A THREE-PHASE AC/AC MATRIX CONVERTER SYSTEM

GEBREGERGIS, ABRAHAM GEBREHIWET

14071916

Thesis presented in partial fulfilment of the
requirements for the degree of Master of
Science in Engineering at the University of Stellenbosch.

Study leaders:

Prof H.du T. Mouton

December 2004

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work, unless otherwise stated, and has not previously, in its entirety or in part, been submitted at any university for a degree.

_____

G. A. Gebrehiwet

December, 2004

Acknowledgement

“For with God nothing will be impossible”.
I thank God Almighty, for giving me strength, direction, and determination.

I would like to thank the following:

My supervisor Prof. H. du T. Mouton for his everlasting patience, understanding, guidance, and support through the whole time, especially during my financial crisis.

My mother for her everlasting love, understanding, and encouragement during all times.

My wife for her everlasting love, devotion, and encouragement during my studies and her patience.

All my colleagues in the Power Electronics Research Group for their help and support, Daleen Kleyn for uncountable willing to assist with administrative matters, the workshop personnel, in particular William Johannes and Marius Jumat, for their support and advice.

Summary

The thesis discusses the analysis and design of a three-phase-to-three-phase direct AC-AC matrix converter. A background study of the various matrix converter topologies and their modulation strategies are presented. The associated PWM strategy of each matrix converter topology is investigated. In addition, a detailed explanation of the three safe commutation strategies is presented.

The research focuses on the design and analysis of the direct AC-AC matrix converter topology. That includes the design of the main bi-directional power converter circuit, gate drive circuit, current direction detection circuit, voltage measurement circuit and protection circuitry. Moreover, it covers the development of the direct control algorithm based on the four-step safe current commutation- and the two-step voltage commutation strategy. A "PEC33" controller board is used to implement the developed control algorithm. Furthermore, simulation results of the direct and the indirect matrix converter topologies are presented.

The results obtained from the experimental test performed on the direct AC-AC matrix converter topology are also presented. The conclusion drawn is discussed at the final stage of the report.

Opsomming

Hierdie tesis bevat die analise en ontwerp van ‘n drie-fase na drie-fase direkte WS-WS matriksomsetter. ‘n Agtergrondstudie van verskeie ander matriksomsetter topologieë word aangebied. Die ge-assosieerde PWM beheerstrategie van elke matriksomsetter topologie is ondersoek. Hierby word ‘n gedetaileerde verduideliking van drie veilige kommutasie strategieë ingesluit.

Die navorsing fokus op die ontwerp en analise van die direkte WS-WS matriksomsetter topologie. Dit sluit die volgende in: die ontwerp van die hoof bi-direksionele drywingsomsetterbaan; die hek aandryfbaan; die stroomrigting deteksiebaan; die spanningsmeetbaan en die beveiliging stroombane. Dit dek ook die ontwikkeling van die direkte beheeralgoritme wat gebasseer is op die vier-stap veilige stroomkommutasie- en die twee-stap spanningskommutasie strategie. ‘n “PEC33” beheerkaart is gebruik om hierdie beheeralgoritme te implementer. Simulasie resultate van beide die direkte sowel as die indirekte matriksomsetter topologieë word ingesluit.

Die eksperimentele resultate wat met die direkte WS-WS matriksomsetter topologie verkry is word aangebied en bespreek. Die gevolgtrekking word in die finale afdeling van die verslag bespreek.

Table of Contents

List of Figures

List of Tables

GLOSSARY

ABBREVIATIONS

AC	:	Alternating current.
ADC	:	Analog to digital converter.
CC	:	Common Collector.
CE	:	Common Emitter.
DAC	:	Digital to analog converter.
DC	:	Direct current.
DSP	:	Digital signal processor
EMI	:	Electro-magnetic interference.
FPGA	:	Field programmable gate array.
IC	:	Integrated circuit.
IGBT	:	Insulated gate bipolar transistor.
IGCT	:	Insulated collector bipolar transistor.
LC	:	Series Inductive-capacitive
LCD	:	Lead control display.
LMSE	:	Least mean square error.
MC	:	Matrix converter
MCT	:	Metal oxide semiconductor controlled thyristor
MOSFET	:	Metal oxide semiconductor field effect transistor.
PCB	:	Printed circuit board.
PEC33	:	Power electronic controller version 33
PWM	:	Pulse-width modulation.
RMS	:	Root mean square.
SV	:	Space vector.
SV PWM	:	Space vector pulse-width modulation.
SMC	:	Sparse matrix converter.
THD	:	Threshold.

USMC	:	Ultra sparse matrix converter.
VHDL	:	Very high development language.
VSI	:	Voltage source inverter.
VSR	:	Voltage source rectifier.

SYMBOLS

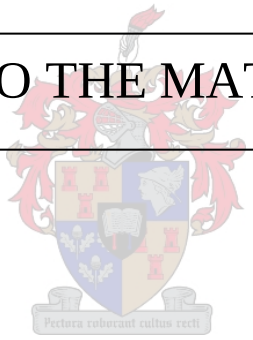
C_c	:	Compensating capacitance
C_C	:	Clamp capacitance
C_{ge}	:	Gate-to-emitter capacitance
C_{in}	:	Input capacitance
C_S	:	Stray capacitance of a resistor
d_{kj}	:	Duty cycle
E_{on}	:	Turn-on energy loss
$E_{off\,(main)}$	:	Turn-off main energy loss
$E_{off\,(tail)}$	:	Turn-off tail energy loss
f_i	:	Fundamental input frequency
f_m	:	Modulation frequency
f_o	:	Fundamental output frequency
i_A	:	Phase A output current
i_B	:	Phase B output current
i_C	:	Phase C output current
i_a	:	Phase a input current
i_b	:	Phase b input current
i_c	:	Phase c input current
I_C	:	Collector current
I_d	:	Diode current
I_F	:	Forward current to gate drive
I_{im}	:	Peak input current
$[i_i(t)]$	:	Input current matrix
$\overline{I_{in}}$	:	Input reference current vector
I_{om}	:	Peak output current
I_o	:	Output current
$[i_{oL}(t)]$	:	Output line current matrix
I_{rr}	:	Current rise

$\left[i_{op}(t)\right]$	:	Output phase current matrix
I_a	:	Current vector of alpha axis
I_b	:	Current vector of alpha axis
K_{ICC}	:	Incrementing collector current constant
l_{in}	:	Per unit inductance
L_{in}	:	Input inductance
L_o	:	Output Inductance
L_{para}	:	Parasitic inductance of the switch
L_{SR}	:	Total leakage inductance of induction motor
m	:	Modulation index
m_I	:	Current modulation index
m_O	:	Voltage modulation index
$\left[M_{phph}(t)\right]$	:	Modulation matrix
P_{cond}	:	Conduction power loss
P_E	:	Emitter power loss of gate drive
$P_{\min}$	:	Minimum power
P_O	:	Total power loss of gate drive
P_{total}	:	Total power loss of the converter
Q_L	:	Total energy stored in a motor inductance
q_{in}	:	Input reference angle
q_o	:	Output reference angle
r_{CE}	:	On-state IGBT resistance slope
r_d	:	On-state diode resistance slope
R_g	:	Gate drive output resistance
R_L	:	Load resistance
R_{qJA}	:	Junction-to-ambient thermal resistance
R_{qCS}	:	Case-to-heat sink thermal resistance
R_{qHA}	:	Heat sink-to-ambient thermal resistance
R_{qJC}	:	Junction-to-case thermal resistance
S_{kjf}	:	Forward switch
S_{kjr}	:	Reverse switch
S_n	:	Switch connected negative rail
S_p	:	Switch connected positive rail
T_A	:	Gate drive temperature
$T_{a\max}$	:	Maximum ambient temperature

$T_{j\max}$	:	Maximum junction temperature
t_{off}	:	Turn-off time of the switch
t_{on}	:	Turn-on time of the switch
t_r	:	Turn-on rise time
t_{rr}	:	Turn-on rise time
T_S	:	Switching period
U_n	:	rated voltage
$\overline{U_s}$	:	Stator voltage
V_A	:	Phase A output voltage
V_B	:	Phase B output voltage
V_C	:	Phase C output voltage
V_a	:	Phase a input voltage
V_b	:	Phase b input voltage
V_c	:	Phase c input voltage
V_{CE}	:	Collector-to-emitter voltage
V_d	:	Diode Voltage drop
V_F	:	Forward voltage to gate drive
V_{im}	:	Peak input voltage
$V_{\max}$	:	Maximum voltage
$V_{\min}$	:	Minimum voltage
V_{om}	:	Peak output voltage
V_o	:	Output voltage
$\overline{V_{oL}}$	:	Output reference voltage vector
$[V_{oL}(t)]$	:	Output line voltage matrix
$\left[V_{op}(t)\right]$	:	Output phase voltage matrix
V_{T0}	:	Diode voltage drop at low current
V_{plat}	:	Voltage drop due to parasitic inductance of IGBT
V_a	:	Voltage vector of alpha axis
V_b	:	Voltage vector of beta axis
w_m	:	Modulation angular frequency
w_o	:	Fundamental output angular frequency
$\overline{y}$	:	Stator flux
t	:	Time constant

CHAPTER 1

INTRODUCTION TO THE MATRIX CONVERTER



1 INTRODUCTION TO THE MATRIX CONVERTER

1.1 Introduction

The study of the matrix converter has been going on for the last 25 years. The progress in the development of power device (silicon) technology and large power integrated circuits encouraged the interest of research to explore an AC-AC matrix converter as an elegant silicon-intensive and efficient way to convert electric power for the following: AC motor drives, uninterruptible power supplies, variable frequency generators, and reactive energy controls. However, the power converter is still not utilized in industry because of the difficulties involved in the practical implementation related to bi-directional switch realization, zero current commutation problems, the complexity of the PWM control method, the synchronization and the protection problems.

It is hoped that the AC-AC matrix converter topology will replace the work of standard AC-DC-AC converters since standard converters are bulky and costly. This converter topology will play a large role in the application of an industrial AC drives. This topology can for instance be used in the following areas: in an industrial AC motor drives, in a marine application, in a military application especially for military vehicles, in an aerospace application.

The control system of the converter used in this thesis is a PEC33 controller board that comprises of an FPGA devices and a DSP device to generate the PWM modulation of the converter. The PWM modulation is proposed in [1-3], [5-7], and [10-12], for a 3∅/3∅ direct matrix converter. Some of the desirable features of the converter are summarized as follows:

1. The converter consists of 9 bi-directional switches.
2. It Generates 3∅ multilevel voltages and a range of frequency.
3. Sinusoidal input currents and output currents are obtained.

4. The converter can able to control the input displacement factor (unity power factor).
5. It is Capable of regenerating energy back to the main supply.
6. There is no reactive component for storage purposes.
7. All-silicon converter, small size, and low cost.
8. Available voltage gain 0.5, 0.866, and 1.15 [1-3], [5-7], and [10-12].

1.2 Scope of the Thesis

The thesis contains four main chapters and a conclusion. The initial focus of this thesis is a discussion of the matrix converter topologies. It proceeds to reveal the construction of the converter, its control, simulation results, and experimental results obtained.

The second chapter discusses the various matrix converter topologies and the modulation methods. These topologies have some advantages and disadvantages over each other. The first topology is known as the conventional inverter based converter, which has intermediate reactive energy storage elements that act as a DC-bus link. The size of this topology is large because of the reactive elements needed to store the intermediate energy. The direct matrix converter is the second topology that consists of 9-bidirectional switches arranged in such a way so that any of the input line can be connected to the output line to give desired frequency and voltage. This topology is small in size, all-silicon converter, provides sinusoidal input and output current waveform, bidirectional energy flow and controllable input power factor regardless of the size or type of the load. All the above advantages are on the cost of complex control system.

A matrix converter topology that has the same approach as the conventional inverter based converter without including any reactive energy storage as an intermediate DC-bus link is discussed. This topology is known as the Indirect SV PWM matrix converter. The indirect SVPWM matrix converter is divided into two portions, a rectifier side, and an inverter side. The rectifier side of the converter is directly connected to the input line

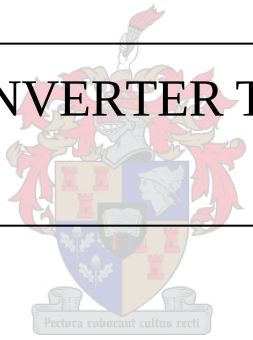
side and converts the three-phase input into DC voltage. This DC voltage is supplied in to the inversion side of the converter, which produces the desired frequency range and voltage level for the load. A brief description of a sparse matrix converter, ultra sparse matrix converter and a new matrix converter is also investigated.

Construction of the direct matrix converter topology will be the focus of chapter three. This chapter will discuss the design and analysis of the main power converter, the gate-drive circuit, the current direction detection circuit, the protection circuit, the fiber optic circuit, the analog isolation circuit, and the voltage measurement circuit. Besides, the power losses of the matrix converter and the heat sink selection are discussed in this chapter.

The development of the control algorithms of the direct matrix converter are presented in chapter four. The control algorithms are based on current commutation and voltage commutation. Chapter five describes a practical system whereby the matrix converter can be analogously simulated and the different controlling algorithms are tested. The results of the simulation and practical tests done on the converter are displayed in the same Chapter. Different PWM modulation mechanisms are tested and the results are discussed. Finally, the conclusion drawn is expressed in Chapter six. Future work and recommendations are also provided in this chapter.

CHAPTER 2

MATRIX CONVERTER TOPOLOGIES



2 MATRIX CONVERTER TOPOLOGIES

2.1 Introduction

A matrix converter system based on high frequency synthesis control was introduced in 1980 [1-2], [5]. Since then, the converter became the center of attention for intensive research, and a number of studies have been done by different groups of researchers. The studies that have been done yet mostly concentrates on the implementation of the matrix converter switches, known as a four-quadrant switch or bidirectional switch [3], [8], and the different switching control strategy of the matrix converter topology [1-3], [5-7], [10-11].Some of the topologies and their switching control strategies are presented in this chapter.

These matrix converter topologies use a bi-directional switch as the basic building block of the converter, and apply the same modulation strategy approach based on a high frequency waveform synthesis technique. Therefore, it is important to investigate the configuration of the bi-directional switch [3], [8] since there is no single module bi-directional switch available in the market. The configuration of the bi-directional switch and the high frequency waveform synthesis technique [5] are discussed before the matrix converter topologies.

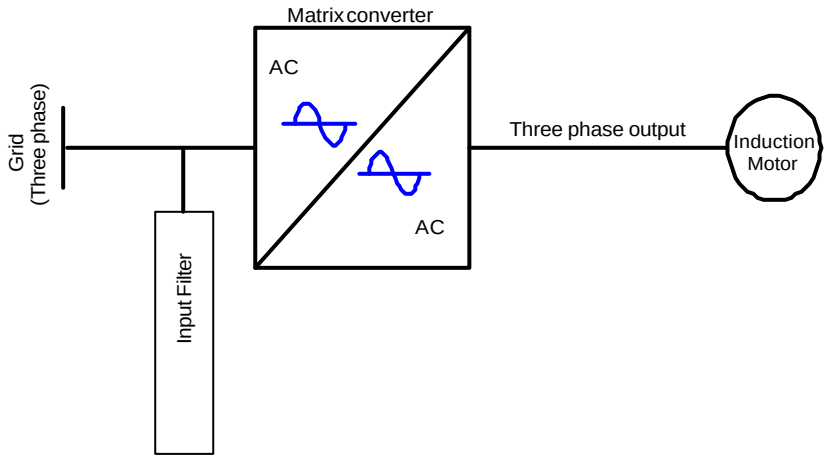


Figure 2-1 A three-phase AC-AC matrix converter block diagram

A block diagram of a three-phase AC-AC matrix converter is shown in Figure 2-1, where the input lines are connected to a three-phase grid and a small series LC filter to eliminate input ripples.

2.2 Step and High Frequency Synthesis

High Power converters consist of active elements (switches) and reactive elements. Because of the switches, the waveform synthesized is discontinuous. However, large sized and high cost reactive elements are used to smooth the discontinuity generated. To avoid the use of expensive reactive elements and yet generate a smooth continuous output waveform a new waveform synthesis techniques are proposed [5]. These waveform synthesis techniques are a step synthesis and a high frequency synthesis that produce a sinusoidal output waveform. A brief discussion of the two techniques of generating continuous waveform follows.

2.2.1 Step Synthesis Analysis

This type of waveform synthesis employs more than two switches that connect the single output line to different input lines, such as tapped transformer, poly phase AC input, and tapped supply, in an appropriate sequence as shown in Figure 2-2(a) to generate the desired continuous output waveform. The typical output waveform is shown in Figure 2-2(b).

(a) (b)

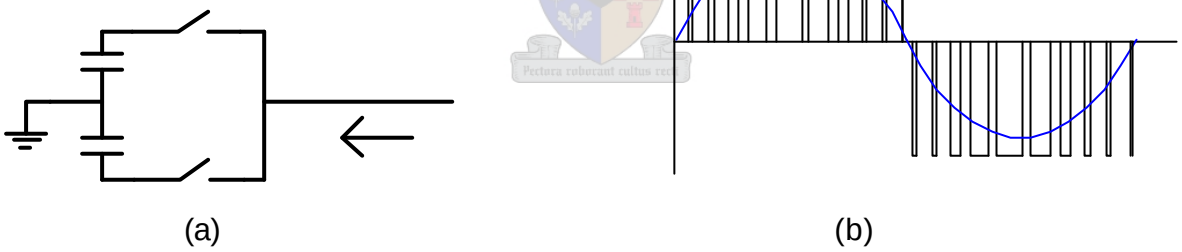
Figure 2-2 A wave form synthesis technique.

a) Multi-switch DC/AC step synthesis converter b) Typical output waveform

2.2.2 High Frequency Synthesis Analysis

High frequency synthesis uses a minimum possible number of switches to connect the input lines with the output line. The switches commutate at a higher frequency than the input and desired output frequency, so that each of the output cycles will consist of enough input "chops" to generate the desired smooth sinusoidal output waveform. The switching frequency is chosen high enough to ensure a wide spectrum of separation between the "unwanted", high frequency output components that are confined around the switching frequency, and the "wanted", low frequency output component. An LC filter is connected at the output line to filter out the unwanted high frequency components of the output.

A simple two-switch DC-AC converter, which uses the advantage of high frequency synthesis, is shown in shown Figure 2-3(a). The chopped input voltage and the filtered low frequency output voltage are also shown together in Figure 2-3(b).



(a) (b)

Figure 2-3 High frequency synthesis technique (a) Two-switch DC-AC step synthesis converter, (b) Typical output waveform of the Converter.

2.3 Bi-directional Switch Construction (Configuration)

The matrix converter topologies require a bi-directional switch capable of blocking voltages of both polarity and capable of conducting current in both directions [3], [8] and [9]. Unfortunately, there is no such switching device commercially available as a single

module to fulfill the need. Hence, discrete devices are configured in such a way to construct a bi-directional switch cell. There are two possible configuration ways of building the bi-directional switch; one consisting of a transistor embedded in a diode bridge arrangement while the alternative configuration consists of two anti-paralleled transistors and anti-parallel diodes. The later configuration can be implemented as either a common emitter (CE) or a common collector (CC). The discrete switching device available for building the bi-directional switch could be IGBT, or other devices like MCTs, IGCTs, and MOSFETs.

2.3.1 Diode Bridge Arrangement

The diode-embedded switch arrangement consists of a single transistor at the center of a single-phase diode bridge arrangement as shown in Figure 2-4. Positive current and negative current are carried by the same switching device, which decreases the required gate driver to one per commutation cell as well as the power supply to the gate drive. The disadvantage of the arrangement is; the relatively high devices loss since there are three devices in each current conduction path, and it is impossible to control the direction of current through the switch device.

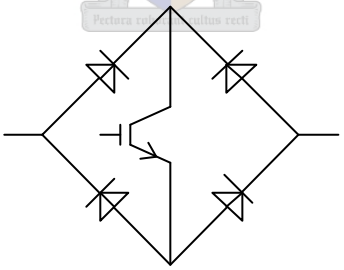


Figure 2-4 Diode bridge arrangement bidirectional switch

2.3.2 Common Emitter Arrangement

This type of bi-directional switch arrangement consists of two IGBTs and two diodes connected in anti-parallel form as shown in Figure 2-5. The arrangement is capable of blocking both voltage polarities and conducting current in both directions. The advantage of this arrangement over the previous diode bridge arrangement is:

- Conduction losses are reduced since there are only two devices that carry the current at any one time.
- There is independent control of the current direction through each switching device.

The disadvantage of the arrangement is the need of two isolated power supplies for the two gate drivers of the IGBT's.

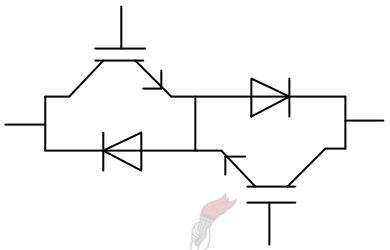


Figure 2-5 Common Emitter arrangement bidirectional switch

2.3.3 Common Collector Arrangement

This arrangement is similar to the previous one, except the IGBT's are arranged in a common collector configuration as shown in Figure 2-6. The conduction losses are the same as the common emitter arrangement, since only two devices are involved in the conduction path of the current.

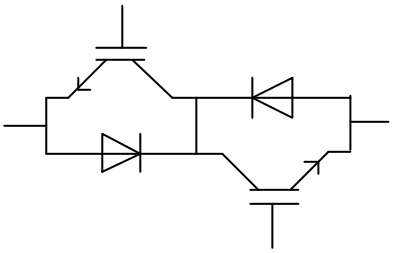


Figure 2-6 Common Collector arrangement bidirectional switch

2.4 Conventional (Standard) Inverter Based AC/DC/AC Converter Topology

The AC-DC-AC converter topology is the only available converter industrially, which is used in the application of driving AC motors. This topology is constituted of a rectifier stage and an inverter stage. The rectifier stage contains energy storage elements, such as a bank of capacitor, which are required to provide the constant voltage (DC-link) to the inverter stage. The DC-link of the converter is bulky and costly. The inverter stage is relatively smaller than the corresponding rectifier stage, but acquires a snubber network circuit that is no longer needed in modern converters. In general, the topology can be described as:

- Consisting of switching device and energy storage (reactive) elements that cause the converter to be large sized and costly.
- It is unable to regenerate energy back to the main supply source.

Either the space vector (SV) modulation technique or the carrier based PWM modulation technique could be used to control the switching of the converter. A block diagram of the conventional inverter based AC-DC-AC converter topology is shown in Figure 2-7.

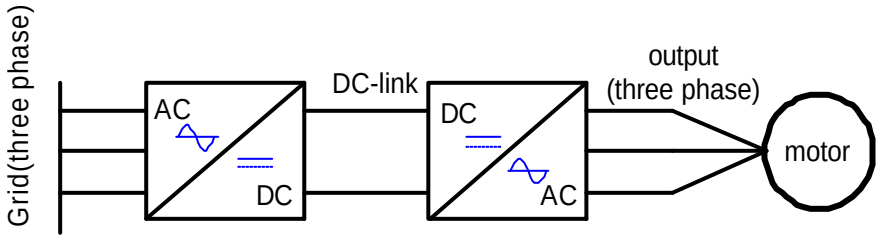


Figure 2-7 Conventional inverter based on AC-DC-AC converter block diagram

2.5 Direct Matrix Converter Topology

The direct matrix converter topology consists of n and p bi-directional switches, connecting the n-input line to the p-output line in order to provide a direct power conversion [1-3], [5]. The converter is characterized by its ability to connect any input phase to any output phase at any instant. An n-line input phase and p-line output phase direct matrix converter topology is shown in Figure 2-8. This direct matrix converter topology has the following attractive features:

- Sinusoidal input current and sinusoidal output voltage
- It employs bidirectional switches, which enables regenerating energy back to the source.
- It ables to adjust the input power factor of the converter despite the type of the load connected. Unity power factor is easily achievable.
- There is no intermediary DC-link energy storage. Since the converter is DC-link less, the size and cost of the converter is relatively reduced. In addition, the power at the input is seen at the output.

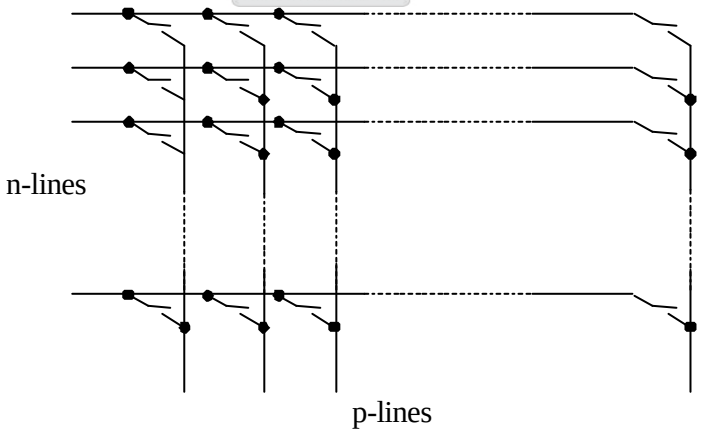


Figure 2-8 A general n-input line to p-output line Matrix Converter topology

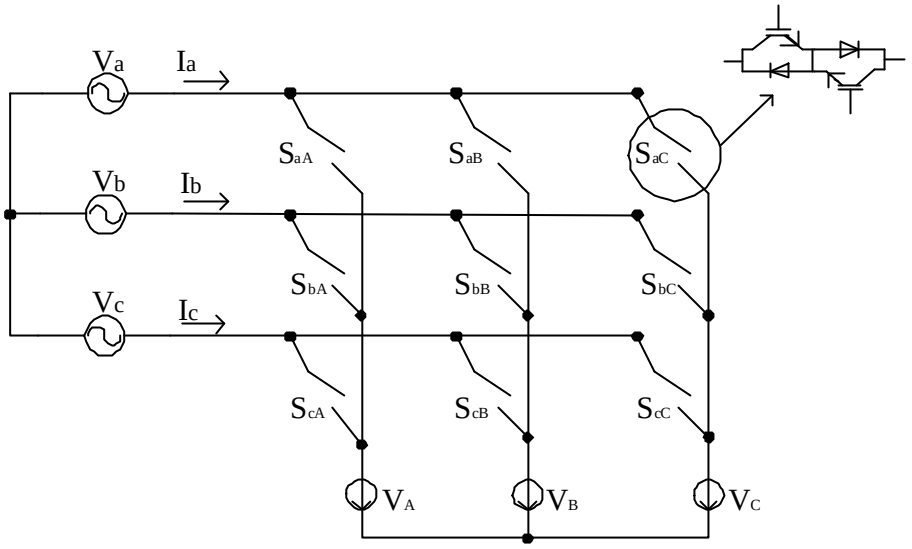


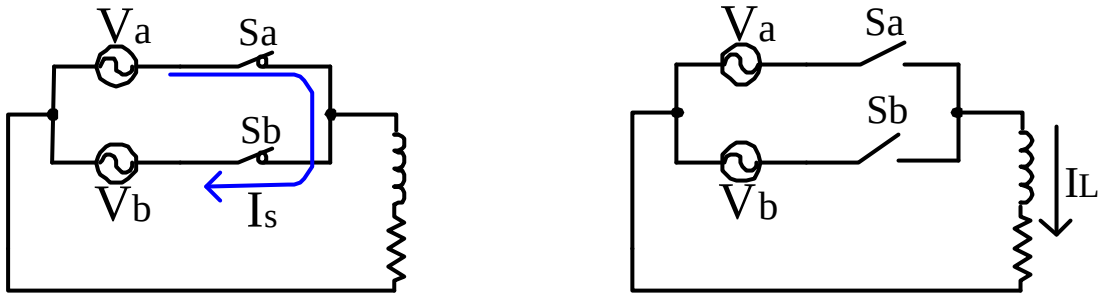
Figure 2-9 A three-phase to three-phase AC-AC matrix converter

Since almost all AC-drive operations depend on a three-phase AC supply, the study of this thesis concentrates on a three-phase-to-three-phase AC-AC direct matrix converter. The study can be extended beyond the three-phase matrix converter to higher number of phases. A three-phase-to-three-phase AC-AC matrix converter topology consists of 9 bi-directional switches arranged so that any of the line-phases are connected to any of the output phases of the converter as shown in Figure 2-9. The bi-directional switches allow the conduction of current in both directions and block voltage of both polarities. The direct matrix converter topology is not available for use in industries currently, because of the complexity of the control implementation and commutation failure.

2.5.1 Commutation Difficulties of the Direct Matrix Converter

The three-phase AC-AC direct matrix converter topology is prone to commutation failure, which could be either a short circuit or an open circuit of the switching devices. The switches cannot turn on or turn off instantaneously and simultaneously at any instant of time since all switches have a definite propagation delay and switching delay time. For safe switching, it is necessary that no two bi-directional switches of the input line voltage be turned on at same time, as shown in Figure 2-10(a). This might result in

a line-to-line short circuit (current spike) and could consequently destroy the converter switches. In addition, the bi-directional switches of each output phase should not all be turned off at any instant, as shown in Figure 2-10(b). Turning off all the output switches creates an open circuit in the converter that results in the absence of a conducting path for the inductive load current, causing a large voltage spike that could destroy the switches.



(a) (b)

Figure 2-10 Two phase to single phase converter commutation problem (a) Short circuit of the switches; (b) Open circuit of the switches

To understand the commutation problem, consider a two-phase input to single-phase output converter, as shown in Figure 2-11. Suppose that S_a is turned on, and conducts the load current I_L. Eventually switching from one phase to another will occur; from input phase voltage V_a to V_b. Hence, switch S_b will carry the current of the inductive load. Theoretically, the switching must be instantaneous and simultaneous so that the load current will have a conducting path, and there will not be a short circuit of the input phase voltage. In practice, the finite switching time and the delay time in the drive circuit as well as the switches has to be considered during switching. Therefore, if the switch S_b is turned on before switch S_a is turned off then a short circuit path is established through $V_a - S_a - S_b - V_b$. Current spikes generated this way will destroy the switches. Similarly, if switch S_a is turned off before switch S_b is turned on, then there will be a need for a natural freewheel path for the inductive load current. Hence, a voltage spike will be induced in the inductive load, which destroys the converter switches.

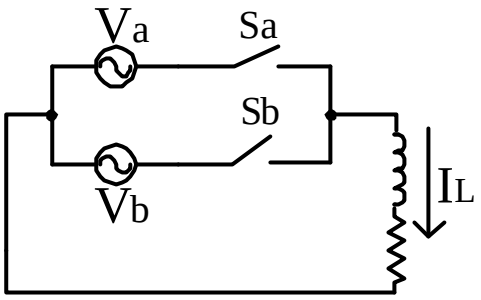


Figure 2-11 Typical two phase to single phase converter

To prevent the commutation failure caused due the practical limitation of the bi-directional switch, a number of commutation approaches have been proposed. However, solving the short circuit and open circuit simultaneously was difficult. For example, a deadtime commutation (used in VSI) allows a delay time between the incoming and outgoing switches so that there will not be an overlap (short circuit of the input phase voltage), but an open circuit will be created that requires a natural freewheeling path (like in VSI) for the inductive load current. An overlap of the switches is also not possible since a short circuit of the input line phase occurs and will destroy the switching devices. Adding large inductors on the input line side to prevent the current from rising to a dangerous level that might destroy the switching devices and applying the overlap switching solution is not desirable, since the commutation time will be increased which complicates the implementation of the switching strategy.

Safe commutation methods that solve both short circuit and open circuit problems are proposed in [3], [4], [9], [11]. These methods depend on the knowledge of a reliable load current direction and/or the relative input phase voltages magnitude.

2.5.2 Current and Voltage Commutation Methods

A reliable knowledge of the load current direction and/or relative input voltages magnitude is necessary for a safe commutation, since any discrepancy might result in a short circuit or open circuit of the switching devices. Thus, a current commutation method, which obeys the two basic rules that are no short circuit of input phase and

open circuit of the inductive load current, which uses a four-step safe semi-soft commutation strategy is proposed in [3]. The direction of the current flow through the switching cells has to be controlled at all times during the commutation, since switching from one phase to another solely depends on the knowledge of the load current direction. Another commutation method that depends on the relative input voltages magnitude with reduced commutation steps denoted as the two-step commutation is also proposed in [3], [4], [9], [11]. The last commutation method proposed is a one-step commutation [3], [4], [9], which depends on the knowledge of both the direction of load current and the relative input voltages. The switching policy solving the commutation problem described above will be introduced with a two-phase to single-phase converter connected to an inductive load as shown in Figure 2-12. The converter can control only the output voltage level, but not the output frequency. Two bi-directional switches are alternatively connecting the inductive load to the input phase voltages, according to the duty cycle determined by the control strategy.

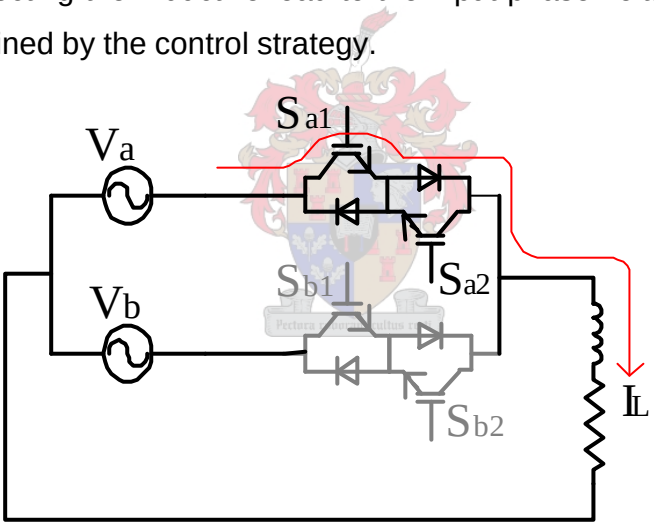
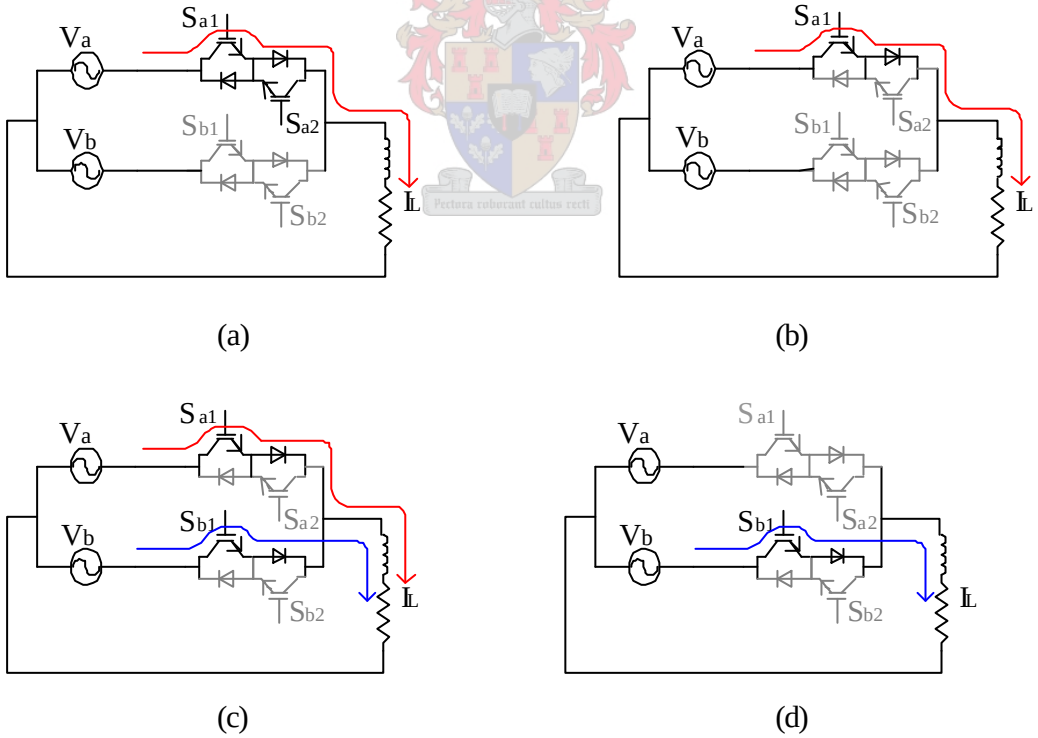


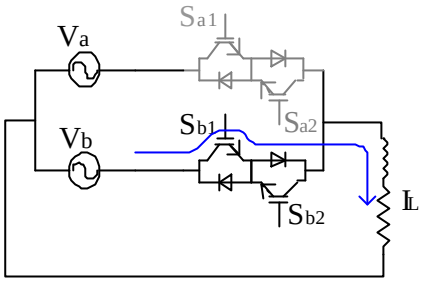
Figure 2-12 A two-phase to single inductive load converter connected via bidirectional switches.

2.5.2.1 Four-Step Semi Soft Safe Commutation Strategy

Consider a two-phase to single-phase converter as shown in Figure 2-13(a). Initially, bi-directional switch cell S_a is turned on and bi-directional switch cell S_b is turned off. Since bi-directional switch cell S_a is active, either of the two active switches S_{a1} or S_{a2}

will be conducting the load current I_L depending on the load current direction. Assume the load current is flowing in the positive direction as shown in Figure 2-13(a); the switch S_{a1} will conduct the load current. Eventually switching from V_a to V_b is required, which depends on the control strategy. The outgoing switch S_{a2} is turned off as shown in Figure 2-13(b), because this switch is not carrying the load current. After a given definite delay time t_d, the incoming switch S_{b1} is turned on which might conduct the load current depending on the magnitude of the input voltages as shown in Figure 2-13(c). If input voltage V_b is greater than input voltage V_a, then a forced (hard) commutation occurs for the switch S_{b1}, otherwise, a soft commutation is going to occur at step four of the commutation. It is safe now to turn off S_{a1} since the load current can be carried by the switch S_{b1} as shown in Figure 2-13(d). The switch S_{b2} is turned-on so that a reverse direction is provided for the load current as shown in Figure 2-13(e).





(e)

Figure 2-13 The four step semi-soft current commutation. (a) A steady state of the converter, where the load current is carried by S_{a1}. (b) Step-one, turning off the idle switch S_{a2}. (c) Step-two, turning on the incoming switch S_{b1}. (d) Step-three, turning off the conducting switch S_{a1} since to is safe. (e) Step-four, turning on switch S_{b2} so that reverse current direction is possible.

A state representation of the four-step commutation method and switching states of the bi-directional switches is shown in Figure 2-14.

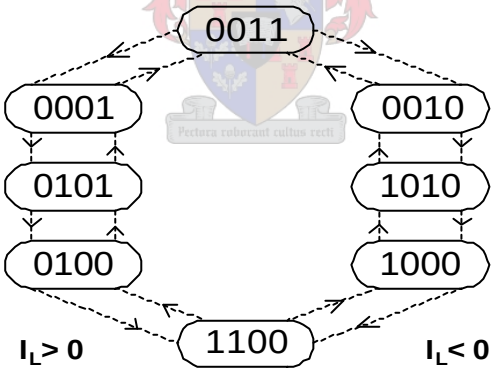


Figure 2-14 State diagram of the bidirectional switches

A timing diagram that shows the sequence of the switches state during commutating from input phase voltage V_a to input phase voltage V_b, is shown in Figure 2-15.

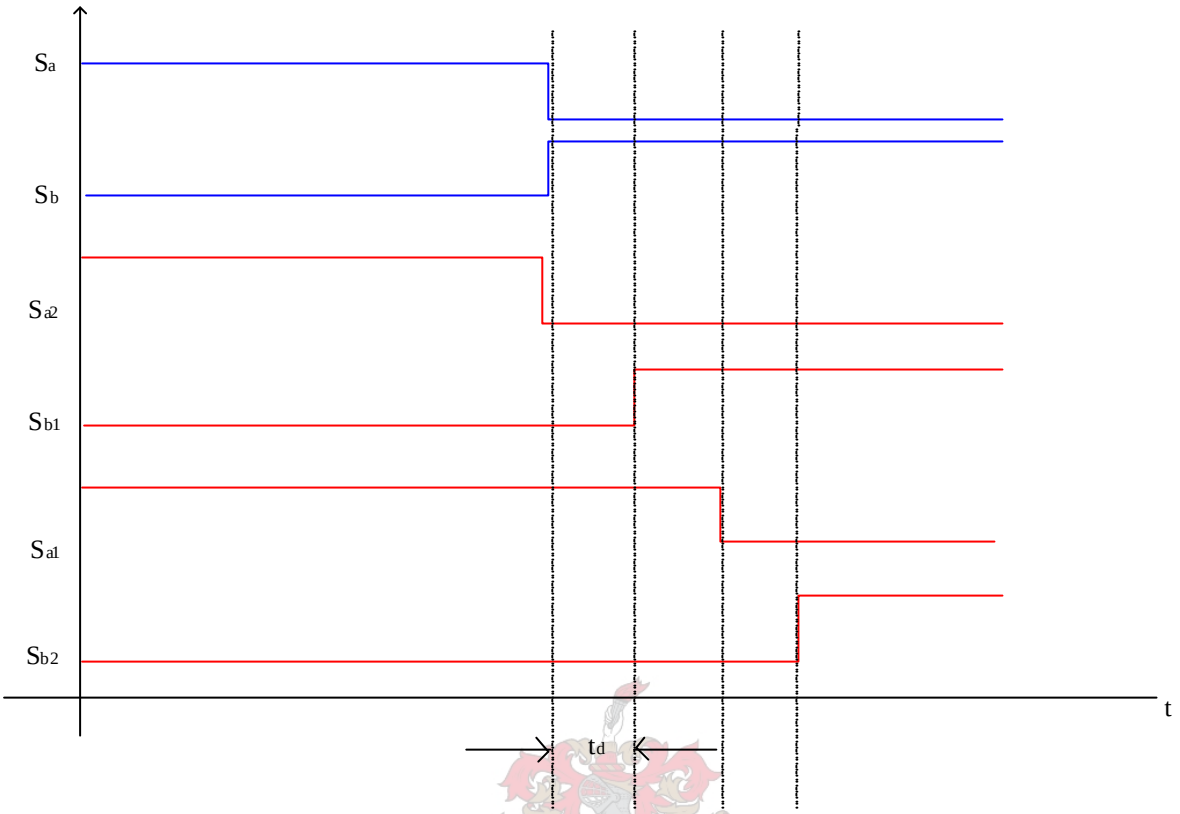


Figure 2-15 A timing sequence of the switches state of the four step semi-soft current commutation

2.5.2.2 Two-Step Semi Soft Safe Commutation Strategy

This commutation method depends on the relative input voltages magnitude of the converter for a safe commutation. Reliable information of the three-phase input voltages is required to achieve a safe switching in the converter. Consider a three-phase to single-phase matrix converter shown in Figure 2-16. Three bi-directional switches connect the three-phase input voltage sources V_a, V_b, and V_c to the inductive load. The load current flows through one of these bidirectional switches depending on its direction. A deadtime is required to avoid current spikes through the non-ideal switches and, at the same time, a current path for the inductive load has to be provided. Therefore, a two-step commutation based on the knowledge of relative input voltages magnitude is used to solve these two problems. The relative input voltage is defined as the voltage difference of the switch-to-switch V_{ab}, V_{bc} and V_{ca} across the bidirectional switches.

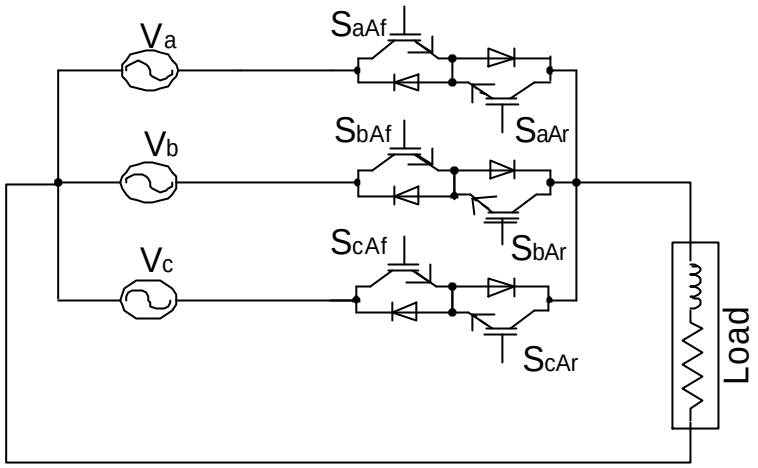


Figure 2-16 A three-phase to single phase matrix converter

Consider only one cycle of the input voltages to explain how this safe switching policy can be applied to the converter as shown in Figure 2-16. Six intervals are identified based on the highest input voltage and lowest input voltage as shown in Figure 2-17. For example, V_a is the highest input voltage and V_c is the lowest input voltage in interval I, and so forth. Let's define $V_{\max}$ as the highest input phase voltage and $V_{\min}$ as the lowest input phase voltage among the input voltages V_a, V_b, and V_c.

$$V_{\max} = \max\{V_a, V_b, V_c\} \quad \textbf{Equation 2-1}$$

$$V_{\min} = \min\{V_a, V_b, V_c\} \quad \textbf{Equation 2-2}$$

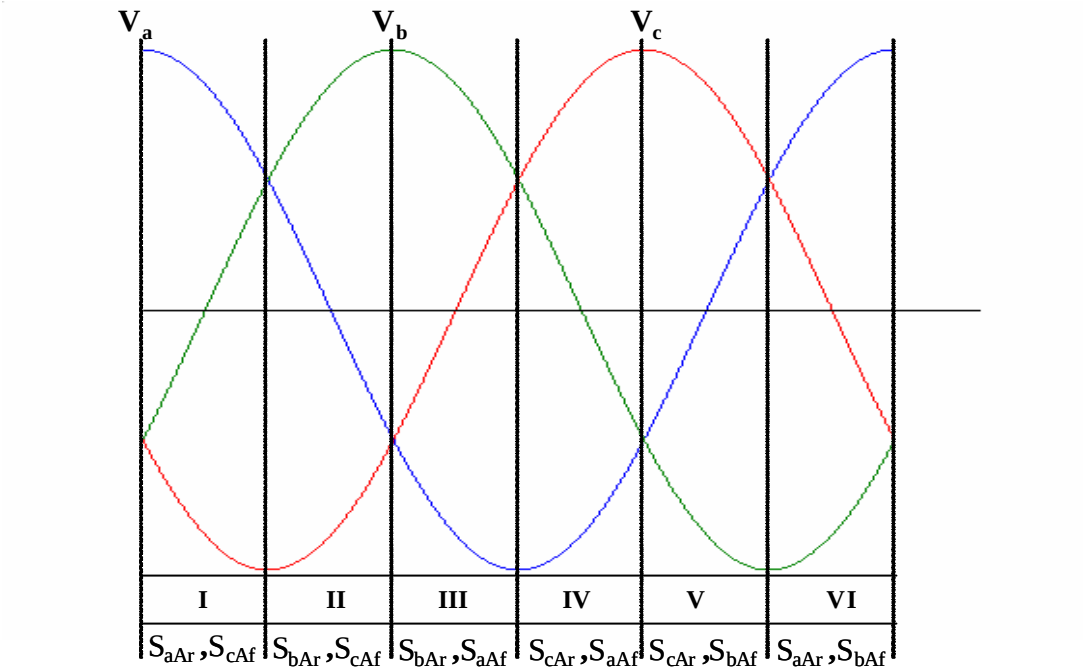


Figure 2-17 Six intervals of the input voltages and the switching states for safe commutation

For a safe switching, the reverse switch corresponding to the highest voltage $V_{\max}$, and the forward switch corresponding to the lowest voltage $V_{\min}$ are turned on to maintain a path for the output current to flow. For example, reverse switch S_{aAr} and forward switch S_{cAf} are turned on and all the rest are turned off in interval I. Reverse switch S_{bAr} and forward switch S_{cAf} are turned on in interval II while all the rest are turned off, and so forth. These six switching intervals and the switching state of the bi-directional switches of the two-step safe commutation method are shown in Figure 2-18.

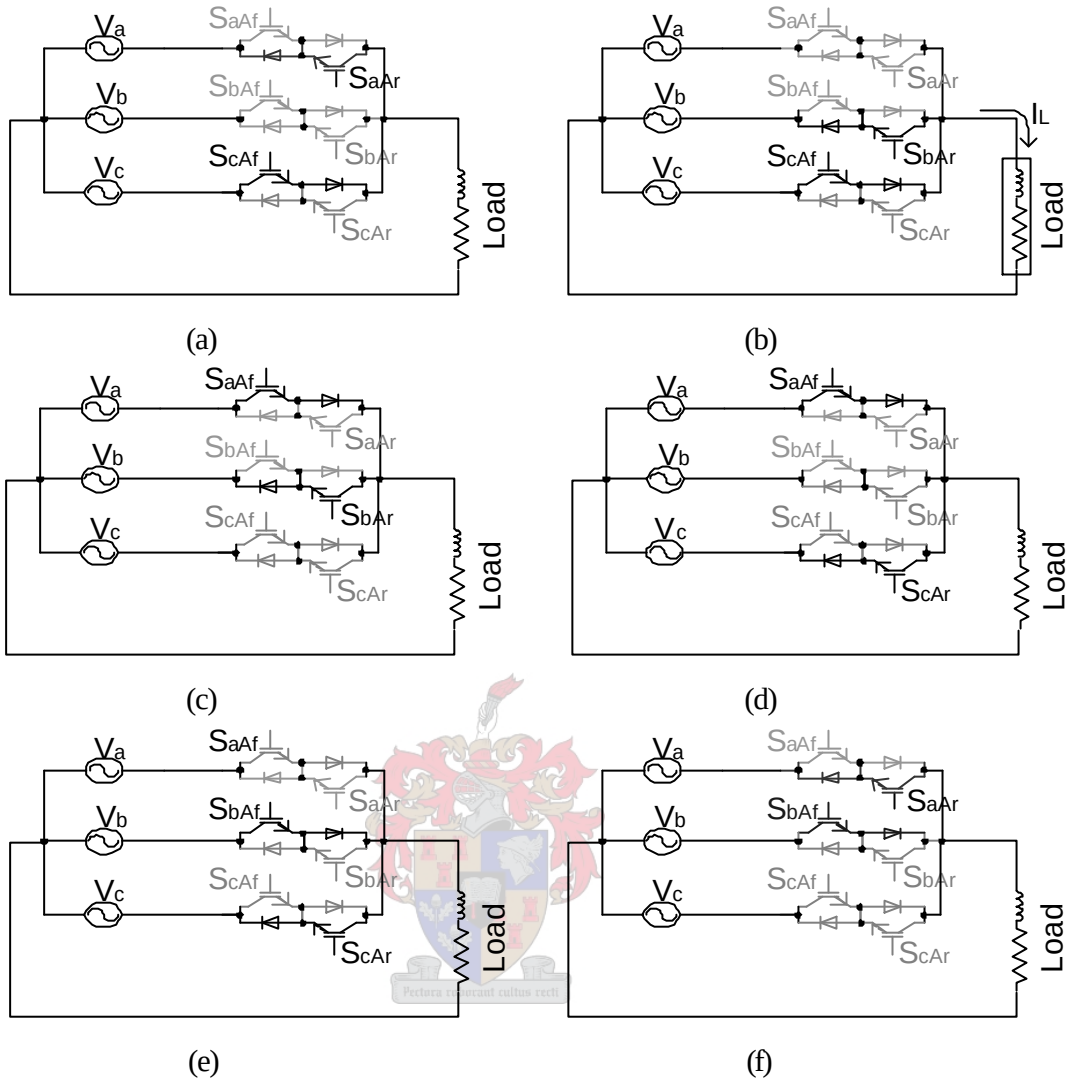
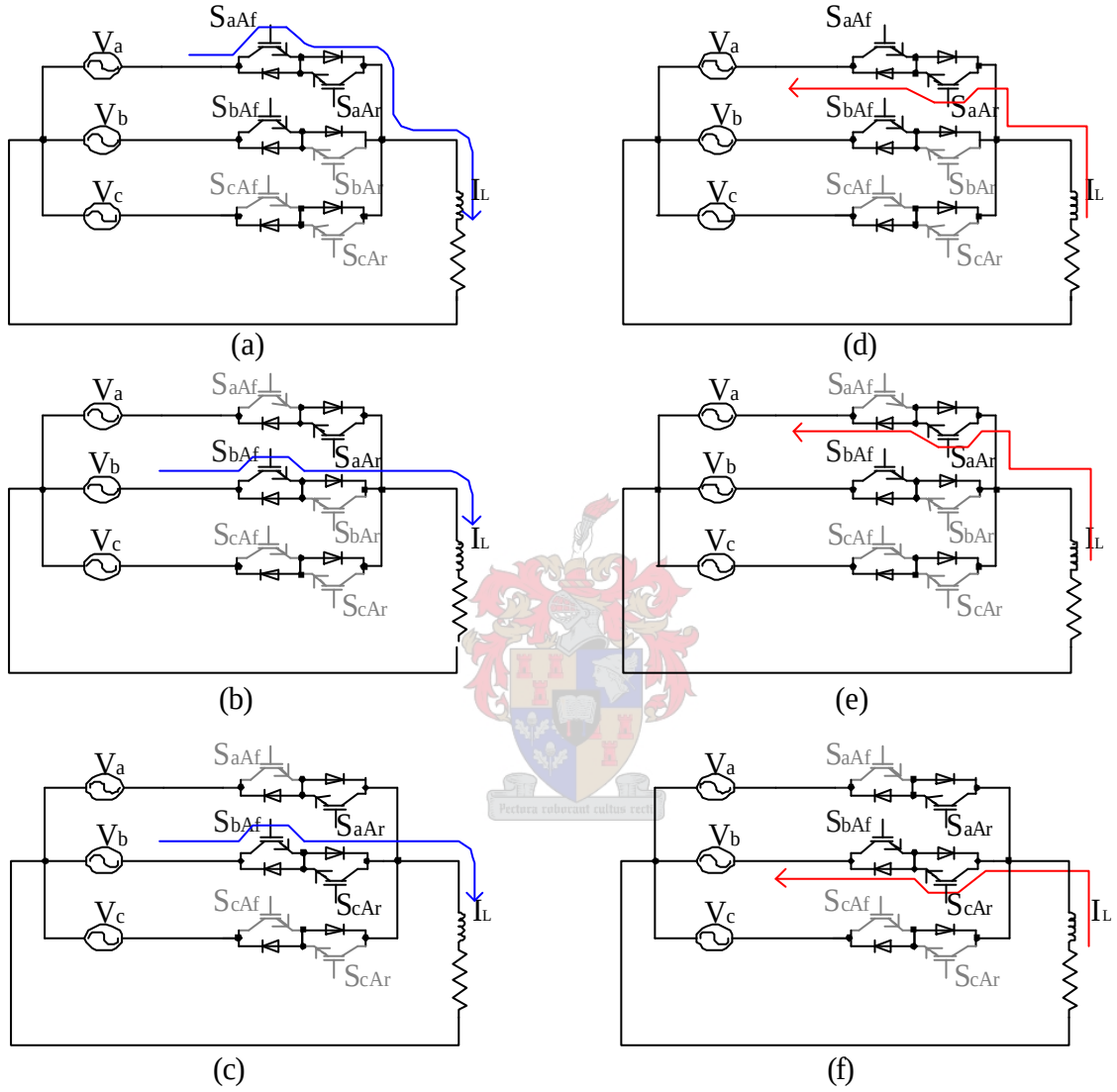


Figure 2-18 The six switching intervals of the voltage commutation identified based on the relative input voltages. (a) Switching state of interval-I, (b) switching state of interval-II, (c) switching state of interval-III, (d) switching state of interval-IV, (e) switching state of interval-V, (f) switching state of interval-VI.

