being present leg **a** has only a single transition compared to all the other legs having 2 transitions. By adding in the third zero state, this balances out the number of switching transitions so that all 4 output legs have 2 transitions per half cycle. This is called the Three Zero method and is listed in Table 6.12.

Table 6.12: Three Zero Switching State Sequence

		Output Leg			
		a	b	c	n
Switching State	ZC	C	C	C	C
	$+5$	C	B	C	C
	-17	C	B	C	B
	-2	C	B	B	B
	ZB	B	B	B	B
	$+1$	A	B	B	B
	$+16$	A	B	A	B
	-4	A	B	A	A
	ZA	A	A	A	A

Examples of the control signals for both of these possible switching sequences are shown in Figure 6.21. Due to all the output legs having the same number of switching transitions in the Three Zero method, it was thought that this would give the better harmonic performance, and so it was chosen to be the initial method used within the simulation, although both types of switching sequence would be investigated.

The final step in the design of the MAST template was to find a way to interface the code which calculates the duty cycles and switching states with the circuit shown in Figure 6.18. The problem that needed to be overcome was this, how to ensure that the control block only performed its calculations once per switching period while still being able to change the control outputs multiple times during this sampling period with high accuracy. The first part of this problem is relatively simple to solve, by using an internal clock which is set to the converter switching frequency, a positive going edge on this clock can be used to trigger the control block into performing the duty cycle and switching state calculations. This does not help with the switching of the control outputs



(a) Two Zero Switching Sequence for the 4-Leg Matrix Converter

(b) Three Zero Switching Sequence for the 4-Leg Matrix Converter

Figure 6.21: 4-Leg Matrix Converter Switching Sequences

however, and it was initially thought that part of the control block would need to be evaluated at every simulation time step to make sure that these were changed at the correct time. This appeared to be a very poor implementation, due to the extra calculations being added at every time step, and so a more efficient solution was sought, and it was then realised that a clock could also be used for this function as well.

When Saber performs the calculations within the control block, simulation time is stopped until all the control block calculations, and any others required at that time step are complete, and so when triggered, the results from the control block appear to be instantaneous within the simulation. Once values for the duty cycles and switching states have been calculated, as time is still stopped, it is simple to load the required values for the first switching state onto the outputs and then set a clock which will trigger when the next switching transition is required. In the example above using the Three Zero method, the first switching state would be Zero state ZC, with timer being set to

$$T_{timer} = \frac{\delta_z}{3} \tag{6.2.3}$$

Once this timer triggers, the new set values for this switching state are loaded onto the outputs and the time to the next switching transition loaded into the clock.

This method has the advantage that the simulation does not need to perform any calculations within the control block at any time other than at one of the clock transitions, to either calculate a new set of duty cycles, or update the output states, and so this will speed up the simulation considerably. This final step completes the MAST template, and a block diagram showing the functions within it is shown in Figure 6.22.

Having already proved that the duty cycle calculations and switching state selection derived in Chapter5 are correct for controlling the converter output voltage, by having simulated and tested them using Matlab in Section 6.1.2, the first step in the Saber simulations was to validate the MAST control block template against these known results. This was done using a series of tests to check on the state of the control block's internal variables.

6.2.1 Saber Simulation Results

The initial testing was used to make sure that the variables representing the input voltage vector were correct, and matched those results found in the Matlab simulations. Figure 6.24 shows the results from this

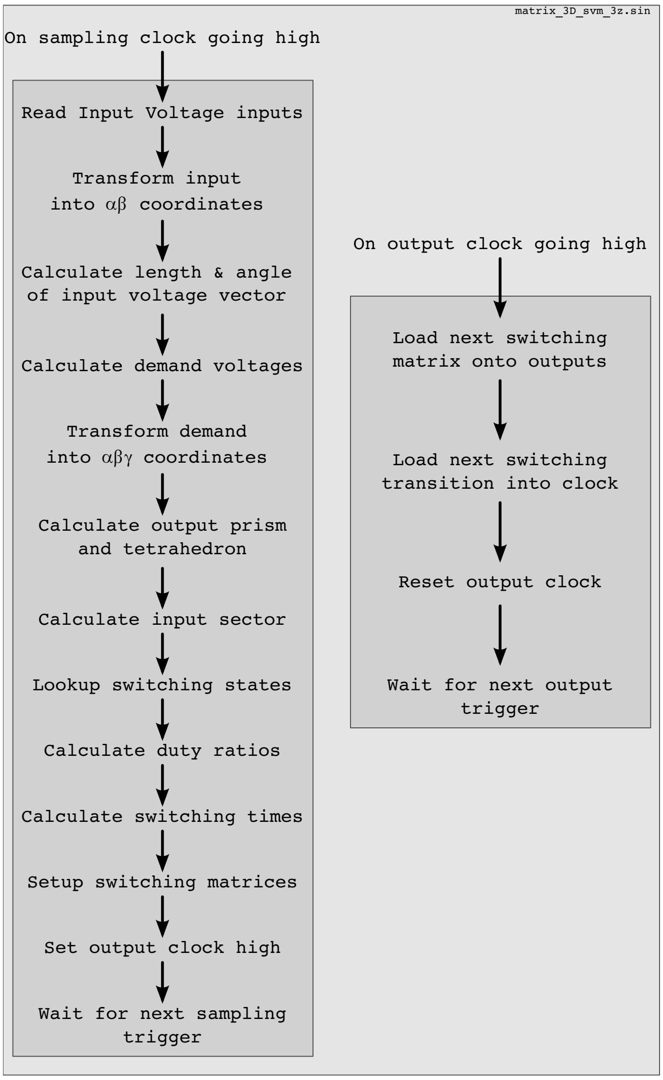

matrix_3D_svm_3z.sin
On sampling clock going high
Read Input Voltage inputs
Transform input into αβ coordinates
Calculate length & angle of input voltage vector
Calculate demand voltages
Transform demand into αβγ coordinates
Calculate output prism and tetrahedron
Calculate input sector
Lookup switching states
Calculate duty ratios
Calculate switching times
Setup switching matrices
Set output clock high
Wait for next sampling trigger
On output clock going high
Load next switching matrix onto outputs
Load next switching transition into clock
Reset output clock
Wait for next output trigger

Figure 6.22: Block Diagram of the MAST template for the Control Block

test where it can be seen that as the input voltage vector angle **beta_in** changes, the input current sector **isector** changes with it in accordance with equation (5.6.1). The changes in both of these are then also matched by how the variable **beta_in_bar** varies, and so by comparing these results to those from the Matlab simulation in Figure 6.12a it can be seen that they match perfectly.

The output variables representing the demand vector were also checked along similar lines, with the results being shown in Figure 6.25. Once again, by comparing the results from this simulation to those from the Matlab simulation, shown in Figure 6.12b it can be seen that the internal variables are working as expected, and match the values from the Matlab simulation. From the results it can be seen that, although the angle that the output vector makes on the $\alpha\beta$ plane is not plotted, the value can be inferred by looking at how the value of **prism** is ascending regularly, as this depends directly its value. This steady rise in output voltage angle is indicative of a balanced 3-phase set of demand voltages, which is indeed the case for this simulation.

It can also be seen in Figure 6.25 how **sign** just alternates between having a value of 2 and 3. When you consider a balanced 3-phase set, then at any one time there are either 2 positive phases, or 2 negative phases, there can never be any other value, so as **sign** is the value

$$\mathbf{sign} = \text{ No. of positive phases} + 1 \tag{6.2.4}$$

for a 3-phase set the value of **sign** will therefore only be 2 or 3, which matches the result in Figure 6.25. This, so far, proves that the operation of the control block MAST template is working in the same way in relation to identifying the input and output vector spaces.

During the previous simulations the duty cycles were also calculated, with a representative set being shown in Figure 6.26, and once again, comparing this with the results from the Matlab simulations shows that the control block is working as expected. The stepped nature of the duty cycle waveforms seen in Figure 6.26 when compared to the Matlab version shown in Figure 6.14a is because the control block duty cycles only get calculated once per sampling period, whereas with the Matlab simulation they were calculated at every iteration step.

With the two different set of internal variables being shown to be correct for these simulations, along with the duty cycles having been calculated correctly, the final stage in the validation of the control block itself is to show that the output voltage from the converter is being generated as expected, and equals the demand voltage. Now this was a relatively easy procedure with the Matlab simulations as the calculated output from the simulations was a time averaged output signal, which to show it was working, exactly matched the demand signal. Now that the output voltage is actually being made up of the space vector modulated output from the converter, it needs a little more care in checking the results.

Figure 6.27 shows a plot comparing the demand voltage V_{dem_a} with the resulting phase-neutral voltage that is generated by the converter between output legs **a** and **n**. This is a valid comparison because the set of demand voltages only contains the 3 values relating to Va, Vb and Vc and due to equation 5.3.2 are therefore all implicitly defined with respect to the neutral leg of the converter. As can be seen, the two plots are exactly in phase with each other, but due to the switching frequency component in the output voltage, as mentioned above, it is very difficult to tell whether the demand voltage for this leg is being met by the output.

To be able to measure this voltage, an fast Fourier transform (FFT) of the output phase-neutral voltage shown in Figure 6.28 was taken and the results of this are shown plotted in Figure 6.29. Now, with a 240V input phase voltage being supplied to the converter, and using the maximum value for q for a balanced set of output voltages, this gives a peak output voltage of

$$V_{dem_a} = \frac{\sqrt{3}}{2}\sqrt{2}V_{in_{rms}} = \sqrt{\frac{3}{2}}.240 = 293.94V \qquad (6.2.5)$$

Reading this value from Figure 6.29, the magnitude of the fundamental frequency is shown to be 293.09V. On checking the other two output phase-neutral voltages, these were also shown to match the phase of their respective demand voltages, along with having very similar spectra to that shown in Figure 6.29. From these it is therefore possible to state that the converter is producing the correct output voltage, and that this results matches up with the results from the Matlab simulations, thus proving that the duty cycle calculations and switching state selection are correct with respect to the output voltage.

Having shown that the simulation is generating the correct output voltages at the terminals to the converter, the resultant voltages which are then produced across the load resistances were then plotted, and these can be seen in Figure 6.30. These show each of the voltages to be almost perfectly sinusoidal, with only a small amount of switching frequency ripple being evident, and this result is backed up by looking at the plot of the FFT of one of these voltages which is shown in Figure 6.31.

The clean spectra for these voltages once again re-enforces the evidence from the spectra and waveforms of the terminal voltages in showing that this space vector modulation technique is working for the 4-leg matrix converter. It shows that the state of the converter switches is correct for each of the required switching states, as the converter is producing the correct voltages at its terminals in order to give the smooth sinusoidal outputs across the load.

So looking now at these switching signals from the control block, Figure 6.32 shows how they change over a single clock cycle, and the effect that they have on the output legs. The gate signals themselves are named so that the input phase is listed first, with the output second, so **SAb** is the switch which connects the input phase **A** to output leg **b**. As can be seen at the start of the cycle all the output legs are switched to input phase **C**, then at the first transition output leg **a** switches to input phase **B** when **SCa** goes low, and **SBa** switches high. Output leg **b** changes on the next transition and so it continues until all of the output legs are switched to input phase **B** at the mid-point in that half-cycle before all switching again to end up all connected to input phase **A** at the end of the half-cycle. As can be seen, this is the Three Zero switching pattern which was chosen for use in the simulations, and just goes to reconfirm that the control block is working correctly.

Further on to looking at the overall pattern of switches for the entire converter, if just a single output leg is looked at, as in Figure 6.33, it can be seen how the output switches between the different input voltages as the switching state changes, which produces the overall waveform shown in Figure 6.34.

Now that it has been confirmed that the simulated converter is working as expected, by both checking that the internal variables are comparable with the Matlab simulations, and also making sure that the output voltage follows the demand voltage, attention can now be focused one of the sections which has not yet been covered in any of the simulations, the input current. Figure 6.35 shows a plot of the input current alongside its respective input phase voltage and it clearly shows how the input current is exactly in phase with the input voltage, giving this converter a unity displacement factor.

Just as for the converter output voltage looked at earlier, to be able to tell if the input current is in fact sinusoidal under the large amount of switching frequency noise an FFT of the input current was taken and the result is shown in Figure 6.36.Once again, it can be seen that there is a large fundamental current at the supply frequency, with a no other discernible frequency spikes until the switching frequency is reached. Figure 6.37 shows this even more clearly, where it plots the 0Hz to 15kHz region of the FFT, but with the y-axis scale re-calculated as a percentage of the fundamental. This plot clearly shows that there is very little low frequency harmonic distortion within the input current.

So, with the output voltage still set at $\frac{\sqrt{3}}{2}V_{inrms}$, and the load resistance being set at 30Ω, this gives a peak input current of

$$I_{A_{peak}} = \sqrt{2}\frac{1}{V_{in_{rms}}}\frac{V_{out}^2}{R} = \sqrt{2}\frac{1}{240}\frac{207.85^2}{30} = 8.48A \qquad (6.2.6)$$

which can be seen ties up well with the simulated result in Figure 6.36. Any small difference with the expected result is due to the voltage drop across the output inductors meaning that the load does not see the full demand voltage across it.

One of the advantages of using a 4-leg matrix converter over a common 3x3 matrix converter is its ability to supply unbalanced loads without causing large voltages fluctuations in the output voltage, and the next simulation looks at the situation where a single resistive load is set to a value of $\frac{1}{3}$ of the other two. Figure 6.38 shows the resultant output currents from the converter, where it is easy to see that one is much larger than the other two showing how the load is unbalanced.

The demand voltage for this simulation is identical to that used in the previous simulations, and this is shown by the set of FFT plots shown in Figure 6.39. This shows that even with an unbalanced load the voltages available at the terminals of the converter remain set to their demand values.

The effect of the unbalanced load on the input current is plotted in Figure 6.40, where it can be seen that while the input current is still in phase with the input voltage, there is a marked difference to the input current waveform shown for a balanced load in Figure 6.35. With the balanced load the shape of the switched current is essentially sinusoidal, however with the unbalanced load, there is an obvious dip in the central portion of each positive and negative half-cycle.

This effect is shown more clearly in Figure 6.41 which shows a plot of the FFT of the input current for the unbalanced load, and shows that as well as the main fundamental current, there are two low order harmonics at 150Hz and 250Hz whose magnitude varies directly with the amount of imbalance in the load. These harmonics are generated at frequencies dependant on both the demand and supply frequencies

$$\text{Harmonics} = 2f_{dem} \pm f_{supply} \tag{6.2.7}$$

and it is these which lead to the distortion of the input current waveform. These low frequency harmonics are extremely undesirable in the input current, and this shows one of the main limitations of the 4-leg matrix converter.

Unlike with a balanced load, an unbalanced load on a 3-phase supply will always draw pulsating power, and because there are no energy storage components within the converter to be able to smooth out the power flow through the converter, they are just passed directly through to the supply. These are seen as the low frequency harmonics in Figure 6.41. So, while the 4-leg matrix converter is perfectly capable of supplying an unbalanced load, the presence of these low order harmonics in the input current limits its use when connected to the normal mains supply.

While the results above prove that the 4-leg matrix converter works for balanced 3-phase sets of output voltages, even when it is powering an unbalanced load, this does not fully prove the operation of the converter. This is because a balanced 3-phase set of demand voltages, when transformed into $\alpha\beta\gamma$ space, always has a γ value of zero, and, in all of the Saber simulations up to this point, the converter has only been operating with a balanced set of demand voltages. This means that it has been operating in a very small region of its overall capabilities, which has, up to this point been entirely within the 'normal' 2-dimensional SVM plane. Unfortunately, although it would certainly be an interesting avenue of work, the active control of the 4-leg matrix converter is outside the scope of this thesis.

However, while it is very extremely unlikely that this particular mode of operation would ever be used, the following simulation was run to be able to demonstrate that the converter itself works when the demand vector trajectory leaves the $\alpha\beta$ plane, and so γ takes on non-zero values. For this simulation run it was decided to reproduce the Matlab simulation, the results of which are shown in Figure 6.8, and see if the Saber simulated converter would be able to produce the same results. For this simulation the demand voltages are set as follows

$$\begin{aligned} V_{dem_a} &= \frac{V_{in_{peak}}}{2}\cos\left(2\pi f_{dem}(t)\right) \\ V_{dem_b} &= \frac{V_{in_{peak}}}{2}\cos\left(4\pi f_{dem}(t) - \frac{2\pi}{3}\right) \\ V_{dem_b} &= \frac{V_{in_{peak}}}{4}\cos\left(2\pi f_{dem}(t) - \frac{4\pi}{3}\right) \end{aligned} \tag{6.2.8}$$

where f_{dem} is set to 100Hz, and $V_{in_{peak}}$ is set to $\sqrt{2}240$.

As to be expected, the results of the voltages developed across the load are not particularly useful, however by looking at the FFT plots of the phase-neutral voltages at the terminals of the converter, shown in Figure 6.42, it is then easy to see that even with these demand voltages that the converter is able to generate them.

Taking the results from this simulation, showing that even with a complex demand trajectory in $\alpha\beta\gamma$ space the converter operates as expected, and then adding with the other simulations shown above, it proves that the theory for the operation of a 4-leg matrix converter using space vector modulation derived in this thesis is valid and correct.

One further set of simulations was also looked at, and those were to do with the comparison between the Two Zero and Three Zero switching sequences. Up until this point, all of the simulations had been performed using the Three Zero switching method, as this had been assumed to be the superior switching sequence based on the results given by Zhang [14]. In Zhang they had found that the best sequence for switching the 4-leg inverter was where all legs were switched, and the switching transitions were symmetrically aligned about the midpoint in the switching period, similar to how it is shown in Figure 6.21, and this was the starting point for both of the switching methods being investigated here.

So, from Zhang, it was thought that the best performance would be from the method which provided the same number of switching transitions in all 4 output phases per half-cycle, and so while the Two Zero method has overall fewer switching transitions one of the phases will always have 2 less changes than the other 3, and so the Three Zero method was thought to be best. Figure 6.43 shows a plot of the FFT for the output phase-neutral voltage when using the Two Zero sequence, while the FFT for the Three Zero sequence is plotted in Figure 6.44. It should be noted that the Three Zero spectrum shown here is different from that shown in Figure 6.29 as the demand voltages for each is different. The reason this needed to be done for this comparison is that as q goes up, the sum of the switching state duty cycles δ^{I} to δ^{VI} approaches 1, and so the amount of time that the Zero states are used approaches 0. As this happens the differences between the two switching sequences shrinks and the spectra then look identical. So, for these simulations, the q of the converter was set to 0.5.

Looking at the two spectra it is easy to see the main differences between them:

- the group of frequencies at around the sampling frequency are smaller for the Two Zero method
- the group of frequencies at around twice the sampling frequency are about half the magnitude in the Three Zero sequence

In both cases, the performance at low frequencies, well below the sampling frequency, was excellent.

These results were a little surprising as it was expected that the Three Zero method would have the superior performance around the sampling frequency, and it was thought that it would be higher around 2 and 3 times the sampling frequency, however, it was thought that the superior harmonic performance throughout the entire range meant that the Three Zero Method was kept and used for the build of the converter itself.

Once again, the results shown here back up the previous simulations, and this is yet more proof that the derivation of the equations for Space Vector Modulation of the 4-Leg Matrix Converter are correct.

6.2.2 Further Saber Simulation Work

Now that a basic set of simulations had been performed, which had shown that the modulation technique was working correctly, a further set of simulations were used to look into some of the possible issues which could arise with the physical implementation of the converter.

6.2.2.1 The Effects of Processing Delays and Quantisation Errors

The first stage of this investigation was to look into the effects on the modulation process of the constraints that would placed upon it by the use of a digital signal processor (DSP) for performing the calculations, and a field programmable gate array (FPGA) for controlling the switching of the devices. The actual design of these elements is described in detail in Chapter 7, but for these simulations a number of assumptions needed to be made about the implementation. The assumptions made are as follows:

- The switching frequency for the converter would be 12.5kHz
- The FPGA clock frequency would be 50MHz
- The A/D converters would have be 14 bit resolution

Each of these three assumptions introduces a different limitation into the previously ideal nature of the simulations, and each of these will be looked at in turn.

With the current ideal nature of the simulation, at any point in time the software performs all the necessary calculations it needs at that time point before moving onto the next time step. This means that while the input voltages are actually sampled, the calculation of the switch timings is all performed at the same instant the voltages are sampled, right at the start of the switching period where they will be used. In practise with a real DSP however, the input voltages will need to be sampled, and the switch timings calculated, during the preceding switching period to allow the correct switching right from the very start of the switching period. As can be seen in Figure 6.45, there is a distinct delay between the actual input voltage waveform and that used to calculate the switching states. This introduces a problem which is that with a changing set of input voltages, which we have by definition in a matrix converter, the input voltages will be different from one sampling period to the next. Now, while this change may not be very large with the relatively fast switching frequency being used in this converter, there will still be an error introduced by this.

The most straightforward method to counteract this error is to simply calculate the difference in input voltage phase between the previous sampling period and the current one, and then to just add this angle onto the current input phase to give the phase at the beginning of the next sampling period. However, this would mean that the voltages matched at the start of the switching period, and diverged after. By adding on another half of the calculated angle, this sets the sampled and actual voltages to match half way through the switching period, thus minimising the error as seen in Figure 6.46.

This is the method what was initially used and the results can be seen in Figure 6.47 showing the FFT of the output voltage with and without compensation. As can be seen, even though the effects of this processing delay are minor anyway, the compensation reduces these effects further on the output voltage.

However, there is a possible problem with this approach due to the assumption that the sampled input voltage space vector rotates at a steady speed. The problem is that as soon as there is noise on the input voltage then the instantaneous phase of that voltage no longer changes at a steady rate, and so the simple calculation of the phase at the next sampling point is no longer necessarily valid. This becomes especially important once an input filter is added to the circuit, as any imbalance in the load will cause harmonics in the input current, and thus distortion in the input voltage to the converter. To try to counter the problem of a noisy or distorted input voltage, another approach is required, and in this instance a phase-locked loop (PLL) is used to track the fundamental frequency component of the input voltage.

The use of a PLL means that the phase of the fundamental frequency of the input voltage can be tracked easily tracked, and so allows the accurate calculation of the phase of the input voltage at the next sample period despite the presence of any noise of distortion that may be present. As can be seen from the waveforms in Figure 6.48, the output straight from the PLL tracks the fundamental of the input voltage, and this happens even when there is significant distortion present.

However, because of the ability of the PLL to track the fundamental frequency of the input, it does have a disadvantage. Depending on the type of distortion, it can lead to slightly higher distortion levels at the output when compared to the direct sampling method, however as will be shown later in Section 6.2.2.3, the PLL does have other advantages. Figures 6.49 and 6.50 show a comparison of the output voltage for the two methods with the same noisy input signal, and in this case the output using the PLL is less distorted than that using the Delta method. The reason for this is that as the voltage changes, the direct sampling method takes into account the change in voltage, whereas when using the PLL the sampled voltages are filtered, and so any change would take a number of cycles to propagate through to the calculations. For lower frequency distortion, where the sampled input voltage doesn't vary greatly between sample points, the Delta method is able to track the distortion and adjust the switching times accordingly. For higher frequency distortion and noise, where the change in sampled input voltage is greater between sample points, the Delta method is not able to track the distortion so well, and the PLL does a better job.

Either of these methods will work for reducing the effects of the processing delay, but the PLL has been chosen for this converter due to its better ability to deal with noise, alongside this there are other reasons which are described later in Section 6.2.2.3. For these simulations, the PLL used is implemented in software, and is exactly the same as that used in the actual design of the converter. It's design is comes from the University of Nottingham's 3x3 matrix converter upon which this 4-leg converter is based, and was not covered in this work, so the design of the PLL is thus outside the scope of this thesis.

Looking next at the limitation imposed due to the clock speed of the FPGA, and this limits the accuracy with which the switching times, calculated by the DSP, can actually be sent out to the switches. The limitation comes about because the period of the FPGA clock defines the smallest unit of time that it is possible to use within the system. In this case, with a 50MHz FPGA clock, this gives a clock period of 20ns, and so all switching times generated need to be multiples of this value.

Now, while this is indeed a short period of time, with a sampling frequency of 12.5kHz, this only gives 4000 FPGA clock cycles for a single switching period, and so it is this which limits the accuracy which the switching transitions can be performed. This can somewhat mitigated by calculating the difference between the ideal and the actual switching points, and then adding this error into the switching times for the next sampling period, however this is still not ideal. Thankfully though, this effect is quite small, and the difference between the ideal and the actual output voltages can be seen in Figure 6.51.

The final limitation that was investigated is because of the fixed word length of the analogue to digital converters, which are used to sample the input voltages. In this case the A/D converters are 14 bit, which means that with a maximum input voltage of 400V the A/D converters are have a resolution of 0.048V per bit. And so the effect of this quantisation error is negligible on the output voltage.

Figure 6.53 shows the results from a simulation which applies all three of these sources of error together, along with the various methods used to eliminate them, and as can be seen the difference in the results from the ideal is negligible.

6.2.2.2 Simulating Device Commutation

In all the simulations up to this point, all of the switching transitions have been ideal in nature, with no attempt at commutating between the incoming and outgoing devices. Obviously when building an actual converter, this would not be possible and so, as discussed in Chapter 4, one of these possible commutation techniques needs to be used. For this converter it was decided that the 4-step commutation process would be used.

As described in Section 4.3, the 4-step commutation process, as it's name states, commutates the current from the outgoing bi-directional switch to the incoming one in four discrete steps. Firstly the non-conducting outgoing device is turned off before the incoming conducting device is turned on. By turning off the non-conducting device first, this ensures that there is no short-circuit current path between the incoming and outgoing input phases. The next step in the process is to turn the conducting outgoing device off, before finally switching the non-conducting incoming device on.

While this is straightforward in principle, there are a number of practical problems which needed to be overcome before this approach could be simulated. Firstly, the switching of each device takes a certain length of time, and with this converter using IGBTs to do the switching they would have the fast turn-on and slower turn-off times which are typical for this type of device, and for the simulation the following were defined:

- $T_{on} = 50ns$
- $T_{off} = 300ns$

With these timings a suitable dead-time between each stage of the commutation process needs to be chosen, and in this case, taking into account the length of the turn-off period, a dead-time of 500ns was chosen. While initially this appears to be a short period of time, over the whole commutation process this adds up to 1.5μs for a single switching transition and when the entire sample period is only 80μs long, this switching time is a significant proportion. On top of this, depending on the direction of current flow and the incoming and outgoing input phase voltages, the actual switching, where the current transfers from one bi-directional switch to the other, can occur at either the second or third stage in the commutation process. Due to this, unless some form of compensation is used, the actual switching instant will be either 500ns or 1μs late, which will then cause distortion in the output voltage.

On top of this issue, the length of the overall commutation process could also limit the minimum possible pulse width, which would severely impact on the operation of the converter, and the reduce the possible range of output voltage that it is possible to produce.

However, it is possible to overcome both of these issues by the use of a simple compensation process, and also by designing the commutation within the simulation so that it can handle pulse-widths shorter than the overall commutation length. The technique used to compensate for the dead-time is to always start the switching process 1μs early and this allows the commutation block to sample the incoming and outgoing phase voltages, along with the current direction. Then, depending on which stage of the commutation process the actual voltage switching takes place on, the commutation block can either start the process immediately, or wait for 500ns before starting the process. This ensures that irrespective of the relative input voltages, or the current direction, the actual switching point where the current commutates from one switch to the other always takes place at the correct time.

Figures 6.54 and 6.55 show both of these switching processes in action. In Figure 6.54 the delayed start to the commutation process can be easily seen, ensuring that the output voltage switches between input phases at the right point, while in Figure 6.55 the commutation process starts immediately, also ensuring that the voltage switching happens at the correct point.

By using this commutation it means it has almost no impact on the output voltage which is generated, and this can be seen in Figure 6.56. Unfortunately there is one part of the commutation process which cannot easily be compensated for manner, and this is at the very start of each SVM period, where the initial zero vector is switched onto the outputs. If this zero vector is going to be held for less than 1μs then there is not enough time for the dead-time compensation to work correctly, and so this will therefore limit the minimum allowed zero time during any one switching period. By limiting the minimum zero time in this fashion, it has a knock on effect on the maximum possible voltage transfer ratio, which occurs when the zero-time within a switching period equals 0. Now, the percentage of zero time in a switching period gives a direct translation into the maximum voltage transfer ratio. So with the 3μs minimum of dead time required when using this compensation scheme, this being 3.75%, which means that the maximum voltage transfer ratio is reduced to 96.25% of theoretical maximum. This is true whether the output is balanced of unbalanced.

It would however be possible to design a compensation scheme which was able to account for this issue, but with the way that the converter has been implemented within the Saber MAST language, this would not be a simple task. So while it would be possible it was felt that, as the scheme which has been implemented above was perfectly acceptable for the purposes of this work, that this additional work was not required.

6.2.2.3 The Effects of Supply Impedance on Converter Operation

With the device commutation scheme in place and working, the next stage in the investigation was to look at the effect that the introduction of a typical supply inductance has on the operation of the converter. As mentioned in Section 6.2.2.1 earlier in this chapter, this is especially important during unbalanced operation. This is where the current drawn from the supply is not purely at the fundamental frequency of the input voltage, and there are also harmonic components related to the fundamental frequency of the unbalanced output voltage.

This issue is important to investigate because the presence of these harmonic currents in the input can cause the input voltage at the converter terminals to become distorted if there is any inductance in the supply. Unfortunately there is always some inductance in the supply network, and so the operation of the converter needs to be checked to ensure this distortion does not cause a problem.

There is also another problem with having serial inductance in the supply which needs to be overcome as well, and that is down to the fundamental operation of the converter. In the initial assumptions about the operation of the matrix converter it was stated that the input port was voltage stiff, while the output port was current stiff. This means that at the output of the converter, the voltage can change instantaneously from the switching between input phases, but the current is continuous and can not change instantaneously. The opposite is true for the input port, where the current can instantly switch between different phases, while the voltage cannot.

So, when considering the inductance of the supply into the converter, it should be remembered that the current through this inductor cannot change instantly, but this is a requirement for the correct operation of the converter, and there are two main problems related to this. The first being what happens when you switch away from a input phase which is carrying current, with the second being when you the output is drawing current and you switch to another input phase.

The first situation is where the converter tries to instantly switch the output current away from one input phase to another. The inductance in supply then tries to keep this current flowing, and so the voltage at the input to the converter could quickly rise, or fall, depending on the direction of current flow. This rapid change in voltage across the device could possibly become larger than the breakdown voltage, which would then end up destroying the device, and disabling the converter.

The second situation arises where there is a no current flowing in the supply inductor in the incoming phase, so when it is switched to an output leg where current was previously flowing the voltage at this point will once again change suddenly to force current to flow. This has two effects in this case, firstly it will distort the voltage and current at the output of the converter, and this change in voltage also has the possibility, as before, of becoming larger than the breakdown voltage of the devices.

One way to counter these problems is to add a rectifier circuit which sits in parallel to the converter, and this arrangement is shown Figure 7.7. This allows any current still flowing in the supply inductance when that phase is no longer connected to the output to just harmlessly flow through the diodes of the rectifier and into a capacitor where it can be safely dissipated. This arrangement will also be able to supply current in the second situation, but this can also cause some voltage distortion at the input to the converter until the load current has fully transferred to the supply.

Another way to counter this problem, which do not have the distortion problems of the parallel diode bridge, is to add some parallel capacitance to the input of the converter as shown in Figure 7.8. Like the diode bridge before, these capacitors are used to handle any current still remaining in the input inductance once a phase has been switched out, while also being able to supply current to an incoming phase before the supply can take over.

Now, obviously with the capacitors arranged like this with the supply inductances they form a passive second order low-pass filter, which in any real application would be required to filter out the high-frequency switching currents from the supply, and so this arrangement works well to solve both of these issues. In a commercial application a more extensive filter than this would be used, with many more filter elements, but for the task of investigating the behaviour of the converter the simple arrangement with the supply inductance and input capacitance is perfectly adequate.

In the design of the actual converter, which is covered in Chapter 7, it can be seen that both the rectifier and the input capacitance are used, however the parallel diode bridge is only really there to provide protection for the IGBTs in case of a problem with the converter. And so, for this investigation the parallel diode rectifier is ignored, and only the effects of the input filter are looked into.

As was discussed in Section 6.2.2.1, one of the methods used to help with the implementation issues of the converter is a phase-locked loop (PLL), which, because it tracks the fundamental input frequency, may well have some influence over the operation when combined with this input filter. So, part of the work looking into the effect of the input filter it also be to see what difference the PLL made.

Figure 6.23 shows the Saber circuit used for the initial simulation work with the input filter, which now includes the switch commutation but at this point there is no PLL.

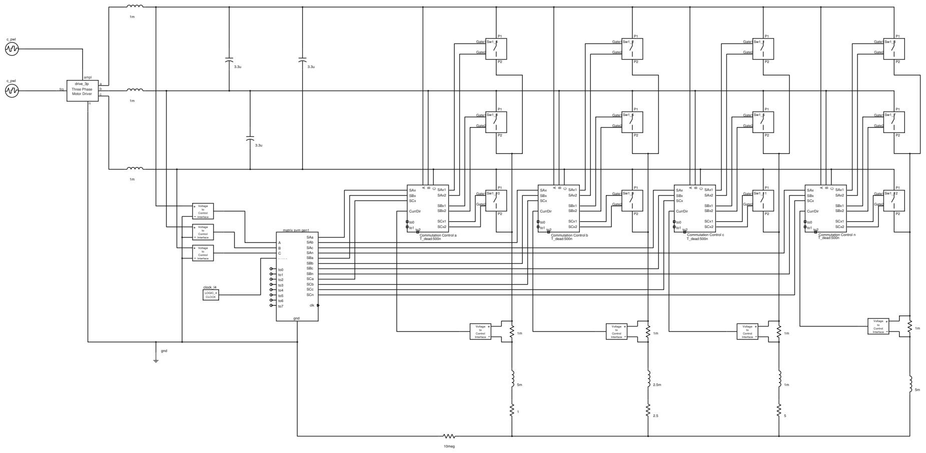


Figure 6.23: Circuit Diagram of the 4-Leg Matrix Converter for simulating the effects of the Input Filter

Now, assuming a typical value of 1mH for the supply inductance, then, using equation (6.2.9) for the resonant frequency of an LC circuit, a 3μ3F capacitor would give a resonant/cut-off frequency of 2.9kHz, which is well below the switching frequency.

$$f_o = \frac{1}{2\pi\sqrt{LC}} \tag{6.2.9}$$

Figure 6.57 shows what happened to the input voltage when this circuit was simulated with the following unbalanced load, which was chosen to ensure that the unbalance not only resulted in giving different load currents, but also that the displacement factor for each output phase would also be different.

$$\begin{aligned} L_a &= 5mH \\ R_a &= 1\Omega \\ L_b &= 2.5mH \\ R_b &= 2.5\Omega \\ L_c &= 1mH \\ R_c &= 5\Omega \end{aligned} \tag{6.2.10}$$

As can be seen in Figure 6.57, the input current, and so also the voltage at the converter terminals is very distorted, and it appears that there is an interaction between the filter and the converter. In this instance, as a consequence of the distortion, the operation of the converter fails as the distorted input voltage drops below that necessary to maintain the demanded output voltage.

This interaction with the filter can be reduced and removed by lowering the LC resonant frequency by increasing the capacitor value to 47μF, and as shown in Figure 6.58 it is possible for the converter to operate correctly with a large enough input filter, however this introduces another problem. One of the objectives with a matrix converter is to be able to produce an input current with a displacement factor of 1, where the fundamental component of the input current is in phase with the supply voltage. Now, when using a filter which does not cause any resonance issues with the converter, the cut-off frequency is then low enough to have a significant effect on the phase of the fundamental component of the input current with respect to the input voltage which can be seen in Figure 6.58.

However, it was found that by incorporating the PLL from Section 6.2.2.1 into the converter once again, the problems with interaction with the input filter disappear, and this is shown in Figure 6.59, which shows the converter operating with the same input filter as was used in the initial simulation in Figure 6.57 above, and with both using the same unbalanced load.

The reason for this behaviour appears to be down to the basic ability of the converter to generate the output voltages from the instantaneous input voltages, alongside this, the converter will then produce an input current in phase with the input voltage. So, if the input voltage seen by the converter includes a harmonic component, then the converter will go on to produce a input current in phase with this which also contains this harmonic component, and this current will then cause further distortion. So, as the unbalanced harmonic current starts to be drawn in the above simulation, the input voltage waveform starts to deform as the harmonic component is below the input filter cut-off frequency. Then as input voltage deforms, the converter adjusts its switch timing to take this into account, and by doing this adds this harmonic element into the current which is being drawn by the converter, thus re-enforcing the voltage distortion.

However, when the PLL is being used in the circuit, this tracks the fundamental frequency of the input and forces the converter to always perform it's calculations based on the ideal PLL waveforms. So whereas previously, the converter automatically adjusted the switching to take into account the initial distortion, the converter using the PLL does not, and therefore the does not end up re-enforcing the distortion. The downside of this is that because the converter is no longer tracking the exact set of input voltages, but rather the ideal PLL set, then any distortion which is present on the input to the converter can end up passing through the converter and leading to some distortion in the output voltages. Figure 6.60 shows the FFT of the output voltage across the load, and it can be seen that there is slightly more low frequency distortion than in the ideal case.

6.2.2.4 The Effects of Changing Input Frequency

Back at the beginning of Chapter 1 it was proposed that one possible application for the 4-leg matrix converter was as part of a generator-based standalone power supply, where the generator would be driven by a variable speed motor, and the 4-leg matrix converter providing a 3-phase plus neutral supply. A diagram of this is shown in Figure 1.1.

For this type of generation scheme to be able to work, it needs to be shown that the converter is able to produce a fixed set of output voltages while the input voltage and frequency is changing, as would happen when being driven by a variable speed generator.

According to the theory set out in Chapter 5, the converter works by taking the instantaneous input voltages and calculating the switching vectors and times based on these. So, irrespective of the input frequency, and as long as the input voltage meets the voltage transfer requirements, then a changing input should not upset the operation of the converter. To prove this a number of simulations were performed using the circuit as would be implemented in Chapter 7, including the commutation, PLL and input filter. Initially these simulations were preformed with a balanced load on the output, but the final simulation will be used to show that using an unbalanced load does not effect the outputs.

The initial simulation was to show the effect that a changing input voltage level would have on the converter output, and the results are shown in Figures 6.61 and 6.62. Figure 6.61 shows the changing input voltage compared to the output, and as can be seen the output stays constant for the entire time. Also included in Figure 6.61 is the input current, and this shows that over the entire range the converter meets the criteria of having a unity displacement factor. Figure 6.62 shows the output voltage alongside its FFT, and as can be seen there is no effect from the changing voltage.

The next simulation kept the voltage level steady, but changed the input frequency instead, with the results being shown in Figures 6.63 and 6.64. Figure 6.63 shows that as the input frequency changes there is no effect on the output, and also shows that the converter keeps the input current in phase with the input voltage. Figure 6.64 shows the output voltage along with its frequency spectra, and as can be seen there is no real effect from the changing frequency.

The next step was to combine the previous two simulations such that both the input voltage and frequency change together, and the results for this simulation are shown in Figures 6.65 and 6.66. In Figure 6.65 the change in both voltage and frequency can easily be seen in the input, while the converter still holds the output steady. Also, as before, the converter keeps the input current in phase with the input voltage. And as before, Figure 6.66 shows the output voltage alongside its FFT, and again, there is no effect on this despite both the changing input frequency and voltage.

The final simulation in this group was to change the load from being balanced to being unbalanced. The same unbalanced load as in Section 6.2.2.3 above was used, as detailed by in (6.2.10). The results for this simulation are shown in Figures 6.67 and 6.68, and once again, despite the changing input conditions and the unbalanced load, the output of the converter remains constant, with the fundamental component of the input current still being in phase with the input voltage.

These results show that the proposed application from Chapter 1 is one which the 4-leg matrix converter is capable of performing.

With the simulations now complete, it allowed the work to progress onto the next stage, to build a converter to allow the technique to be demonstrated.

