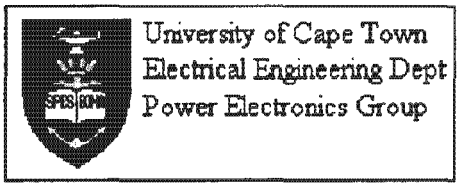
The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

# University of Cape Town




# DSP Controlled Three Phase to Single Phase Uninterruptible Power Supply

University of Cape Town

**Prepared by:** D G Beber
University of Cape Town
South Africa

**Prepared for:** Mr. M. Malengret
Department of Electrical Engineering
University of Cape Town

**Due Date:** 30 March 2001

# Acknowledgements

The author wishes to thank the following people for their invaluable contribution towards this project:

- Mr. Michel Malengret, my thesis supervisor for his support and for making this project possible
- The Reaserch Group: Azeem, Huey, Elvis, Sven, Sam and the technical Staff Chris, who were always willing to help when needed and who made my time spent at UCT very enjoyable.

University of Cape Town

# Terms of Reference

This Project was commisioned by Mr. Michel Malengret of the Electrical Engineering Department, University of Cape Town. The project was completed in order to achieve an MSc degree in Electrical Engineering. The objective was to design and build a Texas Instruments DSP controlled three to single phase UPS system according to the following specifications:

- Build a three phase to single phase UPS system using a Texas Instruments DSP
- Control the DC-Bus voltage using the phase angle technique
- The UPS should draw sinusoidal currents from the mains supply at close to unity power factor
- Phase lock the inverter output to the mains voltages using the space phasor technique

University of Cape Town

# Synopsis

With the increase in use of electronic equipment such as personal computers, network stations and AC drive controllers, an increased "polution" of the AC mains has resulted in the form of unwanted harmonics generated by the switching currents associated with these devices. Combined with the problem of mains "pollution" is the problem of balancing a large number of single phase loads onto a three phase supply, such as in a large computer network. This has resulted in more frequent supply voltage failure, causing substantial data and financial loss and in the extreme case death due to hospital equipment not being powered. Thus the need for an uninterruptible power supply (UPS) which is capable of resolving these problems. Taking these considerations into account, a three to single phase converter topology was developed and tested, using a Texas Instruments TMS320F243 Digital Signal Processor (DSP) for control of the system. The topology is based on a technique that utilises minimal semiconductor devices whilst still allowing for sinusoidal current to be drawn from the mains supply. This is achieved by linking the utility mains to a three phase IGBT module through link inductors and controlling the power flow to and from the utility mains by advancing and retarding the phase between the respective voltage space phasors. This in turn allows for the converter to operate as a phase controlled battery charger during normal mains supply by controlling the DC-Bus Voltage. It also allows for real power to be drawn equally from the three phases and supplied to the single phase loads, thus providing the functionality of load balancing.

In this thesis the converter was controlled using the Space Vector PWM technique, due to its advantages over other regularly sampled PWM strategies. It has the advantage of providing better DC-Bus Voltage Utilisation and less harmonic distortion compared to regularly sampled PWM. The calculation and generation of the Space Vector PWM signals was implemented in software on the DSP evaluation board.

Results from a 650W test system show that the system was able to draw balanced power from the three phase mains supply, and supply it to a variable single phase load, while maintaining a constant DC-Bus Voltage. Results show that the input powerfactor for the system was maintained at close to unity throughout the power range.

University Of Cape Town

# Table of Contents



University of Cape Town

University of Cape Town

University of Cape Town

# List of Figures



University of Cape Town

University of Cape Town

University of Cape Town

# List of Tables



University of Cape Town

# Chapter 1
# Introduction

This thesis aims to investigate the viability of a new component minimised Uninterruptible Power Supply (UPS) topology. The proposed topology consists of a single full-bridge three phase IGBT module, which rectifies the incoming three phase supply, and acts as both an inverter and phase controlled battery charger. The topology also allows for power to be drawn equally from a three phase system, and supply a regulated single phase output to the load.

## 1 The need for an Industrial three to single phase UPS

The need for Uninterruptible Power Supplies has grown dramatically over the last decade. This growth in the market has been fuelled by the increased use of electronic equipment such as personal computers, network stations and AC drive controllers.

The proliferation of electronic equipment has resulted in the "pollution" of the ac mains in the form of unwanted harmonics. Switching currents associated with computer power supplies, motor drives and inverters inject these harmonics into the local power grid.

An increased number of consumers connected to the local power grid, combined with the problems associated with harmonic injection (reactive currents, losses etc.) have reduced the ability of the local grid to supply quality power to its users. This has meant more frequent line sags, over-voltages, spikes and power failures.

The effect of a total loss of power depends on the duration of the failure and the equipment design. Personal computer power supplies are designed to maintain output for a maximum of two cycles (»100ms), [1] . After which a logic signal is generated within the computer that allows the CPU a further 50ms to back up existing information. Beyond this all power supply voltages decay rapidly, and the user will lose any information not backed up. Uninterruptible power supplies are thus vital for critical systems, as a power outage can result in substantial data and financial loss.

The function of the UPS system is to act as a buffer and provide clean and reliable power to the critical load in question. Generally power is stored in a backup battery set during normal operation and is supplied to the load through dc to ac conversion during a power failure or outage

## 2 Objectives of this Thesis

This thesis presents the results of tests carried out on a prototype of the proposed online three to single phase UPS system. The objectives of this thesis were to:

University of Cape Town

- Develop a three to single phase UPS based on a Texas Instruments DSP, using the phase shift technique
- Implement Space Vector Modulation
- Implement phase locking based on the three phase voltage space phasor
- Test the UPS system under various load conditions

This thesis begins with an overview of uninterruptible power supply topologies, where the different merits of the available topologies is briefly discussed. Then the proposed topology is introduced and the the relevant theory for this project is covered. A detailed explanation of the software implementation and hardware design process used for this project is then described, and the control strategy is explained. Finally results of the built UPS are given and based on these results conclusions and recommendations are made.

University of Cape Town

# Chapter 2
# Review of Uninterruptible Power Supply Topologies

## 1 Introduction

This chapter outlines commercially available uninterruptible power supply topologies, and the merits of each are discussed and compared. The proposed topology is then presented with its relevant merits.

UPS systems are intended to provide the following two functions:

- Power quality improvement
- Backup source of power

The function of power quality improvement is implemented to different degrees in the various UPS topologies. Possible power quality improvements include the following:

- Correction of supply under/overvoltage caused by sudden changes in system load
- Correction of harmonic distortion caused by converters connected to the same circuit
- Filtering of surges/spikes caused by system faults, sudden load changes etc.

Although it is possible to minimise some of these effects using low cost passive devices to filter the power being supplied to the load, these passive devices are unable to correct for power outages and frequency deviation. An uninterruptible power supply by definition is capable of protecting against these disturbances by having its own source of power.

Uninterruptible power supplies are generally classified as being either *online* or *standby*, depending on which path is chosen as the main power path. Other classifications such as *Standby-ferro* and *line interactive* are variations of these topologies and are further described below.

University of Cape Town

## 2 Currently available UPS Topologies

Following is a short description of the currently available UPS topologies, outlining each topologies merits. **Note:**In the following UPS topology diagrams, the solid power path is the primary power path, and the dashed path is the backup power path.

### 2.1 Online and Standby UPS

The online and standby UPS topologies are essentially the same, the difference being which power path is chosen as the primary power path. Fig.2-1 shows the online UPS topology. In this configuration the bypass switch is set such that the inverter supplies power to the load under

normal operation. During this mode of operation, all of the current supplied to the load passes through the rectifier/battery charger and the inverter, therefore incurring a power loss of 25-30% [12] . In the event of a mains failure the inverter continues supplying the load until the battery backup power is depleted. The bypass switch is only activated in the event of an inverter failure, and thus allows for the continuation of unregulated but filtered power from the utility supply.

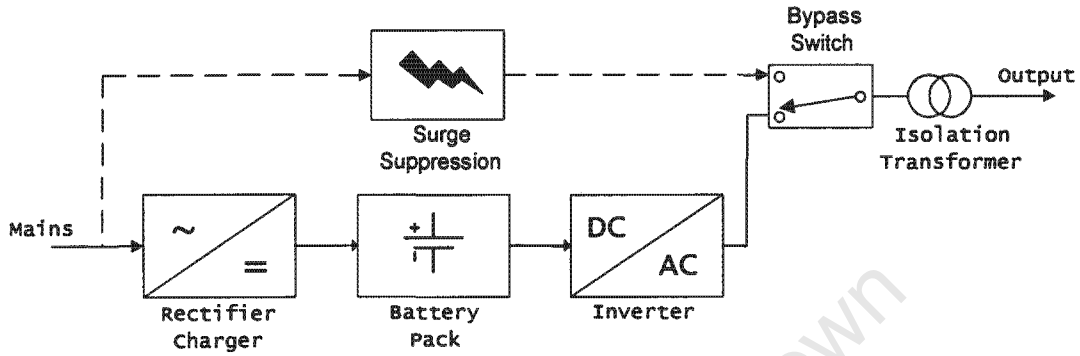



Figure 2-1.Online UPS Topology

In the case of a standby UPS the filtered ac mains is chosen as the primary power path, as shown in Fig.2-2. On a mains failure the transfer switch is activated and power is then supplied to the load from the inverter. The standby topology has the following advantages over the online UPS topology:

- the battery charger can be sized much smaller than in the online UPS topology, as it only has to supply enough current to charge the batteries
- the system efficiency is high because the inverter is active only when the mains supply fails

It however has the disadvantage that on a power failure there is a transfer time (4-30milliseconds), and thus it is not truly an uninterruptible power supply. A variation on this topology, which tries to address this is the standby ferro UPS which is described below.

University of Cape Town

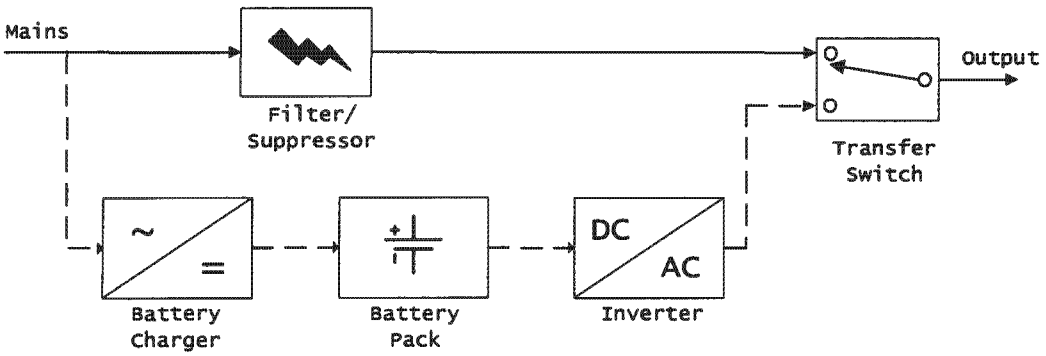



Figure 2-2.Standby UPS Topology

## 2.2 Standby ferro UPS

This topology tries to eliminate the transfer time associated with the standby UPS topology, by including a ferro-resonant transformer, as shown in Fig.2-3. When the mains supply fails, the energy stored in the transformer as flux is used to supply the load curent until the inverter has been switched online. Although this topology is classified as a standby UPS, the ferro-resonant transformer is constantly online and due to it being inherently inefficient it produces a lot of heat. Despite the fact that this makes the topology less efficient, the ferro-resonant transformer does however have the advantage that it acts as a isolation transformer and is also capable of voltage regulation over a limited range.

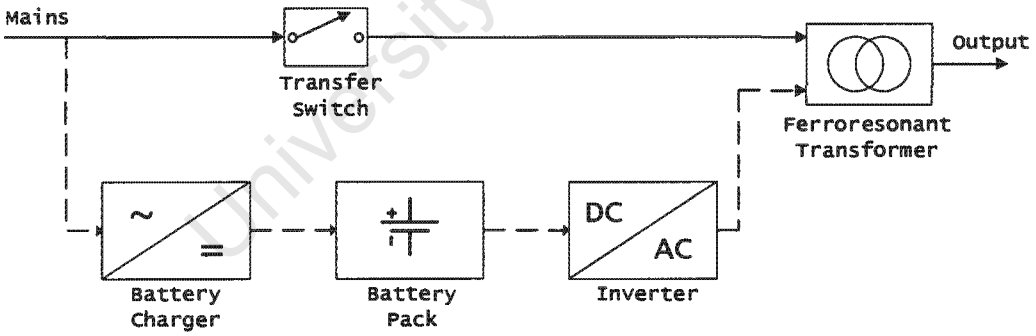



Figure 2-3.Standby Ferro UPS Topology

## 2.3 Line Interactive UPS

In this design, shown in Fig.2-4, the battery/inverter is permanently connected to the output of

University of Cape Town

the UPS. During normal operation power is supplied to the load from the mains, and the inverter is operated in reverse at low power to charge the battery set. On a mains failure, the transfer switch is opened to prevent power flow back into the mains, and power is supplied to the load from the batteries through the inverter. This topology combines the performance benefits (no transfer time) of the online UPS with the reliability and efficiency benefits of the standby UPS.