The steps that are applied for a safe commutation using the two-step safe commutation strategy [3],[4], [9], [11] when switching from input phase voltage V_a to V_b is shown in Figure 2-18. Detection of the load current direction is not necessary. A reliable knowledge of relative input voltages magnitude is enough for the safe commutation.

Figure 2-19 shows a safe commutation of the three-phase to single-phase converter in interval VI and for $I_L > 0$, and $I_L < 0$.



$V_1 > V_2 > V_3 : I_L \geq 0$ (Forced commutation) $V_1 > V_2 > V_3 : I_L \leq 0$ (Natural commutation)

Figure 2-19 Two-Step safe Commutation for a three-phase to single-phase converter

Consider that the input voltages are in interval VI as shown in Figure 2-17. The maximum and minimum input phase voltages in this interval are V_a and V_b respectively, and their corresponding safe commutating switches are S_{aAr} and S_{bAf}. For input phase

voltage V_a supplying positive load current, where initially the switches S_{aAf}, S_{aAr} and S_{bAf} are turned on, and S_{aAf} is conducting the load current, while the remaining switches are turned off is shown in Figure 2-19(a). Eventually commutation to input phase voltage V_b is required. The Switch S_{aAf} is turned off and the switch S_{bAf} will conduct the load current through as shown in Figure 2-19(b), and a hard turn off occurs on the switch S_{aAf}. This type of commutation is known as a forced (hard) commutation. Switch S_{bAr} is turned on to allow the reverse flow of load current as shown in Figure 2-19(c). But, if the load current I_L is in the negative direction, the switch S_{aAr} will conduct the load current. Eventually, when commutation is needed; the switch S_{aAf} is turned off. Since the switch S_{aAf} was not conducting the load current, a natural commutation occurs, as shown in Figure 2-19(d) and Figure 2-19(e). The switch S_{bAr} is turned on at step two to conduct the load current as shown in Figure 2-19(f).

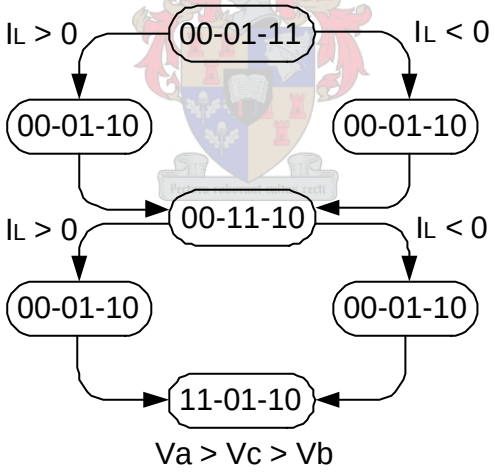


Figure 2-20 State diagram of a semi-natural two-step commutation

A state diagram that shows the sequence of commutation from phase-a to phase-b and then to phase-c for both load current directions is shown in Figure 2-20. Both sequences are the same, but for the positive load current direction $I_L > 0$, a forced commutation occurs, and a natural commutation occurs for the negative load current direction $I_L < 0$, as shown in Figure 2-19.

The two-step commutation method described in the above section for a three-phase to single-phase converter can be extended to a three-phase-to-three-phase direct matrix converter as shown in Figure 2-21. Table 2-1 shows the switching state of the bi-directional switches for the six intervals. It also contains the maximum and minimum input voltage during one period of the input voltages.

Table 2-1 Switch state during the six intervals for a 3Æ/3Æ AC/AC matrix converter

Angle ($?_{in}$)	V_{max}	V_{min}	Interval	Active Switch states		
				Phase-A	Phase-B	Phase-C
$p/2 \square 5p/6$	V_b	V_a	*I*	$S_{bAr} = 1, S_{aAf} = 1$	$S_{bBr} = 1, S_{aBf} = 1$	$S_{bCr} = 1, S_{aCf} = 1$
$5p/6 \square 7p/6$	V_c	V_a	*II*	$S_{cAr} = 1, S_{aAf} = 1$	$S_{cBr} = 1, S_{aBf} = 1$	$S_{cCr} = 1, S_{aCf} = 1$
$7p/6 \square 3p/2$	V_c	V_b	*III*	$S_{cAr} = 1, S_{bAf} = 1$	$S_{cBr} = 1, S_{bBf} = 1$	$S_{cCr} = 1, S_{bCf} = 1$
$3p/2 \square 11p/6$	V_a	V_b	*IV*	$S_{aAr} = 1, S_{bAf} = 1$	$S_{aBr} = 1, S_{bBf} = 1$	$S_{aCr} = 1, S_{bCf} = 1$
$-p/6 \square p/6$	V_a	V_c	*V*	$S_{aAr} = 1, S_{cAf} = 1$	$S_{aBr} = 1, S_{cBf} = 1$	$S_{aCr} = 1, S_{cCf} = 1$
$p/6 \square p/2$	V_b	V_c	*VI*	$S_{bAr} = 1, S_{cAf} = 1$	$S_{bBr} = 1, S_{cBf} = 1$	$S_{bCr} = 1, S_{cCf} = 1$

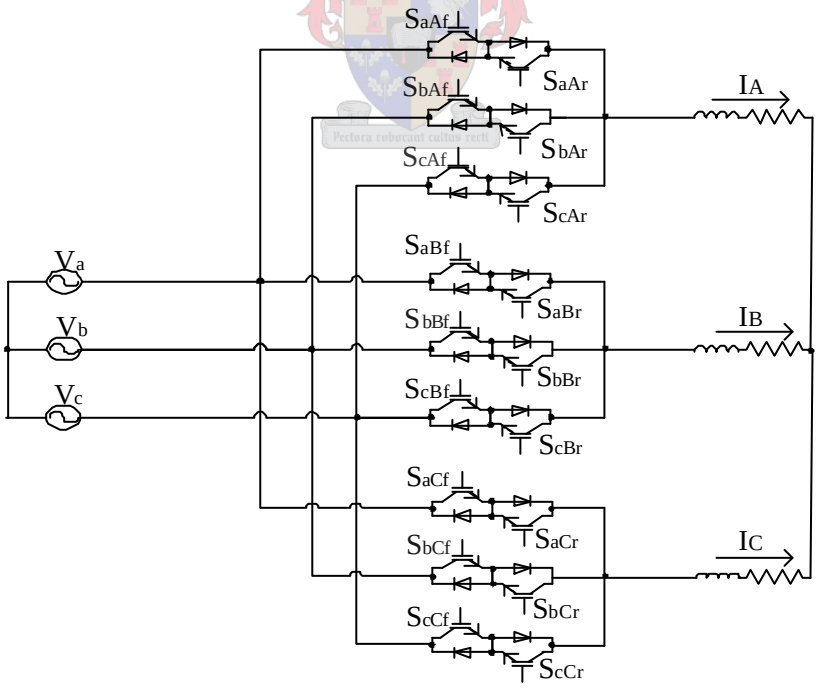
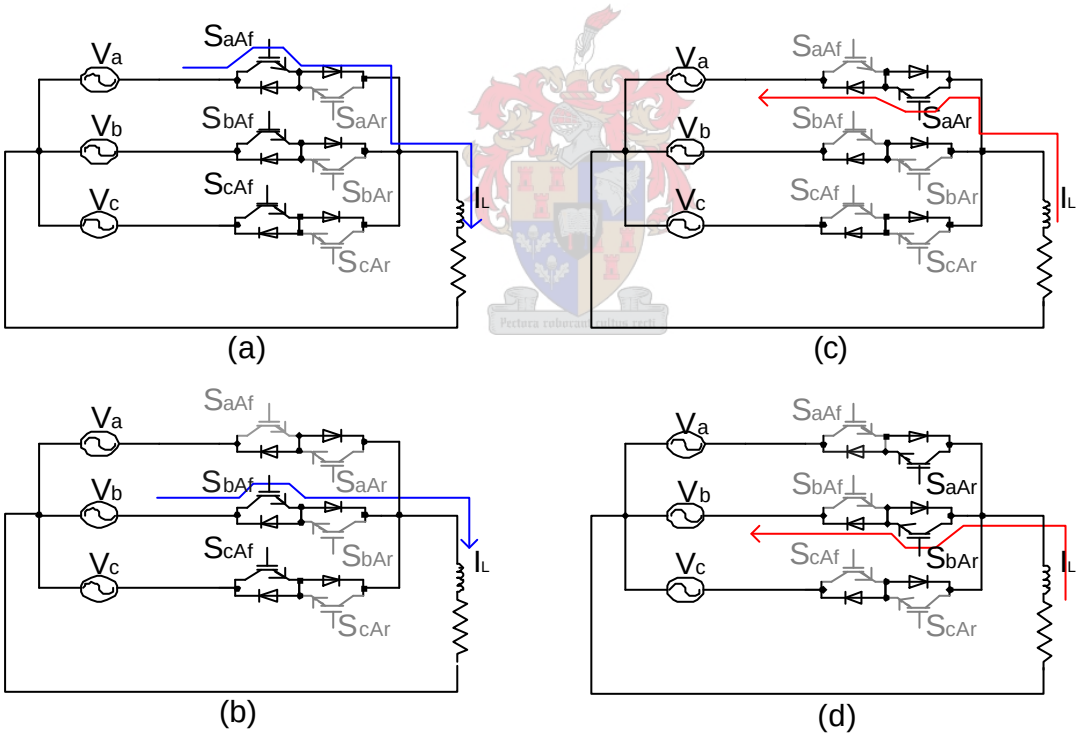


Figure 2-21 The bi-directional switches of the three-phase AC/AC matrix converter

The rest of the bi-directional switches of the three-phase AC-AC direct matrix converter are modulated depending on the control strategy used. In this case, a direct modulation technique is applied to control the switching of the bi-directional switches at each interval.

2.5.2.3 One-Step Safe Commutation Strategy

The one-step commutation depends on the knowledge of both load current direction and relative input voltage information. The incoming switch that will conduct the load current is known and turned on before turning off the out-going switch. The steps of this commutation method are shown in Figure 2-22 for both negative and positive polarity of the load current direction.



$V_1 > V_2 > V_3 : I_L \geq 0$ $\qquad$ $V_1 > V_2 > V_3 : I_L \leq 0$

Figure 2-22 One-Step safe Commutation

Consider interval I, where the relative input voltages are given as $V_a > V_b > V_c$, and positive load current direction, $I_L > 0$ as shown in Figure 2-22(a). Since the input voltage V_a is the maximum, the forward diodes of phase-b and phase-c are reverse biased for S_{aAf}, S_{bAf} and S_{cAf} are turned on while the remaining switches are turned off. Initially, the forward switch S_{aAf} conducts the load current. Eventually, when commutation to phase-b is needed, the outgoing switch S_{aAf} is turned off as shown in Figure 2-22(b) and the forward switch S_{bAf} will take over the action of conducting the load current. Only one-step is required to commutate between phases. The incoming switch is already turned on before the outgoing switch is turned off. For the same interval I of V_a the maximum, but reversed load current direction ($I_L < 0$), the load current conducting switch S_{aAr} is turned on only at the start while the remaining switches are turned off as shown in Figure 2-22(c). The moment the reverse switch S_{bAr} is turned on the load current is immediately carried by this switch S_{bAr} as shown in Figure 2-22(d). Only one-step is needed to switch from input phase-a to input phase-b. The same approach is extended to commutate to the third phase-c as well.

2.5.3 Modulation Technique for Direct Matrix Converter

The modulation technique is proposed in [1-3], [5] to control the switching state of the 9 bi-directional switches of the converter so that the desired variable frequency and multilevel voltage is generated. The output waveform is generated by selecting each of the input voltage phases in sequence for a defined time according to the modulation strategy. The first modulation strategy was introduced in 1980 [5], and is called the direct modulation approach. This modulation strategy produces 50% of the input phase voltage as the output voltage of the converter. Further improvement of the strategy increased the ratio of output-to-input to 86.67% [1-3]. Other modulation strategies have also been proposed [6], [10] that treat the matrix converter as a two-stage transformation. These modulation strategies are termed as the indirect modulation

approaches, since the modulation strategy controls a rectifier stage circuit and an inverter stage circuit.

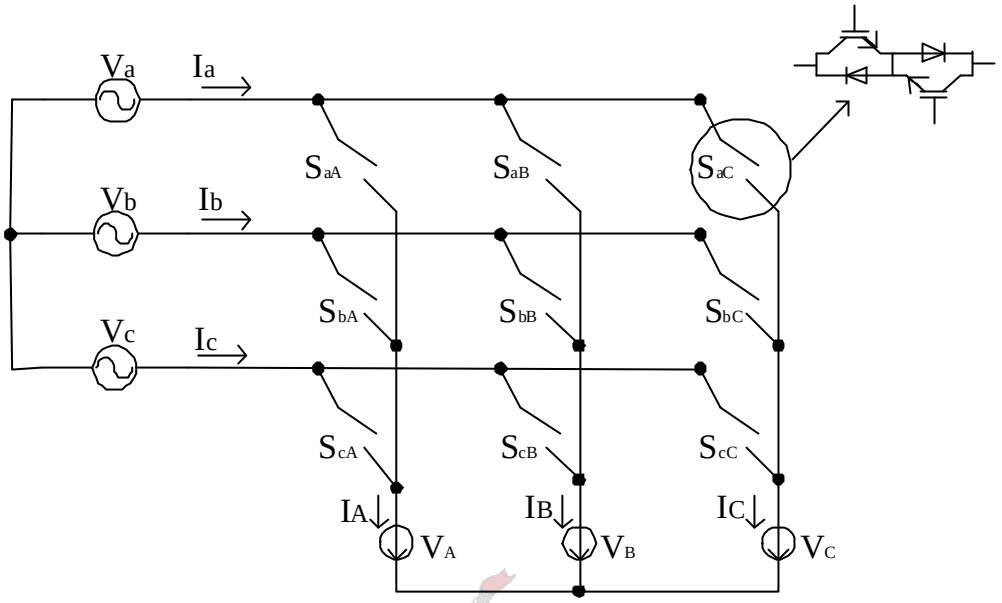


Figure 2-23 Three-phase AC -AC direct matrix converter

Consider a three phase to three phase matrix converter consisting of 9 bi-directional switches that connect the input phases a, b, and c to the output phases A, B, and C as shown in Figure 2-23. The switching function of the switches, $S_{kj}(t)$ in Figure 2-23 is defined as '0' when it is open and '1' when it is closed as given by Equation 2-3.

$$S_{kj}(t) = \begin{cases} 1 \\ 0 \end{cases}, \quad \text{for } k \in \{a, b, c\} \text{ and } j \in \{A, B, C\}. \qquad \textbf{Equation 2-3}$$

Since the input supply is a voltage source, short circuit of the input phases is not allowed, and open circuit of the load is not possible either because an inductive load generates an overvoltage. Therefore, this constraint of the switches is expressed by Equation 2-4.

$$S_{aj} + S_{bj} + S_{cj} = 1, \quad \text{for } j \in \{A, B, C\} \qquad \textbf{Equation 2-4}$$

With the above constraints, the bidirectional witches can assume only 27 allowed switching combination modes as shown in
Table **2-2** from 512 possible combinations. These switching combination modes are categorized into three groups as follows:

1. Group 1 consists of switching combinations, where each output phase is connected to a different input phase. Three switching combination modes belong to this group.

2. The second group consists of those switching combinations where only two output phases are shorted,

3. The third group consists of those switching combinations where all the three output phases shorted. This combination is called a zero output phase switching combination.

Table 2-2 Switching combinations of the switches for three-phase AC-AC direct matrix converter

Group		Phase Connection			Output line voltage			Input phase current		
		A	B	C	V_{AB}	V_{BC}	V_{CA}	i_a	i_b	i_c
I		a	b	c	V_{ab}	V_{bc}	V_{ca}	i_A	i_B	i_C
		a	c	b	$-V_{ca}$	$-V_{bc}$	$-V_{ab}$	i_A	i_C	i_B
		b	a	c	$-V_{ab}$	$-V_{ca}$	$-V_{bc}$	i_B	i_A	i_C
		b	c	a	V_{bc}	V_{ca}	V_{ab}	i_C	i_A	i_B
		c	a	b	V_{ca}	V_{ab}	V_{bc}	i_B	i_C	i_A
		c	b	a	$-V_{bc}$	$-V_{ab}$	$-V_{ca}$	i_C	i_B	i_A
II	A	a	c	c	$-V_{ca}$	0	V_{ca}	i_A	0	$-i_A$
		b	c	c	V_{bc}	0	$-V_{bc}$	0	i_A	$-i_A$
		b	a	a	$-V_{ab}$	0	V_{ab}	$-i_A$	i_A	0
		c	a	a	V_{ca}	0	$-V_{ca}$	$-i_A$	0	i_A
		c	b	b	$-V_{bc}$	0	$-V_{ca}$	0	$-i_A$	i_A
		a	b	b	V_{ab}	0	$-V_{ab}$	i_A	$-i_A$	0
	B	c	a	c	V_{ca}	$-V_{ca}$	0	i_B	0	$-i_B$
		c	b	c	$-V_{bc}$	V_{bc}	0	0	i_B	$-i_B$
		a	b	a	V_{ab}	$-V_{ab}$	0	$-i_B$	i_B	0
		a	c	a	$-V_{ca}$	V_{ca}	0	$-i_B$	0	i_B
		b	c	b	V_{bc}	$-V_{bc}$	0	0	$-i_B$	i_B
		b	a	b	$-V_{ab}$	V_{ab}	0	i_B	$-i_B$	0
	C	c	c	a	0	V_{ca}	$-V_{ca}$	i_C	0	$-i_C$
		c	c	b	0	$-V_{bc}$	V_{bc}	0	i_C	$-i_C$
		a	a	b	0	V_{ab}	$-V_{ab}$	$-i_C$	i_C	0
		a	a	c	0	$-V_{ca}$	V_{ca}	$-i_C$	0	i_C
		b	b	c	0	V_{bc}	$-V_{bc}$	0	$-i_C$	i_C
		b	b	a	0	$-V_{ab}$	V_{ab}	i_C	$-i_C$	0
III		a	a	a	0	0	0	0	0	0
		b	b	b	0	0	0	0	0	0

	c	c	c	0	0	0	0	0	0

The output phase voltage of the matrix converter is a product of the switching function and the input phase voltages given by Equation 2-5.

$$\begin{pmatrix} V_A \\ V_B \\ V_C \end{pmatrix} = \begin{pmatrix} S_{aA} & S_{bA} & S_{cA} \\ S_{aB} & S_{bB} & S_{cB} \\ S_{aC} & S_{bC} & S_{cC} \end{pmatrix} * \begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix} \qquad \textbf{Equation 2-5}$$

Equivalently Equation 2-5 can be presented in a more concise form as given by Equation 2-6.

$$\left[V_{op}(t)\right] = \left[M_{phph}(t)\right] * \left[V_{ip}(t)\right] \qquad \textbf{Equation 2-6}$$

Where: $[M(t)]$ is the modulation matrix function, $\left[V_{ip}(t)\right]$ is the input phase voltages matrix, and $\left[V_{op}(t)\right]$ is the output phase voltage matrix.

Similarly, the input phase current $[i_i(t)]$ is expressed as the product of the transpose of the modulation matrix function $[M(t)]$ and the output phase current $\left[i_{op}(t)\right]$.

$$\begin{pmatrix} i_a \\ i_b \\ i_c \end{pmatrix} = \begin{pmatrix} S_{aA} & S_{bA} & S_{cA} \\ S_{aB} & S_{bB} & S_{cB} \\ S_{aC} & S_{bC} & S_{cC} \end{pmatrix}^T * \begin{pmatrix} i_A \\ i_B \\ i_C \end{pmatrix} \qquad \textbf{Equation 2-7}$$

$$\left[i_{ip}(t)\right] = \left[M_{phph}(t)\right]^T * \left[i_{op}(t)\right] \qquad \textbf{Equation 2-8}$$

Alternatively, the output line voltage can be expressed as:

$$\begin{pmatrix} V_{AB} \\ V_{BC} \\ V_{CA} \end{pmatrix} = \begin{pmatrix} S_{aA} - S_{aB} & S_{bA} - S_{bB} & S_{cA} - S_{cB} \\ S_{aB} - S_{aC} & S_{bB} - S_{bC} & S_{cB} - S_{cC} \\ S_{aC} - S_{aA} & S_{bC} - S_{bA} & S_{cC} - S_{cA} \end{pmatrix} * \begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix} \quad \textbf{Equation 2-9}$$

$$[V_{oL}(t)] = [M_{phL}(t)] * [V_{ip}(t)] \quad \textbf{Equation 2-10}$$

Where $[V_{oL}(t)]$ is the output line voltage.

Similarly, the input phase current is expressed as:

$$[i_{ip}(t)] = [M_{phL}(t)]^T * [i_{oL}(t)] \quad \textbf{Equation 2-11}$$

Where $[i_{oL}(t)]$ is the output line to line current.

The switching time of the modulation matrix function $[M(t)]$ of the converter is given as:

$$[M(t)] = \begin{pmatrix} m_{aA} & m_{bA} & m_{cA} \\ m_{aB} & m_{bB} & m_{cB} \\ m_{aC} & m_{bC} & m_{cC} \end{pmatrix} \quad \textbf{Equation 2-12}$$

Where: m_{aA}, m_{bA}, and m_{cA} are the switching time assigned to phase voltages V_a, V_b, and V_c respectively to generate the output phase voltage V_A, and so forth.

The modulation duty cycle for each switch of the converter is calculated as the ratio of the switching time $m_{kj}(t)$ to the switching period T_s.

$$d_{kj}(t) = \frac{m_{kj}(t)}{T_s} \text{ , for k} \in \{a, b, c\} \text{ and j} \in \{A, B, C\}. \quad \textbf{Equation 2-13}$$

Where: $m_{kj}(t)$ is the switching time interval of switch $S_{kj}(t)$.

Given a set of input phase voltages and an assumed set of output phase currents,

$$\begin{aligned} V_a &= V_{im}\cos(w_i t) \\ V_b &= V_{im}\cos(w_i t - 2p/3) \\ V_c &= V_{im}\cos(w_i t - 4p/3) \end{aligned} \qquad \textbf{Equation 2-14}$$

$$\begin{aligned} i_A &= I_{om}\cos(w_o t + \varphi_o) \\ i_B &= I_{om}\cos(w_o t + \varphi_o - 2p/3) \\ i_C &= I_{om}\cos(w_o t + \varphi_o - 4p/3) \end{aligned} \qquad \textbf{Equation 2-15}$$

Where: V_{im} is the peak value of the input phase voltages and w_i denotes an angular frequency of the input voltages, and I_{om} is the peak value of the assumed output phase currents and w_o denotes an angular frequency of the output currents.

The matrix converter is then designed and controlled in such a manner that the fundamental of the output phase voltages and input phase currents are:

$$\begin{aligned} V_A &= V_{om}\cos(w_o t) \\ V_B &= V_{om}\cos(w_o t - 2p/3) \\ V_C &= V_{om}\cos(w_o t - 4p/3) \end{aligned} \qquad \textbf{Equation 2-16}$$

$$\begin{aligned} i_a &= I_{im}\cos(w_i t + \varphi_i) \\ i_b &= I_{im}\cos(w_i t + \varphi_i - 2p/3) \\ i_c &= I_{im}\cos(w_i t + \varphi_i - 4p/3) \end{aligned} \qquad \textbf{Equation 2-17}$$

Where: V_{om} is the peak value of the desired output phase voltages, w_o denotes an angular frequency of the desired output voltages, I_{im} is the peak value of the desired input phase currents, and w_i denotes an angular frequency of the input currents.

The peak value of the desired output phase voltage V_{om} is expressed as the product of modulation index m and the peak value of the input phase voltage V_{im}.

$$V_{om} = mV_{im} \qquad \textbf{Equation 2-18}$$

The modulation matrix function $[M(t)]$ of Equation 2-12 has to be calculated so that the desired frequency and magnitude controlled output voltage matrix $\left[V_{op}(t)\right]$ and input current $\left[i_{ip}(t)\right]$ vector can be generated.

There are two basic solutions that satisfy the equations given by Equation 2-14 and 2-15, which is proposed in [1-3], [5].These modulation matrices are of a sinusoidal function of frequency $(w_o \pm w_i)\cdot t$.

$$[M_1(t)] = \begin{pmatrix} 1+2m\cos(w_m t) & 1+2m\cos(w_m t-\frac{2p}{3}) & 1+2m\cos(w_m t-\frac{4p}{3}) \\ 1+2m\cos(w_m t-\frac{2p}{3}) & 1+2m\cos(w_m t-\frac{4p}{3}) & 1+2m\cos(w_m t) \\ 1+2m\cos(w_m t-\frac{4p}{3}) & 1+2m\cos(w_m t) & 1+2m\cos(w_m t-\frac{2p}{3}) \end{pmatrix} w_m = w_o - w_i \qquad \textbf{Equation 2-19}$$

This solution yields the same power factor at the input side and the output side of the converter, $\varphi_{in} = \varphi_{out}$.

$$[M_2(t)] = \begin{pmatrix} 1+2m\cos(w_m t) & 1+2m\cos(w_m t-\frac{2p}{3}) & 1+2m\cos(w_m t-\frac{4p}{3}) \\ 1+2m\cos(w_m t-\frac{4p}{3}) & 1+2m\cos(w_m t) & 1+2m\cos(w_m t-\frac{4p}{3}) \\ 1+2m\cos(w_m t-\frac{2p}{3}) & 1+2m\cos(w_m t-\frac{4p}{3}) & 1+2m\cos(w_m t) \end{pmatrix} w_m = w_o + w_i \qquad \textbf{Equation 2-20}$$

Whereas the modulation matrix function $[M_2(t)]$ generates a reversed power factor of the output at the input side of the converter, $\varphi_{in} = -\varphi_{out}$. Combining the above two

solutions provides a means for controlling the input power factor. The combined result of the modulation functions is given by:

$$[M(t)] = \alpha_1 [M_1(t)] + \alpha_2 [M_2(t)] \qquad \textbf{Equation 2-21}$$

α_1 and α_2 can be set equal to give a unity input power factor regardless of the load power factor. In addition, a leading power factor (capacitive) at the input side of the converter or a lagging power factor (inductive) at the load side of the converter and vice versa, is possible by selecting proper value of the two constants, α_1 and α_2.

If $\alpha_1 = \alpha_2$, then the modulation matrix can be written as:

$$d_{kj}(t) = \frac{m_{kj}(t)}{T_s} = \frac{1}{3}\left(1 + \frac{2 * V_j * V_k}{V_{im}^{\ 2}}\right) \quad \text{for k = a, b, c and j = A, B, C.} \qquad \textbf{Equation 2-22}$$

Using this solution the maximum value of input to output voltage ratio m the converter can achieve is 0.5, as shown in Figure 2-24.

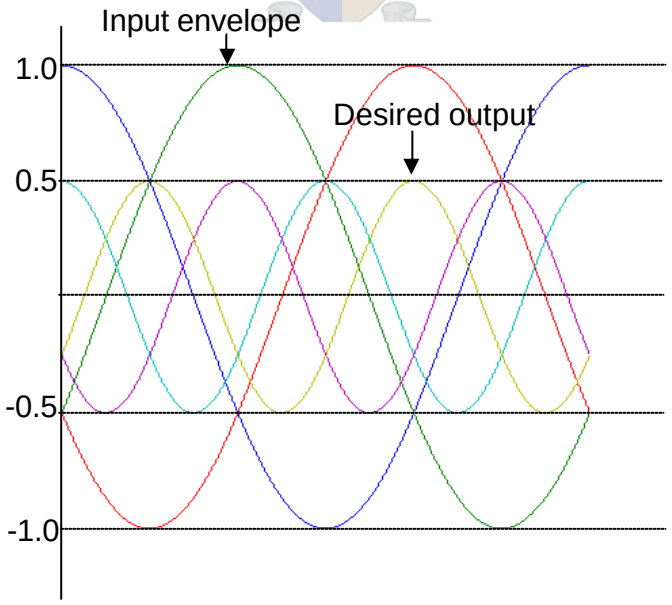


Figure 2-24 The input phase voltages and the generated output phase voltages for m=0.5

The maximum gain of the converter is inherently limited to 50%. The modulation matrix $M(t)$ yields a maximum 50% of the input to output voltage ratio, which is the limitation of the converter. The gain of the converter can be improved to 86.6% by injecting the third harmonic components of the input and output frequencies [1-3], [5]. These third harmonic components of the converter will be cancelled in a three-phase load in the same way as the third harmonic component addition does in a conventional inverter. In this case, the modulation matrix equation given in Equation2-22 will be:

$$d_{kj}(t) = \frac{1}{3}\left(1 + \frac{2 * V_j * V_k}{V_{im}{}^2} + \frac{4m \cdot sin(3w_i t + b_k)sin(w_i t)}{3\sqrt{3}}\right) \qquad \textbf{Equation 2-23}$$

for k = a, b, c , j = A, B, C and $\beta_k = 0, 2\pi/3, 4\pi/3$ for k = a, b, c respectively.

Figure 2-25 shows the input envelope voltage and the generated output voltage for modulation index of 0.866.

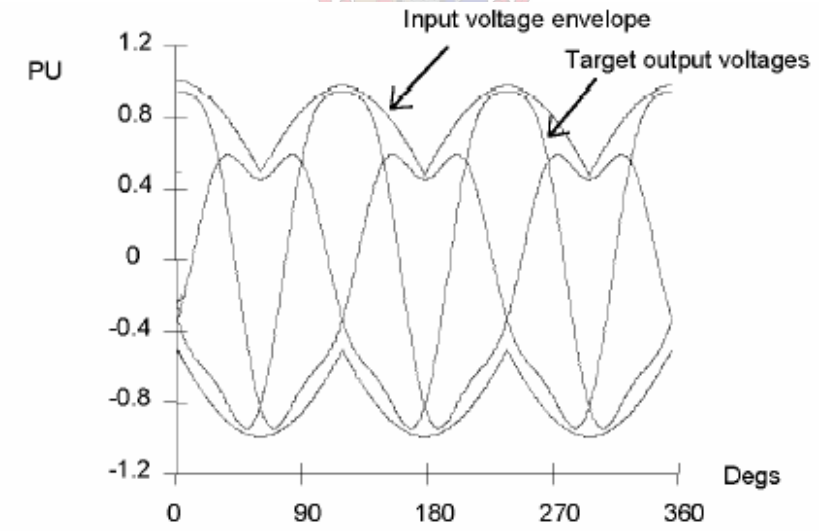


Figure 2-25 Input phase envelop voltages and the generated output phase voltages for m=0.866.

2.6 Indirect Matrix Converter Topology

The direct AC-AC matrix converter topology has a simple structure and many attractive features; but the complexity of its conventional PWM control strategy and the commutation problem prevent it from being used in industry. An alternative approach to overcome these failures is proposed [6], [10-11]. It is a two-stage converter topology known as an indirect matrix converter. This topology is similar to the conventional inverter-based converter topology without any reactive DC-link energy storage components for the intermediate imaginary DC-link bus. A block diagram of the indirect matrix converter topology is shown in Figure 2-26.

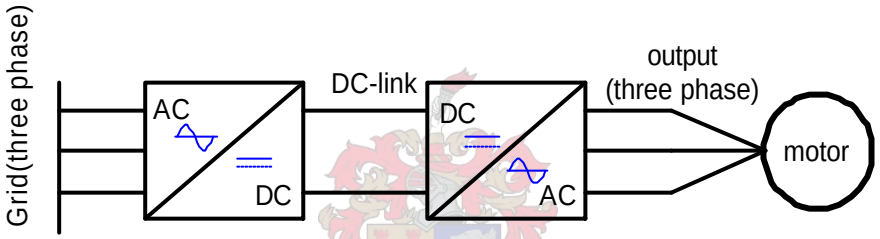


Figure 2-26 Indirect matrix converter topology block diagram feeding to an AC-drive

All the desired features of the direct matrix converter topology, such as sinusoidal input current and sinusoidal output voltage, four-quadrant operation, unity power factor, no DC-storage elements are achieved by this indirect matrix converter topology. In addition, this topology simplifies the complexity of the conventional PWM control strategy, and overcome the commutation problems of the previous topology. Besides, commutation at zero input current is possible. The indirect matrix converter topology shown in Figure 2-27 is treated as a two-stage transformation converter: A rectifier stage to provide an imaginary DC-link during the switching cycle and an inverter stage to generate the three phase desired output voltages.

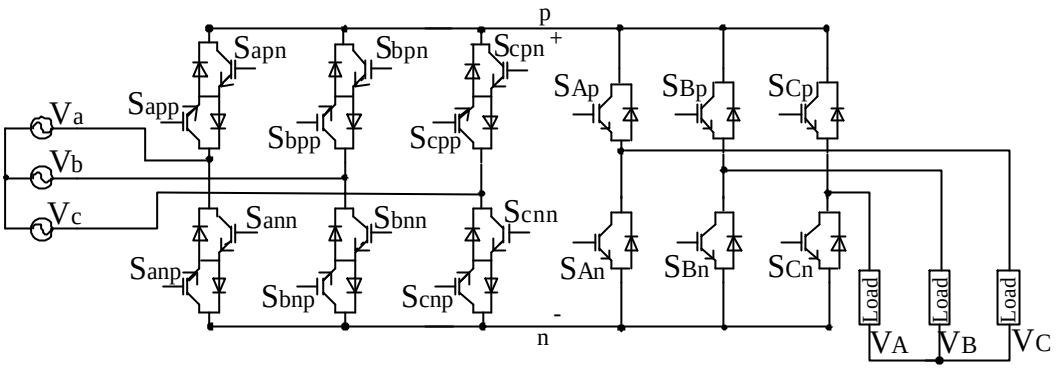


Figure 2-27 Indirect matrix converter topology

2.6.1 Derivation of Indirect Matrix Converter Topology

The modulation function of a single-stage direct matrix converter topology for a three-phase converter in Equation 2-12 is given again in Equation 2-24 here.

$$[M(t)] = \begin{pmatrix} m_{aA} & m_{bA} & m_{cA} \\ m_{aB} & m_{bB} & m_{BcB} \\ m_{aC} & m_{bC} & m_{cC} \end{pmatrix} \qquad \textbf{Equation 2-24}$$

The modulation function of the indirect matrix converter topology is derived from the single modulation function in Equation 2-24 and is expressed as the product of two different transfer functions.

$$[M(t)] = [I(t)] * [R(t)] \qquad \textbf{Equation 2-25}$$

Where: $I(t)$ is the Inverter transfer function and $R(t)$ is the rectifier transfer function.

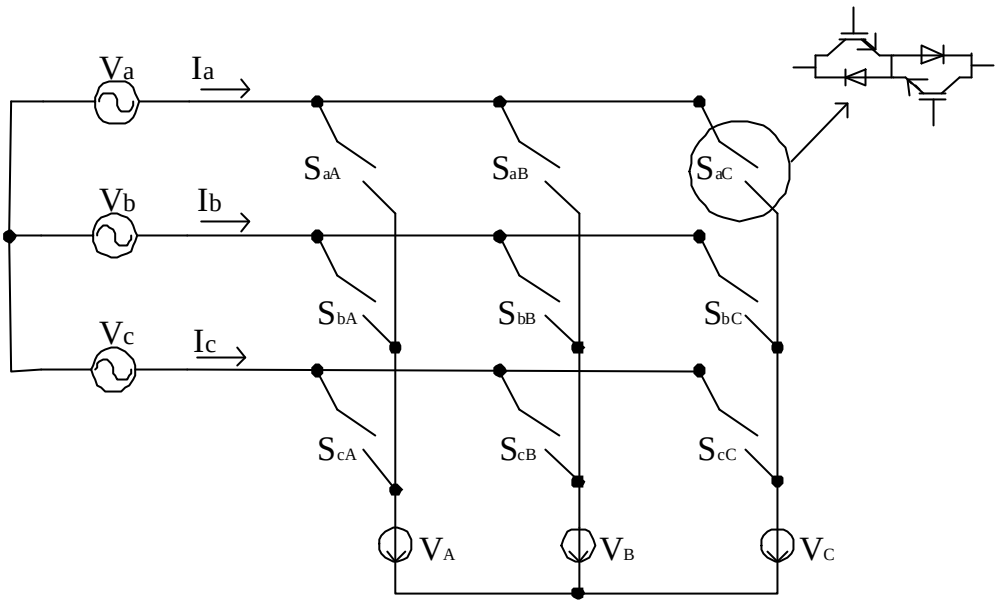


Figure 2-28 Three-phase AC-AC direct matrix converter

Consider the direct matrix converter topology of Figure 2-28 . The input terminals of the converter are connected to the three-phase input voltages V_a, V_b, and V_c. The three-phase desired output voltages V_A, V_B, and V_C of the converter are obtained using Equation 2-26.

$$\begin{pmatrix} V_A \\ V_B \\ V_C \end{pmatrix} = \begin{pmatrix} S_{aA} & S_{bA} & S_{cA} \\ S_{aB} & S_{bB} & S_{cB} \\ S_{aC} & S_{bC} & S_{cC} \end{pmatrix} * \begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix} \qquad \textbf{Equation 2-26}$$

Where the switching function $S_{jk}(t)$ is given as

$$S_{kj}(t) = \begin{cases} 1 \\ 0 \end{cases} \text{, for k} \in \{a, b, c\} \text{ and j} \in \{A, B, C\}. \qquad \textbf{Equation 2-27}$$

For balanced three-phase input voltages, the neutral voltage is equal to zero. Mathematical the relationship of the input voltages can be expressed by Equation 2-28.

$$V_a + V_b + V_c = 0$$ **Equation 2-28**

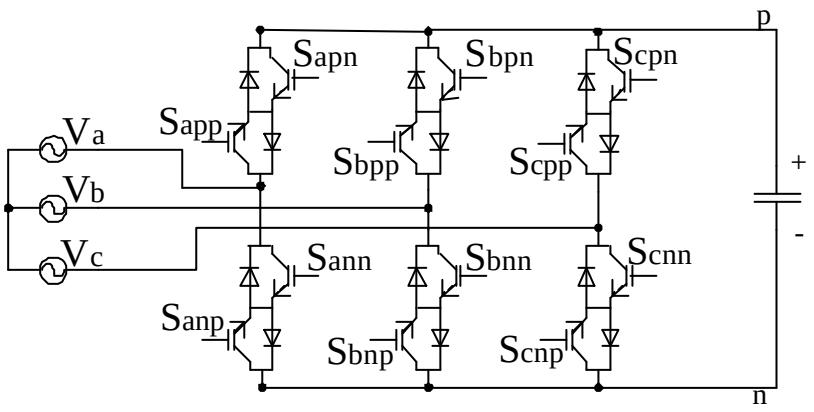
Equation 2-26 can be reduced into a two-stage converter as given by Equation 2-29 and 2-30.

$$\begin{pmatrix} V_p \\ V_n \end{pmatrix} = \begin{pmatrix} S_{ap} & S_{bp} & S_{cp} \\ S_{an} & S_{bn} & S_{cn} \end{pmatrix} * \begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix}$$ **Equation 2-29**

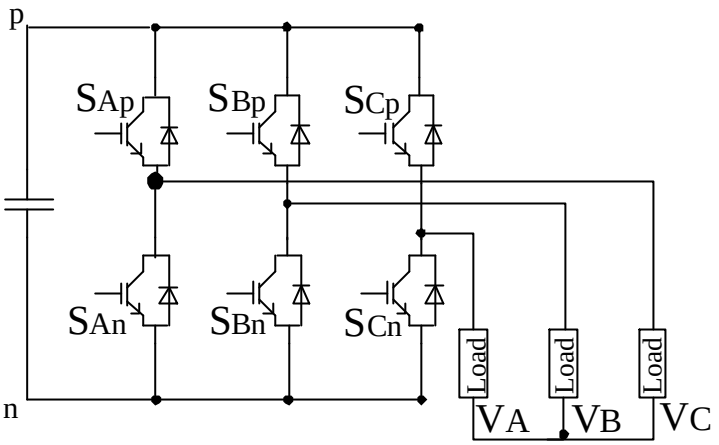
$$\begin{pmatrix} V_A \\ V_B \\ V_C \end{pmatrix} = \begin{pmatrix} S_{Ap} & S_{An} \\ S_{Bp} & S_{Bn} \\ S_{Cp} & S_{Cn} \end{pmatrix} * \begin{pmatrix} V_p \\ V_n \end{pmatrix}$$ **Equation 2-30**

Where: S_{ap}, S_{an}, S_{bp}, S_{bn} S_{cp} and S_{cn} are the switches connecting the three-phase input voltages to the rectifier side of the converter. The switches S_{Ap}, S_{An}, S_{Bp}, S_{Bn}, S_{Cp} and S_{Cn} connect the DC-link voltage to the inverter side of the converter. Nodes p and n are the positive and negative rails of the DC-link respectively.

Equation 2-29 is equivalent to the operation of a voltage source rectifier VSR, which provides a constant voltage V_{pn} as shown in Figure 2-29(a), and Equation 2-30 is equivalent to the operation of conventional inverter VSI connected to a constant voltage V_{pn} to produce three phase voltages as shown in Figure 2-29(b).



(a)



(b)

Figure 2-29 Stand-alone stages of the indirect matrix converter topology. (a) Rectifier stage; (b) Inverter stage.

2.6.2 Modulation Techniques of Indirect Matrix Converter Topology

There are two different modulation techniques proposed [6], [10] to control the switching of the indirect matrix converter. Both modulation techniques treat the matrix converter as a two-stage transformation. The first technique [6] uses space vector modulation for the inversion stage and a relative input voltage magnitude technique for the rectification stage. The second modulation technique proposed [10] uses two different approaches. These approaches are; the application of the space vector modulation (SVM) on both the rectification stage and the inversion stage of the indirect matrix converter, while the second approach involves a combination of the space vector modulation to modulate the rectifier stage, while a polygonal flux modulation (PFM) is applied to modulate the inverter stage.

A combined modulation strategy of the rectifier stage circuit and inverter stage circuit is used since it generates better output voltages, and smooth input currents are seen at the input line side. However, independent control of the rectifier stage and inverter stage is possible.

2.6.2.1 Modulation Technique Based on the Relative Input Voltages and SVM

The rectifier stage and inverter stage modulation strategy proposed in [6] are presented in this section. The modulation of the rectifier stage based on the relative input voltages method will first be presented and is followed by the space vector modulation (SVPWM) technique, which is applied to control the inverter stage. A combined modulation strategy will be presented at the end of the section.

2.6.2.1.1 Modulation Technique of the Rectifier Stage

The rectifier modulation strategy is defined to control the rectifier stage switches so that a constant voltage V_{pn} is provided to the inverter stage. This modulation strategy is based on the proposed paper in [6]. It uses the input voltage synchronization angle to identify six intervals in a given switching period.

The given input source voltages and the assumed output currents are:

$$\begin{aligned} V_a &= V_{im}\cos(w_i t) = V_{im}\cos(q_a) \\ V_b &= V_{im}\cos(w_i t - 2p/3) = V_{im}\cos(q_b) \\ V_c &= V_{im}\cos(w_i t - 4p/3) = V_{im}\cos(q_c) \end{aligned} \qquad \textbf{Equation 2-32}$$

$$\begin{aligned} i_A &= I_{om}\cos(w_o t + \varphi_o) \\ i_B &= I_{om}\cos(w_o t + \varphi_o - 2p/3) \\ i_C &= I_{om}\cos(w_o t + \varphi_o - 4p/3) \end{aligned} \qquad \textbf{Equation 2-33}$$

Where: w_i and w_o is the input and output angular frequency, φ_o is the initial phase angle of the phase current. V_{im} and I_{om} are the amplitudes of the input voltage and the output current respectively.

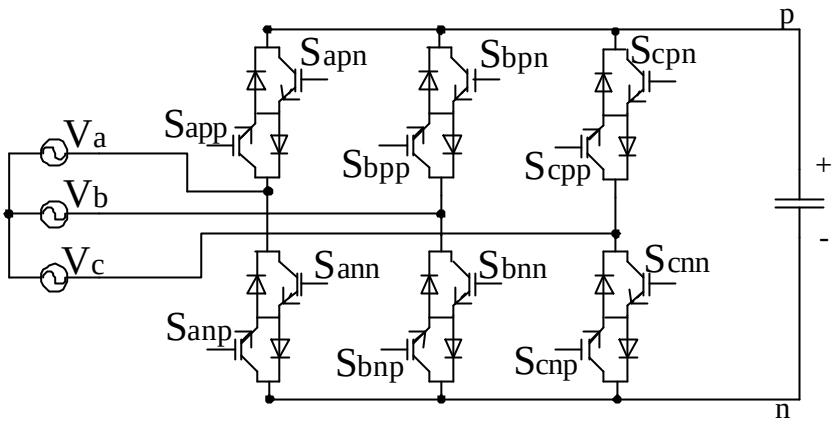


Figure 2-30 A stand alone rectifier converter

Consider the stand-alone rectifier shown in Figure 2-30. The modulation strategy applied to control the rectifier switches ensures a constant DC voltage is provided to the inverter side of the converter. This strategy is presented as follows:

A. Six switching intervals based on the line side voltage

From Figure 2-31 shown below, six switching intervals are identified depending on the input voltage synchronization angle. There is only one input voltage, which has the largest absolute value during each interval, out of the three input voltages. For example, V_a has the largest absolute voltage in the interval I and V_b has the largest absolute voltage in the interval III and so forth.

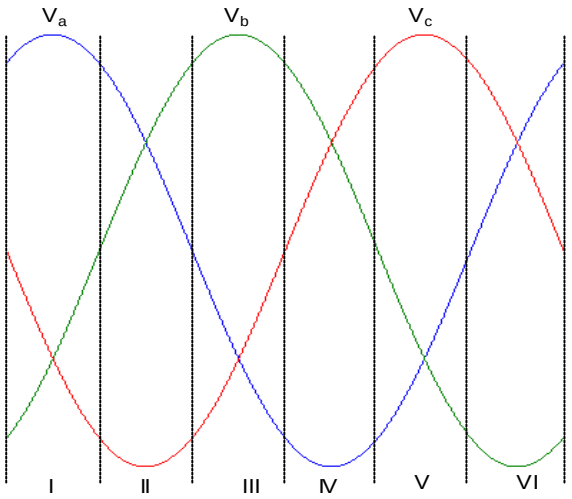


Figure 2-31 Six intervals based on the input synchronization angle

B. Two portions in each switching cycle

The switching cycle T_s is divided into two portions, where the switching state of the line side switch is fixed and the DC side voltage V_{pn} is equal to one of the two highest positive line voltages in each portion. Consider interval I, V_a has the largest absolute voltage and the two largest positive line voltages are $V_{ab} = (V_a - V_b)$ and $V_{ac} = (V_a - V_c)$ respectively. The line side switching states in each portion is determined by the following:

- Portion I: Switches S_{app}, S_{apn}, S_{bnn}, and S_{bnp} are turned on, and the rest of the line side switches are turned off. The DC voltage V_{pn} is then equal to the line voltage V_{ab}, and the DC current i_{pn} of the rectifier is equal to i_a and $-i_b$. The current i_c equals zero, since the switches are turned off. The duty cycle of this portion is defined as:

$$d_{ba} = \frac{-V_b}{V_a} = \frac{-\cos(q_b)}{\cos(q_a)} \qquad \textbf{Equation 2-34}$$

The constant DC voltage V_{pn} during this first portion is given as:

$$V_{pn} = d_{ba} * V_{ab} = d_{ba} * (V_a - V_b) \qquad \textbf{Equation 2-35}$$

- Portion II: The switches $S_{app}, S_{apn}, S_{cnn}$ and S_{cnp} are turned on and the rest of the line side switches are turned off. The DC voltage V_{pn} is then equal to the line voltage V_{ac}, and the DC current i_{pn} of the rectifier is equal to i_a and $-i_c$. The current i_b equals zero, since the switches are turned off. The duty cycle of this portion is defined as

$$d_{ca} = \frac{-V_c}{V_a} = \frac{-\cos(q_c)}{\cos(q_a)} \qquad \textbf{Equation 2-36}$$

The constant DC voltage V_{pn} during this second portion is given as:

$$V_{pn} = d_{ca} * V_{ac} = d_{ca} * (V_a - V_c) \qquad \textbf{Equation 2-37}$$

The constant DC voltage V_{pn} provided by the rectifier converter stage for a given switching period T_s is:

$$V_{pn} = d_{ba} * V_{ab} + d_{ca} * V_{ac}$$
$$V_{pn} = d_{ba} * (V_a - V_b) + d_{ca} * (V_a - V_c) \qquad \textbf{Equation 2-38}$$

Where: $d_{ba} + d_{ca} = 1$.

Substituting Equation 2-30, 2-32, and 2-35 in 2-38 will result in Equation 2-39:

$$V_{pn} = \frac{3}{2 \cdot \left|\cos(q_a)\right|} V_{im} \qquad \textbf{Equation 2-39}$$

Table 2-3 shows the switching state of the rectifier converter switches and the DC-link voltage V_{pn} provided for the given one cycle of the input synchronization angle.

Table 2-3 The duty cycle, switching state and DC-link voltage determined of the rectifier circuit

Intervals	I		II		III		IV		V		VI	
Input angle θ_a	$-\pi/6 \sim \pi/6$		$\pi/6 \sim \pi/2$		$\pi/2 \sim 5\pi/6$		$5\pi/6 \sim 7\pi/6$		$7\pi/6 \sim 3\pi/2$		$11\pi/2 \sim -\pi/6$	
Duty cycle	d_{ba}	d_{ca}	d_{ac}	d_{bc}	d_{cb}	d_{ab}	d_{ba}	d_{ca}	d_{ac}	d_{bc}	d_{cb}	d_{ab}
Conducting switches	S_{ap}		S_{cn}		S_{bp}		S_{an}		S_{cp}		S_{bn}	
	S_{bn}	S_{cn}	S_{ap}	S_{bp}	S_{cn}	S_{an}	S_{bp}	S_{cp}	S_{bn}	S_{an}	S_{ap}	S_{cp}
DC voltage	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}	V_{ab}

2.6.2.1.2 Modulation Technique of the Inverter Stage

Consider a three-legged voltage source inverter circuit shown in Figure 2-32, as a stand-alone voltage source inverter connected to a constant DC-voltage source.

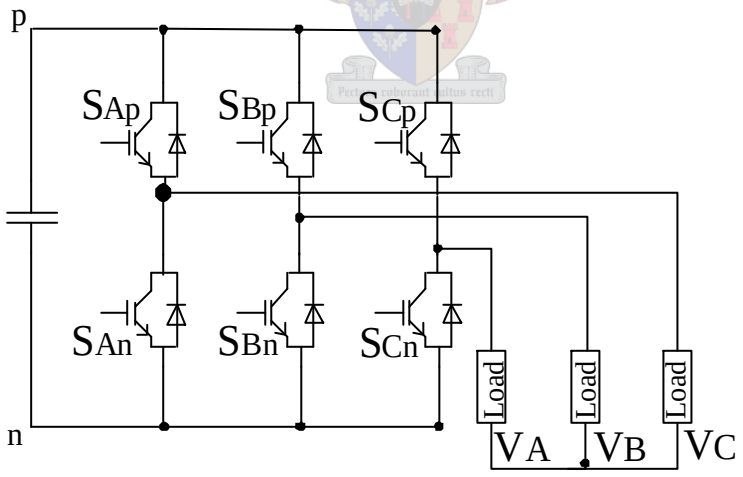


Figure 2-32 An Inverter stage circuit

Since the input lines of the VSI must never be shorted and a conducting path for the inductive load current must be provided, the switches of the VSI can only assume eight

combinations, as shown in Figure 2-33. The first six combinations produce non-zero output voltages and are known as the non-zero switching states. Whereas the remaining two combinations are zero output voltages, and are known as the zero switching states, as shown in Figure 2-33.