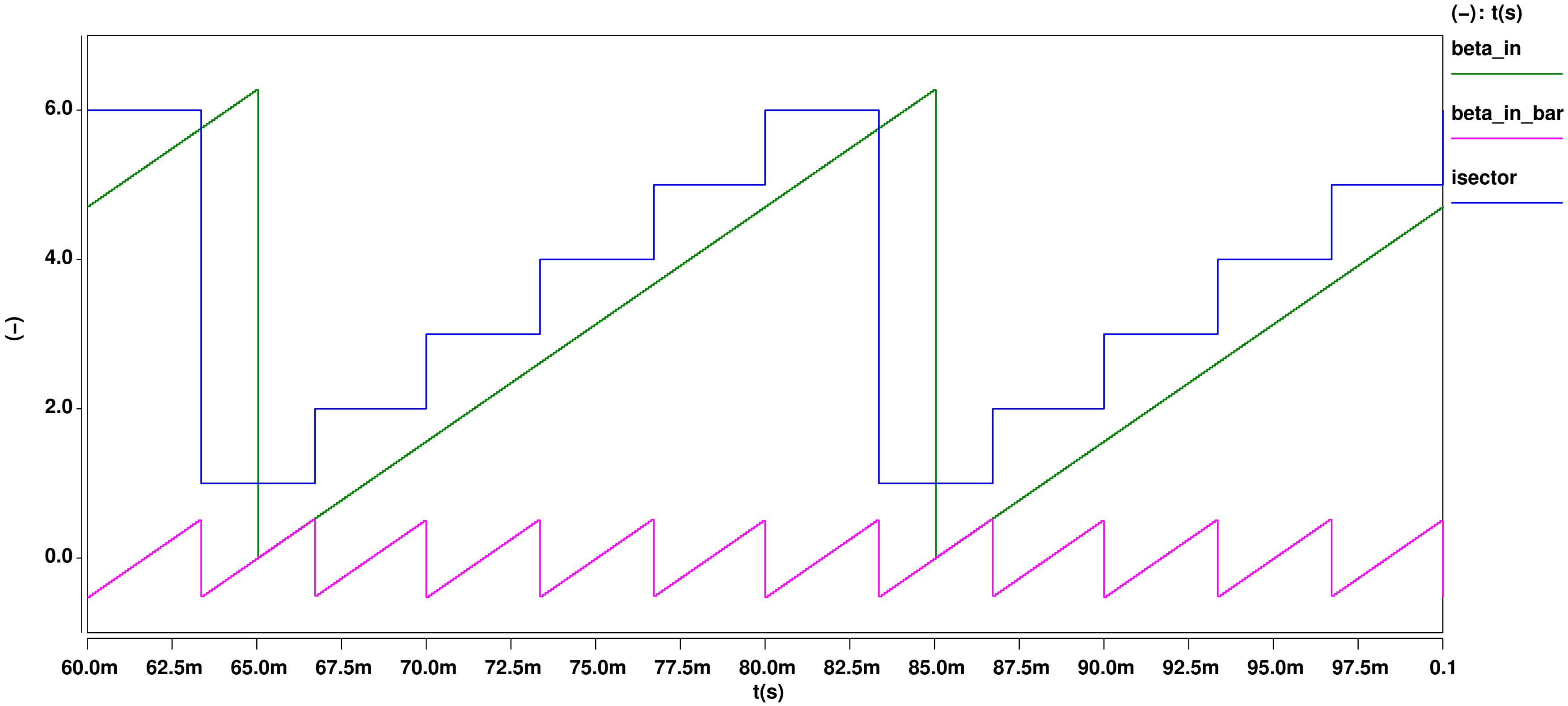


Figure 6.24: Plot of Control Block Internal Input Variables
50Hz Supply Frequency- 12.5kHz Sampling Frequency

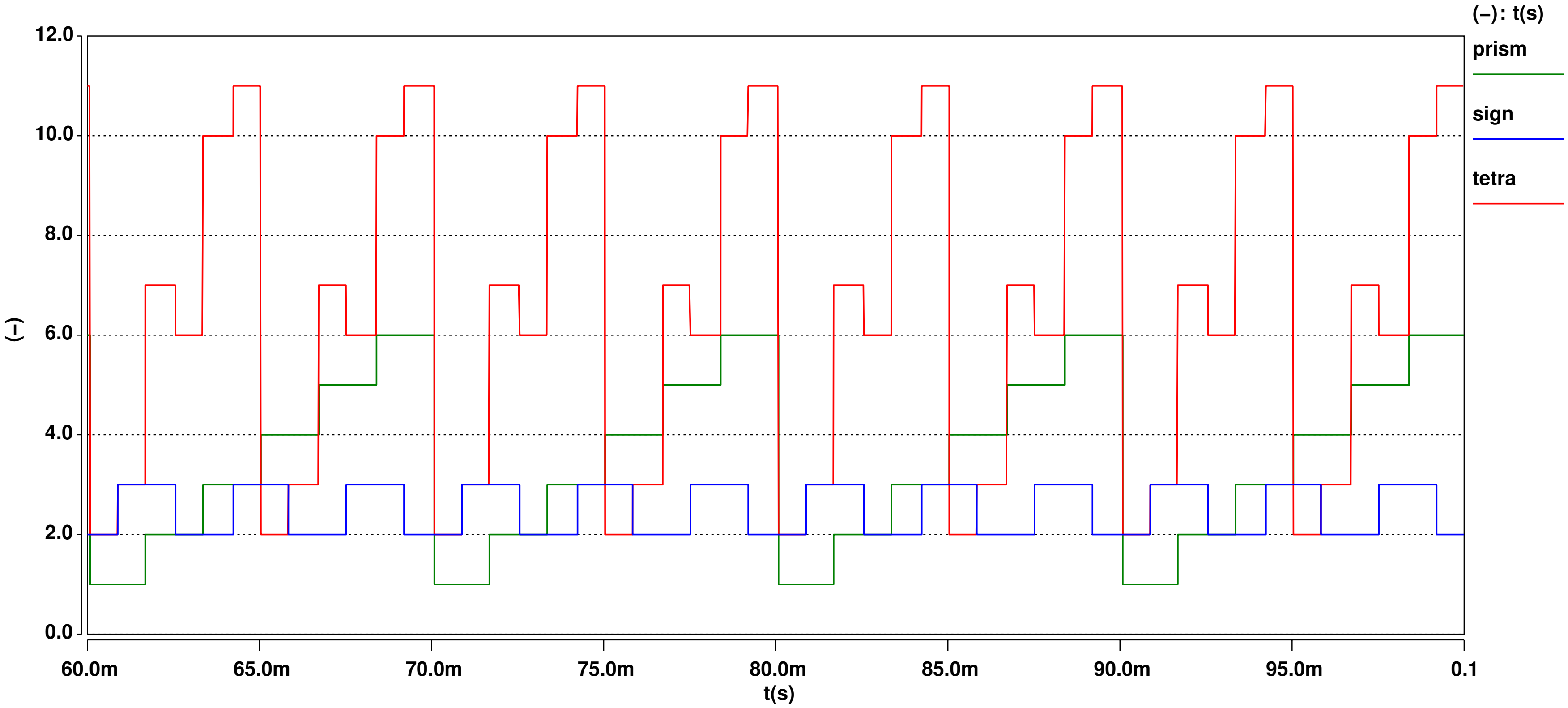


Figure 6.25: Plot of Control Block Internal Output Variables
100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

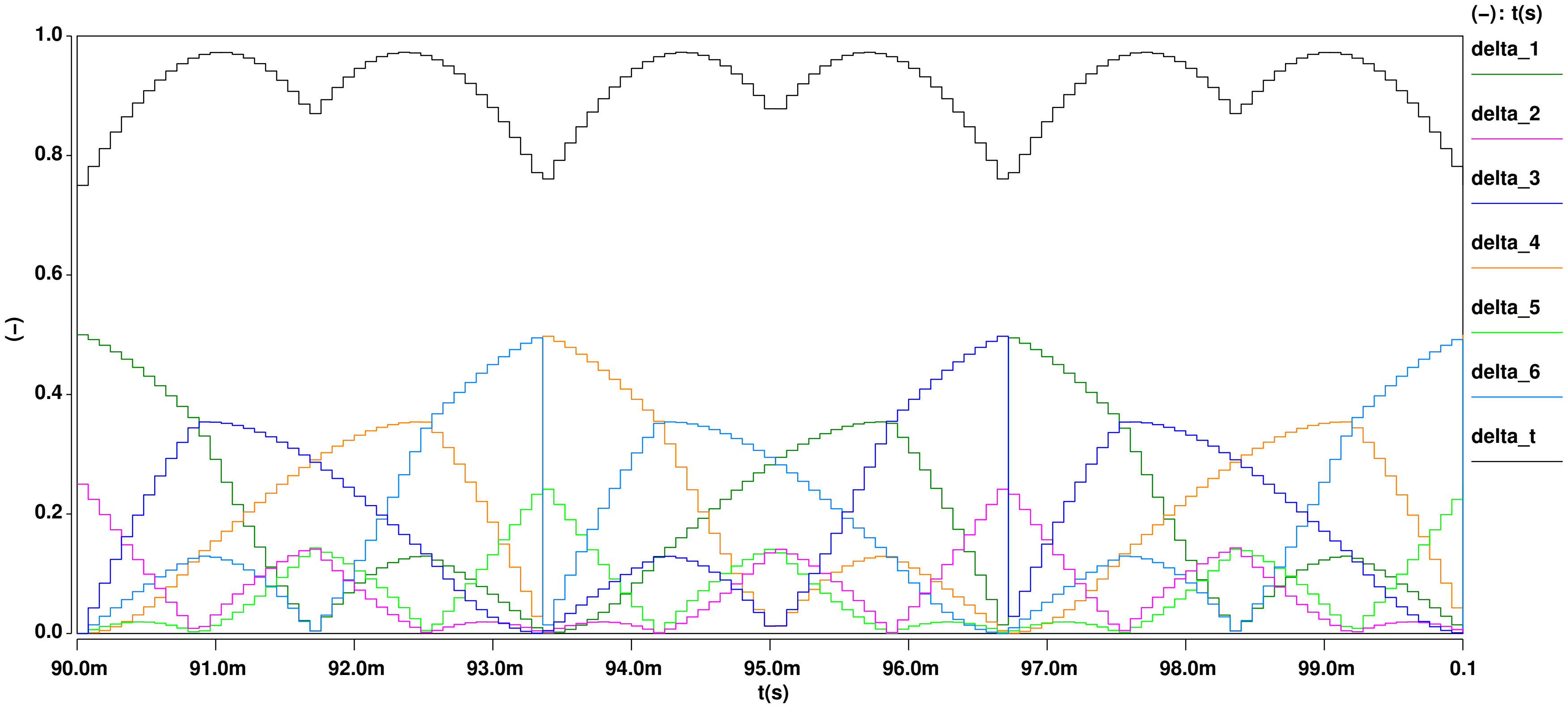


Figure 6.26: Plot of Control Block Switching State Duty Cycles
100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

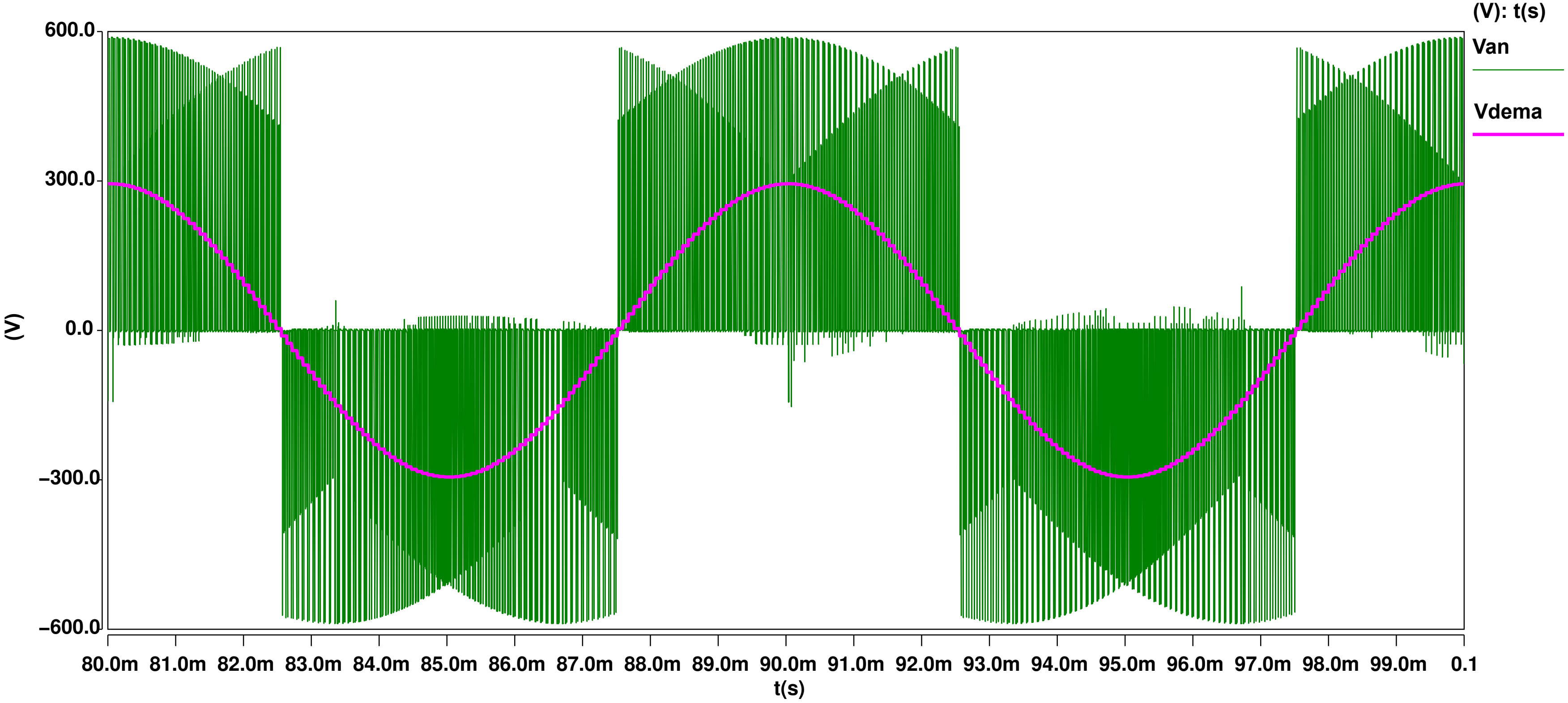


Figure 6.27: Plot of Demand and Output Phase-neutral Voltage for Leg **a**
100Hz Demand Frequency- $q = \frac{\sqrt{3}}{2}$ - 12.5kHz Sampling Frequency

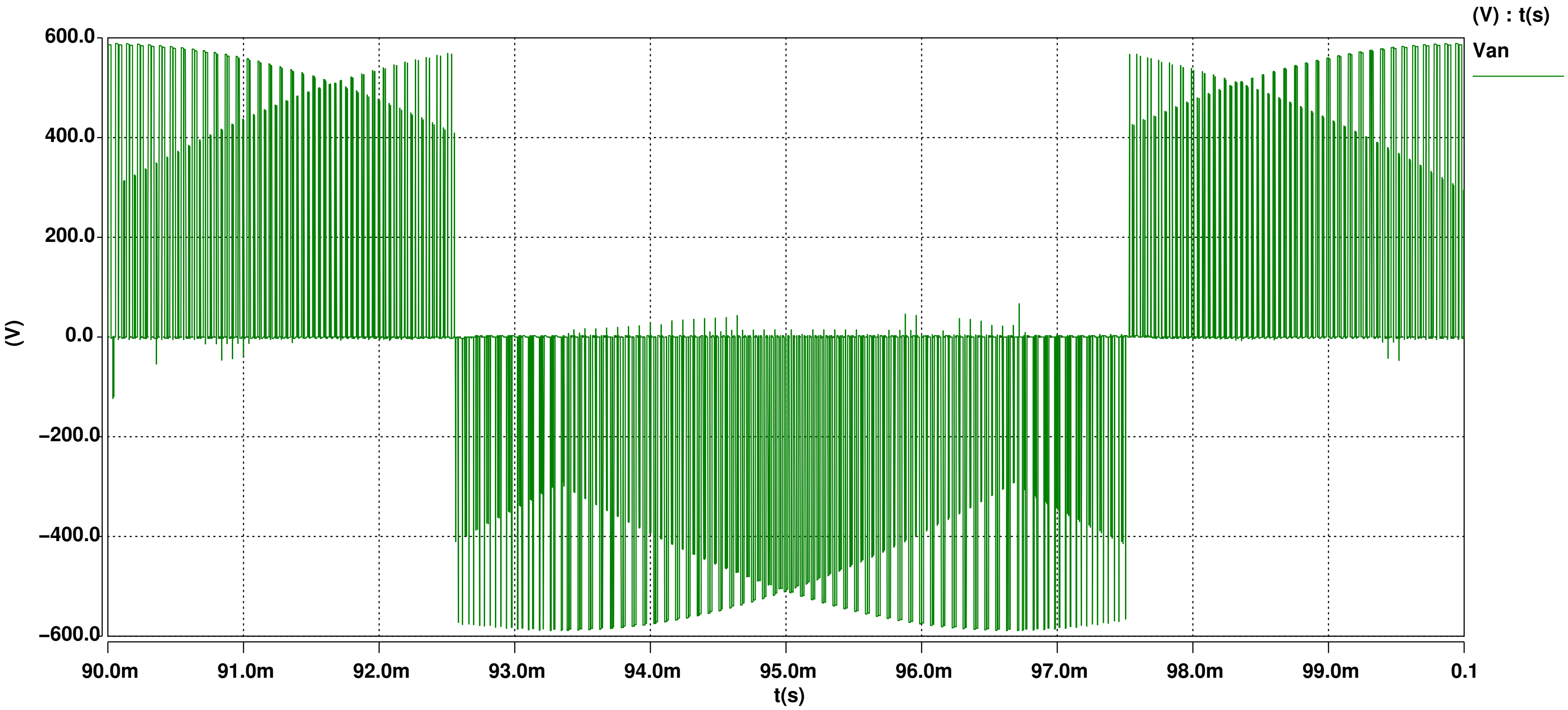


Figure 6.28: Plot of Output Phase Voltage for Leg **a** referenced to Leg **n**
100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

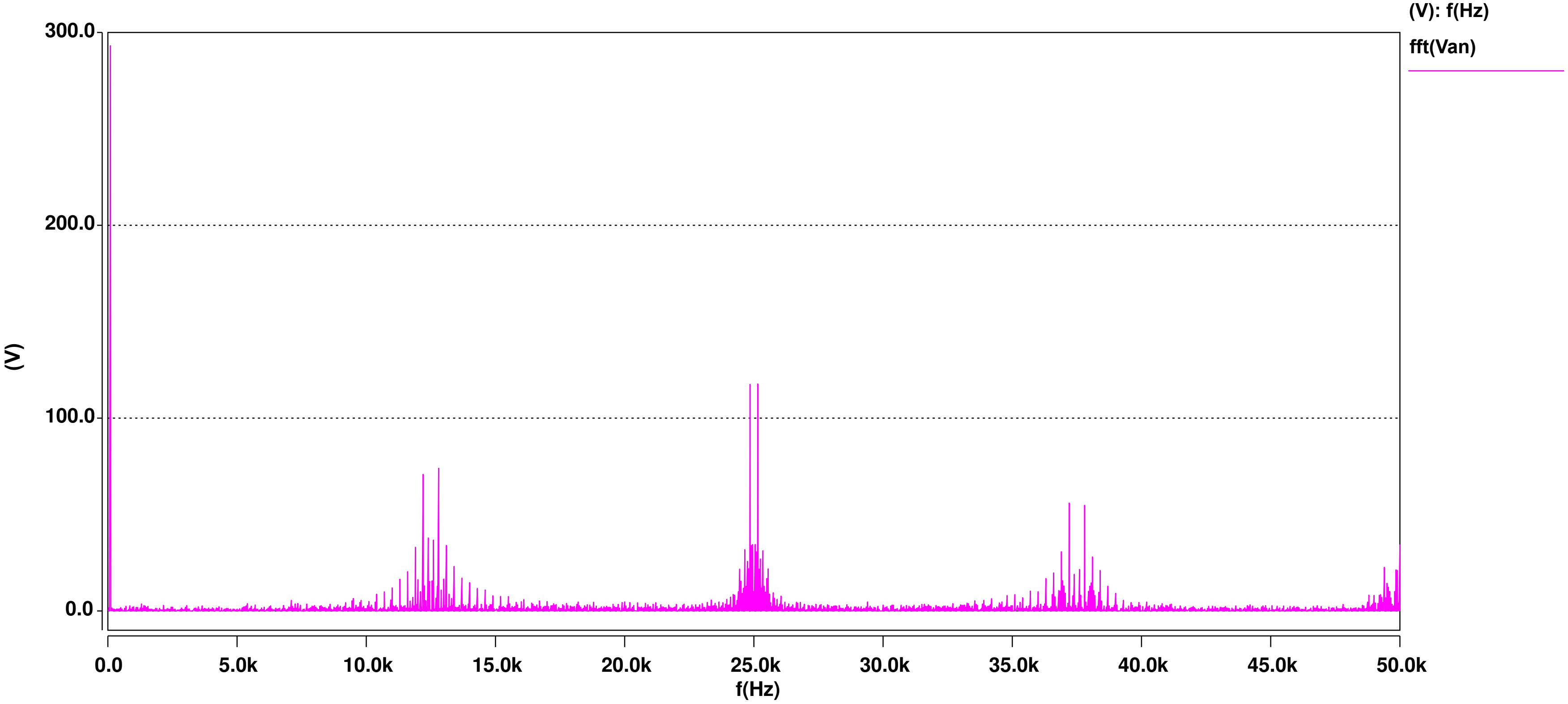


Figure 6.29: Plot of the FFT of the Output Phase-neutral Voltage on Leg **a** with respect to Leg **n** 100Hz Demand Frequency- $q = \frac{\sqrt{3}}{2}$ - 12.5kHz Sampling Frequency

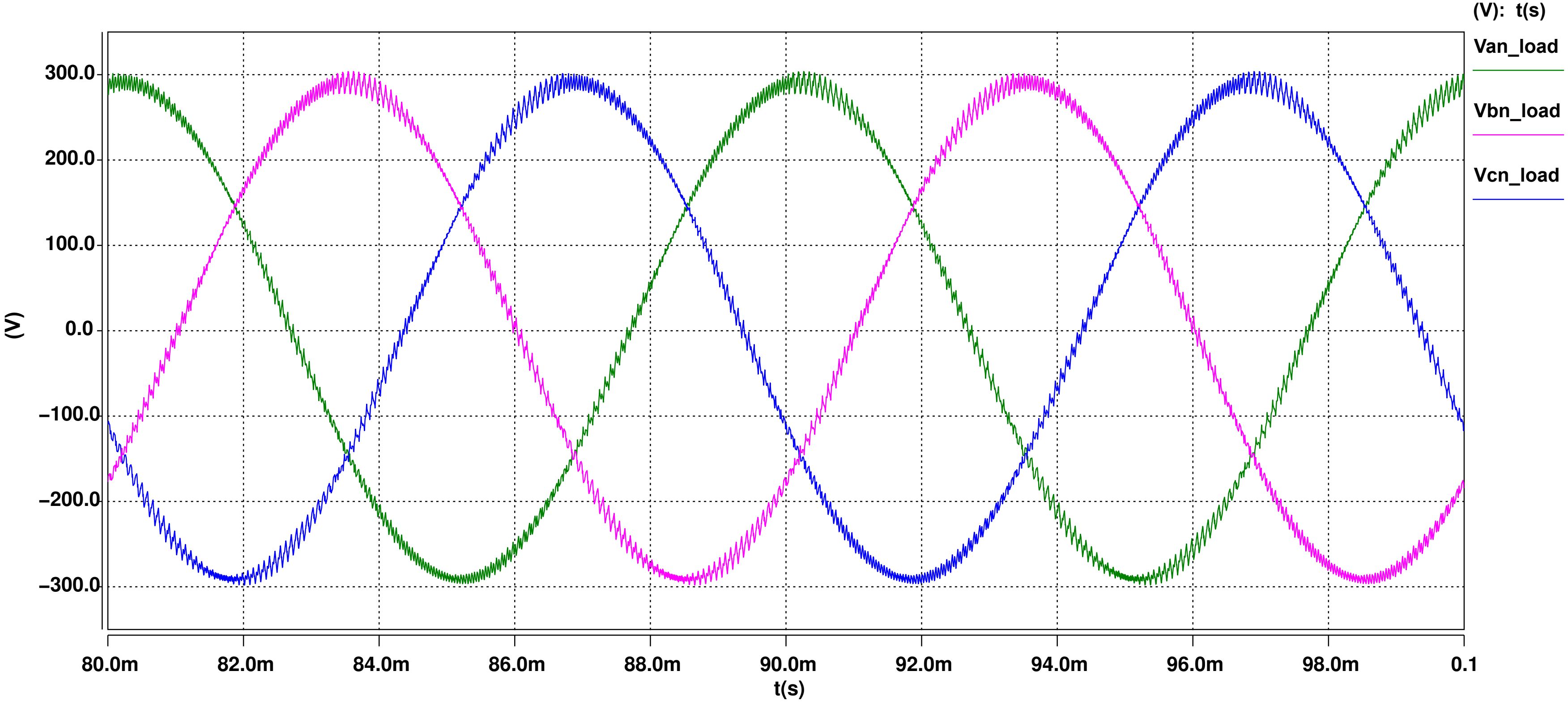


Figure 6.30: Plot of the 3 Output Phase-neutral Voltages at the load and reference to the load neutral point 100Hz Demand Frequency- $q = \frac{\sqrt{3}}{2}$ - 12.5kHz Sampling Frequency

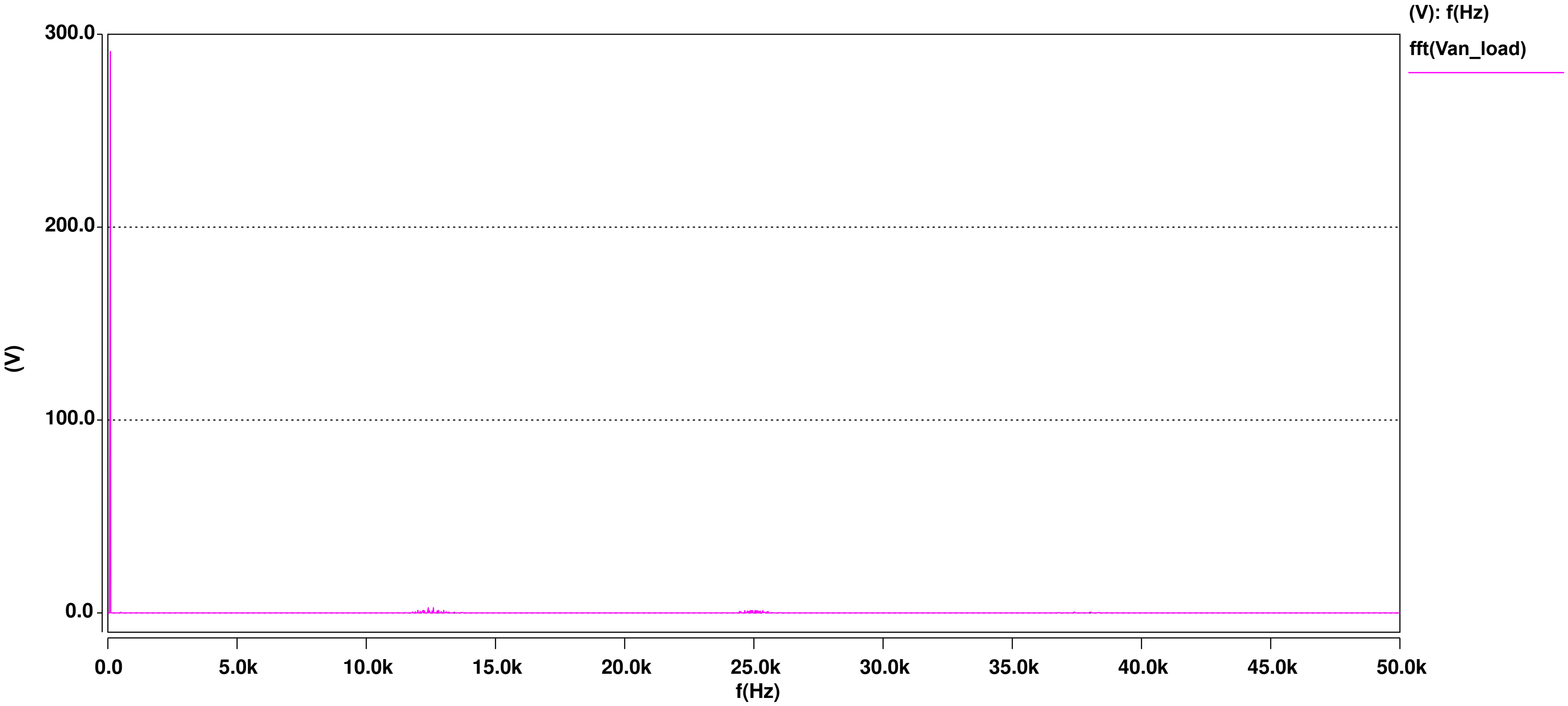


Figure 6.31: Plot of the FFT of the Output Phase-neutral Voltage on Leg **a** at the load 100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

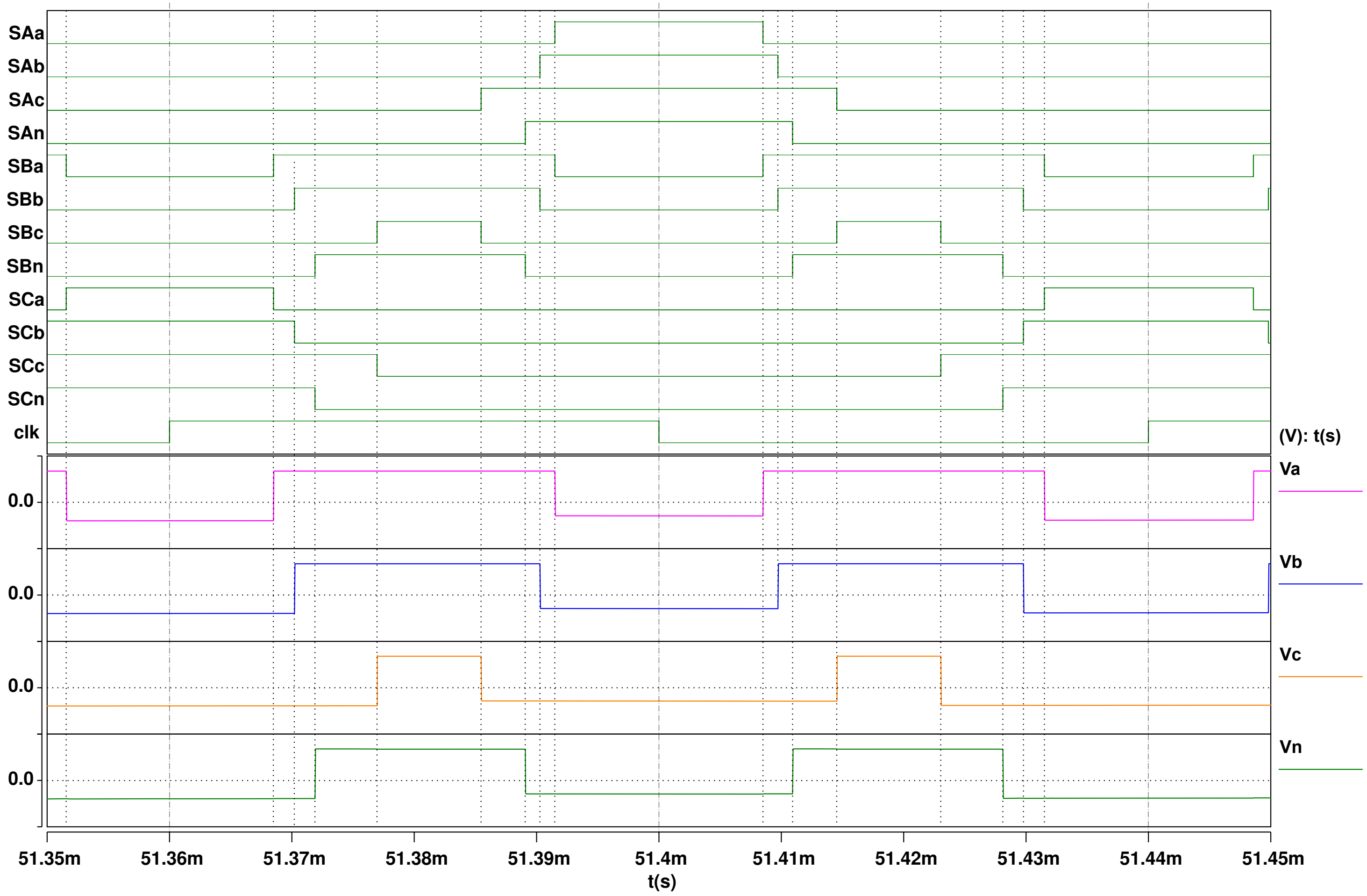


Figure 6.32: Plot of the Gate Control Signals
100Hz Demand Frequency- $q = 0.5$ - 12.5kHz Sampling Frequency

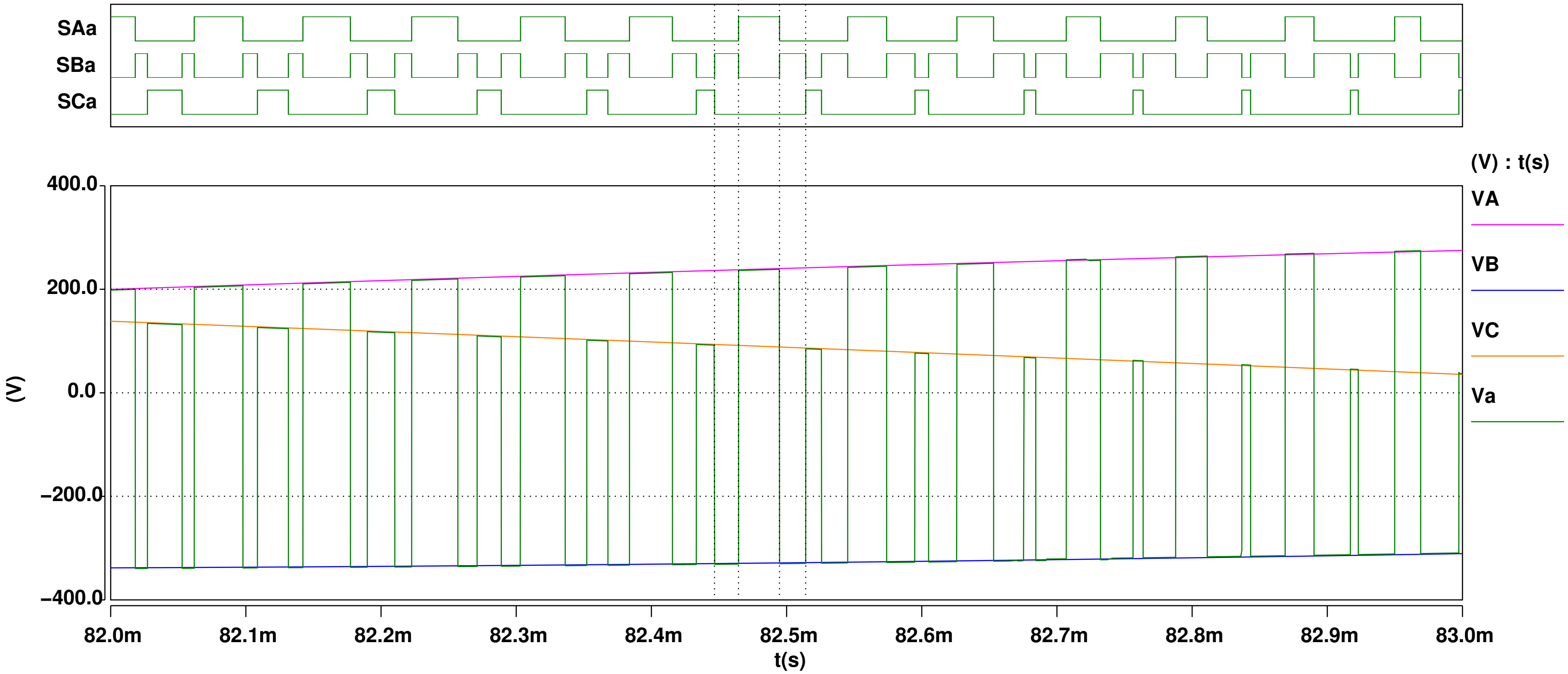


Figure 6.33: Plot showing the Switching of the Voltage on Leg **a** between the Input Phases 100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

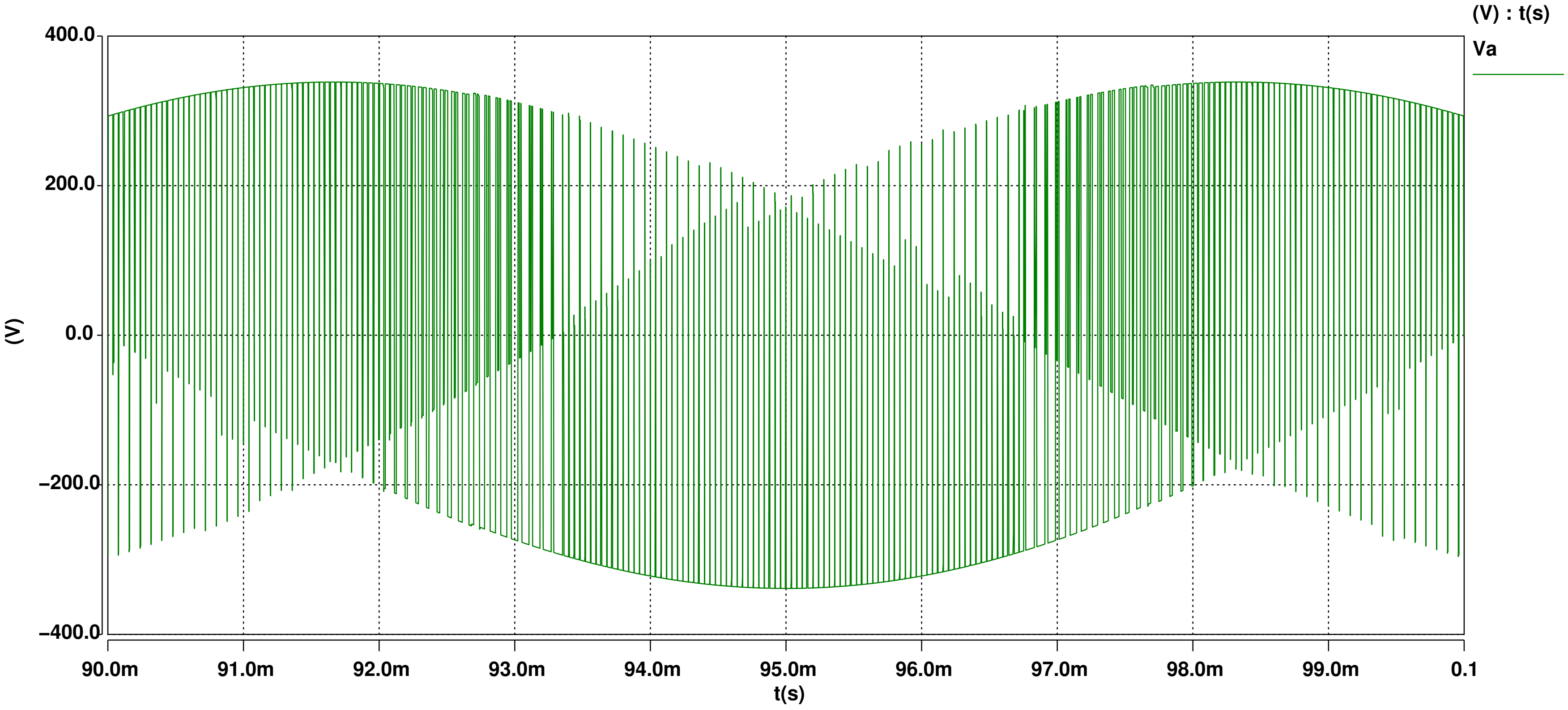


Figure 6.34: Plot of Output Phase Voltages for Leg **a** referenced to the supply neutral
100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

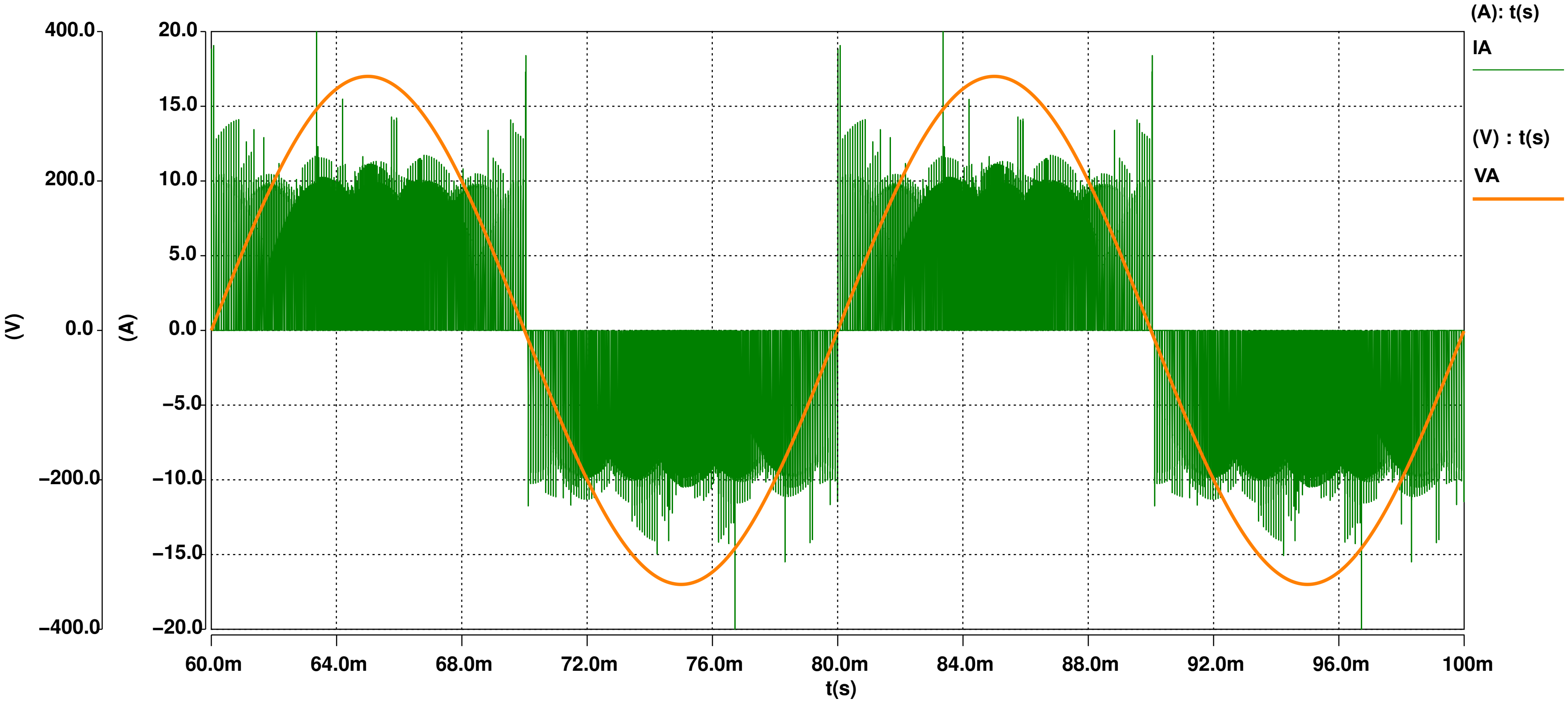


Figure 6.35: Plot of the Input Current and Voltage for Supply Phase A
100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