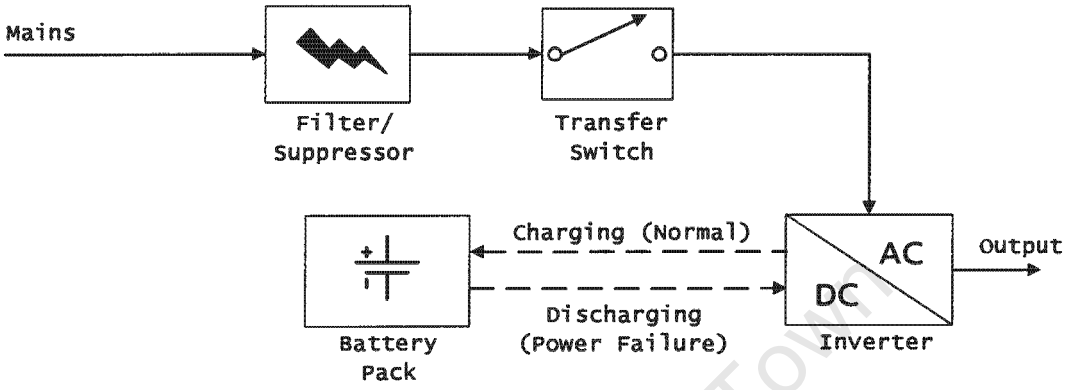



Figure 2-4.Line Interactive UPS Topology

As the inverter is connected permanently to the output, an additional advantage is that improved filtering and reduced switching losses are achieved over the conventional standby UPS. This means that continuous regulation of the output voltage is possible, allowing for brown out correction without the associated transfer time.

## 2.4 Proposed UPS

The proposed UPS topology is shown in Fig.2-5. It is essentially a variation of the line interactive topology that has been component minimised. This topology is suited for supply of single phase loads such as in large computer networks, due its ability to draw balanced power from the utility and supply it to an unpredictable combination of parallel single phase loads. During a power failure the UPS system is diconnected from the mains supply by a contactor and power is supplied from the battery set to the single phase loads until the mains voltages are restored, or the battery power is depleted.

This configuration has a number of advantages, in addition to the advantages of the line interactive topology, namely:

- There is no transfer time on power failure
- High efficiency during normal[1] operation

[1] Normal refers to the mode of operation whereby the mains voltages are at their rated values and the battery set is fully charged

University of Cape Town

- Low implementation cost due to it being component minimised
  (Both rectification and inversion are carried out by a single three phase IGBT module)
- Capable of operating at near unity power factor
- Draws balanced power from the utility, eliminating problem of unbalanced single phase loads

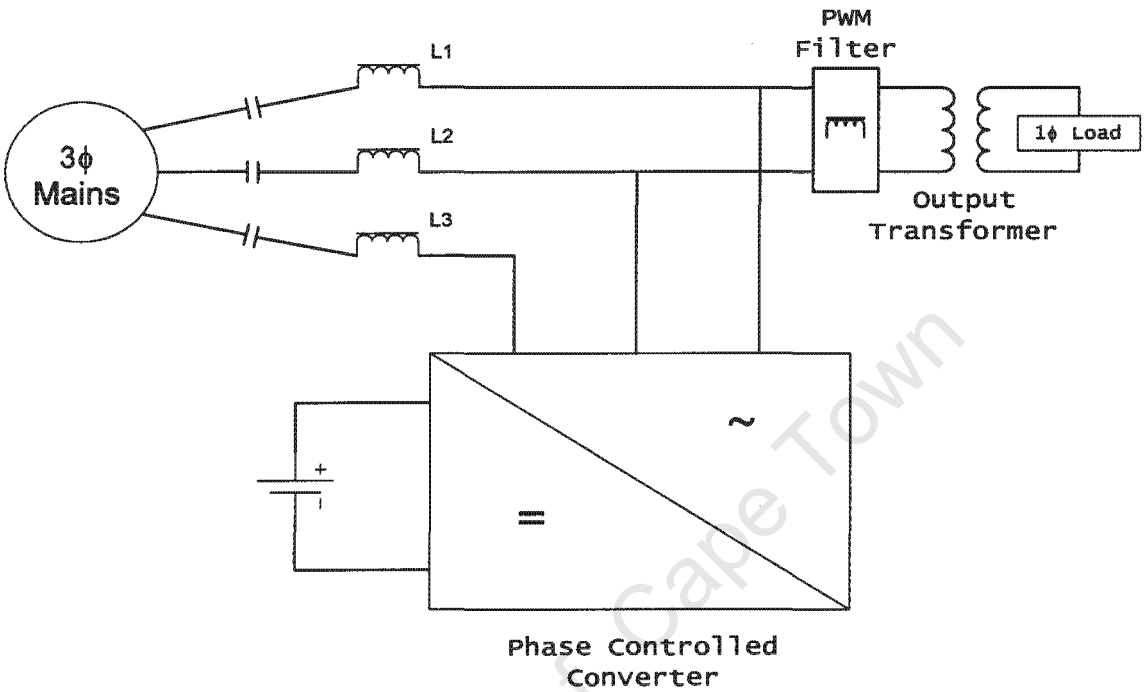



Figure 2-5.Proposed UPS Topology

It should be noted that the line to line output voltage from the inverter is of the same magnitude as the line to line voltage of the three phase mains supply. It is thus necessary to have an output transformer as shown in Fig.2-5, so as to supply the correct voltage level for the single phase load.

University of Cape Town

# Chapter 3

# Proposed Topology Description

## 1 Power Circuit description

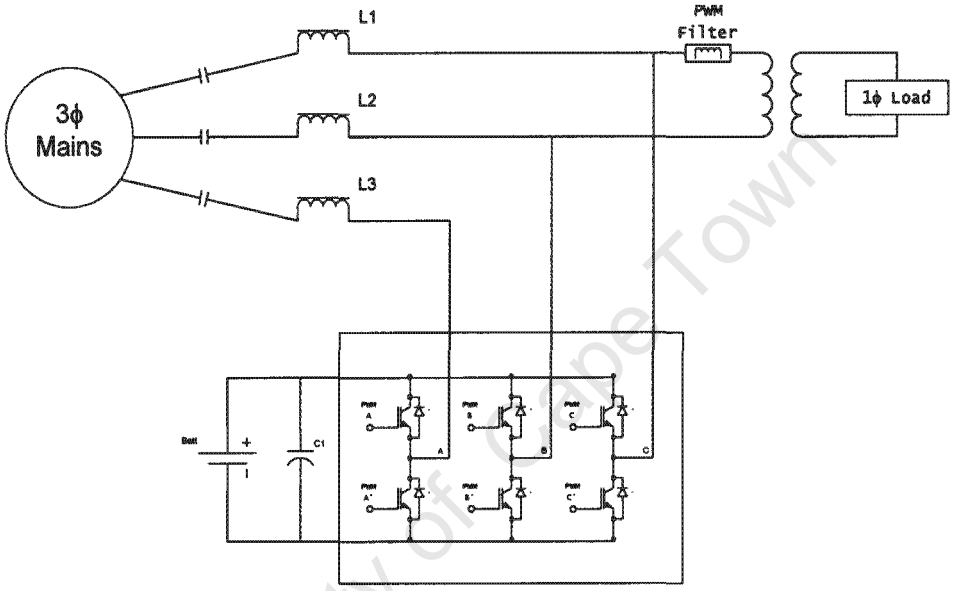



Figure 3-1.Proposed Power Circuit Schematic

The configuration of the proposed UPS topology is shown in Fig.3-1. It consists of a component minimised power circuit, using fewer semicondutor devices, making it a cost effective design. The design utilises a single, three phase full bridge IGBT module for rectification of the ac mains which is operated in a PWM fashion to supply a regulated ac voltage to the load. The link inductors L1,L2,L3 provide an interface between the mains supply and the output of the inverter, allowing this minimised topology to:

- Act as a phase controlled battery charger during normal operation
- Operate bi-directionally, either as an inverter or battery charger
- Operate at close to unity power factor during normal operation
- Suppress transients between the mains supply and the output

By controlling both the magnitude and phase angle of the inverter output, relative to the

University of Cape Town

incoming mains it is possible to control the flow of power through the inductors. By using this strategy it has also been suggested that it is possible to operate the UPS at close to unity power factor during normal operation, supplying a regulated voltage output to the load. The function of the output filter is to remove the high switching frequencies associated with PWM so as to supply a sinusoidal current to the load.

**NOTE:** As the UPS is of the Line interactive type, the inverter is run continuously and on a mains failure the power source is transferred seamlessly to the inverter battery set. This mode of operation continues until the battery set is either depleted, or mains supply returns.

# 2 Theory of operation[3]

The principle of operation of the proposed topology, can best be explained with the help of an equivalent single phase circuit as shown in Fig.3-2, and its phasor diagram Fig.3-3. Essentially the power flow from the mains supply through the link inductor can be controlled by varying the phase angle $\delta$ between the mains supply $V_{in}$ and the converter output voltage $V_{out}$.

## 2.1 Control of Power Flow

In this equivalent circuit $X_z$ represents the link inductance, $I_{in}$ the supply current, $I_{out}$ the load current and $I_{ups}$ the converter current.

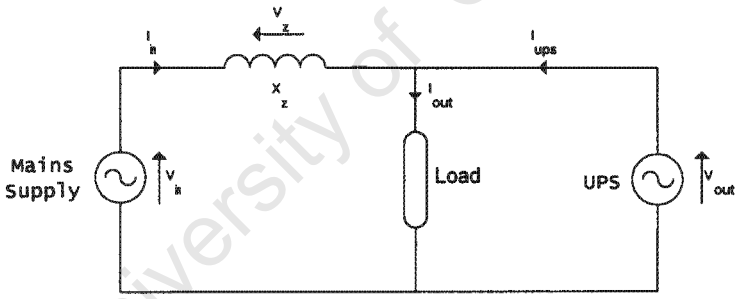



Figure 3-2.Single Phase Equivalent Circuit for proposed UPS

Assuming that the mains voltage and converter voltage are both sinusoidal, they can be expressed as follows:

$$v_{in}(t) = V_{in}\sin(\omega t) \quad (1)$$

$$v_{out}(t) = V_{out}\sin(\omega t - \delta) \quad (2)$$

which can be represented on the phasor diagram as two phasors displaced by an angle $\delta$.

Next an equation for the input current can be derived by analysing the equivalent circuit and

University of Cape Town

phasor diagram. Applying Kirchoff's voltage law to the equivalent circuit, the voltage across the link inductor $V_z$, can be represented by a phasor joining the converter output phasor $V_{out}$ to the mains supply phasor $V_{in}$. Now since the current through an inductor lags the voltage by 90 degrees, the input current phasor $I_{in}$ can be drawn at 90 degrees to $V_z$, lagging the input voltage by an angle $\theta$. The magnitude of the input current phasor $I_{in}$ can then be determined from the phasor diagram by simple trigonometry to be:

$$I_{in} = \frac{V_z}{X_z} = \frac{\sqrt{(V_{out} - V_{in}\cos(\delta))^2 + (V_{in}\sin(\delta))^2}}{X_z} \tag{3}$$

Similarly an equation for the load current can be derived by assuming that the load is reactive, meaning that the load current $I_{out}$ will lag the inverter output voltage by an angle $\beta$, and can be expressed in equation form as:

$$i_{out}(t) = I_{out}\sin(\omega t - \delta - \beta) \tag{4}$$

which can represented in Fig.3-3 as a phasor at an angle $\beta$ to $V_{out}$.

From these equations it should be noted that the input current is independent of the load current, and can be controlled by varying the phase angle $\delta$, and that the current for a given phase shift is determined by the size of the link inductor.

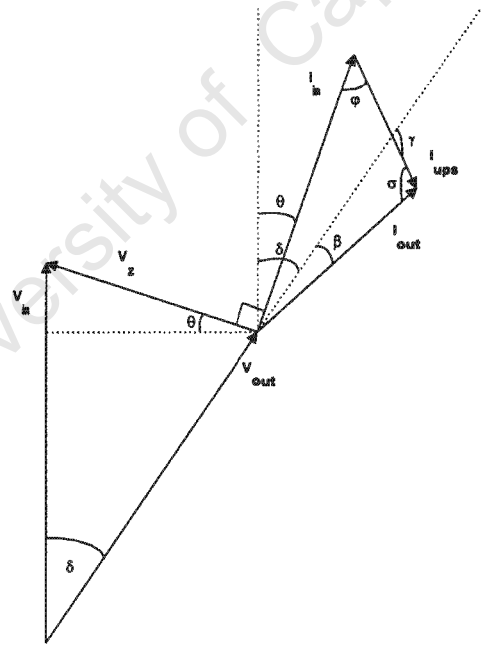



Figure 3-3.Phasor Diagram of Single Phase Equivalent Circuit

University of Cape Town

The central idea behind this topology is thus to control the flow of power through the link inductors and in doing so regulate the DC-Bus voltage, by varying the phase angle $\delta$. In order to be able to understand how to control the power flow, a relationship between the phase angle $\delta$ and the power flow needs to be derived. Using simple trigonometry and referring to Fig.3-3 the equation can be derived as follows:

$$V_{out}\sin(\delta) = V_z\cos(\theta) \tag{5}$$

and by applying this to the standard power equation Eqn.6[1] :

$$P = V_{in}I_{in}\cos(\theta) \tag{6}$$

and substituting for $\cos(\theta)$ we get:

$$P = \frac{V_{in}V_{out}}{X_z}\sin(\delta) \tag{7}$$

Which gives us the relationship between the flow of real power through the link inductors, and the phase angle $\delta$.