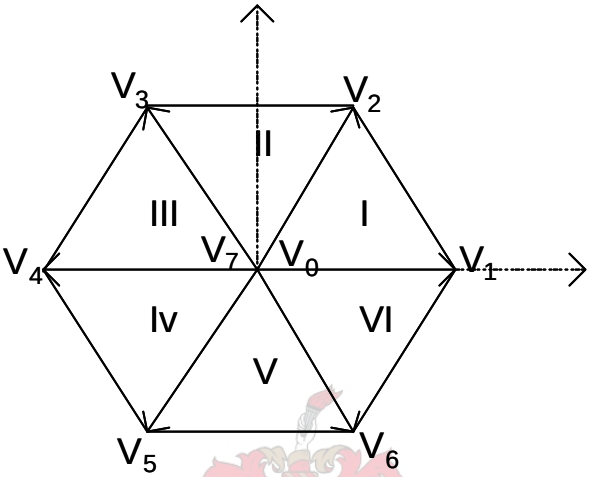


Figure 2-33 The active switching state vectors and the six sectors of SVM.

The switching states (non-zero and zero switching states) of the VSI generate seven discrete values as shown in Figure 2-33, and are known as voltage vectors $V_0 - V_7$. The tips of these voltage vectors form a hexagon of 60^0. Voltage vectors V_1 - V_6 are of the same magnitude displaced by 60^0 as shown in Figure 2-33. The remaining vectors, V_0 and V_7, are zero magnitude voltage vectors placed at the origin of the α-β plane as shown in Figure 2-33. The area enclosed by two adjacent vectors within the hexagon as shown in Figure 2-33, is known as a sector. The desirable output line voltages of the inverter can be represented by an equivalent space vector $\overline{V_{oL}}$ rotating in a counter clockwise direction at a speed of w_o angular frequency as shown in Figure 2-34.

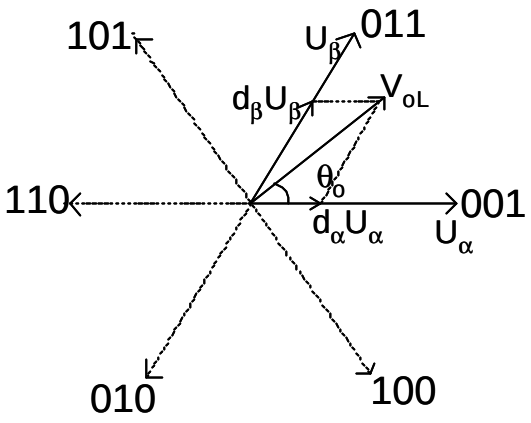


Figure 2-34 A space vector reference voltage $\overline{V_{oL}}$ in the $a-b$ plane.

The resulting space vector of the desired output line voltages is given by Equation 2-40.

$$V_{oL} = V_{AB} + V_{BC}e^{j2p/3} + V_{CA}e^{-j2p/3} \qquad \textbf{Equation 2-40}$$

Let's consider the VSI is a stand alone converter with three-phase output voltage V_A, V_B, and V_C supplied by a DC voltage source V_{pn}. It is assumed that the desired output voltages of the inverter are given as:

$$\begin{aligned} V_A &= V_{om}\cos(w_o t) \\ V_B &= V_{om}\cos(w_o t - 2p/3) \\ V_C &= V_{om}\cos(w_o t - 4p/3) \end{aligned} \qquad \textbf{Equation 2-41}$$

And the desired output line voltages are given by:

$$\begin{aligned} V_{AB} &= \sqrt{3}V_{om}\cos(w_o t + p/6) \\ V_{BC} &= \sqrt{3}V_{om}\cos(w_o t + p/6 - 2p/3) \\ V_{CA} &= \sqrt{3}V_{om}\cos(w_o t + p/6 - 4p/3) \end{aligned} \qquad \textbf{Equation 2-42}$$

In complex form, the space vector of the desired output line voltages is :

$$\begin{aligned} \overline{V_{oL}} &= V_{AB} + V_{BC}e^{j2p/3} + V_{CA}e^{-j2p/3} \\ \overline{V_{oL}} &= \frac{3}{2}kV_{om}\angle q_o \end{aligned} \qquad \textbf{Equation 2-43}$$

Where: $0 < k \le \sqrt{3}/2$ is the modulation index of the inverter stage.

Suppose the angle q_o is in the sector I, the vector reference $\overline{V_{oL}}$ can be approximated by two adjacent non-zero switching state vector voltages V_1 and V_2 as shown in Figure 2-35. A zero switching state vector voltage V_0 or V_7 is also added to improve the magnitude of the voltage vector.

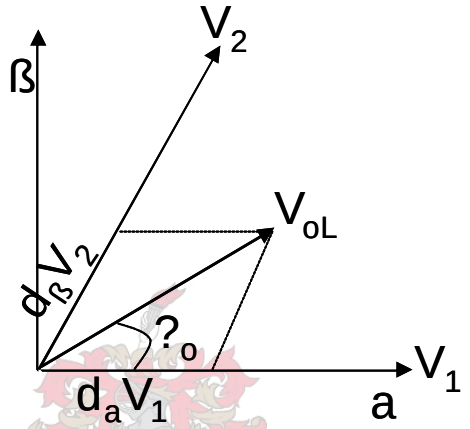


Figure 2-35 Duty cycle and adjacent vector voltage

The desired space vector output voltage $\overline{V_{oL}}$ rotating at an angle q_o approximated by the two non-zero active voltage vectors and the zero voltage vector is given by

$$\overline{V_{oL}} = d_\alpha V_1 + d_\beta V_2 + d_0 V_0 \qquad \textbf{Equation 2-44}$$

The corresponding duty cycles of the non-zero active state voltages and the zero state voltages are:

$$\begin{aligned} d_\alpha &= \frac{2}{\sqrt{3}} k \sin(\frac{p}{3} - q_o) \\ d_\beta &= k \sin(q_o) \\ d_0 &= 1 - d_\alpha - d_\beta \end{aligned} \qquad \textbf{Equation 2-45}$$

If the desired output space vector voltage is in sector I and the synchronization input phase angle is in interval I, the duty cycles are calculated as:

- During the first portion,

$$d_1 = d_\alpha * d_{ba}$$
$$d_2 = d_\beta * d_{ba}$$
$$d_0 = 1 - d_1 - d_2$$

Equation 2-46

- And during the second portion of the interval;

$$d_1 = d_\alpha * d_{ca}$$
$$d_2 = d_\beta * d_{ca}$$
$$d_0 = 1 - d_1 - d_2$$

Equation 2-47

The switching time corresponding to the calculated duty cycle is given as the product of the duty cycle and the switching period T_s. A time sequence of both rectifier side and inverter side switching is shown in Figure 2-36.

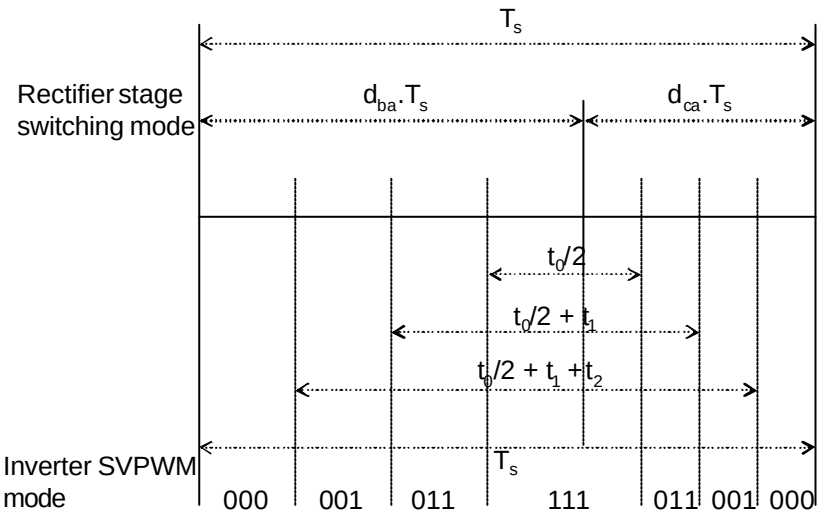


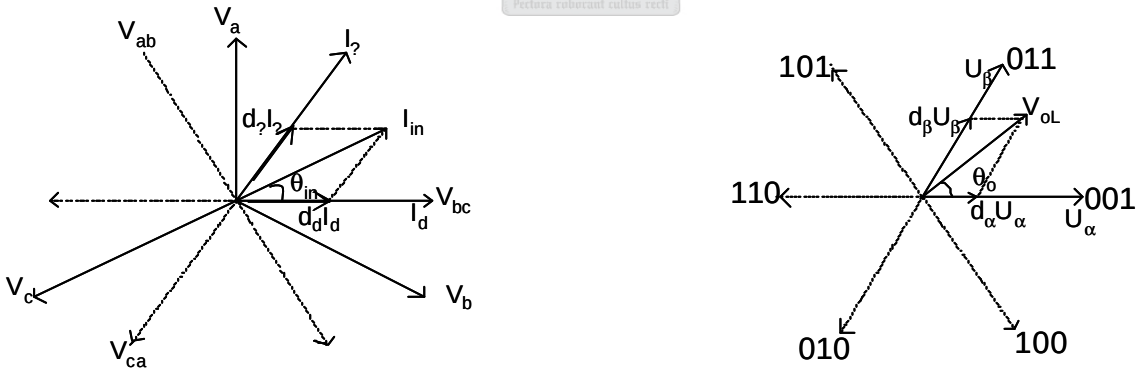
Figure 2-36 PWM sequence of the indirect matrix converter

All line side switches can be turned on or turned off at zero current and all load side switches can commutate similarly to a conventional DC/AC inverter. The line side switches commutate at both the beginning and end of each portion. If a zero vector is selected for the inverter on the load side at this time, then the line side switches can commutate at zero current [6]. Figure 2-36 shows the PWM sequence of the converter. It can be found that the load side VSI utilizes the zero vector, while the line switches are commutating. Hence, at this time instant, all currents on the rectifier side are zero so that zero current commutation can be guaranteed for the line side switches.

2.6.2.2 Modulation Technique Based on SVM and PFM

2.6.2.2.1 Space Vector Modulation (SVM)

Both the rectification stage and the inversion stage of the converter apply a space vector modulation [6]. An input current $\overline{I_{in}}$ that corresponds to the rectifier stage as shown in Figure 2-37(a) and an output voltage $\overline{V_{oL}}$ that corresponds to the inversion stage as shown in Figure 2-37(b) are the reference vectors.



(a) (b)

Figure 2-37 Generation of the reference vectors using SVM. a) Rectification stage; b) Inversion stage

Consider the input current $\overline{I_{in}}$ and the output voltage $\overline{V_{oL}}$ reference vectors are in sector I as shown in Figure 2-37. $\overline{I_{\delta}}$ and $\overline{I_{\gamma}}$ are defined as the active switching state vectors that approximate the input current reference vector $\overline{I_{in}}$ of the rectification stage as shown in Figure 2-37(a). While $\overline{V_{\alpha}}$ and $\overline{V_{\beta}}$ are defined as the active switching state vectors to represent the output voltage reference vector $\overline{V_{oL}}$ of the inversion stage as shown in Figure 2-37(b).The duty cycles that corresponds to these active switching states are calculated with Equation 2-48, and 2-49 for the rectifier stage, and with Equation 2-51 and 2-52 for the inversion stage.

For the rectification stage:

$$d_{\delta} = m_I \sin(\frac{p}{3} - q_{in}) \qquad \textbf{Equation 2-48}$$

$$d_{\gamma} = m_I \sin(q_{in}) \qquad \textbf{Equation 2-49}$$

$$d_0 = 1 - d_{\delta} - d_{\gamma} \qquad \textbf{Equation 2-50}$$

For the inversion stage:

$$d_{\alpha} = m_o \sin(\frac{p}{3} - q_o) \qquad \textbf{Equation 2-51}$$

$$d_{\beta} = m_o \sin(q_o) \qquad \textbf{Equation 2-52}$$

$$d_0 = 1 - d_{\alpha} - d_{\beta} \qquad \textbf{Equation 2-53}$$

Where: m_I and m_o are the rectification stage and inversion stage modulation indexes, and q_{in} and q_o are the angles with in the respective sectors of the input current and output voltage reference vectors.

These duty cycles given in Equation 2-48, 2-49, 2-51, and 2-52 are for a stand-alone rectifier as well as a stand-alone inverter, so it is difficult to get a correct balance of the input currents and output voltages. Therefore, a combined pattern of the two-stage duty cycles during a given switching period are used to generate balanced input currents and

output voltages. A combined pattern of the duty cycles is given by Equation 2-54 and 2-55.

$$\begin{aligned} d_{\alpha\delta} &= d_{\alpha} \cdot d_{\delta} \\ d_{\alpha\gamma} &= d_{\alpha} \cdot d_{\gamma} \\ d_{\beta\gamma} &= d_{\beta} \cdot d_{\gamma} \\ d_{\beta\delta} &= d_{\beta} \cdot d_{\delta} \end{aligned}$$ **Equation 2-54**

$$\begin{aligned} d_{0\delta} &= 1 - d_{\alpha\delta} - d_{\alpha\delta} \\ d_{0\gamma} &= 1 - d_{\alpha\gamma} - d_{\beta\gamma} \end{aligned}$$ **Equation 2-55**

Figure 2-38 shows a timing sequence of the combined pattern of the duty cycles of the rectification stage and inversion stage.

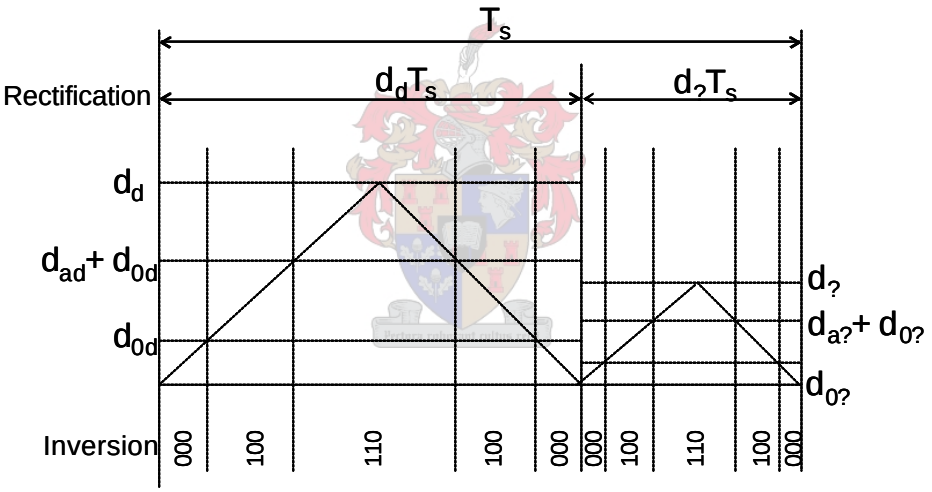


Figure 2-38 PWM sequence of the converter in sector I

2.6.2.2.2 The Novel Indirect Modulation (PFM) Technique

This modulation technique uses a stator flux as a vector reference $\bar{\psi}$ in the inversion stage, and is denoted as the polygonal flux modulation (PFM) [10]. The stator flux vector reference $\bar{\psi}$ is approximated by one active switching state vector and a corresponding zero vector, so that the stator flux error is minimized. If a single active vector is used to represent the input current vector reference in the rectification stage,

then only two sequences are in a switching period. However, the synthesis causes an error. Error in the inversion stage is compensated for due to the integrative nature of the stator flux reference given by Equation 2-56:

$$\overline{\psi_s}(N) = \overline{\psi_s}(N-1) + \int \overline{U_s} \cdot dt \qquad \textbf{Equation 2-56}$$

Where: $\overline{\psi_s}(N)$ and $\overline{\psi_s}(N-1)$ are the stator flux vectors in two successive switching periods, when the resistance is neglected and $\overline{U_s}$ is the stator voltage vector and $t = 0..T_s$ is the time variable.

The generation of the output voltage vector using the stator flux reference in the inversion stage based on the PFM is shown in Figure 2-39.

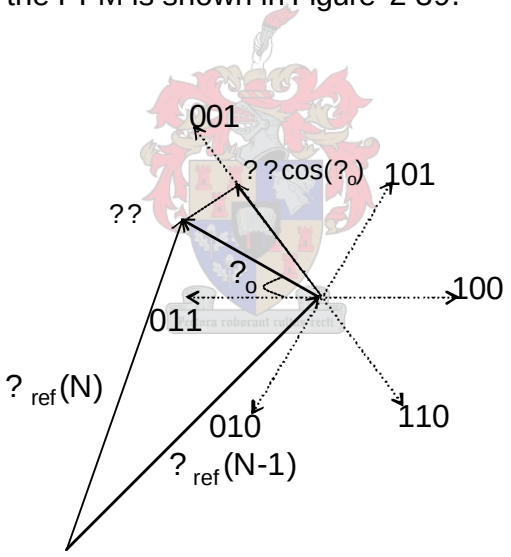


Figure 2-39 Generation of voltage vector reference using stator flux based on PFM

2.7 Sparse Matrix Converter Topology

This topology was developed based on the structure of the indirect matrix converter topology [6], [10]. Since the topology employs a reduced number of switches compared to the conventional direct matrix converter or indirect matrix converter topologies, it is denoted as a sparse matrix converter (SMC) topology [6-7]. All the important

characteristics of the indirect matrix converter topology are possibly provided by this topology. Furthermore, the sparse matrix converter topology can be reduced to a lower number of switches, and is called as the ultra sparse matrix converter (USMC) topology [6-7].

2.7.1 Derivation of Sparse Matrix Converter Topology

The rectifier stage circuit of the indirect matrix converter topology has a fixed DC-link polarity. However, the switches can operate with both DC-link polarities. Therefore, the number of switches can be reduced to 15 from 18 switches of the rectifier stage circuit of the indirect matrix converter topology as shown in Figure 2-40.

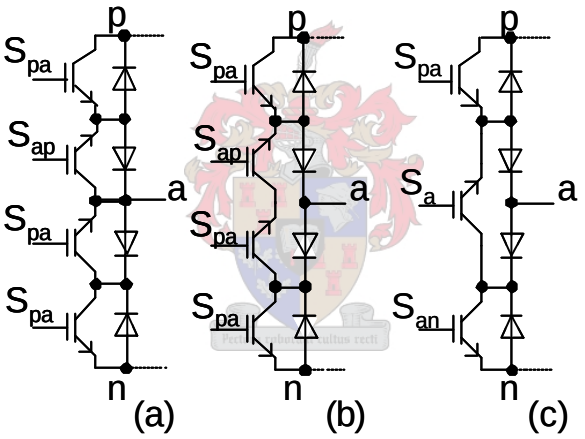


Figure 2-40 Derivation of sparse matrix converter topology a) rectifier stage circuit leg; b) modified rectifier stage circuit leg c) sparse matrix converter leg

The inverter stage of the sparse matrix converter is the same as the indirect matrix converter topology. The same method of modulation technique, which is used by the indirect matrix converter topology, can be applied to the three-phase sparse matrix converter topology. A three-phase sparse matrix converter topology is shown in Figure 2-41 .

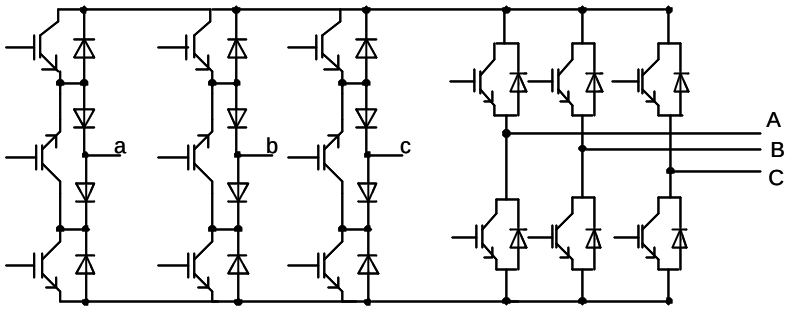


Figure 2-41 Sparse matrix converter topology (SMC)

Further reduction of the switches of the sparse matrix converter (SMC) topology to 9 switches is possible. The reduced topology is termed as the ultra sparse matrix converter (USMC) topology [7] as shown in Figure 2-42. There will only be a unidirectional power flow $(V_{pn} > 0, I_{pn} > 0)$, since the reduction of the topology is based on unidirectional power flow. The ability to control the power factor of the input voltage and input current is limited to $\pm p/6$. Besides, the maximum allowed output displacement factor is limited to $\pm p/6$. The sparse matrix converter (SMC) modulation technique used can be extended to control the ultra sparse matrix converter (USMC) topology.

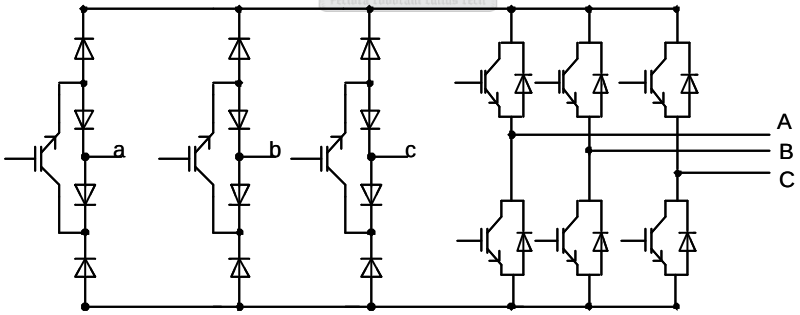


Figure 2-42 Ultra Sparse matrix converter topology

Alternative sparse matrix converter topologies are also proposed [7]. These are:

- Very sparse matrix converter topology
- Inverting link matrix converter topology

2.8 New Matrix Converter Topology

A new matrix converter topology, which employs 10 bidirectional switches, is proposed in [12] to increase the maximum gain of the converter to 1.15 as shown in Figure 2-43. This topology has 45 switching combinations as shown in Table 2-4, which obey the two basic rules: no short circuit of input line voltages and no open circuit of the output line current. The modulation technique used is based on the least mean square error (LMSE) [12]. The advantage of this matrix converter topology over the conventional direct matrix converter topology is that; it has a simplified control algorithm, minimum loss of the converter and the converter has 1.15 gain of the input voltage that can term it as an amplifier. In addition, the converter has low THD [12].

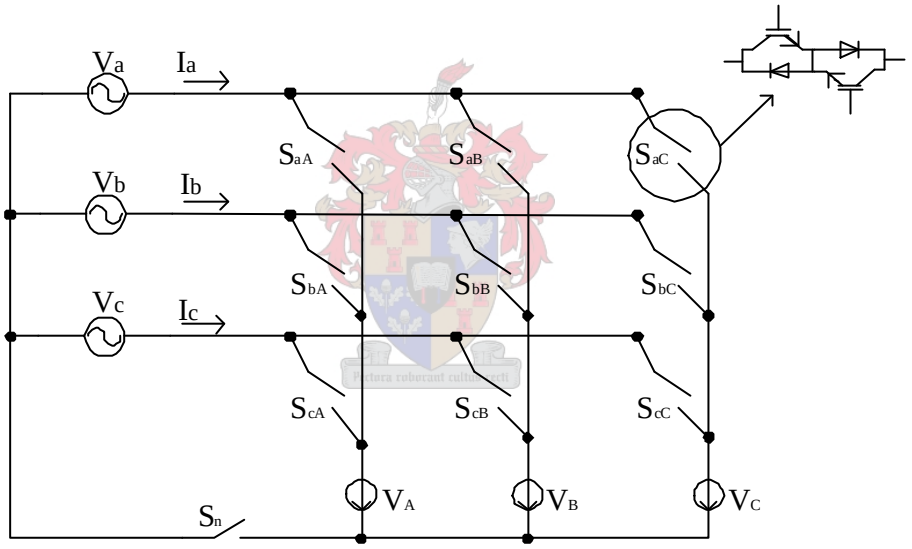


Figure 2-43 A New matrix converter topology of 10 bidirectional switches

Table 2-4 shows the switching combination modes of the converter. Only 45 possible switching combination modes obey the two basic rules: no short circuit and no open circuit.

Table 2-4 Switching combination modes of the new matrix converter topology

Modes	On switches				V_A	V_B	V_C	I_a	I_b	I_c
1	S_{aA}	S_{bB}	S_{cC}	S_n	V_a	V_b	V_c	I_A	I_B	I_C
2	S_{aB}	S_{bC}	S_{cA}	S_n	V_c	V_a	V_b	I_B	I_C	I_A
3	S_{aC}	S_{bA}	S_{cB}	S_n	V_b	V_c	V_a	I_C	I_A	I_B
4	S_{aC}	S_{bB}	S_{cA}	S_n	V_c	V_b	V_a	I_C	I_B	I_A
5	S_{aA}	S_{bC}	S_{cB}	S_n	V_a	V_c	V_b	I_A	I_C	I_B
6	S_{aB}	S_{bA}	S_{cC}	S_n	V_b	V_a	V_c	I_A	I_B	I_C
7	S_{aA}	S_{bB}	S_{bC}	S_n	V_a	V_b	V_b	I_A	$I_B + I_C$	0
8	S_{bA}	S_{cB}	S_{cC}	S_n	V_b	V_c	V_c	0	I_A	$I_B + I_C$
9	S_{aB}	S_{aC}	S_{cA}	S_n	V_c	V_a	V_a	$I_B + I_C$	0	I_A
10	S_{aB}	S_{aC}	S_{bA}	S_n	V_b	V_a	V_a	$I_B + I_C$	I_A	0
11	S_{bB}	S_{bC}	S_{cA}	S_n	V_c	V_b	V_b	0	$I_B + I_C$	I_A
12	S_{aA}	S_{cB}	S_{cC}	S_n	V_a	V_c	V_c	I_A	0	$I_B + I_C$
13	S_{aB}	S_{bA}	S_{bC}	S_n	V_b	V_a	V_b	I_B	$I_A + I_C$	0
14	S_{bB}	S_{cA}	S_{cC}	S_n	V_c	V_b	V_c	0	I_B	$I_A + I_C$
15	S_{aA}	S_{aC}	S_{cB}	S_n	V_a	V_c	V_a	$I_A + I_C$	0	I_A
16	S_{aA}	S_{aC}	S_{bB}	S_n	V_a	V_b	V_a	$I_A + I_C$	I_B	0
17	S_{bA}	S_{bC}	S_{cB}	S_n	V_b	V_c	V_b	0	$I_A + I_C$	I_B
18	S_{aB}	S_{cA}	S_{cC}	S_n	V_c	V_a	V_c	I_B	0	$I_A + I_C$
19	S_{aC}	S_{bA}	S_{bB}	S_n	V_b	V_b	V_a	I_C	$I_A + I_B$	0
20	S_{bC}	S_{cA}	S_{cB}	S_n	V_c	V_c	V_b	0	I_C	$I_A + I_B$
21	S_{aA}	S_{aB}	S_{cC}	S_n	V_a	V_a	V_c	$I_A + I_B$	0	I_C
22	S_{aA}	S_{aB}	S_{bC}	S_n	V_a	V_a	V_b	$I_A + I_B$	I_C	0
23	S_{bA}	S_{bB}	S_{cC}	S_n	V_b	V_b	V_c	0	$I_A + I_B$	I_C
24	S_{aC}	S_{cA}	S_{cB}	S_n	V_c	V_c	V_a	I_C	0	$I_A + I_B$
25	S_{aA}	S_{aB}	S_{aC}	S_n	V_a	V_a	V_a	I_A	$I_B + I_C$	0
26	S_{bA}	S_{bB}	S_{bC}	S_n	V_b	V_b	V_b	0	I_A	$I_B + I_C$
27	S_{cA}	S_{cB}	S_{cC}	S_n	V_c	V_c	V_c	$I_B + I_C$	0	I_A
28	S_{aA}	S_{bB}	S_{cC}		$\frac{2}{3}(V_a - V_b)$	$\frac{1}{3}(V_b - V_a)$	$\frac{1}{3}(V_a - V_b)$	I_A	$I_B + I_C$	0

29	S_{bA}	S_{cB}	S_{cC}	$\frac{2}{3}(V_b-V_c)$	$\frac{1}{3}(V_c-V_b)$	$\frac{1}{3}(V_c-V_b)$	0	I_A	$I_B + I_C$
30	S_{aB}	S_{aC}	S_{cA}	$\frac{2}{3}(V_c-V_a)$	$\frac{1}{3}(V_a-V_c)$	$\frac{1}{3}(V_a-V_c)$	$I_B + I_C$	0	I_A
31	S_{aB}	S_{aC}	S_{bA}	$\frac{2}{3}(V_b-V_a)$	$\frac{1}{3}(V_a-V_b)$	$\frac{1}{3}(V_a-V_b)$	$I_B + I_C$	I_A	0
32	S_{bB}	S_{bC}	S_{cA}	$\frac{2}{3}(V_c-V_b)$	$\frac{1}{3}(V_b-V_c)$	$\frac{1}{3}(V_b-V_c)$	0	$I_B + I_C$	I_A
33	S_{aA}	S_{cB}	S_{cC}	$\frac{2}{3}(V_a-V_c)$	$\frac{1}{3}(V_c-V_a)$	$\frac{1}{3}(V_c-V_a)$	I_A	0	$I_B + I_C$
34	S_{aB}	S_{bA}	S_{bC}	$\frac{1}{3}(V_b-V_a)$	$\frac{2}{3}(V_a-V_b)$	$\frac{1}{3}(V_c-V_a)$	I_B	$I_A + I_C$	0
35	S_{bB}	S_{cA}	S_{cC}	$\frac{1}{3}(V_c-V_b)$	$\frac{2}{3}(V_b-V_c)$	$\frac{1}{3}(V_c-V_b)$	0	I_B	$I_A + I_C$
36	S_{aA}	S_{aC}	S_{cB}	$\frac{1}{3}(V_a-V_c)$	$\frac{2}{3}(V_c-V_a)$	$\frac{1}{3}(V_a-V_c)$	$I_A + I_C$	0	I_A
37	S_{aA}	S_{aC}	S_{bB}	$\frac{1}{3}(V_a-V_b)$	$\frac{2}{3}(V_b-V_c)$	$\frac{1}{3}(V_a-V_b)$	$I_A + I_C$	I_B	0
38	S_{bA}	S_{bC}	S_{cB}	$\frac{1}{3}(V_b-V_c)$	$\frac{2}{3}(V_c-V_b)$	$\frac{1}{3}(V_b-V_c)$	0	$I_A + I_C$	I_B
39	S_{aB}	S_{cA}	S_{cC}	$\frac{1}{3}(V_c-V_a)$	$\frac{2}{3}(V_a-V_c)$	$\frac{1}{3}(V_c-V_a)$	I_B	0	$I_A + I_C$
40	S_{aC}	S_{bA}	S_{bB}	$\frac{1}{3}(V_b-V_a)$	$\frac{1}{3}(V_b-V_a)$	$\frac{2}{3}(V_a-V_c)$	I_C	$I_A + I_B$	0
41	S_{bC}	S_{cA}	S_{cB}	$\frac{1}{3}(V_c-V_b)$	$\frac{1}{3}(V_c-V_b)$	$\frac{2}{3}(V_b-V_c)$	0	I_C	$I_A + I_B$
42	S_{aA}	S_{aB}	S_{cC}	$\frac{1}{3}(V_a-V_c)$	$\frac{1}{3}(V_a-V_c)$	$\frac{2}{3}(V_c-V_a)$	$I_A + I_B$	0	I_C
43	S_{aA}	S_{aB}	S_{bC}	$\frac{1}{3}(V_a-V_b)$	$\frac{1}{3}(V_a-V_b)$	$\frac{2}{3}(V_b-V_a)$	$I_A + I_B$	I_C	0
44	S_{bA}	S_{bB}	S_{cC}	$\frac{1}{3}(V_b-V_c)$	$\frac{1}{3}(V_b-V_c)$	$\frac{2}{3}(V_c-V_b)$	0	$I_A + I_B$	I_C
45	S_{aC}	S_{cA}	S_{cB}	$\frac{1}{3}(V_c-V_a)$	$\frac{1}{3}(V_c-V_a)$	$\frac{2}{3}(V_a-V_c)$	I_C	0	$I_A + I_B$

2.9 Summary

This chapter gave the background of the various different topologies of the matrix converter and a brief description of the conventional standard AC-DC-AC inverter based converter topology. Furthermore, it briefly touched on the construction of the basic building blocks of the converter (bidirectional switch). The complexity of the PWM controlling mechanism and the difficulty involved in commutation has been described. It has been showed that the output-input transfer ratio can be improved by adding the third harmonic components of the input and desired output voltages to the components of the desired output voltage from 50% to 86.6%. The three proposed safe commutation strategies of the direct matrix converter based on current and voltage commutation strategies were presented.

The derivation of the remaining topologies from the direct matrix converter was shown. The indirect matrix converter topology circuit structure and the controlling methods such as, relative input voltage method of the input and SVPWM method in the output side [6], space vector PWM control of the input and output [10], as well as SVPWM in the input and the novel indirect modulation (PFM) [10] were investigated. It was described that this converter overcomes the zero current commutation problem of the direct matrix converter. The subsequent subsections presented the reduction of the indirect matrix converter into the sparse and ultra spares matrix converter [7]. Finally, the amplifier type topology of the matrix converter that applied LMSE to generate 115% transfer ratio of the input was presented. In addition, the 45 possible switching combination modes for the 10 bi-directional switches were also shown.

It was proved that theoretically a small size, capable of energy regenerating back to the utility, capable of generating sinusoidal input and output current, able to produce controllable input power factor, independent of the connected load, and cheap converter construction is possible. This converter is applicable especially for AC drives in industrial application.

CHAPTER 3

DESIGN & ANALYSIS OF DIRECT MATRIX CONVERTER

3 DESIGN & ANALYSIS OF DIRECT MATRIX CONVERTER

3.1 Introduction

The previous chapter discussed the different matrix converter topologies and their modulation techniques. In this chapter, the discussion will concentrate on the analysis and design of the direct matrix converter topology. The design of the direct matrix converter topology will examine the design of the small, building-block parts of the whole converter system. An overview of the designed direct matrix converter printed circuit board and a block diagram to be analyzed and subsequently designed is shown in Figure 3-1 and Figure 3-2. This converter system is required to generate a maximum output voltage of 200V/(0-200Hz) when connected to a three-phase supply of 400V/50Hz.

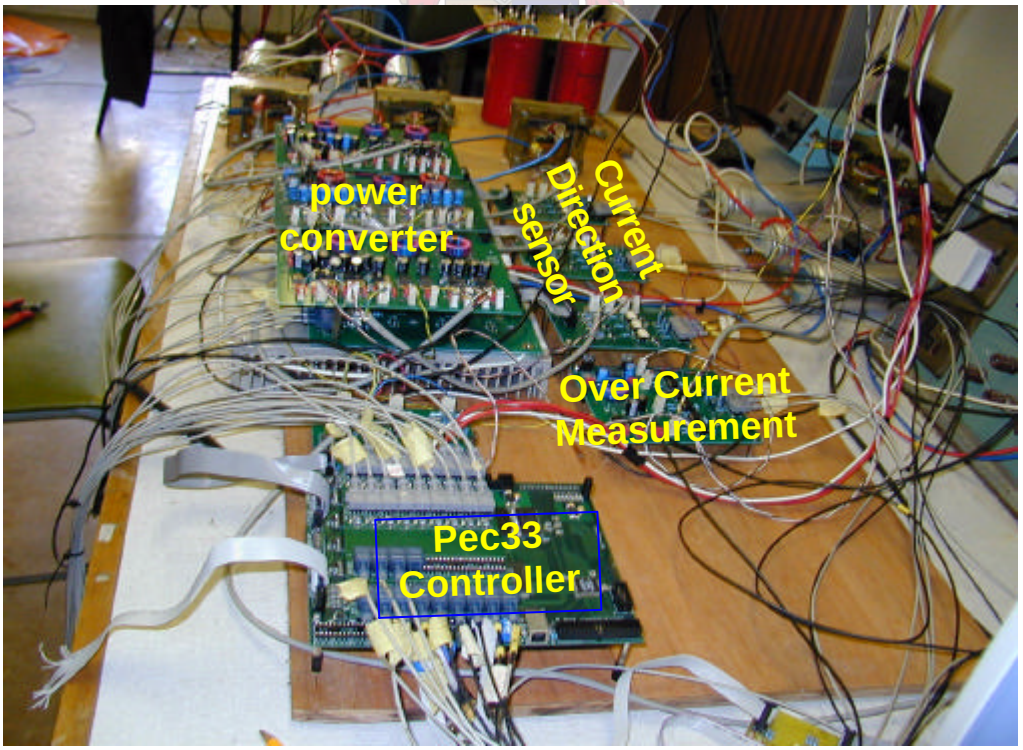


Figure 3-1 A printed circuit board of the matrix converter

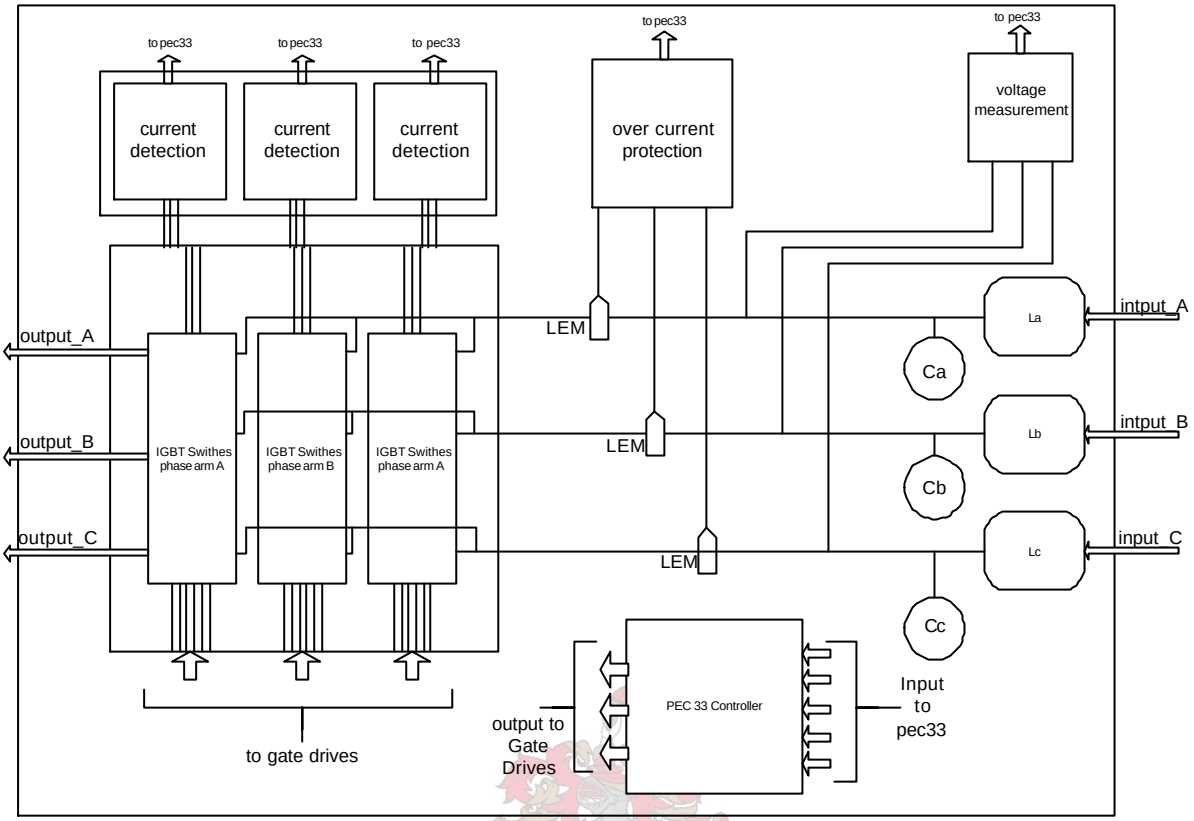


Figure 3-2 Block diagram of the direct matrix converter building components

The system consists of the following parts:

- A small sized input filter to reduce the ripple of the input current;
- A gate drive circuit that drives the IGBT switch of the power converter;
- A main power converter where the actual switching operation is performed;
- A current direction detection circuit which is used to determine the direction of the load current;
- An input voltage measurement to measure the three-phase input voltages;
- A clamp circuit that will protect the power converter switches from destruction when a fault occurs;
- An over-current sensor circuit to detect a short circuit and instantaneous increase of the input current to a dangerous level that might destroy the power converter switches;

- A fiber optic circuit, which is needed for transferring control signals between the PEC33 controller board and the direct matrix converter circuit. Table 3-1 shows the specification of the direct matrix converter circuit designed in this chapter.

Table 3-1 Specification of the direct matrix converter circuit

Symbol	Parameter	Value	Unit
$3\varnothing$ V_i	Input Voltage	400	V
f_i	Input frequency	50	Hz
$3\varnothing$ V_o	Output Voltage	0..200	V
f_o	Output frequency	0..200	Hz
P	Power rating	2.25	kW
q	Modulation index	0..0.5	No unit

3.2 Input/Output Filter Circuit Design & Analysis

One of the important desirable characteristics of the direct matrix converter topology mentioned in chapter two is the design of a small power converter. Therefore, the matrix converter building-block parts have to be designed small enough to ensure the small size and low cost of the converter circuit. The input filter circuit, which is designed to reduce the input ripples with minimum energy on the reactive element, will be discussed in this section. The output filter circuit designed is required only for experimental test purposes, since a resistive load is connected to the output terminal of the converter circuit. Several circuit topologies are recommended for this purpose except that they are complex topologies. The simplest form of filter topology, which can be designed to fit the requirement of the matrix converter, is an LC series circuit [13]. The LC series circuit is easy to design and provides all the necessary characters required from a filter circuit. The designed input filter shown in Figure 3-3 is required to provide the following functions:

1. To produce an input filter with a cut-off frequency lower than the switching frequency.

2. To maximize the displacement power factor $\cos(q)$ for a given minimum output power $P_{\min}$ [13-14]
3. To reduce the input filter volume or weight for a given reactive power, by taking into account different energy densities for capacitors.
4. To minimize the voltage drop on the filter inductance at the rated current in order to provide the highest voltage transfer ratio.

3.2.1 Design of Input Filter Circuit

The design of the filter is done based on the following two criteria:

- The cut-off frequency of the input filter w is chosen to provide certain attenuation at the switching frequency.
- The value of the capacitor and/or inductor is chosen based on previous criteria; the other component, therefore, has to satisfy Equation 3-1.

$$L_{in} \cdot C_{in} = \frac{1}{w_i^2} \qquad \textbf{Equation 3-1}$$

The inductance value should verify the maximum voltage drop condition at full load as proposed in [13].

$$\frac{\Delta V}{U_n} = \sqrt{1-(w_i \cdot L_{in})^2 \left(\frac{I_n}{U_n}\right)^2} = \sqrt{1-l_{in}^{\ 2}} \qquad \textbf{Equation 3-2}$$

Where ΔV is the maximum voltage drop at full load, w_i is the grid frequency, C_{in} and L_{in} are the input filter Capacitance and inductance, I_n and U_n are the rated input phase current and voltage, and l_{in} is the filter inductance in per unit.

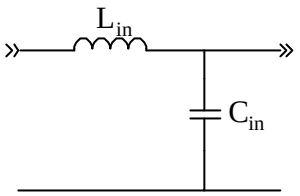


Figure 3-3 Input filter circuit

3.2.2 Design of Output Filter Circuit

The design of the output filter circuit is not necessary; it is explained in here only because a resistive load is connected to the output terminal of the matrix converter circuit for experimental testing purposes. The direct matrix converter is designed to drive an induction motor. Hence, the induction motor has an internal inductance that can be used as a filter to the unfiltered output of the converter. Therefore, an output filter circuit design is redundant to the converter circuit. Though, the same approach can be applied to design the output filter for the output resistive load. A simple series LC filter circuit is shown in Figure 3-4.

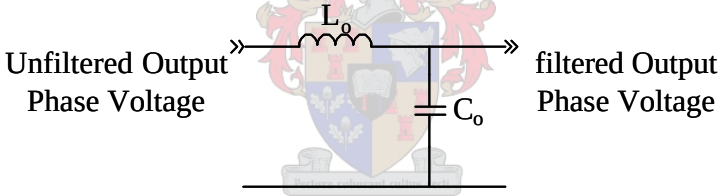


Figure 3-4 Output filter circuit for the resistive load

3.3 Main Power Converter Circuit Construction

Three possible arrangements of bi-directional switch cell construction were discussed in the previous chapter of this thesis. Their advantages and disadvantages have also been clearly mentioned. Since the Common Emitter and Common Collector arrangement allows full control of current direction through the bi-directional switch at all times, the power loss in the switching device is less. Either of these arrangements can be used for the construction of the main power converter circuit of the matrix converter topology. Since the Common Emitter arrangement has the following advantages over the Common Collector; access to all IGBT terminals is possible and suitable for V_{CE}

monitoring. If the bi-directional switches are integrated in a 3∅/1∅ power module no external connection to other modules is necessary and the gate driver may be integrated with the power module. It is possible to implement an over-current protection and adaptive switching techniques by using the V_{CE} [15] as shown in Figure 3-5. The power converter switches are designed to drive an induction motor with a maximum input power of 2.25 kW. The IGBT switching device selected for constructing the power converter of the direct matrix converter circuit is required to handle this input power. The selection depends on the maximum input voltage applied to the terminal of the switching device and the maximum output current that passes through the switching device. Since the switching device will be connected to 400V input supply, the maximum V_{CE} of the switching device has to be higher than the supply input voltage. The maximum output voltage generated by the converter is 50% of the mains supply input voltage as proposed by Venturini [5]. Therefore, the output current can be calculated as shown in Equation 3-3.

$$I_o = \frac{P_o}{V_o \cdot \cos(q_o)}$$ **Equation 3-3**

Where: V_o 200V is the peak output voltage, I_o is the peak output current, and the output power factor is unity.

$$I_o = \sqrt{2} \cdot \frac{2250\ W/3}{200\ V/\sqrt{2}} = 7.5\ A$$ **Equation 3-4**

The collector current I_C of the switching device has to be greater 7.5 A.

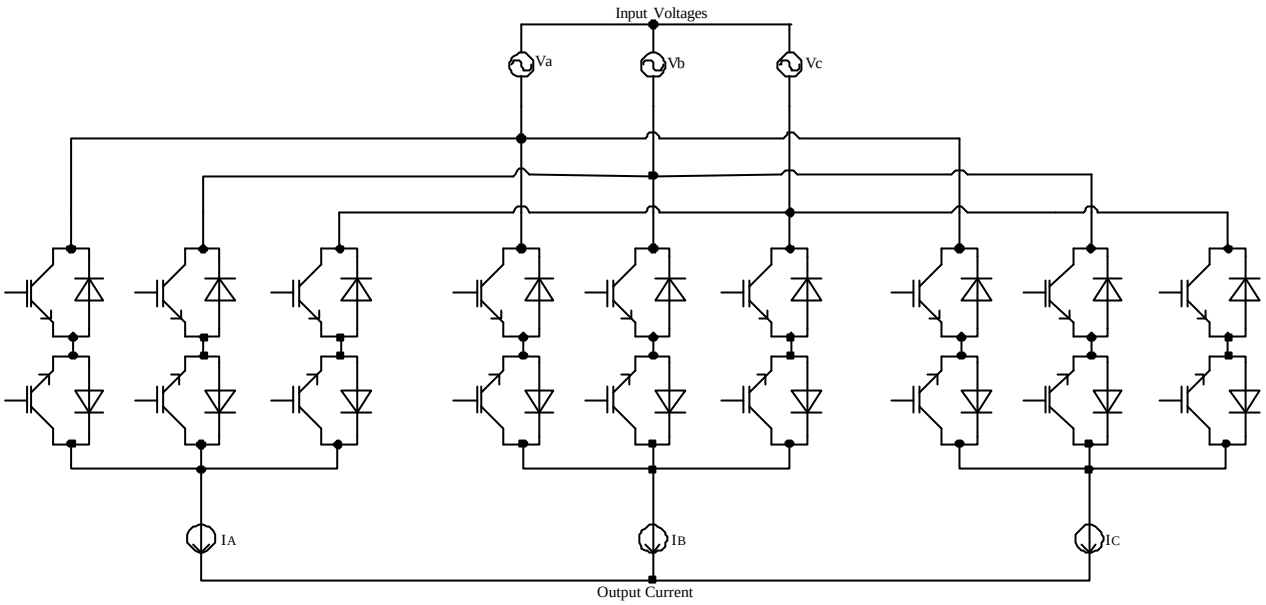


Figure 3-5 The power converter section circuit diagram of the direct matrix converter.

AN IGBT switching device with higher I_C than the calculated output current I_O and higher maximum V_{CE} voltage than the supply input voltage V_{in} is selected to construct the power converter part of the direct matrix converter circuit. IRG4PH40KD IGBT device rated with typical collector current of I_C = 15A and maximum V_{CE} voltage of 1200V satisfies the desired requirements.

Nine bi-directional switch cells are used to build a three-phase direct matrix converter circuit as shown in Figure 3-5. Each of the phases consists of three bi-directional switch cells that connect the three-phase input lines to a single-phase output. Therefore, 18 IRG4PH40KD IGBT switches are required. Detailed information of the IRG4PH40KD IGBT switching device is given in the datasheet specification [20]. Table 3-2 shows some important parameters of the IRG4PH40KD switching device.

Table 3-2 Absolute maximum rating of the IRG4PH40KD IGBT switch

	Parameters	Maximum	Units
V_{CES}	Collector to emitter Voltage	1200	V
V_{CE}(on) @ V_{GE}=15V,I_c=15A	Collector to emitter Voltage	2.74	V
T_{SC}	Short circuit withstand time	10	µs
I_C @ T= 25 °C	Continues collector current	30	A
I_C @ T= 100 °C	Continues collector current	15	A
V_{GE}	Gate to emitter voltage	±20	V
t_d(on)	Turn-on delay time	49	ns
t_d(off)	Turn-off delay time	290	ns
t_t	Total delay time	730	ns

3.4 Gate Drive Circuit Design

The analysis and design of a gate-drive circuit will be examined in this section. The function of the gate-driver circuit is to generate high current output to drive the IGBT switches of the main power converter circuit. An hcpl-314j chip with 0.4 A maximum output current is select to generate the high current output that drives the IGBT device. This gate-drive chip consists of two isolated gate-driving circuits as shown in Figure 3-6. It requires two +15V isolated power supplies connected across the two isolated gate drivers, so that high current output of 0.4 A is generated. A single gate-drive chip drives two IGBT switches as shown in Figure 3-6, therefore 9 HCPL-314J gate-drive chips are required to drive the 9 bi-directional switch cells of the direct matrix converter circuit.

The output of the gate-drive is connected to the gate terminal of the IGBT via a gate resistor R_g. This resistor will be calculated later on this section. The IGBT and R_g in Figure 3-6 can be analyzed as a simple RC circuit with a voltage supplied by the HCPL-314J gate drive chip.

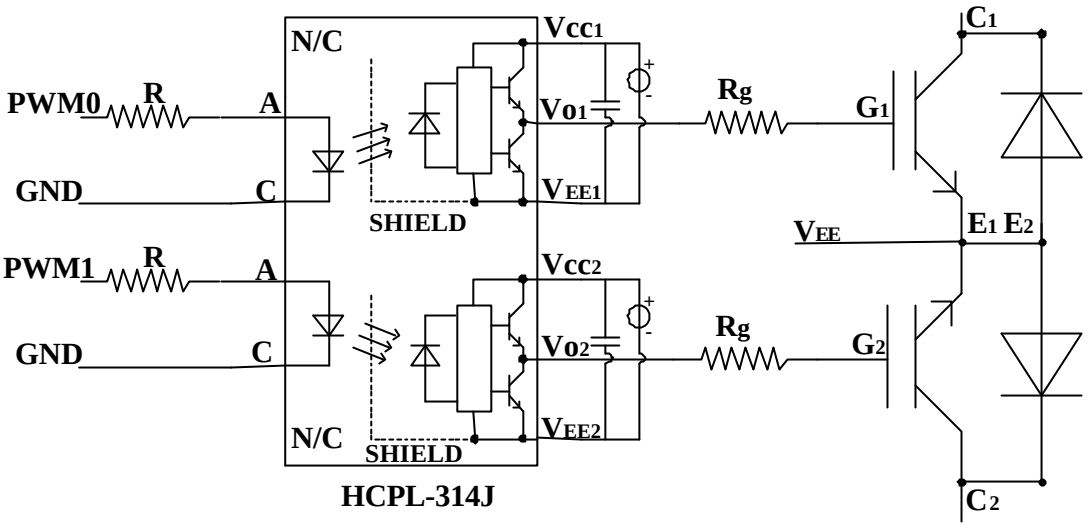


Figure 3-6 The gate drive circuit of the converter system

The value of R is taken directly from the HCPL-314J chip datasheet specification [19], but it can easily be calculated to confirm its validity with the given design parameters in the datasheet specification. The function of this resistance is to limit the input current I_F to the gate drive chip. The voltage of the PWM0 connected to the resistance can be either 0 or 5 V. The maximum voltage drop V_F and maximum current I_F of the diode of the gate drive chip is 1.8 V and 20 mA respectively as given in the datasheet specification [21] of the HCPL-314J.