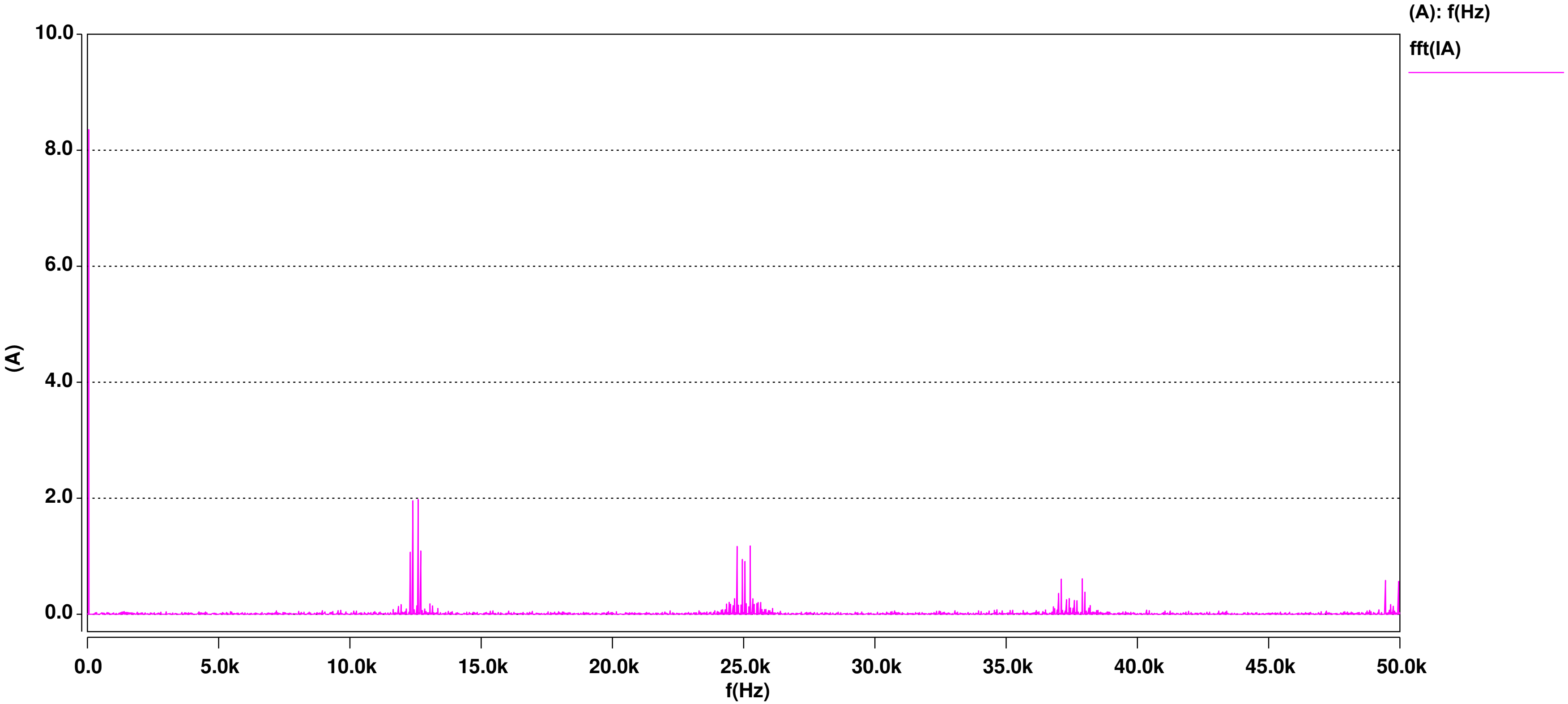


Figure 6.36: Plot of the FFT of Input Current for Supply Phase A
100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

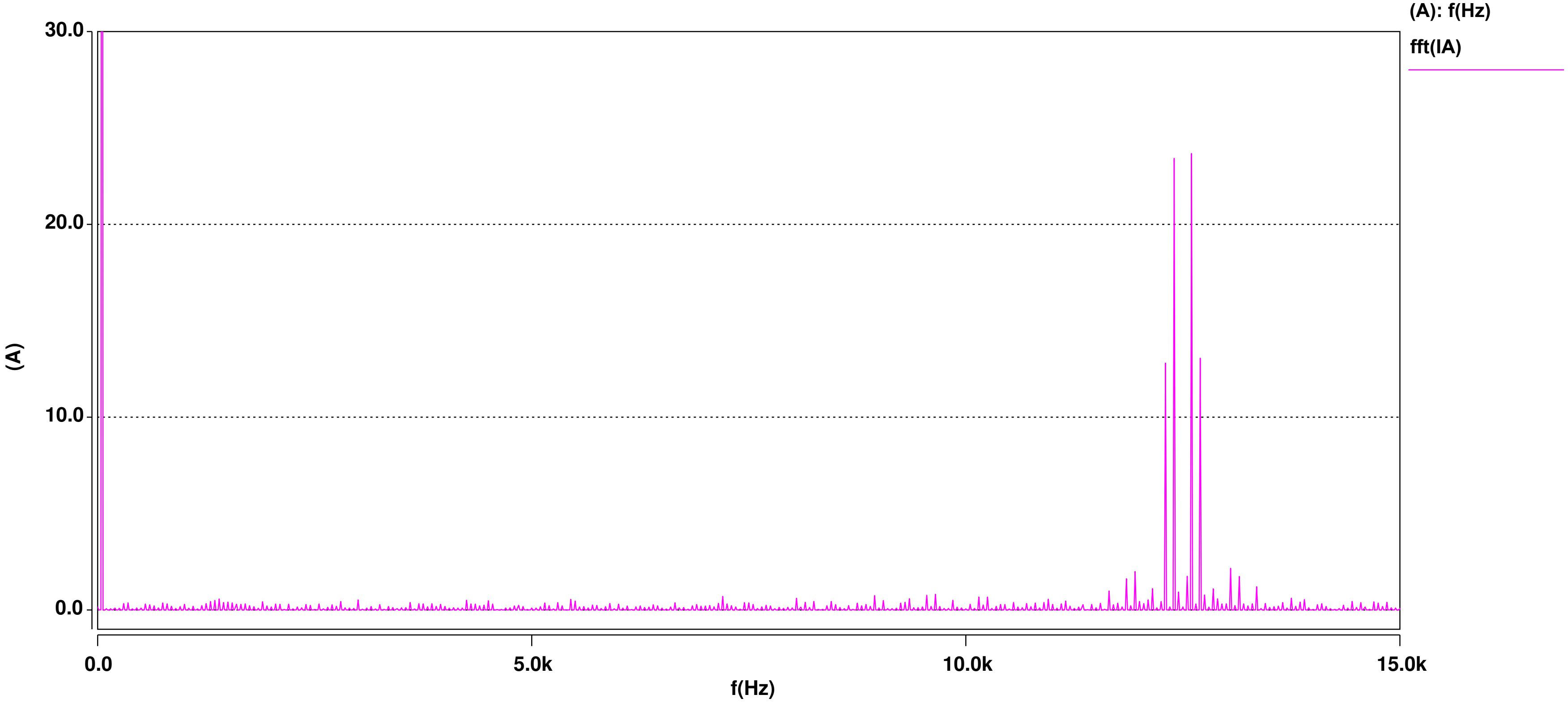


Figure 6.37: Plot of the FFT of Input Current for Supply Phase A scaled as a % of the fundamental 100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

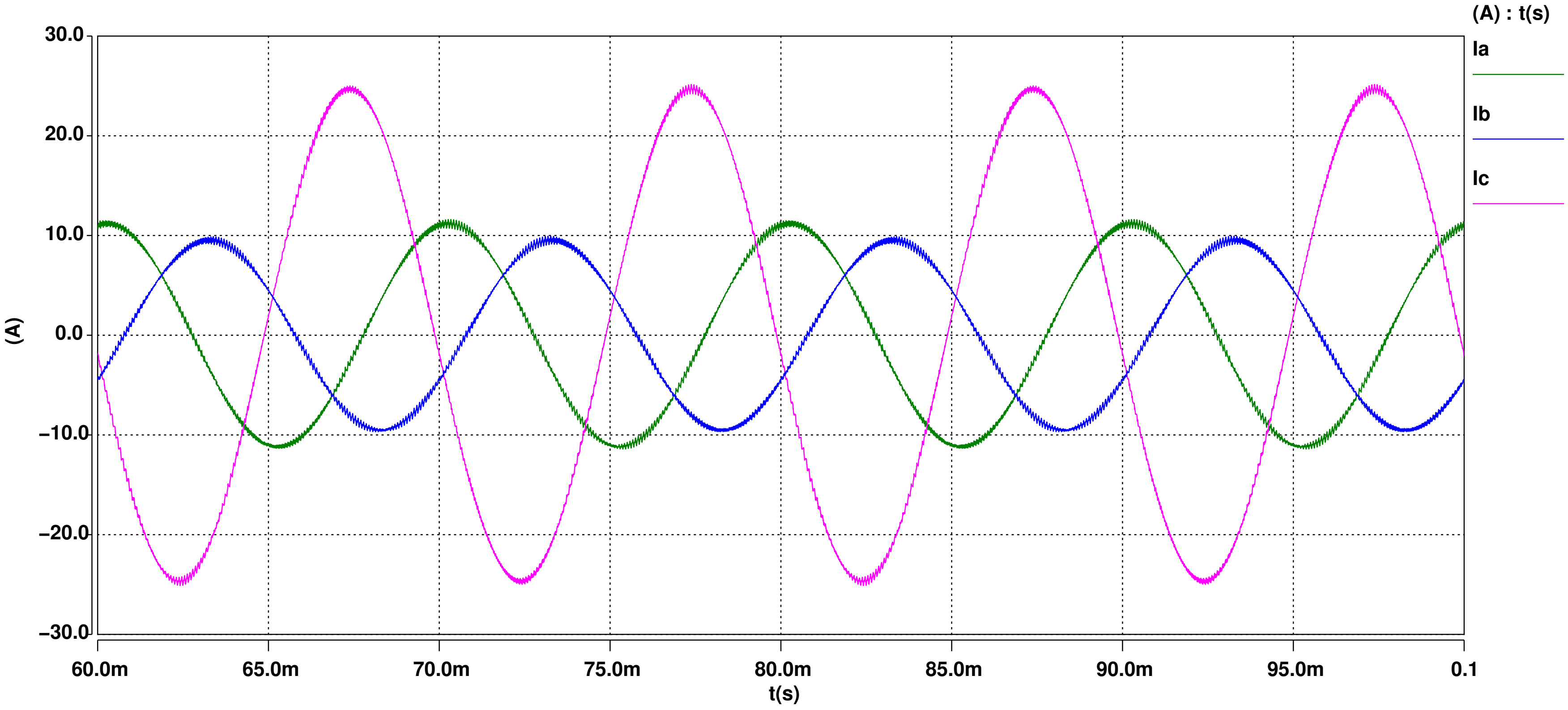


Figure 6.38: Plot of the Output Currents showing an Unbalanced Load
100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

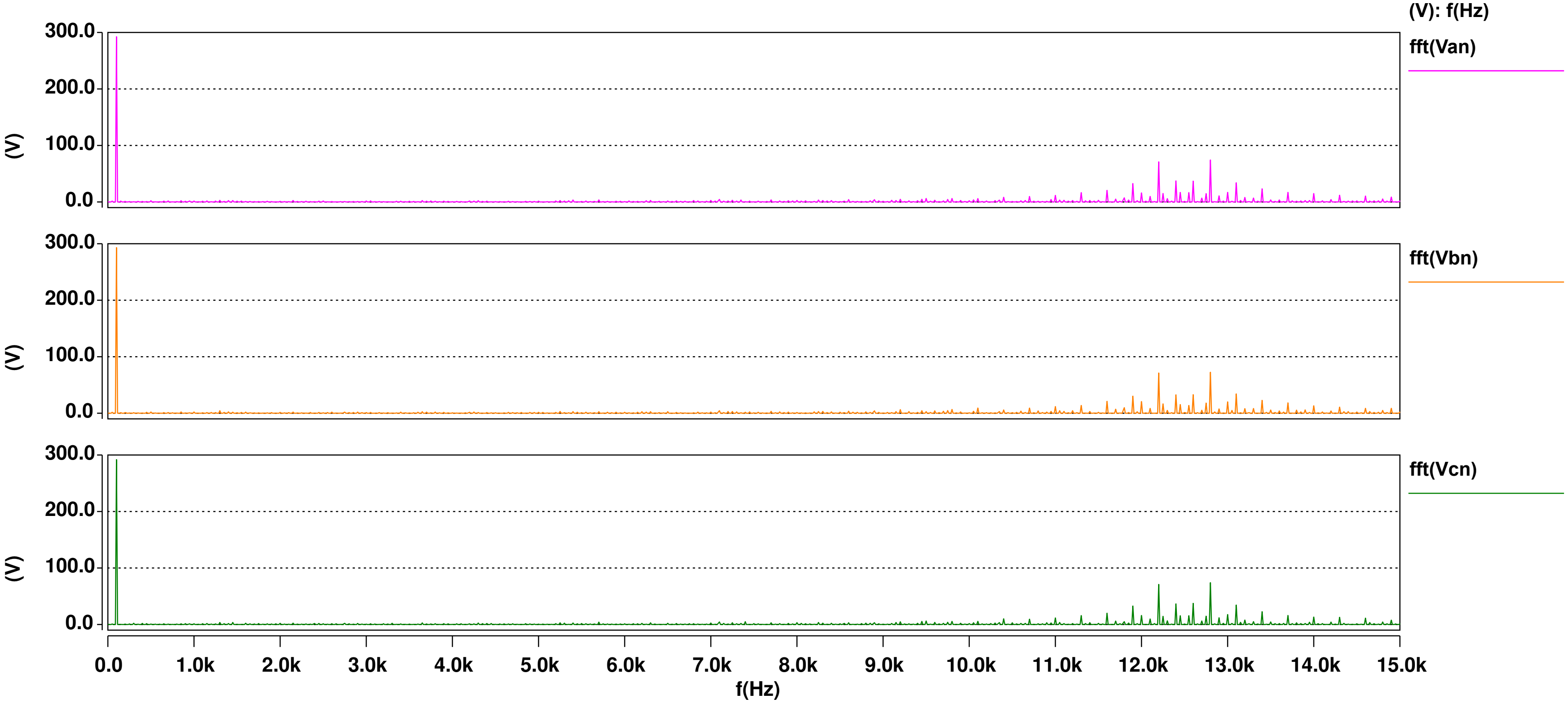


Figure 6.39: Plot of the FFT of Output Phase-neutral Voltages with an Unbalanced Load 100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

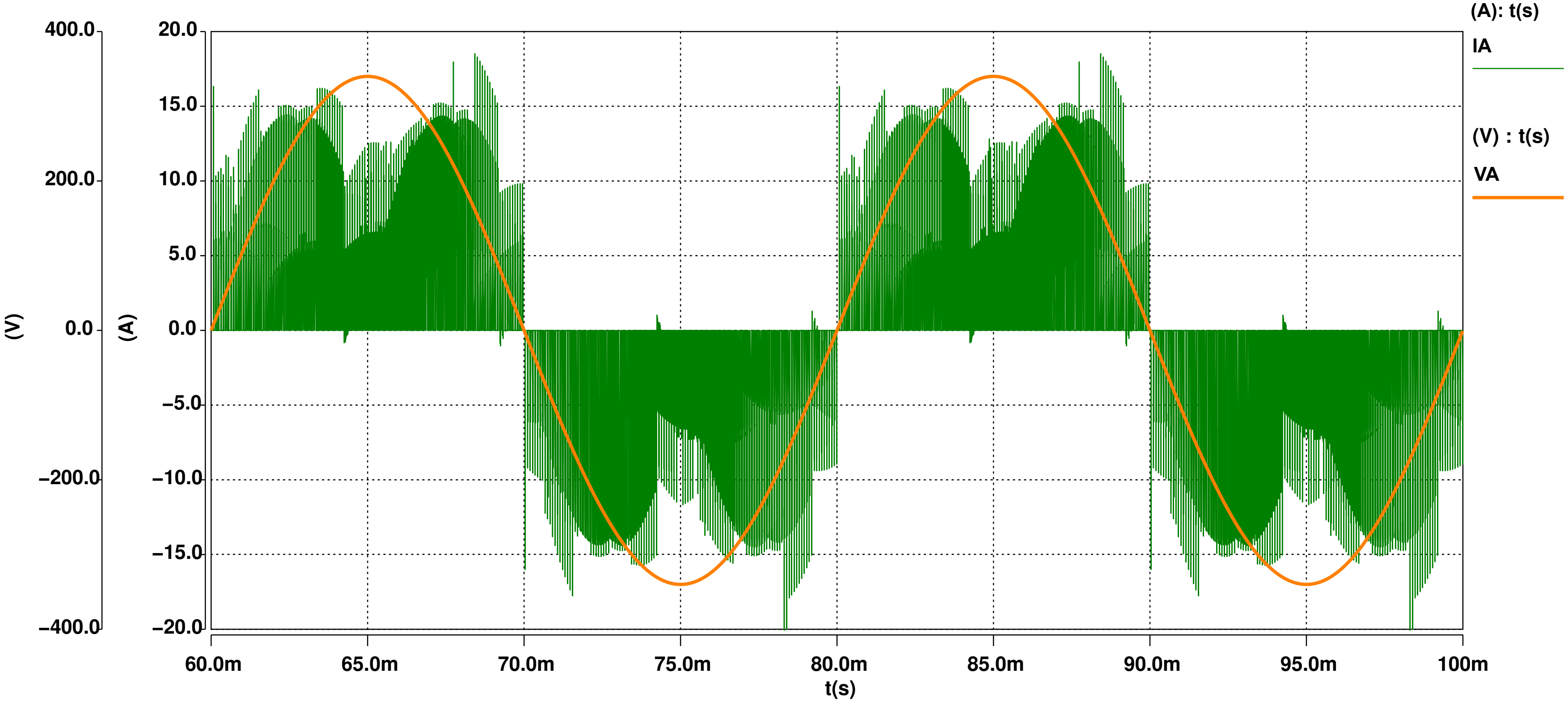


Figure 6.40: Plot of the Input Current and Voltage for Supply Phase A with an Unbalanced Load 100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

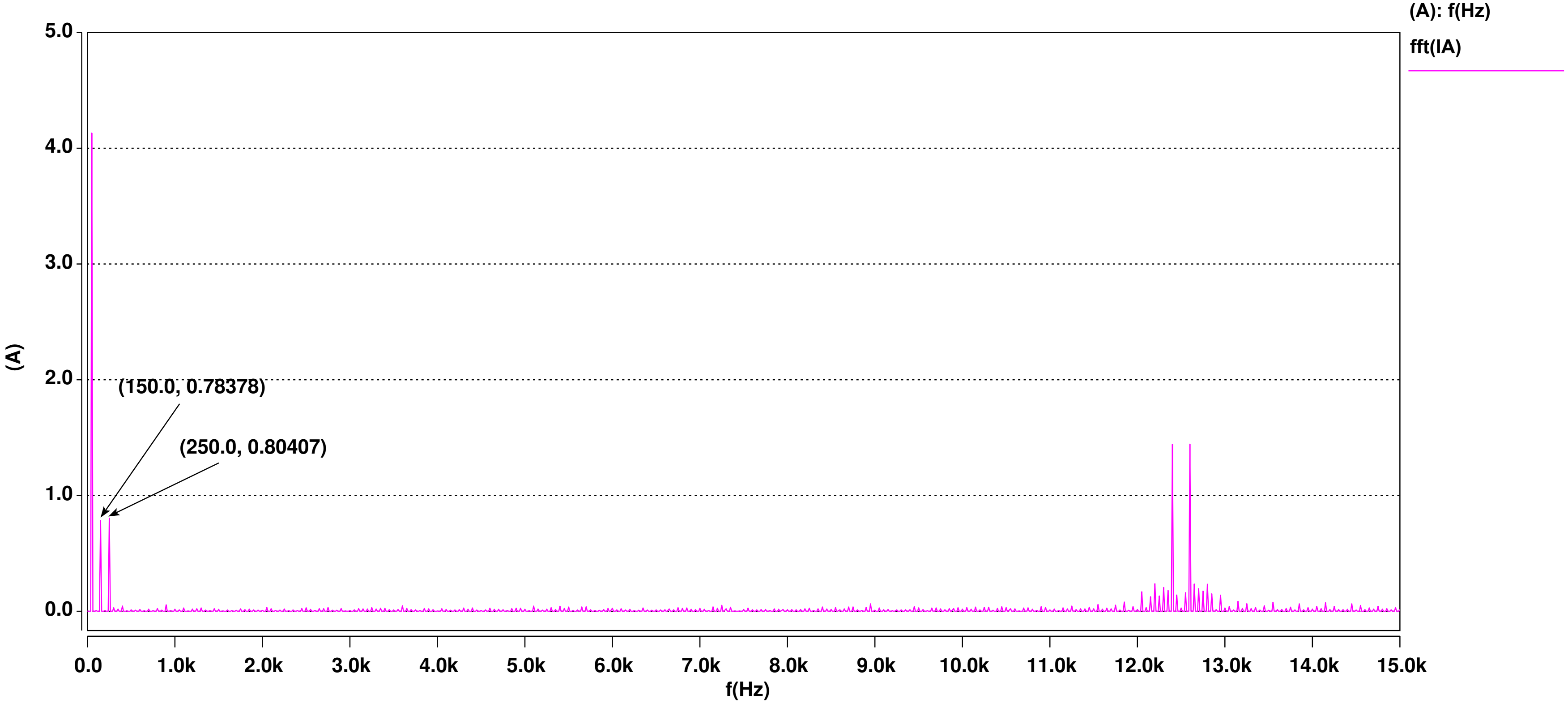


Figure 6.41: Plot of the FFT of Input Current for Supply Phase A with an Unbalanced Supply 100Hz Demand Frequency- $\frac{\sqrt{3}}{2}$ modulation index - 12.5kHz Sampling Frequency

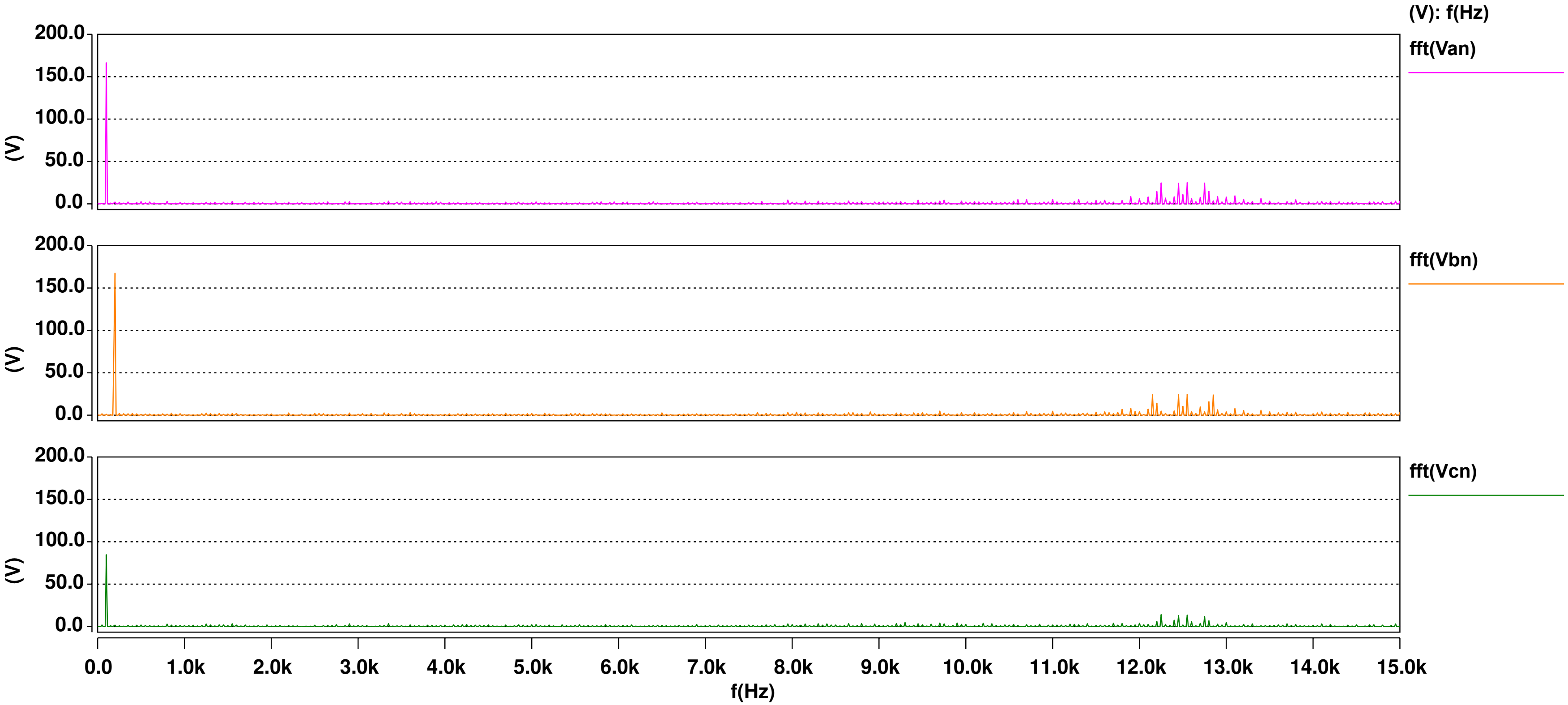


Figure 6.42: Plot of FFT of the 3 Output Phase-neutral Voltages for Unbalanced Demands $V_{dem_a} = V_{in_{peak}}\ 100Hz$, $V_{dem_b} = V_{in_{peak}}\ 200Hz$, $V_{dem_b} = \frac{V_{in_{peak}}}{2}\ 100Hz$ - 12.5kHz Sampling Frequency

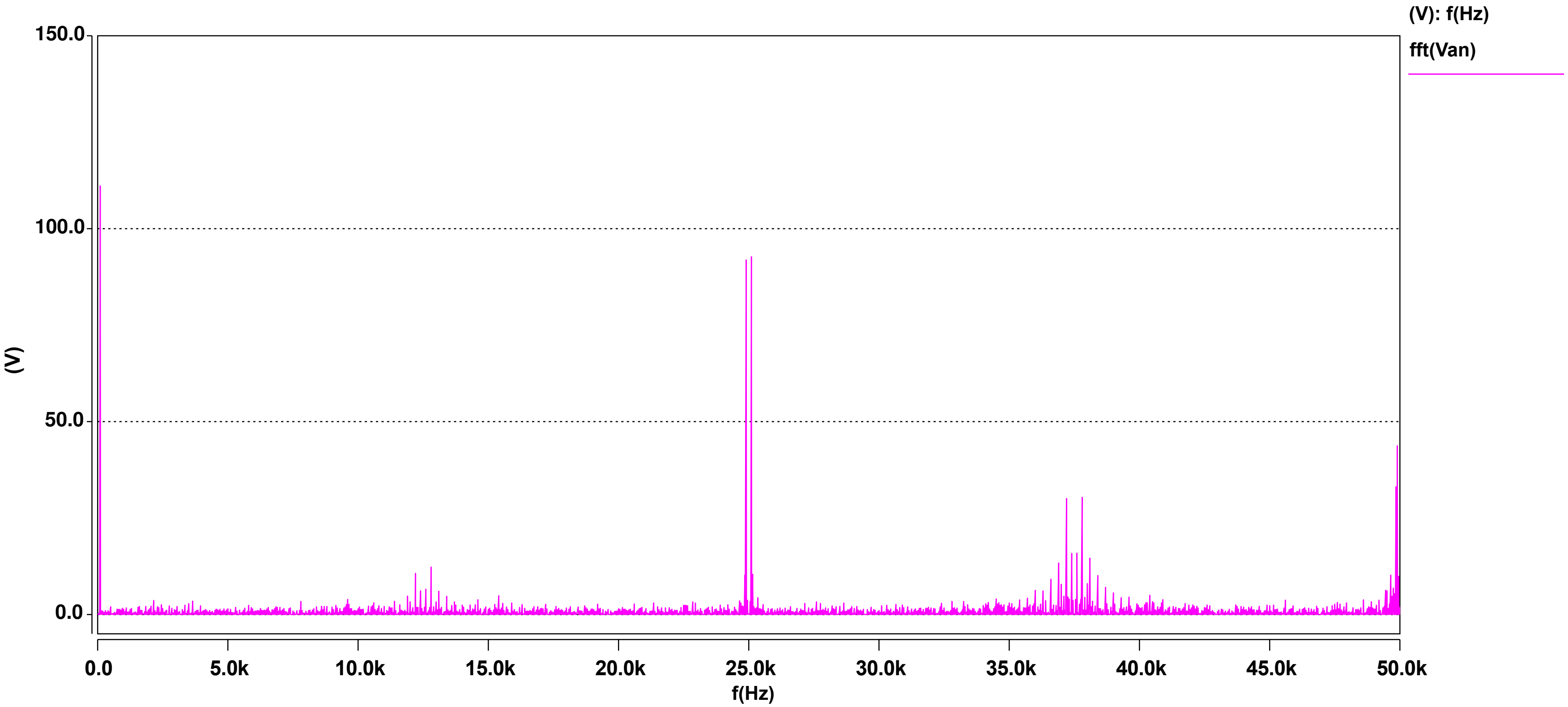


Figure 6.43: Plot of the FFT of the Output Phase-neutral Voltage on Leg **a** with respect to Leg **n** using the Two Zero Switching Sequence
100Hz Demand Frequency- $q = 0.5$ - 12.5kHz Sampling Frequency

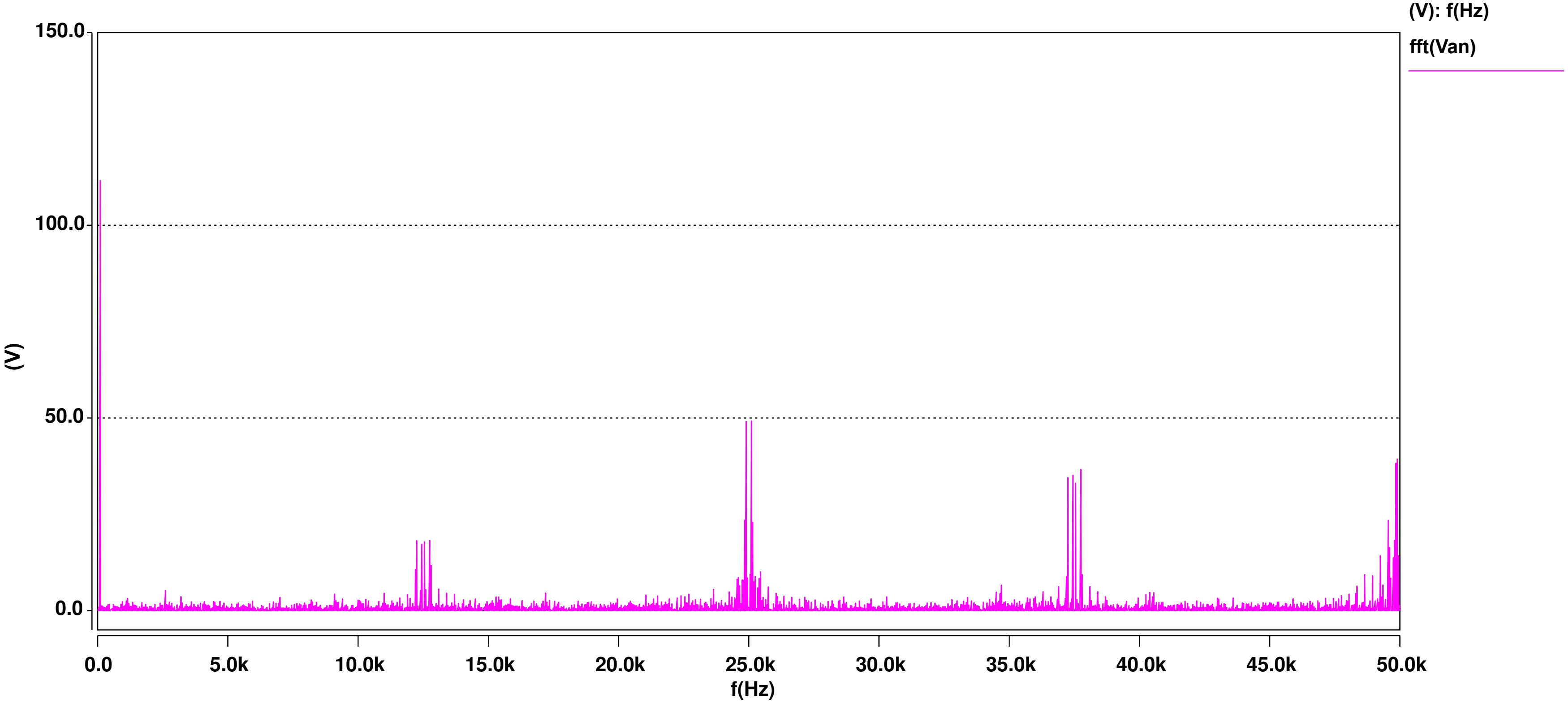


Figure 6.44: Plot of the FFT of the Output Phase-neutral Voltage on Leg **a** with respect to Leg **n** using the Three Zero Switching Sequence
100Hz Demand Frequency- $q = 0.5$ - 12.5kHz Sampling Frequency

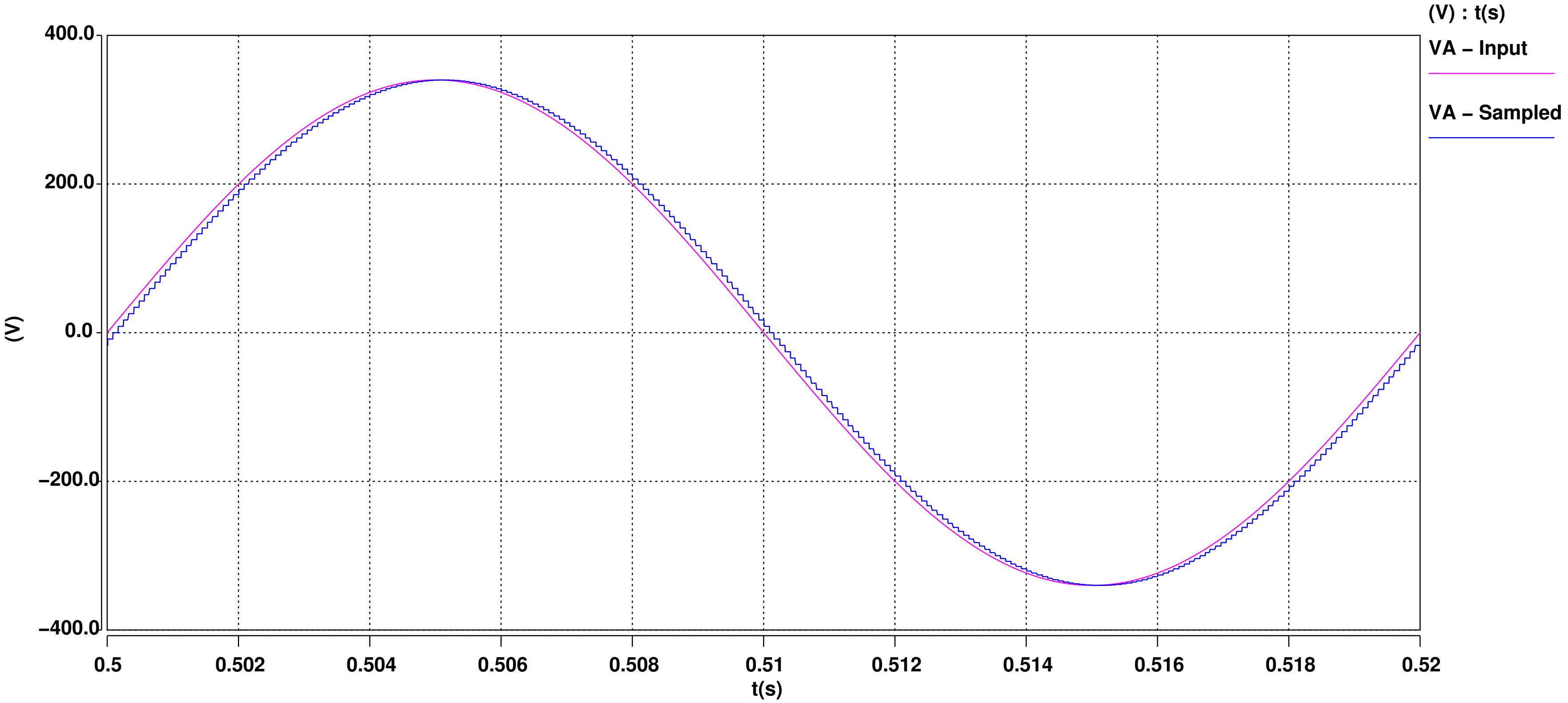


Figure 6.45: Plot of the Supply Voltage alongside the Sampled Voltage showing the Processing Delay within the Converter 50Hz Input Frequency - 12.5kHz Sampling Frequency

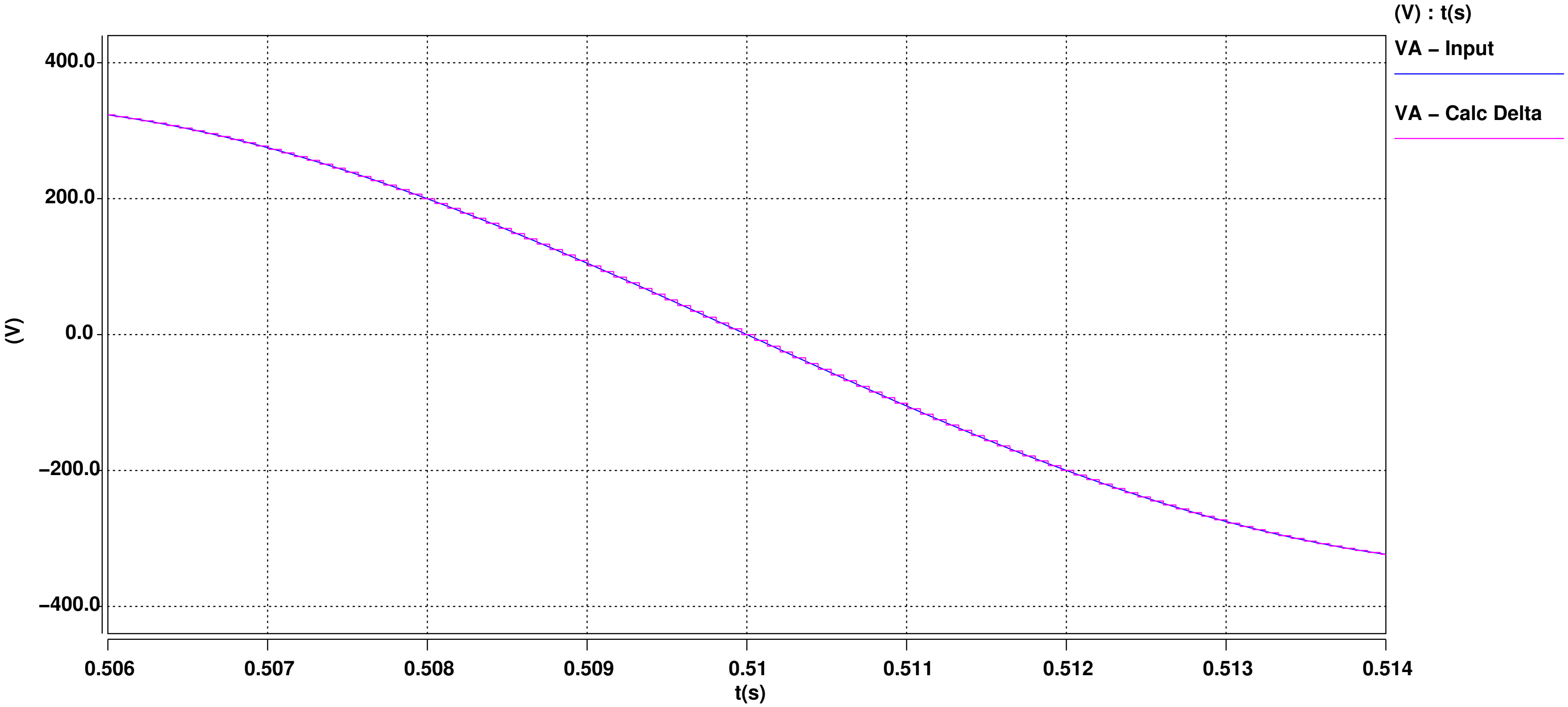


Figure 6.46: Plot of the Supply Voltage alongside the Sampled Voltage showing the Processing Delay Compensation
50Hz Input Frequency - 12.5kHz Sampling Frequency