## 2.2 Power Factor Correction

Similarly from the phasor and equivalent circuit diagrams, we can derive an equation for the input power factor of the system, so as to be able to prove that a close to unity input power factor can be achieved during normal operation. Substituting for $I_{in}$ and $P$ in eqn.3, eqn.6 and eqn.7, an equation relating the input power factor of the system to $\delta$ can be derived to be:

$$\cos(\theta) = \frac{V_{out}\sin(\delta)}{\sqrt{(V_{out} - V_{in}\cos(\delta))^2 + (V_{in}\sin(\delta))^2}} \tag{8}$$

In order to prove that the inverter can be operated bidirectionally, it is then necessary to derive an equation relating the power flow into the UPS and the phase angle $\delta$. Using the standard power equation and refering to Fig.3-3, an equation for the real power flowing into the UPS can be defined as:

$$P_{ups} = V_{out}I_{ups}\cos(\gamma) \tag{9}$$

and by simple trigonometry it can be shown that:

$$\gamma = 180 - \varphi - \delta + \theta$$

By considering the above equations, it can be seen that by varying the phase angle $\delta$, we can:

- Control the angle $\gamma$, and thus control the real power flow into and out of the UPS. This allows for operation as either a phase controlled battery charger or inverter.
- Vary the input power factor of the system.

University of Cape Town

# Chapter 4

# Space Vector Modulation

Traditionally UPS's have used conventional analogue PWM techniques, whereby a triangular waveform is compared with a sinusoidal modulating function.Using this technique for a three phase system, requires three independent modulation stages, one for each of the phases. The control thus looks like three separate single phase controls rather than one control of a three phase system.

In recent years the implementation of PWM signals has however become a digital problem, due to the widespread use of digital microprocessors and it is therefore inevitable that some form of digital PWM algorithm is required. It is possible to generate a direct digital equivalent of the analogue technique, by using a close approximation to the traditional analogue sine-triangle PWM strategy. This however is still only an approximation to the analogue technique.

A more recent development is a technique known as space vector modulation. This scheme offers an inherently digital computation suitable for the processors available today. This method produces identical PWM signals to those obtained by comparing a suitable modulating waveform with a triangular switching waveform. The main advantages of this modulation technique are:

- Simple, inherently digital calculation
- 15% increase in dc-link utilisation compared with traditional sine-triangule techniques
- Lower harmonic distortion in output voltages, compared with conventional sine-triangle techniques.
- Less switching losses for the same THD

## 1 Space Vector Definition and Projection

The objective of the Space Vector PWM technique is to approximate a reference voltage $v_s$ instantaneously by a combination of switching states corresponding to the basic space vectors. To achieve this requires that for a small time period $T$ the average inverter output be the same as the average reference voltage output $v_s$.

University of Cape Town

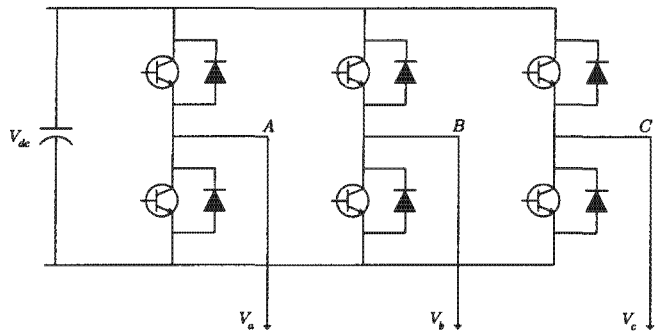



Figure 4-1.Three Phase Voltage Source Inverter

On the assumption that we are using a three phase full bridge voltage source inverter (VSI) as in Fig.4-1, the three-phase voltages can be analysed in terms of complex space vectors. Assuming $v_a, v_b, v_c$are the instantaneous voltages in the three phase system, then the complex voltage space vector can be defined by:

$$\overline{v}_s = \frac{2}{3}(v_a + \alpha v_b + \alpha^2 v_c) \tag{1}$$

where $\alpha = e^{j\frac{2}{3}\pi}$ and $\alpha^2 = e^{j\frac{4}{3}\pi}$ ,represent the spacial operators displaced by 120°. This can be seen more clearly in Fig.4-2.

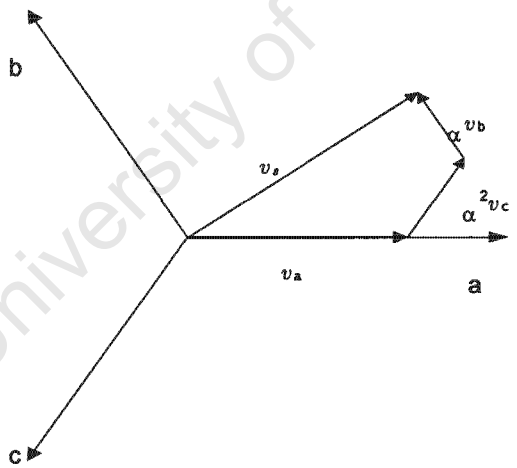



Figure 4-2.Voltage space vector in a,b,c reference frame

University of Cape Town

# 2 Generation of inverter switching states

For a three-phase voltage source inverter as in Fig.4-1 each pole voltage may assume only one of two values, depending on whether the upper or lower switch is turned on. Consequently there are only eight possible switching states for this configuration of inverter, as shown in Fig.4-3.

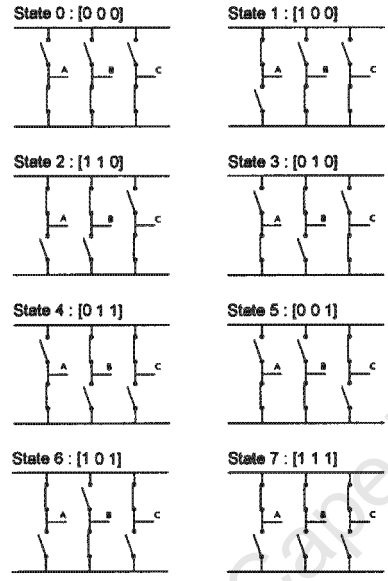



Figure 4-3.Three Phase VSI possible operating states

It emerges that of the eight inverter states, six of these are active states, State 1 to State 6, and two of them zero states, State 0 and 7. The six active states occur when either one upper and two lower or two lower and one upper inverter devices conduct simultaneously. The zero states occur when either the three upper or the three lower devices are turned on. These zero states are often referred to as freewheeling states, since all motor currents are freewheeling during operation in these configurations i.e. some of the freewheeling diodes are in the conduction state during the Zero States.

When connected to a three-phase AC motor, each inverter state imposes a set of voltages across the motor windings. During each inverter state, the phase voltages can be summed vectorially using Eqn.10, to obtain the resultant space-phasor in the machine.

For example for State1 corresponding to having the upper switch of inverter pole A and the lower switches of inverter poles B and C turned on simultaneously, the resultant inverter phase voltages are as shown in Table.4-1, where the reference point $N$ is taken as the negative DC-Bus. The resulting state $vs1$ is calculated by summing the inverter phase voltages vectorially using Eqn.10, and is shown in the complex space vector plane of Fig.4-4 as a vector of length

University of Cape Town

| $a$ | $b$ | $c$ | $V_{AN_1}$ | $V_{BN_1}$ | $V_{CN_1}$ | $v_{sk}$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | $V_{dc}$ | 0 | 0 | $\frac{2}{3}V_{dc}\exp(0)$ |
| 1 | 1 | 0 | $V_{dc}$ | $V_{dc}$ | 0 | $\frac{2}{3}V_{dc}\exp(\frac{\pi}{3})$ |
| 0 | 1 | 0 | 0 | $V_{dc}$ | 0 | $\frac{2}{3}V_{dc}\exp(\frac{2\pi}{3})$ |
| 0 | 1 | 1 | 0 | $V_{dc}$ | $V_{dc}$ | $\frac{2}{3}V_{dc}\exp(\pi)$ |
| 0 | 0 | 1 | 0 | 0 | $V_{dc}$ | $\frac{2}{3}V_{dc}\exp(\frac{4\pi}{3})$ |
| 1 | 0 | 1 | $V_{dc}$ | 0 | $V_{dc}$ | $\frac{2}{3}V_{dc}\exp(\frac{5\pi}{3})$ |
| 1 | 1 | 1 | $V_{dc}$ | $V_{dc}$ | $V_{dc}$ | 0 |

Table 4-1.Device On/Off States, corresponding Outputs of a Three-Phase VSI and resulting Space Vectors

$\frac{2}{3}V_{dc}$ lying along the real axis. Similarly, it is possible to compute the voltage space vectors for the other seven inverter states shown in Figure4-4.

The general expression for the eight realisable voltage vectors is therefore given by Eq.11:

$$v_{sk} = \begin{cases} \frac{2}{3}V_{dc}\exp(\frac{(k-1)\pi}{3}) & k = 1, \bigwedge, 6 \\ 0 & k = 0, 7 \end{cases} \tag{11}$$

The six active vectors subdivide the space vector plane into six equal sectors, and the two zero vectors $v_0, v_7$ are located at the centre of the hexagon, as in Fig.4-2.

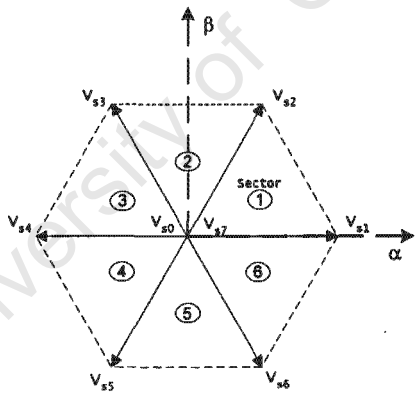



Figure 4-4.Space Vector Sector Representation

By using a combination of these eight voltage vectors any arbitrary voltage space vector within the bounds of the hexagon is realisable. However the maximum realisable fundamental voltage for a given level of DC bus voltage, obtained under six step operation, is only possible at the cost of significant low frequency distortion[8] .

For sinusoidal PWM it can be shown that the maximum achievable fundamental voltage is

University of Cape Town

78.5% of the value that would be reached by square wave operation (inverter capacity) and is given by Eqn.12[8] :

$$V_{1,\sin-pwm} = \frac{V_{dc}}{2} \tag{12}$$

Like conventional sinusoidal PWM, there is also a modulation index associated with Space Vector Modulation. The modulation index is defined as the ratio of the desired peak fundamental line to neutral magnitude to half the dc link voltage[8] :

$$M = \frac{V_1}{\frac{V_{dc}}{2}} \tag{13}$$

The corresponding reference voltage space vector can then be given by:

$$v_s = V_1 e^{-j\omega_m t} = M\frac{V_{dc}}{2}e^{-j\omega_m t} \tag{14}$$

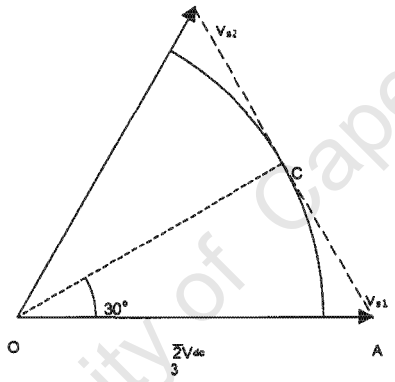



Figure 4-5.Derivation of Space Vector Maximum Modulation Index

From Fig.4-5 it can be seen that the maximum reference space vector attainable is a circle inscribing the hexagon. This circle is tangential to the midpoints of the lines connecting the active space vectors, and by simple trigonometry we can derive the maximum fundamental phase voltage attainable to be:

$$OA = \frac{2}{3}V_{dc}$$

$$OC = OA\cos(30) = \frac{V_{dc}}{\sqrt{3}}$$

and thus:

$$V_{1,svm} = \frac{V_{dc}}{\sqrt{3}} \tag{15}$$

University of Cape Town

therefore from the definition of modulation index Eqn.13, the maximum modulation index can be shown to be:

$$M_{\max} = \frac{V_{1,svm(\max)}}{\frac{V_{dc}}{2}} = 1.15 \tag{16}$$

which is a 15% increase over conventional sinusoidal pwm.

## 2.1 Generation of a Space Vector within a sector

The generation of a space vector can follow any trajectory within the limits of the inverter. The inverter however is only capable of generating eight discrete vectors of fixed magnitude. To be able to generate vectors inbetween these states it is necessary to switch between these states for a predetermined time. This can be more clearly explained using Fig.4-6:

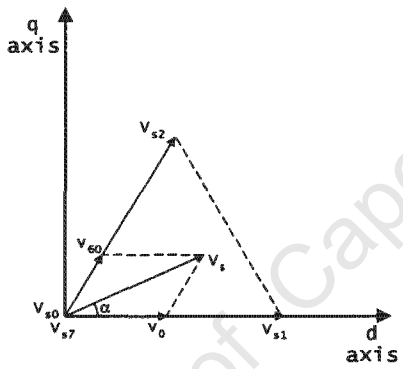



Figure 4-6.Generation of Voltage Space Vector from realisable voltage vectors

From this it can be seen that the required space vector $v_s$, can be created by summing the two vectors $v_0$ and $v_{60}$. "Although it is not necesary to use the two adjacent inverter states in the synthesis of the output voltage, it can be shown that superior harmonic performance is obtained when this condition is satisfied"[8] .Therefore:

$$v_s = v_0 + v_{60} \tag{17}$$

Where vectors $v_0$ and $v_{60}$ are obtained by time modulating vectors $v_{s1}, v_{s2}$ and the zero vectors $v_{s0}, v_{s7}$. Thus for any arbitrary space vector within this sector, the switching times for each vector are given by:

$$v_s T_s = v_{s1} T_{s1} + v_{s2} T_{s2} + v_{s0} T_{s0} + v_{s7} T_{s7} \tag{18}$$

where the total switching time $T_s$ is given by:

$$T_s = T_{s1} + T_{s2} + T_{s0} + T_{s7} \tag{19}$$

University of Cape Town

From Eqn.18and Eqn.19 above it can be seen that since $v_{s0}, v_{s7}$ are both equal to zero, their respective times $T_{s0}, T_{s7}$ control the length of the vector $v_s$. Likewise it can be seen that the times for vectors $T_{s1}, T_{s2}$ control the angle at which the vector $v_s$ is positioned at.