Therefore, the resistance R is calculated as:

$$R = \frac{5\ V - V_F}{I_{F(MAX)}}$$
$$= \frac{5\ V - 1.8\ V}{20\ mA}$$
$$= 160\ \Omega$$

Equation 3-5

$R = 160\ \Omega$ is the minimum resistance value that has to be used, so that the driver's input current I_F will not exceed the limiting $I_{F(MAX)}$ and subsequently wouldn't damage the input diode of the gate drive chip.

Calculating the minimum R_g:

The IGBT and R_g shown in Figure 3-6 can be analyzed as a simple RC circuit with a voltage supplied by the HCPL-314J gate drive chip [21]. The minimum resistor R_g is then calculated as:

$$\begin{aligned} R_g >= & \frac{V_{CC} - V_{OL}}{I_{OL(PEAK)}} \\ = & \frac{15\ V - 5\ V}{0.6\ A} \\ = & 16.667\ \Omega \end{aligned}$$

Equation 3-6

Where: V_{OL} is the output voltage of the HCPL-314J gate drive and V_{OL} = 5 V is given at the peak current $I_{OL(PEAK)}$ of 0.6 A.

The power dissipation of the gate drive HCPL-314J chip calculated at the minimum calculated value of R_g has to be smaller than the given power dissipation value of the gate drive chip in the datasheet. Otherwise, the value of R_g has to be increased until the calculated power dissipation is lower than the given power dissipation of the gate drive chip. The total power dissipation of the chip is obtained as the sum of the emitter power (P_E) and the output power (P_O).

$$P_T = P_O + P_E$$

Equation 3-7

Where P_E is the power dissipated at the input side of the chip is given by Equation 3-8.

$$P_E = I_F \cdot V_F \cdot Duty\ Cycle$$

Equation 3-8

Where I_F and V_F are the forward input current and forward input voltage to the HCPL-314J respectively.

And P_O is given as

$$\begin{aligned} P_O &= P_O(bias) + P_O(switching) \\ &= I_{CC} \cdot V_{CC} + E_{SW} \cdot (R_g \cdot Q_g) \cdot f_s \\ &= (I_{CC(BIAS)} + K_{ICC} \cdot Q_g \cdot f_s) \cdot V_{CC} + E_{SW} \cdot (R_g \cdot Q_g) \cdot f_s \end{aligned}$$ **Equation 3-9**

Where $K_{ICC} \cdot Q_g \cdot f_s$ is the increase of the current I_{CC} due to switching, and K_{ICC} is a constant of $0.001\ mA/(nC \cdot KHz)$ given in the datasheet [21]. For the circuit in Figure 3-6 with $I_F(worst\ case) = 10\ mA$, $R_g = 16.667\ \Omega$, Max duty cycle of 80%, $Q_g = 100\ nC$, $f_s = 5\ KHz$ and $T_{A(MAX)} = 85\ ^oC$, P_O is calculated as given in Equation 3-11 .

The emitter power P_E and the output power P_O calculated values are given in Equation 3-10 and 3-11 respectively.

$$\begin{aligned} P_E &= 10\ mA \cdot\ 1.8\ V \cdot\ 0.8 \\ &= 14\ mW \end{aligned}$$ **Equation 3-10**

$$\begin{aligned} P_O &= (3\ mA + (0.001\ mA/(nC \cdot KHz) \square 5\ KHz \square 100\ nC)) \square 15\ V + 0.4\ mJ \square 5\ KHz \\ &= 54.5\ mW \end{aligned}$$ **Equation 3-11**

The total power is given as in Equation 3-12:

$$\begin{aligned} P_T &= P_O + P_E \\ &= 68.5\ mW\ <\ 260\ mW\ \ (P_{O(MAX)}\ @\ 85\ ^oC) \end{aligned}$$ **Equation 3-12**

The value of $3mA$ for I_{CC} in the previous Equation 3-11 is the max I_{CC} over entire operating temperature range. Since P_O for this case is less than $P_{O(MAX)}$, $R_g = 16.667\ \Omega$ is within the limits for the power dissipation.

The maximum resistance value of R_g is calculated also as the ratio of the time constant t of the gate driver and the capacitance C_{ge} between gate and emitter of the gate driver as shown in equation 3-7.

$$t = R_g \cdot C_{ge} \qquad \textbf{Equation 3-13}$$

The time constant of the gate driver circuit has to be lower than the sum of the turn-on delay time and rise time of the HCPL-314J chip as shown in Table 3-3.

$$\begin{aligned} t &< t_d + t_r \\ &< 50\ ns + 31\ ns \\ &< 81\ ns \end{aligned} \qquad \textbf{Equation 3-14}$$

Equation 3-13 can be rewrite as given in Equation 3-15:

$$\begin{aligned} R_g \cdot C_{ge} &< 81ns \\ R_g &< \frac{81\ ns}{1.467\ nF} \\ &< 55.23\ \Omega \end{aligned} \qquad \textbf{Equation 3-15}$$

Therefore, a value of the resistor that falls in the range between $R_g = 16.667\ \Omega$ and $R_g = 55.23\ \Omega$ is selected.

Table 3-3 Specification parameter of IRG4PH40KD IGBT device

Parameters	Symbol	Values
Turn-on Delay time (turn-on)	t_d (on)	$50\ ns$
Rise time	t_r	$31\ ns$
Gate-Emitter Charge (turn-on)	Q_{ge} (max)	$22\ nC$
Gate-Emitter Voltage	V_{GE}	$15\ V$
Gate-Emitter Capacitance	$C_{ge} = \frac{Q_{ge}}{V_{GE}} = \frac{22\ nC}{15\ V}$	$1.467\ nF$

3.5 Current Direction Detection Circuit Design

This section of the chapter examines the analysis and design of the load current direction detection circuits. The direction of the load current can be detected either by placing a LEM module (transducer sensor) [25] at the output side of the matrix converter circuit or just measuring the voltage across the bi-directional switch cell [16]. The later method of load current direction detection is more reliable than the LEM module method, since the LEM module introduces a phase shift to the actual voltage measured due to its inherited inductive characteristic. But, the second method of detection circuit consists of only resistive and active elements. A detailed analysis and design of this detection circuit will be discussion as follows.

The bi-directional switch cell discussed in the previous chapter used for constructing the power converter system is shown again in Figure 3-7. It contains two IGBTs each with an anti-parallel diode, connected in series. This arrangement allows the direction of current within the switch to be controlled. Assuming I_L is in the direction shown; IGBT1 and D_2 will be conducting. The voltage drop across IGBT1 will cause V_1 to be about 2.75 V depending on the type of the devices used as shown in the datasheet of the IRG4PH40KD IGBT [20] and V_2 will be about -1 V because of the voltage drop across D_2. If the load current is in the opposite direction, the reverse situation exists. IGBT2 and D_1 will be conducting and V_1 and V_2 will be -1 V and 2.75 V respectively. A simple comparator circuit looking at the voltages V_1 and V_2 can therefore be used to determine the direction of the current through the bidirectional switch cell.

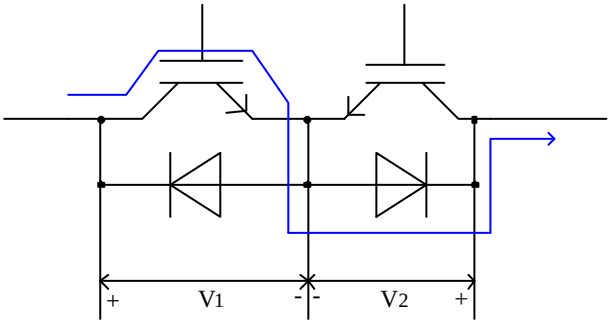


Figure 3-7 Bi-directional switch cell voltage drops

Figure 3-9 shows a current direction detection circuit proposed in [16], which is designed to measure the voltage across the bi-directional switch device and determine the load current direction. This can be implemented using a simple comparator circuit. However, the comparator circuit has to be protected from over-voltages since the switch device that is being measured sees the full mains input voltages while the switches are switched off. The over-voltage problem is protected by using a circuit shown in Figure 3-8. The circuit contains a resistor and a BYV26E diode connected in series. The time constant of the BYV26E diode is 250 X 10^3 s [27]. And a 5 V DC supply is connected to the negative terminal of the diode as shown in the Figure 3-8. The function of this circuit is to reduce the measured voltage whenever the bi-directional switch cell is switched off, otherwise the measured voltage will get high enough to destroy the comparator. The input to the current direction detection circuit is clamped by the diode to a 5V plus one diode voltage drop. The whole mains input voltage is then dropped across the resistor. The resistor has to be large enough to keep the power dissipation to a minimum. This is only possible if a high quality comparator is used in the voltage detection circuit and since this will take a negligible input current, there is very little voltage drop across the resistor when the switch cell is gated.

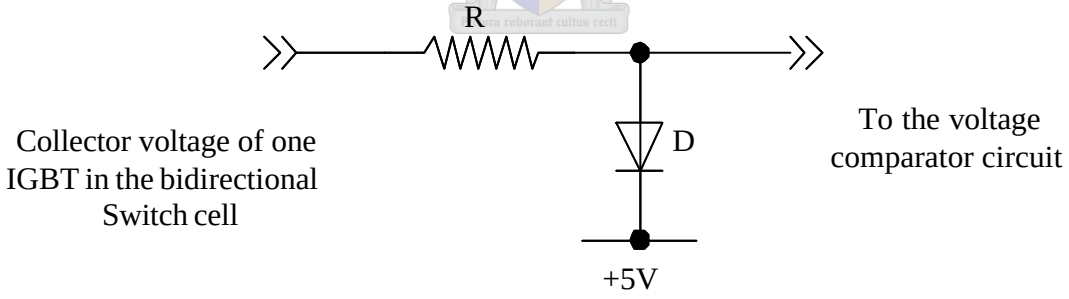


Figure 3-8 Over-voltage protection circuit

Figure 3-9 shows the current direction detection circuit. It consists of the protection circuit mentioned, a comparator chip, an opto-coupler to isolate the three phase bi-directional switch cells and a fiber optic transmitter circuit. The resistance R_1 used in the protection circuit is 100 KΩ, so that the input current to the comparator chip will be as small as possible. TL3016ID ultra fast comparator chip is used to compare the voltage

drop V_1 and V_2 across the bidirectional switch cell as shown in Figure 3-9. TL3016ID is a Texas Instrument which is an ultra fast comparator designed to interface to TTL logic while operating from either a single 5V power supply or ±5V supplies. The output of the comparator is connected to the opto-coupler through a limiting resistor R. The purpose of the limiting resistor R is to limit the current impressed to the input of the opto-coupler chip to 40 mA maximum as given in the datasheet of the 6N137 opto-coupler. 6N137 opto-coupler of super-high speed is used during the circuit design since the use of a fast response from each component ensures a safe commutation process between the converter switches. The 6N137 opto-coupler chip operates from a single 5 V power supply and a rise or fall delay time of 45 ns is possible if a resistor R_L of 250 ohms is used.

The value of the limiting resistor R connecting the output of the TL3016ID comparator to the input of the opto-coupler 6N137 is calculated as in equation 3-13.

$$R = \frac{V_{CC} - V_{F(MAX)}}{I_F} \qquad \textbf{Equation 3-13}$$

Where the forward voltage $V_{F(MAX)}$ = 1.75 V is given for a forward current I_F = 10mA as given in the datasheet specification of the 6N137 opto-coupler chip [26]. However, the rated forward current is 40 mA.

$$R = \frac{5\ V - 1.75\ V}{40\ mA} = 81.25\ \Omega \qquad \textbf{Equation 3-14}$$

The minimum value of the resistance R has to be used for the opto-coupler circuit is 81.25 Ω.

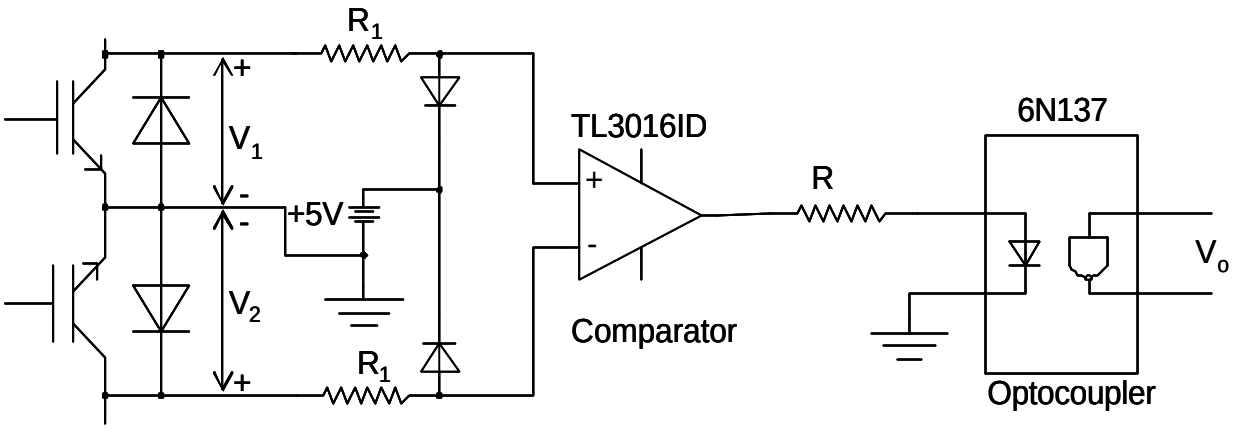
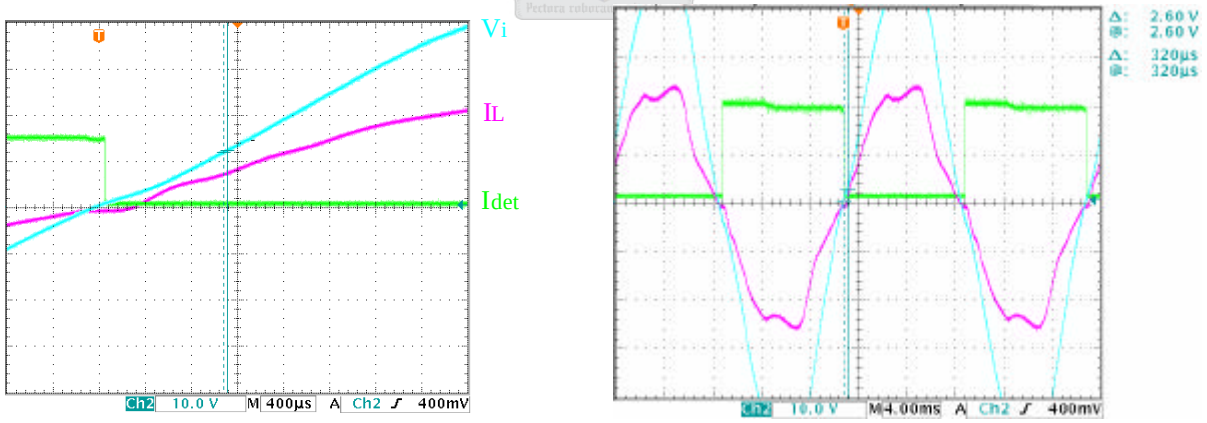


Figure 3-9 The current direction detection circuit.

The output of the opto-coupler is send to the PEC33 controller board via a fiber optic transmitter, which will be discussed in the fiber optic circuit design section.

Figure 3-10 shows a practical result of the current direction detection circuit when the load current changes sign. A long delayed resoponse of 320 µs is shown, which really affects the commutation process. Table 3-4 shows the delay time calculated for the current direction detection circuit.



(a) (b)

Figure 3-10 Current direction detection output when current changing sign. (a) Detection output during the load current direction is changed from negative to positive; (b) Detection output during load current zero crossing.

Table 3-4 The time delay of the load current detection circuit

Components	Delay Time associated
TL3016I High precision Comparator	12.2 ns max
6N317 Optocoupler	75 ns max @ R_L= 350Ω
HFBR2521	4 ns
HFBR1521	80 ns
Optic Fiber	140 ns @ 0.5m
SN75451 Fiber optic driver	25 ns max
FPGA	26.6 ns
Total Time Delay	362.8 ns

3.6 Voltage Measurement Circuit

Since the high phase voltage of the mains supply cannot be connected directly to the ADC of the PEC33 controller board, an equivalent low-level input phase voltage has to be generated, so that the ADC will read the input phase voltage correctly. A voltage measurement circuit is designed to lower the level of the voltage and isolate the DSP from the AC supply. The analysis and design of the voltage measurement required to measure the input phase voltages in the direct matrix converter system will be discussed in this section of the chapter. The designed voltage measurement circuit is designed to measure voltages up to 500V.

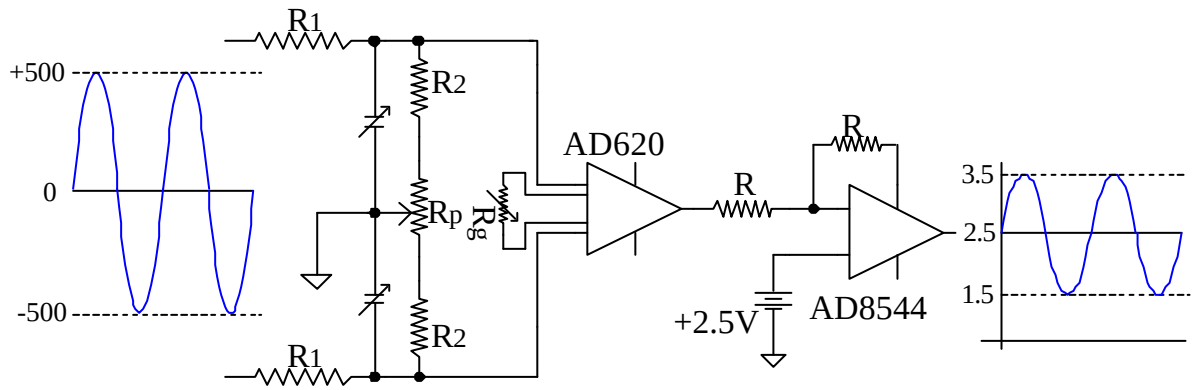


Figure 3-11 Voltage Measurement circuit

The voltage measurement reads high voltage up to a 500 V peak and converts to a lower equivalent 1.5 V peak. The voltage measurement consists of a voltage divider and an instrumentation amplifier as shown in Figure 3-11. The voltage divider circuit converts the high input voltage level into the equivalent lower voltage level. The instrumentation amplifier is used to interface the divider circuit with a TLV1570 serial ADC. The AD620 instrumentation amplifier is a low cost, dual supply device from Analog Devices [22]. The converted phase voltage is biased to +2.5 V by another buffer amplifier since the ADC chip range is greater than zero. Figure 3-12 shows the output of the voltage measurement circuit.

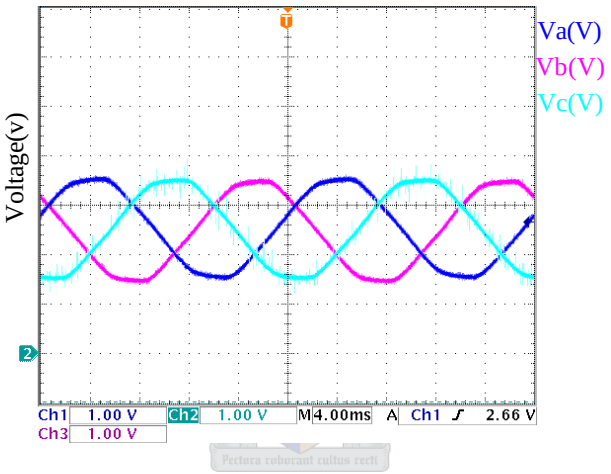


Figure 3-12 The measured three-phase input voltage

Calculation of the voltage divider circuit resistors:

The relationship of the resistors in the voltage divider circuit of the voltage measurement is given in Equation 3-18.

$$\frac{V_O}{V_i} = \frac{2 \cdot R_2 + R_P}{2 \cdot R_1 + 2 \cdot R_2 + R_P}$$ **Equation 3-15**

Where V_O is the output voltage, V_i is the input phase voltage, $R_1 = 5M\Omega$, $R_2 = 10K\Omega$ and the variable resistor $R_P = 10K\Omega$.

The Variable capacitors connected in parallel to R_2 are used to compensate the inductance of the resistors. Their capacitance value is calculated as shown in Equation 3-19.

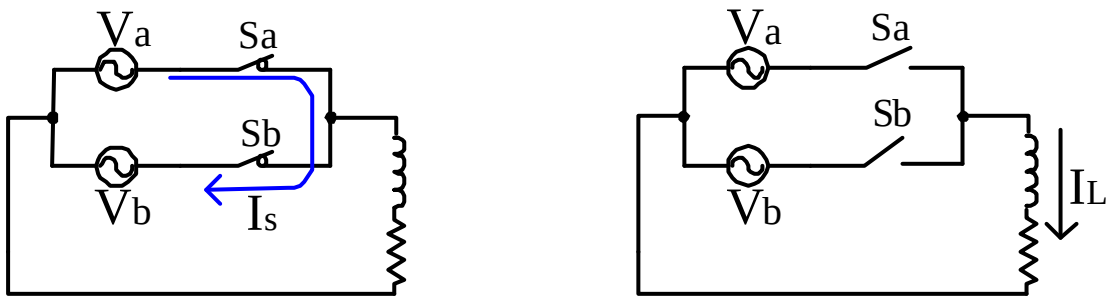
$$C_c = \left(\frac{R_1 + R_2 + \frac{R_P}{2}}{R_2 + \frac{R_P}{2}} - 1 \right) * C_S$$
$$= 16.316\ pF$$

Equation 3-19

Where: $R_1 = 5M\Omega$, $R_2 = 10K\Omega$, $R_P = 10K\Omega$ and a stray capacitor of the resistors.

3.7 Over-Current & Over-Voltage Protection Circuits Design

A fault could occur any time an error PWM control signal is generated by the PEC33 controller board or an error caused due to incorrect load current direction detection by the current direction detection circuit. When a fault occurs, it could be of two forms. A short circuit as shown in Figure 3-13(a), which subsequently results in the rise of a circulating input current to a dangerous level (over-current) that might destroy the converter switches. On the other hand, an open circuit, as shown in Figure 3-13(b), which will in turn produce an over-voltage due to the inductive load and/or the inductance of the wiring that might destroy the converter switches as described in the previous chapter.



(a) (b)

Figure 3-13 A two-phase to single-phase converter circuit. (a) Short circuit of the switches; (b) Open circuit of the switches

The over-current and over-voltage generation has to be detected before it reaches a dangerous level that might destroy the converter switches. Therefore, in order to protect the direct matrix converter switches, an over-current protection circuit and a clamp circuit [17] are used. The over-current protection circuit detects the over-current and the clamp circuit provides a conducting path for the current when the converter switches are switched off. The discussion of the analysis and design of these protection circuits will be examined as follows.

3.7.1 Over-Current Protection Circuit Design

The direct matrix converter circuit is vulnerable to a fault since it lacks a foolproof circuit protection as was mentioned in the previous sections. Therefore, it is important to implement the over-current protection circuit to detect the existence of an over-current or short circuit. Figure 3-14 shows the over-current protection circuit. The function of the over-current protection circuit is to determine whether an over-current occurs due to an error PWM control signal generated by the PEC33 controller board. It then sends a signal that notifies the PEC33 controller board to turn off immediately the entire converter switches.

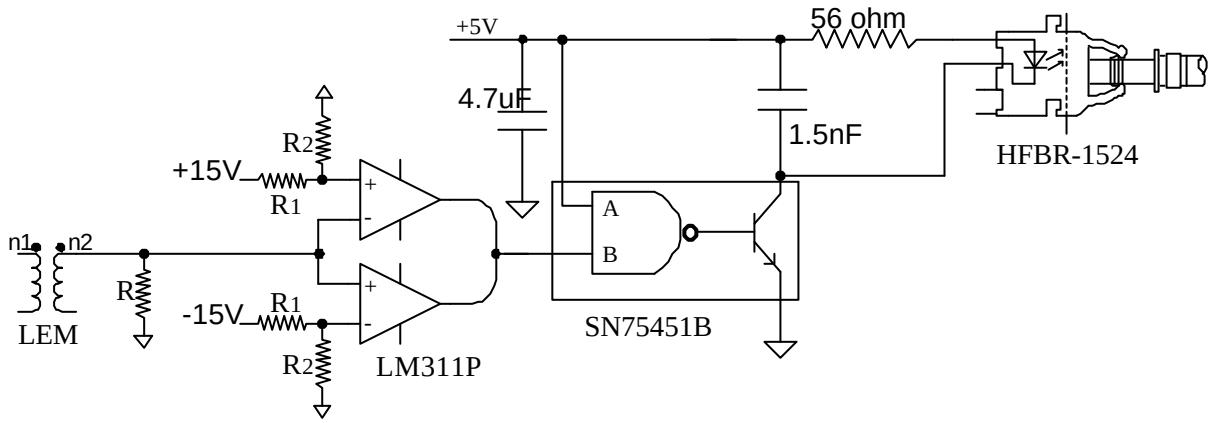


Figure 3-14 Over-current protection circuit

The protection circuit consists of a current probe, an analog comparator circuit, a fiber optic data transferring circuit, and fiber optic driving gate as shown in Figure 3-14

above. The current probe or current transducer shown in Figure 3-14 is a LEM "HTP 25". The conversion ratio of the turns in the LEM "HTP 25" current probe is 1:1000 as given in the datasheet specification [25], but the external wire that carry the actual current has two turns which changes the new turns ratio to 2:1000. The current delivered to the analog comparator circuit by the current transducer is 60 mA, since the maximum allowable current that can be handled by the converter switches is 30 A as obtained from the IGBT "IRG4PH40KD" switch device datasheet specification [20].

The analog comparator circuit measures a voltage drop across a resistor R connected in parallel with the current probe output terminal. This voltage drop across the resistor is compared to two reference voltages generated as an output from a voltage divider circuit as shown in Figure 3-13. The value of the resistor R is 91 Ω, which will produce a voltage drop of 3.64 V. The voltage divider circuit is designed to generate 3.64 V drop across the small resistor R_2. A resistor value of R_1 = 6.8 KΩ and R_2 = 2.2 kΩ will generate a voltage drop of 3.64 V across the resistor R_2 = 2.2 kΩ of the voltage divider circuit. The output voltage of the voltage divider circuit is +3.64 V and -3.64 V as shown in Figure 3-15.

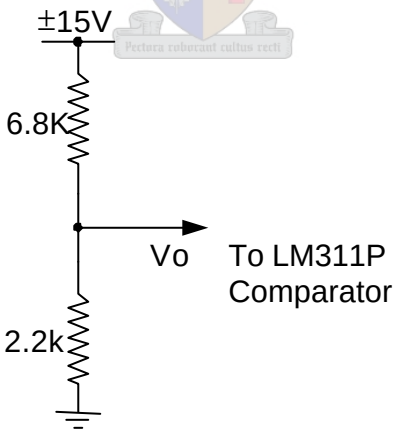


Figure 3-15 Voltage divider circuit

During an over-current or a short circuit fault, the IGBT "IRG4PH40KD" switch device has $t_{SC} = 10\ ms$ short circuit withstanding time. Hence the time delay of the protection circuit has to be lower than the t_{SC} of the IGBT.

Table 3-5 shows the time delay of each component constituting the protection circuit, and the sum of the time delays of the protection circuit are found to be 2.2556 μs, which is fast enough to protect the IGBT switch from destruction.

Table 3-5 The time delay of the over-current protection circuit when a fault occurs

Components	Delay Time associated
LEM	1 μs
LM311P comparator	165 ns
HFBR2521	4 ns
HFBR1521	80 ns
FPGA	26.6 ns
Optic Fiber	140 ns @ 0.5m
HCPL-314j gate drive	0.7 μs
DSP	13.333 ns
Total Time Delay	2.2689 μs

3.7.2 Clamp Circuit Design

The standard AC-DC-AC power converter consists of the DC-bus capacitors and freewheeling diodes, which form a clamp circuit that protects the converter circuit from overvoltage generated because of an open circuit. The direct matrix converter circuit requires a protection when an over-voltage or open circuit occurs. The proposed protection method known as the clamp circuit [17] will be discussed in this subsection of this thesis. Figure 3-16 shows the clamp circuit designed protects the IGBT switches from being destroyed during a sudden open circuit at both sides of the converter terminal due to a generation of an error PWM signal or an over-current failure detected by the over-current protection circuit. Two clamp circuits are connected at both terminals of the direct matrix converter circuit, at both the input side and the output side of the converter circuit.

A number of circuit protections are recommended for the matrix converter circuit. In a conventional rectifier/inverter circuit the input rectifier diodes, the DC-link capacitor, and the freewheeling paths provided by the diodes in the output bridge all contribute to very straightforward circuit protection strategies, particularly when the mains supply is turned off immediately. Since none of these components exists in the matrix converter circuit, a clamp circuit is required to cope with the various possible fault modes.

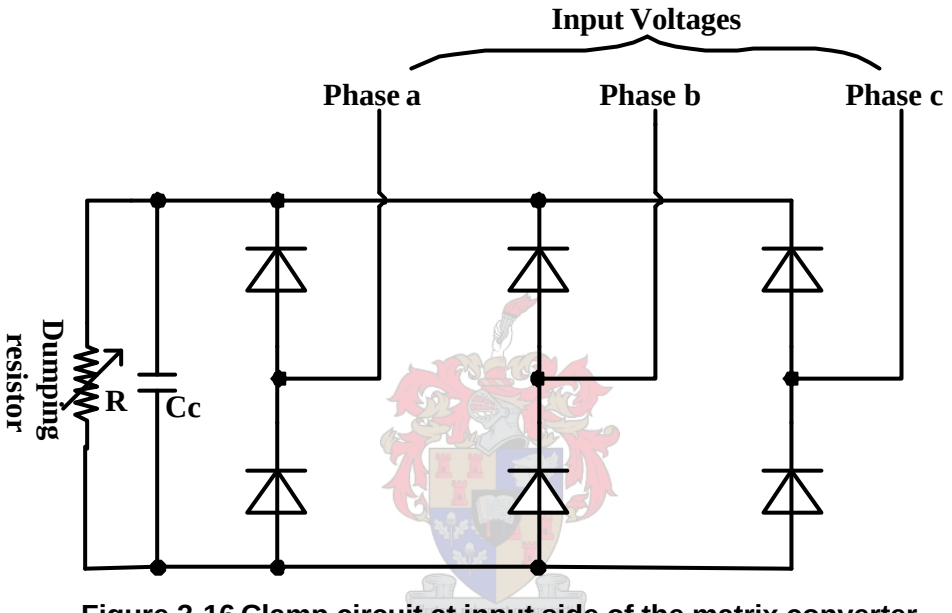


Figure 3-16 Clamp circuit at input side of the matrix converter

A typical clamp circuit for protection of the matrix converter circuit is shown in Figure 3-16. The clamp circuit configuration consists of two rectifier bridge circuits using a fast IRG4PH40KD IGBT at the input and output terminals. These IGBT switches act as normal fast diode rectifiers. A capacitor C_c takes the commutation energy. A dumping resistor will discharge the energy stored in the clamp capacitor. Under normal conditions, the clamp only has to cope with the energy stored in the leakage inductances. The capacitor C_c is designed considering for a critical situation of fault time. The design of the clamp circuit is carried out under the assumption that a fault has generated an error signal to the power converter. All the switches in the converter are turned off immediately. The task of the clamp circuit is to de-energize the load without

damaging the power switches. The total energy stored in a three-phase inductance carrying sinusoidal balanced output current is calculated as:

$$Q_L = \frac{1}{2} L_{SR} (i_a{}^2 + i_b{}^2 + i_c{}^2) = \frac{3}{4} L_{SR} \cdot I_{rms}{}^2$$

Equation 3-16

Where: Q_L is the total energy stored in the three load inductances, L_{SR} is the total leakage inductance of the induction motor, i_a, i_b and i_c are the three output phase currents, and I_{rms} is the rms value of the line currents.

Q_L is the energy to be transferred to the clamp circuit in case the matrix converter feeds a passive load.

$$Q_L = \frac{1}{2} C_C [U_{MAX}{}^2 - 565^2]$$

Equation 3-20

Where U_{MAX} is the maximum allowed overvoltage.

The energy stored in the clamp capacitor of the clamp circuit is de-energized by connecting a dumping resistor parallel to the clamp capacitor as shown in Figure 3-16. Figure 3-16 shows one clamp circuit only connected across the input side of the direct matrix converter circuit, the same clamp circuit is also connected at the output side of the converter circuit.

A clamp capacitance C_C=4.7 µF and a variable damping resistor R = 25 Ω are used in the clamp circuit shown in Figure 3-16.

3.8 Fiber Optic (TX & RX) Circuit Design

Signals are transferred between the matrix converter circuit and the PEC33 controller board via a fiber optic circuit. A brief discussion of the fiber optic circuit is as follows.

Figure 3-17 shows the fiber optic circuit and a gate driver for the fiber optic transmitter device. HFBR1524 (HFBR1521) transmitter device is used to convert and transmit the electrical signal into an optic signal, and the HFBR2524 (HFBR2521) receiver device is used to receive and convert the optic signal into an electrical signal.

30 transmitters and receivers fiber optics are used to transfer the signal between the matrix converter circuit and the PEC33 controller board.

- 18 transmitters and receivers are employed to transfer the PWM control signal to the gate drive circuit of the matrix converter.

- 9 transmitters and receivers are used to transfer information of the load current direction to the FPGA of the PEC33 controller board.

- The remaining 3 transmitters and receivers are used to transfer the overcurrent protection circuit signal to the FPGA of PEC33 controller.

The gate-driving chip is a SN75451B gate chip. This gate chip is used to drive the transmitter fiber optic as shown in the Figure 3-17.

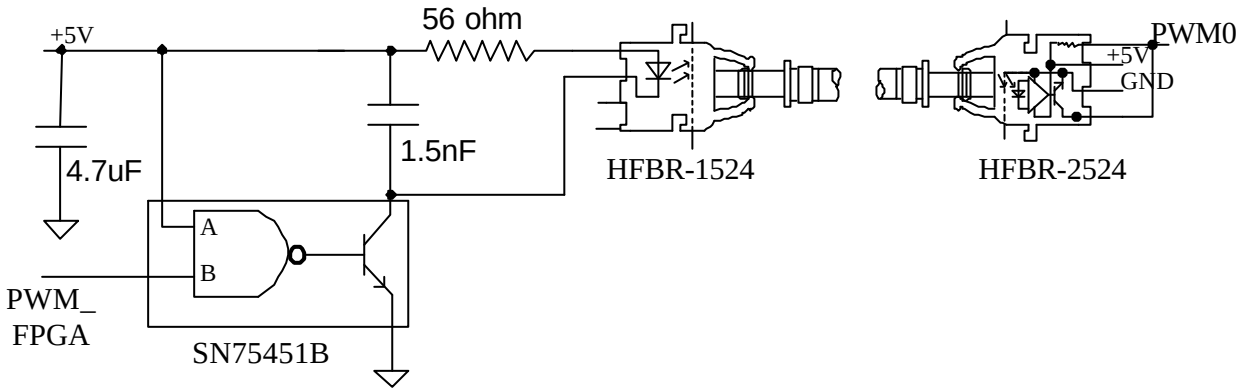


Figure 3-17 Fiber optic data transmission circuit

3.9 Analog Isolation Circuit

The function of the isolation amplifier is to isolate the voltage measurement circuit from the PEC33 controller circuit board since the voltage measurement circuit is directly connected to the mains supply three-phase input. Three analog isolation chips are employed to isolate the three output phase voltages of the voltage measurement. An ISO122 analog isolation amplifier is used to perform the isolation operation. A block diagram of the isolation amplifier chip is shown in Figure 3-18.

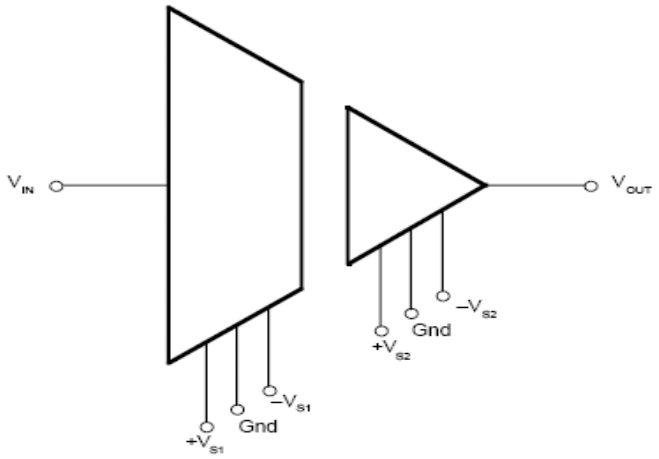


Figure 3-18 Amplifier Isolation Block diagram

A detailed circuit diagram of the isolation amplifier employed is shown in Figure 3-19. The electrical characteristics of the chip can be obtained from the datasheet specification given on the Texas Instrumentation website [23]. The ISO122 is a precision isolation amplifier incorporating a novel duty cycle modulation-demodulation technique. The key specifications are 0.020% maximum non-linearity, rated isolation 1500 V_{rms} and 50 KHz signal bandwidth. It operates from two +15 V and -15 V power supplies.

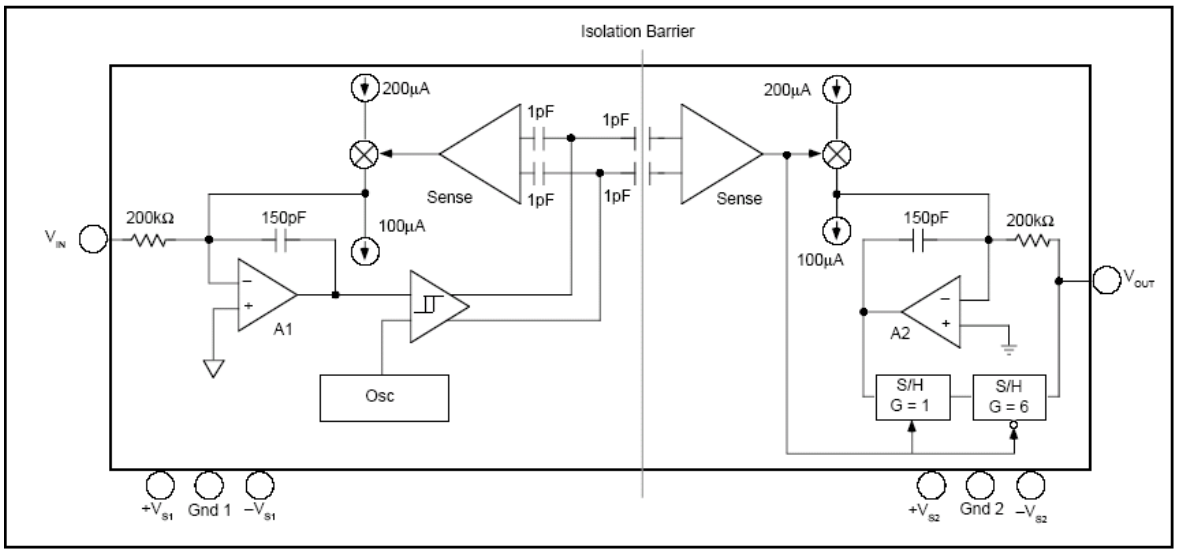


Figure 3-19 A detailed circuit diagram of the ISO122 isolation amplifier

3.10 Direct Matrix Converter Losses

The Loss calculation of the matrix converter includes calculating the switching losses and the conduction losses. Conduction loss calculation is straightforward, but total switching loss calculation is more complex and requires a numerical model of the modulation process to be set up. A brief discussion of both the switching loss and conduction loss follows based on the paper "Comparison of Calculated and Measured Losses in Direct AC-AC Converter" [18].

3.10.1 Switching Losses

Commutation in the matrix converter is not straightforward as in conventional inverters since there are no natural freewheeling paths. Current is transferred from one controlled switch to another at each commutation. To determine switching losses, it is important to understand the mechanisms of commutation. Figure 3-20(a) shows a two-phase to single-phase converter including the parasitic inductance of the IGBT device and stray inductance of the interconnections.

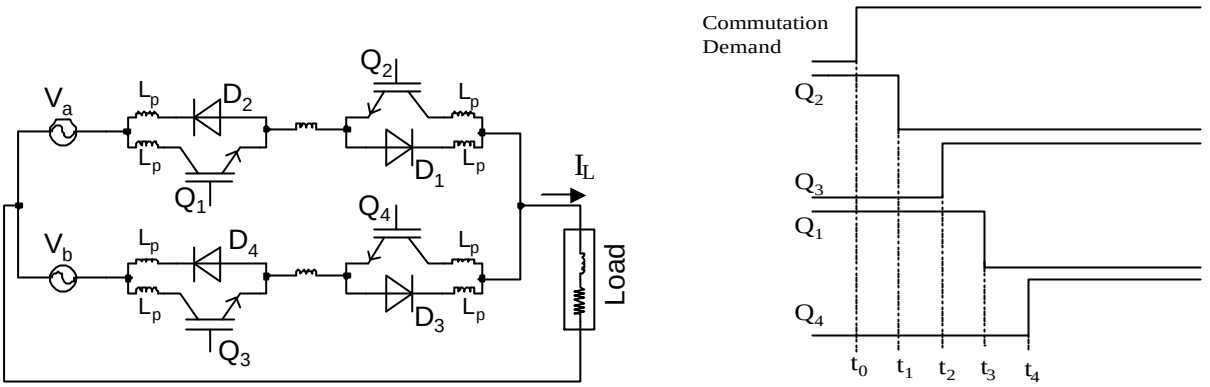


Figure 3-20 AC-AC Matrix converter. (a) Two phase to single phase converter including parasitic inductance; (b) Timing diagram of the switching sequence of a 4 step commutation strategy when I_L is +ve.

Figure 3-20(b) shows the timing diagram of the gates for a four-step commutation method [3] when switching from S_a to S_b with the load current I_L in the direction shown. If V_a is more positive than V_b, commutation will occur at t_3 resulting in a hard turn-off in the switch Q_1 and a soft turn-on in the switch Q_3. Conversely, if V_b is more positive than V_a, commutation takes place at t_2 resulting in a hard turn-on in the switch Q_3 and a soft turn-off in the switch Q_1. Note that there is no switching loss at all in Q_2 and Q_4 for either situation, since neither conduct current when I_L is positive. A similar, yet different sequence of events to those above occurs for negative I_L.

The soft commutations are not completely lossless, but the energy involved is at least an order of magnitude less than for the hard commutations. This discussion therefore concentrates on analyzing and predicting the losses due to hard commutations only.

1. Turn-on Losses

Figure 3-21 shows typical experimental waveforms for a switch undergoing hard "turn-on" given in the paper [18]. The waveform shown is across the entire bi-directional

switch cell. A voltage drop occurs across the parasitic inductance of the switch undergoing hard "turn-on". This voltage drop is given by Equation 3-21:

$$V_{plat} = L_{para} \cdot \frac{di}{dt} \qquad \textbf{Equation 3-21}$$

Where: $L_{para} = 2L_p + L_b$ is the total inductance in the commutating loop external to the switch, V_{plat} is the voltage drop across this inductance and di/dt is the fairly constant derivative of the rising through the incoming switch.

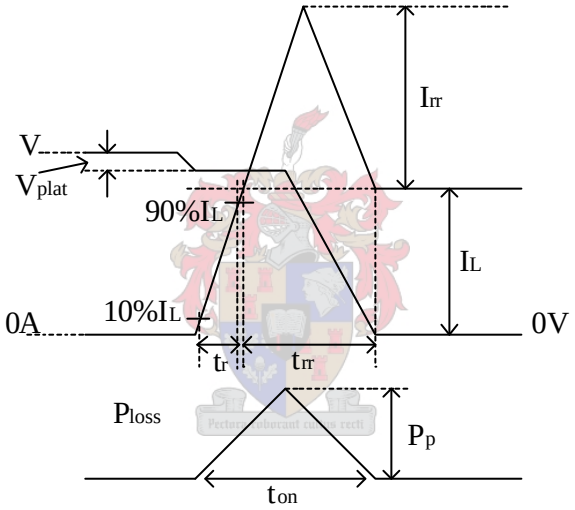


Figure 3-21 Approximation for loss calculation during "turn-on"

The switch current reaches the peak value of $(I_L + I_{rr})$ at t_1. It then decays towards I_L as the voltage across the reverse blocking diode in the opposite switch reaches ΔV. Figure 3-21 shows approximations to the practical waveforms for loss analysis. Therefore, the "turn-on" energy loss, E_{on} at each switching instant is given by Equation 3-22 [18]:

$$E_{on} = 1/2 \cdot t_{on} \cdot P_p \qquad \textbf{Equation 3-22}$$

Where, the power P_p is given by Equation 3-23.

$$P_p = (|\Delta V| - V_{plat}) \cdot (|I_L| + I_{rr}) \qquad \textbf{Equation 3-23}$$

And t_{on} is approximated by $t_{on} = t_r + t_{rr}$. The time t_r and t_{rr} and the current I_{rr} are given in the datasheet of the device [19].

Substituting Equation 3-23 on 3-22 will result in Equation 3-24 [18]:

$$E_{on} = \tfrac{1}{2} \cdot (t_r + t_{rr}) \cdot \{(|\Delta V| - V_{plat}) \cdot (|I_L| + I_{rr})\} \qquad \textbf{Equation 3-24}$$

2. Turn-off Losses

The loss energy during the "turn-off" time is also calculated according to the same approach as the "turn-on" loss calculation. Figure 3-22 shows typical experimental waveforms for hard "turn-off" together with the approximation waveforms made for analysis. The "turn-off" loss is divided into 2 parts, the main switching loss and the loss due to the 'tail' current [18].

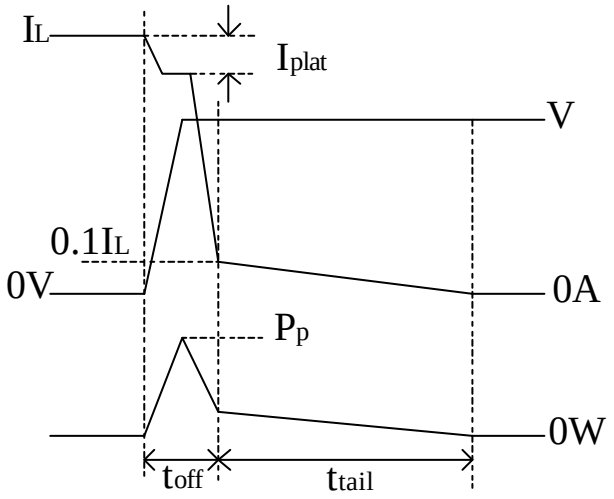


Figure 3-22 Turn-off waveform approximation for power loss calculation

The main energy loss can be approximated by Equation 3-25 [18].

$$E_{off(main)} = \frac{1}{2} \cdot t_{off}(ns) \cdot |\Delta V| \cdot |I_L| \cdot 10^{-9} \quad \textbf{Equation 3-25}$$

The loss due to the 'tail' current is approximated by Equation 3-26 [18].

$$E_{off(tail)} = \frac{1}{2} \cdot 10^{-6} \cdot (0.1 \cdot |\Delta V| \cdot |I_L|) \quad \textbf{Equation 3-26}$$

3.10.2 Conduction Losses

The instantaneous conduction loss $P_{cond}(t)$ of an IGBT device is given by [18]:

$$P_{cond}(t) = V_{CE}(sat) \cdot I_C(t) \quad \textbf{Equation 3-27}$$

Where $V_{CE}(sat)$ and I_C are the collector-emitter saturation voltage and the collector current of the device respectively. The relationship of the $V_{CE}(sat)$ and I_C is given in the datasheet and can normally be linearised into the form given by Equation 3-28 [18].

$$V_{CE}(sat) = V_{CE}(0) + r_{CE} \cdot I_C \quad \textbf{Equation 3-28}$$

Where r_{CE} is the approximated on-state resistance slope and $V_{CE}(0)$ is the forward voltage drop across the device when the collector current is small.

The conduction loss Equation 3-29 is obtained when Equation 3-28 is substituted in Equation 3-27.

$$P_{cond,IGBT}(t) = (V_{CE}(0) + r_{CE} \cdot I_C) \cdot I_C \quad \textbf{Equation 3-29}$$

For the IGBTs and a similar process for the diodes gives:

$$P_{cond,DIODE}(t) = (V_{T0} + r_d \cdot I_d(t)) \cdot I_d(t) \qquad \textbf{Equation 3-30}$$

Where V_{T0} is the voltage drop across the diode at low current and r_d is the approximated on-state resistance slope.

Since the output current flows through one active device and one diode in a matrix converter, regardless of the switching pattern or load current direction. Therefore, the total instantaneous conduction loss is given in terms of the instantaneous output current, $I_O(t)$ as:

$$P_{cond,ph}(t) = \{V_{T0} + V_{CE}(0)\} \cdot |I_O(t)| + \{r_d + r_{CE}\} \cdot I^2{}_O(t) \qquad \textbf{Equation 3-31}$$

The average conduction loss per phase for peak amplitude I_{om} of a sinusoidal output current is given by Equation 3-32.

$$\overline{P}_{cond\,ph}(t) = 2 \cdot \frac{I_{om}}{p}(V_{T0,d} + V_{CE}(0)) + \frac{1}{2} I^2{}_O \cdot (r_d + r_{CE}) \qquad \textbf{Equation 3-32}$$

In general, loss calculation for a complete matrix converter in any operating regime can be calculated using the given Equations 3-24, 3-25, 3-26 and 3-32 as proposed in [17]. The calculation of Conduction loss is simple and straightforward. But, Total switching loss calculation is more complex and requires a numerical model of the modulation process to be set up. This together with an assumed set of output currents yields the load current and input voltages at each commutation instant. This information in turn enables determination of whether the commutation involves hard "turn-on", or hard "turn-off" and allows energy loss for each commutation to be calculated using the appropriate expression.

3.11 Heat Sink Design & Analysis

The design of an appropriate heat sink for the matrix converter requires the knowledge of power dissipation of the IGBT switches. Equation 3-33 [28] is used to calculate the average conduction power dissipated of a single switch over one half-cycle of a 50 Hz supply and 50 Hz output for a given fixed duty cycle. The switching losses are given by Equation 3-34 as proposed in [28]. Furthermore, the parameters required for the calculation are given in the datasheet of the switching device [20]. The total power loss of the switch is then given as the summation of the conduction loss and the switching loss as shown in Equation 3-35.

$$P_{cond\,(IGBT+Diode)} = d^2 V_{CE}\frac{V_m}{pR} + d^2 V_{To}\frac{V_m}{pR} \qquad \textbf{Equation 3-33}$$

$$P_{switching} = \frac{dV_m^{\,2}}{16T_S R}(t_{on}+t_{off}) \qquad \textbf{Equation 3-34}$$

$$\begin{aligned} P_{total} &= 9\cdot 2\cdot (P_{cond\,(IGBT+Diode)} + P_{switching}) \\ &= 18\cdot (d^2 V_{CE}\frac{V_m}{pR} + d^2 V_{To}\frac{V_m}{pR} + \frac{dV_m^{\,2}}{16T_S R}(t_{on}+t_{off})) \\ &= 18\cdot (\frac{1}{3}\cdot 2.8\cdot 400/10\cdot p + (\frac{1}{3})^2\cdot 2.6\cdot 400/10\cdot p + \frac{1/3\cdot 2\cdot (400)^2}{16\cdot 200us\cdot 10}(81ns+470ns)) \\ &= 313.168\ W \end{aligned} \qquad \textbf{Equation 3-35}$$

Where the resistive load R is 10Ω and the duty cycle d is 0.3333.

Equation 3-36 gives the relationship between the junction-to-ambient thermal resistance, R_{qJA}, the total power loss of a single IGBT switch, P_{total}, the maximum junction temperature, $T_{j\max}$, and maximum ambient temperature, $T_{a\max}$. The value of the maximum junction temperature, $T_{j\max}$, is given in the IGBT IRG4PH40KD datasheet specification [20].

$$R_{qja} = \frac{T_{j\max} - T_{a\max}}{P_{total}}$$ **Equation 3-36**

Equation 3-37 gives the relationship between R_{qJA}, the junction-to-case thermal resistance, R_{qJC}, the case-to-heat sink, R_{qCS}, and the heat sink-to-ambient thermal resistance, R_{qHA}.

$$R_{qHA} = R_{qJA} - R_{qJC} + R_{qCS}$$ **Equation 3-37**

$$R_{qHA} = \left(\frac{150-40}{313.33/18}\right) - 0.77 - 0.24 \quad {}^{o}C/W$$

$$= 5.31\ {}^{o}C/W$$

The value of the heat sink-to-ambient thermal resistance R_{qHA} is used to select the appropriate heat sink.

3.12 Summary

The development, design, and analysis of the low cost all silicon direct matrix converter was introduced in this chapter including the converter's input filter and output filter. This chapter discussed the hardware design of the three-phase-to-three-phase direct matrix converter circuit. The converter was designed to generate a three-phase output voltage to replace the conventional standard AC-DC-AC inverter based converter, because of the desired features of the direct matrix converter mentioned. The main power converter was designed to conduct the load current through to an induction motor. The gate drive circuit and its isolated power supply have been designed and investigated. The current direction detection circuit was designed to measure the voltage drop across the bidirectional switch. The voltage measurement, over-current protection, over-voltage protection (clamp circuit), heat sink, and analog isolation circuits were analyzed and designed.