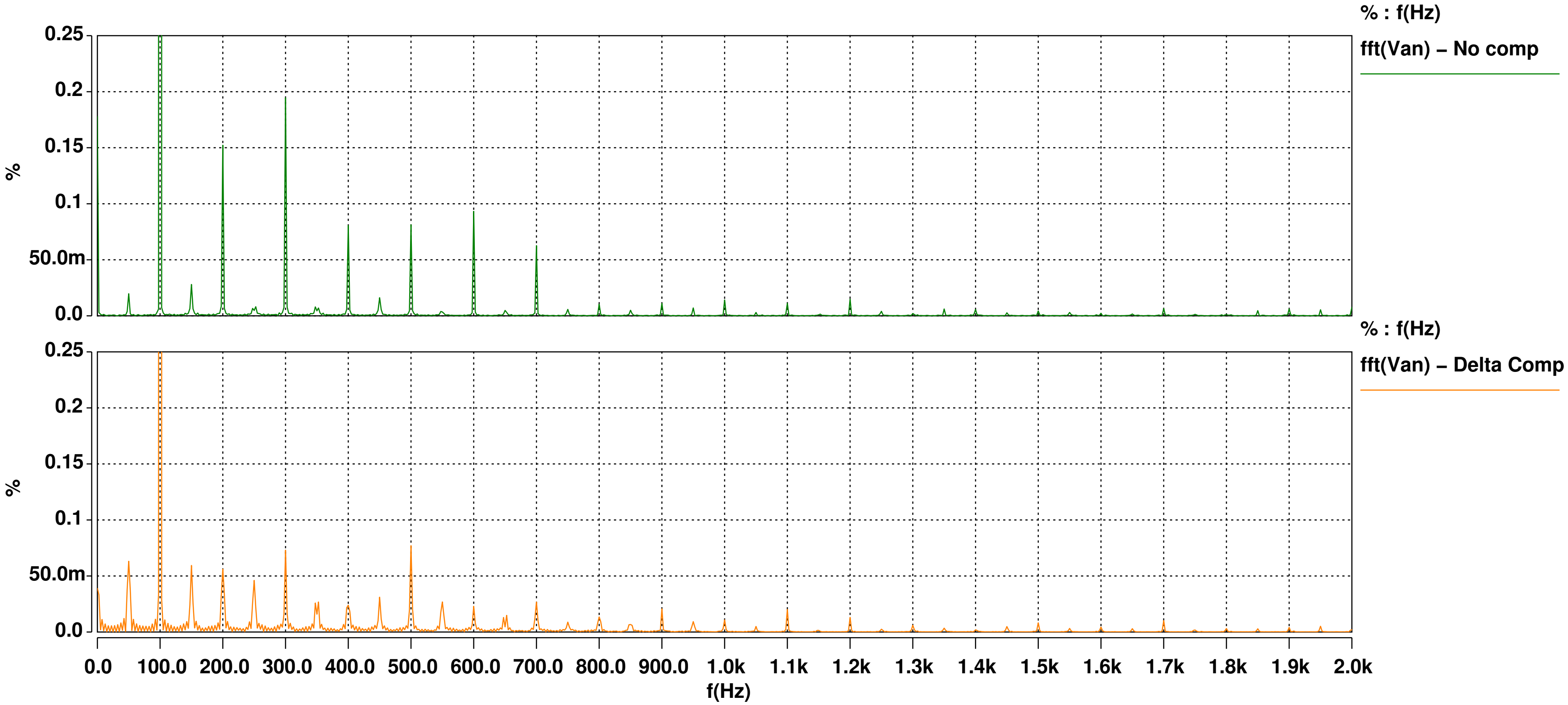


Figure 6.47: Plot of the FFT of the Output Voltage, Van, with and and without compensation for the processing delay using the Delta method
100Hz Output Frequency - 12.5kHz Sampling Frequency

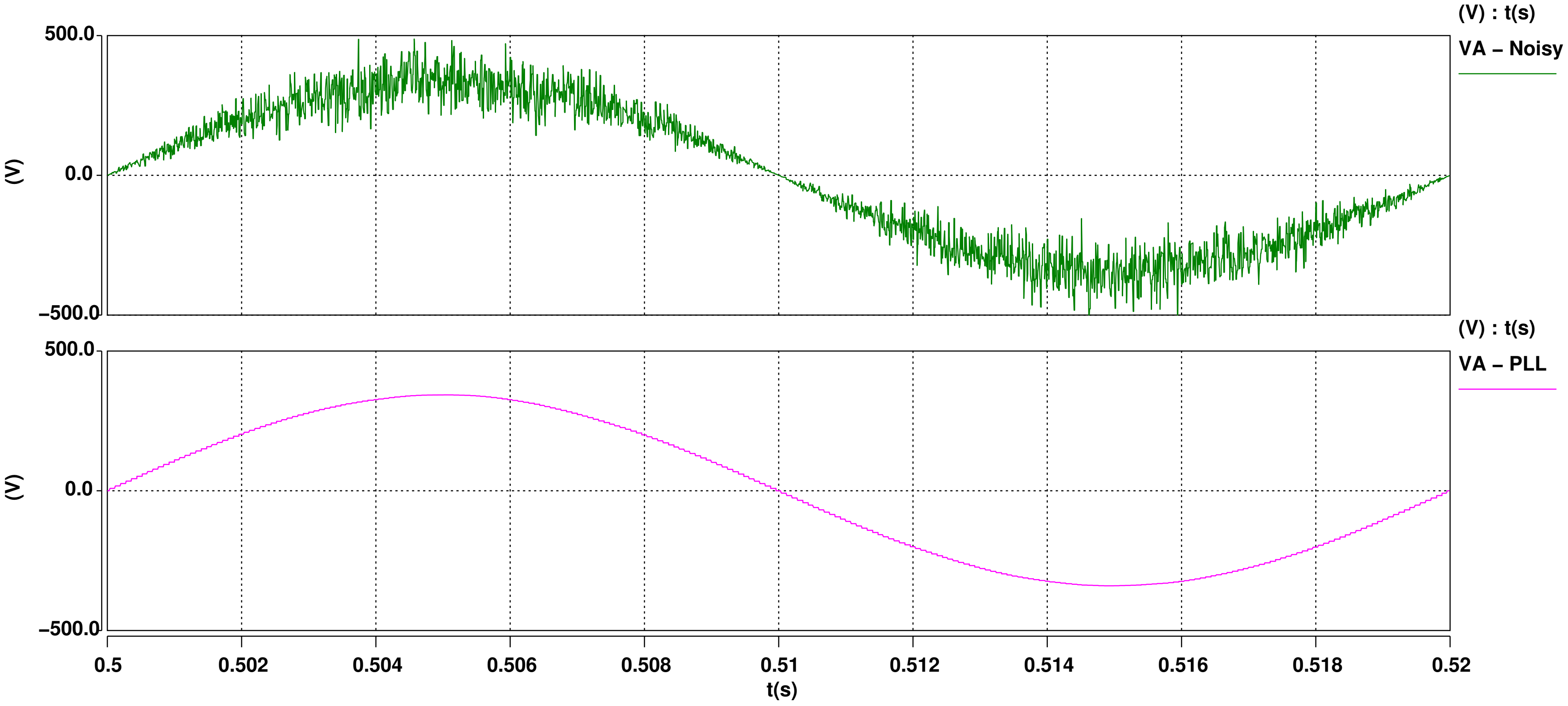


Figure 6.48: Plot of the noisy Input Voltage, VA, alongside the PLL tracked fundamental frequency output used by the converter
50Hz Input Frequency - 12.5kHz Sampling Frequency

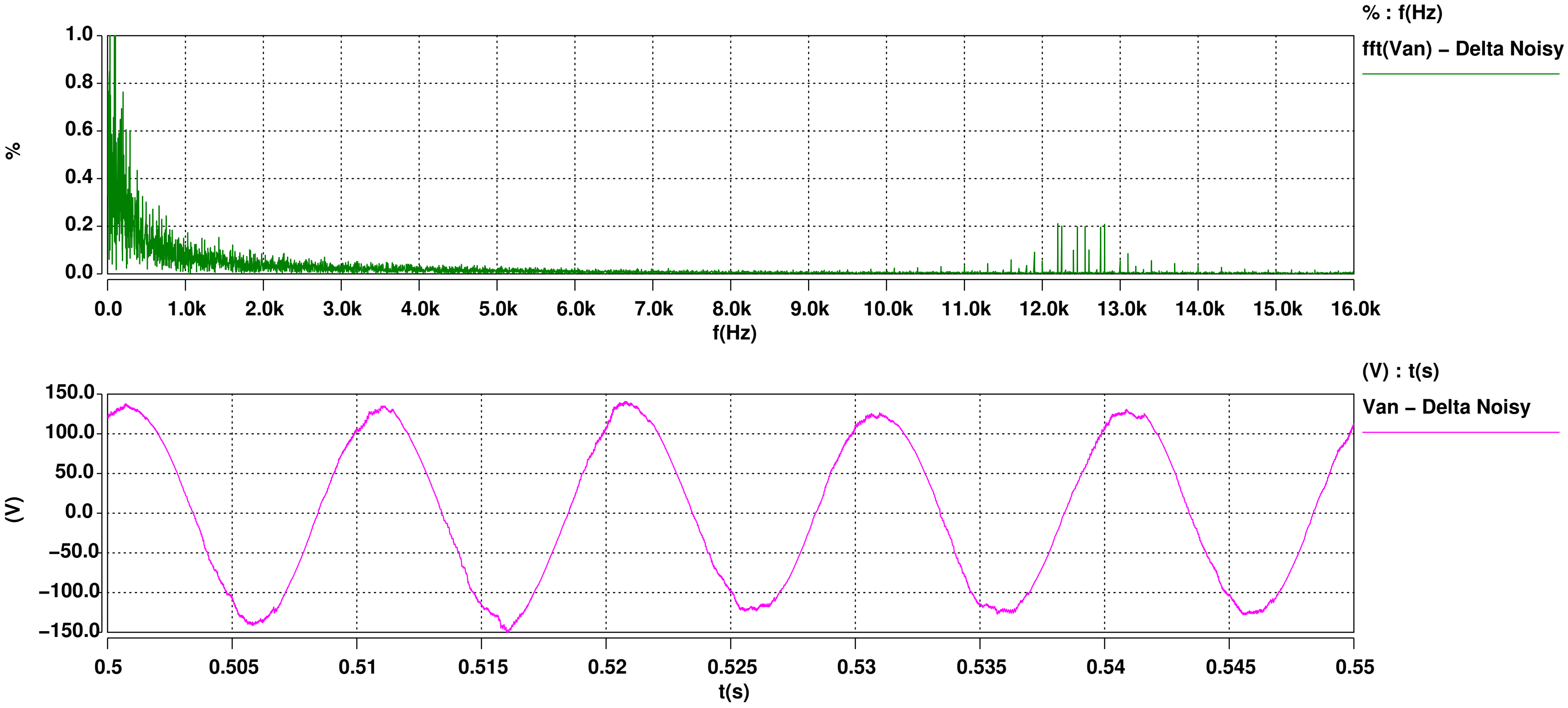


Figure 6.49: Plot of the Output Voltage, Van, and its Spectra showing the effect of Noise when using the Delta method of delay compensation
100Hz Output Frequency - 12.5kHz Sampling Frequency

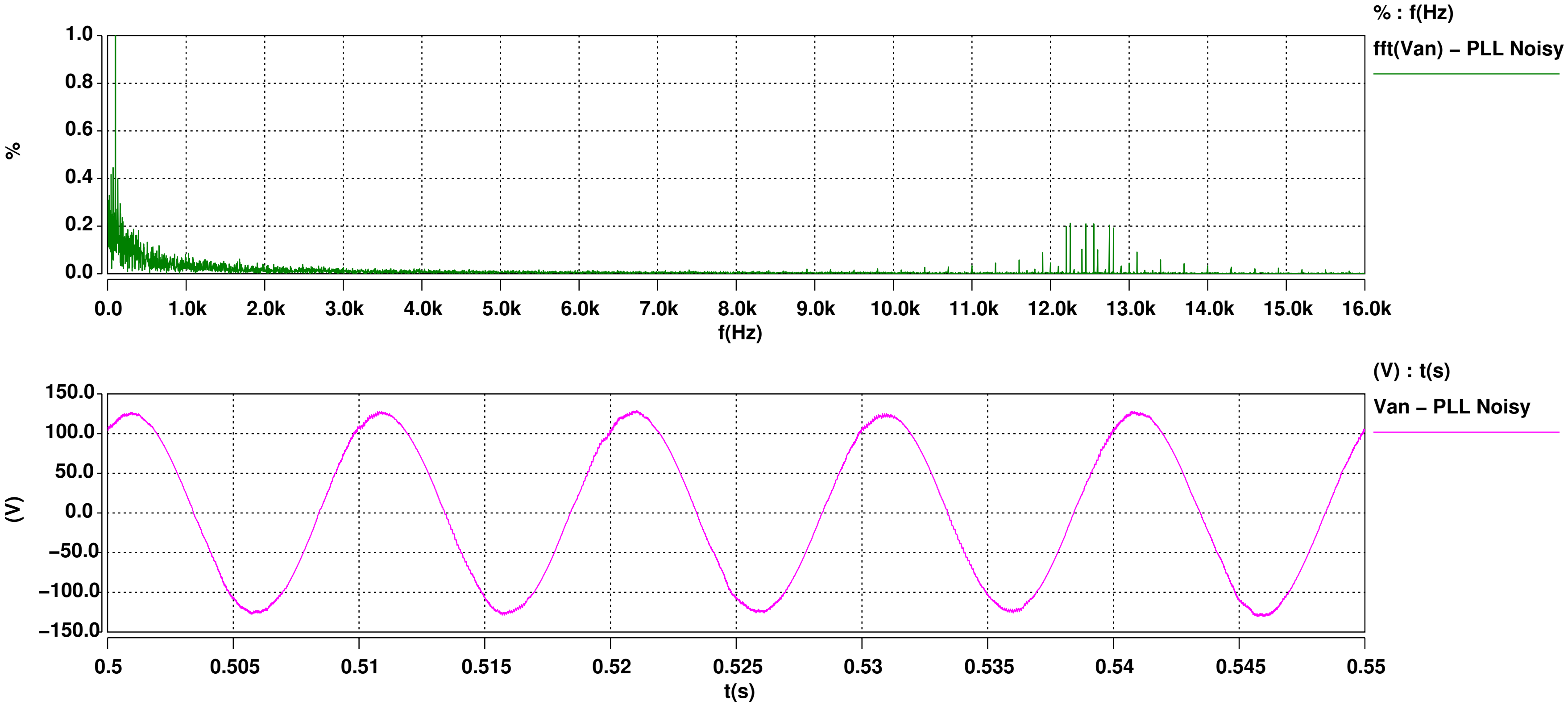


Figure 6.50: Plot of the Output Voltage, Van, and its Spectra showing the effect of Noise when using the PLL for delay compensation
100Hz Output Frequency - 12.5kHz Sampling Frequency

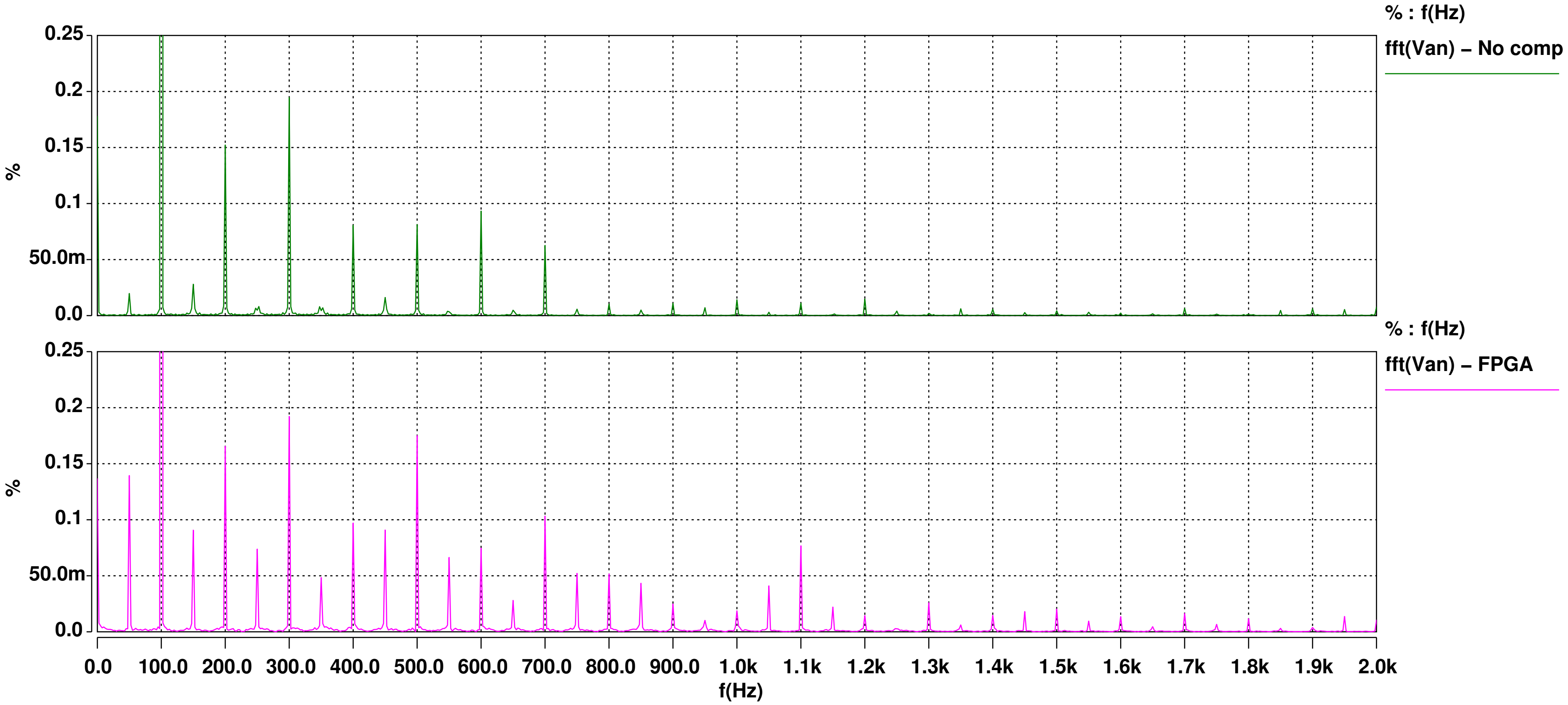


Figure 6.51: Plot of the FFT of the Output Voltage, Van, with and without the truncation effect of the FPGA minimum step size
100Hz Output Frequency - 12.5kHz Sampling Frequency

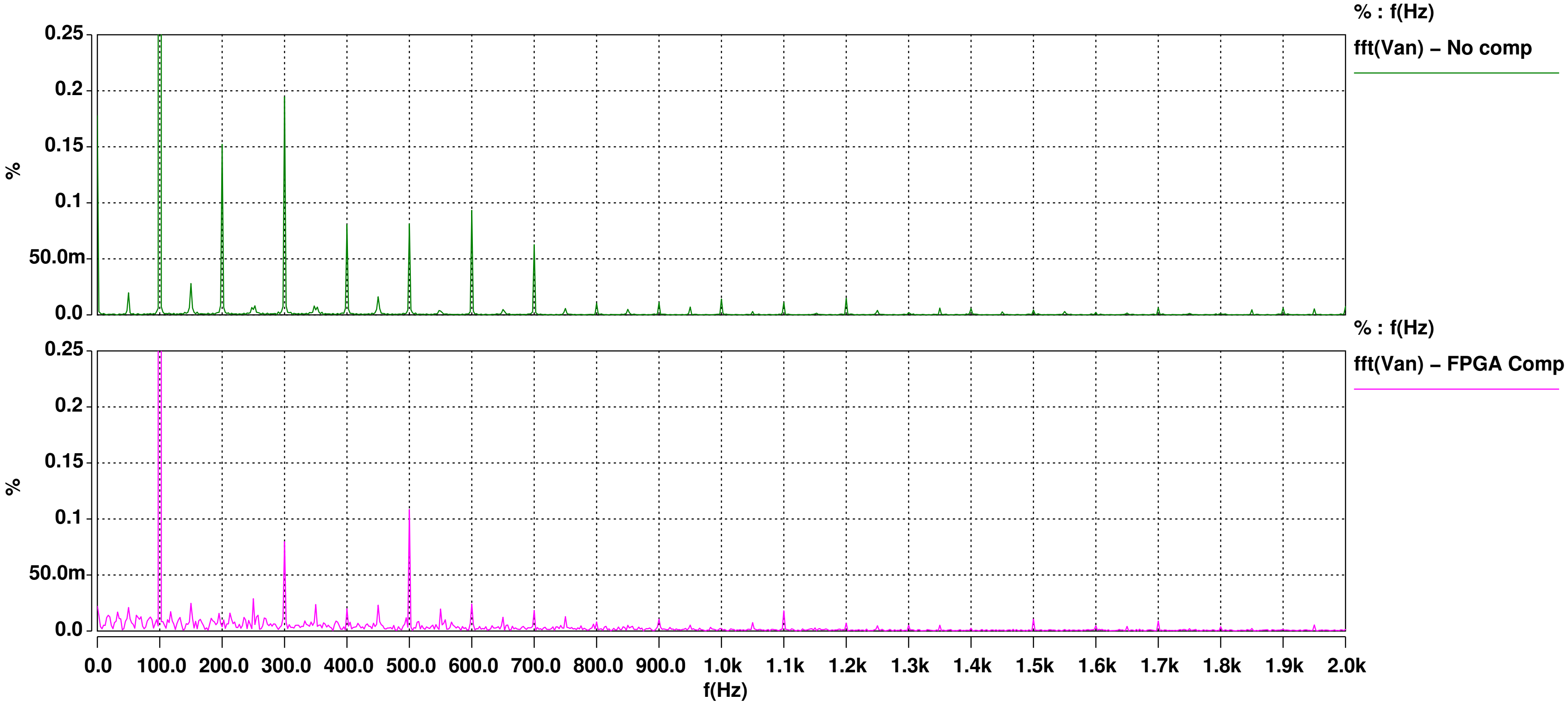


Figure 6.52: Plot of the FFT of the Output Voltage, Van, showing the original along with the output which is compensated for the truncation effect of the FPGA minimum step size
100Hz Output Frequency - 12.5kHz Sampling Frequency

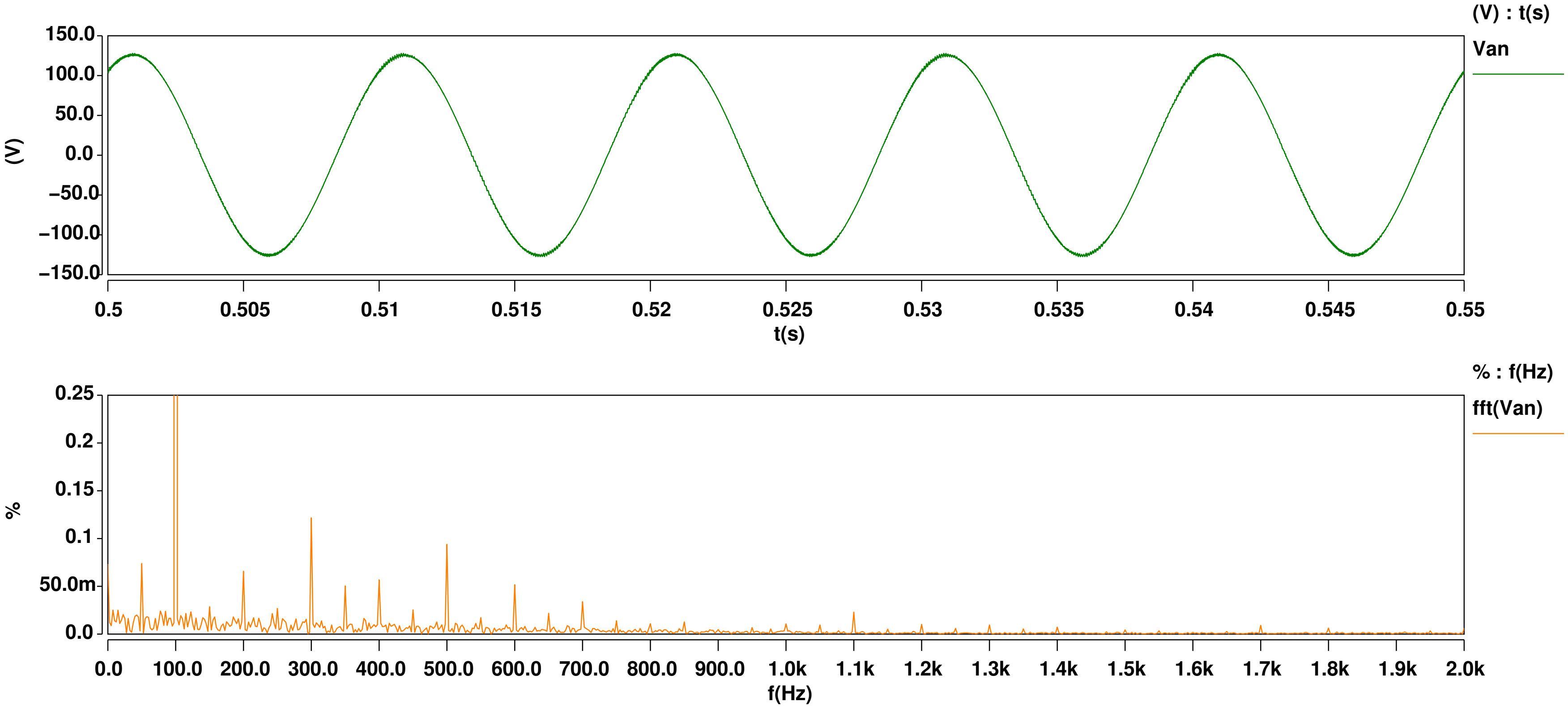


Figure 6.53: Plot of the Output Voltage, Van, and it's Spectra, showing the combined effect of the different types of compensation
100Hz Output Frequency - 12.5kHz Sampling Frequency

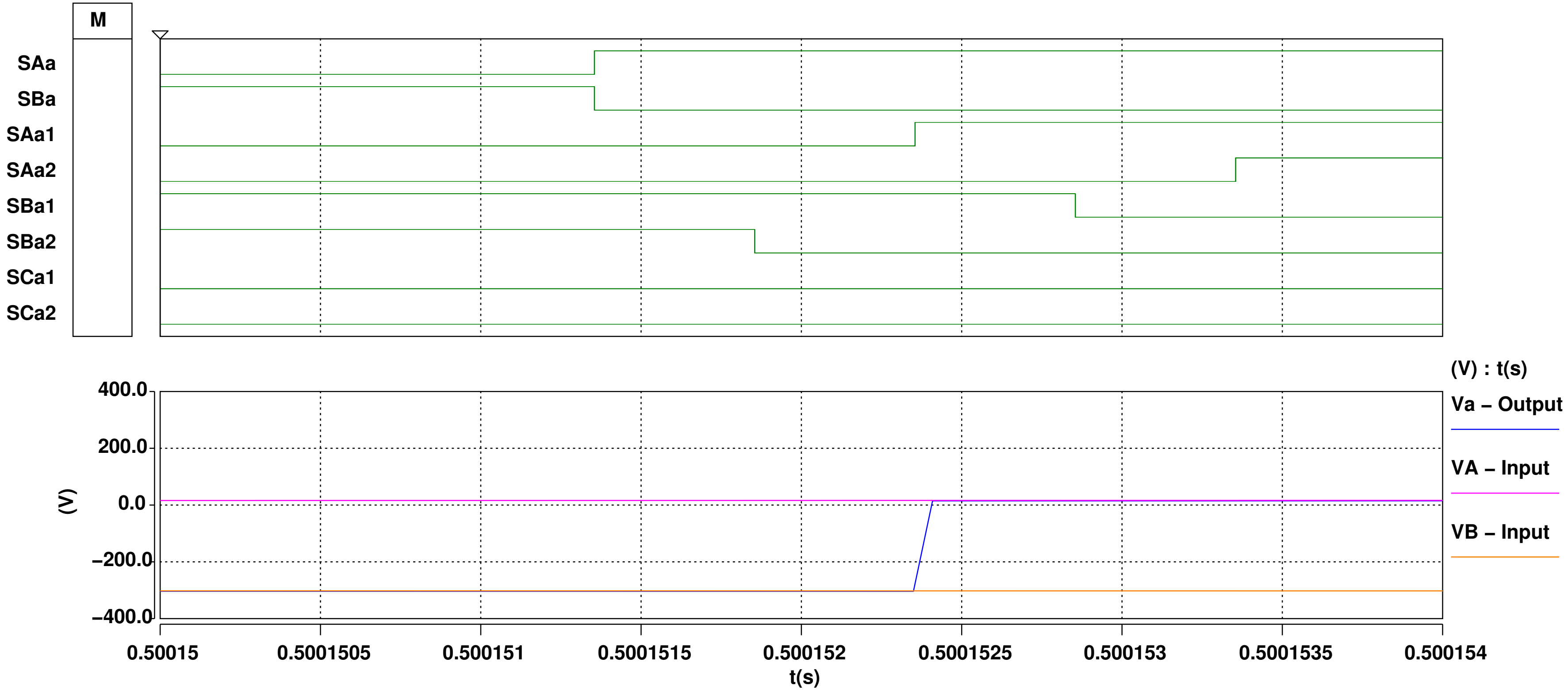


Figure 6.54: Plot showing the switching sequence used for commutation going from a lower to a higher voltage input phase

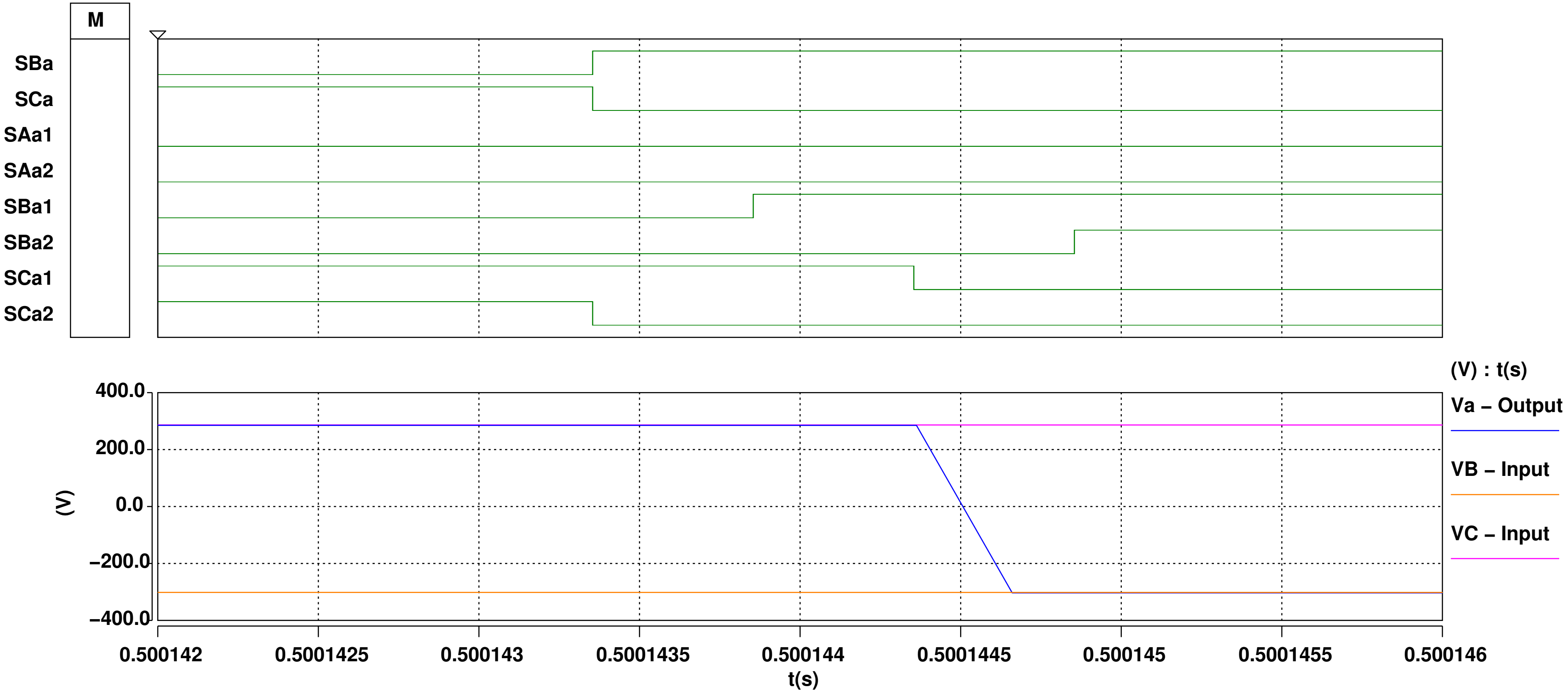


Figure 6.55: Plot showing the switching sequence used for commutation going from a lower to a higher voltage input phase

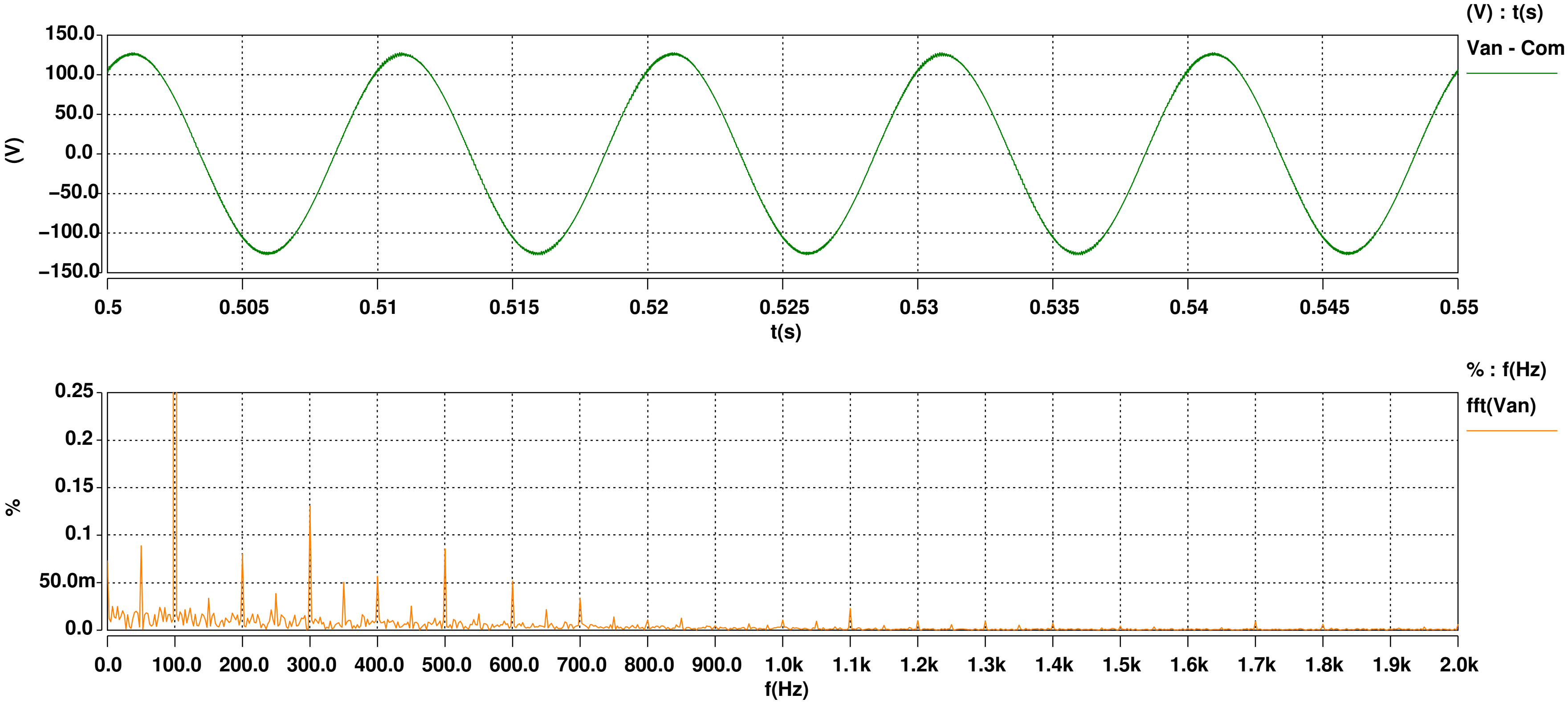


Figure 6.56: Plot of the FFT of the Output Voltage, Van, comparing the commutated output to the ideal 100Hz Output Frequency - 12.5kHz Sampling Frequency

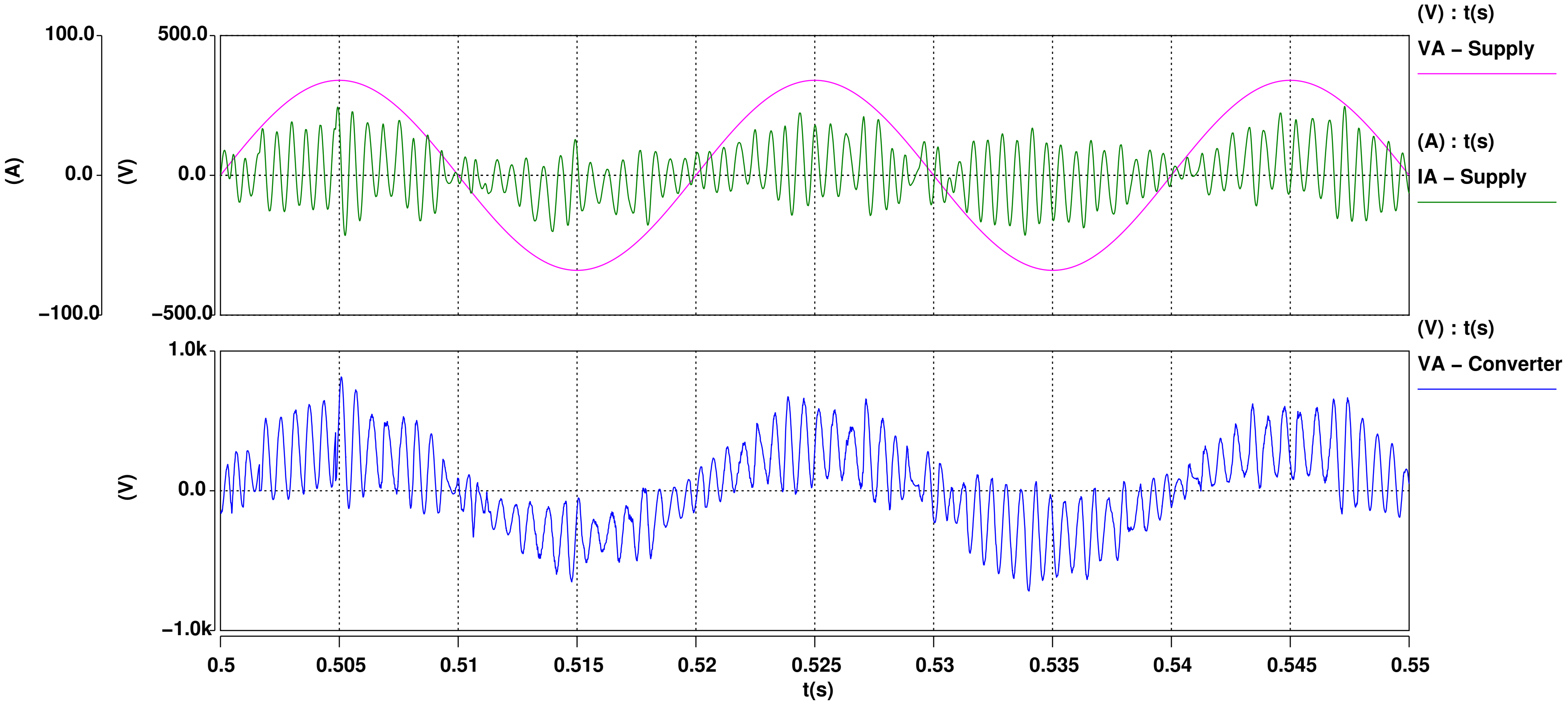


Figure 6.57: Plot of Input Phase A Supply Voltage and Current, along with the voltage seen at the converter, when using a 1mH/3μ3F Input Filter with no PLL
50Hz Input Frequency - 12.5kHz Sampling Frequency