The non-zero vector switching times can be given by Eqn.20 and Eqn.21[11] :

$$T_{s1} = \frac{\sqrt{3}}{2} T_s M_a \sin(\frac{\pi}{3} - \alpha) \tag{20}$$

$$T_{s2} = \frac{\sqrt{3}}{2} T_s M_a \sin(\alpha) \tag{21}$$

Knowing $T_s, T_{s1}, T_{s2}$, times $T_{s0}$ and $T_{s7}$ can be determined from Eqn.19, and from the type of modulation scheme chosen. In the technique chosen for this project, termed "double edged space vector modulation" the remaining time is split equally between $T_{s0}$ and $T_{s7}$.

## 2.2 Generation of the switching signals

To be able to achieve the minimum possible switching losses in the generation of space vector modulation, it is necessary to minimise the inverter switching frequency. This can be achieved if the transition from one inverter state to another is obtained by switching only one inverter pole[8] . A further reduction in switching frequency can be achieved by using "bus clamped modulation techniques"[9] , where only one zero state is used in each switching period. These however introduce extra harmonics, when using "dead times" and thus will not be discussed further in this document.

In order to avoid introducing extra harmonics into the system when using "dead times", it is convenient to start and end each switching period in one or other zero state, as they are common to all sectors. All of these conditions can be met by the use of symmetrical switching patterns as shown in Fig.4-7 below.

University of Cape Town

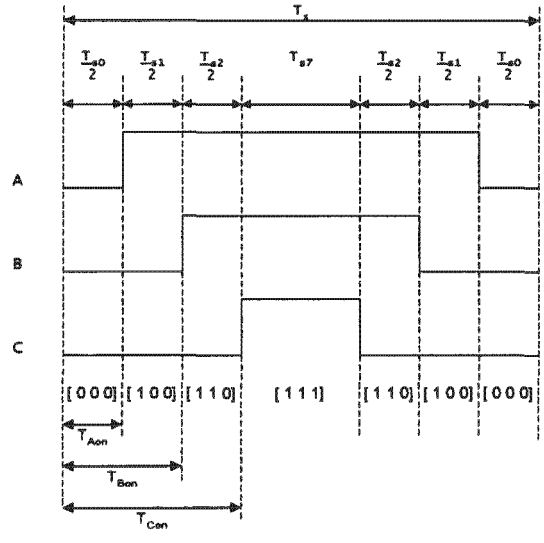



Figure 4-7.SVM Switching Times

This switching technique is termed "double edge space vector modulation" and is by far the most common technique used due to its harmonic characteristics[8] . Fig.4-7 shows for sector1 how the switching times $T_{Aon}, T_{Bon}, T_{Con}$ for inverter poles A,B,C are derived, from vector times $T_{s1}, T_{s2}, T_{s0}, T_{s7}$ for the modulation technique used in this project, completing the generation of the Space Vector.

University of Cape Town

# Chapter 5

# Design Considerations

## 1 Introduction

The operation of the uninterruptible power supply is based on the control of power flow between the utility mains and the inverter/load. Although this is the primary objective of the system, it is also necessary to ensure that the other goals of the project are also achieved. Some of the desired objectives of the project are:

- Operation at close to unity input power factor during normal operation
- Maintain a controlled DC-Bus Voltage, to be able to regulate battery set charging
- Inverter to run at maximum efficiency (Minimise real power consumption by inverter)
- Maintain a regulated voltage to the load

Since some of these objectives are only obtainable at the cost of sacrificing one of the other objectives, it is thus necessary to find a suitable compromise solution.

University of Cape Town

## 2 Design Considerations

In deciding the control strategy for this Uninterruptible power supply, it was necessary to investigate the relationships between phase angle, inductor size, input power flow and input power factor. Following are the results of this analysis.

### 2.1 Phase Angle / Input Power relationship vs. Inductor size

The central idea of this thesis is to control the flow of power through the link inductors by controlling the phase angle $\delta$, and at the same time regulate the DC-Bus voltage to enable battery set charging. The flow of power however is determined by the choice of the link inductor and the relationship is given by Eqn.7. This gives us the real power flowing from the mains through a single phase link inductor, for a three phase system as in our inverter the power flow will be three times this value as in Eqn.22.

$$P_{in-3ph} = 3\frac{V_{in}V_{inv}}{X}\sin(\delta) \tag{22}$$

In order to visualise this relationship more clearly, a three dimensional plot of the equation can be made as in Fig.5-1, resulting in a surface. From this it can be seen that as the phase angle $\delta$ is increased, the power flowing through the link inductors increases up to a maximum

at $1.57rad$ (90°). The rate at which the power increases for a given change in phase angle can be seen to be affected by the inductor size. The smaller the inductor the faster the rise in power flow for a given change in phase angle, thus as the power rating of the required system increases the link-inductor size decreases. It should be noted that negative values of phase angle are possible and would result in an identical surface, but in the negative power range, meaning that power would be flowing from the converter to the utility mains.

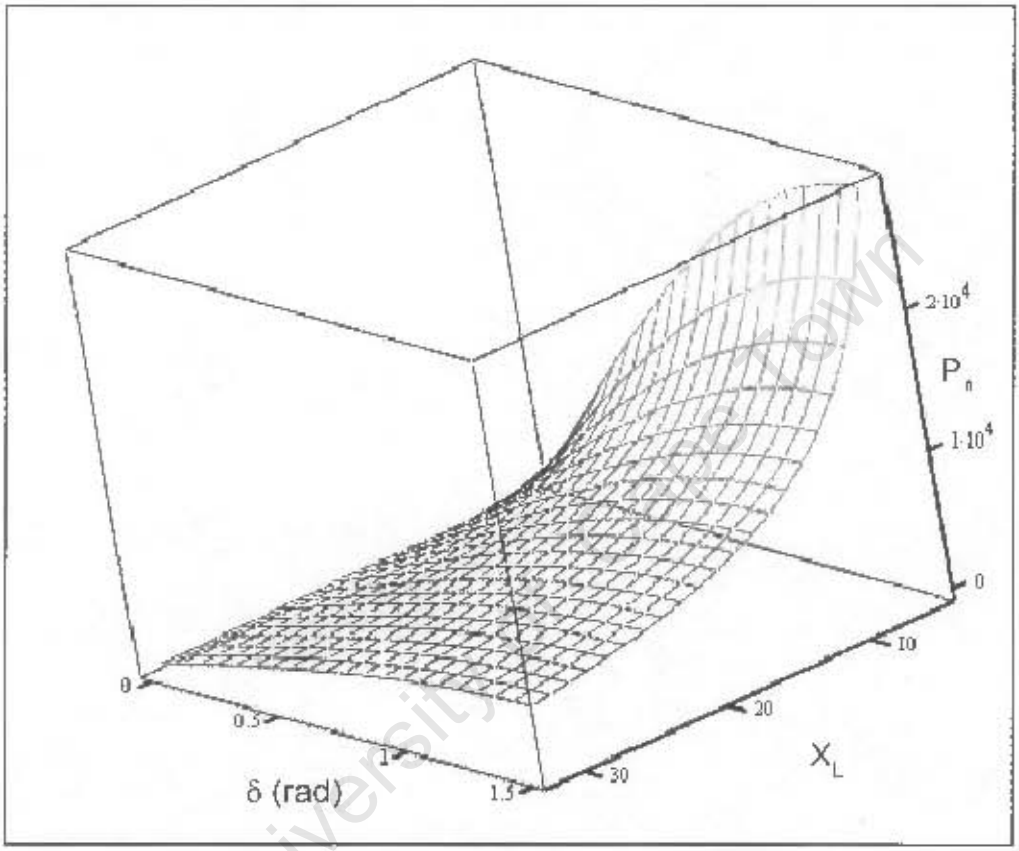



Figure 5-1.Relationship between Input power, inductor size and Phase Angle

In order to simplify the process of control of the input power it would be beneficial to select a large value of inductor as this will reduce the sensitivity of the change of input power with respect to a change in phase angle, making the system easier to control. It will thus not be necessary to have a high resolution on the phase angle to be able to accurately control the power flow. From Fig.5-1 it can be seen however that this would limit the maximum possible input power that can be transferred, and thus a compromise solution is required.

University of Bath

## 2.2 Phase Angle / DC-Bus Voltage relationship vs. Inductor size

Since one of the other main objectives of a UPS is to regulate the DC-Bus voltage, so as to implement the functionality of a phase controlled battery charger, it is important to find the relationship between $\delta$ and the DC-Bus voltage with respect to the choice of inductor size.

Since it is easier to obtain this relationship for a specific operating condition, a relationship is derived for the normal operating condition of the system. As this system is expected to operate for the majority of the time in the state whereby the converter is idling with minimal switching losses, we can assume that the all of the power flowing through the link inductors will be supplied to the load in this mode of operation. If we then assume the load to be resistive, the power consumed by the load can be defined as in Eqn.23.

$$P_{out} = \frac{V_{inva-b}^2}{R} \tag{23}$$

and since the power flowing through the link inductors is equal to the power being consumed by the load we can equate Eqn.22 with Eqn.23 to give:

$$3\frac{V_{in}V_{out}}{X}\sin(\delta) = \frac{V_{inva-b}^2}{R}$$

where by substituting in :

$$V_{inva-b} = \sqrt{3}V_{inv}$$

and

$$V_{inv} = V_{out}$$

we get:

$$V_{inv} = \frac{V_{in}R}{X}\sin(\delta) \tag{24}$$

Next from the definition of the modulation index for a space vector, Eqn.13 we can write:

$$V_{inv} = \frac{V_1}{\sqrt{2}} = M\frac{V_{dc}}{2\sqrt{2}}$$

and substituting for $V_{inv}$ from Eqn.24, we obtain Eqn.25 which relates the DC-Bus voltage to the output load and link inductor size.

$$V_{dc} = \frac{2\sqrt{2}V_{in}R}{MX}\sin(\delta) \tag{25}$$

Assuming that the modulation index is unity and the input voltage is constant, this relationship can be represented as a surface as shown in Fig.5-2:

University of Cape Town

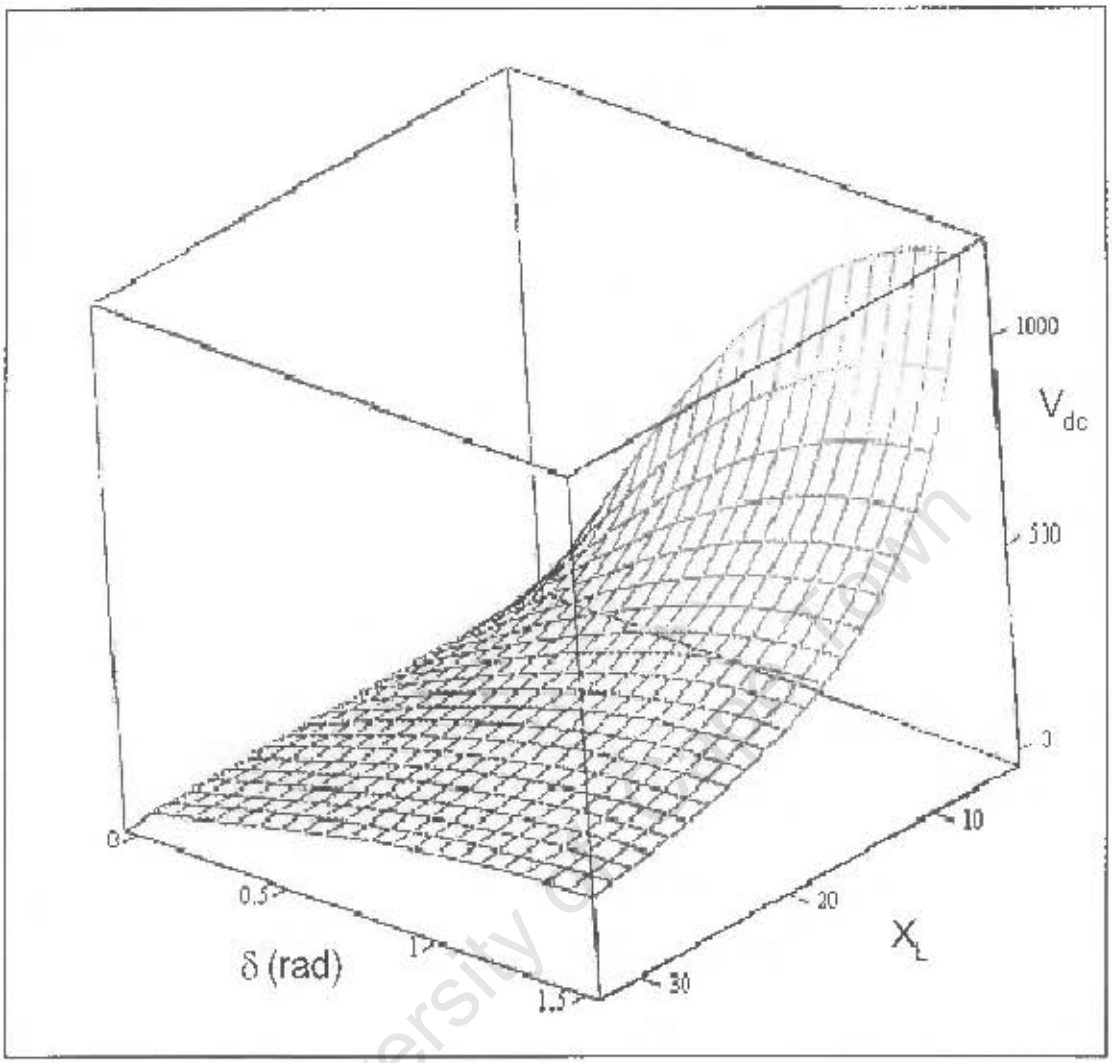



Figure 5-2.Relationship between DC-Bus Voltage, Inductor Size and Phase Angle

In order to be able to control the DC-Bus voltage easily, it is desireable to have a relatively large change in phase angle to cause a relatively small change in DC-Bus voltage. From Fig.5-2 it can be seen that to achieve this we need to choose an apropriately large inductor inductor size and operate over the full phase angle range, or select a smaller inductor size and operate over a limited phase angle range at close to $1.57rad$ $(90°)$.