CHAPTER 4

CONTROL ALGORITMS OF THE DIRECT MATRIX CONVERTER TOPOLGY

4 CONTROL ALGORITHMS OF THE DIRECT MATRIX CONVERTER TOPOLGY

4.1 Introduction

The previous two chapters of the thesis discussed the different topologies of the matrix converter system, and the analysis and design of the matrix converter circuit. This chapter will discuss the control algorithms for modulating the AC/AC direct matrix converter system based on the paper by Venturini [1-2] and [5]. Figure 4-1 shows a block diagram of the direct matrix converter system: the matrix converter circuit and the controller board.

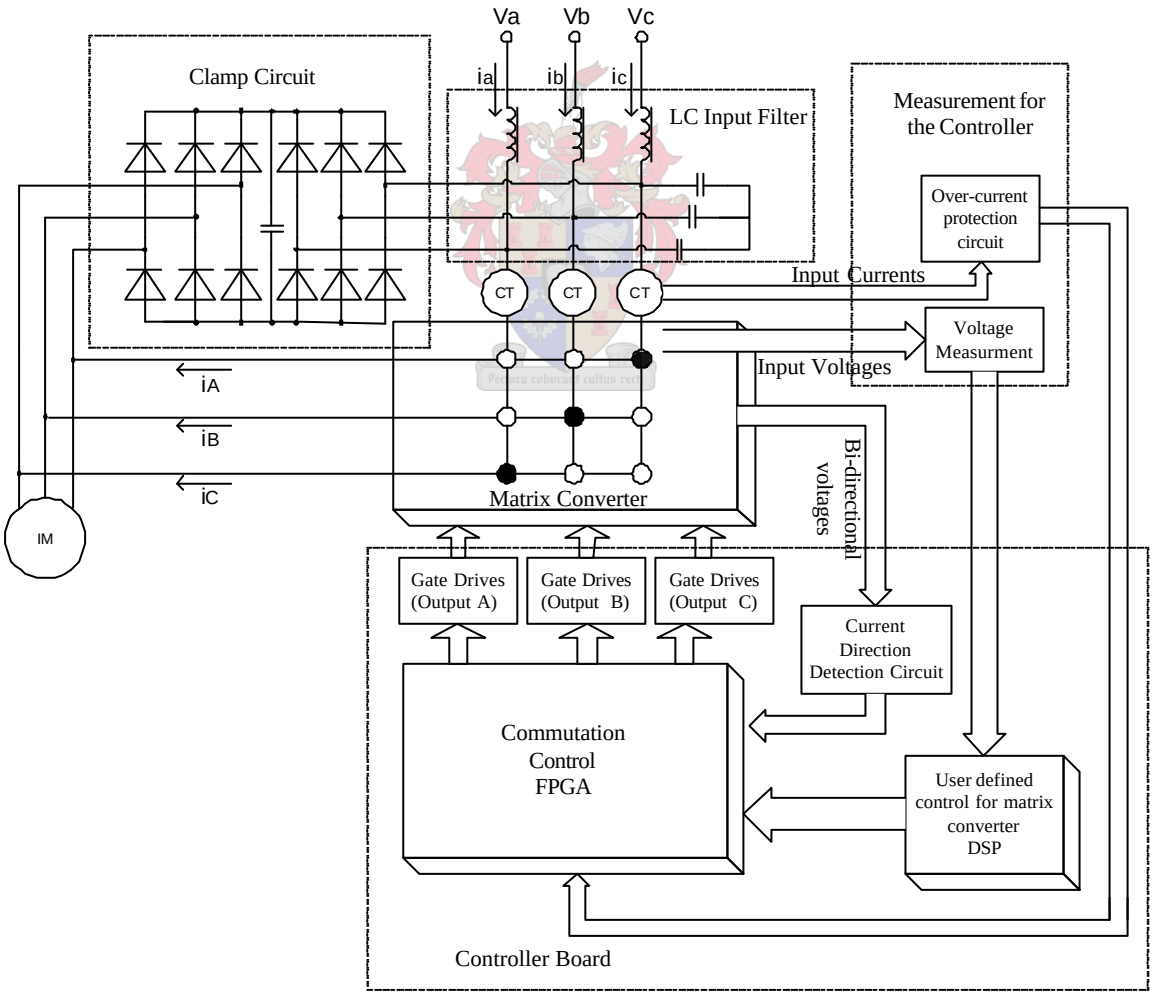


Figure 4-1 Block diagram of Direct Matrix Converter Controller

This chapter is divided into two sections: where the first section discusses a control algorithm based on the current commutation method [3], while the second section explains controlling the matrix converter switches using the voltage commutation method, [4] and [9]. These methods of commutation are intended to ensure safe switching of the converter switches explained in chapter two. Both commutation methods apply the same controlling mechanism of calculating the duty cycles, and a single PEC33 controller board for the implementation of the PWM control signal outputs to the converter switching devices. The PEC33 controller board consists of a built-in DSP device, which calculates the duty cycles and identifies the six sectors of the input phase voltage using space vector representation, and built-in FPGA devices used to generate the required PWM control signals to the converter switches, and controlling the addressing and data transfer. The DSP device is loaded with a C code to read digital measured input phase voltages, to generate three-phase desired output voltage references, and subsequently calculate the 9 duty cycles required by the FPGA device. There are two FPGA devices: an analog FPGA and a main FPGA device in the controller board. A VHDL program is loaded into the FPGA main and analog using Quartus II compiling software. The calculated duty cycles are fed to the analog FPGA device via a parallel bus, which is controlled, by the main FPGA device. The purpose of the analog FPGA device is to generate a sequence of PWM control signal outputs to modulate the IGBT switches of the AC/AC direct matrix converter. Since the matrix converter circuit contains 18 IGBT switches, the controller board generates 18 PWM control signal outputs that modulate the switches so that a controlled output voltage level and frequency range is generated.

The matrix converter system consists of three parts as shown in Figure 4-1: the power converter plant (matrix converter circuit), the input filter, an integrated protection system, and a controller board. The analysis and design of the first two parts of the converter were discussed in the previous chapter. The controller board and its functions will be explained in this chapter. The control algorithm based on the current commutation method will be discussed first and is followed by the discussion of the control algorithm using the voltage commutation method.

4.2 Control Algorithm Based on Current Commutation Method

The current commutation method based control algorithm proposed in [3] applies a four-step commutation strategy for a safe switching during commutating from one input phase to another input phase. This four-step safe commutation strategy ensures that no short circuit will occur and that a current conducting path provides that the load current flow (conducts) at all times. This section of the control algorithm is divided into a DSP controller and an FPGA controller subsection. The DSP controller subsection will examine the calculation of the duty cycles from the generated desired output voltage references in the DSP device and read measured digital input voltages. Subsequently, the generation of 18 PWM output control signals, which are required to control the modulation of the converter switches will be discussed in the FPGA controller subsection.

The DSP controller discussion will be introduced first, and then the FPGA controller discussion will proceed.

4.2.1 DSP Control Algorithm and Analysis

This subsection examines the C code employed to calculate the desired duty cycles based on the paper of Venturini [1], [2] and [5]. A functional block diagram of the DSP controller is shown in Figure 4-2. This controller consists of an interrupt function block, a main function block, and an output phase voltage reference generating function block as shown in Figure 4-2. The main function block contains C code that initializes the ADC, DAC, and LCD display. It sends a request signal to the analog FPGA so that a new measured input phase voltage is ready for reading. The analog FPGA device is responsible for providing the proper sampling and buffering time required by the ADC converter as given in the datasheet [28].

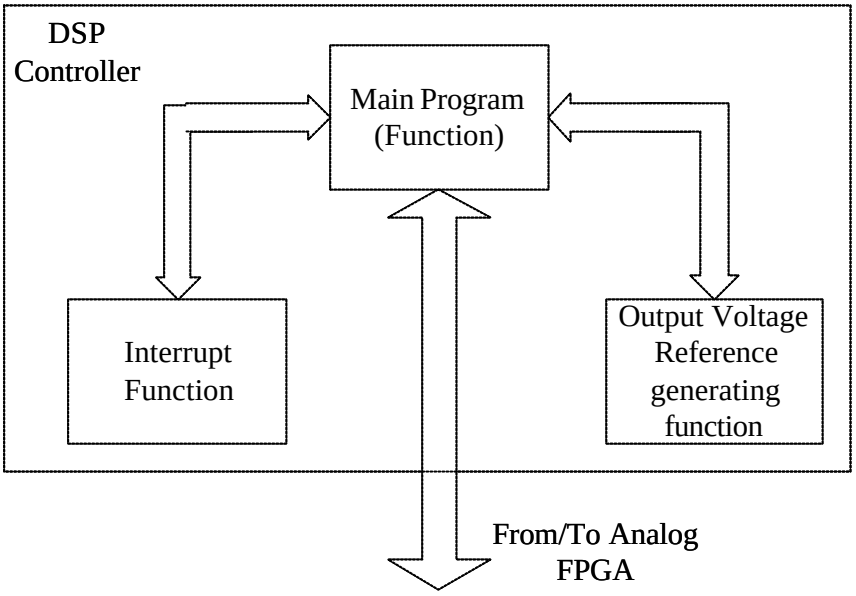


Figure 4-2 A functional block diagram of the DSP controller

The measured input phase voltages sent to the DSP device from the analog FPGA buffer is given by Equation 4-1:

$$V_k(t) = V_{im} \cdot \sin(w_i t - q_k), \ for\ \mathrm{k} \in \{a,\ b,\ and\ c\}\ and\ q_k = \{0,\ 2p/3\ and\ 4p/3\} \qquad \textbf{Equation 4-1}$$

Where $V_k(t)$ is the input phase voltage, V_{im} is the peak voltage of the input phase voltage, and w_i is the input angular frequency.

The desired reference output phase voltage function block generates the desired reference output phase voltages given by Equation 4-2.

$$V_j(t) = m \cdot V_{im} \cdot \sin(w_o t - q_j), \ for\ j \in \{A,\ B,\ and\ C\}\ and\ q_j = \{0,\ 2p/3\ and\ 4p/3\} \qquad \textbf{Equation 4-2}$$

Where: $V_j(t)$ is the desired reference output phase voltage, m is the modulation index, V_{im} is the peak voltage of the input phase voltage, and w_o is the output angular frequency.

The interrupt function is employed to calculate the duty cycles. The duty cycles are required for the generation of the PWM control signals. The formula used to calculate duty cycles is given by the following Equation 4-3:

$$d_{kj}(t) = \frac{Ts}{3} \cdot (1 + \frac{2 \cdot V_j(t) \cdot V_k(t)}{V_{im}{}^2}), \quad for\ j \in \{A, B,\ and\ C\}\ and\ k \in \{a,\ b,\ and\ c\}$$ **Equation 4-3**

Where: $d_{kj}(t)$ is the duty cycle, Ts is the switching period, $V_j(t)$ is the desired reference output phase voltage, $V_k(t)$ is the input phase voltage, and V_{im} is the peak voltage of the input phase voltage.

A flow chart of the C code loaded to the DSP device to perform the above-mentioned functions is shown in Figure 4-3. The DSP device runs at a speed of 75 MHz clock cycle. At first, the ADC is initialized to start sampling and make ready the measured input phase voltages to be read by the DSP device. Reference output phase voltages of desired magnitude and frequency are generated in the output voltage reference generating function part of the code. Then, 10 μs before the end of every switching period the duty cycles are calculated for the next commutation by the interrupt function. These values are sent to the analog FPGA at the end of the interrupt function of the C code. A complete C code is given in appendix A.1.

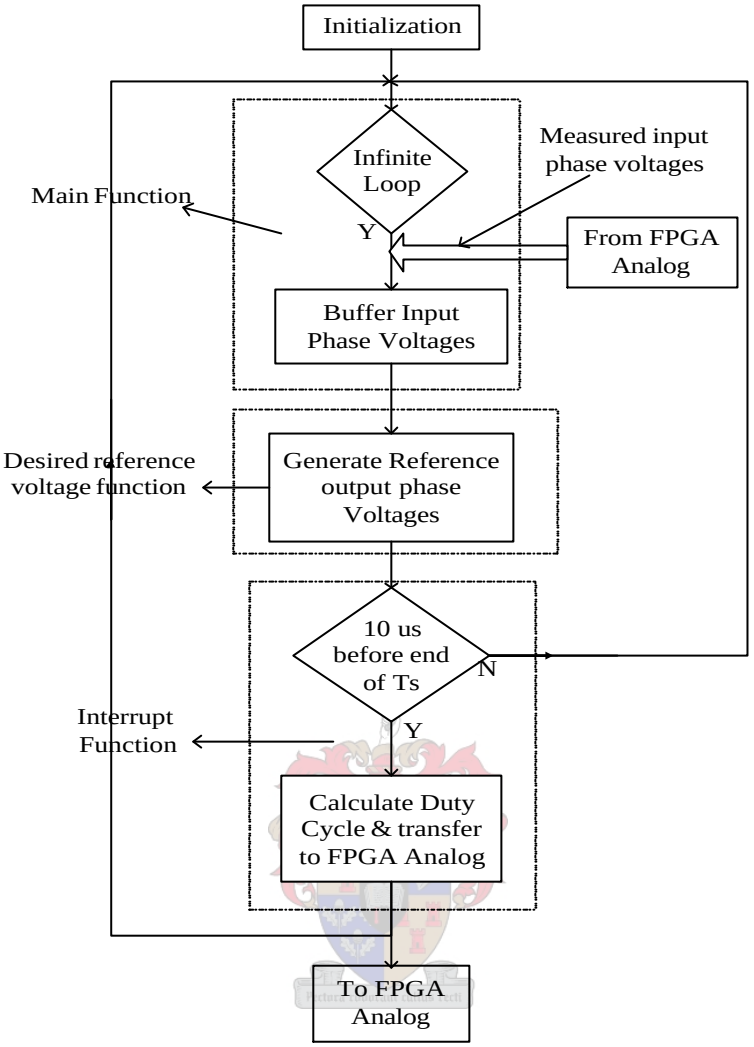


Figure 4-3 The DSP flow chart for reading and calculating

4.2.2 FPGA Control Algorithm and Analysis

Figure 4-4 shows an overview of the functional block diagram of the FPGA controller of the PEC33 Controller board. The data bus connected to the DSP controller is included in the block diagram to show bus connectivity for data and address transfer between the FPGA device and DSP device only. The controller consists of two FPGA devices. Firstly, the main FPGA, which contains the main firmware that controls the serial communication to the flash RAM and to generate the chip-select and real-time clock for the PEC33 board. Secondly, the analog FPGA that generates the PWM control signals

to modulate the bidirectional switches of the direct matrix converter system. The PWM signal outputs are generated by comparing the duty cycles received from the DSP device/controller with internally generated sawtooth carrier signals in the FPGA analog. The FPGA analog controller consists of two main VHDL blocks that constitute the controller of the direct matrix converter topology, a duty cycle buffer block and three PWM controller blocks. The duty cycle buffer block runs at the same frequency as the DSP device (75 MHz) so that an error of data transmission will not occur due to asynchronous data transfer. The three PWM controller blocks run at a lower frequency of 30 MHz than the buffer block which generates the 18 PWM control signals required to modulate the power converter switches to produce the desired output voltage magnitude and frequency range.

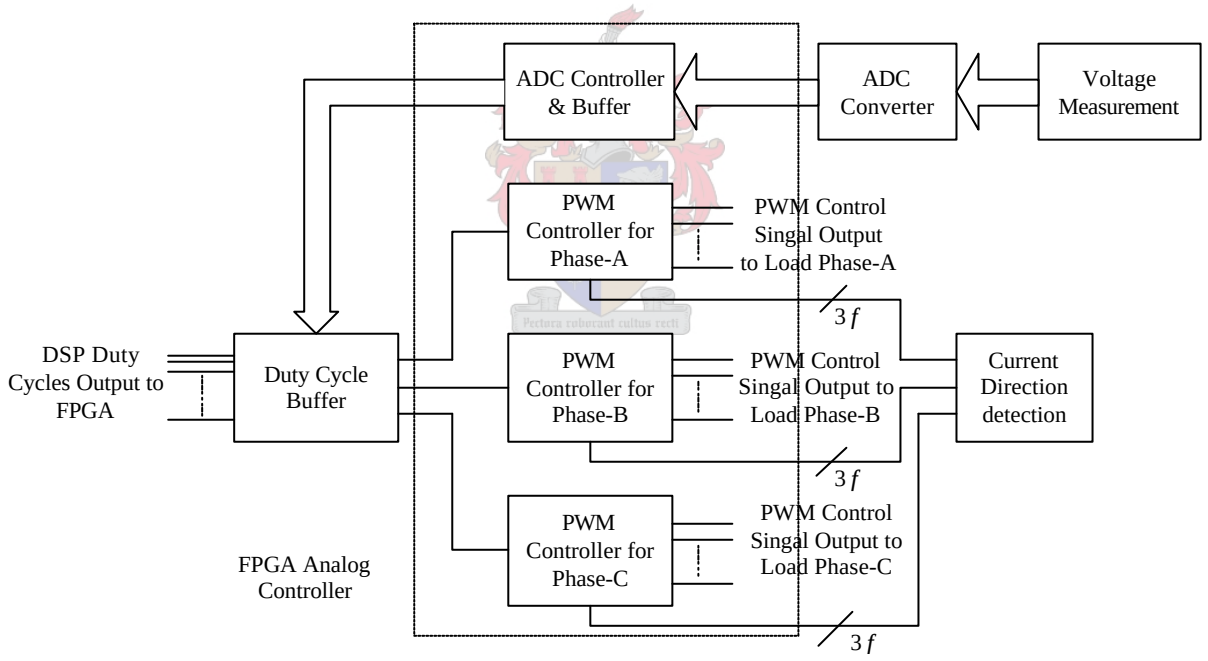


Figure 4-4 A functional block diagram of the FPGA controller

The ADC controller and buffer block store the equivalent digital voltages of the three-phase input voltages measured by the voltage measurement. It then sends these to DSP device/controller. The DSP device calculates the duty cycle values as

discussed in the above section and sends these values back to the duty cycle buffer block of the analog FPGA device/controller.

A request signal (interrupt request) is sent to the DSP controller once every 200 µs because 5 kHz switching frequency is used. Therefore, a new duty cycle has to be available. Figure 4-5 shows the timing sequence of generating the "interrupt" request signal by the duty cycle buffer block to the DSP device. The request is sent 10 µs before the end of every switching period Ts so that the duty cycles will be ready in the duty cycle buffer block for the next commutation to start immediately without delay or interruption when receiving a new data (duty cycles).

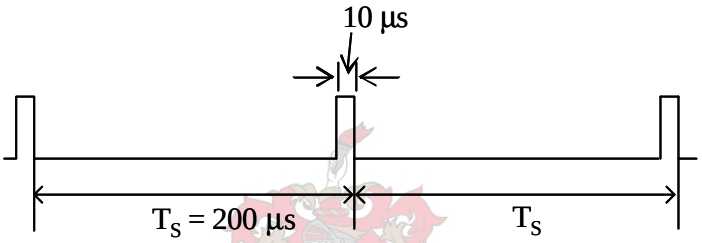


Figure 4-5 The timing sequence of generating the interrupt request

The duty cycles $d_{kj}(t)$ are stored in the FPGA buffer controller after being transferred from the DSP controller until a new value is available for reading again. The three PWM controller blocks employed are required to generate the PWM control signal outputs taking into account these values stored in the buffer controller block.

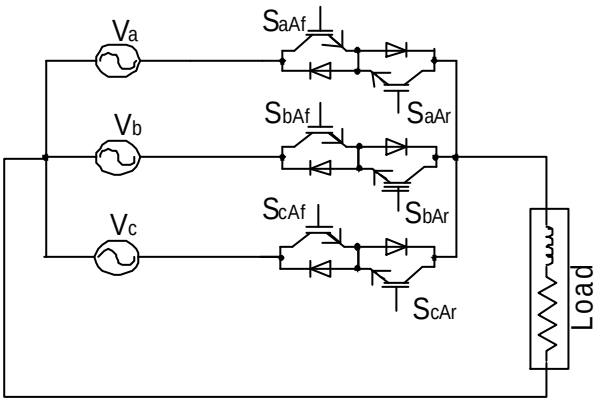


Figure 4-6 A three-phase to single phase connection using bidirectional switches

Consider a single PWM controller block, which is used to control the modulation of the three-phase bi-directional switches connected to load phase-A, as shown in Figure 4-6. The remaining two PWM controller blocks can apply the same control algorithm. The PWM controller block consists of two VHDL processes. Process one generates the sawtooth carrier signal and the signals that initiate commutation from one input phase to another input phase. Process two generates the actual PWM control signals needed to modulate the power converter switches of the direct matrix converter system based on the four-step safe commutation strategy [3].

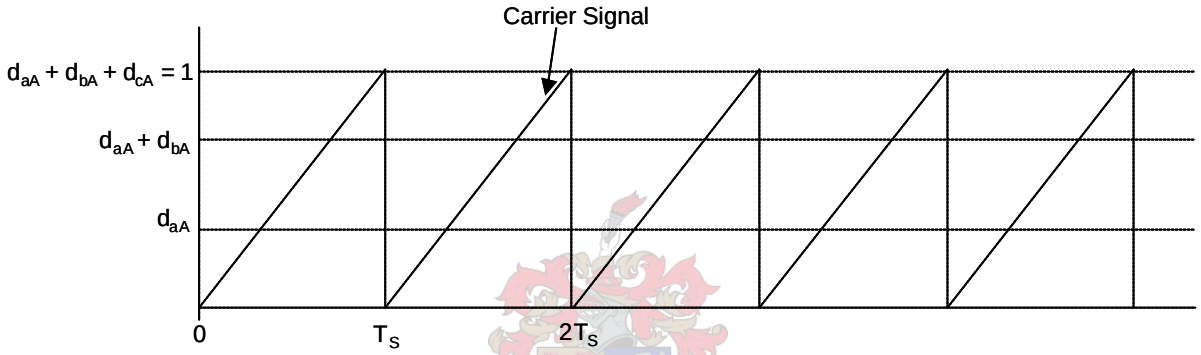


Figure 4-7 A carrier wave signal generated in the analog FPGA controller

Figure 4-7 shows the generated sawtooth carrier signal and a sample of the duty cycles. This carrier signal is compared with the three duty cycles from the DSP device buffered in the duty cycle buffer block of the analog FPGA to generate the commutation initiating signals as shown in Figure 4-8.

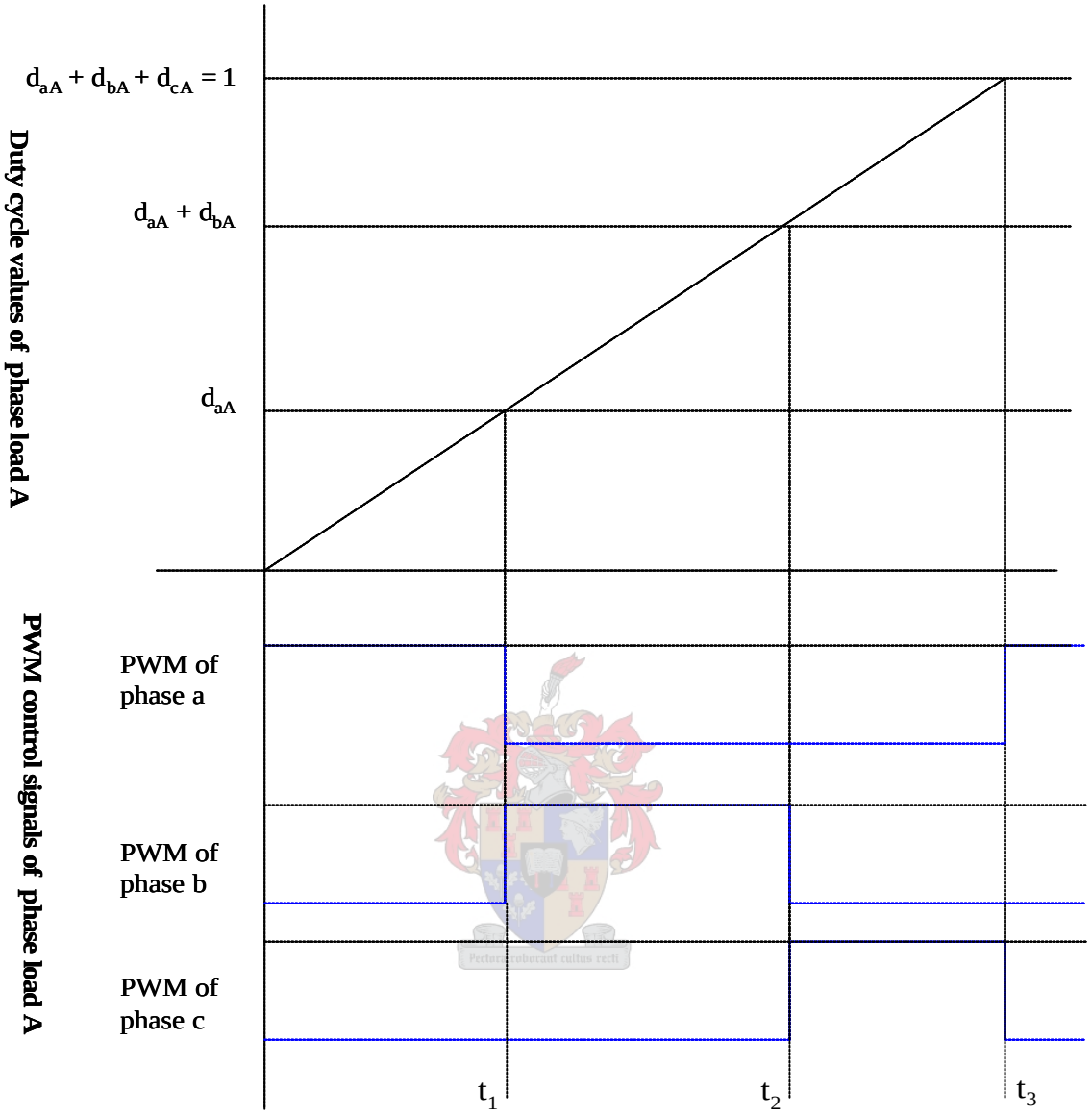


Figure 4-8 Commutation initiating signal generated

The duty cycles d_{aA}, d_{bA} and d_{cA} correspond to phase-a, phase-b and phase-c respectively. Commutation from phase-a to phase-b starts at time t_1 and input phase-b supplies the load current for the time $t_2 - t_1$. Commutation from phase-b to phase-c begins at time t_2. Input phase-c supplies the load current for the time $t_3 - t_2$ until commutation from phase-c to phase-a begins at time t_3 as shown in Figure 4-8. These

same processes are repeated every switching period T_s to generate the desired output voltage of load phase-A. The remaining PWM controller blocks apply the same approach to generate the output load phases-B and phases-C.

A flow chart that shows the generation of the commutation initiating signals is shown in Figure 4-9. Where, Max is constant value equal to the sum of the duty cycles that are calculated to control a single load output phase, and is unity (peak value of the sawtooth reference carrier signal).

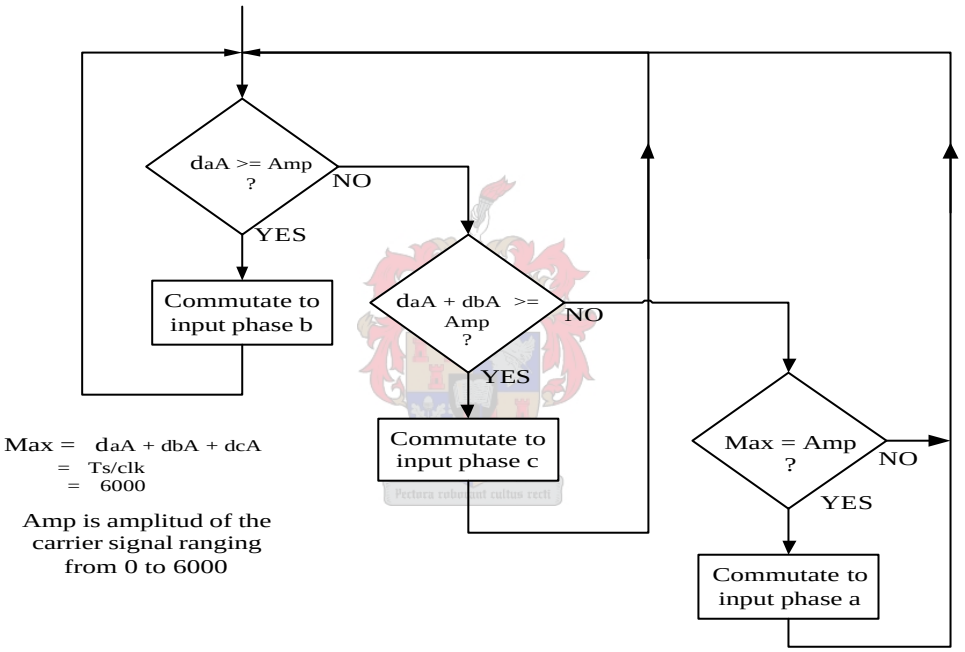


Figure 4-9 A flow chart of generating switching initiating signals of load phase A

The three-phase bi-directional switches connecting the single load output phase-A to the three-phase supplies modulated using the four-step safe commutation strategy is shown in Figure 4-10.

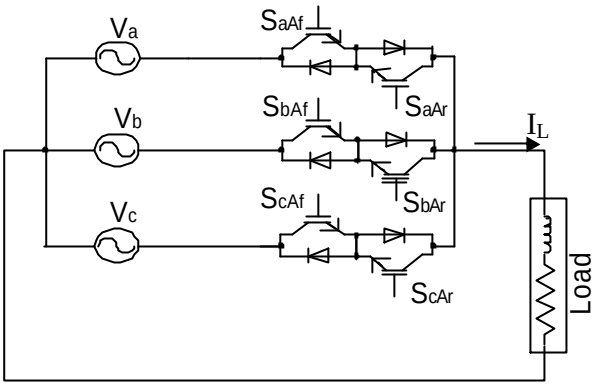


Figure 4-10 A three-phase to single-phase Matrix converter circuit

Figure 4-11 shows a flow chart of the four-step safe commutation strategy. The commutation strategy has four steps so that the right switch is gated at each step for safe commutation. These steps are explained as follows:

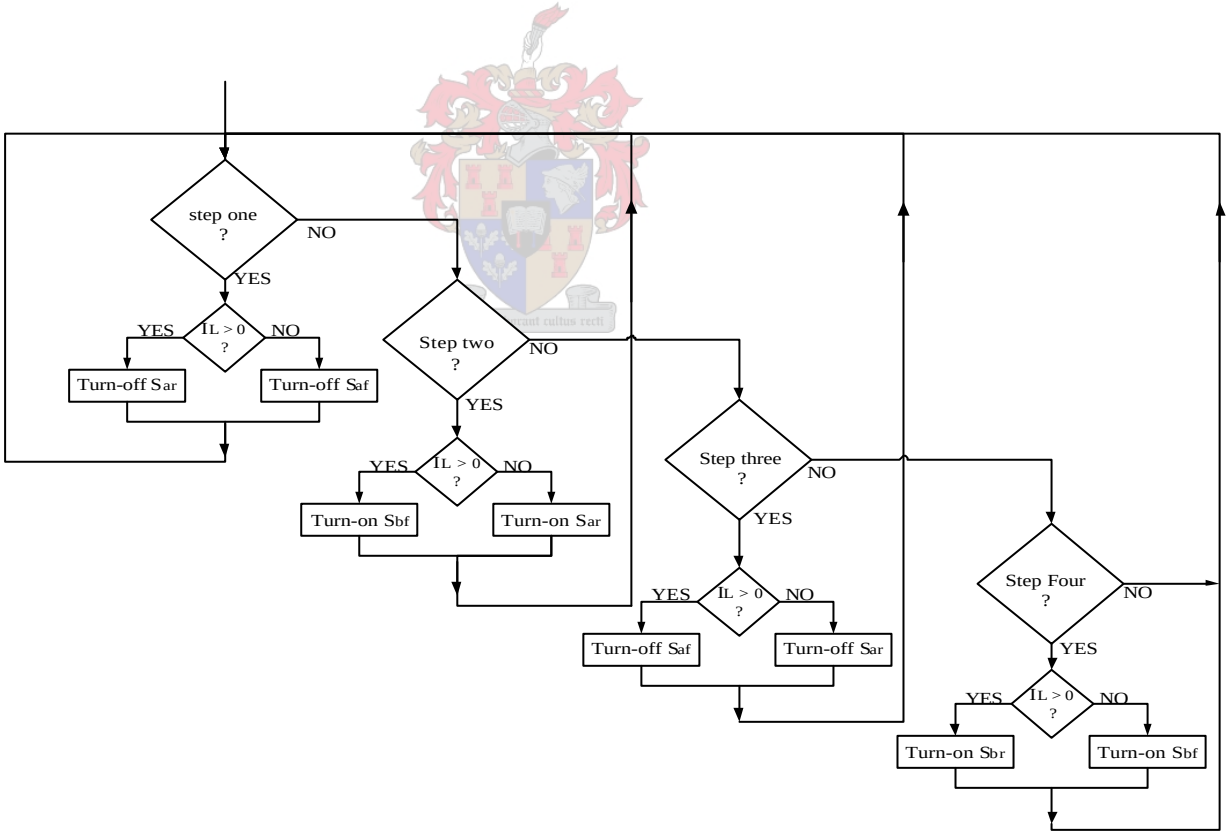


Figure 4-11 A flow chart of the four-step safe commutation strategy

- Step one: if condition checks the direction of load current I_L, if the load current I_L is in the negative portion, then the forward switch S_{aAf} is turned off or else the reverse switch S_{aAr} is turned off since the forward switch is conducting the load current (positive load current direction). Where I_L is a variable signal, which stores the load current direction information from the current direction detection circuit of the converter.

- Step two: The existence of a conducting path for the load current is ensured by gating the right switch in this step. If the load current is positive, then the forward switch S_{bAf} is turned on, otherwise the reverse switch S_{bAr} is turned on to connect the input phase-b to the load. Since both input phases are connected to the load, only one phase is going to supply the load current depending on the relative input phase voltage magnitude.

- Step three: Hence, a conducting path is ensured by turning on the incoming switch, therefore, it is safe to turn off the outgoing switch depending on the load current information of I_L as shown in the flow diagram of Figure 4-11.

- Step four: A bi-directional flow of the load current is possible in this step since the remaining switch of the incoming bi-directional switch cell is turned-on.

Each step introduces a 5 μs delay time so that the switches will be either completely turned off or turned on. This delay time is generated in one of the processes discussed above. A complete VHDL code of the switching process is given in appendix A.2.

4.3 Control Algorithm Based on Voltage Commutation Method

The control algorithm of the direct matrix converter based on the voltage commutation method solely depends on the knowledge of a reliable input voltage. The detection of load current is not necessary. A reliable knowledge of the input phase voltages only is

enough. The voltage measurement device used to measure the input phase voltages have to be efficient enough to provide the exact replica of the high-level input voltages. A sampling rate of 20 μs is used in the ADC converter so that a fast reading of the input voltages will be achieved to determine the exact sector location of the input phase voltages. The DSP calculates and sends the duty cycles to the FPGA once every T_s=200μs switching period, in the same way as discussed in the control algorithm based on the current commutation method section. Besides, the six sectors of the input voltage are determined by the DSP device. These sectors are determined based on the measured input voltage represented in the space vector representation to identify the maximum and minimum input phase voltages respectively.

4.3.1 DSP Control Algorithm and Analysis

The DSP control algorithm of the current and voltage based commutation method is similar, except that the voltage commutation method has an additional function that determines the sectors of the input phase voltages as shown in Figure 4-12.

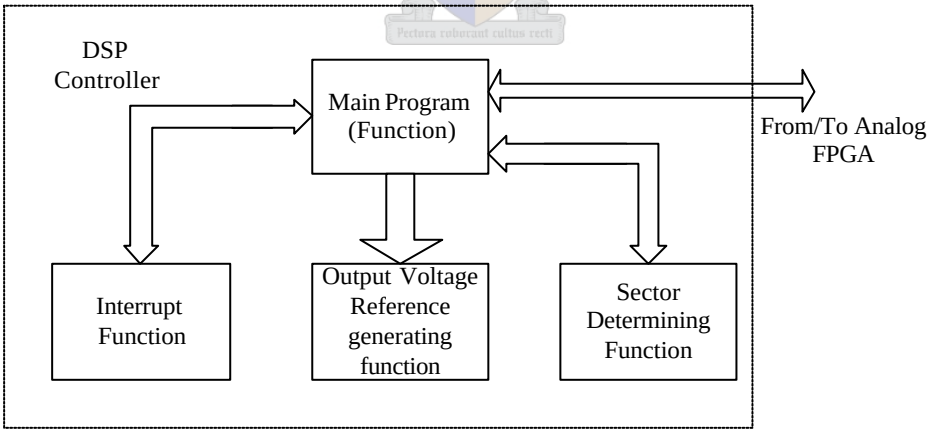


Figure 4-12 A functional block diagram of the DSP controller

Determination of the input phase voltage sector is based on the space vector representation. The three-phase input voltages are represented by two-phase space

vector voltages in the alpha-beta plane as shown in Figure 4-13. The transfer matrix formula from three-phase to two-phase is given in Equation 4-4.

$$\begin{bmatrix} U_a \\ Ub \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

Equation 4-4

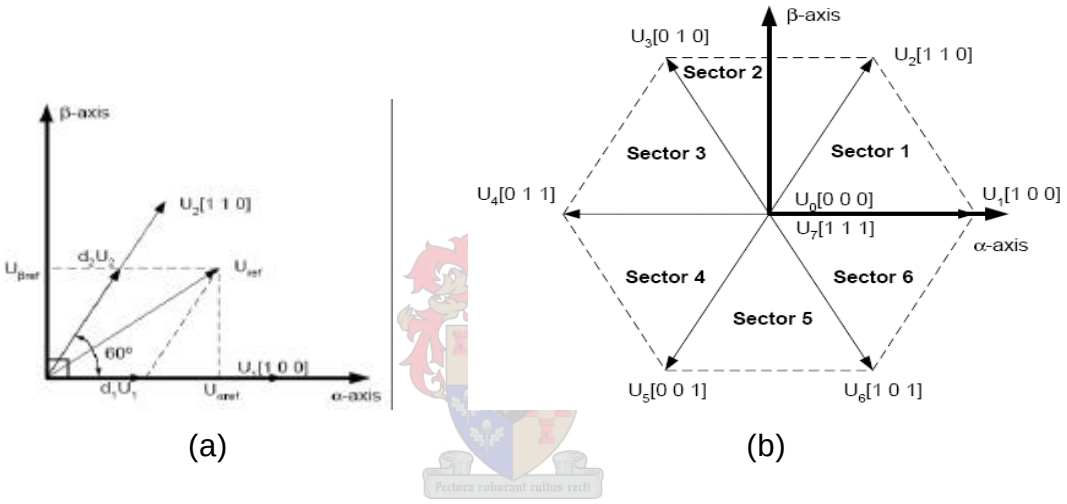


Figure 4-13 Space vector representation (a) The approximated reference voltage vector on the a-ß axes; (b) The six sectors, the non-zero voltage vectors and the zero vector voltages.

The discussion of determining (identifying) the six sectors based on the space vector representation was examined in chapter two. Figure 4-14 shows a flow chart developed to identify these six sectors. In every sector, one maximum and one minimum input phase voltages exist simultaneously. These voltages determine which IGBT switches have to be turned on in that specific sector in order to have a safe commutation between input phase voltages. A complete C code is given in appendix A.1.

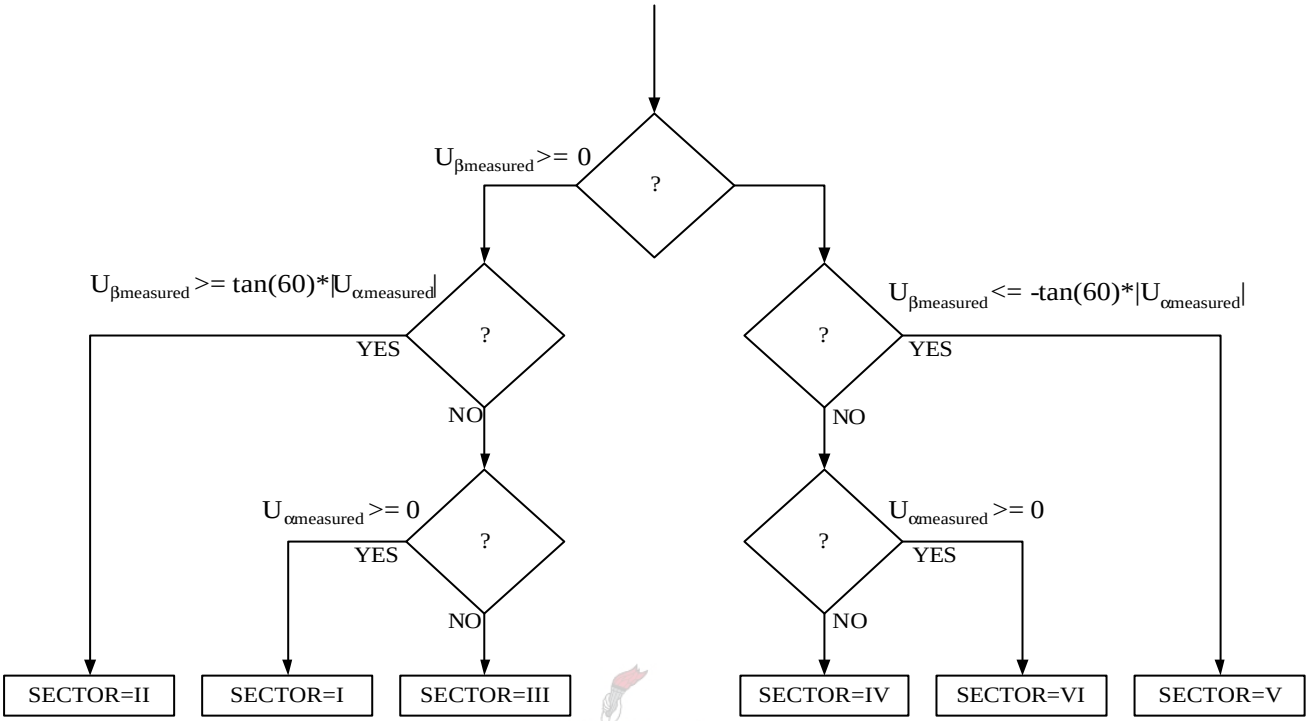
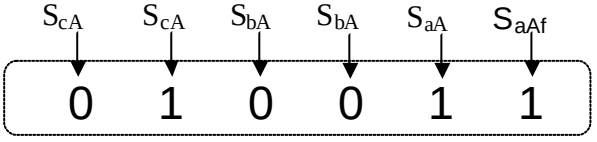


Figure 4-14 A flow chart of identifying a sector in SV representation

4.3.2 FPGA Control Algorithm and Analysis

The two-step safe commutation method used to commutate from one phase to another phase based on the voltage commutation method will be discussed in this subsection. The switching PWM control signals are generated in the FPGA controller of the PEC33 controller board. The FPGA device produces 18 PWM controls for the 9 bi-directional switches. A flow chart of the two-step safe commutation strategy [4] is shown in Figure 4-15. This flow chart shows commutation from one phase to another phase for only one sector since the same approach is applied to the remaining of the sectors. Where '0' means the switch is open and '1' means the switch is closed.



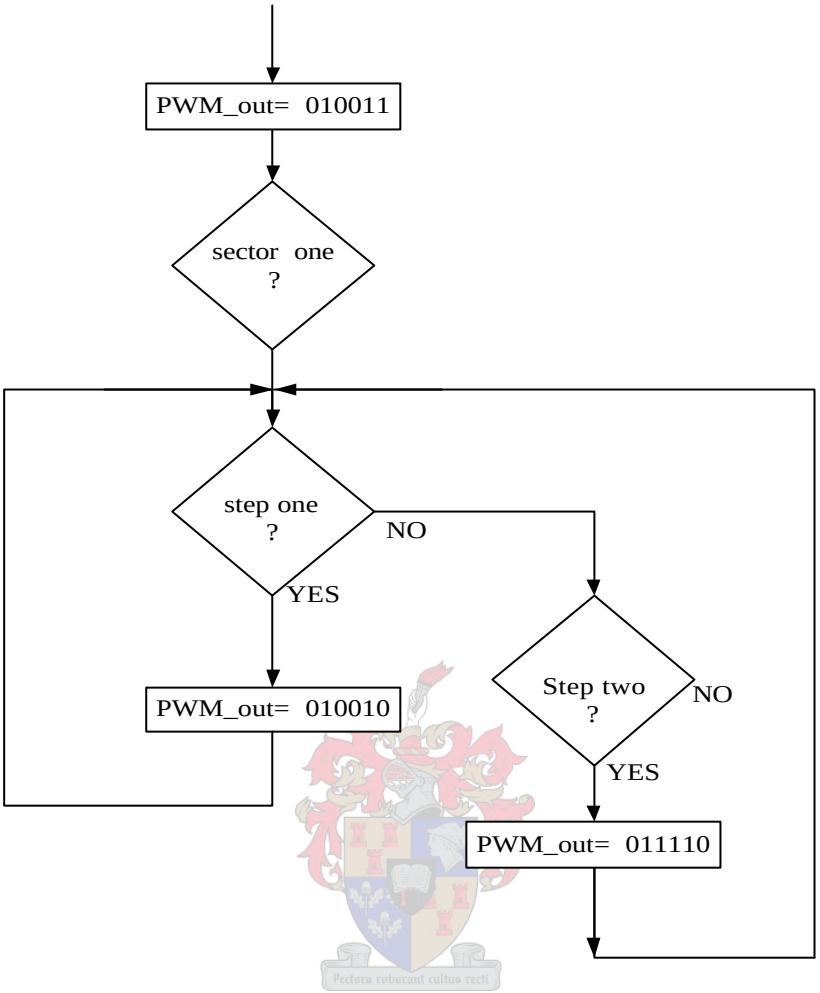


Figure 4-15 A two step voltage commutation

Consider the three-phase input voltages are in sector III as shown in Figure 4-18. In this sector, the maximum and minimum input phases are V_a and V_c respectively in this sector. The corresponding switches gated are reverse switch S_{aAr} and forward switch S_{cAf}, where S_{aAr} is the reverse switch connecting input phase-a to load phase-A and S_{cAf} forward switch connecting input phase-c to load phase-A as shown in Figure 4-16. Since input phase-a is a maximum, gating the reverse switch S_{aAr} will never create any short circuit of the input voltages. The steps followed to change from input phase-a to either phase-b or phase-c depending on the duty cycle calculated is as follows:

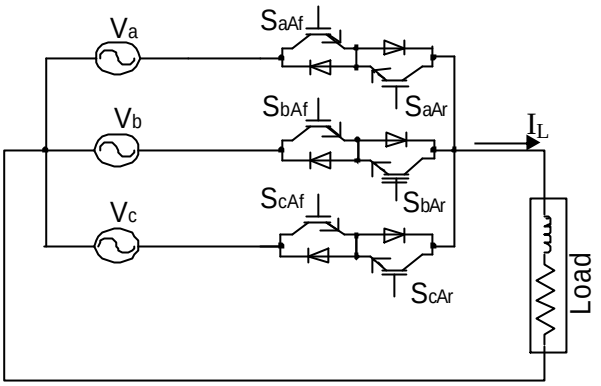


Figure 4-16 A three-phase to single-phase Matrix converter circuit

- Step one: Initially input phase-a could supplying the load current to either positive direction or negative direction. Eventually, when commutation is required, the forward switch S_{aAf} is turned off and a delay of 5us is introduced to ensure the switch is turned off properly. Initially the PWM control output for load phase-A is "01-00-11". Therefore, after the forward switch S_{aAf} is turned off, the PWM control output will be "01-00-10", the reverse switch S_{aAr} and forward switch S_{cAf} are left turned on alone to ensure the flow of load current on either direction. Where xx-xx-xx represents the three-phase bi-directional switches state shown in Figure 4-16. The value of 'x' could either be '0', which represents turned off switch state or '1', which means the switch, is turned on. Figure 4-17 shows which switch state belongs the three bidirectional switches of the converter circuit

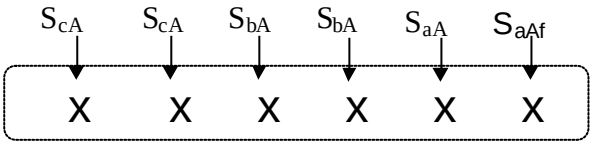


Figure 4-17 Switches state

- Step two: Since the commutation from one phase to another is dependent on the calculated duty cycle values, the commutation could be either to phase-b or to phase-c. The minimum duty cycle value used to commutate between the phases

has to ensure that there will not be any short circuit during commutation. The minimum switching time required for a complete switching between the phases is 10 us. Therefore, the duty cycles have to generate an equivalent value greater than 10 us, so that a complete commutation will occur. An if condition is used to check this duty cycle value and decide whether to commutate to phase-b or phase-c as shown in the flow chart diagram of Figure 4-15. For example, if the commutation is to phase-b, the PWM control output is set to "01-11-10" or else if it is to phase-c then the PWM control output is set to "11-00-10".

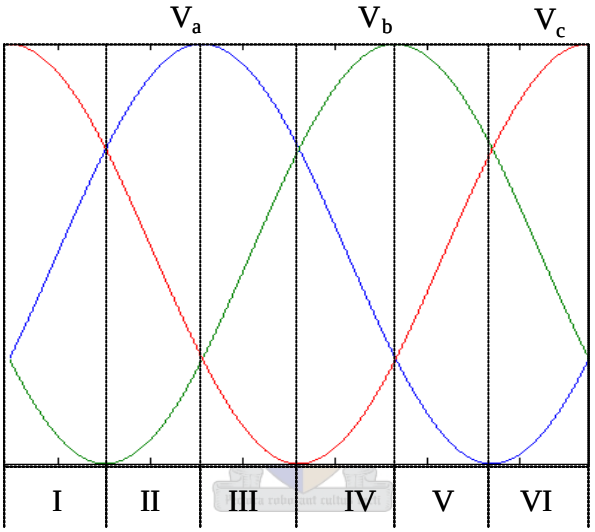


Figure 4-18 A three-phase input voltage and the six sectors

The same approach is applied to commutate from phase-b to either phase-c or phase-a, and from phase-c to either phase-a or phase-b. A complete VHDL code of the switching process between the three input phases to generate the single output load phase-A is given in appendix A.4.

4.4 Summary

The analyses and development of the three-phase-to-three-phase direct matrix converter modulation strategies were presented in this chapter. The Direct Control

algorithm [1-3], [5] described in chapter two was developed based on the two commutation strategies [3-4]. At first the development and analysis of the control algorithm based on the four-step safe current commutation strategy was investigated to provide the PWM switching to the matrix converter. An investigation of the two-step safe voltage commutation strategy was undertaken. The flow chart diagram of the C code and VHDL code developed was presented for both commutation strategies. The current commutation strategy depends on the knowledge of load current direction, while the voltage commutation strategy depends on the knowledge of relative input voltage. Space vector representation was used to identify the relative input section. Reliable knowledge of the input voltage and load current information is required for the control algorithm to operate as desired.

A PEC33 controller board was used to implement the controller algorithms developed. The PEC33 controller board consists of two FPGA devices and a single DSP Device.

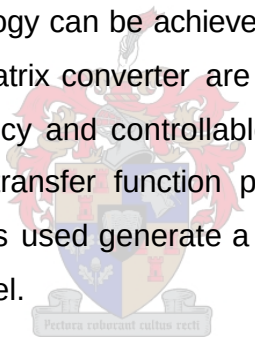
CHAPTER 5

SIMULATION AND PRACTICAL RESULTS OF MATRIX CONVERTER

5 SIMULATION & EXPERIMENTAL (PRACTICAL) RESULTS

5.1 Introduction

The simulation and practical results obtained for the matrix converter circuit are presented in this chapter. The chapter is divided into two main sections; a section that presents the simulation results of both the direct and the indirect matrix converter topologies, while the other section presents the practical results of the 3∅/3∅ AC/AC direct matrix converter topology. Practical results of the indirect matrix converter topology are not included in this section, since this topology is not part of the thesis. The simulation results of this topology are presented to show a comparison of the advantages and disadvantages, as well as to verify whether the desired characteristics of the direct matrix converter topology can be achieved by this topology. Some practical results obtained from the direct matrix converter are presented. Fixed duty cycles that results only 50 Hz output frequency and controllable voltage level, and variable duty cycles calculated based on the transfer function proposed in [1], [2] and [5] were applied. These variable duty cycles used generate a controlled output frequency range and a controlled output voltage level.



The simulation results of the direct matrix converter topology will be examined and are followed by presenting the results of the indirect matrix converter topology. Finally, some of the practical results of the direct matrix will be presented.

5.2 Simulation Results of the Direct Matrix Converter

5.2.1 Introduction

The converter circuit of the 3∅/3∅ AC/AC direct matrix converter setup is shown in Figure 5-1. Four different modulation transfer functions of generating the control switching signals were applied to the ideal switches of the converter. These are a forward transfer function, a reversed transfer function, a combined transfer function with

50% maximum gain, which can able to control the input phase shift of the converter to unity, and the final transfer function with 86.67% maximum gain, as well able to control the input phase shift of the converter to unity. All four-transfer functions are proposed in [1-3]. Table 5-1 shows the specification of the three-phase-to-three-phase direct matrix converter circuit.