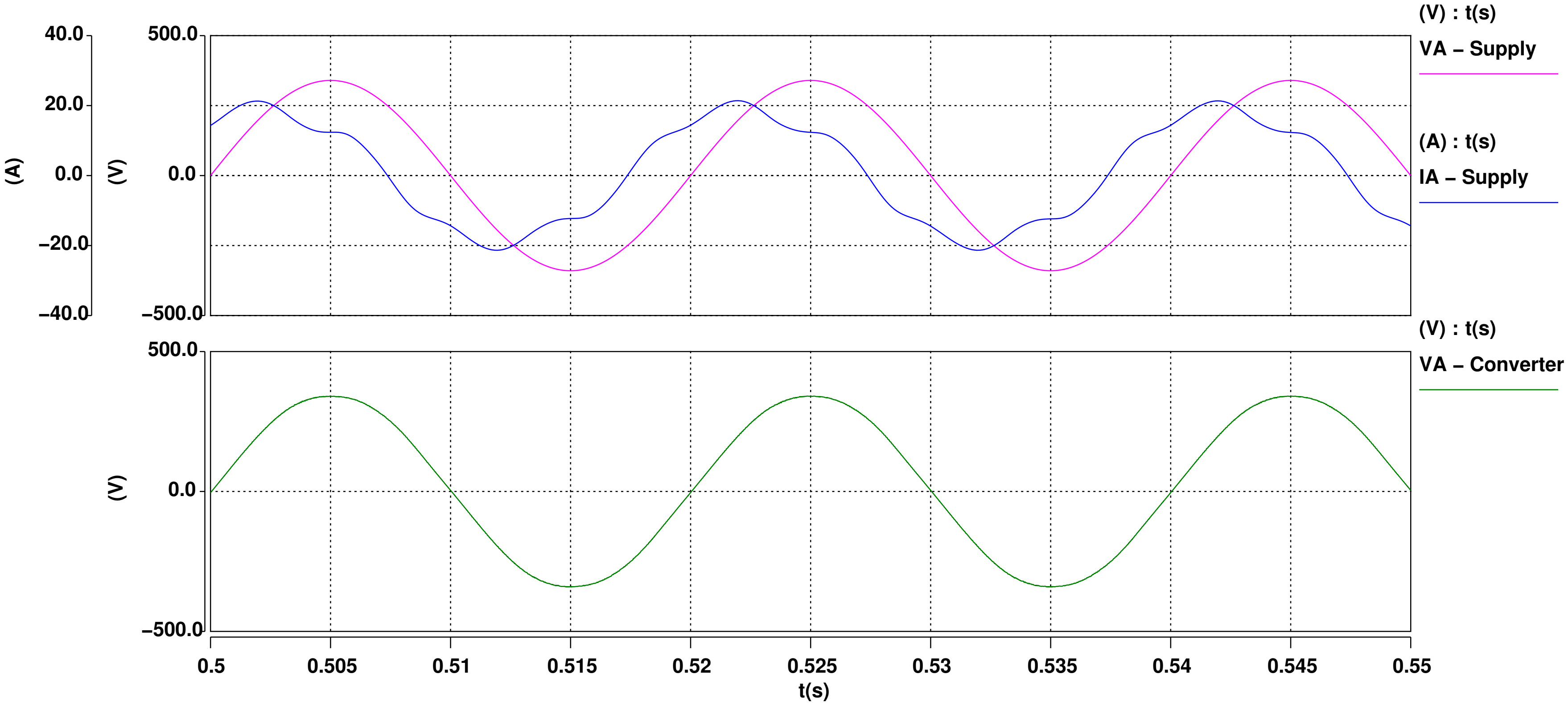


Figure 6.58: Plot of Input Phase A Supply Voltage and Current, along with the voltage seen at the converter, when using a 1mH/47μF Input Filter with no PLL
50Hz Input Frequency - 12.5kHz Sampling Frequency

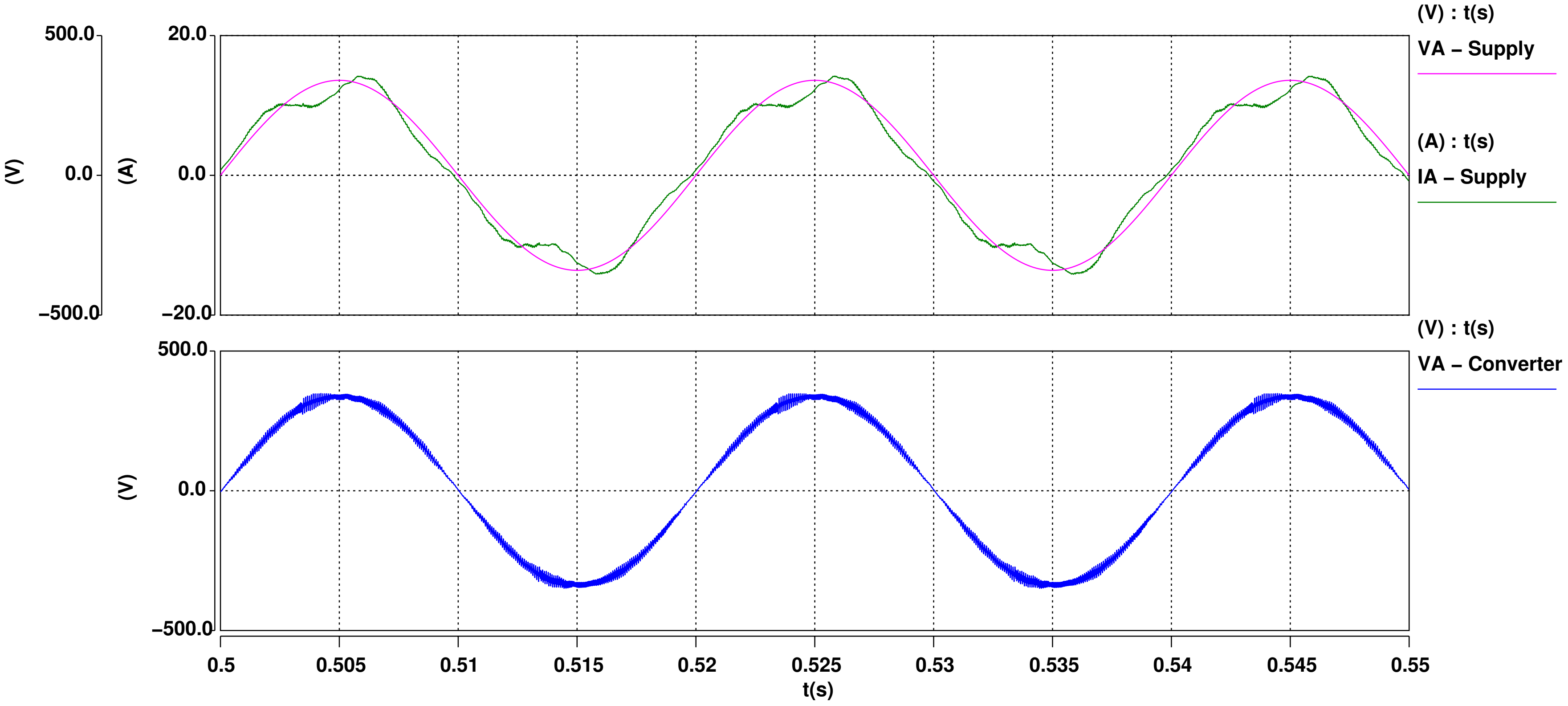


Figure 6.59: Plot of Input Phase A Supply Voltage and Current, along with the voltage seen at the converter, when using a 1mH/3μ3F Input Filter using a PLL
50Hz Input Frequency - 12.5kHz Sampling Frequency

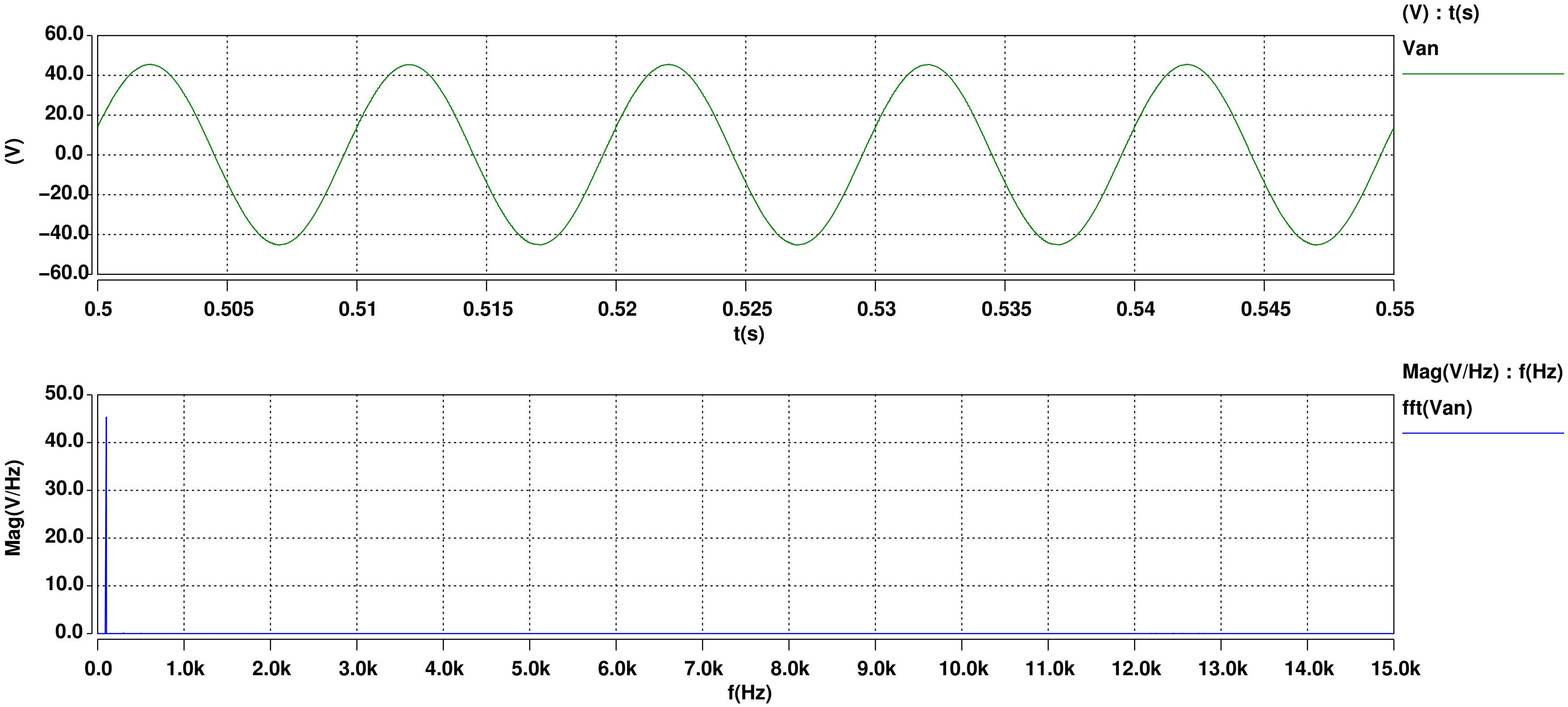


Figure 6.60: Plot of Output Voltage, Van, along with its Spectra, when using a 1mH/3μ3F Input Filter using a PLL 100Hz Output Frequency - 12.5kHz Sampling Frequency

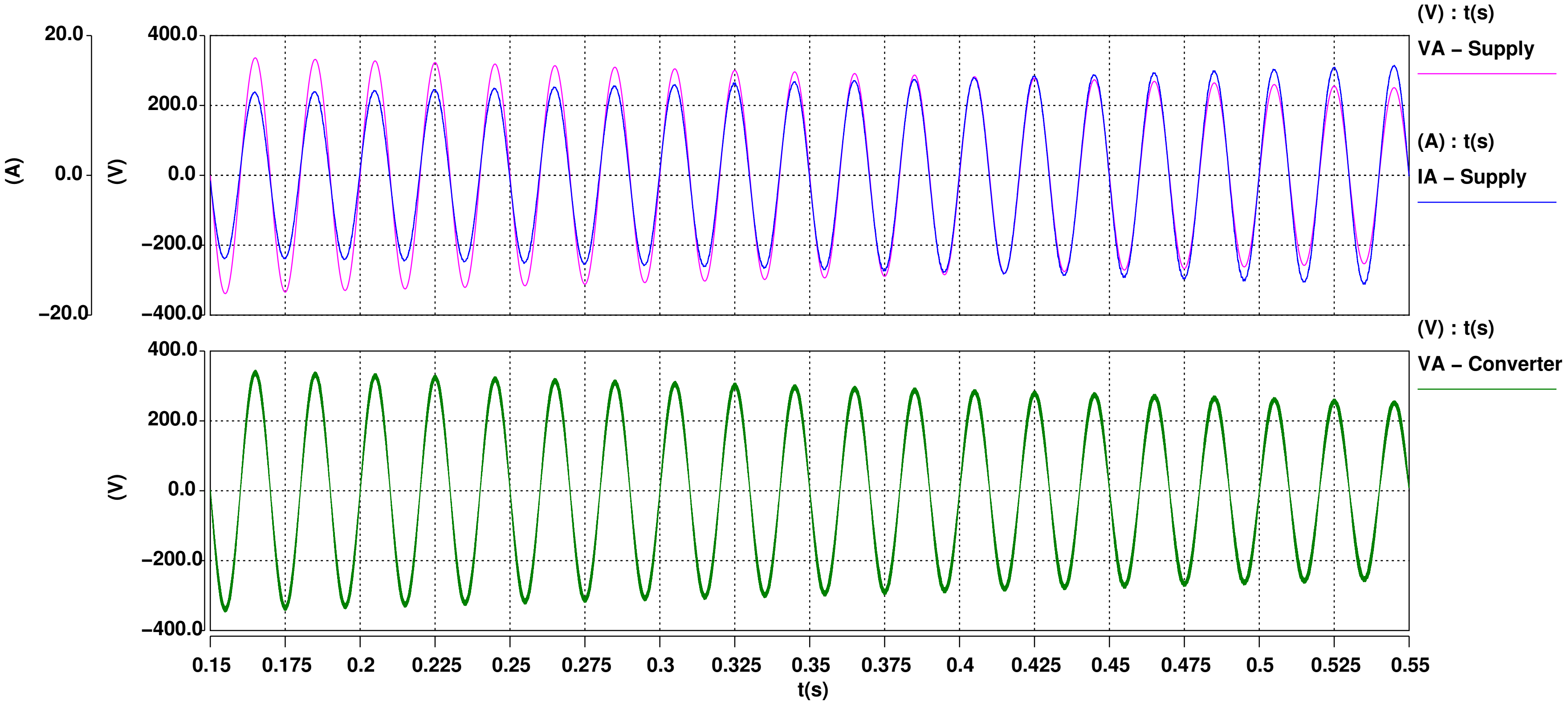


Figure 6.61: Plot of Input Phase A Supply Voltage and Current, along with the voltage seen at the converter, with the peak input voltage decreasing from 340V to 250V
50Hz Input Frequency - 12.5kHz Sampling Frequency

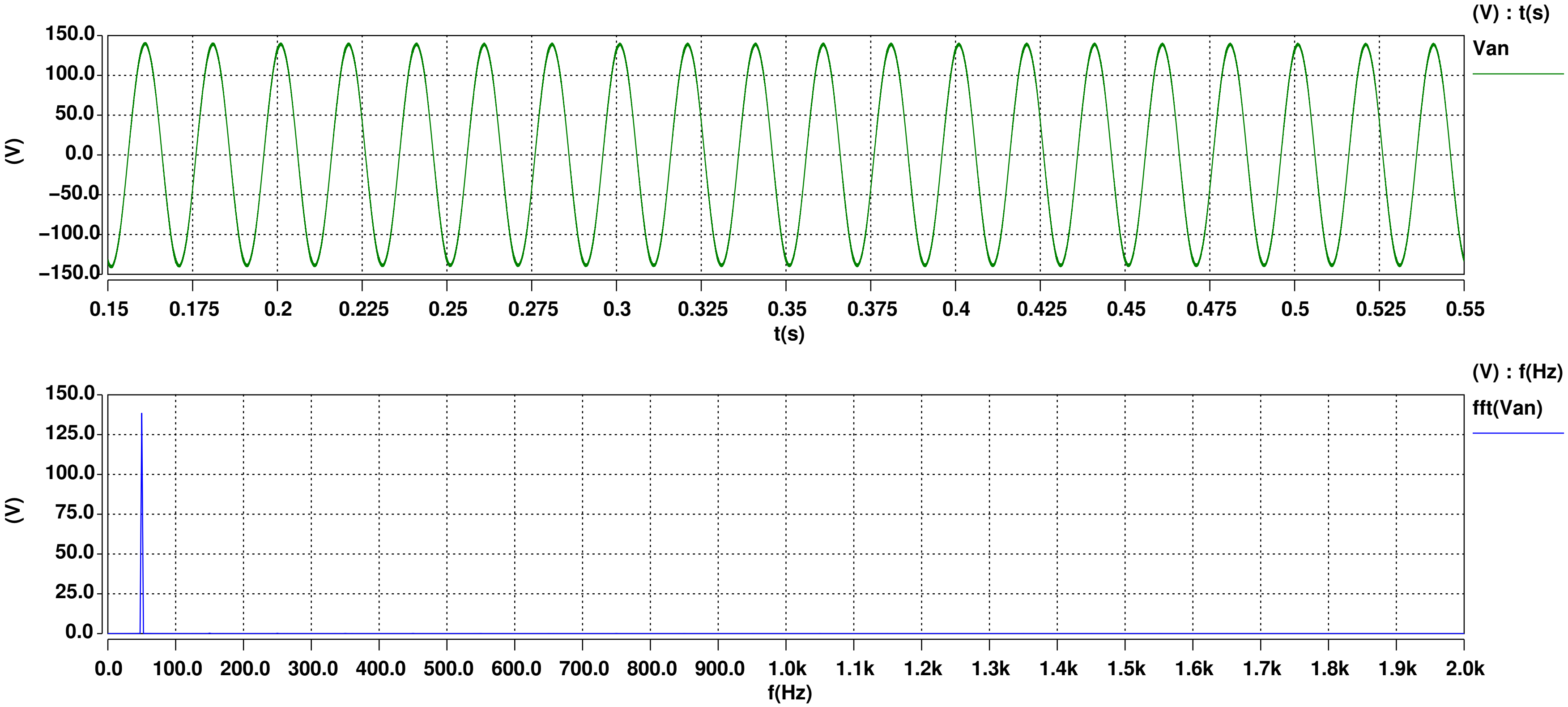


Figure 6.62: Plot of Output Voltage, Van, along with its Spectra, with the peak input voltage decreasing from 340V to 250V 100Hz Output Frequency - 12.5kHz Sampling Frequency

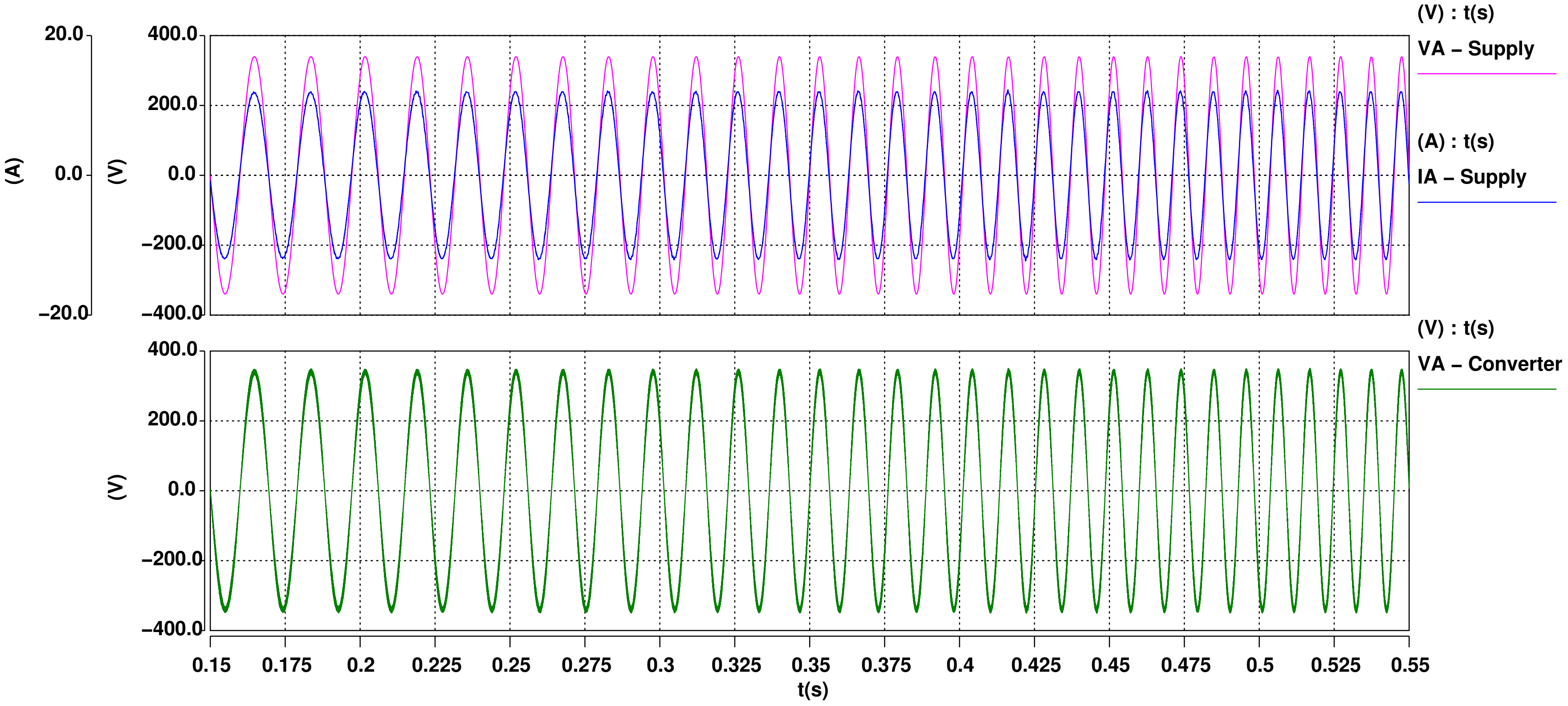


Figure 6.63: Plot of Input Phase A Supply Voltage and Current, along with the voltage seen at the converter, with the input frequency increasing from 50Hz to 100Hz
12.5kHz Sampling Frequency

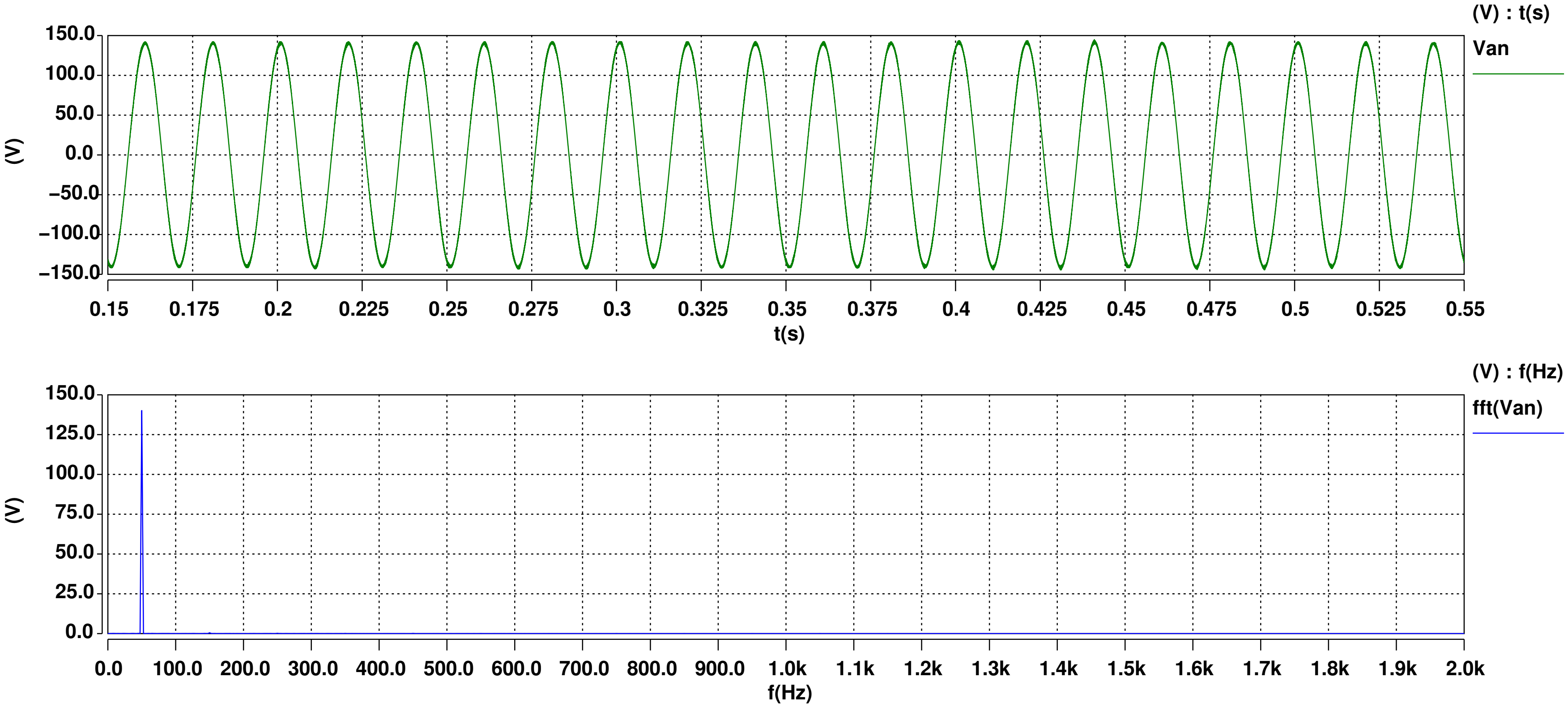


Figure 6.64: Plot of Output Voltage, Van, along with its Spectra, with the input frequency increasing from 50Hz to 100Hz
100Hz Output Frequency - 12.5kHz Sampling Frequency

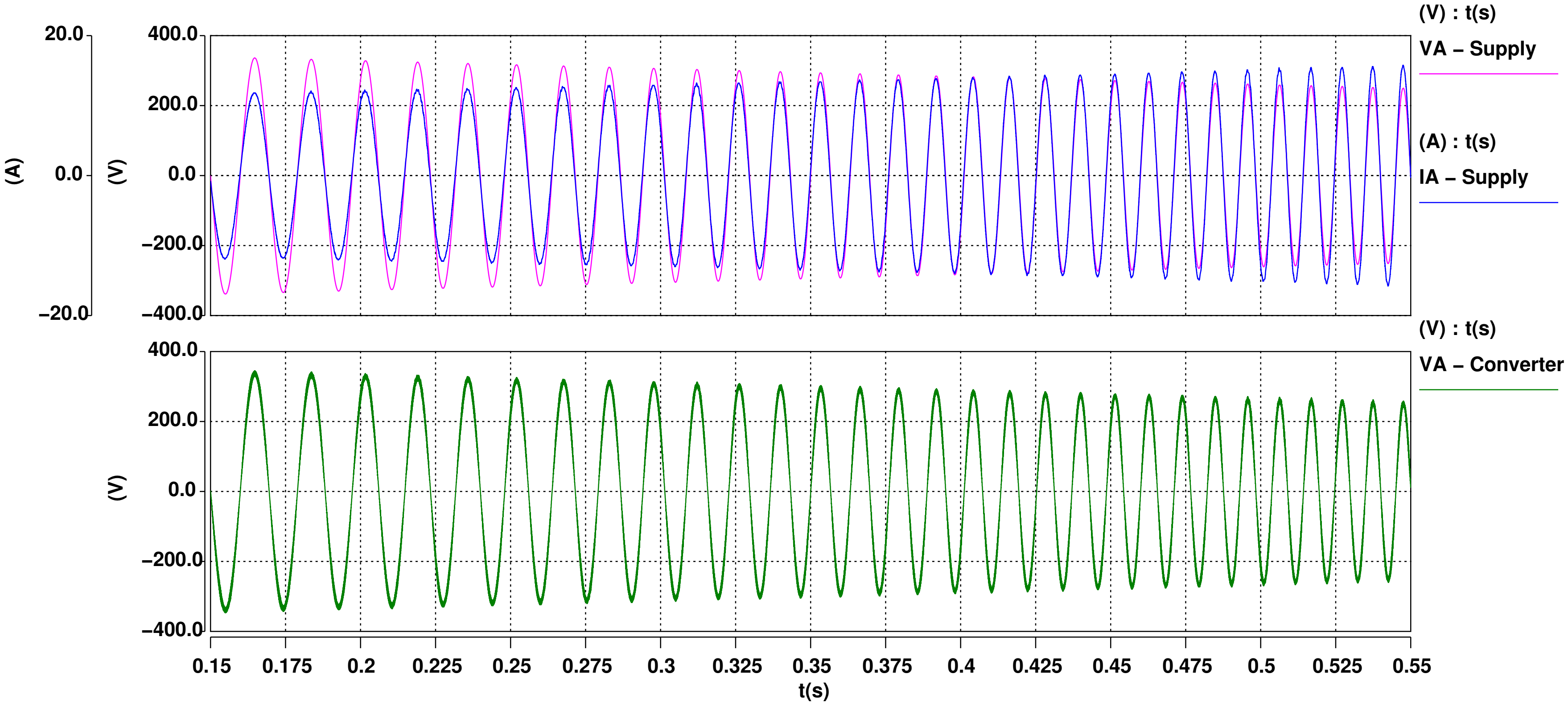


Figure 6.65: Plot of Input Phase A Supply Voltage and Current, along with the voltage seen at the converter, with the peak input voltage decreasing from 340V to 250V while the input frequency increases from 50Hz to 100Hz
12.5kHz Sampling Frequency

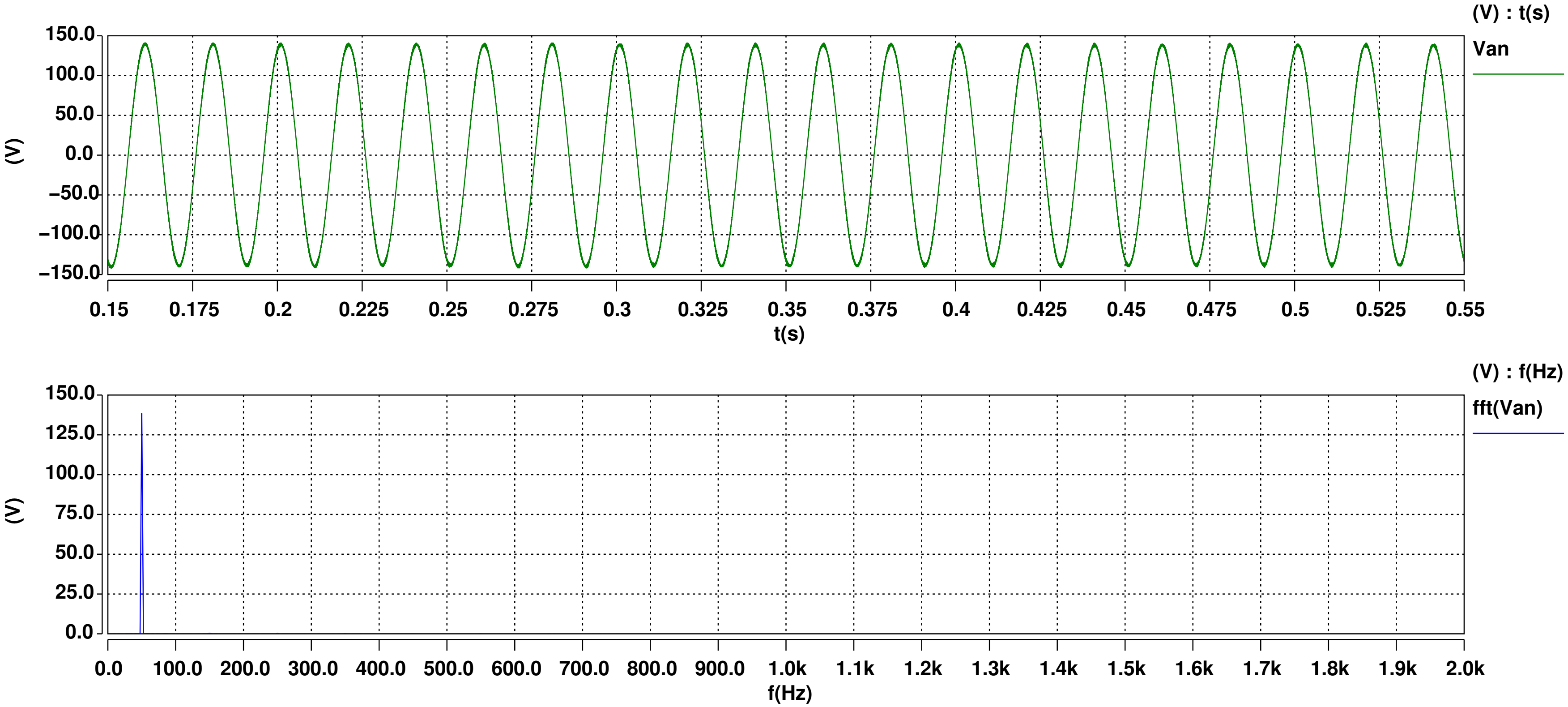


Figure 6.66: Plot of Output Voltage, Van, along with its Spectra, with the peak input voltage decreasing from 340V to 250V while the input frequency increases from 50Hz to 100Hz
100Hz Output Frequency - 12.5kHz Sampling Frequency

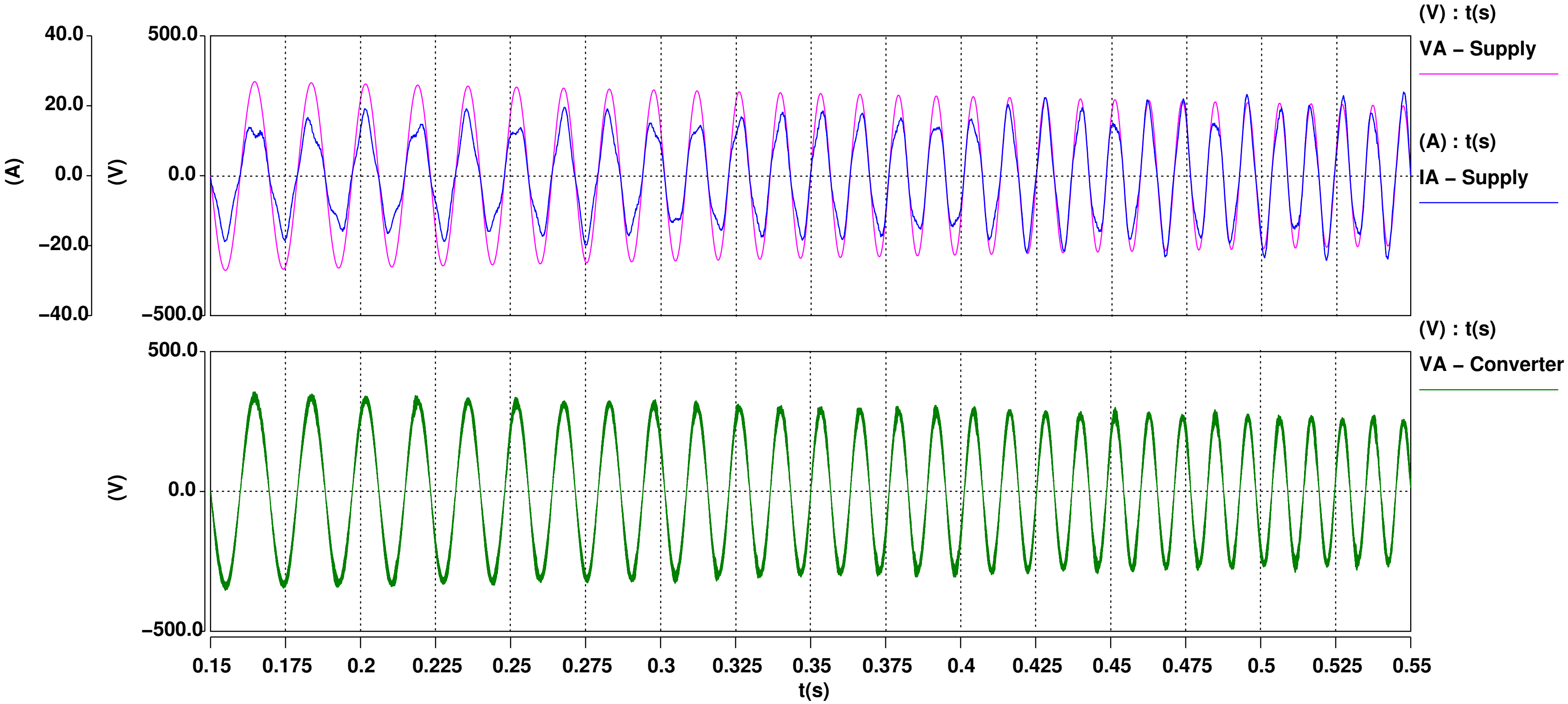


Figure 6.67: Plot of Input Phase A Supply Voltage and Current, along with the voltage seen at the converter, with the peak input voltage decreasing from 340V to 250V while the input frequency increases from 50Hz to 100Hz, driving an unbalanced load
12.5kHz Sampling Frequency

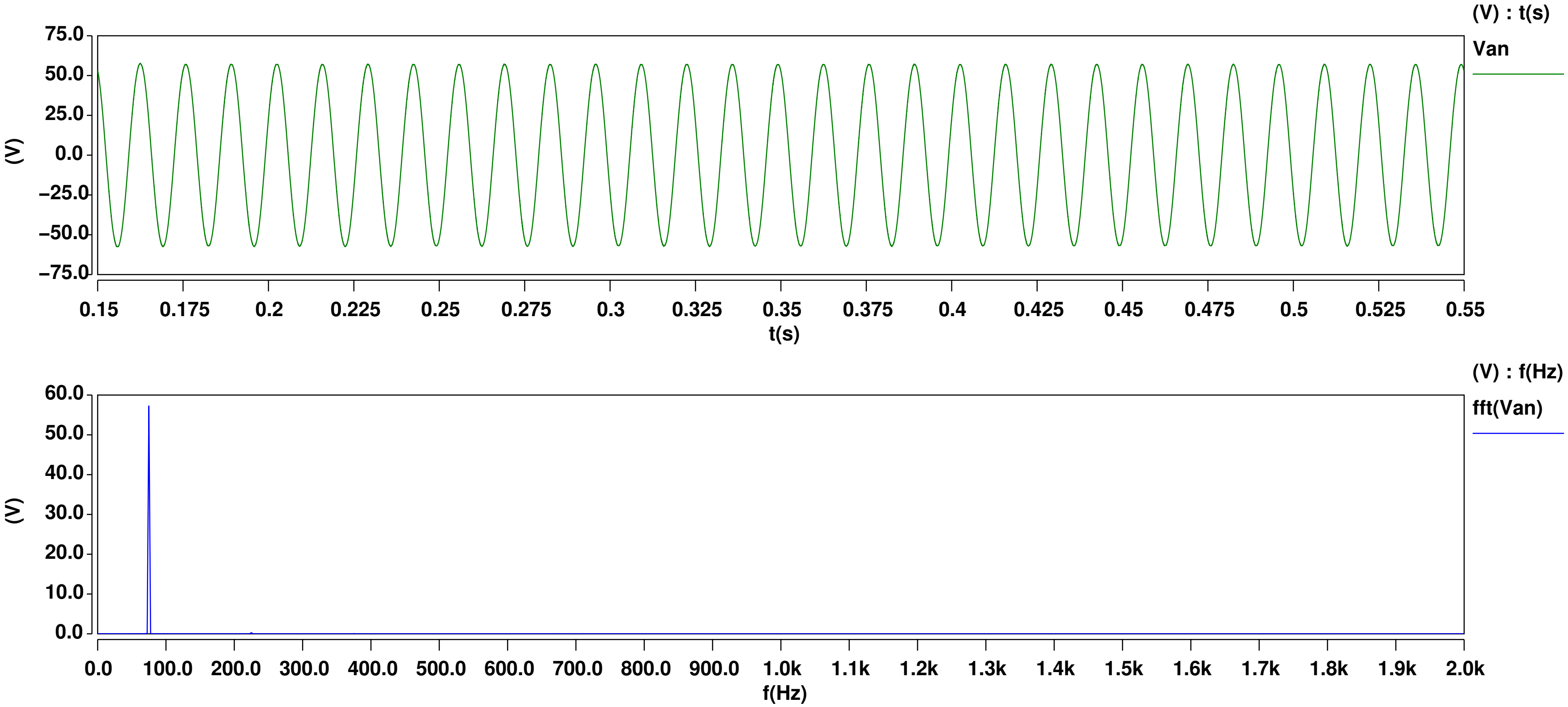


Figure 6.68: Plot of Output Voltage, Van, along with its Spectra, with the peak input voltage decreasing from 340V to 250V while the input frequency increases from 50Hz to 100Hz, driving an unbalanced load
75Hz Output Frequency - 12.5kHz Sampling Frequency

CHAPTER 7

Matrix converter design and construction

7.1 Introduction and overview

This chapter will discuss the construction of a 4-leg matrix converter in order to be able to demonstrate that both the theoretical and simulation work carried out in Chapters 5 and 6 is correct. As such, it should be noted that the converter has been built with a simple open loop control system and is initially based upon the common 3x3 matrix converter which is in use at the University of Nottingham, which was then extended to become the 4-leg (3x4) matrix converter. Because the main scope of this work was to derive the theory behind the 4-leg matrix converter and its basic operation, this physical converter is the proof of the concept, and as such any more complex control system would be outside the scope of this work.