**NOTE:**In the case of a battery DC-Bus, the DC-Bus Voltage has to be controlled during battery charching so as to limit the current.

University of Cape Town

## 2.3 Input Power / Power Factor relationship vs. Inductor size

Another important objective of this project is to try ensure that a close to unity power factor is obtained for the majority of the operation time from the supply point of view. In order to be able to achieve this it is necesary to derive a formula that relates the input power factor to the link inductor size and the input power. We can obtain this equation by substituting for $\delta$ from the standard power equation Eqn.7 into Eqn.8 to obtain Eqn.26.

$$pf = \frac{P_{out}X}{V_{in}\sqrt{(V_{out} - V_{in}\cos(\arcsin(\frac{P_{out}X}{V_{in}V_{out}})))^2 + (\frac{P_{out}X}{V_{out}})^2}} \tag{26}$$

From this we can plot a three dimensional surface for the input power factor during normal UPS operation, as in Fig.5-3. We can see from this plot that to be able to maintain a high power factor, either the inductor size must be kept small or the value of power being transferred must be kept small. Since keeping the power transfer small defeats the purpose of this project, it is preferrable to keep the value of the link inductors as small as is feasibly possible while trying to satisfy the other important constraints.

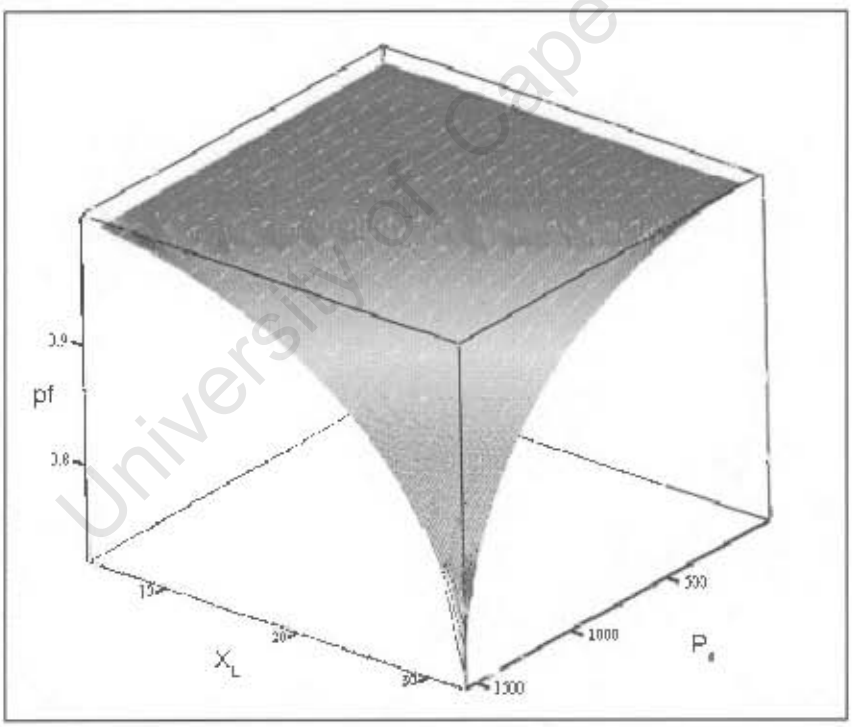



Figure 5-3.Relationship between Power Factor, Inductor size and Input Power flow

University of Cape Town

# 3 Control of Power Factor

Although it is possible to control the input powerfactor of the system as described in Chap.3 by controlling the phase angle $\delta$, by doing this both the DC-Bus Voltage and Input power flow also change. As can be seen from Fig.5-1 ,Fig.5-2 and Fig.5-3, as $\delta$ is increased for a given link inductor size, the DC-Bus voltage and input power increase as the power factor decreases. This demonstrates that if we want to regulate the DC-Bus Voltage by varying the phase angle, we will not be able to control the power factor of the system independently.

It is however possible by careful design to try and limit the the range of the attainable power factor during normal operation, by selecting an appropriate inductor size and phase angle range dependent on the maximum power of the system.

Assuming that the magnitude of $V_{in}$ and $V_{out}$ are equal in magnitude, by considering Fig.5-4 and using simple trigonometry, it can be seen that the input power factor angle $\theta$ will be equal to $\frac{\delta}{2}$. Thus by limiting the the maximum phase shift $\delta_{\max}$, it will be possible to limit the minimum input power factor to $\cos(\frac{\delta_{\max}}{2})$.

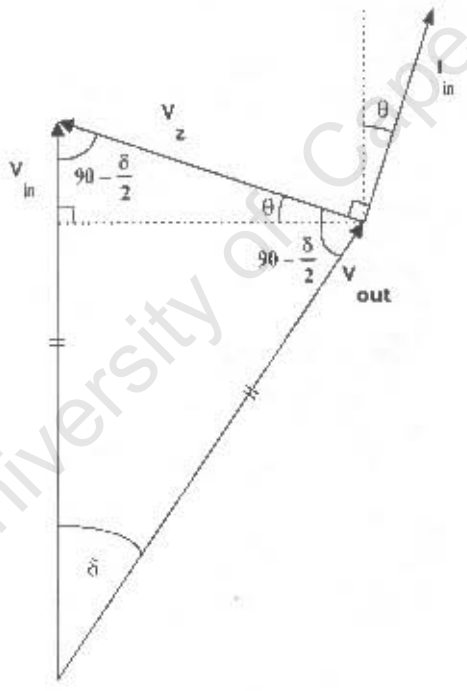



Figure 5-4.Determining Power Factor Limits

This is however heavily dependent on the magnitudes of $V_{in}$ and $V_{out}$ , as a change in their magnitudes for a constant phase angle will affect the power factor as shown in Fig.5-5. In

University of Cape Town

this case as $V_{in}$ increases, $\theta$ increases and the input power factor decreases, likewise when $V_{in}$ decreases the power factor improves. Since $V_{out}$ is constant as it is a regulated output, and $V_{in}$ is not controllable as it depends on the utility mains it is not possible to directly control the effects of the difference in magnitude between the voltage vectors. The use of this strategy is thus only an attempt to limit the power factor and will be most effective during normal operating conditions.

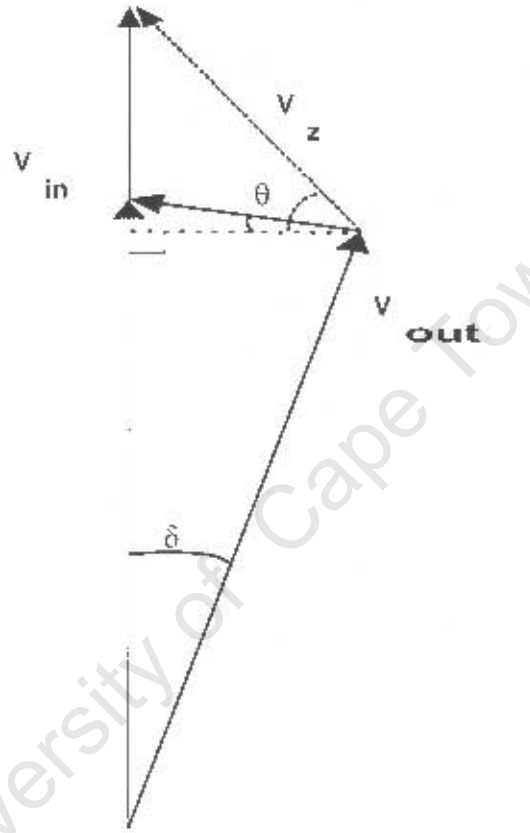



Figure 5-5.Change in Power Factor

University of Cape Town

# 4 Selection of Link Inductors

Since the power factor and DC-Bus Voltage cannot be controlled independently by varying the phase shift angle, it is important to select the impedance value of the link inductors $X$, such that the input power factor remains within acceptable limits. In the process of selecting the link inductor it is also necessary to take into account the above design considerations. The selection

of the link inductor was carried out as follows:

Firstly a maximum phase shift angle was selected to be a reasonably small value being 30°, so as to ensure the ease of control of the DC-Bus with the available phase angle resolution, and also to ensure that the input power factor remains close to unity power factor. Selecting 30° as the maximum phase shift angle results in a minimum power factor of $\cos(15°) = 0.966$, assuming that $V_{in}$ and $V_{out}$ remain equal in magnitude.

Secondly, knowing the the maximum load to be attached to the UPS and the incoming mains voltage, we can calculate a value of inductor from Eqn.7, as maximum power is transferred at the maximum phase shift angle. As the maximum power defined for the system is 15kW, which is distributed equally between the three phases, for a single phase we can calculate the inductor impedance as follows:

$P_{\max-1ph} = 5kW$
$V_{in} = V_{out} = 220V$
$\delta_{\max} = 30°$
$f = 50Hz$

$$X = \frac{V_{in}V_{out}}{P_{\max-1ph}}\sin(\delta_{\max}) = \frac{220^2}{5000}\sin(30) = 4.84\Omega$$

$$L = \frac{X}{2\pi f} = \frac{4.84}{2\pi(50)} = 15.4mH$$

For this project three inductors of $14.4mH$ were used as these were available off the shelf, and allow for a slightly higher system power rating.

## 5 Selection of DC-Bus Voltage

In order for the system to draw sinusoidal currents from the supply it is necessary to ensure that the DC-Bus Voltage is kept above a minimum voltage, by having a large enough battery set. The equation giving this relationship is Eqn.27[1] :

$$V_{dc} > 1.634V_{LL} \tag{27}$$

## 6 Phase locking

Phase locking of the inverter output to the utility mains, has been conventionally implemented by analogue techniques and through the use of analogue/digital phase locked loops. To achieve phase locking using this method it is necessary to determine one or both voltage zero point crossings for one phase, generate a pulse at each of these points, and then use the pulse as a reference for an analogue or digital phase locked loop. This results in a situation where the frequency and position of the voltage space vector can only be determined once or twice per

University of Cape Town

cycle, at these zero point crossing.

Several problems are associated with this method and can be summarised as follows:

- not possible to determine the exact zerocrossing as there is usually noise on the waveform
- not possible to follow reference waveform closely as frequency is updated only on zero crossings
- there is a maximum of two reference points per cycle.

## 6.1 Phase Locking Strategy

The phase locking strategy employed by this project overcomes some or all of these problems, due to it being a more robust method of locking. Since the Space Vector Modulation technique has been chosen for this Uninterruptible Power Supply, due to it being an inherently digital form of Modulation and due to it having a higher converter utilisation for the the same DC-Bus Voltage, it was realised that the phase locking technique could also be based around this modulation technique.

As described in Chapter4, in order to be able to generate a desired three phase voltage Space Vector output, it is necessary to know both the magnitude and position of the reference space vector. By using this already available information, a more robust phase locking technique can be implemented. Assuming that the utility mains consists of three voltage vectors spacially separated by 120°, it is then possible to mathematically sum these vectors as shown in Fig.5-2, resulting in a single vector that contains information about the magnitude and phase of all three phases.

The resulting mains voltage space vector can then be compared to the reference space vector used in generating the inverter space vector output. The difference in phase found between these two vectors can then be used to increase or decrease the converters output fundamental voltage frequency, so as to reduce the phase difference until phase locking results. A point to note about this method is that at any point within the utility mains cycle, the position and magnitude of the mains space vector can be determined, and can be compared with the reference space vector $v_s$ used in the generation of the converters Space Vector Modulation Output, allowing for more accurate phase tracking.

## 6.2 Implementation of phase locking

To be able to phase lock in the way described above, it is first necessary to sum the mains voltage vectors together to determine the position and magnitude of the utility mains voltage space vector. There are several methods in which this can be done depending on whether hardware cost or processing power is a determining factor. Following are several of the methods investigated:

- Software Implementation utilising three mains voltage vectors directly
  The three scaled mains voltage vectors are sampled by the processor. Scaling of the voltage signals is carried out using three signal transformers which also act as voltage isolators. The



University of Cape Town

processor then adds these three vectors together, and determines the magnitude and phase angle of the resulting space vector. It must be noted that this is the most processor intensive of the methods shown here, since this method requires several mathematically intensive steps.

- Hardware Implementation using two signal transformers
  A more elegant solution to the problem was found to be a hardware implementation of the vector summing as shown in Fig.5-6[2] . This method has the advantage that only two isolation transformers are required for the interface circuitry instead of three, and since there are only two signals to process this also reduces the processing required by the DSP.
  The three input phase voltages are connected as shown in the diagram to two signal transformers, the output of these transformers result in two sinusoidal waveforms displaced by 60°. To be able to easily determine the position of the space vector from these waveforms it is necessary to shift one of these waveform by 30° so as to have two sinusoidal waveforms $u_x$,$u_y$ at 90° to eachother. This is achieved by summing the two signals $u_A$,$u_B$ through an operational amplifier with resistor weightings as shown in the diagram.
  The output signals $u_x$,$u_y$ are then fed into the DSP, and since they are in quadrature it is a much simpler process for the DSP to calculate both the magnitude and phase of the resulting space vector. It should be noted that the sampled voltages in this configuration are line to line voltages, and thus the resulting space vector will be phase shifted from the line to neutral space vector generated by the inverter by 30°.