Table 5-1 Input/output specifications of the 3Æ/3Æ AC/AC direct matrix converter

Symbol	Parameter	Vlaue	Unit
$3\varnothing$ V_i	Input supply Voltage	400	V
f_{in}	Input frequency	50	Hz
L_{in}	Input inductance	200	µH
C_{in}	Input capacitance	50	µF
R_{in}	Input resistance	0.02	Ω
$3\varnothing$ V_o	Output Voltage	0..200	V
f_o	Output frequency	0..200	Hz
I_o	Output current	0..10	A
R_L	Resistive load	10	Ω
L_o	Output inductance	2000	µH
C_o	Output capacitance	50	µF
f_s	Switching frequency	5	kHz

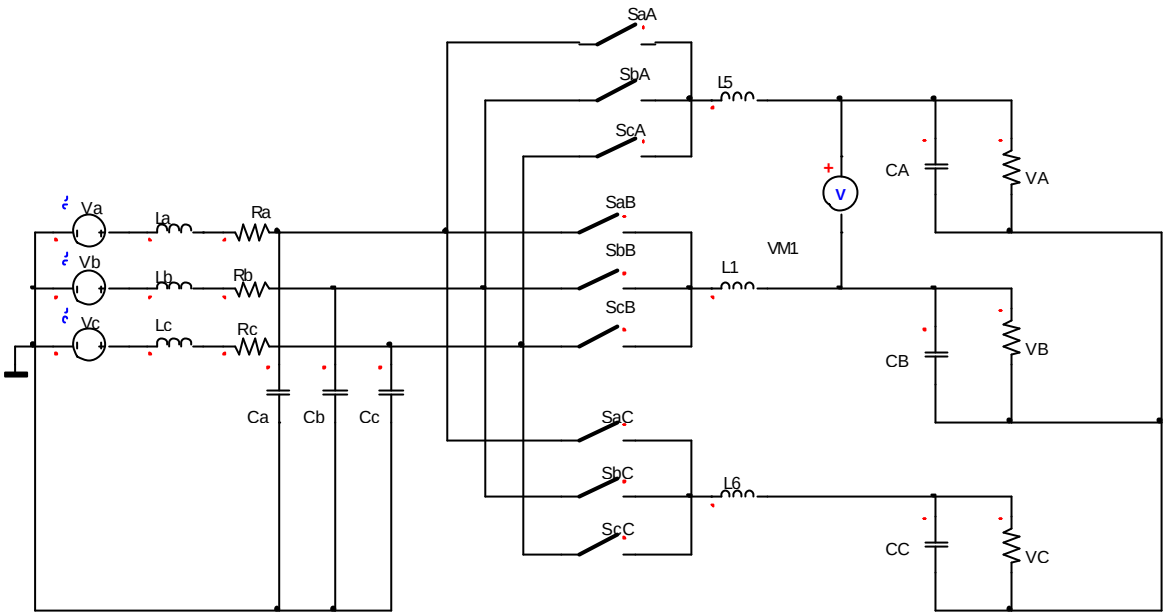
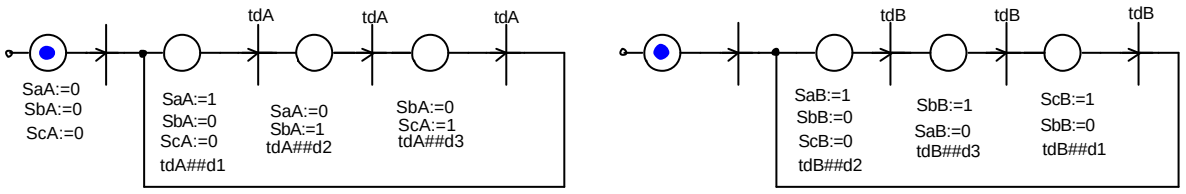


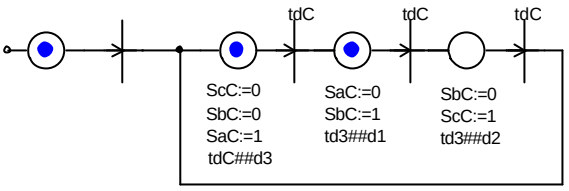
Figure 5-1 The 3Æ/3Æ AC/AC direct matrix converter circuit with resistive load

5.2.2 Forward Transfer Function

Figure 5-1 shows a circuit setup of the 3∅/3∅ AC/AC direct matrix converter circuit consisting of 9 ideal switches connecting the input lines to the output lines of the resistive load. The transfer function given in Equation 5-3 was applied to generate the PWM gating signals, so that a controlled output frequency range and multilevel three-phase output voltages were produced. Three switch state control blocks were used to control the switching of the idle switches as shown in Figure 5-2.



(a) (b)



(c)

Figure 5-2 Switch state control Block of the 9 ideal switches of the direct matrix converter. a) Control block for load phase-A, b) Control block for load phase-B, c) Control block for load phase-C

The three duty cycles of the forward transfer function are given by Equation 5-1 as proposed in [1], [2] and [5]. The timing function is easily calculated by multiplying the switching period T_S and the evaluated duty cycle is given by Equation 5-2.

$$\begin{aligned} d_1 &= 1 + 2 \cdot m \cdot \sin(w_m \cdot t) \\ d_2 &= 1 + 2 \cdot m \cdot \sin(w_m \cdot t - 2 \cdot p/3) \\ d_3 &= 1 + 2 \cdot m \cdot \sin(w_m \cdot t - 4 \cdot p/3) \end{aligned} \qquad \textbf{Equation 5-1}$$

$$t_j = \frac{T_S \cdot d_j}{3}, \quad for\ j \in \{1, 2, and\ 3\} \qquad \textbf{Equation 5-2}$$

Where: d_j is the duty cycle and $w_m = w_o + w_i$ is the modulation function angular frequency, w_o the output angular frequency, $w_i = 2 \cdot p \cdot f_i$ is the input angular frequency and $f_i = 50\ Hz$ is the input frequency, t_j is the switching time, m is the modulation index and $T_S = 200\ ms$ is the switching period.

The timing modulation transfer function for a 3∅/3∅ in a matrix form is given by Equation 5-3.

$$M_1(t) = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_2 & t_3 & t_1 \\ t_3 & t_1 & t_2 \end{bmatrix} \qquad \textbf{Equation 5-3}$$

This timing modulation function was applied to the three switch state control blocks in Figure 5-2 to modulate the ideal switch shown in Figure 5-1. Consider the switch state control block that modulates the three ideal switches S_{aA}, S_{bA} and S_{cA} connecting the three-phase input to the single resistive load of output phase-A shown in Figure 5-2(a). The principle of operation of this switch state control block is, to turn on switch S_{aA}, and turn off the remaining switches S_{bA} and S_{cA} at time 0 for a time delay of t_1. During this time delay, the input voltage V_a supplies the load current. The switch S_{aA} is turned off and simultaneously the switch S_{bA} is turned on at time t_1, and the input voltage V_b supplies the load current for the delay time of t_2. At time $(t_1 + t_2)$, the switch S_{cA} is turned on and the switch S_{bA} is turned off to allow the input voltage V_c supply the load current for the time delay t_3. The same process is repeated every switching period $T_S = 200\ ms$ to generate a chopped voltage of the three-phase input voltages in the output side. Besides, a continuous input current waveform is seen at the input side of the converter.

Figure 5-3 shows the duty cycle waveform results of the simulation for 50 Hz and 100 Hz modulation frequencies and modulation indexes of m = 0.25 and m=0.5. The duty cycle waveforms are similar for a given value of modulation index m of any modulation frequency. However, the period of these waveforms is dependent on the modulation frequency as shown in Figure 5-3. For example, using the same modulation index m, for modulation frequency of 50 Hz and 100 Hz, the waveform is similar as shown in Figure 5-3, but the duty cycle waveform of 100 Hz modulation frequency runs twice as fast as the cycle of the duty cycle waveforms of 50 Hz modulation frequency.

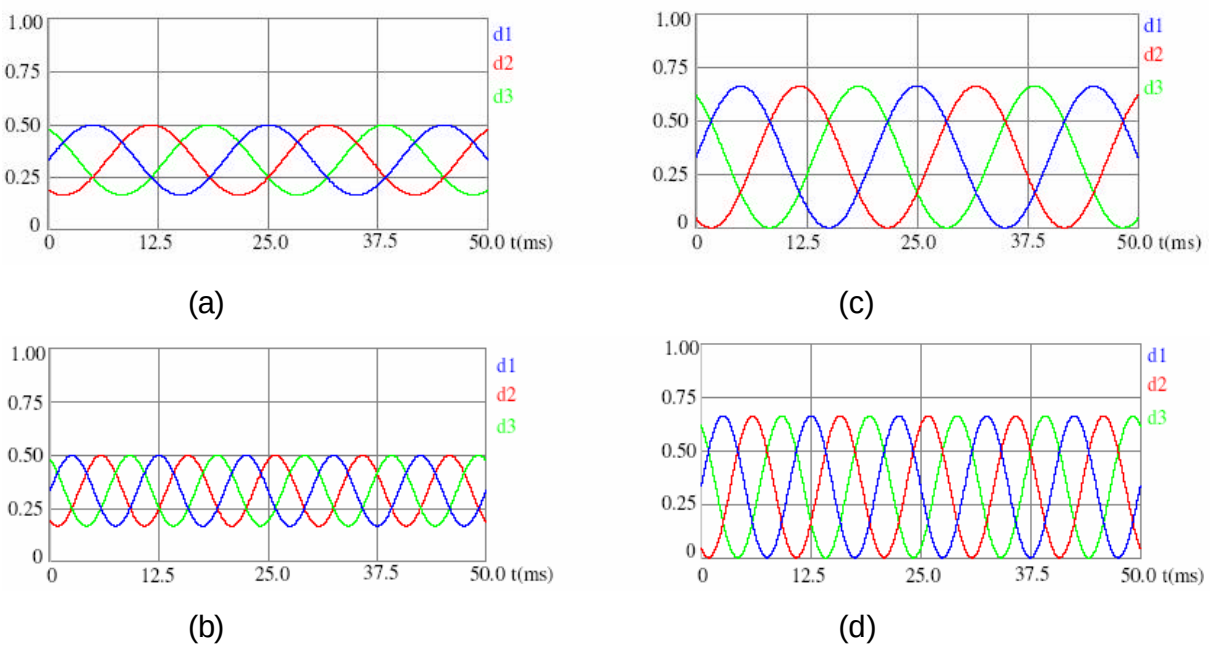


Figure 5-3 Duty cycle values for 50 Hz and 100 HZ modulation frequency. a) f_m = 50 Hz modulation frequency and m = 0.25 modulation index, b) f_m = 100 Hz modulation frequency and m = 0.25 modulation index, c) f_m = 50 Hz modulation frequency and m = 0.50 modulation index, d) f_m = 100 Hz modulation frequency and m = 0.50 modulation index

Simulation results obtained applying the above duty cycle waveforms for various output frequencies will be presented. To show the capability of the direct matrix converter circuit generating multi-range output frequency (0..200 Hz), various results of the simulation for 25 Hz, 50 Hz, 100 Hz, and 200 Hz output frequencies are presented below. The converter was connected to a three-phase 400V/50Hz supply. The maximum output-input transfer ratio used was 0.5. Unbalanced and distorted output waveform was generated for transfer ratios higher than m =0.5.

Figure 5-4 shows the DC output voltage waveforms obtained from the simulation. The modulation frequency f_m used was 50 Hz. The summation of the three DC output voltages of Figure 5-4(a) is zero, which means it is a balanced system. Two of the output voltages are negative and the remaining output voltage is positive. Even though

the generated output voltage waveform was DC, the input current was a continuous sinusoidal waveform as shown in Figure 5-4 (b). Figure 5-4(c) shows the input voltage, lagging input current supplied, and DC output voltage waveforms. It is not possible to control the lagging power factor of the converter using the forward transfer function of Equation 5-1, 5-2 and 5-3.

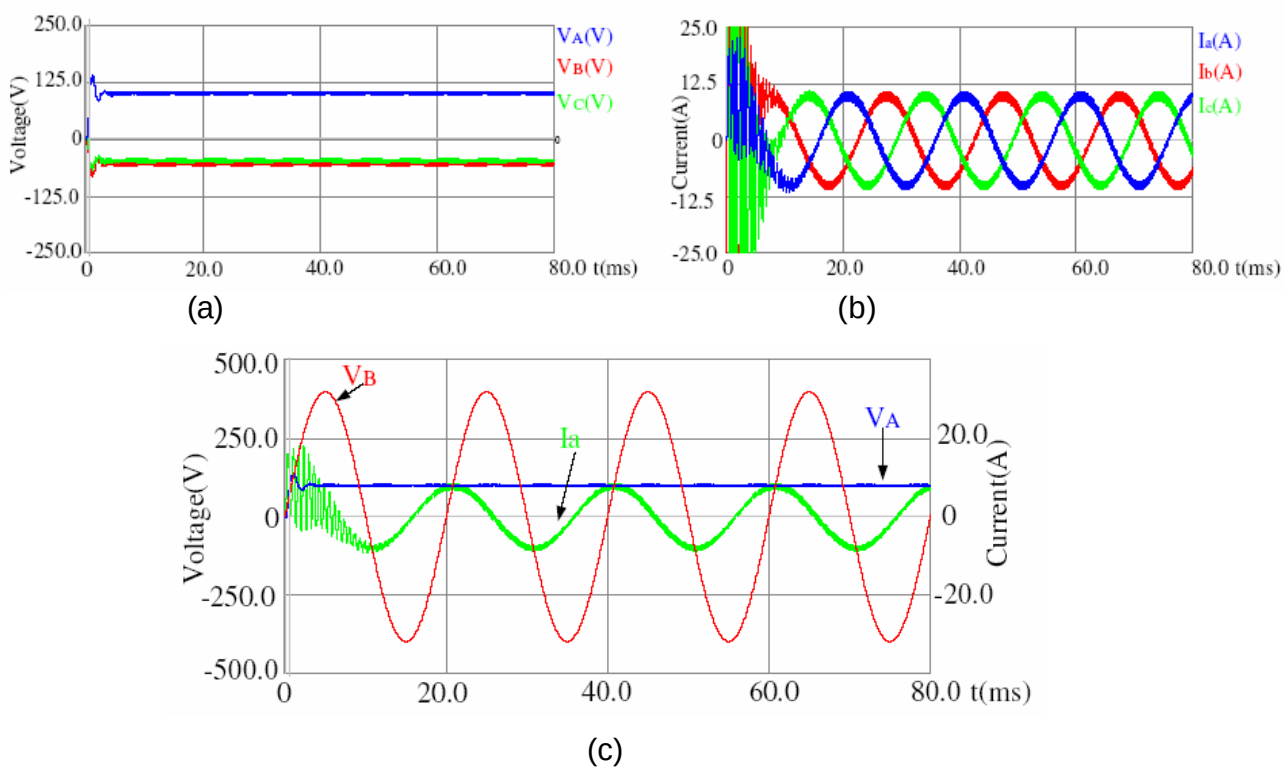
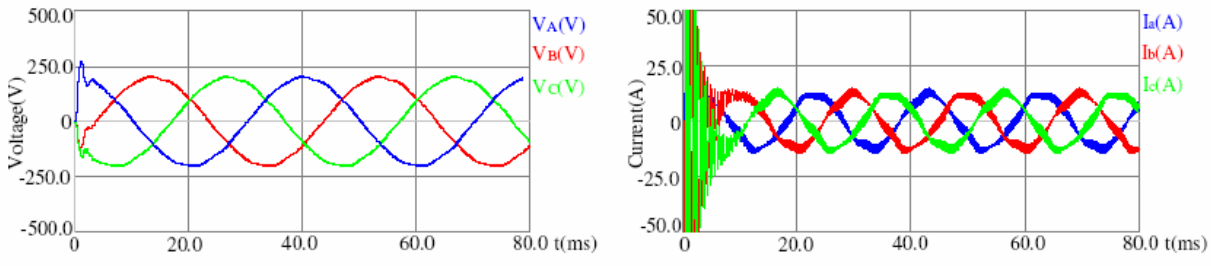
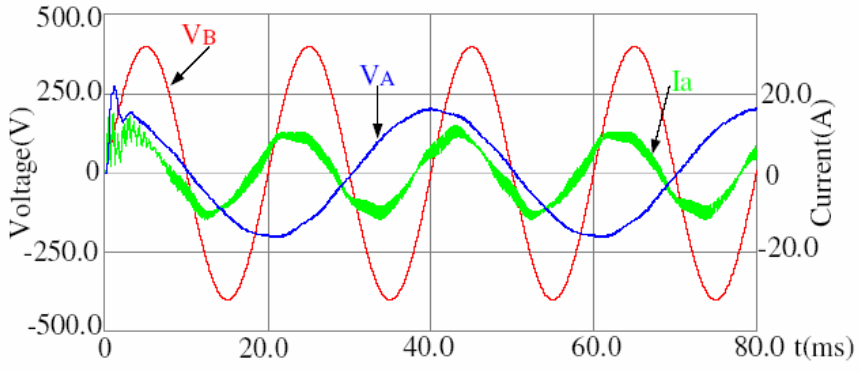


Figure 5-4 Output and input waveforms for f_o =0z. (a) Three DC output voltages, where V_A is positive, while V_B and V_C are negative; (b) Three-phase input currents; (c) Green-Input current, Red-input voltage and Blue-DC output voltage

Figure 5-5 shows simulation results of the three-phase output voltage waveform of 25 Hz output frequency and m=0.5 modulation index generated by the direct matrix converter. A sinusoidal input current and output voltage waveforms were produced as shown in Figure 5-5(a) and Figure 5-5(b) respectively.



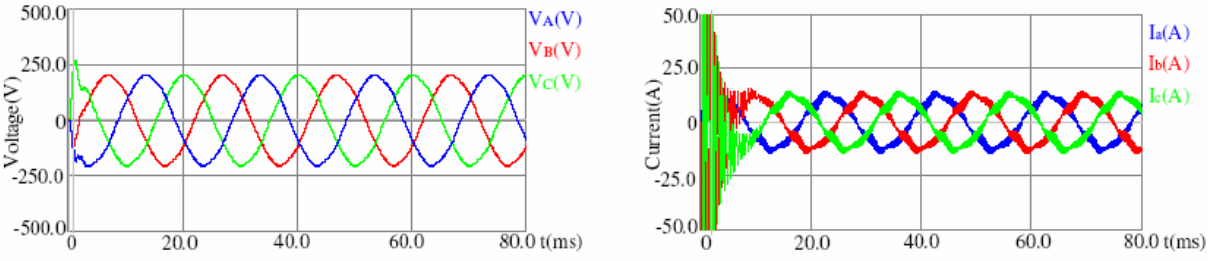
(a) (b)



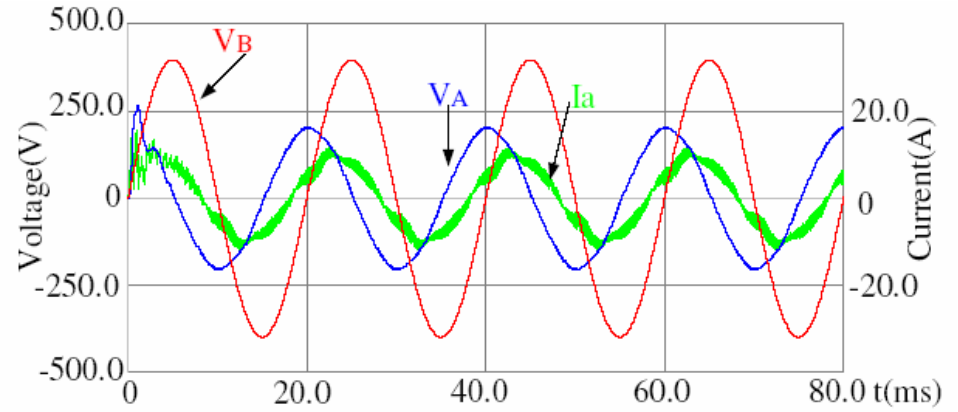
(c)

Figure 5-5 Output and input waveforms for f_o =25 Hz. a) 3Æ output voltages, b) 3Æ input currents c) green-Input current, red-input voltage and blue -output voltage.

Figure 5-6 shows the three-phase output voltage waveform generated and three-phase input current supplied for a 50 Hz output frequency and modulation index m=0.5. The output voltage waveforms are shown in Figure 5-6(a), and the input current waveforms are shown in Figure 5-6(b).

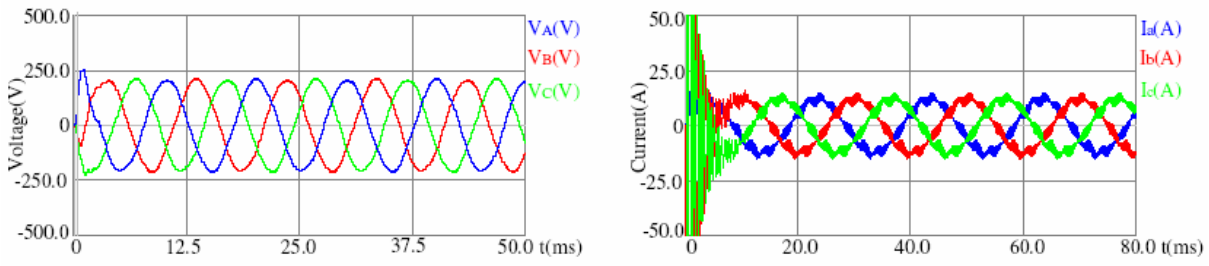


(a) (b)

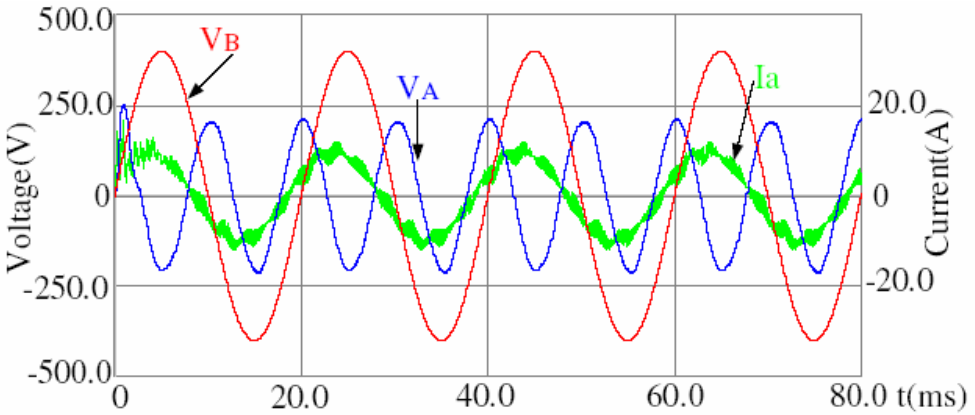


(c)

Figure 5-6 Output and input waveforms for f_o =50 Hz. a) 3Æ output voltages, b) 3Æ input currents c) green-Input current, red-input voltage and blue-output voltage.



(a) (b)



(c)

Figure 5-7 Output and input waveforms for f_o =100 Hz. a) 3Æ output voltages, b) 3Æ input currents c) green-Input current, red-input voltage and blue-output voltage.

Figure 5-7 shows simulation waveforms generated for a 100 Hz output frequency and m=0.5 modulation index. The output voltage waveforms are shown in Figure 5-7(a), and the input current waveforms are shown in Figure 5-7(b).

Figure 5-8 shows simulation waveforms generated for a 200 Hz output frequency and modulation index m=0.5. The output voltage waveforms are shown in Figure 5-8 (a), and the input current waveforms are shown in Figure 5-8 (b). Figure 5-8(c) shows the input phase voltage, input current, and the output voltage waveforms.

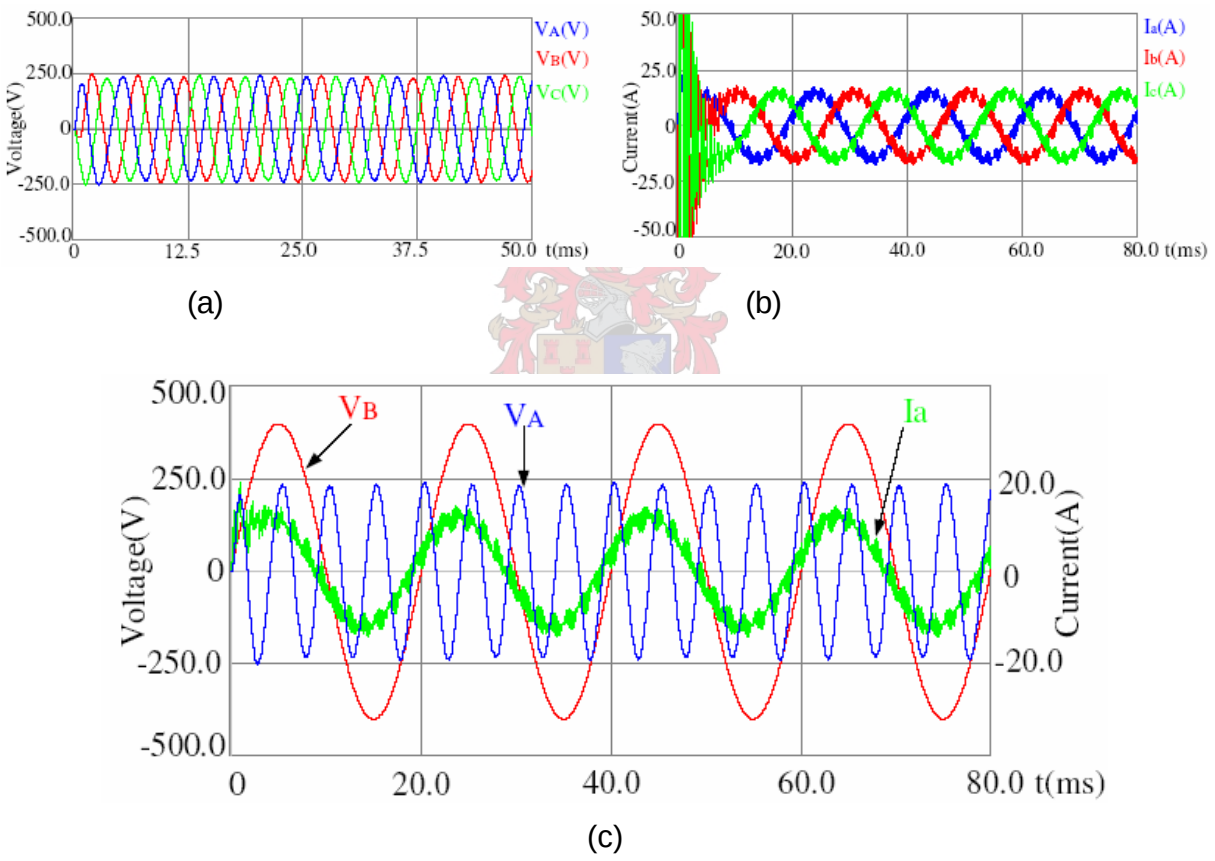


Figure 5-8 Output and input waveforms for f_o =200 Hz. a) 3Æ output voltages, b) 3Æ input currents c) green-Input current, red-input voltage and blue-output voltage.

5.2.3 Reversed Transfer Function

The same circuit setup shown in Figure 5-1 was set up; the 3∅/3∅ direct matrix converter circuit consisting of 9 ideal switches that connect the input lines to the output lines of the resistive load. The simulation results shown in Figure 5-9 were obtained, when the reversed transfer function given in Equation 5-4 was applied. The switching times t_1, t_2 and t_3 are the same for both the forward transfer and reversed transfer function, except that the arrangement in the matrix function differ as given by Equation 5-4 and Equation 5-5.

$$M_2(t) = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_3 & t_1 & t_2 \\ t_2 & t_3 & t_3 \end{bmatrix} \qquad \textbf{Equation 5-4}$$

$$M_1(t) = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_2 & t_3 & t_1 \\ t_3 & t_1 & t_2 \end{bmatrix} \qquad \textbf{Equation 5-5}$$

The modulation angular frequency w_m for the reversed transfer function is given by Equation 5-6.

$$w_m = w_0 - w_i \qquad \textbf{Equation 5-6}$$

Since the duty cycles of both the forward and the reversed transfer functions are the same, the simulation results are similar, except for a slight difference because of the different modulation frequencies of the transfer functions. Therefore, results of 50Hz and 100 Hz output frequencies are only presented, as shown in Figure 5-9.

Figure 5-9(a) and (b) show the output voltage and input current waveforms of the 50 Hz output frequency, while Figure 5-9(c) and (d) present the output voltage and input current waveforms of 100 Hz output frequency.

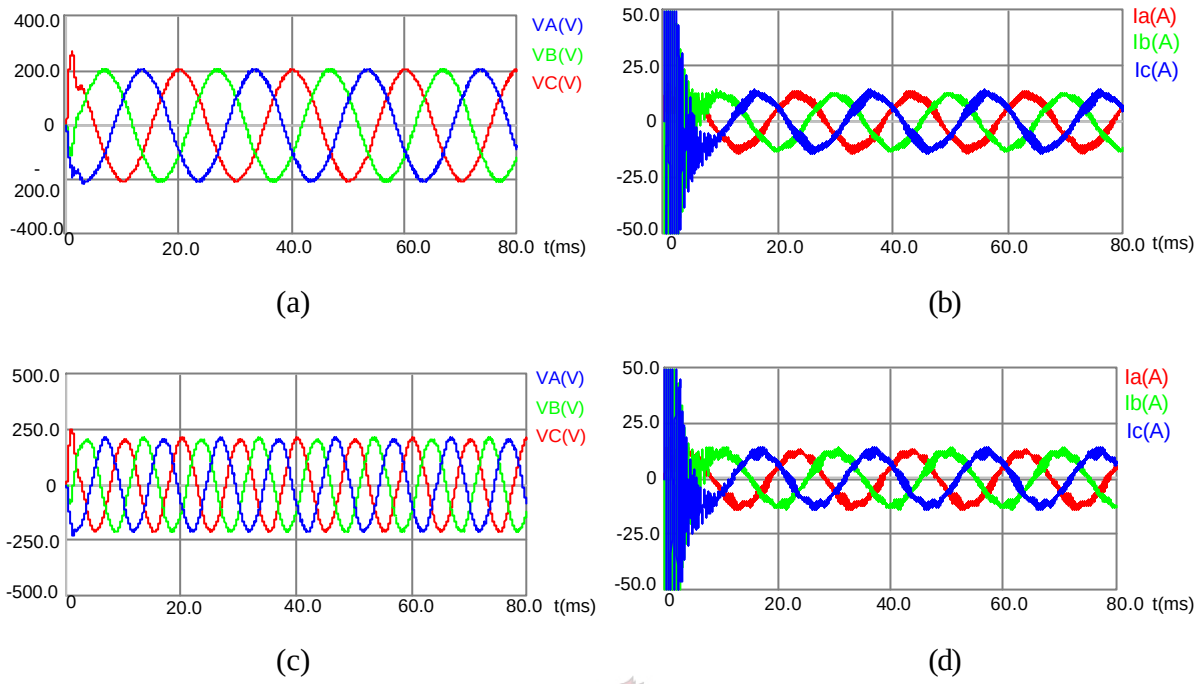


Figure 5-9 Output and input waveforms of matrix converter. a) 3Æ output voltages of 50 Hz output frequency, b) 3Æ input currents of 50 Hz output frequency c) 3Æ output voltages of 100 Hz output frequency, d) 3Æ input currents of 100 Hz output frequency.

5.2.4 Combined Transfer Function of Forward and Reverse Transfer Function

The two modulation transfer functions described in the above sections are added to produce a combined transfer function given by Equation 5-8 proposed in [1-3], [5]. This formula is applied to generate the PWM gating signal for the ideal switches of the 3∅/3∅ AC/AC direct matrix converter circuit setup shown in Figure 5-1. The transfer function has the ability to generate a unity power factor in the input side of the converter regardless of the load connected, and a maximum gain of 50%.

$$d_{kj}(t) = \frac{1}{3}\left(1 + \frac{2 \cdot V_j(t) \cdot V_k(t)}{V_m{}^2}\right), \quad for\ j \in \{A, B,\ and\ C\}\ and\ k \in \{a,\ b,\ and\ c\}$$ **Equation 5-7**

$$t_{kj}(t) = d_{kj}(t) \cdot T_S, \quad for\ j \in \{A, B,\ and\ C\}\ and\ k \in \{a,\ b,\ and\ c\}$$ **Equation 5-8**

Where $t_{kj}(t)$ is the modulation time, $d_{kj}(t)$ is the duty cycles, T_S is the switching period, $V_j(t)$ is the desired reference output voltage, and $V_k(t)$ is the input voltages.

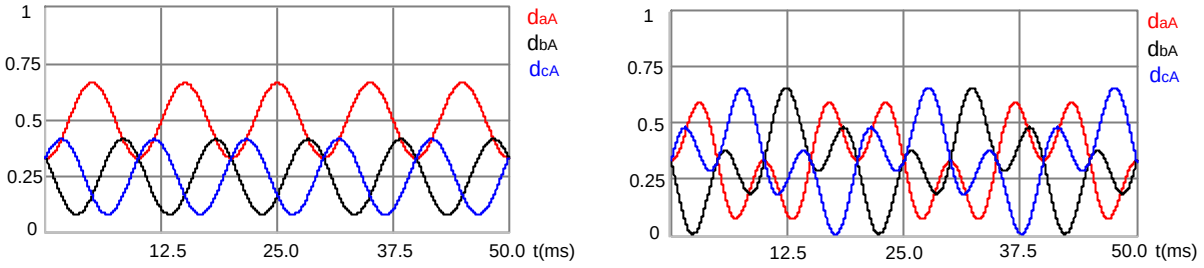
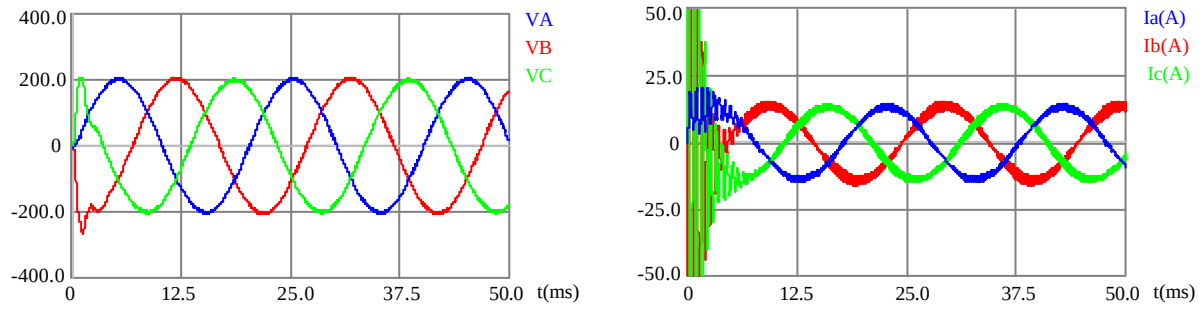
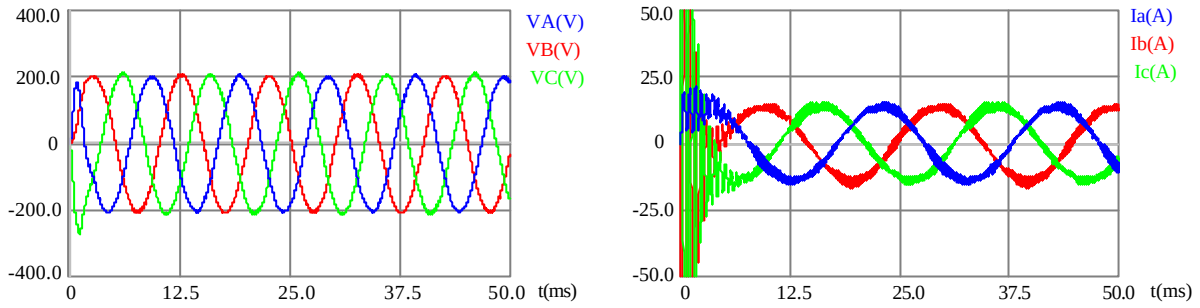


Figure 5-10 Duty cycle waveforms for f_o 50 Hz and 100 Hz. a) f_o = 50 Hz modulation frequency and m = 0.5 modulation index, b) f_o = 100 Hz modulation frequency and m = 0.5 modulation index.

Figure 5-10 shows the duty cycle waveforms applied to produce a single-phase output voltage for a 50 Hz and a 100 Hz output frequencies, and a maximum modulation index m =0.5. The remaining 0 to 200 Hz frequency ranges produce similar waveforms, except that the periods of the waveforms differ. The duty cycle waveforms of the 50 Hz output frequency are shown in Figure 5-10(a). Figure 5-10(b) shows the waveform of the duty cycles of the 100 Hz output frequency.

The various waveforms of the simulation obtained applying the duty cycles in Figure 5-10 for the 50 Hz and 100 Hz output frequency are shown in Figure 5-11.





(c) (d)

Figure 5-11 Output and input waveforms of matrix converter. a) 3Æ output voltages of 50 Hz output frequency, b) 3Æ input currents of 50 Hz output frequency c) 3Æ output voltages of 100 Hz output frequency, d) 3Æ input currents of 100 Hz output frequency.

5.2.5 86.67% Maximum Transfer Ratio of the Input Voltage

A third harmonic components of the input and output voltages are added into the combined transfer function to give 86.6% maximum gain of the input voltage as an output voltage of the direct matrix converter circuit. Equation 5-9 is the transfer function that results in a maximum gain of 86.6%. This transfer function is also able to produce unity power factor in the input side of the converter independent of the connected load. The simulation results obtained for several output frequencies are presented below.

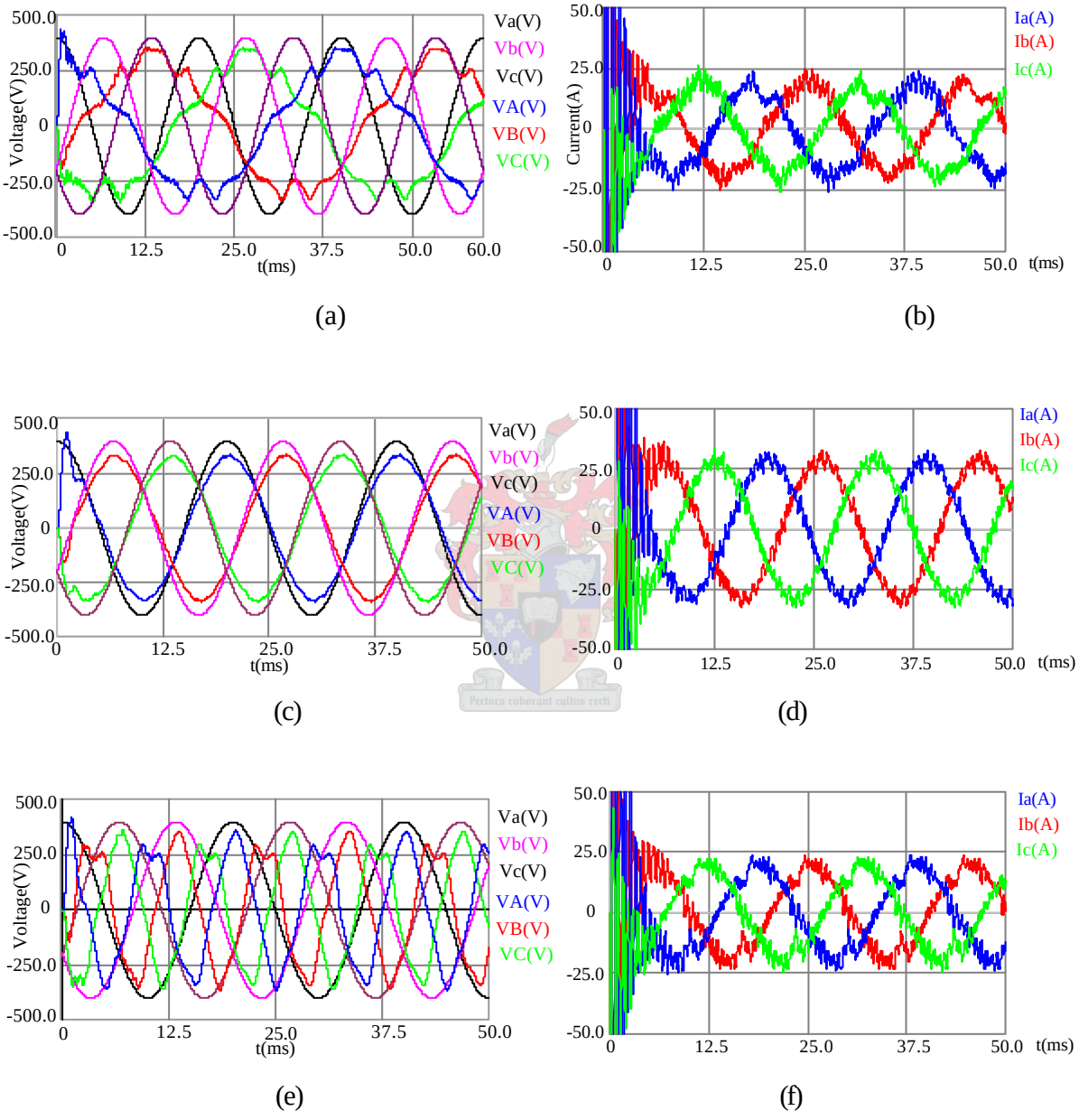
$$d_{kj}(t) = \frac{1}{3}\left(1 + \frac{2 * V_j * V_k}{V_{im}^{\ 2}} + \frac{4m \cdot sin(3\omega t + b_k)sin(\omega t)}{3\sqrt{3}}\right) \qquad \textbf{Equation 5-9}$$

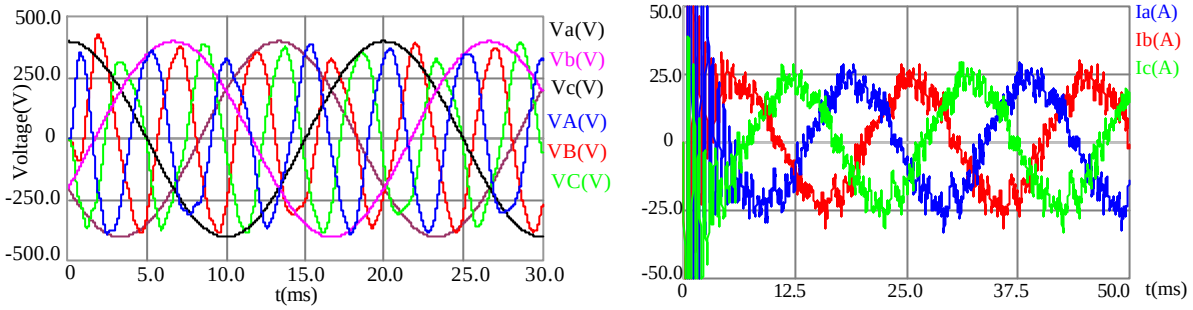
$$b_k = \{0,\ 2p/3,\ and\ \ 4p/3\},\ \text{for}\ k \in \{a,\ b,\ and\ \ c\}\ and\ j \in \{A,\ B,\ and\ \ C\}$$

$$t_{kj}(t) = d_{kj}(t) \cdot T_S \qquad \textbf{Equation 5-10}$$

Where: $t_{kj}(t)$ is the modulation time, $d_{kj}(t)$ is the duty cycle, m is the modulation index, T_S is the switching period, $V_j(t)$ is the desired reference output voltage, and $V_k(t)$ is the input voltage.

Figure 5-12 shows the waveform results obtained for the transfer function given by Equation 5-10. Modulation index of 0.866 was used to obtain the simulation results. These results are similar to the results proposed in [1-3].

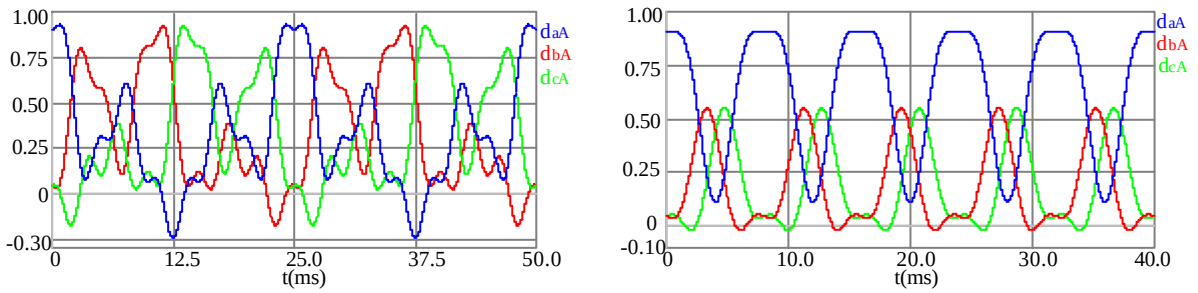




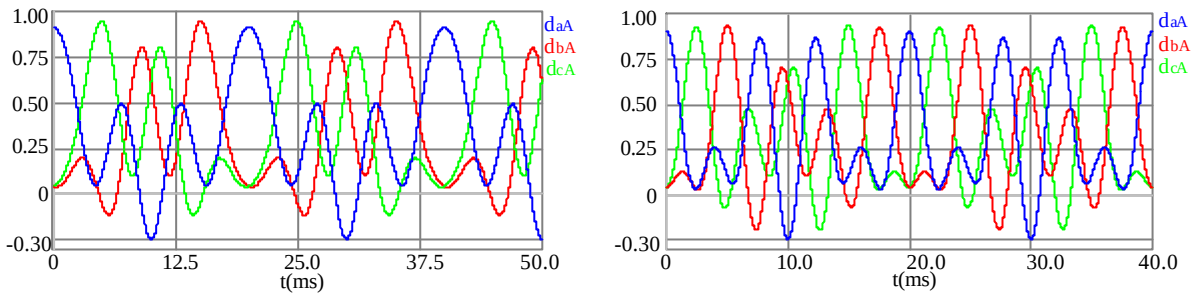
(g) (h)

Figure 5-12 Output and input waveforms of matrix converter. (a) 3Æ output voltages of 25 Hz output frequency and m=0.866; (b) 3Æ input currents of 25 Hz output frequency and m=0.866; (c) 3Æ output voltages of 50 Hz output frequency and m=0.866; (d) 3Æ input currents of 50 Hz output frequency and m=0.866; (e) 3Æ output voltages of 100 Hz output frequency and m=0.866; (f) 3Æ input currents of 100 Hz output frequency and m=0.866; (g) 3Æ output voltages of 200 Hz output frequency and m=0.866; (h) 3Æ input currents of 200 Hz output frequency and m=0.866.

The addition of the third harmonic components of the input and output voltages into the components of the output voltages influence the duty cycle value to achieve values lower than zero. Figure 5-13 shows the waveform results for modulation index of m=0.866. The duty cycle values explain the reason of the distorted output waveforms obtained.



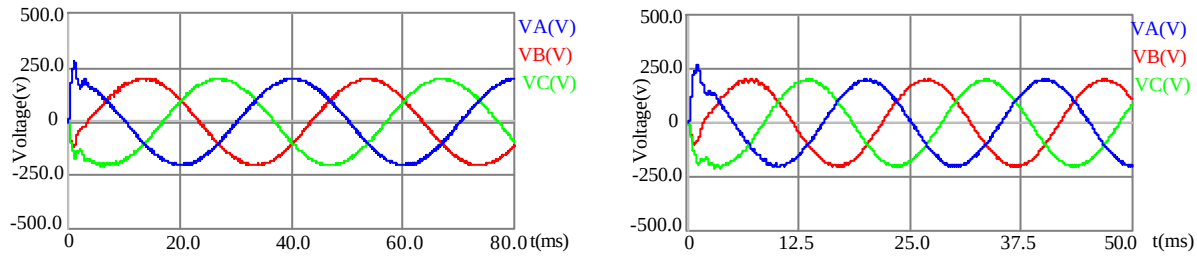
(a) (b)



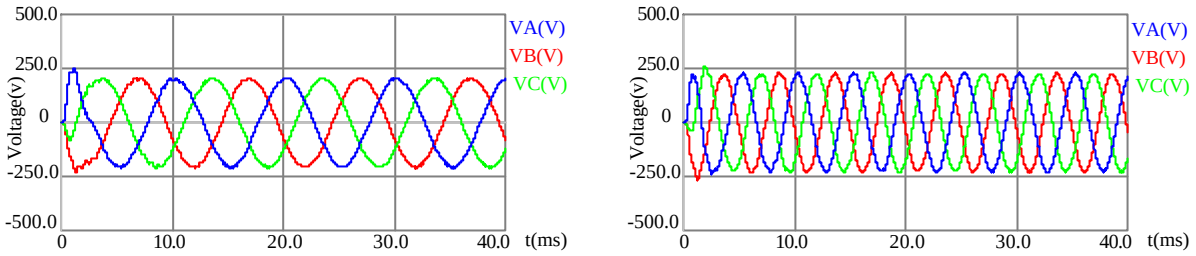
(c) (d)

Figure 5-13 Duty cycle waveforms. (a) 25 Hz output frequency and 0.866 modulation index; (b) 50 Hz output frequency and 0.866 modulation index; (c) 100 Hz output frequency and 0.866 modulation index; (d) 200 Hz output frequency and 0.866 modulation index.

Undistorted and balanced sinusoidal output voltage waveform results are displayed in Figure 5-14. These results show the waveforms generated using the combined transfer function and the transfer function with 86.6% gain for the same modulation index m less than 0.5 are similar. Figure 5-14(a) shows the generated output voltage waveforms of a 25 Hz output frequency, Figure 5-14(b) shows the output waveform results of a 50 Hz output frequency. The output voltage waveforms for a 100 Hz and a 200 Hz output frequency are shown in Figure 5-14(c) and Figure 5-14(d) respectively. The modulation index used to generate these results is m =0.5.



(a) (b)

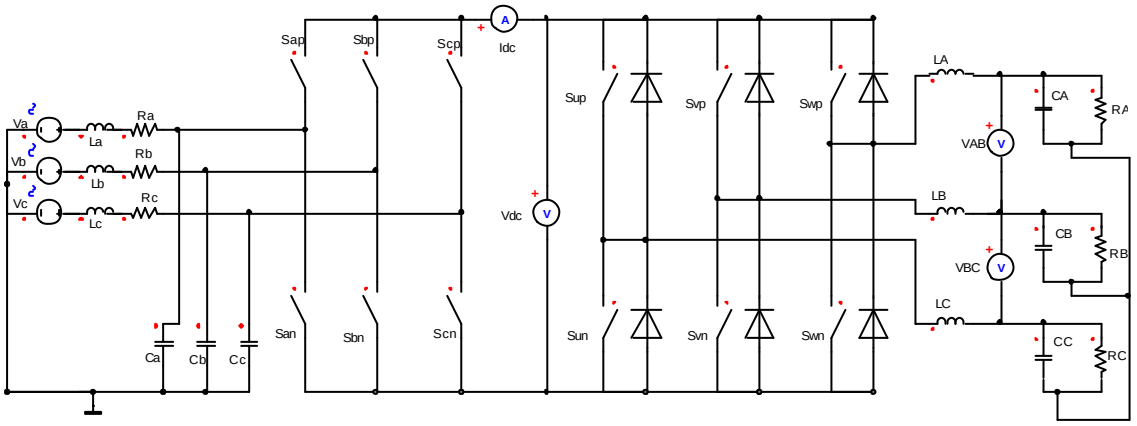


(c) (d)

Figure 5-14 Output waveforms of direct matrix converter (a) 3Æ output voltages of 25 Hz output frequency m=0.5; (b) 3Æ output voltages of 50 Hz output frequency and m=0.5; (c) 3Æ output voltages of 100 Hz output frequency and m=0.5; (d) 3Æ output voltages of 200 Hz output frequency and m=0.5

5.3 Simulation Results of Indirect Matrix Converter

The indirect matrix converter circuit shown in Figure 5-15 was set up to generate various output frequencies using the modulation strategy proposed in [6]. The circuit setup consists of a rectifier side and an inverter side. The rectifier side switches applied the relative input voltage method of generating the PWM gating technique. The SV modulation technique was applied into the inverter switches. Figure 5-15(a) shows the indirect matrix converter circuit set up, The PWM control generating state block of the rectifier switches are shown in Figure 5-15(b), and Figure 5-15(c) shows the SV PWM generating state block for the inverter switches. The converter circuit was connected to a three-phase 326 V/ 50 Hz supply.