The proposed matrix converter would be designed to meet the following specification:

- Switching Frequency: 12.5KHz
- Input Voltage: 415Vac
- Input Frequency: 50Hz
- Output Voltage: 0 - $\sqrt{3}/2V_{in}$
- Output Frequency: 0 - 400Hz
- Maximum Output Power: 10kW

The basic circuit of a matrix converter is essentially fixed, in that it is a matrix of bi-directional switches which connect a set of inputs directly to a set of outputs, although the layout and type of construction can differ widely. With the power levels involved with this build being relatively small, it was possible to build this converter using a large multi-layered printed circuit board, and this allowed the design of this converter to based on an existing University of Nottingham design. As such, a large amount of the circuit design itself had already been completed, and so the major part of this work was in then going on to modify the existing circuit layout to accommodate the extra output leg.

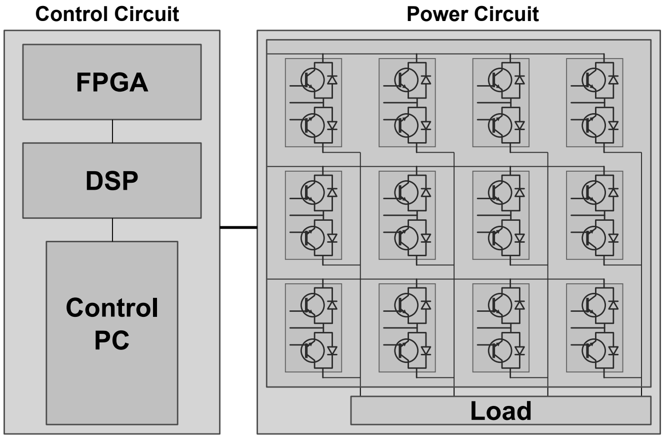


Figure 7.1: Broad Overview of the different areas in a Matrix Converter

Figure 7.1 shows how the system fits together as a whole, with each of the major components shown, and this shows that the matrix converter is easily separated into two main areas:

- The power circuit is the main physical bulk of the converter, it includes the printed circuit board (PCB) which holds the switching devices and their respective gate drive circuits, the along with the current and voltage sensing.
- The control circuit is however the more complex of the two, it is based around a digital signal processor (DSP) board, which connects through to the power circuit through a field programmable gate array (FPGA) daughter-board, and is also connected to a PC to allow the converter to be controlled.

This section will now take a look at each of these areas, breaking each one down into its major components and going into detail on the design and operation where required.

7.2 Matrix Converter Power Circuit

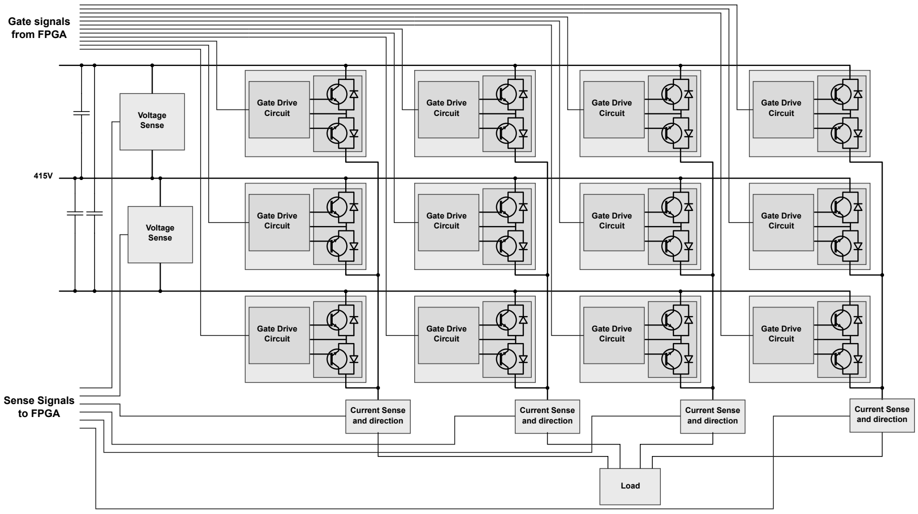


Figure 7.2: Block Diagram of the Matrix Converter Power Circuit

Figure 7.2 shows the general overview of this circuit, and it can be seen that the core of the circuit is very simple, it consists of 12 bi-directional switches arranged in three legs of four switches. For each of these switches there is a dedicated isolated gate drive circuit, and then the output of each of these legs passes through a current sense circuit, before being output to the load. Additionally, there are a pair of back to back diode rectifiers and clamp circuit, while the input voltage has a separate sense circuit as well.

- The bi-directional switches
- Gate drive circuits for the switches
- Voltage and current sensing
- Circuit protection

As stated above, within the PEMC group at the University of Nottingham there already existed a working design for a 3x3 matrix converter, and with the 4-leg matrix converter being a extension of the standard 3x3 matrix, it provided an excellent base for the design of this converter. This matrix design is based around a large multi-layered PCB, using self-contained switching packs with isolated gate drive circuits, and also including the required sensing and protection circuits for the converter. As such the details in the sections below are a description of the need for, and operation of that element.

The following subsections will look in more detail at each of the individual parts of this circuit.

7.2.1 Devices

The bi-directional switches required by this converter need to be capable of passing current in both directions when turned on, and yet blocking current in both directions when turned off. While the required behaviour is possible by using GTOs, the switching times required by this converter means they are not a practical solution in this instance. For the switching speeds required in this circuit, where the overall switching frequency is 12.5kHz, but due to the modulation scheme used the devices themselves switch at 25kHz there the two types of device which it would be possible to use, and these are field effect transistors (FETs) and IGBTs, but neither of which fulfils the bi-directional nature of the switch requirements on its own.

However, by using FETs in a back to back arrangement, so the drains of the devices are connected together, this allows a pair of FETs to exhibit the required behaviour, however, while this situation may work in theory, the reverse conduction characteristics of both the body diodes in the FETs are just not as good as the forward conduction of a discrete diode, and so for this reason anti-parallel diodes are used alongside them. The arrangement for using IGBTs in the directional switch is exactly the same, although with no body diode, they require the anti-parallel diode to operate in the required way. Finally, due to the power level that this converter is being designed for IGBTs are favoured over FETs due to their superior switching and conduction losses. Figure 7.3 shows a circuit diagram of one of these switches.

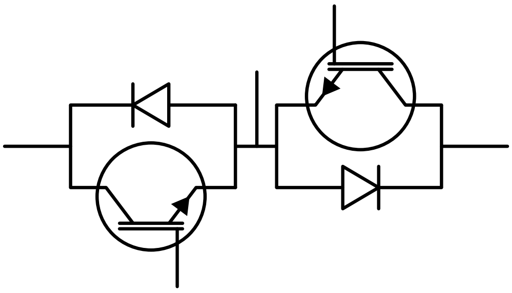


Figure 7.3: A Single Bi-directional Switch

The requirements for the elements of the bi-directional switch in this circuit are the following:

- Reverse breakdown voltage: >600V
- Forward IGBT Current: over 30A
- Forward Diode Current: over 30A

The actual devices used for the converter are the Semikron SK60GM123, which are a pair of IGBTs with anti-parallel diodes in a common emitter configuration, as shown in Figure 7.3, all held in a single package. These are a standard device which is used within the PEMC Group at the University of Nottingham, and are the same devices as those used in the standard 3x3 matrix converter that this converter is based upon. These devices are rated at 1200V and 50A, and so meet the requirements for this converter. The data-sheet for these devices can be found in Appendix B.

7.2.2 Gate Drive Circuit

There are a number of requirements that need to be considered for a gate drive circuit for an IGBT, as well as a specific requirement for a matrix converter:

- Current drive
- Switching speed
- Isolation

As with all switching power converters, the main switching devices are only ever operated in either their fully on or fully off states, with the gate drive circuit designed to switch between the two states as quickly as possible. If we first consider the requirements to fully turn an IGBT on, this requires that the gate-emitter voltage is set to above the threshold voltage level, and for the Semikron SK60GM123 devices used in this project that level is typically 5.5V. However, as there is an inherent capacitance in the gate of an IGBT which needs to be charged in order for the gate emitter voltage to rise. It is therefore a good idea to set the gate voltage higher than is required in order to charge this capacitance more quickly, and thus switch the device on as quickly as possible. To allow this fast switching the gate drive needs to also be able to supply enough current at this higher voltage, in the short period of time that is required for the switching.

Alongside the need to switch on the device as quickly as possible is the need to switch it off as quickly as possible as well, but as well as needing to discharge the gate capacitance there is an extra complication in switching off an IGBT, and this is the tail current. This current comes about by the need to remove electrons from the gate/base of the IGBT as the device switches off in order to stop the device from conducting, and the device can continue to conduct as long as a tail current is present. So, to make sure the device turns off as quickly as possible, the gate voltage is taken negative with respect to the emitter.

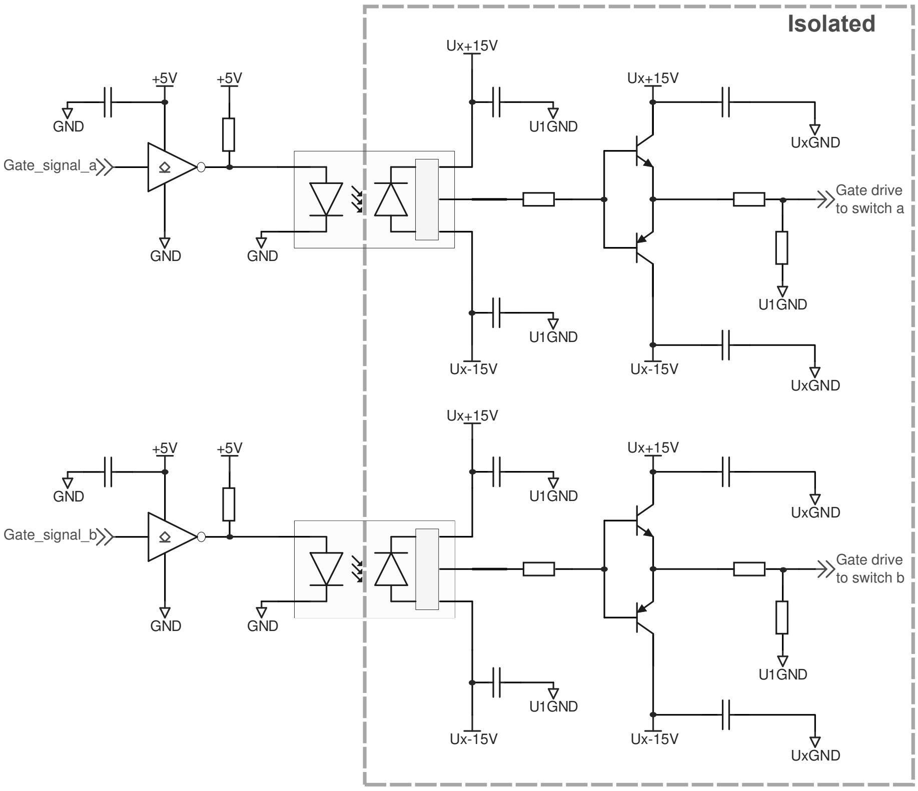


Figure 7.4: Circuit Diagram of the Gate Drive Circuit for a single Bi-directional Switch

For the gate drive circuit for this project, the IGBT gate is driven from +15V and -15V sources. Figure 7.4 shows the circuit diagram for a gate drive for a single bi-directional switch. As can be seen there are a pair of bipolar transistors which are driven in anti-phase to switch the gate circuit between the two voltages, and the gate resistor is used to control the maximum current fed into/drawn from the gate.

Alongside the specific requirements for driving the IGBT itself, the matrix converter also has specific requirement for its gate drives, and that is that each of the gate drives needs to be isolated. This is because the gate drive circuits must always reference the emitter voltage of the devices, and as this voltage can be tied to any input phase, or even just be floating when a device is turned off, the gate drive needs to be able to float as well. To achieve this the entire gate drive circuit, with its respective ±15V rails, is isolated, and these can be seen in Figure 7.4 as the opto-isolated driver for the gate signal, and the isolated power supply for the ±15V rails.

For this circuit, the gate drive signal comes in from the FPGA control board, this then gets inverted by a logic gate before driving the opto-isolated driver. The output signal from this device is then used to drive a pair of transistors connected between the +15V and -15V rails. The output from this drives the IGBT gate through a small resistor.

7.2.3 Output Current Sensing

As this converter is going to be utilising the 4-step commutation sequence, as examined in Section 4.3 above, the FPGA, which generates the control signals, needs to know at any switching instant which direction the current is flowing through each bi-directional switch. To achieve this a current direction sensing circuit is built into the output of each switching leg, and is made up of a pair of anti-parallel diodes. Each pair of anti-parallel diodes are used to monitor the current direction in their particular switching leg, with the voltage across the diodes being positive for current flow in one direction, and negative in the other depending on which diode is conducting. This voltage is then fed into a comparator to produce a logic level signal that is capable of being fed back, through an isolation barrier, to the FPGA. This logic level is then used to determine the commutation sequence required when switching between output states. By using the diodes in this way it is possible to detect down to very low current values so ensuring that the optimum commutation sequence is used. Below these current values where the circuit stops working, the current levels are so small that it makes no real difference which commutation sequence is used.

Alongside the current direction sensing, there is also an inline current transducer which is used for monitoring the output current from each converter leg. This uses a device from a company called LEM which has an isolated output which can then be fed directly out to the A/D converter circuits on the FPGA card. These signals are then used by the DSP to primarily protect the power circuit against any prolonged over-current conditions, although in more complex control schemes these currents can be used as part of the control loop.

Figure 7.5 shows the current sensing circuit that is used for each output leg.

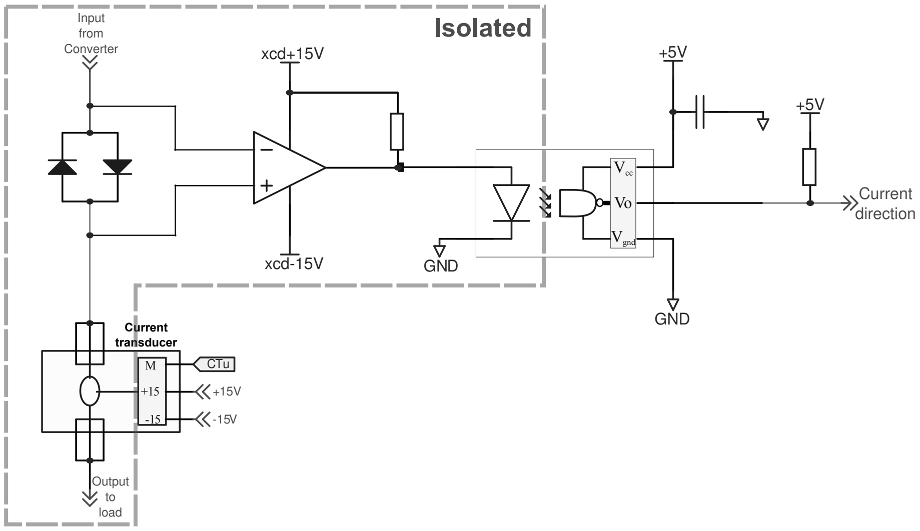


Figure 7.5: Circuit Diagram of the Current Sense Circuit

7.2.4 Input Voltage Sensing

In order for Space Vector Modulation to work it is vital to know both the direction and phase of the input voltage space vector, and to do this we need to know each of the three input voltages. This is achieved in this instance by the use of LEM voltage transducers. Even though the values of all three input phases needs to be known, from the circuit diagram in Figure 7.6, it can be seen that there are only two voltage transducers, and these are being used to measure the two line-to-line voltages V_{AB} and V_{BC}. The reason that the voltage sensing is performed in this manner is that there is no neutral line at the input side of the converter, as the idea behind this type of converter is to be able to generate the 3-phase plus neutral from a balanced 3-phase supply.

So, as it is known that the input phases will form a three phase set where

$$V_A + V_B + V_C = 0 \tag{7.2.1}$$

then, if V_{AB} is known and is defined by 7.2.2

$$V_{AB} = V_A - V_B \tag{7.2.2}$$

and V_{BC} is known and is defined by 7.2.3

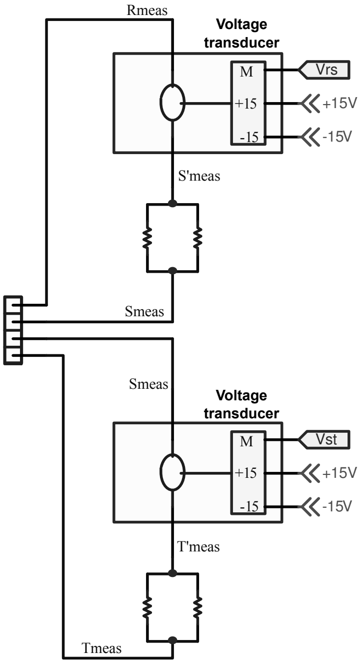


Figure 7.6: Circuit Diagram of the Voltage Sensing Circuit

$$V_{BC} = V_B - V_C \tag{7.2.3}$$

then re-arranging 7.2.3 and substituting into 7.2.1 gives

$$V_A + 2V_B - V_{BC} = 0 \tag{7.2.4}$$

Now doing the same with 7.2.2 gives

$$3V_A + 2V_{AB} - V_{BC} = 0 \tag{7.2.5}$$

and by re-arranging it then gives

$$V_A = \frac{1}{3}\left(2V_{AB} - V_{BC}\right) \tag{7.2.6}$$

From 7.2.6 it can be seen that by just knowing these two line-to-line voltages it is possible to calculate one of the line-to-neutral voltages, and from there the other two line-to-neutral voltages are trivial to solve, such that

$$\begin{bmatrix} V_A \\ V_B \\ V_C \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\left(2V_{AB} - V_{BC}\right) \\ V_A - V_{AB} \\ -V_A - V_B \end{bmatrix} \tag{7.2.7}$$

So, the two line-to-line voltages are all that is required to define the input space vector, and the output from both of these sensors is then fed directly out to the A/D converter circuits on the FPGA card.

7.2.5 Clamp Circuit

This is a relatively simple protection circuit which gives the input and output currents a freewheel path in the case of a device not being switched correctly, and is the parallel diode bridge as discussed in Section 6.2.2.3. While it helps the converter during normal commutation, it also provides some protection for the matrix converter if problems arise a commutation by operating as a voltage clamp. The circuit layout for this is shown in Figure 7.7, and as can be seen, it is a pair of back to back diode rectifiers, one connected to the input and the other to the output. The rectified output from these bridges is then connected to a pair of series capacitors. If one of the switches fails for some reason, then any current in that input or output phase will be able to flow through its respective diode in the clamp and then into/through the capacitor. This stops any large increase in voltage which could occur due to inductances within the input and output circuits if an output leg was allowed to go open circuit. If this is just a problem with a single commutation the time would be small and so the associated voltage increase would also be small, but if there is a larger problem then the increase in capacitor voltage can then be used to trigger a trip within the control circuit and the converter can be stopped. Any currents still flowing in the input and output legs can then be allowed to safely freewheel through the clamp circuit. The associated resistor is there to provide a discharge path for the capacitor, so that when the converter is tripped the freewheeling energy can be safely dissipated.

Included in this circuit is another LEM voltage transducer which is used to sense the clamp circuit voltage, the output of which is fed into a comparator. Once the clamp voltage rises above the comparators reference voltage the output goes high and this signal is fed back into the FPGA board and thus the DSP to enable it to stop the converter.

7.2.6 PCB Design and implementation

This is a multi-layer PCB which along with the top and bottom signal layers also has four internal layers, three of which are shared between the three input and three output phases, with an input and output phase

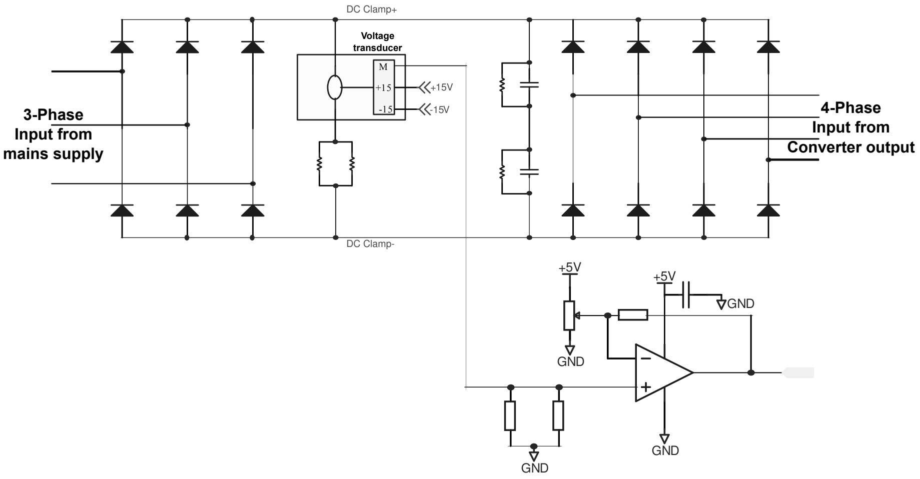


Figure 7.7: Circuit Diagram of the Voltage Clamp used for Converter Protection

per layer, and then another layer to allow a separate ground plane to be used. By using the internal layers in this way, it allows as large an amount of copper as possible to be used for connecting the phases to and from the devices. There is the obvious advantage with having as much copper as possible as it keeps the track resistance down and allows higher power levels to be used, but this also has the additional benefit of allowing the tracks to be as wide as possible and this helps to reduce the stray inductance in the design. While having any stray inductance in the output circuit is not really an issue due to the inductors on each of the output phases, keeping the inductance of the input circuits as low as possible is an advantage. This is because of the nature of the switching, where only two input phases are usually conducting, and so the input currents are discontinuous. As such any stray inductance could cause spikes on the input voltage.

The layout for this board is an extension of the layout for the 3x3 matrix circuit. This layout was arranged with the input phases being connected to the switches in columns down the board, and had the output legs connected across the board. This arrangement was kept, and allowed the 4th leg to be placed along the bottom edge of the PCB.

7.2.7 Input Filter

As the simulations in Section 6.2.2.3 demonstrated, the presence of impedance in the power supply to the converter the deviates from the theoretical ideal and as such the converter requires a number of input capacitors to compensate for this. These capacitors combine with the supply inductance to also form the filter which is required to prevent any switching frequency components of the input current from reaching the supply, and so causing further distortion and problems from other users of that supply.

For this converter, the filter will simply use the supply impedance coupled with a set of three input capacitors, with the arrangement demonstrated in Figure 7.8. This same setup was used as part of the simulations, and assumes a supply impedance of approximately 1mH, and will then use 3μF capacitors to produce a resonant/cut-off frequency of just under 3kHz.

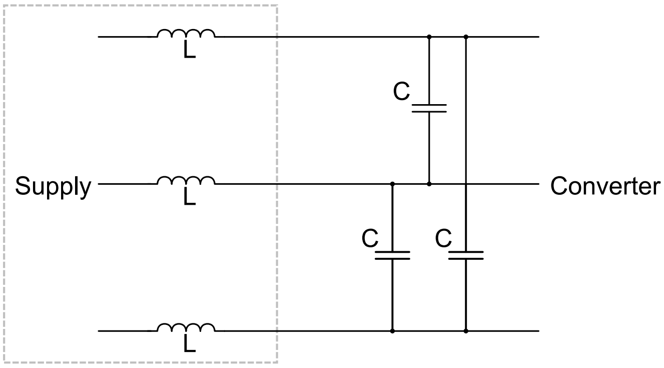


Figure 7.8: Circuit Diagram of the Supply and Input Filter for the 4-Leg Matrix Converter

Choosing a cut-off frequency at this point means that it is still well below the switching frequency being used and so be able to effectively filter it, while also trying to keep the phase shift of the filter as close to zero over the entire input frequency range, up to 400Hz. This last point is important as it allows the converter to have a input displacement factor as close to unity as possible.

7.2.8 Output Inductors and Load

The final part of the power section is the load, and this itself is broken up into two parts, the output inductors and the load resistance.

For a matrix converter to operate it ideally requires some inductance on each of the 4 output legs, thus giving it the current stiff characteristic. The actual inductors used for the converter are a set of 8mH 3-phase inductors attached to the three main output legs, with a pair of 4mH inductors connected in series on the neutral leg. This arrangement was limited in this way because there were no 4-phase inductors available within the PEMC test lab, as 4-phase output inductors are relatively uncommon. While this is not the ideal setup, it has been tested and shown to be acceptable for use with this converter.

The outputs from these inductors are then taken on and connected to the load, which for this converter it is a relatively simple setup, initially consisting of a star connected set of three 10Ω resistors, with the output legs connected to the star points, while the neutral is connected to the centre point.

7.3 Matrix Converter Control

7.3.1 Digital Signal Processor

Due to the relatively intensive nature of the mathematics that is required for the operation of a matrix converter every switching cycle, and with the short time-scales that are available for performing these calculations, a fast processor is required and a digital signal processor (DSP), with its dedicated maths functions and parallel execution paths, is ideally suited to this task. In this case one of the Texas Instruments TMS320C67xx family of digital signal processors has been chosen. While there are other processors available, this family of processors was chosen as it is widely used within the University of Nottingham PEMC research group, and this then also allowed the use of a previously designed FPGA daughter board. This daughter board was necessary as even though the DSP is fast it is still not fast enough to be able to achieve the required resolution for the output of the switching signals to the IGBT gate drives.

In this instance the TMS320C6713 processor was used, this is because, at the time, this was the latest generation of the TMS320C67x family. To allow a quick and relatively easy development process the processor was bought as part of an evaluation kit made by Spectrum Digital. This kit has the processor mounted on a PCB with all the required memory along with the associated peripherals that allow it to be connected directly to a PC and used with the Texas Instruments development environment, Code Composer Studio. Not only does this package allow the software to be written and compiled specifically for this processor, but also that the processor can be simulated within the development environment prior to being run directly on the hardware.

This processor runs at a clock speed of 225MHz, with 4 clock cycles per instruction cycle, then with a sampling frequency of 12.5kHz for the converter, this gives a maximum of 4500 instruction cycles per sampling period.

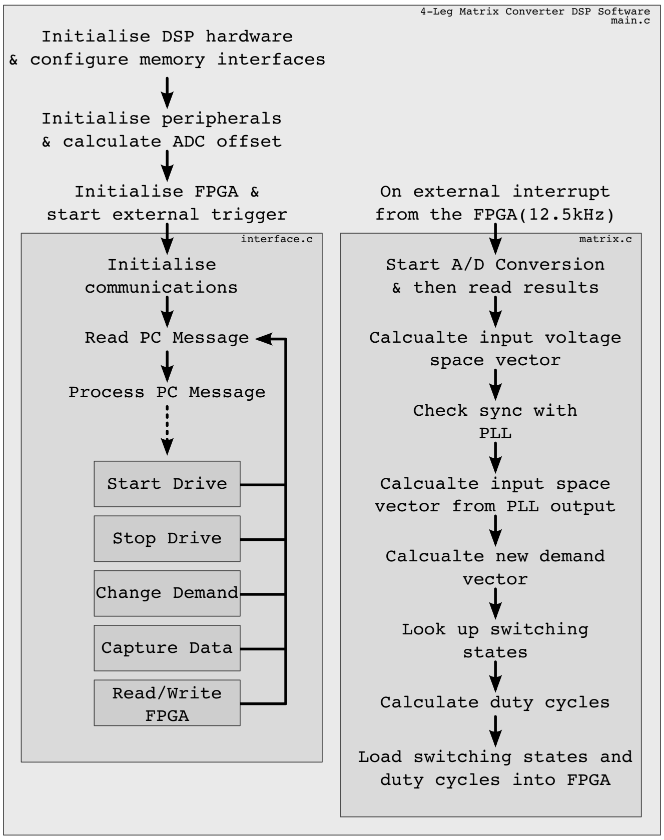


Figure 7.9: Block Diagram for the DSP Software for the 4-Leg Matrix Converter

The software for the DSP is broken up in the 3 main parts, as can be seen in Figure 7.9, with these being the:

- Initialisation and peripheral setup
- Interface and main control loop
- Matrix SVM interrupt code

The initialisation and peripheral setup code block is the first block of code to be run by the DSP when the application starts. This is a straightforward set of functions which are used to initialise and set up the various parts of the DSP and its evaluation board, initialising the main variables and then setting up the FPGA. This code is just run at the start of the program, and its final task is to call the AwaitPCMessage which is the main communication and control loop.

The communication and control loop is the part of the DSP software which, as its name suggests, is where theDSP handles its communication with the controlling PC, and then interprets the messages from the PC to control the operation of the converter. There are a number of commands which the DSP is able to perform, and these include:

- Start the converter
- Stop the converter
- Change the demand voltage or frequency
- Capture some DSP data for the PC
- Read from or write to the FPGA

This is designed as an endless loop which the DSP will never leave, except for servicing the external interrupt from the FPGA which is triggered at the start of every SVM period. There are more details about how the DSP and PC communicate in Section 7.3.3 below.

The main core of the software for the DSP is contained within the interrupt routine which is triggered by an external pulse from the FPGA, and it is this routine which performs all the calculations required to generate the duty cycles and switching states.

The software for performing these calculations is very similar to that used for the Matlab simulations and the MAST template, however, there are a number of notable differences.

The biggest of these differences between the implementation in the simulations and that required for the actual converter is that for the simulations the time stops while all the calculations are being completed for that time instant. This means that the sampled input voltages can be used to calculate the required switching vectors and times and then be immediately used during that sample period, however, with the actual converter this cannot happen as the processing always takes a finite amount of time. For example, the instant the new sampling period starts, the DSP would need to know what switching state to send to the FPGA, but at that instant the DSP has not even had change to read the input voltages, so this first switching state cannot be known at this point.

For this reason the DSP always needs to be calculating the switching vectors and times for the next switching period. But to enable this to happen the DSP needs to know the behaviour of the input voltages in the next sampling period. To achieve this the DSP employs a software phase locked loop, which locks an internally generated space vector to the one calculated from the voltages on the input phases. By knowing the magnitude of this vector, along with its phase and its frequency, it makes it a relatively simple task to be able to calculate the expected input voltages for the next sampling period. The software PLL used in this software, is taken from the software used to control the 3x3 matrix converter upon which the hardware design is based.

Now, for every sampling period, even though the times involved are short, the voltages at the end of the period will be different from those at the start. So if the input voltages are sampled at the start of a period, as time approaches the end of the period, the input voltages will have changed, and so the duty cycles used at that point will be slightly out. This then means that the output voltage at this instant will be slightly different to that expected, and that the input current vector will also be slightly misaligned from the voltage vector, thus giving a small error, which would be seen as a degradation in the harmonic performance of the converter.

However, by extending the way that the output space vector from the PLL is used, this error can be minimised. So, using the fact that the input voltages can be predicted at some time in the future, this means that it allows the choice of any point within the next sampling period, for example the mid-point in the sampling period, to base the duty cycle and switching state calculations on. Then taking this once step further again by considering the type of switching sequence being used, it will be noticed that the switching sequence is separated into two half cycles$seeFigure$ 6.21b. This, combined with being able to predict the voltages, means that instead of just performing the duty cycle and switching state calculations once for a whole sampling period, it is possible to do these calculations for each half-cycle separately, using the voltages predicted at the mid-point in each half-cycle as the input reference.

Obviously, by using this approach it will double the time taken for the DSP to perform the duty cycle and switching state calculations, and so during the development process a check needed to be kept in place to ensure that the time taken to execute the interrupt routine was always less than the sampling period. During testing of the DSP software it has been shown that the interrupt routine takes 60μs to complete which is well within the sampling period of 80μs.

The next difference is how the DSP outputs the switching information to the converter itself. In the Saber simulations these signals were sent directly from the block performing the calculations, however, due to the time resolution that is required for accurate switching the DSP is just not fast enough, and so the FPGA is used to manage the actual gate switching signals for the devices. However, there still needs to be some method for the DSP to tell the FPGA what switching states and timing to used. This task is performed by a First-In First-Out (FIFO) buffer built into the FPGA, the DSP then loads this with the switching state and timing information for each sampling period. There is some care which is needed by the DSP when handling this information though, firstly to ensure that the time for a single to switching state does not fall below a certain value, and secondly to make sure that the total time for all the switching states, including the zero states, for this sampling period exactly equals the length of the sampling period. Once these arrangements are in place, the switching states with their respective timings are loaded into the FIFO, first to last, for both half-cycles of the sampling period.

The interrupt routine then exits, and the DSP returns to the communication and control loop to await either the next interrupt, or a command from the PC.

The DSP software has been written in C and compiled specifically for the C6713 processor using the Texas Instruments Code Composer Studio - Platinum Edition which is supplied with the evaluation board.

7.3.2 Field Programmable Gate Array

Due to the complex nature of the switching sequences involved with the operation of a matrix converter, and their need for short and precise timing within each switching sequence, a field programmable gate array (FPGA) is used to interface the DSP to the gate drive circuits. Using the FPGA also provides an easy way to provide an interface between the DSP and a number of other external functions which are required, the most important of which is the A/D Converters, and by placing these under the control of the FPGA, it frees up the DSP to concentrate on the calculations.

Figure 7.10 is a block diagram of the main components of the FPGA, which shows that it handles the following functions:-

- A FIFO which holds the vector information
- A PWM timing module which performs the timing for each vector
- 4 off Four-Step Commutation blocks which translate the vector information into switching signals

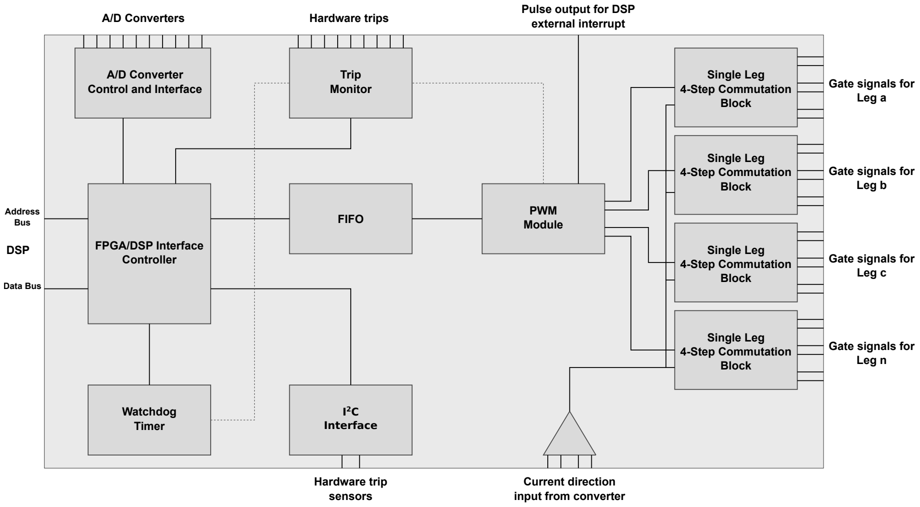

A/D Converters
Hardware trips
Pulse output for DSP external interrupt
A/D Converter Control and Interface
Trip Monitor
Single Leg 4-Step Commutation Block
Gate signals for Leg a
Single Leg 4-Step Commutation Block
Gate signals for Leg b
Address Bus
DSP
Data Bus
FPGA/DSP Interface Controller
FIFO
PWM Module
Single Leg 4-Step Commutation Block
Gate signals for Leg c
Single Leg 4-Step Commutation Block
Gate signals for Leg n
Watchdog Timer
I²C Interface
Hardware trip sensors
Current direction input from converter

Figure 7.10: Block Diagram showing the main components of the FPGA

- An A/D Converter module which controls the A/D converters, and then demultiplexes the resulting data
- A Watchdog timer that will trip the converter if required
- A set of Hardware and software trips

The overall design of the FPGA is such that it is seen by the DSP as a series of memory addresses, which it is able to read from and write to, and this interface is briefly described in Appendix C. By having the interface arranged in this way, it allows the registers and functions of the FPGA to be accessed directly by the DSP, this is greatly advantageous as it means that no special routines are required for reading from or writing to it, and especially so with respect to the A/D data.

The core functions of the FPGA are set up by the DSP on initialisation, the most important here being the PWM sampling period, as this controls the interrupt that is sent to the DSP, along with providing all the timing information for the 4 different commutation blocks. Once the sampling period is set, the various parts of the FPGA can be initialised, the outputs turned on, and the FPGA at this point is ready to go.

The most important function for the FPGA is that of generating the gate control signals which go straight to the gate drive circuits on the power board. The sequence for this originates in the previous sampling period where the DSP has calculated the switching states and times for this sampling period, and this data has been loaded into the FIFO buffer in the FPGA. Then, as the new sampling period starts, the first switching state out of the FIFO buffer gets output by the PWM module, which is then read by the commutation blocks. This information stays on the outputs of the PWM module for its specified switching time, before being changed for the next switching state. The length of time it stays on the outputs is controlled by the timing information which the DSP sent. This states how many FPGA clock cycles that switching state is used for, and with the FPGA clock running at 50MHz, this gives a possible resolution of 20nS in the switching times.

The commutation blocks are constantly looking for changes in their inputs, which will then cause the commutation process to start. With each commutation block looking after a single switching leg, this means that there are only 3 possible final output states as only a single switch in a leg can be turned on at the same time. So, the commutation block reads in the value from the PWM module, and if that is currently the same as already on the outputs, it does nothing, if this value is different it then commutates to the specified leg using the 4-step process, until once again the output of the block is the same as the input. The only other input to this block is from the current direction. As was shown in Section 4.3, exactly how the 4-step process operates during a single commutation depends on the current direction in that output leg.

This process with the DSP filling up the FIFO, and the PWM module emptying it needs to be carefully controlled, as it is a fine balance. If the sum of all the times the DSP places in the FIFO in a sampling period, to be used in the next sampling period, does not equal the length of the sampling period, the FIFO will start to under, or over, run and as this happens, rouge switching states will sent to the outputs, so this process has to be carefully controlled by the DSP.

Overall, the code for the FPGA is based on, and very similar to that used by the 3x3 converter, however the extra output leg did require a number of changes to the overall design, and these changes were made using Actels release of the Libero FPGA design platform.

The layout for the FPGA board itself is no different to that used by the 3x3 matrix converter, and primarily consists of a number of A/D converters which are used for sampling the voltage and current sense signals from the power board. Built into this circuit there are also a number of hardware trips, with controllable trip levels via an integrated I^2C bus which has been built into the FPGA.

7.3.3 Host Port Interface

In order to interface to the software running on the DSP, and thus to control the operation of the converter, a PC needs to attached to the DSP. Initially this was performed using the Universal Serial Bus (USB) port built into the Spectrum Digital DSP evaluation board. This had the advantage that there already existed a set of commands within the DSP evaluation board that allowed the PC to communicate with the DSP, and so it allowed quick and easy communication. There was a problem with this method, that although this method worked for small amounts of data, it always required the intervention by the DSP as part of the communication process, and as the inbuilt commands used an interrupt driven structure this meant that the DSP could be interrupted during a time-critical part of the software, and so interfere with the operation. This was not an issue when during the early stages of the development work, but did start to cause problems during the later phases.

This problem was cured by using the Host Port Interface (HPI) facility of the evaluation board which can be utilised by using a small daughterboard made by Educational DSP Ltd. This small board plugs into the HPI port, which is one of the multi-way headers on the DSP board, and through this board a PC is allowed direct access to the memory on the DSP board. This then lets the PC to read and write directly into the DSP memory space without taking up any processor time with the communication. With this method, the only overhead in the DSP is the time required to check if a new message/command is waiting to be read, and to then respond appropriately, and most importantly, as this does not need a DSP interrupt to operate, there is no chance of the communication between the PC and DSP interfering with any of the time-critical sections of code.

The way this interface is achieved is by having the DSP reserve an area of memory within its memory space, which would be used specifically for the transfer of data between the PC and DSP. This area is then split up into four regions:-

- PC -> DSP Messages
- Trip Status
- Captured Data Area

The first area, for PC -> DSP Messages, is used for the PC to send control messages to the DSP, such as to start or stop the converter, or to change the demand voltage or frequency. During the period of time when the DSP is effectively idling, so when not performing the main interrupt routine, the DSP sits and loops round a short section of code which continuously checks the PC -> DSP Message area for a new message. It will check to see if the message checksum is correct, and at that point it will process the message.

The next memory area that the PC has access to is the Trip Register. This is a copy of the FPGA trip register which is updated every time the DSP restarts the idling loop, so it is continuously update

The final memory area which is made available to the PC is the Capture Data area, this, as its name suggests, it where the PC can read any data which the DSP has been asked to save. The data stored in this area depends on which of the three different commands has been sent to the DSP by the PC.

The first command just asks for a single snapshot of a series of the DSPs internal variables to be saved which is then read by the PC and displayed. This is useful for quickly checking that the converter is functioning as expected and is making sane calculations.

The second, and most useful, command is for the DSP to capture a set of its data on every interrupt cycle, over a period of time. This data is then uploaded to the PC where, for example, the Matlab simulation files can be run using the uploaded set of input data, and the results compared.

The third command asks for the DSP to interrogate each of the FPGA registers in turn and save the result into the Capture Data area. This then allows the PC to read the FPGA information, to check on register settings and ensure that the FPGA is working correctly.