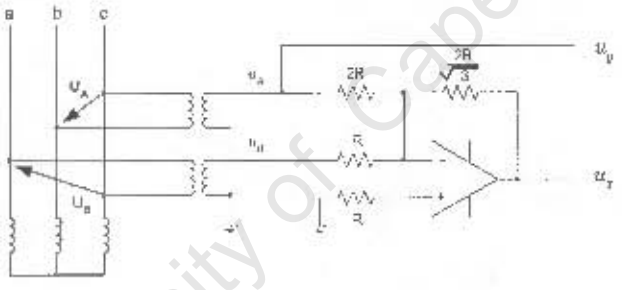



Figure 5-6.Hardware to Phase Shift Reference Waveforms

- Software Implementation using two signal transformers
  This solution is the same as the hardware implementation, however the two signals from the transformers are fed directly into the ADC of the DSP, eliminating the need for an external operational amplifier and thus provides a more cost effective solution. This however places more processing load on the DSP, as the conversion from two phases at 60° to two phases at 90° has to be implemented in software. A set of equations that produce this phase shift without unecessarily loading the DSP are Eqn.28 and Eqn.29[2] , shown below:

$$u_x = u_A \tag{28}$$

$$u_y = -\frac{1}{\sqrt{3}}u_A + \frac{2}{\sqrt{3}}u_B \tag{29}$$

University of Cape Town

# 7 Maintaining Constant Output Voltage

During all modes of operation it is necessary to maintain a constant voltage output on the load. However during battery charging and discharging the DC-Bus voltage will vary which will affect the output voltage magnitude. In order to maintain a constant voltage output it is thus necessary to have a control feedback loop whereby the magnitude of the output phasor is monitored and used to control the modulation index $M$. Voltage space phasor theory simplifies this control by being able to easily monitor the output voltage phasor constantly and keeping its magnitude fixed.

# 8 Battery Charging/Discharging mode

During normal operation the DC-Bus Voltage is maintained at a fixed value by controlling the phase shift angle. This strategy can only be used while the battery set is fully charged and the mains supply is normal due to the other system constraints. Thus different strategies are needed during the other operating modes:

## 8.1 Battery Discharging

On a mains power failure the UPS is disconnected from the mains supply to prevent power flowing from the inverter back into the supply, and power to the load is supplied from the batteries causing them to discharge. As power is no longer flowing from the supply it is also not possible to control the DC-Bus Voltage level using the phase shift strategy and consequently the output voltage will vary if there is no feedback control as described above. Since in this mode of operation there is no reference phasor on which to phase lock onto, it is necessary to keep the fundamental frequency constant during this mode until the mains supply returns to normal.

## 8.2 Battery Charging

On the return of the mains supply, phase locking must occur before the UPS is reconnected to the supply to avoid large transient currents. As the batteries will be invariably depleted during this operating mode, it will not be possible to control the DC-Bus voltage to the desired value without exceeding the desired battery charging rate. It is thus necessary to modify the control strategy during this mode such that the phase angle is current limited until the batteries return to the desired voltage level.

University of Cape Town

# Chapter 6

# System Hardware Description

## 1 Introduction

This section gives a brief introduction to the hardware used to realise the three to single phase Uninterruptible Power Supply. A brief description of the MSK243 Digital Signal Processing (DSP) evaluation board and development software is given, with the features that lead to the selection of this development kit for the project. Following this is a description of the important DSP features that were used in the project, with the interface boards built for sampling signals and driving the converter. For more detailed information regarding the TMS320F243 DSP see references [4] , [5] .

## 2 Choice of DSP Development Kit

The DSP development kit chosen for this project was the MSK243 Pro Kit from Technosoft S.A. Switzerland. This specific kit was chosen due to its superior Integrated Development Environment (IDE), whereby programming can be carried out in both ANSI-C and assembler, and has advanced graphical debugging tools which greatly reduce development time.

The development kit is based on the Texas Instruments TMS320F243 motor control DSP, and comes with the most advanced programming and debugging software for motor control DSPs currently available on the market.

### 2.1 Technosoft DSP Development Board Key Features

The hardware part to this development kit is the DSP development board, a picture of which is shown Fig.6-1.

University of Cape Town

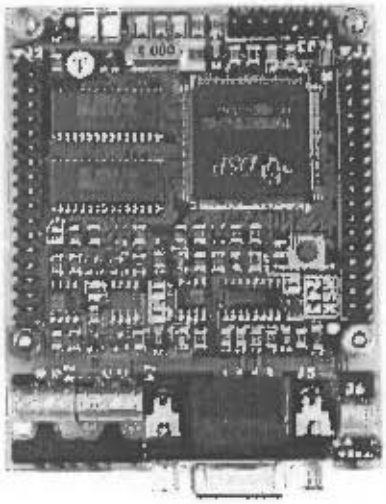


Figure 6-1.Technosoft MSK243 DSP Development Board

The main features of this DSP evaluation board are as follows:

- Texas Instruments TMS320F243 motor control DSP
  This is a 16 bit fixed point DSP running at 20MHz designed specifically for digital motor control. It has 8kword of flash, 4kword of which is occupied by the Technosoft monitor program. This program provides the interface for programming and debugging through the serial communication interface, using the Technosoft development environment. The DSP also has 544 words of internal data RAM and specific peripherals necessary for motor control.
- 32-kword 0-wait state SRAM external data/program memory
  The function of this memory is specifically for system development, it is used as both program execution memory and for storage of trace variables during program debugging. Once a system has been fully developed this memory would normally no longer be used, and the program would be burnt onto the 8kword of flash memory for production purposes.
- RS-232 serial communication port
  All communication with the evaluation board is carried out through this port, both for the programming and debugging.
- CAN field bus communication port
  This is used for inter-processor communication, it is however not used in this project.
- Standard I/O headers
  These headers allow for direct connection of signals to the more important pins of the DSP, they do however require external interface circuitry. Not all of the available DSP output pins are accesible from these headers.
- Jtag interface (XDS510 emulator connector)
  The Jtag interface allows for real time emulation of the target system, through the use of a Jtag pod cable and DOS based emulation software. This system was not used because the

University of Cape Town

Technosoft debugging software was found to be superior to the debugging tools available for this interface.

- Power Supply (5V ±10%)

## 2.2 Technosoft Development Environment Key Features

Software is supplied with the kit specifically for programming the DSP and for debugging purposes. The software supplied with the kit was the DMC Developer Pro version of the software, which is a fully integrated DSP software development environment. A screen capture of this software is shown in Fig.6-2, showing the powerful graphical debugging capabilities of the software.

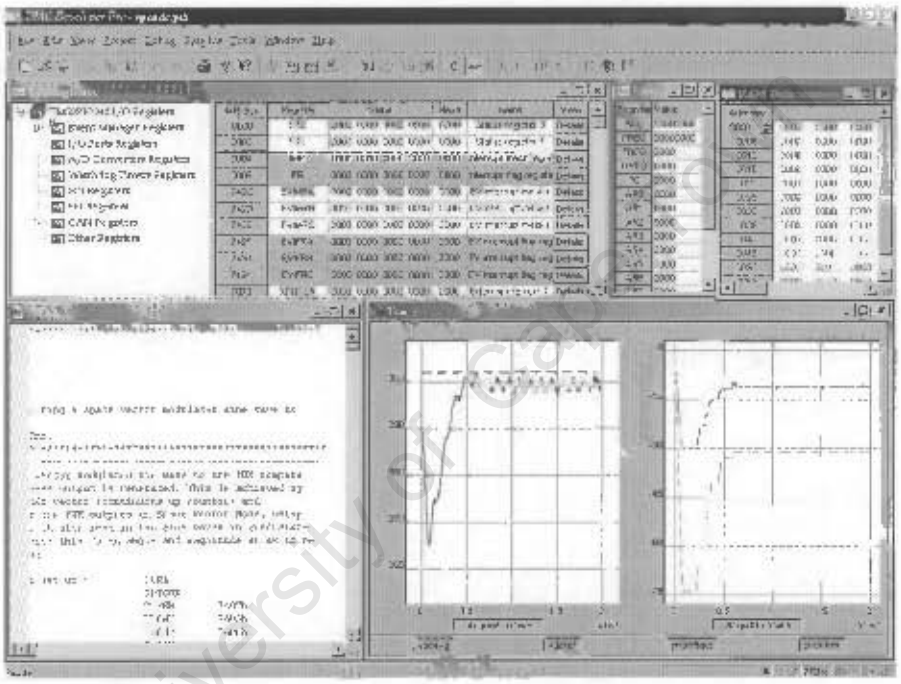


Figure 6-2.Screen Capture of DMCD-Pro Software

University of Calgary

The main features of this environment are:

- Fully integrated DSP software development environment
  The easy to use graphical environment ensures a relatively short learning curve to get started.
- Project management system
  An integrated file management system provides an effective way of quickly visualising, accessing and manipulating all project files and their dependencies, which allows for more a more efficient development process.
- Programming of target system in both ANSI-C and Assembler
  The technosoft development environment is fully compatible with the Texas Instruments C-

Compiler and Assembly tools. It is possible to program the target system in either ANSI-C or assembler, however programming of the target system using C promotes a more efficient development process by greatly reducing the development time.

- Incorporated debugger
  It is possible to store program variables in external memory during program execution for later viewing and analysis. This allows the user to quickly observe system variables during the debugging process. Debugging can also be carried out using standard debugging techniques such as single stepping, break points and watch windows.
- Tracing Module
  System variables stored during program execution in the debugging process, can be uploaded to a PC after execution has completed, and then visualised in graphical form, as shown at the bottom right of Fig.6-2.

# 3 Features of the TMS320F243 DSP

The TMS320F243 device is a member of the TMS320 family of digital signal processors which have been designed to meet a wide range of digital motor control and other embedded control applications. This DSP combines the enhanced TMS320 architectural design of the C2xx core CPU and several advanced peripherals optimised for motor control applications. This DSP has an external memory interface and is primarily intended for development purposes, it has the following main features:

- 16-bit Fixed-Point DSP C2xx Core CPU (enhanced Harvard architecture)
- Operates at 20 MIPS, 50-ns Instruction Cycle Time
- 8K words x 16bits of Onchip Flash ROM
- 544x16words of Dual Access RAM (DARAM)
- External Memory Interface - 224K Words x 16 Bits of Total Memory Address Reach
- Multiple Peripherals Associated with Motor Control
- 26 Individually Programmable, Multiplexed General-Purpose I/O (GPIO) Pins
- Six Dedicated GPIO Pins
- Five External Interrupts (Power Drive Protection, Reset, NMI, and Two Maskable Interrupts)
- Scan-Based Emulation
- Three Power-Down Modes for Low-Power Operation
- 5V operation

An important part of the DSP is its internal peripheral set which includes devices such as:

- Event Manager : Timers and PWM generators for digital motor control
- CAN Interface: Controller Area Network
- ADC: 10bit pseudo-dual Analogue to Digital Converter (ADC) Module, 1.7$\mu$s dual conversion time
- SPI: Serial Peripheral Interface
- SCI: Serial Communication Interface

University of Cape Town

- Watchdog timer
- General Purpose bi-directional digital I/O pins

For more information regarding the TMS320F243 and its specific peripherals, refer to [5] . Following is a more detailed description of the peripherals used during this project and how they were setup:

## 3.1 Event Manager (EV2)

The EV2 provides a broad range of functions and features that are particularly useful in motor control applications. A block diagram of the Event Manager can be seen in Fig.6-3[5] . In this project the EV2 has been primarily used for the generation of the PWM switching signals. Since most of the EV2 device pins are shared with general purpose digital I/O pins and thus need to be setup appropriately. Only the sections of the EV2 used during this project will be described further.

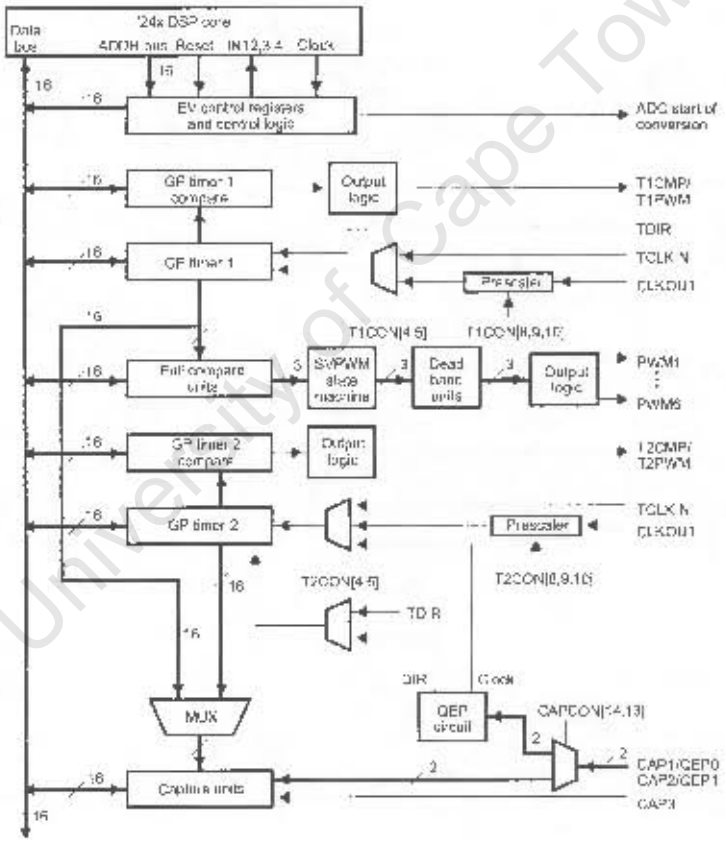



Figure 6-3.Event Manager (EV2) Block Diagram

University of Cape Town

### 3.1.1 Timer1and Peripherals dependent on Timer1

The functioning of the entire UPS program is entirely dependent on Timer1 and the peripherals dependent on it. Following is a brief overview of Timer1 and its dependent peripherals functionality, relevant to the project.