(a)

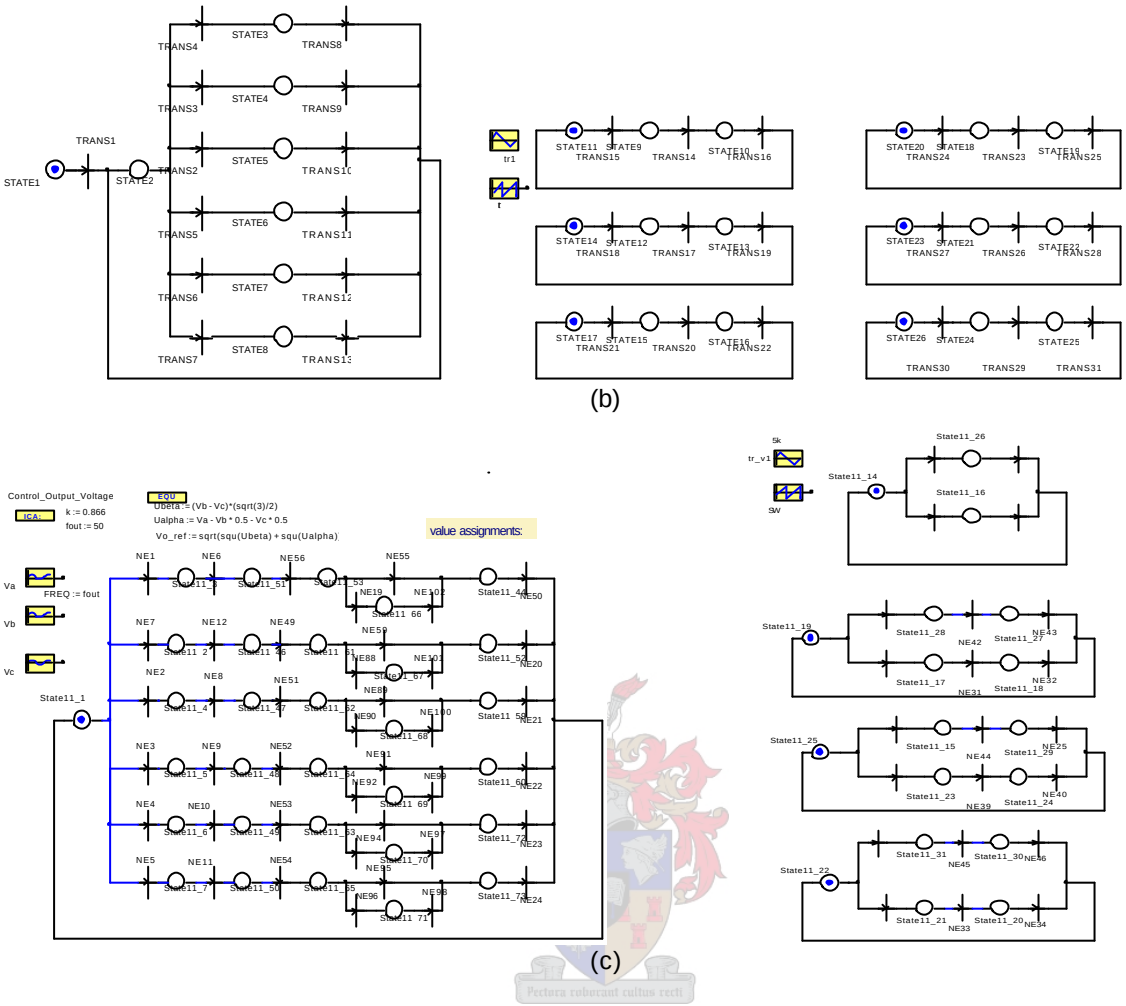
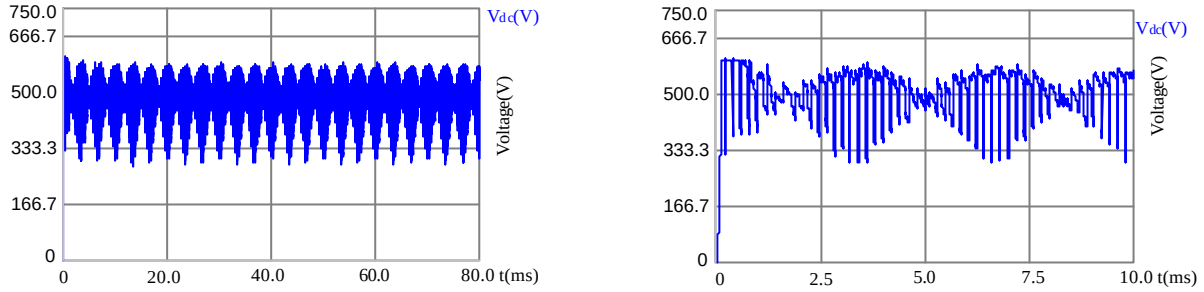


Figure 5-15 The indirect matrix converter circuit setup. a) 3Æ-3Æ converter circuit; b) State control block of the rectifier switches, and c) State control block of the inverter switches.

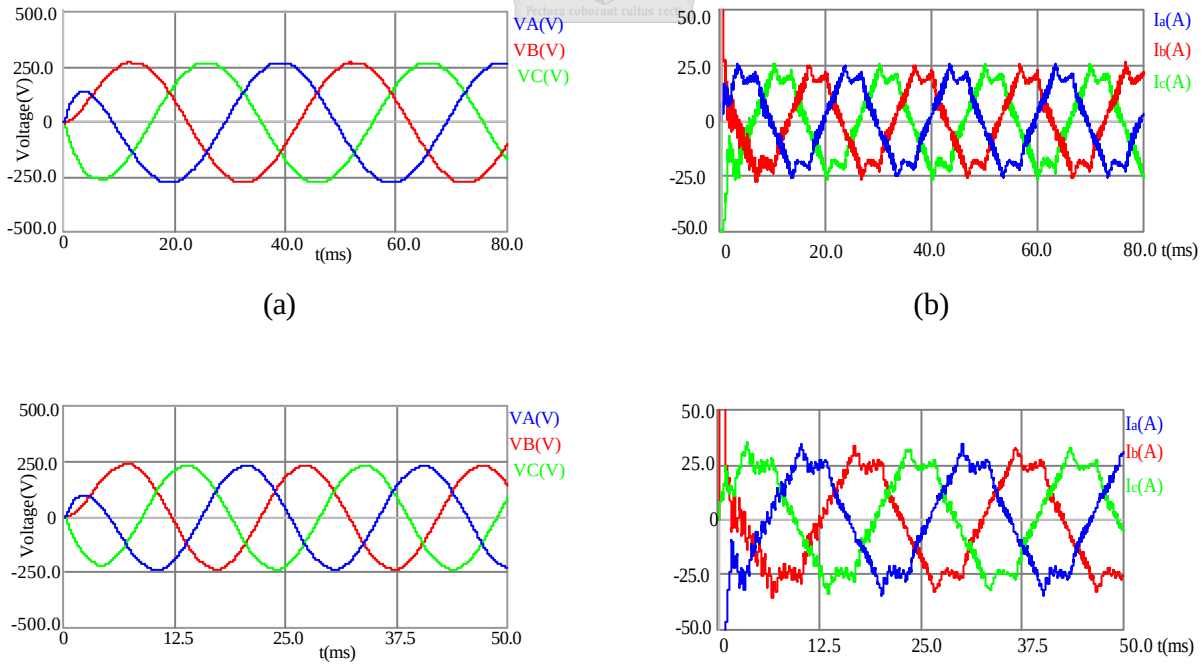
The DC output voltage waveform generated by the rectifier of the converter circuit is shown in Figure 5-16. The rectifier switches were controlled to produce the continuous three-phase sinusoidal input currents displayed in the subsequent figures.



(a) (b)

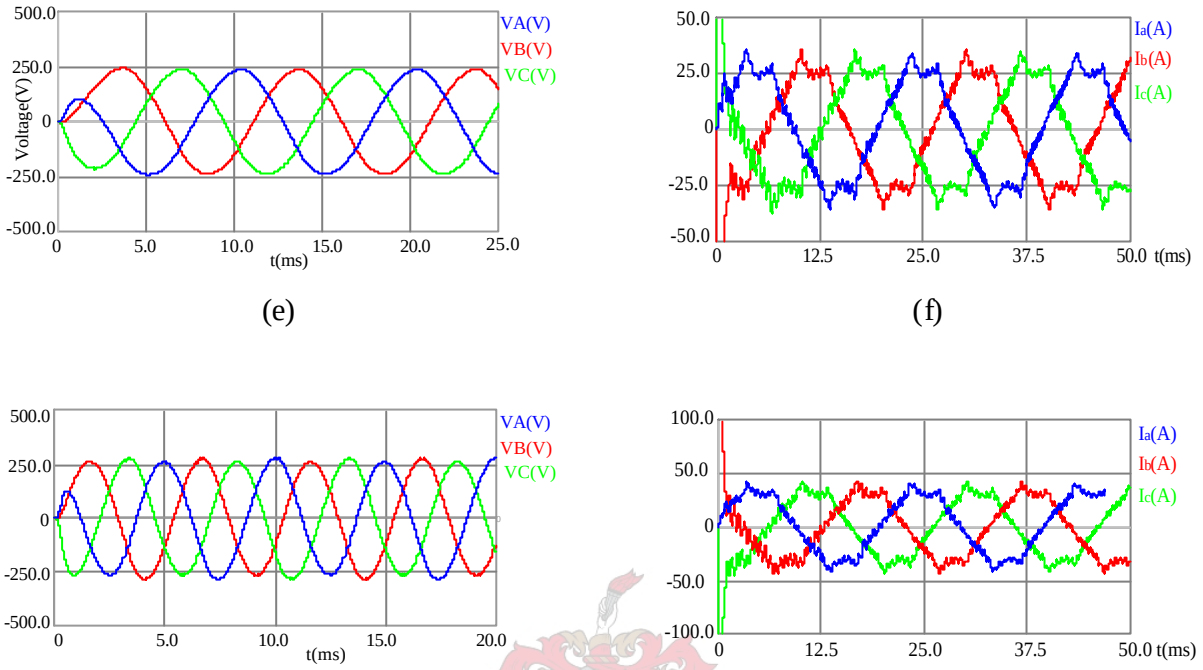
Figure 5-16 The DC output voltage of the rectifier of the converter

The same controlling strategy known as the SV PWM modulation used on a standard inverter was applied on the inverter side of the indirect matrix converter circuit. The results of this inverter were three-phase sinusoidal output voltages. The waveforms obtained from the simulation of the inverter for various output frequencies and the input currents to the rectifier are shown in Figure 5-17. The output voltage and input current waveforms for a 25 Hz output frequency are shown in Figure 5-17(a) and (b) respectively. Figure 5-17(c), (d), (e), (f) and (g) show the output voltage and input current waveform results for a 50 Hz, 100 Hz and 200 Hz output frequency respectively.



(a) (b)

(c) (d)



(e) (f)

(g) (h)

(h)

Figure 5-17 Waveforms generated by the indirect matrix converter circuit. (a) 3Æ output voltages of 25 Hz output frequency and m=0.866; (b) 3Æ input currents of 25 Hz output frequency and m=0.866; (c) 3Æ output voltages of 50 Hz output frequency and m=0.866; (d) 3Æ input currents of 50 Hz output frequency and m=0.866; (e) 3Æ output voltages of 100 Hz output frequency and m=0.866; (f) 3Æ input currents of 100 Hz output frequency and m=0.866; (g) 3Æ output voltages of 200 Hz output frequency and m=0.866; (h) 3Æ input currents of 200 Hz output frequency and m=0.866.

5.4 Practical Results of the Direct Matrix Converter

5.4.1 Introduction

Figure 5-18 was set up to generate a sinusoidal three-phase multilevel voltage and multi frequency output. The control strategy applied for switching the bi-directional switches of the converter is proposed in [1-2] and [5]. The converter was connected to a three-

phase 250V/50Hz and to a resistive load of 10Ω. Various results of the test are presented in the subsequent subsections. The schematic diagram of the power converter circuit is shown in appendix C. The complete functional block diagram of the three-phase-to-three-phase AC/AC direct matrix converter topology, which includes all the components, is also given in appendix D.

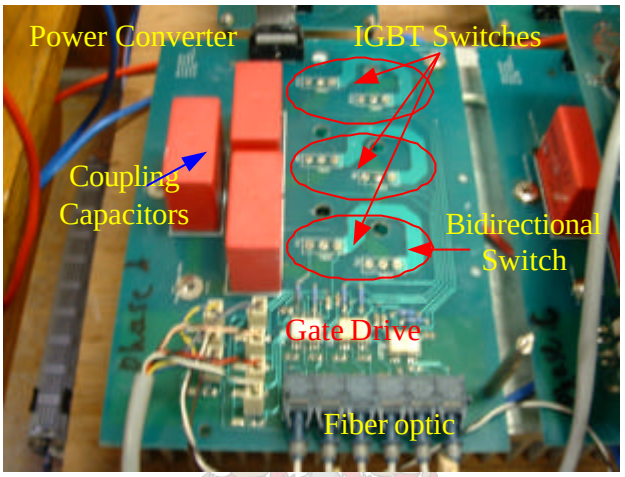


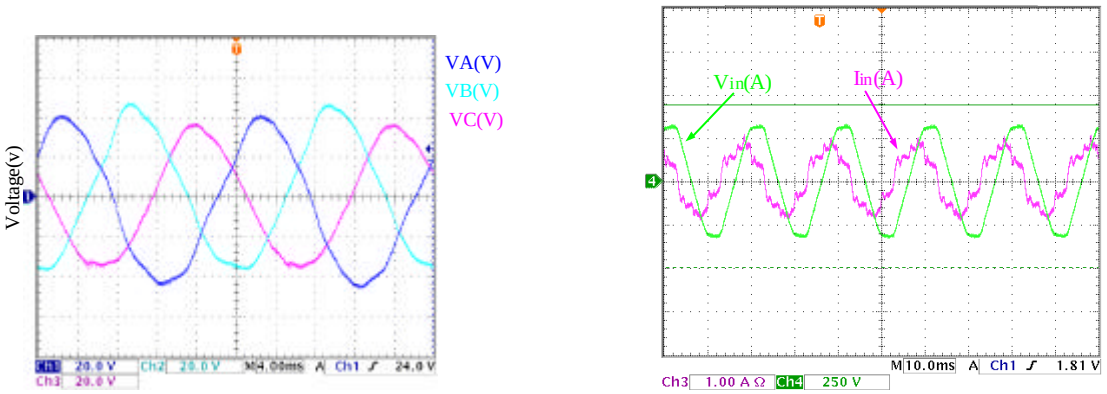
Figure 5-18 The power converter circuit of direct matrix converter.

5.4.2 Experimental Results of Fixed Duty Cycle

5.4.2.1 Based on Current Commutation Strategy

Test results of the direct matrix converter topology based on the current commutation method proposed in [3] and [4] are presented in here. Since fixed duty cycle values were applied to generate the PWM gating signal, the result generated was 50 Hz output frequency only. Figure 5-19 shows experimental results of the output voltage and input current waveforms for the fixed duty cycle values used. Figure 5-19(a) shows the filtered three-phase 50 Hz output voltages and Figure 5-19(b) shows the input current and input voltage waveforms. The fixed duty cycle values used to generate the results shown in Figure 5-19 is given in Table 5-2. The reason for the unbalanced output

voltage waveforms in Figure 5-19(a) is due to the use of the unbalanced output filter inductance and the incorrect load current detection.



(a) (b)

Figure 5-19 Output Voltage waveform practical results of direct matrix converter a) Three-phase generated output voltage waveform, b) Input voltage and input current waveforms

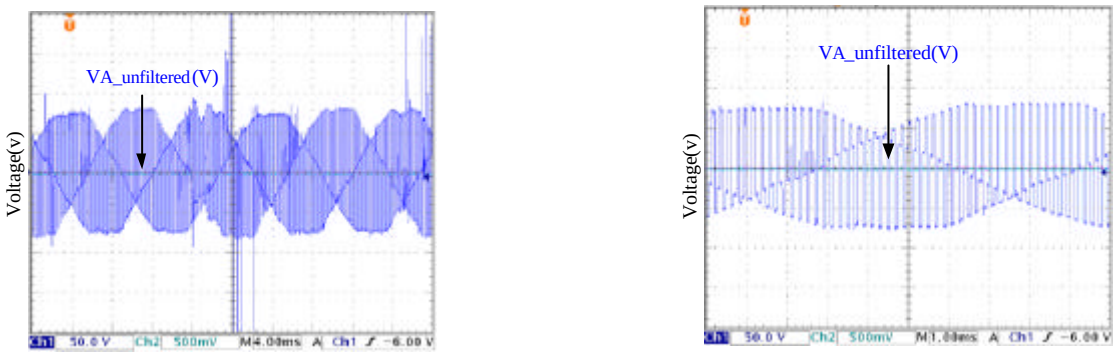
Table 5-2 Fixed duty cycle values for current commutation strategy

	d_{aA}	d_{bA}	d_{cA}	d_{aB}	d_{bB}	d_{cB}	d_{aC}	d_{bC}	d_{cC}
value	0.667	0.1667	0.1667	0.1667	0.667	0.1667	0.1667	0.1667	0.667

Figure 5-20 shows the unfiltered single-phase output voltage waveform. This waveform follows the envelope of the three-phase input voltages. The 9 duty cycle values used to generate the result shown in Figure 5-20(a) and (b) is given in Table 5-3.

Table 5-3 Fixed duty cycles with zero output voltage for current commutation method

	d_{aA}	d_{bA}	d_{cA}	d_{aB}	d_{bB}	d_{cB}	d_{aC}	d_{bC}	d_{cC}
value	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333



(a) (b)

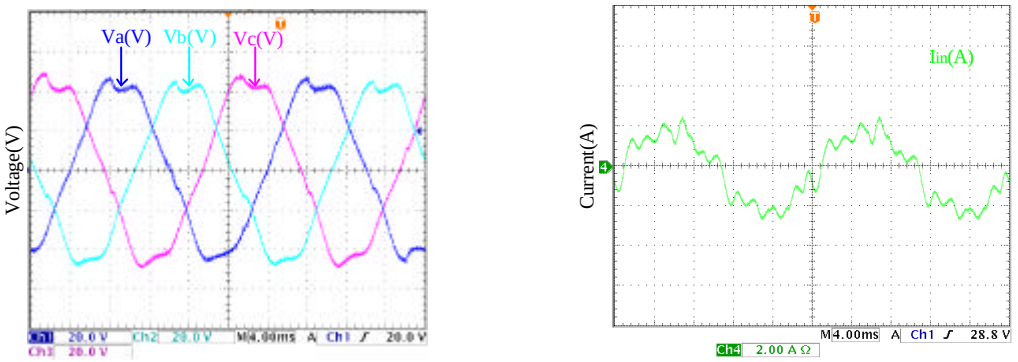
Figure 5-20 Output Voltage waveform test results based on current commutation (a) Three-phase generated unfiltered output voltage waveform; (b) Scaled three-phase generated unfiltered output voltage waveform.

5.4.2.2 Based on Voltage Commutation Strategy

Test results of the direct matrix converter based on the voltage commutation strategy proposed in [4] and [9] are presented below. Since fixed duty cycle values were applied to generate the PWM gating signal again, the result generated was only 50 Hz output frequency as the above-mentioned method. Figure 5-21 shows test results of the output voltage waveforms and the input current waveform for the fixed duty cycle values used. Figure 5-21(a) shows the filtered three-phase 50 Hz output voltage waveforms and Figure 5-21(b) shows the input current waveform. The fixed duty cycle values used to generate the result shown in Figure 5-21(a) are given in Table 5-4.

Table 5-4 Fixed duty cycle values for voltage commutation strategy

	d_{aA}	d_{bA}	d_{cA}	d_{aB}	d_{bB}	d_{cB}	d_{aC}	d_{bC}	d_{cC}
value	0.667	0.1667	0.1667	0.1667	0.667	0.1667	0.1667	0.1667	0.667

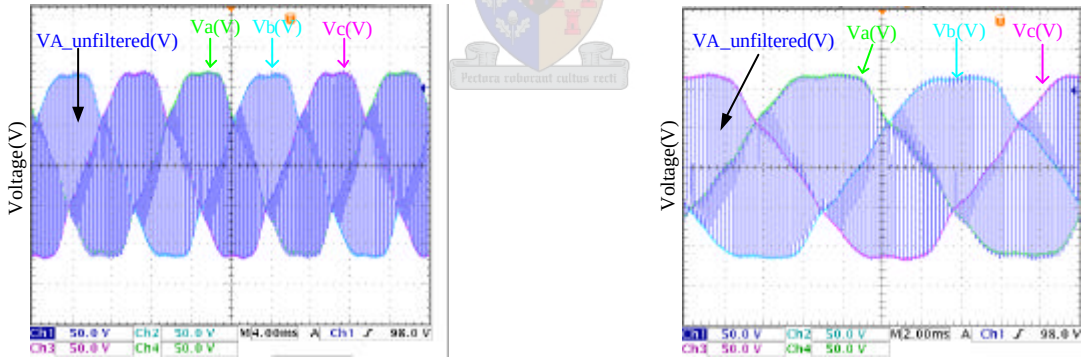


(a) (b)

Figure 5-21 Test result waveforms based on voltage commutation strategy a) Three-phase generated output voltage waveform; b) Input voltage and input current waveforms.

Figure 5-22 shows the unfiltered output voltage waveform obtained from the experimental tests. The control algorithm applied was based on the voltage commutation method. The waveform of the output voltage follows the envelope of the three-phase input voltages.

Table 5-5 shows the fixed duty cycle values used to generate the unfiltered output voltage.



(a) (b)

Figure 5-22 Output Voltage waveform test results of direct matrix converter based on voltage commutation method a) Three-phase generated unfiltered output voltage waveform; b) Scaled three-phase generated unfiltered output voltage waveform.

Table 5-5 Fixed duty cycles for voltage commutation strategy with zero output voltage

	d_{aA}	d_{bA}	d_{cA}	d_{aB}	d_{bB}	d_{cB}	d_{aC}	d_{bC}	d_{cC}

value	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333

5.4.3 Experimental Results for Variable Duty Cycle

5.4.3.1 Experimental Results Based on Current commutation

The test results obtained using the control algorithm based on the current commutation strategy for 50 Hz output frequency is all that is presented below. The current direction detection circuit had problems with detecting the direction of load current correctly. This was due to the low reverse load current value of the converter circuit. The IGBT device requires a reverse current of I_{rr} = 8 A for a fast reverse recovery, but the load current through the IGBT was less than 4 A. The delayed response of the detection circuit is shown in Figure 5-23. Hence, a zero current commutation problem made the switching more complex, distorted output voltage waveforms was obtained. Moreover, EMI noise and the TLV1570 ADC device were affecting the calculated duty cycle to fall below zero and/or give incorrect values, which changed the pattern of the generated PWM gating signal.

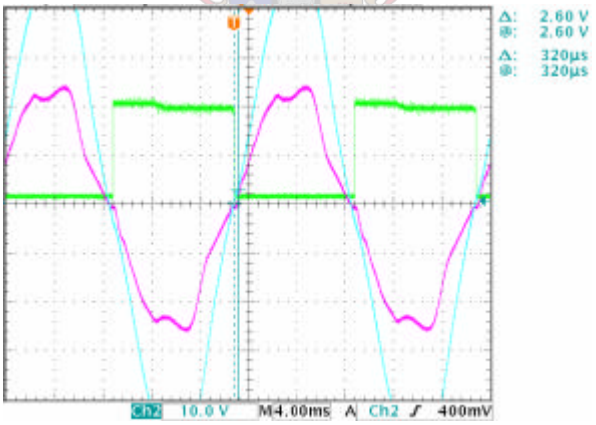
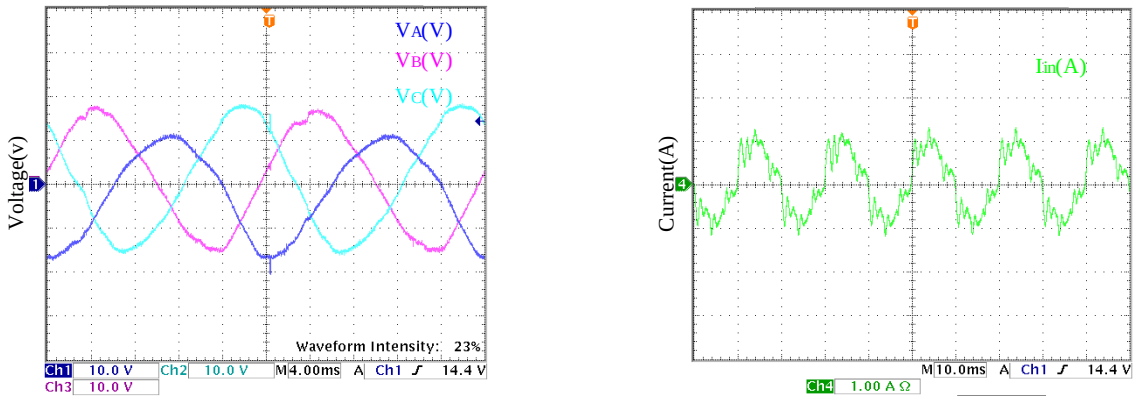


Figure 5-23 Output of the current direction detection circuit

Figure 5-24 shows the output voltage and input current waveforms obtained from the experimental test for modulation index of m=0.5 and 50 Hz output frequency. The three-phase output voltage waveforms are shown in Figure 5-24(a), and the input current waveform is shown in Figure 5-24(b).

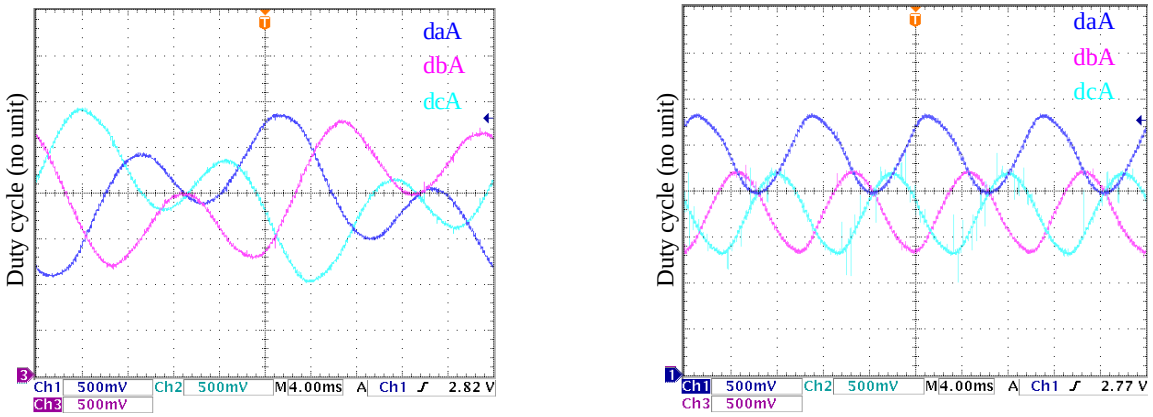


(a) (b)

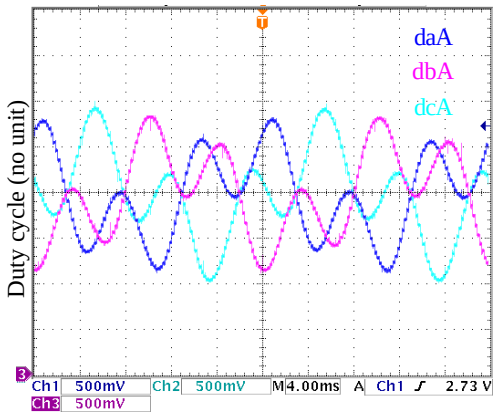
Figure 5-24 Test result waveforms, (a) output voltage of 50 Hz output frequency, (b) Input current of 50 Hz output frequency

5.4.3.2 Experimental Results Based on Voltage Commutation Strategy

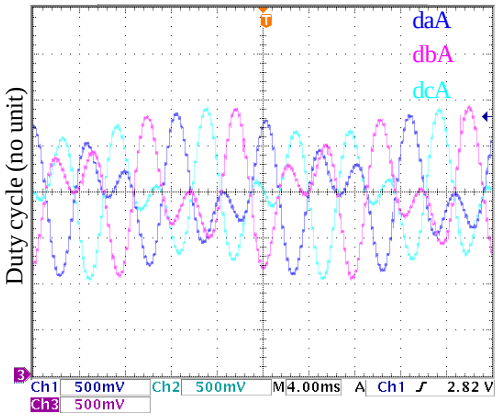
The Various duty cycle waveforms calculated using the given Equation 5-7 are presented in this subsection. Figure 5-25(a), (b), (c) and (d) show the duty cycle waveforms applied to generate the PWM gating signal of the direct matrix converter topology of Figure 5-18 for a 25 Hz, a 50 HZ, a 100 HZ, and a 200 Hz output frequencies respectively.



(a) (b)



(c)

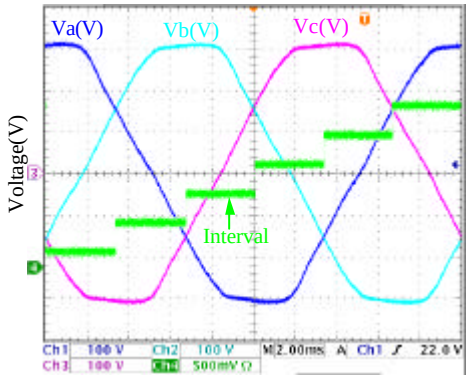


(d)

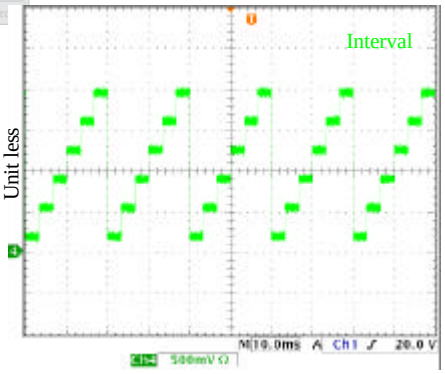
Figure 5-25 Duty cycle waveforms (a) for 25 Hz output frequency, (b) for 25 Hz output frequency, (c) for 25 Hz output frequency, (d) for 25 Hz output frequency.

The results displayed in Figure 5-10 and Figure 5-25 confirm the similarity of the duty cycle waveforms obtained from the simulation and experimental test done.

Figure 5-26 shows the six identified intervals obtained directly from the DSP device. These intervals were used to determine the safe switching state for the converter circuit.



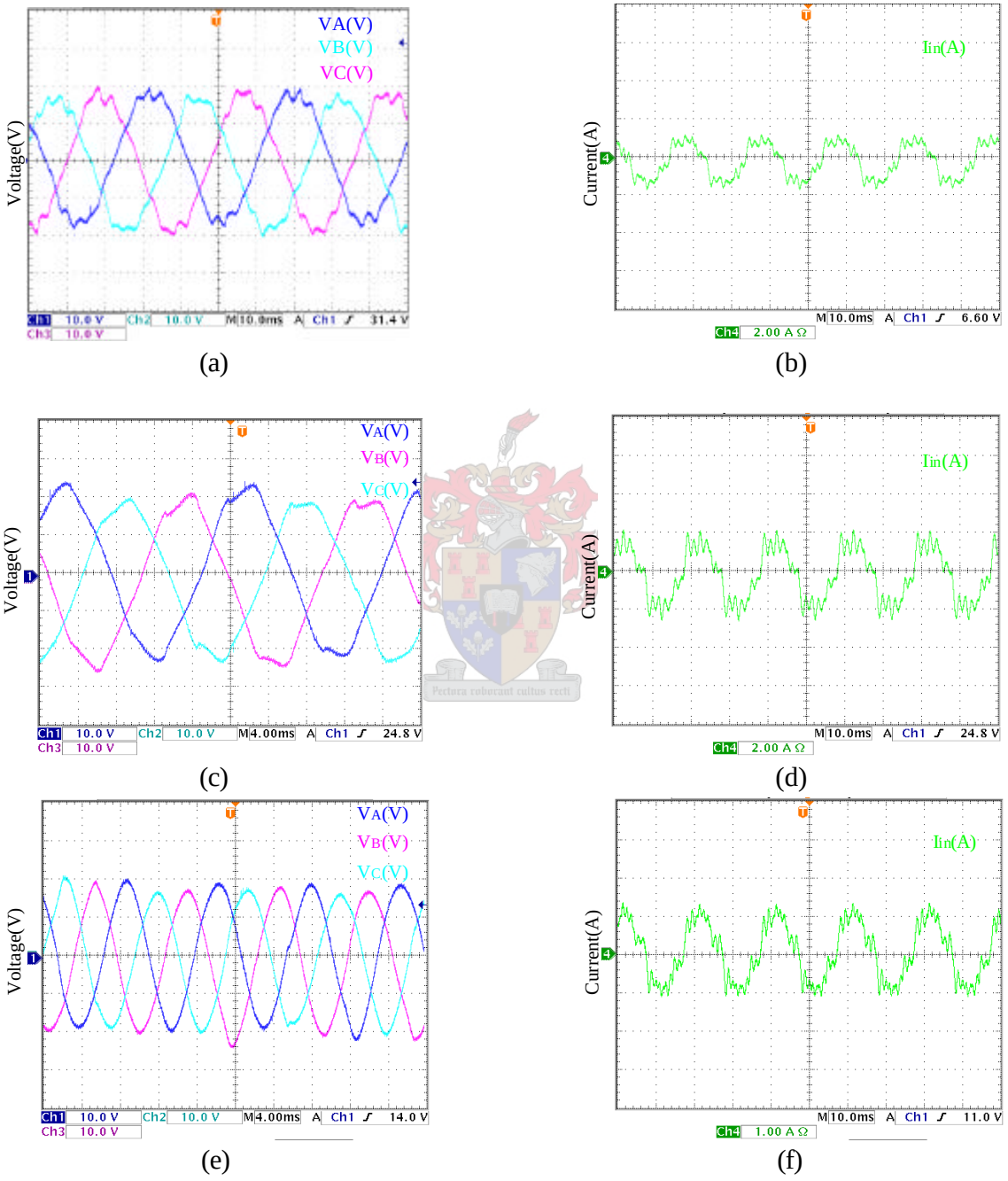
(a)

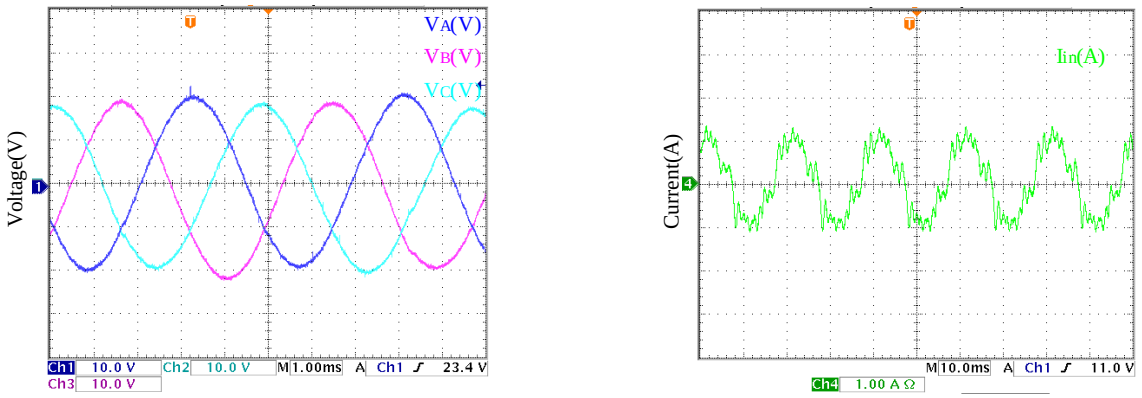


(b)

Figure 5-26 Output of DSP interval detection (a) Relationship between the three-phase input voltages and the six identified intervals; (b) The six detected intervals.

Experimental results of a 25 Hz, a 50 Hz, a 100 Hz, and a 200 Hz output frequency obtained using the above duty cycle waveforms of Figure 5-25 are shown Figure 5-27. Two-step safe voltage commutation strategy was used to control the converter circuit.



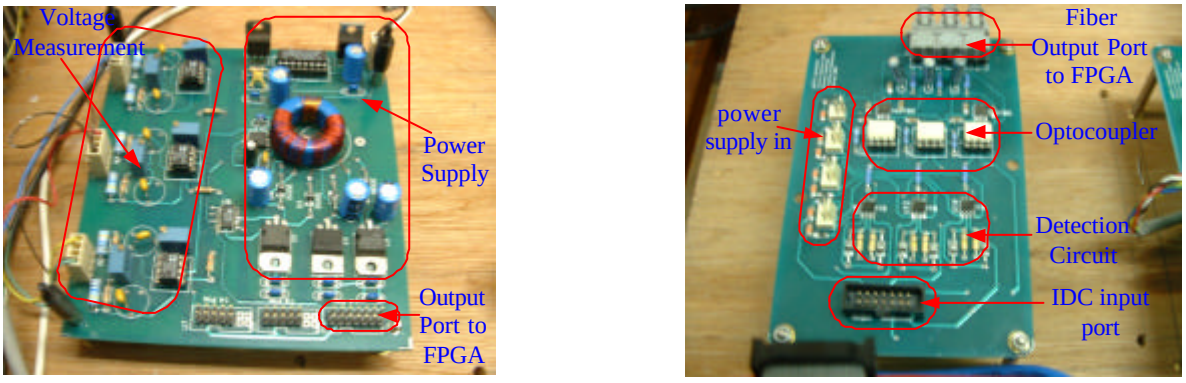


(g) (h)

Figure 5-27 Waveforms generated by the indirect matrix converter circuit (a) 3Æ output voltages of 25 Hz output frequency, (b) 3Æ input currents of 25 Hz output frequency; (c) 3Æ output voltages of 50 Hz output frequency, (d) 3Æ input currents of 50 Hz output frequency, (e) 3Æ output voltages of 100 Hz output frequency, (f) 3Æ input currents of 100 Hz output frequency, (g) 3Æ output voltages of 200 Hz output frequency, (h) 3Æ input currents of 200 Hz output frequency.

5.4.4 Results of Voltage Measurement & Current Detection Circuit

The voltage measurement was setup to measure and convert the three-phase input voltages into a lower level value. The current direction detection circuit was setup to measure the voltage drop across the bi-directional switch of the converter. Figure 5-28 shows the printed circuit board diagram of both the voltage measurement and current direction detection circuits respectively.



(a) (b)

Figure 5-28 Voltage measurement and current direction detection circuits

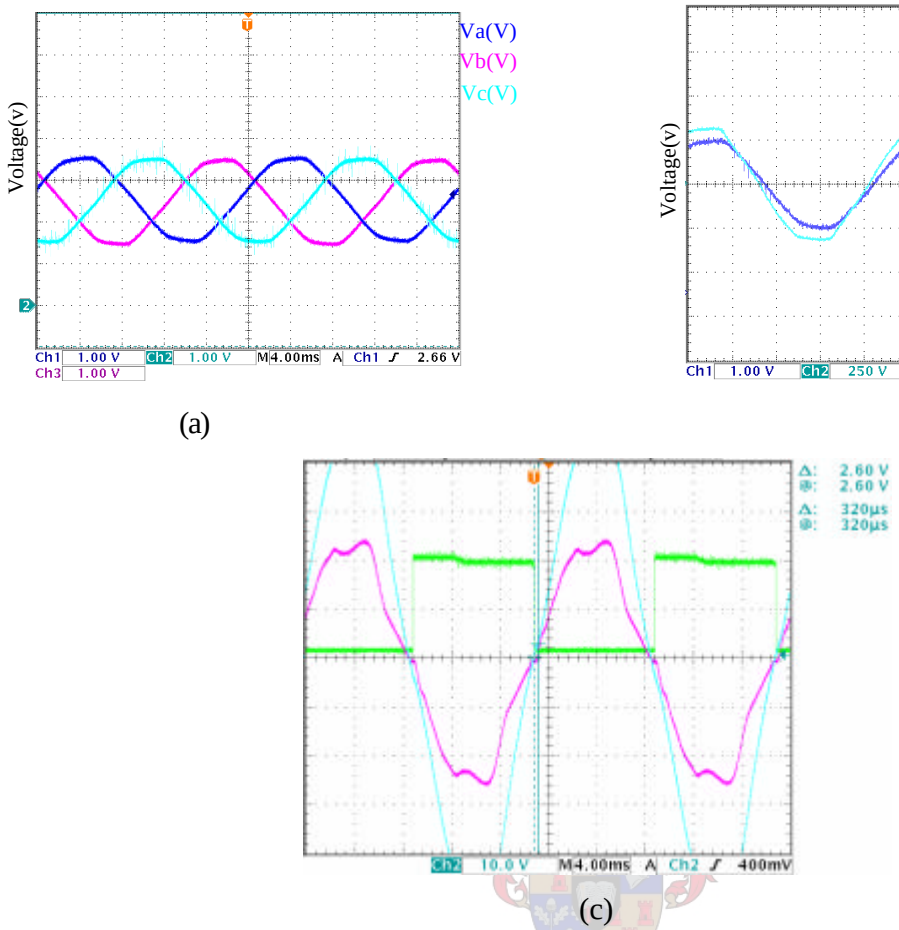


Figure 5-29 Output of voltage measurement and current direction detection circuits. (a) three-phase measured input voltages, (b) supply input voltage and measured input voltage, (c) Output of the current direction detection circuit.

Figure 5-29(a) shows the waveform of the three-phase input voltages measured using the voltage measurement shown in Figure 5-28(a). The phase shift shown in Figure 5-29(b) is due to the compensating capacitor in the measurement circuit. This compensating capacitor was used to eliminate or decrease the level of the noise and compensate for the inductance of the resistance of the measurement circuit. The result of the current direction detection circuit for only one bi-directional switch is shown in Figure 5-29(c).

5.5 Summary

The main part of the chapter presented the simulation and experimental results obtained from the test performed on the three-phase-to-three-phase direct matrix converter circuit of controlled output frequency and output voltage. In addition, simulation results of the indirect matrix converter topology are also discussed. The simulation results obtained from the direct matrix converter topology were for the various controlling methods proposed to modulate the bi-directional switches. The first three controlling methods produced 50% output-input transfer ratio. By adding the third harmonic components of the input and desired output voltages to the components of the desired output voltage, the output-input transfer ratio was improved to 86.6% and the results were shown. The simulation tests show that the controlling methods applied on the direct matrix converter circuit generate a controlled three-phase voltage of various frequencies. It can be concluded that the direct matrix converter is suitable for induction motor drive speed control.

The DC-voltage output of the rectifier as well as the sinusoidal output voltage of the inverter of the indirect matrix converter, were presented in this chapter. The SV PWM used for a standard DC/AC inverter was used on the inverter side of the converter. The distorted sinusoidal input current waveforms obtained from the simulation of the rectifier were presented. The relative input voltages method was used on the rectifier switches.

Experimental tests of the direct matrix converter topology based on the current and the voltage commutation strategy, for the fixed and the variable duty cycle, were presented. From the experimental results obtained, it is shown that if an accurate and reliable voltage measurement circuit is used, the two-step safe voltage commutation strategy provides a safe and better performance than the four-step current commutation strategy. The experimental tests conducted in the direct matrix converter topology were successful. The results obtained showed that the converter circuit achieved its basic goals. The measurement problem encountered due to the EMI was solved using first order RC digital IIR filter.

The overvoltage protection circuits was tested separately and showed the potential to protect the converter against overvoltage. The current direction detection circuit was tested and showed slit time delay, which is not good for the commutation process of the converter circuit. This time delay is due to the long recovery time of the IGBT device for low reverse current.

CHAPTER 6

CONCLUSION

6 CONCLUSION

6.1 Conclusion

This thesis demonstrated the analysis, design, and development of the PWM switching of the three-phase-to-three-phase direct AC-AC matrix converter. A background study of the various topologies of the matrix converter and the associated controlling algorithms was investigated. Theoretical background of the simulation results of both the direct and the indirect three-phase-to-three-phase AC-AC matrix converter topologies were presented. These results presented the output voltage and input current waveforms of a 0 Hz, a 25 Hz, a 50 Hz, a 100 Hz, and a 200 Hz output frequencies.

Furthermore, experimental tests were done on the three-phase-to-three-phase direct AC-AC matrix converter only. Test results of the output voltages and input current for a 25 Hz, a 50 Hz, a 100 Hz, and a 200 Hz output frequency were showed. The experimental tests were done using the direct control algorithm method based on the two commutation strategies, namely the four-step safe current commutation strategy and the two-step safe voltage commutation strategy. The results of the simulation and experimental test proved that the converter successfully complied with its specifications. The voltage measurement faced an EMI problem, which created a difficulty in identifying the correct current interval of the relative input voltages necessary for the voltage commutation method. Moreover, there was calculation error of the duty cycles. First-degree RC Digital IIR filter was used to eliminate the EMI successfully.

Based on the simulation and the experimental results shown, the three-phase direct matrix converter can be used as a replacement of the conventional rectifier-inverter based converter. The advantages of the matrix converter in short are:

- Inherent four-quadrant operation
- Absence of bulky DC-link electrolyte capacitors

- Pure input power characteristics with unity power factor
- Small sized all-silicon, and cost effective converter.

6.2 Future Work

Future work should focus on the improvement of the developed code to control the switching of the converter switches to obtain a better performance of the direct matrix converter. The design of accurate voltage measurement and high sensitivity current direction detection circuits is necessary. Since safe commutation depends solely on either reliable detection of the load current direction or accurate voltage measurement. Besides, improvement of both measurement circuits would solve the exiting problems. Furthermore, future work could also focus on the evaluation of the proposed SV PWM control strategy in the application of controlling the direct matrix converter topology. The use of the SV PWM controlling mechanism offers full control of the generated output voltage and input current waveforms.

An IGBT matrix module (FM35R12KE3) is available now in some areas for research purpose. It is advisable to use an IGBT module for constructing the direct matrix converter, because by doing so, the research will only emphasize on the development of the controlling algorithm.

References:

[1] Alesina, A. and M.G.B. Venturini, *Analysis and design of optimum-amplitude nine-switch direct AC-AC converters.* Power Electronics, IEEE Transactions on, 1989. 4(1): p. 101-112.

[2] Alesina, A. and M.G.B. Venturini. *Intrinsic amplitude limits and optimum design of 9-switches direct PWM AC-AC converters*. in *Power Electronics Specialists Conference, 1988. PESC '88 Record., 19th Annual IEEE*. 1988.

[3] Wheeler, P.W., et al. *Matrix Converters: The Technology and Potential for Exploitation*. in *The Drives and Controls Power Electronics Conference*. 2001. London.

[4] Marcus Ziegler and W. Hofmann, *New One-Step Commutation Strategies in Matrix Converters.* PEDS'01 Bali Proceedings, 2001: p. 560-564.

[5] Venturini, M.G.M., *"A New Sine Wave in, Sine Wave out Conversion Technique Eliminates Reactive Elements,".* in Proc. POWERCON 7,, 1980: p. E3-1-E3-15.

[6] Wei, L. and T.A. and Lipo, *A noval Matrix Converter Topology with Simple Commutation.* Record on the 26th IEEE Industrial Applications Society Annual Meetings, Chicago,, 2001. 3: p. 1749-1754.

[7] Kolar, J.W., et al., *Noval Three-Phase AC-DC-AC Sparse Matrix Converter.*

[8] Burany, N., *"Safe Control of Four Quadrant Switches".* IEEE Conference Record of the Industrial Application Society Annual Meeting,, 1989: p. 1190-1194.

[9] Marcus Ziegler and W. Hofmann, *Semi-natural Two steps commutation strategy for Matrix Converter.* PESC,Fukuoka/Japan, 1998: p. 727-731.

[10] Klumpner, C., F. Blaabjerg, and P. and Nielsen, *Speeding -up the Maturation Process of the Matrix Converter Technology.* Proceedings of the 32nd IEEE Power Electronics Specialist Conference, Vancounver, Canada, 2001. 2: p. 1083-1088.

[11] Youm, J. and B. Kwom, *"Switching Techniques for Current-Controlled AC-to-AC Converter,".* IEEE Trans. Industrial. Electronics, 1999. 46(2): p. 309-318.

[12] Babaei, E., Hosseini, *A New Circuit For Three-Phase to Three-Phase AC/AC Matrix Converters.*

[13]. Klumpner, C., et al. *A new matrix converter-motor (MCM) for industry applications*. in *Industry Applications Conference, 2000. Conference Record of the 2000 IEEE*. 2000.

[14]. D. Casadei, G.S., A. Tani, P. Nielsen, *"Theoretical and Experimental Analysis of SVM-controlled Matrix Converters Under Unbalanced Supply Conditions".* Trans. on ELECTOMOTION, 1997(no. 4): p. pp.28-37.

[15]. Christian Klumpner, P.N., Ion Boldea, and Frede Blaabjerg, *"New Solutions for a Low-Cost Power Electronic Building Block for Matrix Converters".* IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, APRIL 2002. VOL. 49, NO. 2: p. 336-344.

[16]. Wheeler, P.W., et al., *Gate drive level intelligence and current sensing for matrix converter current commutation.* Industrial Electronics, IEEE Transactions on, 2002. 49(2): p. 382-389.

[17]. Peter Nielsen, F.B., and John K. Pedersen, *"New Protection Issues of a Matrix Converter: Design Considerations for Adjustable-Speed Drives".* IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS,, SEPTEMBER/OCTOBER 1999. VOL. 35, NO. 5,: p. pp.1150-1161.

[18]. Bland, M., Wheeler, P., Clare, J., and Empringham, L., *"Comparison of Calculated and Measured Losses in Direct AC-AC Converters"*. in Power Electronics Specialists Conference, 2001. PESC 2001 IEEE 32nd Annual 2001.