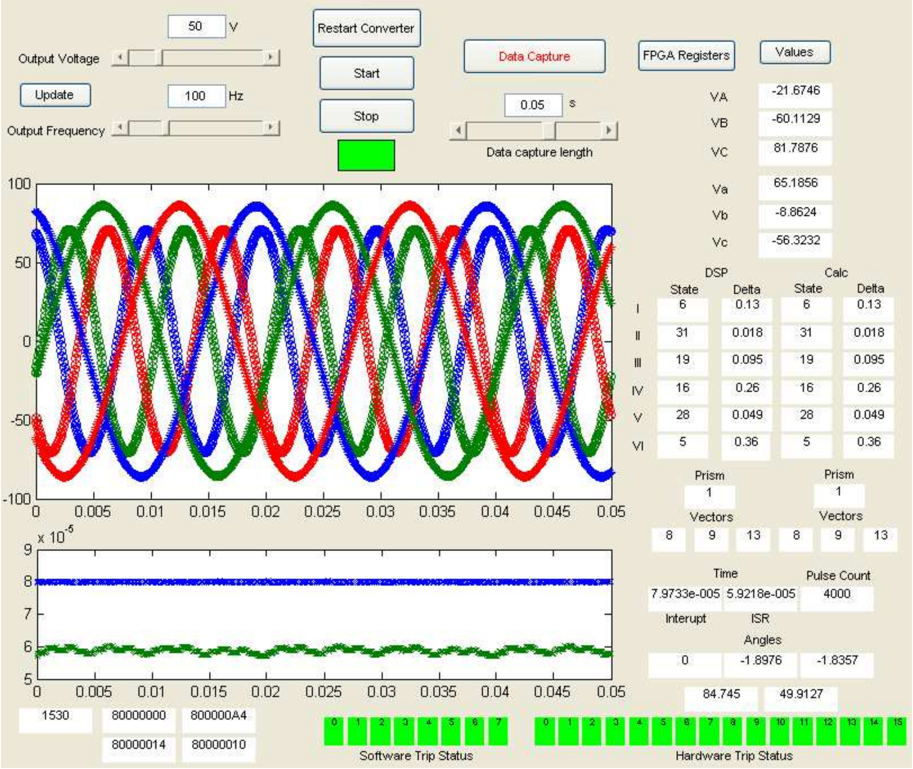
To be able to access this memory from the PC required a series of Matlab executable (MEX) files to be written. These allow Matlab function calls to interface with the HPI libraries supplied with the daughterboard, and for each type of read and write function a separate MEX file is required.

7.3.4 Matlab/PC

To enable some control over the DSP, and thus the converter, a Matlab graphical user interface (GUI) was designed that would allow the PC to stop and start the converter, reset it, and also change the demand voltage and frequency. During the development of the DSP software, it became apparent that due to the complexity of the calculations, it would be difficult to verify if the DSP was actually working correctly, so alongside the command functions, a number of additional functions was added so that the PC could request data from the DSP. This data could then used to verify the DSP results, using the Matlab functions which has already been designed for the simulation.

Figure 7.11 shows the main window of the interface. The graph which is plotted in the centre of the window is from a data capture, and shows the input voltage, demand voltage and calculated output voltage all read from the DSP. As well as those three plots, there is a fourth set of data plotted which is the set calculated on the PC. If the DSP is working as expected, then the demand voltage and two calculated voltage plots should all match.

The second plot window shows the interrupt execution time and the sampling period plotted together, and was used during the development of the software to ensure that the execution time for the interrupt routine stayed well within the sampling period.



Output Voltage: 50 V

Output Frequency: 100 Hz

Data capture length: 0.05 s

VA	-21.6746
VB	-60.1129
VC	81.7876
Va	65.1856
Vb	-8.8624
Vc	-56.3232

	DSP		Calc	
	State	Delta	State	Delta
I	6	0.13	6	0.13
II	31	0.018	31	0.018
III	19	0.095	19	0.095
IV	16	0.26	16	0.26
V	28	0.049	28	0.049
VI	5	0.36	5	0.36

Prism: 1 | Prism: 1

Vectors: 8 9 13 | Vectors: 8 9 13

Time		Pulse Count
7.9733e-005	5.9218e-005	4000
Interupt	ISR	

Angles: 0 -1.8976 -1.8357

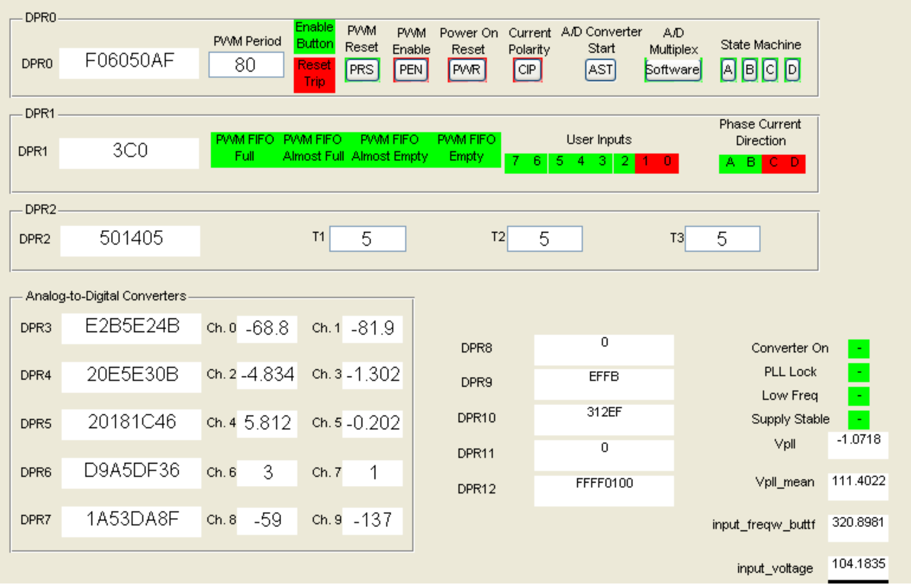
84.745 49.9127

1530 80000000 800000A4

80000014 80000010

Figure 7.11: The Main Control Interface for the 4-Leg Matrix Converter

The second window in the Matlab interface, which is shown in Figure 7.12. is used to show information about the state of the FPGA registers, the outputs from the A/D converters, and of a number of other internal DSP variables. This data is regularly sampled by the PC, and so this window automatically updates its values.



DPR0

DPR0 F06050AF

PWM Period 80

DPR1

DPR1 3C0

DPR2

DPR2 501405

T1 5

T2 5

T3 5

Analog-to-Digital Converters

DPR3 E2B5E24B Ch. 0 -68.8 Ch. 1 -81.9

DPR4 20E5E30B Ch. 2 -4.834 Ch. 3 -1.302

DPR5 20181C46 Ch. 4 5.812 Ch. 5 -0.202

DPR6 D9A5DF36 Ch. 6 3 Ch. 7 1

DPR7 1A53DA8F Ch. 8 -59 Ch. 9 -137

DPR8 0

DPR9 EFFB

DPR10 312EF

DPR11 0

DPR12 FFFF0100

Converter On -

PLL Lock -

Low Freq -

Supply Stable -

Vpll -1.0718

Vpll_mean 111.4022

input_freqw_buttf 320.8981

input_voltage 104.1835

Figure 7.12: The FGPA Registers Display Window

CHAPTER 8

Experimental Results

This chapter describes the testing which was performed on the 4-Leg Matrix Converter described in the previous chapter. The initial testing was all designed around ensuring that the converter was operational, before moving on to use the converter to drive a real load, both balanced and unbalanced.

8.1 Initial Testing

The start of the initial testing of the converter proved to be tricky as the code for the DSP, although tested and simulated within the design environment, was still untried with real world signals. In the end the problems mostly due to problems with the timing between the DSP and the FPGA, centring around the FIFO for the PWM module. However, after the initial teething troubles were overcome the converter was shown to be working fully. One small trick that helped here was to be able to use the onboard D/A converters on the DSP card to produce a simulated pair of voltage signals. This allowed the testing of the phase-locked loop and a number of other functions without needing to have sensors connected to a mains supply all the time.

During the latter stages of the initial testing of the converter, when it was undergoing mains level testing, it was discovered that when in the input voltage from the variac was turned up above approximately 50% of the mains supply it would cause a number of diodes in the protection circuit to fail. This initially was put down at a problem with a device shorting to one of the clamp rails, but after replacement the same problem was re-occurring. At this point the problem is still unidentified, although it is suspected that this might be down to problems during high frequency spikes of current from during the commutation phase, which are then causing the clamp voltage to rise.

However, even with this problem, it allowed the converter to function, although a reduced voltage and power levels, and even at these lower levels the results gained are still valid.

8.2 Balanced Load

Once initial testing had been completed, the converter was connected to a 415V supply, through a variac, and then attached to its load, made up of 4 8mH inductors and a star connected set of 3 10Ω resistors as described in the previous chapter. The following results show results show that the converter is working as expected.

Due to the trouble with the diode protection circuit, the converter was being run with an input voltage of approximately $80V_{rms}$, as such the demand voltage was set to a figure of $50V_{rms}$ and with a demand frequency of 100Hz. The results for this test follow.

Figure 8.1 shows the set of output voltages measured across the load resistances.

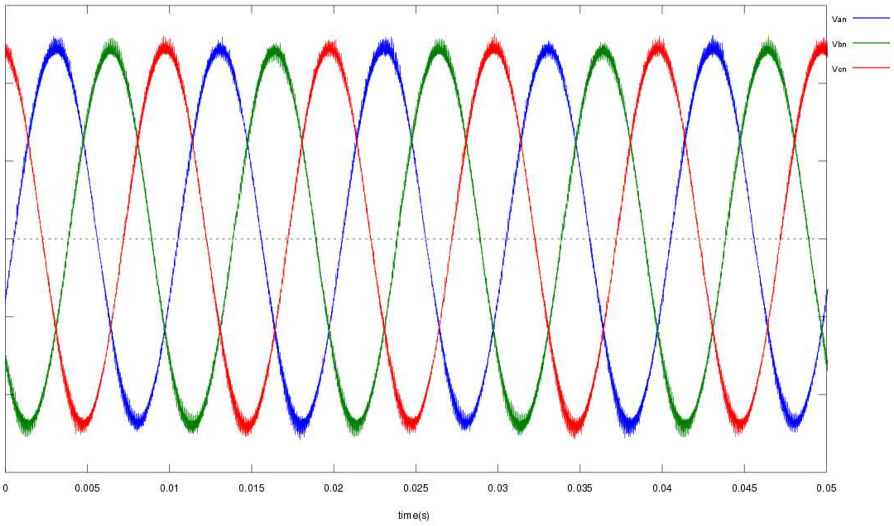


Figure 8.1: Plot of the 4-Leg Matrix Converter Output Phase-neutral Voltages at the Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

Figure 8.2 then provides a closer look at these waveforms, and it shows that while there is high frequency switching noise, present and especially noticeable at the peaks of the sine waves, the overall shape appears to be good, and this is born out by the FFT of the output voltage on leg Van which is shown in Figure 8.3.

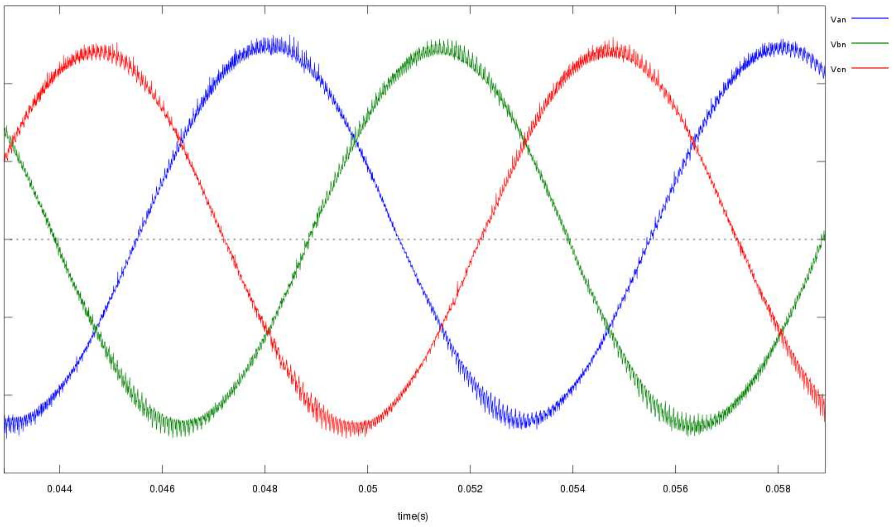


Figure 8.2: Plot of the 4-Leg Matrix Converter Output Phase-neutral Voltages at the Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

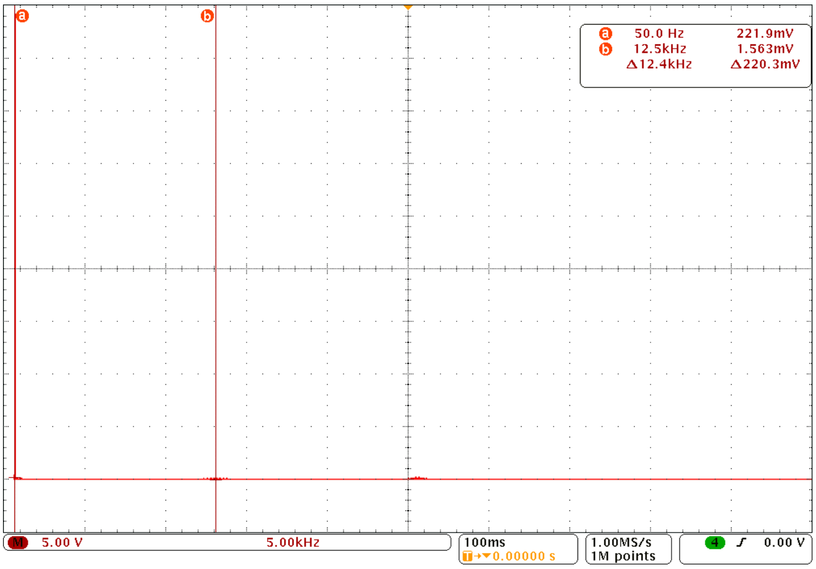


Figure 8.3: Plot of the FFT of the Output Phase-neutral Voltage on Leg **a** at the Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

Figure 8.4 shows the an area of the FFT in Figure 8.3 which has been zoomed in vertically 25 times. As can be seen, the switching frequency noise at around 12.5kHz is at this point only at approximately 0.5% of the magnitude of the fundamental. As will also be noted, there does appear to be some small amount of low frequency harmonics present as well, and these are most likely to be due to differences in the PLL-derived input voltage and the actual input voltage.

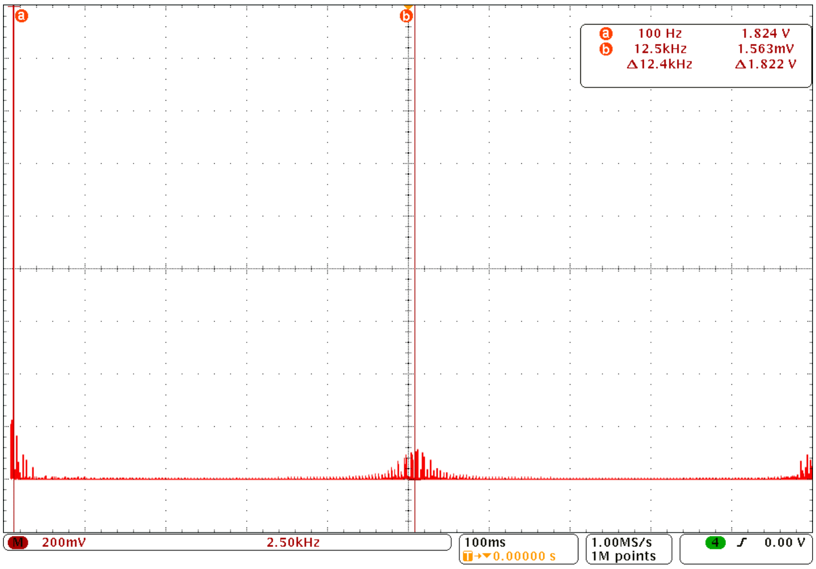


Figure 8.4: Closeup of the FFT of the Output Phase-neutral Voltage on Leg **a** at the Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

Looking at the plots for both the spectra and waveforms for the output voltage at the load, these both in tie in extremely well with the plots from the Saber simulations shown in Figures 6.30 and 6.31.

Taking a step back towards the converter terminals now, Figure 8.5 shows the FFT of the voltage at the terminals of the converter. Once again, this waveform looks very close to the simulated waveform in Figure 8.3.

Moving onto the front end of the converter now, to look at the input currents. These are shown in Figure 8.6, and as can be seen, these are also nicely sinusoidal, although there is some amount of ripple in the waveform in places. These points appear to tie in with the boundaries between input current segments, however despite efforts to find the cause of this problem, they have remained. One suspicion was that these are caused by the difference between the actual input voltages and the PLL derived ones, and as can be seen in Figure 8.9 this ripple does appear to some extent on the input voltages.

Now looking at the FFT for the input current in Figure 8.7, once again this shows a clean spectrum. The ripple which was noted in the current

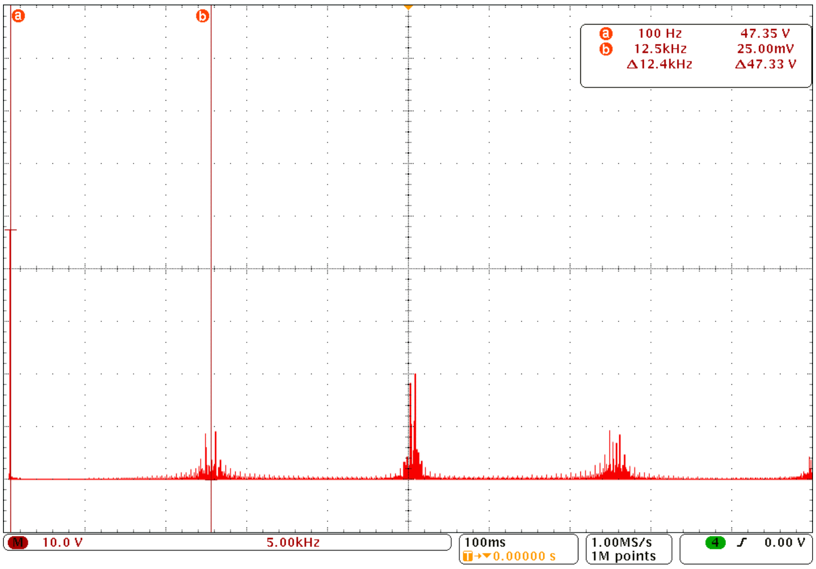


Figure 8.5: Plot of the FFT of the Output Phase-neutral Voltage on Leg **a** at the Converter
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

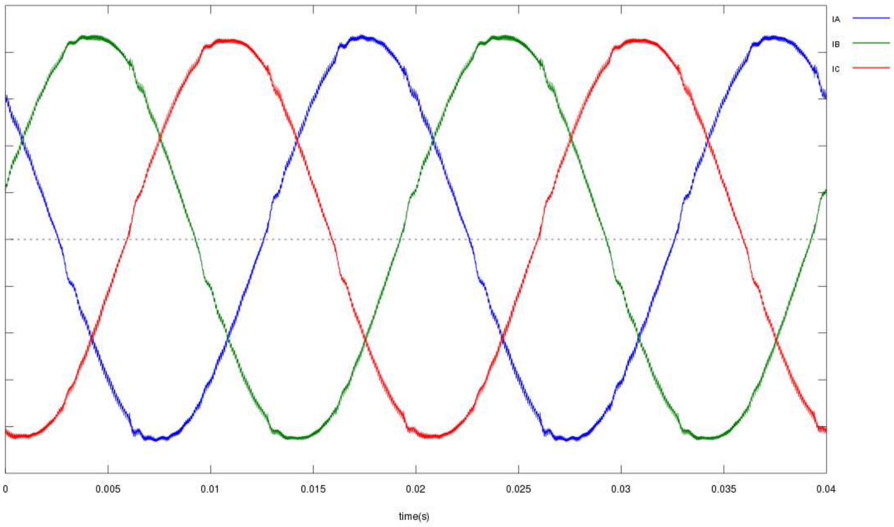


Figure 8.6: Plot of the 4-Leg Matrix Converter Input Currents
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

waveforms can be seen as the low frequency noise, and looking more closely at that region in Figure 8.8 it can be seen that for the input, this ripple is of greater magnitude that the switching noise, however in this case, the ripple is still well below 1% of the magnitude of the fundamental.

While the input current being sinusoidal is important, it is also a re-

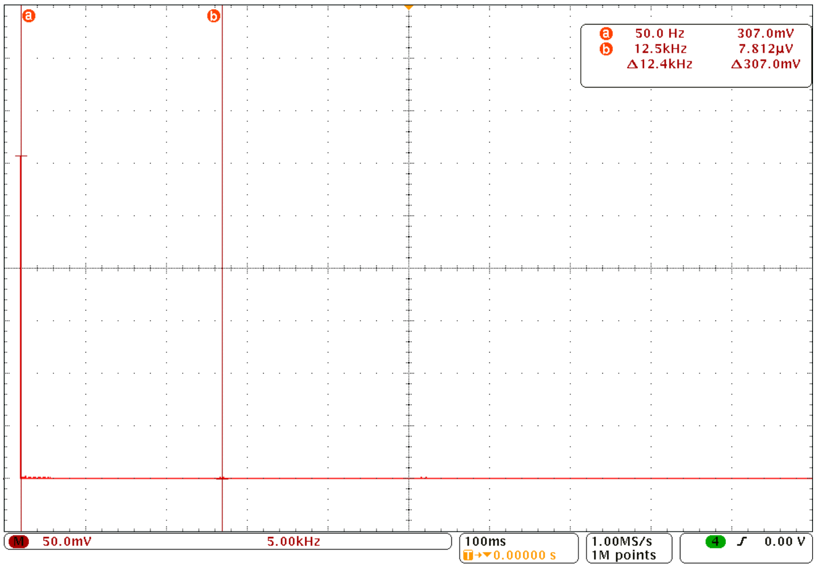


Figure 8.7: Plot of the FFT of the Input Current of Phase **A** $50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

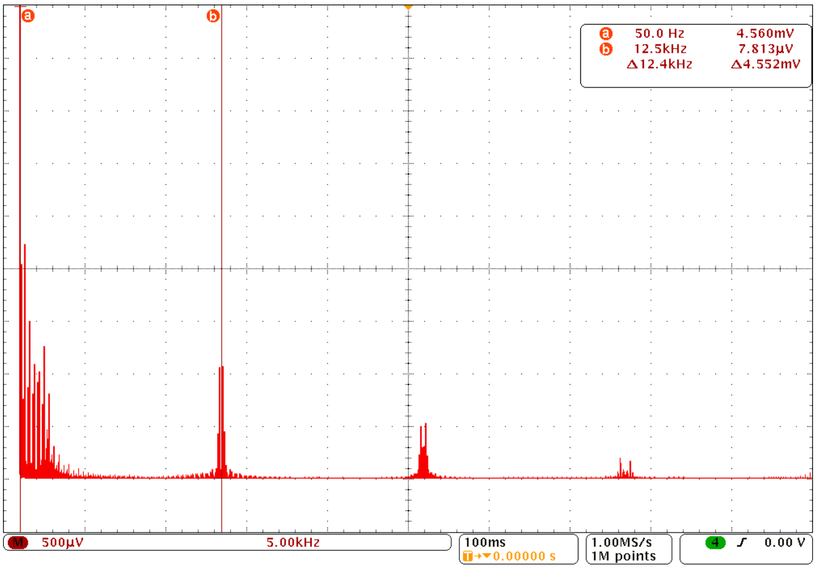


Figure 8.8: Closeup of the FFT of the Input Current of Phase **A** $50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

quirement for the input current to be in phase with the input voltage, and Figure 8.9 shows that well, and so means that while supplying a balanced set of 3-phase voltages to a balanced load, the converter has a unity displacement factor.

Due to the Matlab controller being able to download various sets of data from the DSP during the converter operation, it is possible to see the

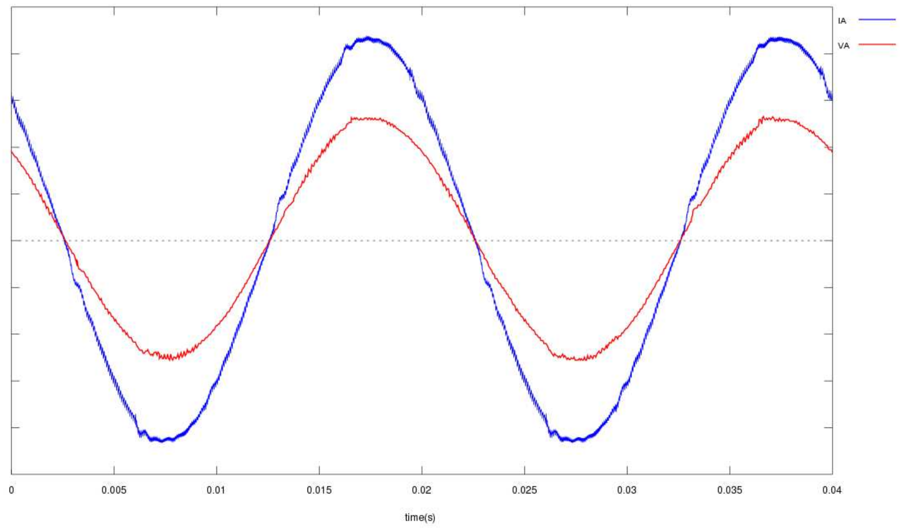


Figure 8.9: Plot of the 4-Leg Matrix Converter Input Current and Voltage for Phase **A**
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

internal state of the converter. This was originally designed to be able to check on the correct operation, but it also allows the later storage and plotting of the results. For these results one of the most interesting of these is the sampled output current data for each of the three phase legs **a**, **b** and **c**, with the neutral current being calculated as the sum of the other 3 legs. Figure 8.10 shows a plot of these currents, and it can be seen here that the current ripple which is present in the input phases, looks to also be present here as well. This ties up with a possible problem with how the converter is behaving around the interfaces between input sectors and tetrahedrons, and is a problem with the implementation in this converter rather than a problem with the theory itself.

Also of interest is the plot of the duty cycles shown in Figure 8.11. These are a snapshot of the duty cycles from the above testing, and as can be seen, they overall, follow the general shape of the duty cycles seen in both the Matlab(Figure 6.15a) and Saber(Figure 6.26) simulations. The slight differences in shape are entirely due to the relative phasing between the input frequency and the output frequency.

And finally, before moving on to look at the behaviour of the converter under an unbalanced load condition, is the plot shown in Figure 8.12. This shows the time it takes for the DSP to perform all the calculations required to determine the switching vectors and times to control the matrix converter, T_i, and plots this value alongside the total time available

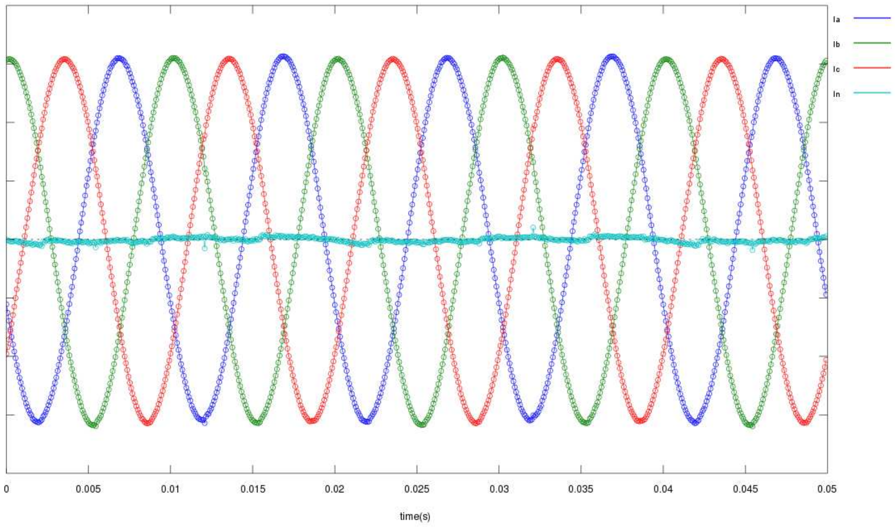


Figure 8.10: Plot of the 4-Leg Matrix Converter Output Leg Currents for Balanced Operation
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

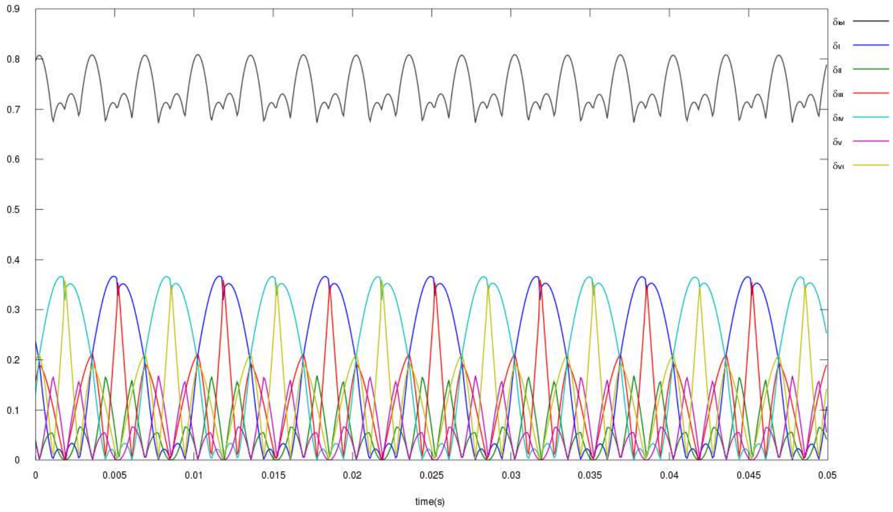


Figure 8.11: Plot of the 4-Leg Matrix Converter Input Current and Voltage for Phase **A**
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

within the SVM switching period, T_s. As can be seen, the interrupt routine takes approximately 60μs to perform it's calculations, leaving around 20μs left over in which to perform all its other duties. In this case that time was taken up with communicating with the host PC and performing any commands, however in a real application this time would also need to be used to implement whatever control strategy was being used for the overall system.

When comparing this to the other types of modulation used for 4-leg matrix converters as discussed in Section 5.11, the 60μs taken by this method does appear to be slow, especially since the Alesina/Venturini method can be used in a converter with a 25kHz switching frequency [30], which has a period of 40μs, and it is believed that using the 'Fictional DC Link' [] would be able to operate even quicker than this. However, the DSP code for this project was written to be able to operate with a switching frequency of 12.5kHz, and as such some of the techniques used may not be the fastest, and with the more extensive use of lookup tables it would be possible to bring down the DSP execution time to at least work at 25kHz.

One thing that should be noted here is that for this method, the DSP is actually calculating two sets of switching vectors per switching period, as it handles each half of the switching cycle separately, and this might also explain why the execution time is longer and this should also be taken into account.

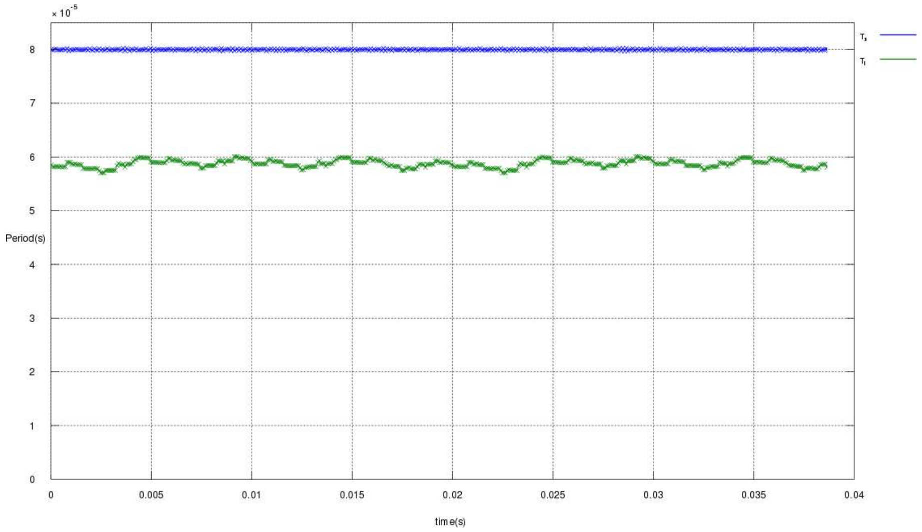


Figure 8.12: Plot showing T_i, the time taken to process the DSP interrupt routine, alongside the SVM Period, T_s

8.3 Unbalanced Load

The next set of testing featured the converter running an unbalanced load, in this case one of the branches of the star connected resistor network was doubled in size to 20Ω, and a number of the tests was run again.

The first investigation in this instance was to look at the input current, as it was expected that this would be where the imbalance in the load would have the greatest effect. Figure 8.13 shows this quite clearly, and although it is broadly sinusoidal, the distortion is very evident. Looking at the spectra of this input current in Figure 8.14 it clearly shows that there is now a significant amount of low frequency distortion present, and this is shown more clearly in the close-up of that region shown in Figure 8.15. Here it can be seen that there are two significant harmonics, one at 150Hz, and one at 250Hz, and these precisely match those found during the Saber simulations.

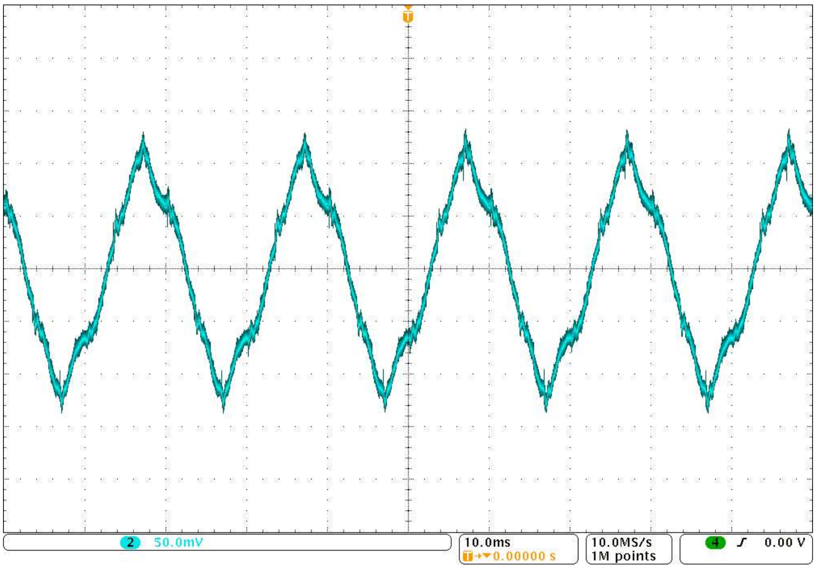


Figure 8.13: Plot of the Input Current of Phase **A** with an Unbalanced Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

Moving to the other end of the converter now, to look at the output voltages at the converter terminals. As expected, and as verified in the Saber simulations, there if little difference between the voltage being produced with the unbalanced load at this point and that for the balanced load, and this is confirmed by the FFT of the output voltage on leg **c** which is shown in Figure 8.16.

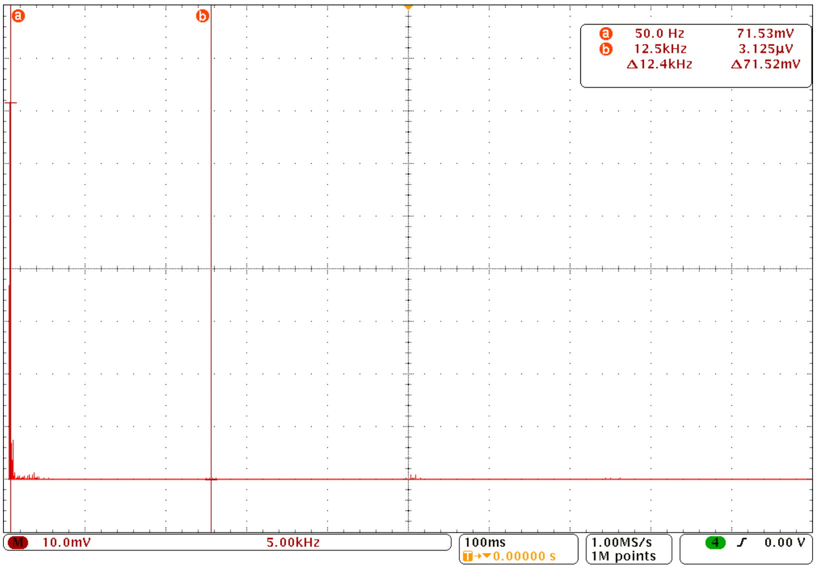


Figure 8.14: Plot of the FFT of the Input Current of Phase **A** with an Unbalanced Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

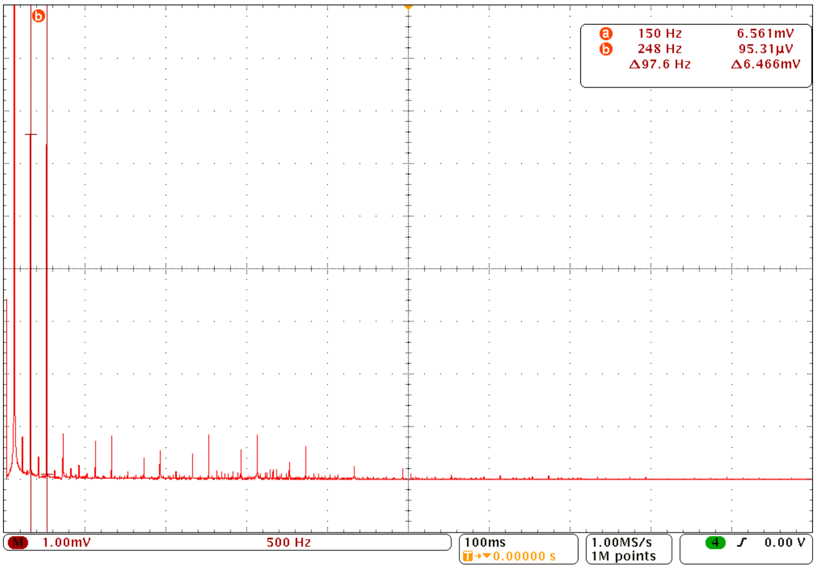


Figure 8.15: Close-up of the FFT of the Input Current of Phase **A** with an Unbalanced Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

Looking now at the voltage that is produced across the load, and as with the Saber simulations, the voltage waveforms were seen to be sinusoidal, which is shown in Figure 8.17 which shows the FFT of the output voltage for leg **c** across the load resistor. Looking more closely at the lower frequency range in Figure 8.18 once again shows there is very

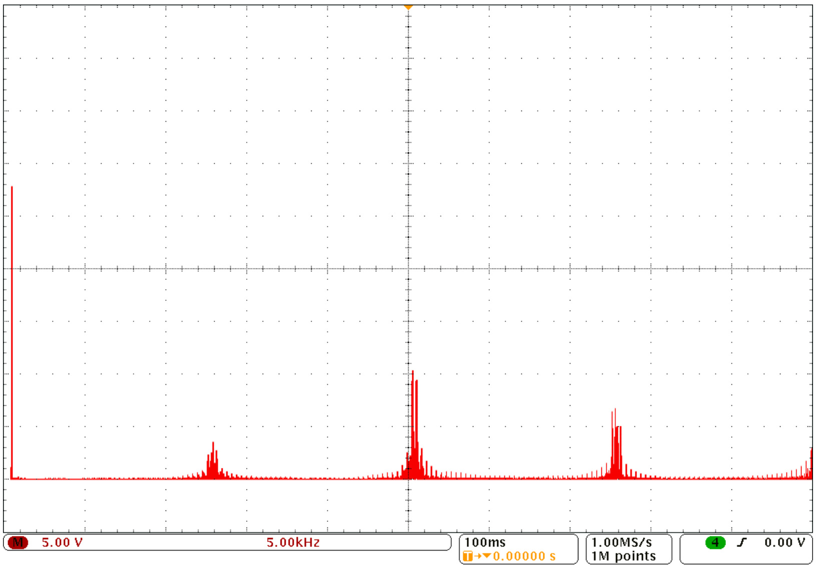


Figure 8.16: Plot of the FFT of the Output Phase-neutral Voltage for Leg **c** at the Converter for an Unbalanced Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

little low frequency distortion, and once again matches very closely with the results found in the Saber simulations, and shown in Fig 6.39.