Timer1 is a highly configurable 16bit timer, suitable for the motor control. The timer has several modes of operation dependent on the type of PWM required, ie Continuous Up-Counting mode, Directional Up/Down-Counting mode and Continuous Up/Down-Counting mode. The mode selected for this project is the Continious up/down-counting mode which is particularly useful in the generation of symmetrical PWM waveforms, and which has significant harmonic advantages over the other counting modes. Fig.6-4[5] shows the basic operation of Timer1 in the continuous up/down-counting mode.

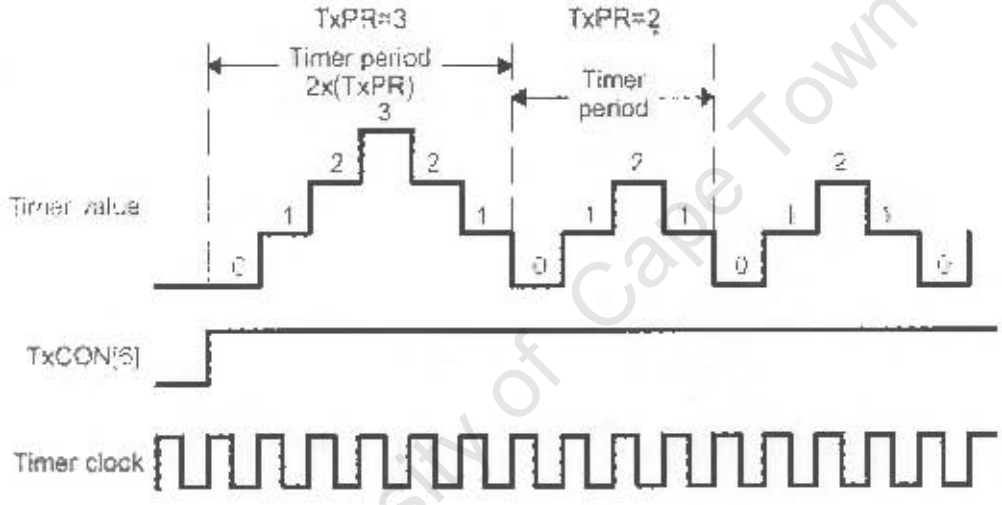



Figure 6-4.Timer1 Continuous Up/Down-Counting Mode

The functions that are dependent on the frequency of Timer1 are as follows:

- Hardware Interrupts
  Harware interrupts are highly configurable and can be setup to occur on several Event Manager events. These interrupts are organised into three groups, each being associated with a different interrupt flag and interrupt enable register. In each group there are several Event Manager peripheral requests, each group having its own priority and each peripheral request having its own priority within the group. For a more detailed listing of these groups and their priorities refer to [5] .
  The way in which these interrupts are handled is as follows: when a peripheral interrupt request is acknowledged, the appropriate peripheral interrupt vector is loaded into the peripheral interrupt vector register(PIVR) by the Peripheral Interrupt Expansion (PIE) controller.

University of Cape Town

The vector loaded into the PIVR is the vector of highest pending priority and determines the starting point of the next section of code to be run.
In this program Timer1 has been set to generate an event manager event and thus cause an interrupt on a Timer1 Underflow and on a Timer 1 Period, and consequently execute the appropriate interrupt service routine.

- ADC start of conversion

The TMS320F243 has a pseudo-dual 10bit ADC unit capable of a dual conversion every 1700ns, as represented in Fig.6-5[5] .

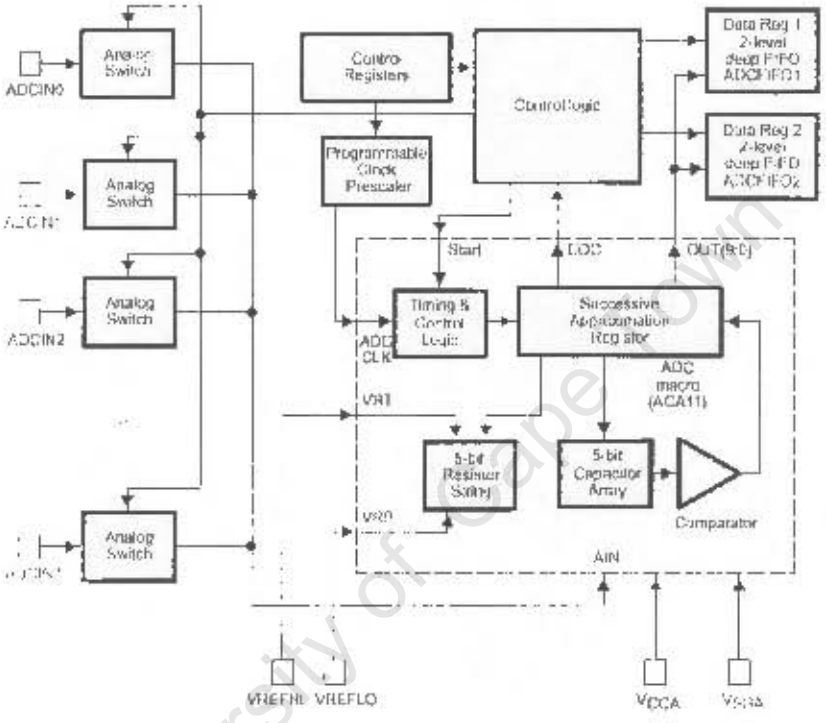



Figure 6-5.Pseudo-Dual ADC Module

As can be seen the ADC has an eight channel input with a single converter that has only one inherent sample and hold circuit. The configuration of the ADC has been designed such that it behaves as if there were two analogue converters. Although there are eight separate input channels, only two of these can be sampled at the same instant (850ns apart), and the values are then stored in the two level deep ADCFIFO1 and ADCFIFO2 registers.
Thus to capture a sample the two channels to be sampled must first be selected, and then a conversion can be started by either setting the appropriate register for an immediate conversion, or on an event manager signal in this program on a Timer1 Underflow. This ensures a regular sampling time, dependent on the Timer1 frequency. The results of the conversion can then be read from the FIFO registers once the conversion is complete.

University of Cape Town

- PWM Generation
  The TMS320F243 DSP has a built-in three phase PWM peripheral generation unit, which must be configured by setting the appropriate configuration registers according to the type of PWM required. The PWM generation unit forms part of the Event Manager as can be seen from the Event Manager Block Diagram in Fig.6-3, and is capable of producing PWM output without placing any load on the processor. All that is required from the processor is to supply the PWM generation unit with the switching period and the relevant compare values. In this project symmetrical PWM is used, the generation of which can be more easily explained using Fig.6-6[5] . The PWM output is generated by the PWM unit using the standard technique of comparing a triangular switching waveform with a reference waveform. The triangular waveform determines the PWM switching frequency and is generated using one of the timers. Dependent on the timer counting mode two types of PWM are possible, namely symmetric and asymmetric PWM. Comparison with the reference wave form is carried out by placing the instantaneous value of the reference wavform into the compare registers associated with the PWM unit, so as to generate complimentary PWM switching waveforms as shown in Fig.6-6

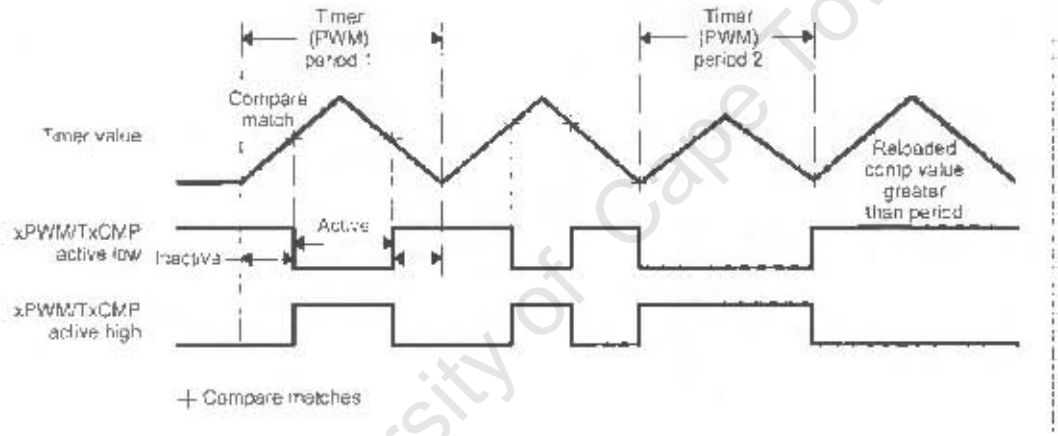



Figure 6-6.Generation of Symmetrical PWM Output

University of Cape Town

# 4 Interface Boards

To be able to connect external signals to the DSP board, and for the DSP board to drive the high power converter, it is necessary to have interfacing circuitry for the protection of the DSP and for signal conditioning. Following is a description of the interface boards that were used for the project.

## 4.1 Isolation Transformer Board

For implementing phase locking as described in Chapter5 and for control purposes it is necessary to sample the mains voltages to determine the phase and magnitude of the mains space vector. Since the DSP operates on a 0-5VDC range and the voltages to be sampled are in the 0-380VAC range, and also from the circuit protection point of view, it is necessary to isolate the two voltage levels from eachother. The simplest and most effective way of achieving this is by using isolation signal transformers, so as to completely separate the two voltage levels. In this project, two transformers have been used for the purpose of sampling the input mains and are connected to the three phase mains as shown in Fig.6-6. The physical implementation of this is shown in Fig.6-7, where the mains voltages enter at the bottom of the figure and the isolated signals exit at the top. The transformers shown in the figure are 380VAC to 15VAC signal transformers.

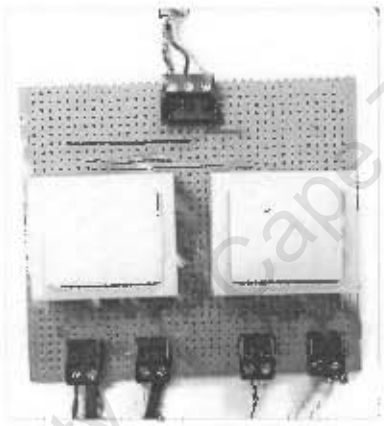


Figure 6-7.Signal Transformers for sampling Mains

## 4.2 Mains voltage Signal Conditioning board

In order for the DSP to read system parameters, it is necessary to condition the signals appropriately to limit the voltage range supplied to the DSP for protection of the analogue to digital converter. The system parameters that need to be read into the DSP to be able to control the system are the three mains voltages and the DC-Bus Voltage.

The first strategy used for this board was the hardware implementation of the phase locking, using an operational amplifier as shown in Fig.6-6. On this board the output signals were then further conditioned by adding a 2.5VDC offset to signals $u_x$ and $u_y$ before supplying the signals to the ADC unit of the DSP. This hardware configuration was however found not to work adequately, as the operational amplifier is required to operate at close to unity gain and under



University of Cape Town

these conditions the majority of operational amplifiers are unstable and introduce noise into the output signal. To try and circumvent this problem an operational amplifier stable at unity gain (MAX474) was used, this dramatically improved the signals to be sampled however still introduced substantial noise.

Finally the hardware implementation of the phase locking was abandoned, due this problem and a fully software solution was implemented, with interface circuitry as shown in Fig.6-8. This board consists of resistor divider circuits to reduce the signal transformer ouputs to a -2.5V to 2.5V range, the operational amplifier then adds a 2.5VDC offset so that the range is in the 0V to 5V range suitable for the ADC unit of the DSP. The board also contains clamping schottky diodes, to ensure that the signals supplied to the ADC do not exceed the voltage rails by more than 0.3V, thus protecting the ADC unit from damage (-0.3V to 7V range).

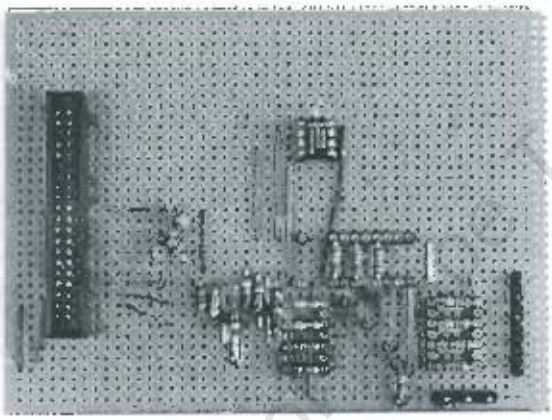


Figure 6-8.Input Signals Conditioning Board

This board also has an input for the DC-Bus voltage that has been scaled down to the 0 to 10V range by a differential amplifier board. This signal is resistor divided to the 0 to 5V range, and then clamped with schottky diodes to ensure protection of the ADC unit.

## 4.3 Differential Amplifier Board

In order to safely sample the DC-Bus, it is necessary to isolate the DSP from the high voltage DC-Bus and convert the voltage to the 0 to 5V range necessary for the ADC unit. To achieve this a prebuilt differential amplifier board was used whereby extremely high value resistors were used on the input to the operational amplifier such that the voltage of the DC-Bus is effectively dropped across these resistors and the gain of the system is below unity. The resulting output is in the 0 - 10V range which is supplied to the mains voltage signal conditioning board which then converts the signal to the 0 to 5V range and provides protection for the ADC unit. The phsical implementation of the board is shown in Fig.6-9, and the schematic shown in Fig.6-10.