Datasheets:

[19]. International IR rectifier, "IGBT IRG4PH40KD", *http://www.irf.com/product-info/datasheets/data/irg4ph40kd.pdf*

[20]. Agilent Technology, "IGBT Gate Drive Optocoupler", *http://cp.literature.agilent.com/litweb/pdf/5980-0462E.pdf*

[21]. Analog Device, "Instrumentation Amplifier AD620", *http://www.analog.com/UploadedFiles/Data_Sheets/37793330023930AD620_e.pdf*

[22]. Texas Instrumentation, "Isolation Amplifier ISO1220", *http://focus.ti.com/lit/ds/symlink/iso122.pdf*

[23]. Hewlett Packard, “Versatile Link /The Versatile Fiber Optic Connection”,

http:// www.ee.iisc.ernet.in/new/people/faculty/vr/DSMixed/HFBRX5X1.pbf

[24]. LEM Data Book 1999 CD, "Current Transducers for Industrial Applications", also *http://www.lem.com*

[25]. Sharp, "6N137 Optocoupler" *www.ee.iisc.ernet.in/new/people/faculty/vr/DSMixed/HFBRX5X1.pbf*

[26]. Texas Instrumentation, "TLV1570ADC Converter" *http://focus.ti.com/lit/ds/symlink/tlv1570.pdf*

[27]. Philips Semiconductors "BYV26 series Fast soft-recovery controlled avalanche rectifiers", http://www.semiconductors.philips.com/acrobat/datasheets/BYV26_2.pdf

Thesis:

[28]. Van Schalkwyk, C , H du T Mouton, H.J. Beukes "*Analysis & Design of a Voltage Regulator Based on AC to AC Converter".* Masters Thesis, University of Stellenboch, 2003

Appendix A

A.1 C-code for the current commutation strategy

```
/* Abraham Gebrehiwet
/* General purpose matrix converter C program based on Voltage commutation
strategy*/

#include <string.h>          /* String function library handler*/
#include <math.h>
#include <stdio.h>           /* Input/output function handler*/
#include <stdlib.h>
#include <float.h>
#include <limits.h>
#include "PEC33_Address.h"   /* header file address declarations*/
#include "sin_table1.h"      /* Sine function lookup table */



#define pi (double)3.14159265359      /* define pi           */
#define q (double)0.35                /* Modulation value    */
#define Vmax 272.5                    /* Amplitude of input Voltage*/
#define Amp 6000.0                    /* Maximum value of duty cycle*/

/* function prototypes */
void poll_IF4(void);
void reference_voltage(void); /* Function for reference voltage*/
interrupt void timer0(void);

interrupt void int0(void);    /* interrupt for duty cycle calculation */
interrupt void int1(void);
interrupt void int2(void);
interrupt void int3(void);

float     adc0_data_chan0;
float     adc1_data_chan0;
float     adc2_data_chan0;
float     adc3_data_chan0;

char lcd0[17],lcd1[17];
int check=0;

/* Reference and measured voltage variable declaration */
float Vref_A=0.0,Vref_B=0.0,Vref_C=0.0;/*Output reference*/
float Vin_a=0.0,Vin_b=0.0,Vin_c=0.0; /*Input voltage measured */

/* Duty cycle variable declaration */
float duty_aA=0.0,duty_aB=0.0,duty_aC=0.0,kA=0.0;
float duty_bA=0.0,duty_bB=0.0,duty_bC=0.0,kB=0.0;
float duty_cA=0.0,duty_cB=0.0,duty_cC=0.0,kC=0.0;

float Interval=0.0,x;
float Ubeta_a[16],Ualpha_a[16];
float Ubeta=0.0,Ualpha=0.0;
```

```
int j1, j2,j3,t;

main(void)
{
  int i,j;      /* integer declaration for iteration control*/

  asm(" ldi 0301h,IE");/* enable int2, van lcd, en timer 0 */
  asm(" OR 2000h,ST");/* status register glob int enable */


  i = 0;                            /* delay  */
    while (i<11250000) {
      i++;
    }


/* setup the DAC*/
   dac0[conf_addr] = 4;    /* DAC0 : bufferd, normal operation*/

  adc0[conf_adc]=samp_chan1;/*configure adc0 to sample chan 0*/
  adc1[conf_adc]=samp_chan1;/*configure adc1 to sample chan 0*/
  adc2[conf_adc]=samp_chan1;/*configure adc2 to sample chan 0*/
  adc3[conf_adc]=samp_chan1;/*configure adc3 to sample chan 0*/


/* main loop of DSP        */
    while ( 1 ) {                   /* inf loop       */

      *adc_samp_cmd = 1; /* command to sample analog to dig */

/* 2.56us dealay before data on adc is valid----79 */
      i= 0;
        while (i<79){ i++; }

/*adc is 10 bit, input 0 to 4.096V, "x & 0x3FF" zero all top 22 bits from
data bus of FPGA, " - 625.0" shifts the value to give signed representation
centered around 0, adc0_data_chan0 -> variable where data is stored in the
DSP,(float) -> converts the data to type float before storing
it,(adc0[adc_chan0_addr])-> address of the register in FPGA that is to be
read  */
/*============================================================*/
adc0_data_chan0=((float)((adc0[adc_chan1_addr])&0x3FF)-650.0);
adc1_data_chan0=((float)((adc1[adc_chan1_addr])&0x3FF)-650.0);
adc2_data_chan0=((float)((adc2[adc_chan1_addr])&0x3FF)-650.0);
adc3_data_chan0=((float)((adc3[adc_chan1_addr])&0x3FF)-650.0);

/*Calculation of measured Input Voltage per unity base */
/*Peak value of input voltage and its corresponding ADC
/*============================================================*/
Vin_a = 0.02469*(adc0_data_chan0/Vmax) + 0.9753*Vin_a;
Vin_b = 0.02469*(adc1_data_chan0/Vmax) + 0.9753*Vin_b;
Vin_c = 0.02469*(adc3_data_chan0/Vmax) + 0.9753*Vin_c;

Ualpha  =  0.66667*(Vin_a - 0.5*Vin_b - 0.5*Vin_c);
Ubeta =  0.57735*(Vin_b - Vin_c);
x = 2*pi*0.5/360;
```

```
Ualpha  =cos(x)*Ualpha - sin(x)*Ubeta;
Ubeta =sin(x)*Ualpha + cos(x)*Ubeta;
Vin_a = Ualpha;
Vin_b = -0.5*Ualpha + 0.8667*Ubeta;
Vin_c = -0.5*Ualpha - 0.8667*Ubeta; /**/

reference_voltage();
sector();

/* loads data in DAC buffers */
dac0[a_data_addr] = duty_aA;
dac0[b_data_addr] = 70*Interval;

dac0[load_dac]    = 1;       /* load dac          */
    }
}


void reference_voltage(void)
{

/* calculating of (wt)*/
/*=================================================================*/
  if  (t<=340) { t++;} else {t=0;  }/* t = 1350 -> 50hz output*/
    j1 =(int)(t/0.34);               /* t = 2700 -> 25hz output*/
                                     /* t = 675 -> 100hz output*/
/* calculating of (wt-2*pi/3)       /* t = 338 -> 100hz output*/
/*=================================================================*/
 if(j1 <= 333)  { j2=(j1 + (int)(2000/3));}
 else if ((j1 >= 333) && (j1 <= 1000)){j2=(j1 -(int)(1000/3));}

/* calculating of (wt-4*pi/3)*/
/*=================================================================*/
 if(j1 <= 666)  { j3=(j1 + (int)(1000/3));}
 else if((j1 >= 666) && (j1 <= 1000)){j3=(j1 - (int)(2000/3));}

/*Calulation of Reference Output Voltages per unity base */
/*=================================================================*/
  Vref_A = q*sin_table1[j1];
  Vref_B = q*sin_table1[j2];
  Vref_C = q*sin_table1[j3];

}
/*******************************************************/
 interrupt void timer0(void){ /* timer */
}
/*************************************************************/


interrupt void int0(void)
{
/* Duty cycle calculation and loading to the pwm_ctrl modula of the Analog
FPGA*/

/* Duty cycles for load phase A */
    duty_aA = (1 + (2*Vref_A*Vin_a))/3;
    duty_bA = (1 + (2*Vref_A*Vin_b))/3;
```

```
    duty_cA = (1 + (2*Vref_A*Vin_c))/3;
    if (duty_aA <= 0.1 )
      {   duty_aA = 0;
      }
    else if (duty_bA <= 0.1  )
      {   duty_bB = 0;
      }
    else if (duty_cA <= 0.1  )
      {   duty_cA = 0;
      }
    kA = duty_aA + duty_bA + duty_cA;
    duty_aA = Amp*(duty_aA/kA);
    duty_bA = Amp*(duty_bA/kA);
    duty_cA = Amp*(duty_cA/kA); /**/

  /* Duty cycles for load phase B*/
    duty_aB=(1 + (2*Vref_B*Vin_a))/3;
    duty_bB=(1 + (2*Vref_B*Vin_b))/3;
    duty_cB=(1 + (2*Vref_B*Vin_c))/3;
    if (duty_aB <= 0.1  )
      {   duty_aB = 0;
      }
    else if (duty_bB <= 0.1  )
      {   duty_bB = 0;
      }
    else if (duty_cB <= 0.1  )
      {   duty_cB = 0;
      }
    kB = duty_aB + duty_bB + duty_cB;
    duty_aB = Amp*(duty_aB/kB);
    duty_bB = Amp*(duty_bB/kB);
    duty_cB = Amp*(duty_cB/kB);

  /*Duty cycles for load phase C*/
    duty_aC=(1 + (2*Vref_C*Vin_a))/3;
    duty_bC=(1 + (2*Vref_C*Vin_b))/3;
    duty_cC=(1 + (2*Vref_C*Vin_c))/3;
    if (duty_aC <= 0.1  )
      {   duty_aC = 0;
      }
    else if (duty_bC <= 0.1  )
      {   duty_bC = 0;
      }
    else if (duty_cC <= 0.1  )
      {   duty_cC = 0;
      }
    kC = duty_aC + duty_bC + duty_cC;
    duty_aC = Amp*(duty_aC/kC);
    duty_bC = Amp*(duty_bC/kC);
    duty_cC = Amp*(duty_cC/kC);  /**/


    *DutyCycle_aA_cmd = duty_aA;
    *DutyCycle_bA_cmd = duty_bA;
    *DutyCycle_cA_cmd = duty_cA;
    *DutyCycle_aB_cmd = duty_aB;
    *DutyCycle_bB_cmd = duty_bB;
```

```
    *DutyCycle_cB_cmd = duty_cB;
    *DutyCycle_aC_cmd = duty_aC;
    *DutyCycle_bC_cmd = duty_bC;
    *DutyCycle_cC_cmd = duty_cC;


}

/*************************************************************/

interrupt void int1(void)
{

}

/*************************************************************/
interrupt void int2(void)
{

}

/*************************************************************/

interrupt void int3(void)
{

}
```

A.2 VHDL code for the Four-Step Current Commutation Strategy

```
--Abraham Gebrehiwet                    26/08/2004
--VHDL program written to control the level of Output voltage
--for a Three-phase to single phase output matrix converter

library ieee;                           -- Library declaration
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;


entity pwm_out_A is port
        (                                       --Port declaration
clk,nreset          : in std_logic;
--Current direction information declaration
       curr_dir_aA,curr_dir_bA,curr_dir_cA : in std_logic;
       amp : in std_logic_vector(12 downto 0);

       Duty_cycle_phase_aA : in std_logic_vector(11 downto 0);
       Duty_cycle_phase_bA : in std_logic_vector(11 downto 0);
       Duty_cycle_phase_cA : in std_logic_vector(11 downto 0);

--Overcurrent measurement
       Ov_current  : in std_logic;

--Output to Gate driveS
       pwm_out_phase_A      : out std_logic_vector(5 downto 0) );

end pwm_out_A;

architecture RTL of pwm_out_A is
-- Signal declaration
-- Machine State for safe switching sequence
    type state_type is (idle,first,second,third,fourth);
    signal current_state,next_state : state_type;

-- Output signal variable
    signal state : std_logic_vector(5 downto 0);
    signal counta : integer range 0 to 255;

-- variable of the reference triangular wave
    signal amp_int: integer range 0 to 8191;
    signal count_delay : integer range 0 to 460;
    signal commA_a_b_c,commA_b_c_a, commA_c_a_b : std_logic;
    signal start_switching : std_logic;

--Dutycycle is defined as := DutyCycle/MAX
    signal DutyCycle_aA : integer range 0 to 4000;
    signal DutyCycle_bA : integer range 0 to 4000;
    signal DutyCycle_cA : integer range 0 to 4000;
    signal MAX     : integer range 0 to 6000;
-- Delay time between switching-on and off(5us maximum)
    constant Delay   : integer:= 150;

--A process that generate the PWM commutating signals to the switches
```

```
--************************************
begin
p0: process(current_state,nreset,clk)
begin

     if (nreset = '0') then
              state <="000000";  --Reset all Switches

      elsif (clk'event and clk= '1') then
--            ********************************
--            **Phase A to Phase B or Phase C*
--            ********************************
          if (Ov_current = '1')  then
           if (commA_a_b_c = '1' and start_switching = '1') then
               case current_state is
                  when idle =>
                        next_state <= first;
                  when first =>
                        if curr_dir_aA = '0'  then
                              state(1) <= '0';
                        elsif curr_dir_aA = '1'  then
                              state(0) <= '0';
                        end if;
                        next_state <= second;
                  when second =>
                        if (DutyCycle_bA >= 500) then
                              if curr_dir_aA = '0' then
                                    state(2) <= '1';
                              elsif curr_dir_aA = '1' then
                                    state(3) <= '1';
                              end if;
                        else
                              if curr_dir_aA = '0' then
                                    state(4) <= '1';
                              elsif curr_dir_aA = '1' then
                                    state(5) <= '1';
                              end if;
                        end if;
                        next_state <= third;
                  when third =>
                        if curr_dir_aA ='0' then
                              state(0) <= '0';
                        elsif curr_dir_aA ='0' then
                              state(1) <= '0';
                        end if;
                        next_state <= fourth;
                  when fourth =>
                        if (DutyCycle_bA >= 500) then
                              state <= "001100";
                        else
                              state <= "110000";
                        end if;
                        next_state <=idle;
             end case;
```

```
--            *************************
--            **Phase B to Phase C or A*
--            *************************
    elsif ((commA_b_c_a = '1')and (start_switching = '1'))then
            case current_state is
              when idle =>
                    next_state <= first;
              when first =>
                      if curr_dir_bA = '0'  then
                              state(3) <= '0';
                      elsif curr_dir_bA = '1'
                              state(2) <= '0';
                      end if;
                      next_state <= second;
              when second =>
                      if (DutyCycle_cA >= 500) then
                              if curr_dir_bA = '0' then
                                      state(4) <= '1';
                              elsif curr_dir_bA = '1'
                                      state(5) <= '1';
                              end if;
                      else
                              if curr_dir_bA = '0' then
                                      state(0) <= '1';
                              elsif curr_dir_bA = '1' then
                                      state(1) <= '1';
                              end if;
                      end if;
                      next_state <= third;
              when third =>
                      if curr_dir_bA ='0' then
                              state(2) <= '0';
                      elsif curr_dir_bA ='1' then
                              state(3) <= '0';
                      end if;
                      next_state <= fourth;
              when fourth =>
                      if (DutyCycle_cA >= 500) then
                              state <= "110000";
                      else
                              state <= "000011";
                      end if;
                      next_state <=idle;
          end case;
--            *************************
--            **Phase C to Phase A or B*
--            *************************
    elsif ((commA_c_a_b='1')and (start_switching = '1')) then
          case current_state is
              when idle =>
                      next_state <= first;
              when first
                      if curr_dir_cA = '0'  then
                              state(5) <= '0';
                      elsif curr_dir_cA = '1'  then
                              state(4) <= '0';
```

```
                    end if;
                    next_state <= second;
              when second =>
                    if (DutyCycle_aA >= 500) then
                        if curr_dir_cA = '0' then
                            state(0) <= '1';
                        elsif curr_dir_cA = '1' then
                            state(1) <= '1';
                        end if;
                    else
                        if curr_dir_cA = '0' then
                            state(2) <= '1';
                        elsif curr_dir_cA = '1' then
                            state(3) <= '1';
                        end if;
                    end if;
                    next_state <= third;
            when third =>
                    if curr_dir_cA ='0' then
                        state(4) <= '0';
                    elsif curr_dir_cA ='1' then
                        state(5) <= '0';
                    end if;
                    next_state <= fourth;
            when fourth =>
                    if (DutyCycle_aA >= 500) then
                        state <= "000011";
                    else
                        state <= "001100";
                    end if;
                    next_state <=idle;
         end case;
        end if;
      else
        state <= "000000";
      end if;
   end if;
end process;
--A process that generate the 5µsec delay time
--****************************************
p1:process(clk,nreset)
begin
 if (nreset='0') then
       current_state <= idle;
       counta <= 0 ;

 elsif (clk'event and clk='1') then
      if (start_switching = '1')and(current_state = idle ) then
           current_state <= first;
      elsif (counta >= Delay )   then
           current_state <= next_state;
      end if;

      if (counta >= Delay) or (current_state = idle) then
           counta <=0;
      else
           counta <= counta + 1;
```

```
      end if;
 end if;
end process;
--A process that generate commutation-initiating signals
--*****************************************************
p2:process(clk,nreset)
 begin
    if nreset='0' then
        amp_int <= 0;
        count_delay  <= 0;
        commA_a_b_c <= '0';
        commA_b_c_a <= '0';
        commA_c_a_b <= '0';
    elsif (clk'event and clk='1') then
        amp_int <= conv_integer(unsigned(amp));
        DutyCycle_aA <= conv_integer(unsigned(Duty_cycle_phase_aA));
        DutyCycle_bA <= conv_integer(unsigned(Duty_cycle_phase_bA));
        DutyCycle_cA <= conv_integer(unsigned(Duty_cycle_phase_cA));
        MAX <= DutyCycle_aA + DutyCycle_bA + DutyCycle_cA;

        if count_delay <= 455 then
                count_delay <= count_delay + 1;
        end if;

        if (amp_int = DutyCycle_aA) and( DutyCycle_aA >= 500)  then
                if ( DutyCycle_bA >= 500) or (DutyCycle_cA >= 500)then
                        start_switching <= '1' ;
                end if;
                commA_a_b_c <='1'; --Start commutating from phase a-to-b
                count_delay <= 0;
        elsif amp_int=(DutyCycle_aA+DutyCycle_bA)and(DutyCycle_bA>=500) then
                if ( DutyCycle_cA >= 500) or ( DutyCycle_aA >= 500)then
                        start_switching <= '1';
                end if;
                commA_b_c_a <='1';--Start commutating from phase a-to-c
                count_delay <= 0;
        elsif amp_int = (MAX-1) and ( DutyCycle_cA >= 500) then
                if( DutyCycle_aA >= 500) or ( DutyCycle_bA >= 500) then
                        start_switching <= '1';
                end if;
                commA_c_a_b <='1';--Start commutating from phase a-to-c
                count_delay <= 0;
        elsif count_delay >= 455 then
                commA_a_b_c<='0';
                commA_b_c_a<='0';
                commA_c_a_b<='0';
                start_switching <= '0';
        end if;
     end if;
  end if;
end process;

pwm_out_phase_A  <= state; --PWM control signal to the gate-drive

end RTL;
```

A.3 C code for the Two-Step Voltage Commutation Strategy

```
/* Abraham Gebrehiwet
/* General purpose matrix converter C program based on Voltage commutation
strategy*/

#include <string.h>          /* String function library handler*/
#include <math.h>
#include <stdio.h>           /* Input/output function handler*/
#include <stdlib.h>
#include <float.h>
#include <limits.h>
#include "PEC33_Address.h"   * header file address declarations*/
#include "sin_table1.h"      /* Sine function lookup table */

#define pi (double)3.14159265359      /* define pi           */
#define q (double)0.5                 /* Modulation value    */
#define Vmax 272.5             /* Amplitude of input Voltage*/
#define Amp 6000.0             /* Maximum value of duty cycle*/

/* function prototypes */
void poll_IF4(void);
void sector(void);

interrupt void int0(void); /* interrupt for duty cycle calculation */
interrupt void int1(void);
interrupt void int2(void);
interrupt void int3(void);

void reference_voltage(void); /* Function for reference voltage*/
interrupt void timer0(void

float     adc0_data_chan0;
float     adc1_data_chan0;
float     adc2_data_chan0;
float     adc3_data_chan0;

char lcd0[17],lcd1[17];
int check=0;

/* Reference and measured voltage variable declaration */
float Vref_A=0.0,Vref_B=0.0,Vref_C=0.0;/*Output reference  */
float Vin_a=0.0,Vin_b=0.0,Vin_c=0.0; /*Input voltage measured*/

/* Duty cycle variable declaration */
float duty_aA=0.0,duty_aB=0.0,duty_aC=0.0,kA=0.0;
float duty_bA=0.0,duty_bB=0.0,duty_bC=0.0,kB=0.0;
float duty_cA=0.0,duty_cB=0.0,duty_cC=0.0,kC=0.0;

float Interval=0.0,x;
float Ubeta_a[16],Ualpha_a[16];
float Ubeta=0.0,Ualpha=0.0;

int j1, j2,j3,t;
```

```
main(void)
{
  int i,j;      /* integer declaration for iteration control*/

  asm(" ldi 0301h,IE");/* enable int2, van lcd, en timer 0 */
  asm(" OR 2000h,ST");/* status register glob int enable */


  i = 0;                              /* delay  */
    while (i<11250000) {
      i++;
    }


/* setup the DAC*/
   dac0[conf_addr] = 4;    /* DAC0 : bufferd, normal operation*/

  adc0[conf_adc]=samp_chan1;/*configure adc0 to sample chan 0*/
  adc1[conf_adc]=samp_chan1;/*configure adc1 to sample chan 0*/
  adc2[conf_adc]=samp_chan1;/*configure adc2 to sample chan 0*/
  adc3[conf_adc]=samp_chan1;/*configure adc3 to sample chan 0*/


/* main loop of DSP        */
    while ( 1 ) {                 /* inf loop      */

      *adc_samp_cmd = 1; /* command to sample analog to dig */

/* 2.56us dealay before data on adc is valid----79 */
      i= 0;
        while (i<79){ i++; }

/*adc is 10 bit, input 0 to 4.096V, "x & 0x3FF" zero all top 22 bits from
data bus of FPGA, " - 625.0" shifts the value to give signed representation
centered around 0, adc0_data_chan0 -> variable where data is stored in the
DSP,(float) -> converts the data to type float before storing
it,(adc0[adc_chan0_addr])-> address of the register in FPGA that is to be
read  */
/*============================================================*/
adc0_data_chan0=((float)((adc0[adc_chan1_addr])&0x3FF)-650.0);
adc1_data_chan0=((float)((adc1[adc_chan1_addr])&0x3FF)-650.0);
adc2_data_chan0=((float)((adc2[adc_chan1_addr])&0x3FF)-650.0);
adc3_data_chan0=((float)((adc3[adc_chan1_addr])&0x3FF)-650.0);

/*Calculation of measured Input Voltage per unity base */
/*Peak value of input voltage and its corresponding ADC
/*============================================================*/
Vin_a = 0.02469*(adc0_data_chan0/Vmax) + 0.9753*Vin_a;
Vin_b = 0.02469*(adc1_data_chan0/Vmax) + 0.9753*Vin_b;
Vin_c = 0.02469*(adc3_data_chan0/Vmax) + 0.9753*Vin_c;

Ualpha  =  0.66667*(Vin_a - 0.5*Vin_b - 0.5*Vin_c);
Ubeta =  0.57735*(Vin_b - Vin_c);
x = 2*pi*0.5/360;
Ualpha  =cos(x)*Ualpha - sin(x)*Ubeta;
Ubeta =sin(x)*Ualpha + cos(x)*Ubeta;
```

```
Vin_a = Ualpha;
Vin_b = -0.5*Ualpha + 0.8667*Ubeta;
Vin_c = -0.5*Ualpha - 0.8667*Ubeta; /**/

reference_voltage();
sector();

/* loads data in DAC buffers */
dac0[a_data_addr] = duty_aA;
dac0[b_data_addr] = 70*Interval;

dac0[load_dac]    = 1;       /* load dac          */
    }
}

void reference_voltage(void)
{

/* calculating of (wt)*/
/*=====================================================================*/
  if  (t<=340) { t++;} else {t=0;  }/* t = 1350 -> 50hz output*/
    j1 =(int)(t/0.34);               /* t = 2700 -> 25hz output*/
                                     /* t = 675 -> 100hz output*/
/* calculating of (wt-2*pi/3)       /* t = 338 -> 100hz output*/
/*=====================================================*/
 if(j1 <= 333)  { j2=(j1 + (int)(2000/3));}
 else if ((j1 >= 333) && (j1 <= 1000)){j2=(j1 -(int)(1000/3));}

/* calculating of (wt-4*pi/3)*/
/*=====================================================*/
 if(j1 <= 666)  { j3=(j1 + (int)(1000/3));}
 else if((j1 >= 666) && (j1 <= 1000)){j3=(j1 - (int)(2000/3));}

/*Calulation of Reference Output Voltages per unity base */
/*=====================================================*/
  Vref_A = q*sin_table1[j1];
  Vref_B = q*sin_table1[j2];
  Vref_C = q*sin_table1[j3];

}
/*******************************************************/
 interrupt void timer0(void){ /* timer */
}
/*******************************************************/


interrupt void int0(void)
{
/* Duty cycle calculation and loading to the pwm_ctrl modula of the Analog
FPGA*/

/* Duty cycles for load phase A */
    duty_aA = (1 + (2*Vref_A*Vin_a))/3;
    duty_bA = (1 + (2*Vref_A*Vin_b))/3;
    duty_cA = (1 + (2*Vref_A*Vin_c))/3;
    if (duty_aA <= 0.1 )
      {   duty_aA = 0;
```

```
    }
  else if (duty_bA <= 0.1  )
    {   duty_bB = 0;
    }
  else if (duty_cA <= 0.1  )
    {   duty_cA = 0;
    }
  kA = duty_aA + duty_bA + duty_cA;
  duty_aA = Amp*(duty_aA/kA);
  duty_bA = Amp*(duty_bA/kA);
  duty_cA = Amp*(duty_cA/kA); /**/

/* Duty cycles for load phase B*/
  duty_aB=(1 + (2*Vref_B*Vin_a))/3;
  duty_bB=(1 + (2*Vref_B*Vin_b))/3;
  duty_cB=(1 + (2*Vref_B*Vin_c))/3;
  if (duty_aB <= 0.1  )
    {   duty_aB = 0;
    }
  else if (duty_bB <= 0.1  )
    {   duty_bB = 0;
    }
  else if (duty_cB <= 0.1  )
    {   duty_cB = 0;
    }
  kB = duty_aB + duty_bB + duty_cB;
  duty_aB = Amp*(duty_aB/kB);
  duty_bB = Amp*(duty_bB/kB);
  duty_cB = Amp*(duty_cB/kB);

/*Duty cycles for load phase C*/
  duty_aC=(1 + (2*Vref_C*Vin_a))/3;
  duty_bC=(1 + (2*Vref_C*Vin_b))/3;
  duty_cC=(1 + (2*Vref_C*Vin_c))/3;
  if (duty_aC <= 0.1  )
    {   duty_aC = 0;
    }
  else if (duty_bC <= 0.1  )
    {   duty_bC = 0;
    }
  else if (duty_cC <= 0.1  )
    {   duty_cC = 0;
    }
  kC = duty_aC + duty_bC + duty_cC;
  duty_aC = Amp*(duty_aC/kC);
  duty_bC = Amp*(duty_bC/kC);
  duty_cC = Amp*(duty_cC/kC);  /**/


  *DutyCycle_aA_cmd = duty_aA;
  *DutyCycle_bA_cmd = duty_bA;
  *DutyCycle_cA_cmd = duty_cA;
  *DutyCycle_aB_cmd = duty_aB;
  *DutyCycle_bB_cmd = duty_bB;
  *DutyCycle_cB_cmd = duty_cB;
  *DutyCycle_aC_cmd = duty_aC;
  *DutyCycle_bC_cmd = duty_bC;
```

```
    *DutyCycle_cC_cmd = duty_cC;

}
/*************************************************************/
interrupt void int1(void)
{

}
/*************************************************************/
interrupt void int2(void)
{

}
/*************************************************************/
interrupt void int3(void)
{

}

void sector(void){
  float temp = 0.0,x;
  int i = 0,j=0;

  temp = 1.73205*fabs(Ualpha);

    if (Ubeta >= 0)
    {
      if (Ubeta > temp){
      Interval = 15.0;
      }
      else{
        if(Ualpha > 0){
        Interval = 5.0;
        }
        else{
        Interval = 25.0;
        }
      }
     }
     else
    {
      if (Ubeta < -temp){
      Interval = 45.0;
      }
      else{
        if(Ualpha > 0){
          Interval = 55.0;
        }
          else{
        Interval = 35.0;
        }
      }
    }
     *Interval_cmd = Interval;
}
```

A.4 VHDL code for the Two-Step Voltage Commutation Strategy

```
--Abraham Gebrehiwet
--VHDL program written to control the level of Output voltage
--for a Three-phase to single phase output matrix converter

library ieee;                  -- Library declaration
USE ieee.std_logic_1164.all; -- Declaring the library for std_logic USE
ieee.std_logic_arith.all;    -- and std_logic_vectors

entity vcommutation_a is port
         (             --Port declaration
--Clock and Reset in put port declaration
        clk       : in std_logic;
        nreset      : in std_logic;
        nINT_0      : out std_logic; -- Interrupt request for new duty cycle
        section_value : in std_logic_vector(5 downto 0);
        amp : in std_logic_vector(12 downto 0);

        Duty_cycle_phase_Aa : in std_logic_vector(12 downto 0);
        Duty_cycle_phase_Ba : in std_logic_vector(12 downto 0);
        Duty_cycle_phase_Ca : in std_logic_vector(12 downto 0);

        pwmoutA       : out std_logic_vector(5 downto 0));

end vcommutation_a;

architecture rtl of vcommutation_a is
  -- Machine State for safe switching sequence
  type section_type is (One,Two,Three,Four,Five,Six);
  signal section : section_type;

  type system_type is (start,delay1,commutate,delay2);
  signal conv_oper     : system_type;

  type state_type is (idle,first,second);
  signal current_state,next_state : state_type;

  type data_rx_conv is (idle,hold);
  signal section_conv  : data_rx_conv;
  signal duty_Aa,duty_Ba,duty_Ca : data_rx_conv;
Amplitude of the triangular wave form (6000)-- 200us*30Mhz
  signal ampA: integer range 0 to 8192;

--Dutycycle is defined as := DutyCycle/MAX
  signal DutyCycle_Aa : integer range 0 to 6000;
  signal DutyCycle_Ba : integer range 0 to 6000;
  signal DutyCycle_Ca : integer range 0 to 6000;

  signal switchA_to_b  :  std_logic;
  signal switchA_to_c  :  std_logic;
  signal switchA_to_a  :  std_logic;
  signal start_in  :  std_logic;
```

```
  signal nINT_0_buf: std_logic;

  signal pwmout_buf    : std_logic_vector(5 downto 0);
  signal count             : integer range 0 to 150;
  signal section_int   : integer range 0 to 60;

  constant Deadtime  : integer:= 150;
  constant Min_val   : integer:= 305;
  constant MAX       : integer:= 6000;                  --
begin
  p0:process(clk,nreset)
  begin
    if nreset = '0' then
      pwmout_buf <= "000000";
      count <=0;
      conv_oper <= start;
      current_state <= idle;
    elsif (clk'event and clk = '1')  then
      case section is
        when Six =>
          case conv_oper is
            when start =>
              pwmout_buf <= "100110";
              conv_oper <= delay1;
              current_state <= idle;
            when delay1 =>
              if count = Deadtime then
                count <= 0;
                pwmout_buf <= "000110";
                conv_oper <= commutate;
              else
                count <= count + 1;
              end if;
            when commutate =>
              if switchA_to_b = '1' then
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "000110";
                    --pwmout_buf(0) <= '0';
                    next_state <= second;
                    conv_oper <= delay2;
                  when second =>
                    if DutyCycle_Ba >= Min_val then
                      pwmout_buf <= "001110";--switch to phase b
                      --pwmout_buf(3) <= '1';
                    elsif DutyCycle_Ca >= Min_val then
                      pwmout_buf <= "110110";--switch to phase c
                    end if;
                    next_state <= idle;
                    conv_oper <= delay2;
                end case;
              elsif switchA_to_c = '1' then
```

```
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "000110";
--                   pwmout_buf(3) <= '0';
                    next_state <= second;
                    conv_oper <= delay2;
                  when second =>
                    if DutyCycle_Ca >= Min_val then
                      pwmout_buf <= "110110";--switch to phase b
                      --pwmout_buf(3) <= '1';
                    elsif DutyCycle_Aa >= Min_val then
                      pwmout_buf <= "000111";--switch to phase c
                    end if;
--                   pwmout_buf(5) <= '1';
  --                 pwmout_buf(4) <= '1';
                    next_state <= idle;
                    conv_oper <= delay2;
                end case;
              elsif switchA_to_a = '1' then
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "000110";
--                   pwmout_buf(5) <= '0';
--                   pwmout_buf(4) <= '0';
                    next_state <= second;
                    conv_oper <= delay2;
                  when second =>
                    if DutyCycle_Aa >= Min_val then
                      pwmout_buf <= "000111";--switch to phase b
                      --pwmout_buf(3) <= '1';
                    elsif DutyCycle_Ba >= Min_val then
                      pwmout_buf <= "001110";--switch to phase c
                    end if;
--                   pwmout_buf(0) <= '1';
                    next_state <= idle;
                    conv_oper <= delay2;
                end case;
              end if;
            when delay2 =>
              if count = Deadtime then
                count <= 0;
                conv_oper <= commutate;
                current_state <= next_state;
              else
                count <= count + 1;
              end if;
          end case;
```

```
          if (section_int < 10)  then
            conv_oper <= start;
          end if;

        when One =>

          case conv_oper is
            when start =>
              pwmout_buf <= "010110";
              conv_oper <= delay1;
              current_state <= idle;
            when delay1 =>
              if count = Deadtime then
                count <= 0;
                pwmout_buf <= "010010";
--                pwmout_buf(0) <= '1';
                conv_oper <= commutate;
              else
                count <= count + 1;
              end if;
            when commutate =>
              if switchA_to_b = '1' then
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "010010";
                    next_state <= second;
                    conv_oper <= delay2;
                  when second =>
                    if DutyCycle_Ba >= Min_val then
                      pwmout_buf <= "011110";--switch to phase b
                      --pwmout_buf(3) <= '1';
                    elsif DutyCycle_Ca >= Min_val then
                      pwmout_buf <= "110010";--switch to phase c
                    end if;
                    --pwmout_buf(3) <= '1';
                    --pwmout_buf(2) <= '1';
                    next_state <= idle;
                    conv_oper <= delay2;
                end case;
              elsif switchA_to_c = '1' then
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "010010";
                    --pwmout_buf(3) <= '0';
                    --pwmout_buf(2) <= '0';
                    next_state <= second;
                    conv_oper <= delay2;
```

```
            when second =>
              if DutyCycle_Ca >= Min_val then
                pwmout_buf <= "110010";--switch to phase b
                --pwmout_buf(3) <= '1';
              elsif DutyCycle_Aa >= Min_val then
                pwmout_buf <= "010011";--switch to phase c
              end if;
              --pwmout_buf(5) <= '1';
              next_state <= idle;
              conv_oper <= delay2;
          end case;
        elsif switchA_to_a = '1' then
          case current_state is
            when idle =>
              if start_in = '1' then
                next_state <= first;
                conv_oper <= delay2;
              end if;
            when first =>
              pwmout_buf <= "010010";
              --pwmout_buf(5) <= '0';
              next_state <= second;
              conv_oper <= delay2;
            when second =>
              if DutyCycle_Aa >= Min_val then
                pwmout_buf <= "010011";--switch to phase b
                --pwmout_buf(3) <= '1';
              elsif DutyCycle_Ba >= Min_val then
                pwmout_buf <= "011110";--switch to phase c
              end if;
              --pwmout_buf(0) <= '1';
              next_state <= idle;
              conv_oper <= delay2;
          end case;
        end if;
      when delay2 =>
        if count = Deadtime then
          count <= 0;
          conv_oper <= commutate;
          current_state <= next_state;
        else
          count <= count + 1;
        end if;
    end case;
    if ((section_int > 10) and (section_int < 20)) then
      conv_oper <= start;
    end if;

  when Two =>
    case conv_oper is
      when start =>
        pwmout_buf <= "011010";
        conv_oper <= delay1;
        current_state <= idle;
      when delay1 =>
        if count = Deadtime then
          count <= 0;
```

```
                pwmout_buf <= "011000";
--               pwmout_buf(1) <= '1';
--               pwmout_buf(0) <= '1';
                conv_oper <= commutate;
              else
                count <= count + 1;
              end if;
            when commutate =>
              if switchA_to_b = '1' then
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "011000";
                    --pwmout_buf(1) <= '0';
                    --pwmout_buf(0) <= '0';
                    next_state <= second;
                    conv_oper <= delay2;
                  when second =>
                    if DutyCycle_Ba >= Min_val then
                      pwmout_buf <= "011100";--switch to phase b
                      --pwmout_buf(3) <= '1';
                    elsif DutyCycle_Ca >= Min_val then
                      pwmout_buf <= "111000";--switch to phase c
                    end if;
                    --pwmout_buf(2) <= '1';
                    next_state <= idle;
                    conv_oper <= delay2;
                end case;
              elsif switchA_to_c = '1' then
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "011000";
                    --pwmout_buf(2) <= '0';
                    next_state <= second;
                    conv_oper <= delay2;
                  when second =>
                    if DutyCycle_Ca >= Min_val then
                      pwmout_buf <= "111000";--switch to phase b
                      --pwmout_buf(3) <= '1';
                    elsif DutyCycle_Aa >= Min_val then
                      pwmout_buf <= "011011";--switch to phase c
                    end if;
                    --pwmout_buf(5) <= '1';
                    next_state <= idle;
                    conv_oper <= delay2;
                end case;
              elsif switchA_to_a = '1' then
                case current_state is
```

```
                when idle =>
                  if start_in = '1' then
                    next_state <= first;
                    conv_oper <= delay2;
                  end if;
                when first =>
                  pwmout_buf <= "011000";
                  --pwmout_buf(5) <= '0';
                  next_state <= second;
                  conv_oper <= delay2;
                when second =>
                  if DutyCycle_Aa >= Min_val then
                    pwmout_buf <= "011011";--switch to phase b
                    --pwmout_buf(3) <= '1';
                  elsif DutyCycle_Ba >= Min_val then
                    pwmout_buf <= "011100";--switch to phase c
                  end if;
                  --pwmout_buf(1) <= '1';
                  --pwmout_buf(0) <= '1';
                  next_state <= idle;
                  conv_oper <= delay2;
              end case;
            end if;
          when delay2 =>
            if count = Deadtime then
              count <= 0;
              conv_oper <= commutate;
              current_state <= next_state;
            else
              count <= count + 1;
            end if;
        end case;
        if ((section_int > 20) and (section_int < 30)) then
          conv_oper <= start;
        end if;

      when Three =>
        case conv_oper is
          when start =>
            pwmout_buf <= "011001";
            conv_oper <= delay1;
            current_state <= idle;
          when delay1 =>
            if count = Deadtime then
              count <= 0;
              pwmout_buf <= "001001";
--              pwmout_buf(1) <= '1';
              conv_oper <= commutate;
            else
              count <= count + 1;
            end if;
          when delay2 =>
            if count = Deadtime then
              count <= 0;
              conv_oper <= commutate;
              current_state <= next_state;
            else
```

```
        count <= count + 1;
      end if;
    when commutate =>
      if switchA_to_b = '1' then
        case current_state is
          when idle =>
            if start_in = '1' then
              next_state <= first;
              conv_oper <= delay2;
            end if;
          when first =>
            pwmout_buf <= "001001";
            --pwmout_buf(1) <= '0';
            next_state <= second;
            conv_oper <= delay2;
          when second =>
            if DutyCycle_Ba >= Min_val then
              pwmout_buf <= "001101";--switch to phase b
              --pwmout_buf(3) <= '1';
            elsif DutyCycle_Ca >= Min_val then
              pwmout_buf <= "111001";--switch to phase c
            end if;
            --pwmout_buf(2) <= '1';
            next_state <= idle;
            conv_oper <= delay2;
        end case;
      elsif switchA_to_c = '1' then
        case current_state is
          when idle =>
            if start_in = '1' then
              next_state <= first;
              conv_oper <= delay2;
            end if;
          when first =>
            pwmout_buf <= "001001";
            --pwmout_buf(2) <= '0';
            next_state <= second;
            conv_oper <= delay2;
          when second =>
            if DutyCycle_Ca >= Min_val then
              pwmout_buf <= "111001";--switch to phase b
              --pwmout_buf(3) <= '1';
            elsif DutyCycle_Aa >= Min_val then
              pwmout_buf <= "001011";--switch to phase c
            end if;
            --pwmout_buf(5) <= '1';
            --pwmout_buf(4) <= '1';
            next_state <= idle;
            conv_oper <= delay2;
        end case;
      elsif switchA_to_a = '1' then
        case current_state is
          when idle =>
            if start_in = '1' then
              next_state <= first;
              conv_oper <= delay2;
            end if;
```

```
                when first =>
                  pwmout_buf <= "001001";
                  --pwmout_buf(5) <= '0';
                  --pwmout_buf(4) <= '0';
                  next_state <= second;
                  conv_oper <= delay2;
                when second =>
                  if DutyCycle_Aa >= Min_val then
                    pwmout_buf <= "001011";--switch to phase b
                    --pwmout_buf(3) <= '1';
                  elsif DutyCycle_Ba >= Min_val then
                    pwmout_buf <= "001101";--switch to phase c
                  end if;
                  --pwmout_buf(1) <= '1';
                  next_state <= idle;
                  conv_oper <= delay2;
              end case;
            end if;
          end case;
          if ((section_int > 30) and (section_int < 40)) then
            conv_oper <= start;
          end if;

        when Four =>
          case conv_oper is
            when start =>
              pwmout_buf <= "101001";
              conv_oper <= delay1;
              current_state <= idle;
            when delay1 =>
              if count = Deadtime then
                count <= 0;
                pwmout_buf <= "100001";
--                pwmout_buf(1) <= '1';
                conv_oper <= commutate;
              else
                count <= count + 1;
              end if;
            when commutate =>
              if switchA_to_b = '1' then
                case current_state is
                  when idle =>
                    if start_in = '1' then
                      next_state <= first;
                      conv_oper <= delay2;
                    end if;
                  when first =>
                    pwmout_buf <= "100001";
                    --pwmout_buf(1) <= '0';
                    next_state <= second;
                    conv_oper <= delay2;
                  when second =>
                    if DutyCycle_Ba >= Min_val then
                      pwmout_buf <= "101101";--switch to phase b
                      --pwmout_buf(3) <= '1';
                    elsif DutyCycle_Ca >= Min_val then
                      pwmout_buf <= "110001";--switch to phase c
```

```
            end if;
            --pwmout_buf(3) <= '1';
            --pwmout_buf(2) <= '1';
            next_state <= idle;
            conv_oper <= delay2;
        end case;
      elsif switchA_to_c = '1' then
        case current_state is
          when idle =>
            if start_in = '1' then
              next_state <= first;
              conv_oper <= delay2;
            end if;
          when first =>
            pwmout_buf <= "100001";
            --pwmout_buf(3) <= '0';
            --pwmout_buf(2) <= '0';
            next_state <= second;
            conv_oper <= delay2;
          when second =>
            if DutyCycle_Ca >= Min_val then
              pwmout_buf <= "110001";--switch to phase b
              --pwmout_buf(3) <= '1';
            elsif DutyCycle_Aa >= Min_val then
              pwmout_buf <= "100011";--switch to phase c
            end if;
            --pwmout_buf(4) <= '1';
            next_state <= idle;
            conv_oper <= delay2;
        end case;
      elsif switchA_to_a = '1' then
        case current_state is
          when idle =>
            if start_in = '1' then
              next_state <= first;
              conv_oper <= delay2;
            end if;
          when first =>
            pwmout_buf <= "100001";
            --pwmout_buf(4) <= '0';
            next_state <= second;
            conv_oper <= delay2;
          when second =>
            if DutyCycle_Aa >= Min_val then
              pwmout_buf <= "100011";--switch to phase b
              --pwmout_buf(3) <= '1';
            elsif DutyCycle_Ba >= Min_val then
              pwmout_buf <= "101101";--switch to phase c
            end if;
            --pwmout_buf(1) <= '1';
            next_state <= idle;
            conv_oper <= delay2;
        end case;
      end if;
    when delay2 =>
      if count = Deadtime then
        count <= 0;
```

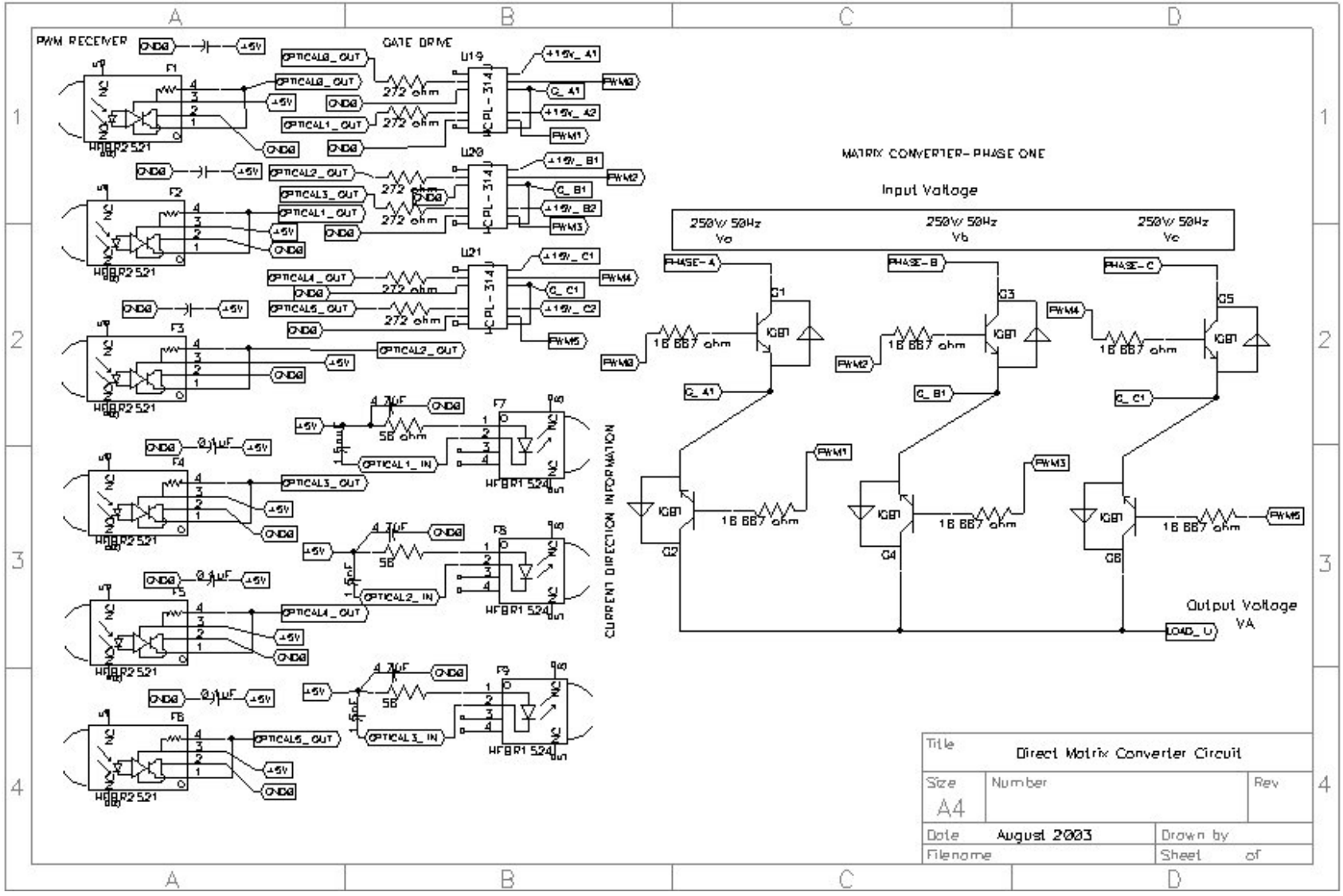
```
                conv_oper <= commutate;
                current_state <= next_state;
              else
                count <= count + 1;
              end if;
            end case;
            if ((section_int > 40) and (section_int < 50)) then
              conv_oper <= start;
            end if;

          when Five =>
            case conv_oper is
              when start =>
                pwmout_buf <= "100101";
                conv_oper <= delay1;
                current_state <= idle;
              when delay1 =>
                if count = Deadtime then
                  count <= 0;
                  pwmout_buf <= "100100";
--                 pwmout_buf(1) <= '1';
--                 pwmout_buf(0) <= '1';
                  conv_oper <= commutate;
                else
                  count <= count + 1;
                end if;
              when commutate =>
                if switchA_to_b = '1' then
                  case current_state is
                    when idle =>
                      if start_in = '1' then
                        next_state <= first;
                        conv_oper <= delay2;
                      end if;
                    when first =>
                      pwmout_buf <= "100100";
                      --pwmout_buf(1) <= '0';
                      --pwmout_buf(0) <= '0';
                      next_state <= second;
                      conv_oper <= delay2;
                    when second =>
                      if DutyCycle_Ba >= Min_val then
                        pwmout_buf <= "101100";--switch to phase b
                        --pwmout_buf(3) <= '1';
                      elsif DutyCycle_Ca >= Min_val then
                        pwmout_buf <= "110100";--switch to phase c
                      end if;
                      next_state <= idle;
                      conv_oper <= delay2;
                  end case;
                elsif switchA_to_c = '1' then
                  case current_state is
                    when idle =>
                      if start_in = '1' then
                        next_state <= first;
                        conv_oper <= delay2;
                      end if;
```

```
                when first =>
                  pwmout_buf <= "100100";
                  --pwmout_buf(3) <= '0';
                  next_state <= second;
                  conv_oper <= delay2;
                when second =>
                  if DutyCycle_Ca >= Min_val then
                    pwmout_buf <= "110100";--switch to phase b
                    --pwmout_buf(3) <= '1';
                  elsif DutyCycle_Aa >= Min_val then
                    pwmout_buf <= "100111";--switch to phase c
                  end if;
                  --pwmout_buf(4) <= '1';
                  next_state <= idle;
                  conv_oper <= delay2;
              end case;
            elsif switchA_to_a = '1' then
              case current_state is
                when idle =>
                  if start_in = '1' then
                    next_state <= first;
                    conv_oper <= delay2;
                  end if;
                when first =>
                  pwmout_buf <= "100100";
                  --pwmout_buf(4) <= '0';
                  next_state <= second;
                  conv_oper <= delay2;
                when second =>
                  if DutyCycle_Aa >= Min_val then
                    pwmout_buf <= "100111";--switch to phase b
                    --pwmout_buf(3) <= '1';
                  elsif DutyCycle_Ba >= Min_val then
                    pwmout_buf <= "101100";--switch to phase c
                  end if;
                  --pwmout_buf(1) <= '1';
                  --pwmout_buf(0) <= '1';
                  next_state <= idle;
                  conv_oper <= delay2;
              end case;
            end if;
          when delay2 =>
            if count = Deadtime then
              count <= 0;
              conv_oper <= commutate;
              current_state <= next_state;
            else
              count <= count + 1;
            end if;
        end case;
        if ((section_int > 50) and (section_int < 60)) then
          conv_oper <= start;
        end if;
    end case;
  end if;
end process;
```

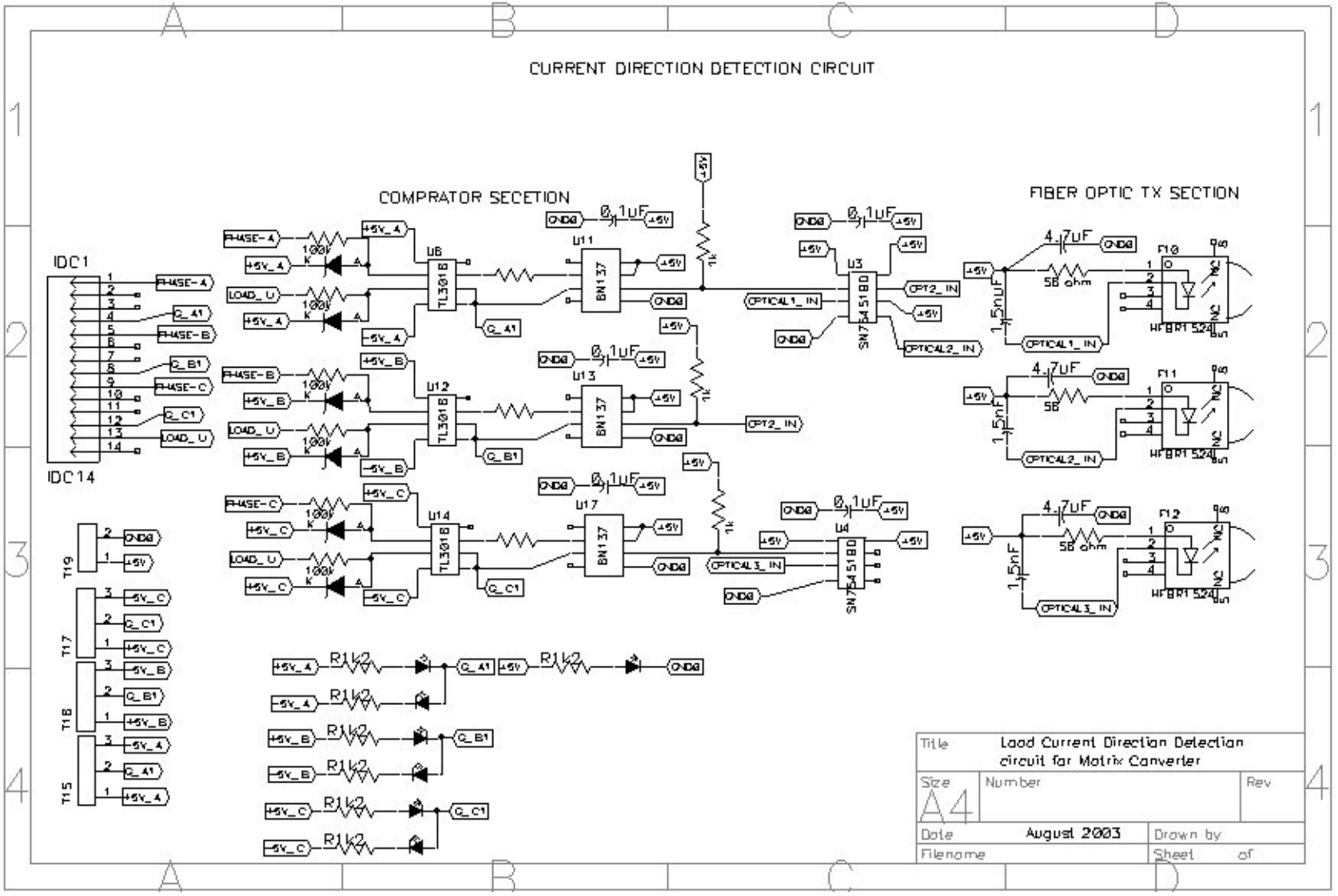
```
  p2:process(clk,nreset)
  begin
    if (nreset = '0') then
      section <= Five;

    elsif (clk'event and clk= '1') then
      section_int <= conv_integer(unsigned(section_value));

      if (section_int < 10) then
        section <= One;
      elsif ((section_int > 10) and (section_int < 20)) then
         section <= Two;
      elsif ((section_int > 20) and (section_int < 30)) then
         section <= Three;
      elsif ((section_int > 30) and (section_int < 40)) then
        section <= Four;
      elsif ((section_int > 40) and (section_int < 50)) then
        section <= Five;
      elsif ((section_int > 50) and (section_int < 60)) then
        section <= Six;
      end if;
    end if;
  end process;

  p3:process(clk,nReset)
  begin
    if (nReset = '0') then
      ampA  <= 0;
      DutyCycle_Aa <= 0;
      DutyCycle_Ba <= 0;
      DutyCycle_Ca <= 0;
      start_in  <=   '0';
    elsif (clk'event and clk='1') then
      ampA <= conv_integer(unsigned(amp));
      DutyCycle_Aa <= conv_integer(unsigned(Duty_cycle_phase_Aa));
      DutyCycle_Ba <= conv_integer(unsigned(Duty_cycle_phase_Ba));
      DutyCycle_Ca <= conv_integer(unsigned(Duty_cycle_phase_Ca));

      if (ampA=DutyCycle_Aa) and (DutyCycle_Aa >= 350) then
--Start commutating from phase a-to-b of load phase-A
        switchA_to_b <='1';
        switchA_to_a <='0';
        switchA_to_c <='0';
        start_in <= '1';

      elsif ampA=(DutyCycle_Aa + DutyCycle_Ba) and (DutyCycle_Ba >= 350) then
--Start commutating from phase b-to-c of load phase-A
        switchA_to_c <='1';
        switchA_to_b <='0';
        switchA_to_a <='0';
        start_in <= '1';
      elsif ampA = MAX and (DutyCycle_Ca >= 350) then
--Start commutating from phase c-to-a of load phase-A
        switchA_to_a <='1';
        switchA_to_c <='0';
        switchA_to_b <='0';
        start_in <= '1';
```

```
      else
        start_in <='0';
      end if;

      if ampA=5700 then
        nINT_0_buf <= '0';
      else
        nINT_0_buf <= '1';
      end if;

    end if;

  end process;

  nINT_0      <= nINT_0_buf; --interrupt request to DSP
  pwmoutA <= pwmout_buf; --PWM control signal to gate0drive


end RTL;
```

Appendix B: A 3Ø-1Ø direct matrix converter schematics



Appendix C: Current direction detection circuit.



Appendix D: Block diagram of the 3Ø-3Ø direct matrix converter circuit.