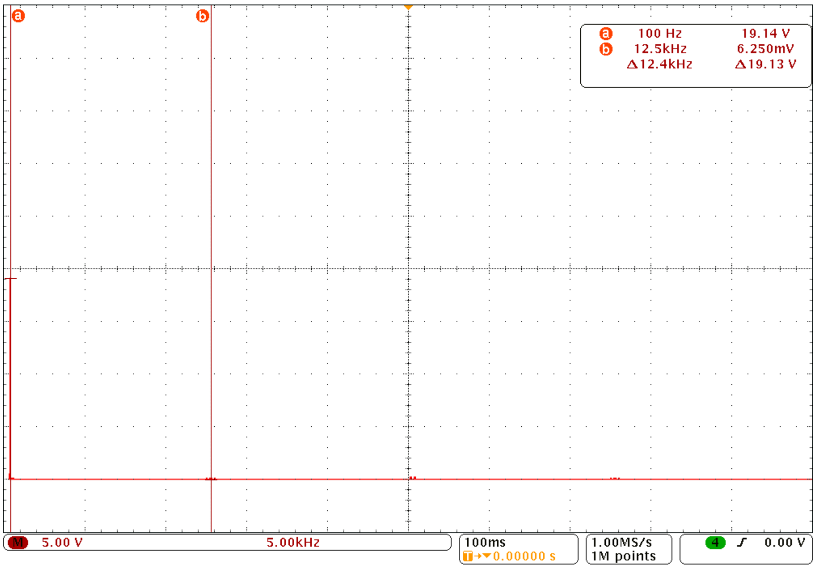


Figure 8.17: Plot of the FFT of the Output Phase-neutral Voltage for Leg **c** across the Load for an Unbalanced Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

As with the testing of the balanced load above, the DSP was used to capture snapshots of the converters operation, and by looking at the

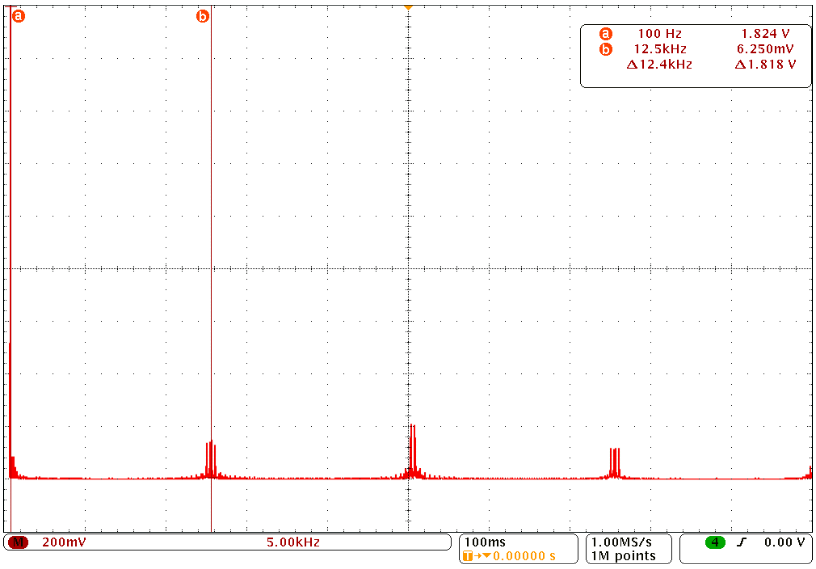


Figure 8.18: Close-up of the FFT of the Output Phase-neutral Voltage for Leg **c** across the Load for an Unbalanced Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

output leg currents being supplied to the unbalanced load, its possible to see that the currents in all three phase output legs are sinusoidal, which once more ties in nicely with the results from the FFT of the voltage across the load.

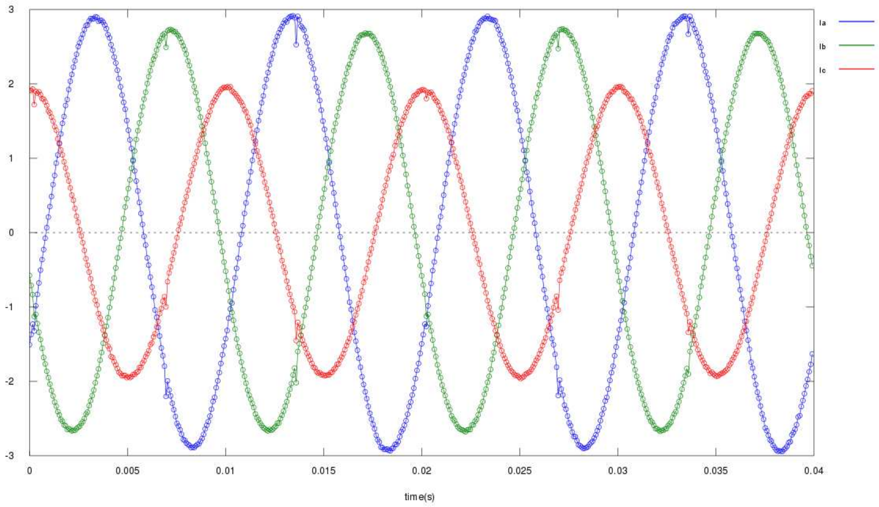


Figure 8.19: Plot of the 4-Leg Matrix Converter Output Leg Currents for Unbalanced Load
$50V_{rms}$ Demand Voltage - 100Hz Demand Frequency - 12.5kHz Sampling Frequency

Unfortunately, at this point the converter suffered another failure within the diode bridge, which prematurely ended the testing, however with the results gathered up to this point, it is considered that there is enough evidence to state the converter is operating as expected, the areas in which it does suffer from issues are entirely due to problems with the implementation, either in the software, or in a problem with the diode bridge.

8.4 Summary

While the converter was working the experimental results which were achieved were very good, even though the converter was having to operate at reduced voltages levels. These results matching up with the simulated results to a high degree, however with the problems in the diode bridge this did limit the testing, and even cut it short of the ideal. However, with the results that have been achieved, taking these in conjunction with the simulation results it is reasonable to state that the converter is operating as expected. As such this proves the theory behind the operation, and shows that the equations derived for calculating the duty cycles and switching states are correct. As such meets the main objective set for this work.

Unfortunately, what these results do not prove is whether the actual implementation of this converter is correct as it was just prior to these tests that it failed once again. Having stated that, during the initial testing the DSP was put through a large number of different tests, involving a range of different waveforms, and although not documented here, as no results were taken during this development work, this did show that the DSP software was working.

CHAPTER 9

Conclusions

The work presented in this thesis has set out to demonstrate the proof of the theory behind the implementation of Space Vector Modulation on a 4-Leg Matrix converter, and it is believed that with the work presented here it has been able to do this. The aims and objectives at the start of the project were simple, to investigate the use of space vector modulation with the new 4-leg matrix converter, and then to derive it's theory and operation. This theory was then to be tested by both simulation and by the construction of a demonstration converter.

This investigation started with a look at how space vector modulation works, and has been implemented, on two related circuits, the 4-leg inverter and matrix converter, from which the 4-leg matrix converter derives. The work then went on to show a derivation of the theory behind the 4-leg matrix converter in Chapter 5. This derivation started with by looking at the input current and output voltages spaces separately, before being able to combine them in a fashion that allowed both to be independently controlled with respect to one another.

Having completed this proof, the simulation work then became the next step, with this forming a large quantity of the work involved in this paper. The initial simulation work went on to first prove itself to be an adequate method, before moving on to demonstrate that the theory worked perfectly, and the results then provided by further circuit simulation backed these results up further. The simulation work then moved onto looking at more advanced concepts using the Saber package, and with these simulations it was possible to show how the converter dealt with a number of issues around the physical implementation of the converter.

The focus of the work then turned on to building a converter to practically demonstrate these results, by expanding on an existing design of converter. While this work resulted in a converter which was capable of operation, a problem which caused a number of failures, limited the experimental results available. However it is felt that despite these problems with the experimental converter, the results that were obtained were more than capable of backing up the simulation results and showing that the theory of operation for the converter was correct. Since this work a 4-Leg Matrix converter using the derived technique has been built and proven to be operational[36]. This evidence can only be strengthened by papers[30, 36] which have been published since this initial theory was published[32].

At the start of this project there had been no research published into the operation or control of the 4-Leg Matrix converter, however interest is starting to be seen with the number of published works increasing as different aspects of the 4-Leg Matrix converter are investigated[29, 34–41]. It is therefore hoped that the work presented here can be seen as being part of the beginnings of this interest.

So, despite the problems which were encountered with the build of the matrix converter, there is sufficient evidence presented here that it is possible to state that the derivation and implementation of Space Vector Modulation in a Four-Leg Matrix Converter shown here is correct, and so the work has been a success.

APPENDIX A

Papers published as a result of this work

During this work the following paper was published:

Wheeler, P.W. and Clare, J.C. and Mason, N. Space Vector Modulation for a 4-Leg Matrix Converter Power Electronics Specialists Conference, 2005. PESC ’05. IEEE 36th

APPENDIX B

IGBT Datasheet

SK 60GM123

SEMITOP® 2

IGBT Module

SK 60GM123

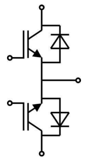
Preliminary Data

Features

- Compact design
- One screw mounting
- Heat transfer and isolation through direct copper bonding aluminium oxide ceramic (DBC)
- High short circuit capability
- Low tail current with low temperature dependence

Typical Applications*

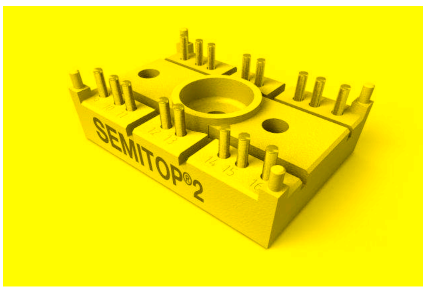
- Switching (not for linear use)
- Inverter
- Switched mode power supplies
- UPS



GM

Absolute Maximum Ratings		T_s = 25 °C, unless otherwise specified		
Symbol	**Conditions**		**Values**	**Units**
IGBT				
V_{CES}	T_j = 25 °C		1200	V
I_C	T_j = 125 °C	T_s = 25 °C	60	A
		T_s = 80 °C	40	A
I_{CRM}	I_{CRM}= 2 x I_{Cnom}		100	A
V_{GES}			± 20	V
t_{psc}	V_{CC} = 600 V; $V_{GE} \leq$ 20 V; VCES < 1200 V	T_j = 125 °C	10	µs
Inverse Diode				
I_F	T_j = 150 °C	T_s = 25 °C	60	A
		T_s = 80 °C	40	A
I_{FRM}	I_{FRM}= 2 x I_{Fnom}		100	A
Module				
$I_{t(RMS)}$				A
T_{vj}			-40 ... +150	°C
T_{stg}			-40 ... +125	°C
V_{isol}	AC, 1 min.		2500	V

Characteristics		T_s = 25 °C, unless otherwise specified				
Symbol	**Conditions**		**min.**	**typ.**	**max.**	**Units**
IGBT						
$V_{GE(th)}$	V_{GE} = V_{CE}, I_C = 2 mA		4,5	5,5	6,5	V
I_{CES}	V_{GE} = V, V_{CE} = V_{CES}	T_j = °C				mA
V_{CE0}		T_j = °C				V
r_{CE}	V_{GE} = V	T_j = °C				mΩ
$V_{CE(sat)}$	I_{Cnom} = 50 A, V_{GE} = 15 V	T_j = 25°$C_{chiplev.}$		2,5	3	V
		T_j = 125°$C_{chiplev.}$		3,1	3,7	V
C_{ies}				3,3		nF
C_{oes}	V_{CE} = 25, V_{GE} = 0 V	f = 1 MHz				nF
C_{res}						nF
$t_{d(on)}$				40		ns
t_r	R_{Gon} = 23 Ω	V_{CC} = 600V		45		ns
E_{on}		I_C= 50A		7		mJ
$t_{d(off)}$	R_{Goff} = 23 Ω	T_j = 125 °C		300		ns
t_f		V_{GE}=±15V		45		ns
E_{off}				5,2		mJ
$R_{th(j-s)}$	per IGBT				0,6	K/W



SEMITOP® 2

IGBT Module

SK 60GM123

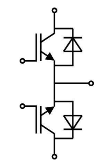
Preliminary Data

Features

- Compact design
- One screw mounting
- Heat transfer and isolation through direct copper bonding aluminium oxide ceramic (DBC)
- High short circuit capability
- Low tail current with low temperature dependence

Typical Applications*

- Switching (not for linear use)
- Inverter
- Switched mode power supplies
- UPS



GM

Characteristics

Symbol	Conditions		min.	typ.	max.	Units
Inverse Diode						
$V_F = V_{EC}$	I_{Fnom} = 50 A; V_{GE} = 0 V	T_j = 25 °$C_{chiplev.}$		2	2,5	V
		T_j = 125 °$C_{chiplev.}$		1,8		V
V_{F0}		T_j = 125 °C		1	1,2	V
r_F		T_j = 125 °C		16	22	mΩ
I_{RRM}	I_F = 30 A	T_j = 125 °C		16		A
Q_{rr}	di/dt = 400 A/µs			5,4		µC
E_{rr}	V_{CC}= 600V			2,4		mJ
$R_{th(j-s)D}$	per diode				0,7	K/W
M_s	to heat sink M1				2	Nm
w				21		g

This is an electrostatic discharge sensitive device (ESDS), international standard IEC 60747-1, Chapter IX.

* The specifications of our components may not be considered as an assurance of component characteristics. Components have to be tested for the respective application. Adjustments may be necessary. The use of SEMIKRON products in life support appliances and systems is subject to prior specification and written approval by SEMIKRON. We therefore strongly recommend prior consultation of our personal.

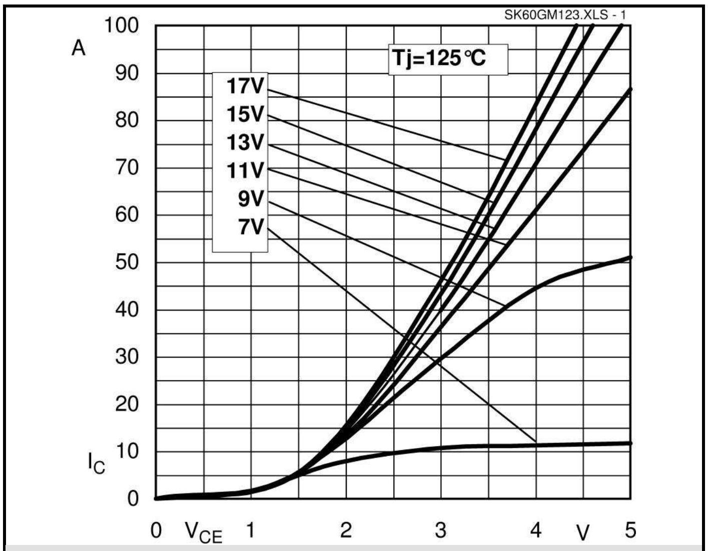


Fig. 1 Typ. output characteristic, inclusive $R_{CC'+ EE'}$

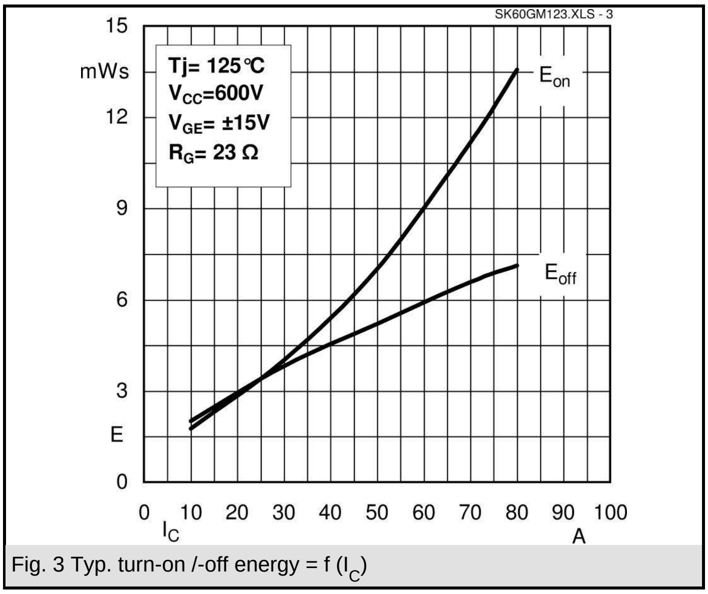


Fig. 3 Typ. turn-on /-off energy = f (I_C)

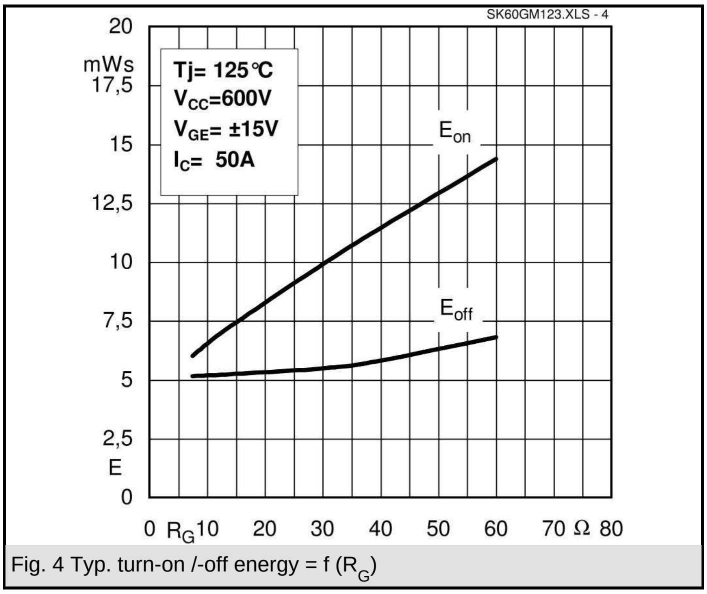


Fig. 4 Typ. turn-on /-off energy = f (R_G)

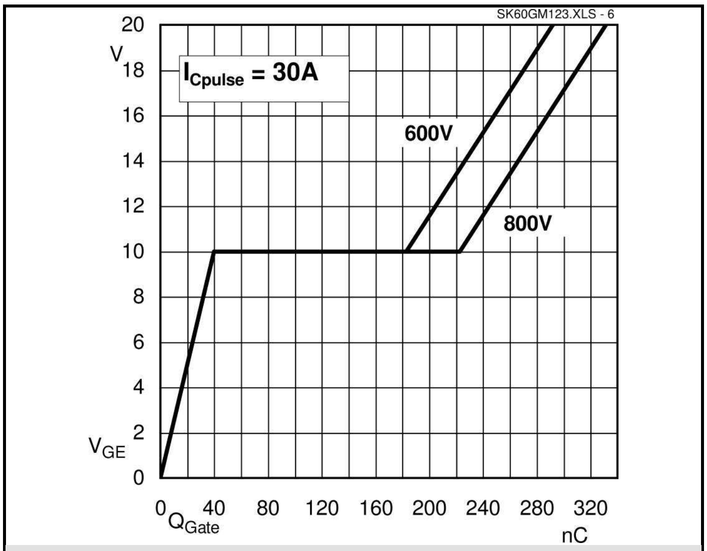


Fig. 6 Typ. gate charge characteristic

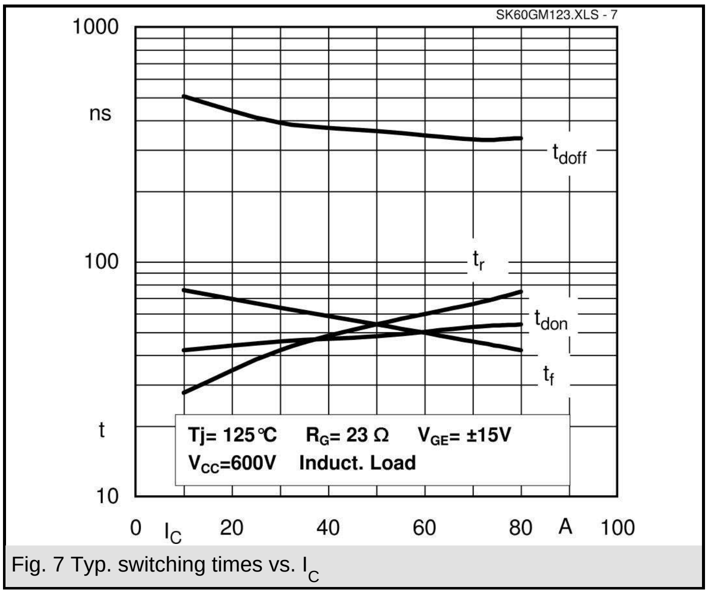


Fig. 7 Typ. switching times vs. I_C

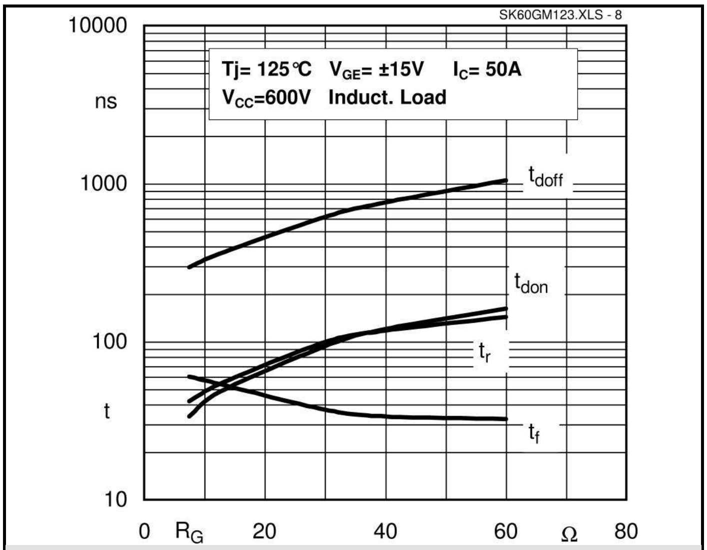


Fig. 8 Typ. switching times vs. gate resistor R_G

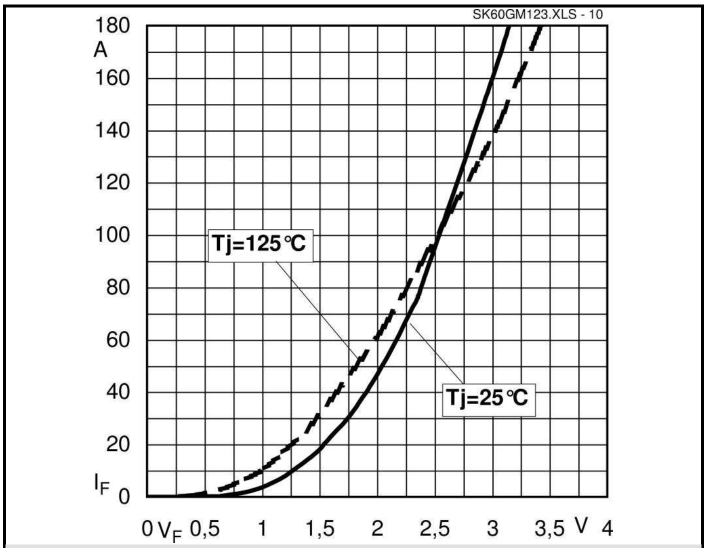
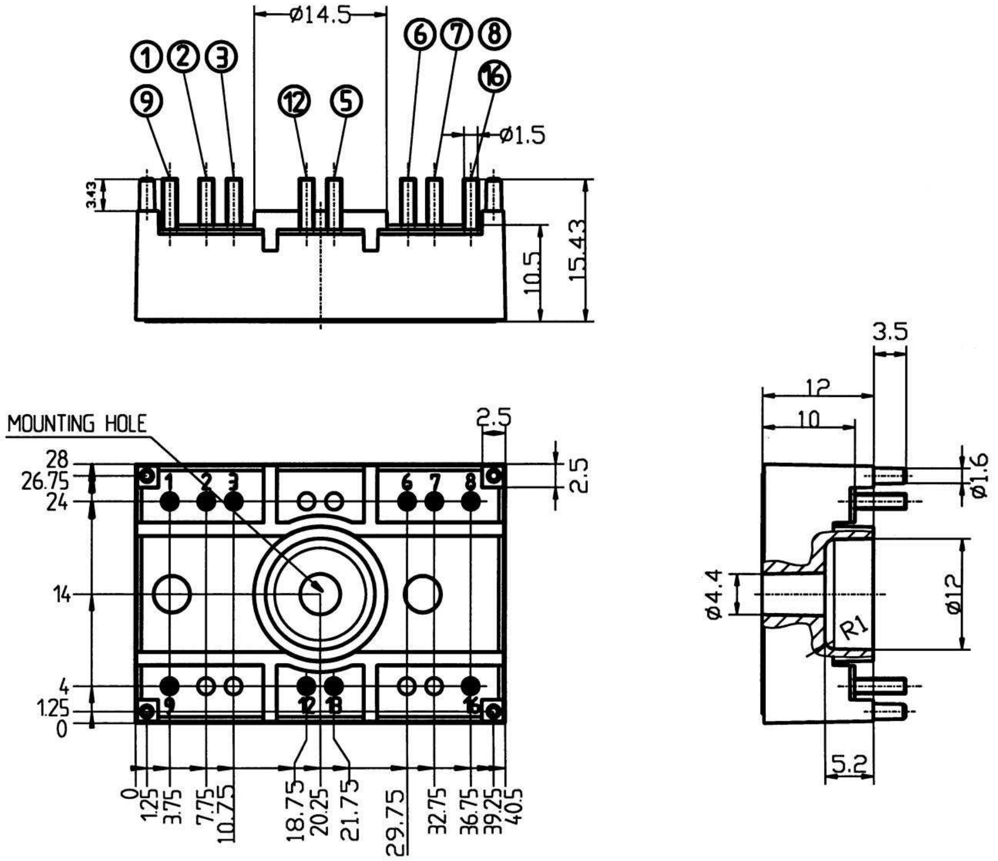


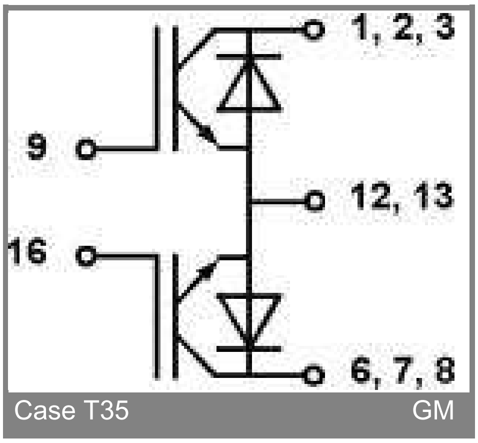
Fig. 10 CAL diode forward characteristic

UL recognized file

no. E 63 532



Case T32 (Suggested hole diameter, in the PCB, for solder pins and plastic mounting pins: 2mm)



Case T35 GM

APPENDIX C

FPGA Memory Locations within the DSP Memory Space

DPR0 0xA0000000;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**PPD**															
Write	**PPD**															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**PRS**	**PEN**		**CIP**	**ADM**				**PWR**	**0**	**RTR**	**MEN**	**SMD**	**SMC**	**SMB**	**SMA**
Write	**PRS**	**PEN**		**CIP**	**ADM**				**PWR**	**AST**			**SMD**	**SMC**	**SMB**	**SMA**

PPD	PWM_PERIOD	0xFFFF – (desired time - 1 clock cycle)
PRS	PWM_RESET	Active High
PEN	PWM_ENABLE	Active High
PFL	PWM_FIFO_LEVEL	
PWR	Power on reset	Active low - temporary
CIP	Current Direction Input Polarity:	0 = Active Low, 1 = Active High
ADM	A2D multiplex	0 = software driven 1=pwm interupt driven.
AST	A2D Converter Start	Active High
SMA	State Machine A enable	Active High
SMB	State Machine B enable	Active High
SMC	State Machine C enable	Active High
SMD	State Machine D enable	Active High
RTR	Reset trip button state	Active low
MEN	Enable button state	

DPR1 0xA0000100;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**ABY**															
Write	**PVE**															

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**UI7**	**UI6**	**UI5**	**UI4**	**UI3**	**UI2**	**UI1**	**UI0**	**CIND**	**CINC**	**CINB**	**CINA**	**PFU**	**PEM**	**PAE**	**PAF**
Write	**PVT**															

PVE	PWM_VECTOR	
PVT	PWM_VECTOR_TIME	
PAF	PWM_FIFO_Almost Full	No. of words in FIFO > 250, Active High
PAE	PWM_FIFO_Almost Empty	No. of words in FIFO < 2, Active High
PEM	PWM_FIFO_EMPTY	Active High
PFU	PWM_FIFO_FULL	Active High
CINA	Current Direction Input, PhaseA	Active High
CINB	Current Direction Input, PhaseB	Active High
CINC	Current Direction Input, PhaseC	Active High
CIND	Current Direction Input, PhaseD	Active High
ABY	A2D converters Busy	1 = Busy
UI1-7	User input 1 to 7	

DPR2 0xA0000200;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read			**T3**										**T2**			
Write			**T3**										**T2**			

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read							**T1**									
Write							**T1**									

T1	First delay timer register for 4-step current commutation
T2	Second delay timer register for 4-step current commutation
T3	Third delay timer register for 4-step current commutation

DPR3 0xA0000300;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**AD0**															
Write																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**AD1**															
Write																

AD0	A2D Data, Channel 0
AD1	A2D Data, Channel 1

DPR4 0xA0000400;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**AD3**															
Write																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**AD2**															
Write																

AD2	A2D Data, Channel 2
AD3	A2D Data, Channel 3

DPR5 0xA0000500;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**AD5**															
Write																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**AD4**															
Write																

AD4	A2D Data, Channel 4
AD5	A2D Data, Channel 5

DPR6 0xA0000600;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**AD7**															
Write																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**AD6**															
Write																

AD6	A2D Data, Channel 6
AD7	A2D Data, Channel 7

DPR7 0xA0000700;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**AD9**															
Write																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**AD8**															
Write																

AD8	A2D Data, Channel 8
AD9	A2D Data, Channel 9

DPR8 0xA0000800;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**HT23**	**HT22**	**HT21**	**HT20**	**HT19**	**HT18**	**HT17**	**HT16**	**HT15**	**HT14**	**HT13**	**HT12**	**HT11**	**HT10**	**HT9**	**HT8**
Write																

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**HT7**	**HT6**	**HT5**	**HT4**	**HT3**	**HT2**	**HT1**	**HT0**	**ST7**	**ST6**	**ST5**	**ST4**	**ST3**	**ST2**	**ST1**	**ST0**
Write									**ST7**	**ST6**	**ST5**	**ST4**	**ST3**	**ST2**	**ST1**	**ST0**

ST0 – ST7	Software Trip	Active High
HT0 – HT23	Hardware Trip	Active High

HT0	PWM FIFO empty trip
HT1	Watchdog Trip
HT2	Channel 0 trip
HT3	Channel 1 trip
HT4	Channel 2 trip
HT5	Channel 3 trip
HT6	Channel 4 trip
HT7	Channel 5 trip
HT8	Channel 6 trip
HT9	Channel 7 trip
HT10	Channel 8 trip
HT11	Channel 9a trip
HT12	Channel 9b trip
HT13	not used
HT14	External trip 2
HT15	PWM FIFO Full

DPR9 0xA0000900;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**UO3**	**UO2**	**UO1**	**UO0**											**WEN**	**0**
Write	**UO3**	**UO2**	**UO1**	**UO0**											**WEN**	**WSR**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**W_PERIOD**															
Write	**W_PERIOD**															

W_PERIOD	Watchdog Period Register	Period = 0xFFFF – W_PERIOD
WSR	Watchdog Service	Active High
WEN	Watchdog Enable	Active High
UO0-3	User Output 0-3	

DPR10 0xA0000A00;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read																
Write														**Address**		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																
Write		**Command**					**Register**		**I2C Data**							

I2C DATA for controlling the Hardware trip limit setting via I2C digital potentiometers(MAX5478)

DPR12 0xA0000C00;

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	**TE23**	**TE22**	**TE21**	**TE20**	**TE19**	**TE18**	**TE17**	**TE16**	**TE15**	**TE14**	**TE13**	**TE12**	**TE11**	**TE10**	**TE9**	**TE8**
Write	**TE23**	**TE22**	**TE21**	**TE20**	**TE19**	**TE18**	**TE17**	**TE16**	**TE15**	**TE14**	**TE13**	**TE12**	**TE11**	**TE10**	**TE9**	**TE8**

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	**TE7**	**TE6**	**TE5**	**TE4**	**TE3**	**TE2**	**TE1**	**TE0**								
Write	**TE7**	**TE6**	**TE5**	**TE4**	**TE3**	**TE2**	**TE1**	**TE0**								

TE = trip enable, number = hardware trip channel, 0 = enable, 1 = disabled

References

[1] D. Katsis, P.W. Wheeler, J.C. Clare, and P. Zanchetta. A three-phase utility power supply based on the matrix converter. In *Proc. IEEE IAS Annual Meeting '04*, Seattle, 2004.

[2] R. Zhang, D. Boroyevich, V.H. Prasad, H.-C. Mao, F.C. Lee, and S. Dubovsky. A three-phase inverter with a neutral leg with space vector modulation. In *Proc. IEEE APEC'97*, volume 2, pages 857–863, February 1997.

[3] Domenico Casadei, Giovanni Serra, and Angelo Tani. Reduction of the input current harmonic content in matrix converters under input/output unbalance. *IEEE Trans. Ind. Electron.*, 45(3):401–411, June 1998.

[4] D. Casadei, G. Serra, A. Tani, and L. Zarri. Matrix converter modulation strategies: a new general approach based on space-vector representation of the switch state. *IEEE Trans. Ind. Electron.*, 49(2):370–381, April 2002.

[5] Alberto Alesina and Marco G. B. Venturini. Solid-state conversion: A fourier analysis approach to generalized transformer synthesis. *IEEE Trans. Circuits Syst.*, 28(4):319–330, April 1981.

[6] Alberto Alesina and Marco G. B. Venturini. Analysis and design of optimum-amplitude nine-switch direct ac-ac converters. *IEEE Trans. Power Electron.*, 4(1):101–112, January 1989.

[7] L. Huber and D. Borojevic. Space vector modulator for forced commutated cycloconverters. In *Proc. IEEE IAS Annual Meeting '89*, San Diego, CA, 1989.

[8] L. Huber and D. Borojevic. Space vector modulation with unity input power factor for forced commutated cycloconverters. In *Proc. IEEE IAS Annual Meeting '91*, 1991.

[9] L. Huber, D. Borojevic, and N. Burany. Analysis, design and implementation of the space-vector modulator for forced-commutated cycloconvertors. *Electric Power Applications, IEE Proceedings B*, 139(2):103–113, March 1992.

[10] L. Huber and D. Borojevic. Space vector modulated three-phase to three-phase matrix converter with input power factor correction. *Industry Applications, IEEE Transactions on*, 31(6):1234–1246, November 1995.

[11] M.J. Ryan, R.D. Lorenz, and R. De Doncker. Modeling of multileg sine-wave inverters: a geometric approach. *IEEE Trans. Ind. Electron.*, 46(6):1183–1191, December 1999.

[12] M.J. Ryan, R.W. De Doncker, and R.D. Lorenz. Decoupled control of a four-leg inverter via a new 4 times;4 transformation matrix. *IEEE Trans. Power Electron.*, 16(5):694–701, September 2001.

[13] V.H. Prasad, D. Borojevic, and R. Zhang. Analysis and comparison of space vector modulation schemes for a four-leg voltage source inverter. In *Applied Power Electronics Conference and Exposition, 1997. APEC '97 Conference Proceedings 1997., Twelfth Annual*, volume 2, pages 864–871, February 1997.

[14] R. Zhang, V.H. Prasad, D. Boroyevich, and F.C. Lee. Three-dimensional space vector modulation for four-leg voltage-source converters. *IEEE Trans. Power Electron.*, 17(3):314–326, May 2002.

[15] R. A. Van Eck. Frequency-changer systems using the cycloconverter principle. *IEEE Trans. Appl. Ind.*, 82(66):163 âĂŞ 168, May 1963.

[16] Louis J. Lawson. The practical cycloconverter. *IEEE Trans. Ind. Gen. Appl.*, 4(2):141–144, March 1968.

[17] Muhammad H. Rashid. *Power Electronics Handbook: Devices, Circuits and Applications*. Academic Press, 2nd edition, 2006.

[18] Bin Wu, J. Pontt, J. Rodriguez, S. Bernet, and S. Kouro. Current-source converter and cycloconverter topologies for industrial medium-voltage drives. *Industrial Electronics, IEEE Transactions on*, 55(7):2786–2797, July 2008.

[19] R. Hagmann. Ac-cycloconverter drives for cold and hot rolling mill applications. In *Industry Applications Society Annual Meeting, 1991., Conference Record of the 1991 IEEE*, pages 1134–1140, September 1991.

[20] Alberto Alesina and Marco G. B. Venturini. Intrinsic amplitude limits and optimum design of 9-switches direct pwm ac-ac converters. In *Proc. IEEE PESC'88*, Kyoto, Japan, 1988.

[21] J. Oyama, T. Higuchi, E. Yamada, T. Koga, and T. Lipo. New control strategy for matrix converter. In *Proc. IEEE PESC'89*, Milwaukee, WI, 1989.

[22] L. Wei and T.A. Lipo. A novel matrix converter topology with simple commutation. In *Industry Applications Conference, 2001. Thirty-Sixth IAS Annual Meeting. Conference Record of the 2001 IEEE*, volume 3, pages 1749–1754, September 2001.

[23] C. Klumpner, F. Blaabjerg, I. Boldea, and P. Nielsen. A new modulation method for matrix converters. In *Industry Applications Conference, 2001. Thirty-Sixth IAS Annual Meeting. Conference Record of the 2001 IEEE*, volume 4, pages 2143–2150, September 2001.

[24] Zhao Yu and Yan Yangguang. A three-phase four-line soft switched inverter for ups applications. In *Industrial Electronics, 1992., Proceedings of the IEEE International Symposium on*, volume 2, pages 696–700, May 1992.

[25] G. Oriti, A.L. Julian, and T.A. Lipo. A new space vector modulation strategy for common mode voltage reduction [in pwm invertors]. In *Power Electronics Specialists Conference, 1997. PESC '97 Record., 28th Annual IEEE*, volume 2, pages 1541 –1546, June 1997.

[26] S.M. Ali and M.P. Kazmierkowski. Pwm voltage and current control of four-leg vsi. In *Industrial Electronics, 1998. Proceedings. ISIE '98. IEEE International Symposium on*, volume 1, pages 196–201, July 1998.

[27] S. Ei-Barbari and W. Hofmann. Digital control of a four leg inverter for standalone photovoltaic systems with unbalanced load. In *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE*, volume 1, pages 729–734, 2000.

[28] M.E. Fraser, C.D. Manning, and B.M. Wells. Transformerless four-wire pwm rectifier and its application in ac-dc-ac converters. *Electric Power Applications, IEE Proceedings -*, 142(6):410–416, November 1995.

[29] D. Katsis, P.W. Wheeler, J.C. Clare, L. Empringham, and M. Bland. A utility power supply based on a four-output leg matrix converter. In

Industry Applications Conference, 2005. Fourtieth IAS Annual Meeting. Conference Record of the 2005, volume 4, pages 2355 – 2359 Vol. 4, October 2005.

[30] P. W. Wheeler, P. Zanchetta, J. C. Clare, L. Empringham, M. Bland, and D. Katsis. A utility power supply based on a four-output leg matrix converter. *IEEE Trans. Ind. Appl.*, 44(1):174–186, February 2008.

[31] Jang-Hwan Kim and Seung-Ki Sul. A carrier-based pwm method for three-phase four-leg voltage source converters. *IEEE Trans. Power Electron.*, 19(1):66–75, January 2004.

[32] Pw. Wheeler, J.C. Clare, and N. Mason. Space vector modulation for a 4-leg matrix converter. In *Power Electronics Specialists Conference, 2005. PESC '05. IEEE 36th*, pages 31–38, June 2005.

[33] L. Empringham, P.W. Wheeler, and J.C. Clare. Intelligent commutation of matrix converter bi-directional switch cells using novel gate drive techniques. In *Power Electronics Specialists Conference, 1998. PESC 98 Record. 29th Annual IEEE*, volume 1, pages 707–713, 1998.

[34] Fan Yue, P.W. Wheeler, and J.C. Clare. A novel four-leg matrix converter. In *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, pages 2694 –2699, November 2006.

[35] Fan Yue, P.W. Wheeler, N. Mason, L. Empringham, and J.C. Clare. A new control method of single-stage 4-leg matrix converter. In *Power Electronics and Applications, 2007 European Conference on*, pages 1 –10, September 2007.

[36] R. CaÌĄ andrdenas, R. PeÃś anda, P. Wheeler, and J. Clare. Experimental validation of a space-vector-modulation algorithm for four-leg matrix converters. *Industrial Electronics, IEEE Transactions on*, 58(4):1282–1293, April 2011.

[37] Sangshin Kwak, Taehyung Kim, and O. Vodyakho. Four-leg based matrix converter with fault resilient structures and controls for electric vehicle and propulsion systems. In *Vehicle Power and Propulsion Conference, 2007. VPPC 2007. IEEE*, pages 519 –523, September 2007.

[38] Yao Sun, Mei Su, Lixun Xia, and Weihua Gui. Randomized carrier modulation for four-leg matrix converter based on optimal markov

chain. In *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*, pages 1 –6, April 2008.

[39] Wesam M. Rohouma, Saul Lopez Arevalo, Pericle Zanchetta, and PatrickW Wheeler. Repetitive control for a four leg matrix converter. In *Power Electronics, Machines and Drives (PEMD 2010), 5th IET International Conference on*, pages 1 –6, April 2010. doi: 10.1049/cp.2010.0208.

[40] S. Khwan-on, L. de Lillo, P. Wheeler, and L. Empringham. Fault tolerant four-leg matrix converter drive topologies for aerospace applications. In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pages 2166–2171, July 2010.

[41] Y. Sun, M. Su, G. Weihua, H. Wang, and X. Li. Indirect four-leg matrix converter based on robust adaptive back-stepping control. *Industrial Electronics, IEEE Transactions on*, PP(99):1, 2010.