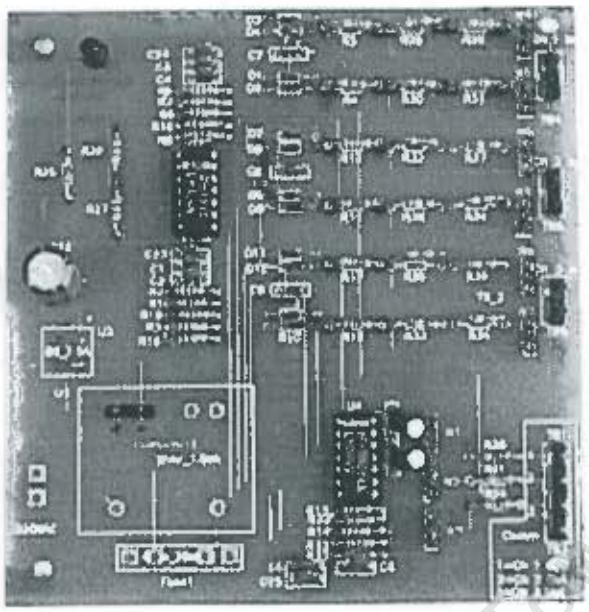


Figure 6-9.Differential Amplifier Board

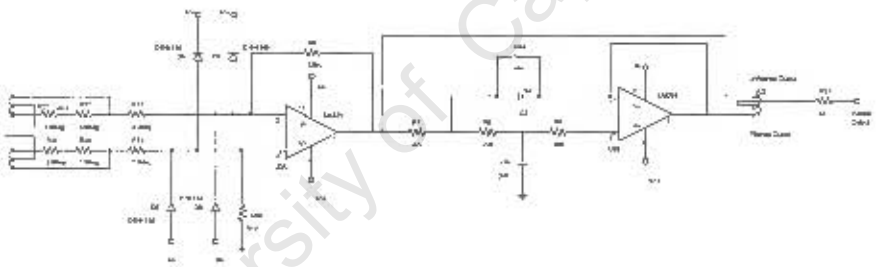


Figure 6-10.Differential Amplifier Schematic

University of Cape Town

## 4.4 Inverter Driver Boards

The UPS in this project was implemented using a three phase IGBT inverter, driven by a Semikron SKHI60 IGBT driver board shown in Fig.6-12. Since the semikron driver board requires 15V signals to drive the its inputs and the DSP operates in the 0 to 5VDC range an interface board was necessary between these boards. This board also acts as an extra protection for the DSP board by acting as a buffer. The interface board built is shown in Fig.6-11, and consists of two voltage level shifting chips (CD4504BE) which convert the signals from the DSP at 5V to the 15V range and provide an extra protection buffer. Also visible on this board is a simple

resistor ladder, connected to an output port of the DSP to implement a simple Digital to Analogue converter used for debugging purposes. The potentiometer seen on the board was also used for debugging purposes, for supplying a manually controllable 0 to 5V signal to the DSP ADC unit.

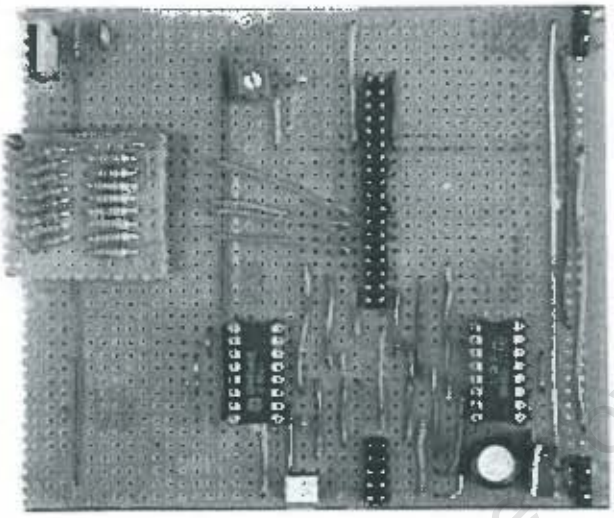


Figure 6-11.PWM Voltage Level Shifting Board

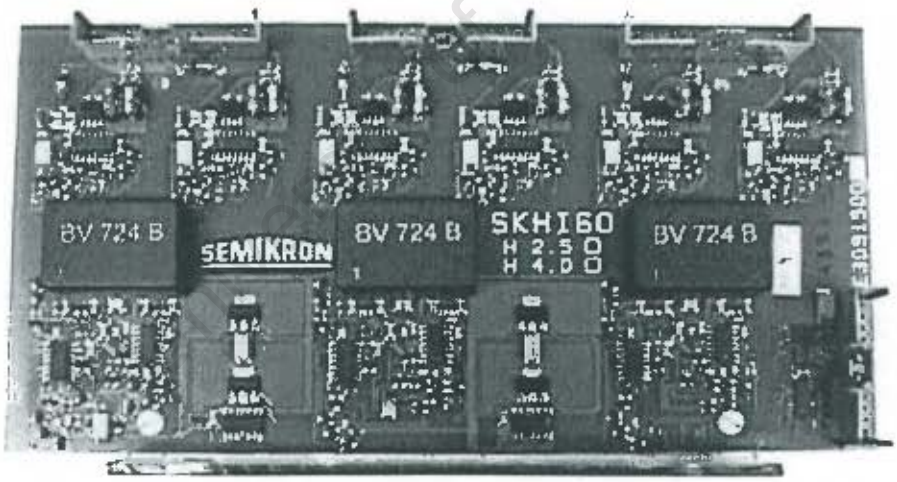



Figure 6-12.Semikron SKHI60 IGBT Driver Board

# 5 Inverter IGBT Module

Fig.6-13 shows the photo of the implemented IGBT inverter, connected to the UPS system. This module was provided by Semikron. The photo shows the inverter module which comprises of the inverter driver board, the DC-Link capacitors, three phase IGBT modules and a cooling fan and heat sink.

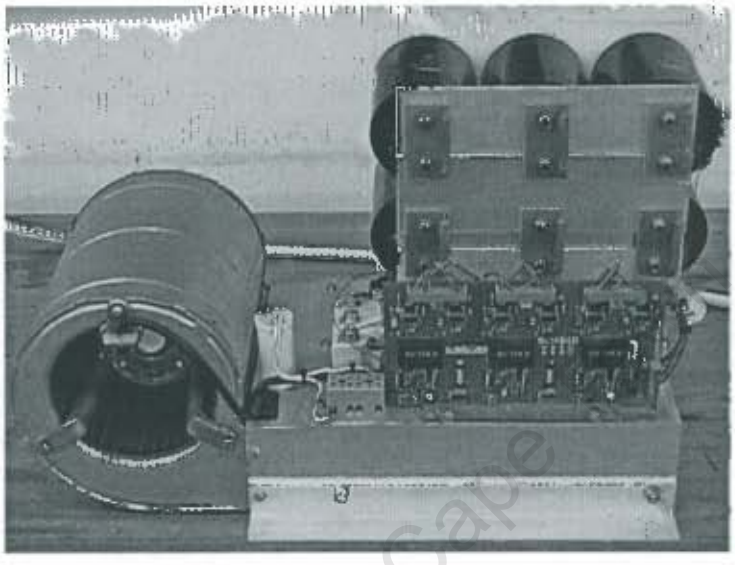


Figure 6-13.Semikron Three Phase IGBT Inverter Module

### 5.0.1 DC-Bus Battery Set

A set of fourteen 12V (7AH) batteries was connected to the DC-Link capacitors of the inverter, through a current limiting resistor and a 25A DC Breaker for protection during testing. A photo of the battery set and breaker is shown in Fig.6-14.

University of Cape

Figure 6-14.UPS Battery Set

University of Cape Town

# Chapter 7
# Software Description

## 1 Introduction

This chapter describes the development of the software for the TMS320F243 DSP. Firstly there is a description of the overall program structure, and following is a more detailed explanation of each routine.

## 2 Limitations

Since the DSP chosen for this project is of 16bit fixed point architecture running at 20MHz, and the programming carried out in the C-Programming language, it was necessary to be very careful on the choice of C commands and structures used. It was found necessary to write the bulk of the program using fixed point routines, making the writing of the program much more challenging.

## 3 Program Structure

Due to the limitations mentioned above it was necessary to structure the program carefully, such that the amount of calculations in a switching cycle is minimised. Due to the nature of the phase locking implemented, it is necessary to know the angular positions of both the mains phasor and inverter phasor at any one instant. Since the calculation of any one of these angles requires substantial processing, a great time saving can be achieved if it is not necessary to calculate one of these angles. By carefully selecting the structure of the program it is possible to achieve this, and also keep the harmonics generated to a minimum.

Fig.7-1 shows one converter switching cycle, around which the entire program structure is based. Timer1 is set up as an up down counter, and its period is selected such that exactly 120 switching cycles occur within one fundamental converter output voltage cycle.Using this strategy it is not necessary to perform complex calculations to determine the inverter output voltage phasor angle for each switching cycle. It is inherent that for every switching cycle, the inverter phasor angle will advance by exactly $3^o$. This reduces the program execution time significantly by:

- being able to calculate the exact inverter output phasor angle for each switching cycle, within one instruction cycle
- keeping the inverter output phasor angle always an integer, such that all calculations associ-

University of Cape Town

ated with the variable are optimised for the DSP architecture

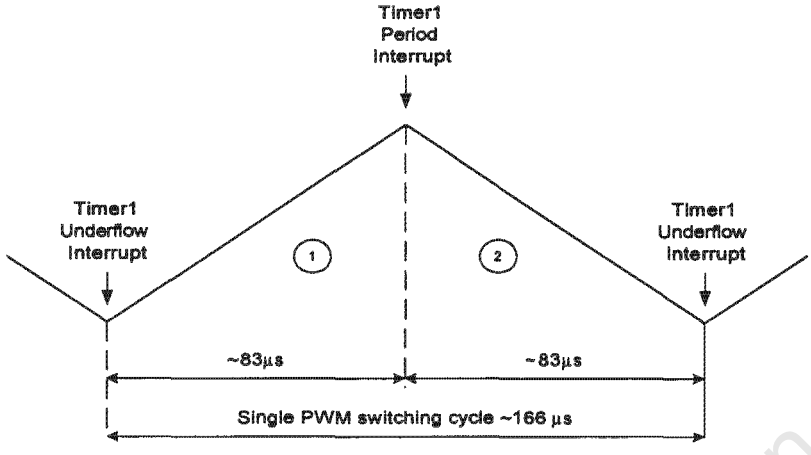



Figure 7-1.Switching Timing Diagram

A further advantage of this strategy is that it ensures that the type of PWM generated is synchronous PWM, which results in less subharmonics (of the fundamental frequency), than asynchronous PWM [1] .

To further improve the optimisation of the code, it was found necessary to separate critical code from non critical code, and to give critical code a higher priority to ensure its execution. To achieve this, each switching cycle has been subdivided into two time intervals by defining two interrupts, as shown in Fig.7-1. The interrupts have been defined such that one occurs on Timer1 reaching a value of zero (Underflow Interrupt) and a second occurs on Timer1 reaching the value defined in its period register (Period interrupt).

Critical functions that require to be executed every switching cycle, such as the calculation of switching times, have been setup to execute within either of the two interrupt service routines. This ensures that these functions are of a high priority and will be executed at fixed intervals, and requires that the code execution time be shorter than the time for half a switching cycle. All other functions of lesser priority, can be setup to run within the main loop of the program, and will be executed during the spare time between interrupt service routines.

# 4 Description of Program Routines

The program implemented on the DSP can be represented in flow chart form as in Fig.7-2. It has has been written in four distinct sections. With reference to the program code in Appendix A, the code has been separated into the following sections:

University of Cape Town

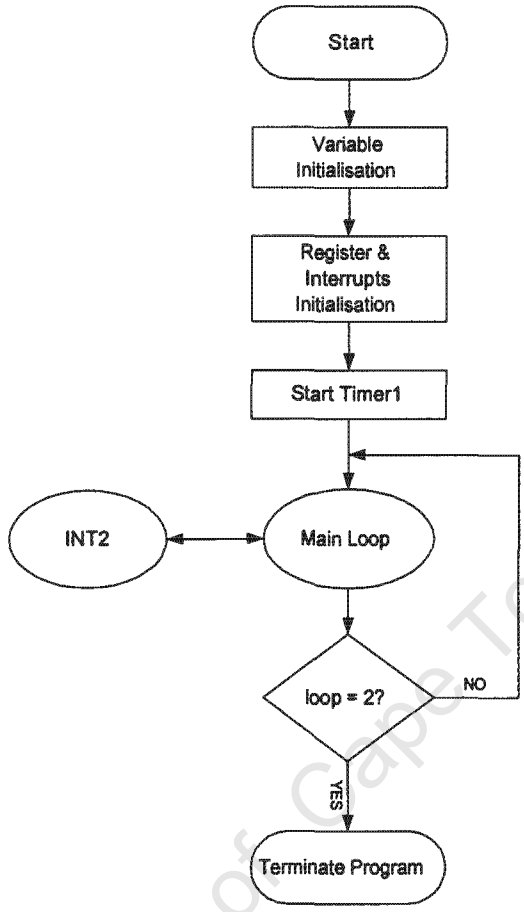



Figure 7-2.Program Flowchart

University of Cape Town

- *Dsp/Variable Initialisation*
  In this section all variables used in the program are initialised, and the DSP system and peripheral registers are setup as required for the main program.
- *Main Program loop*
  This section of code ensures, that the program has defined bounds. All routines of low priority, such as for the display of system parameters, can be placed within this section of code.
- *Interrupt Service Routines*
  Two interrupt service routines have been defined, code in this section has been setup with a high priority and is executed every switching cycle.
- External Functions and Lookup tables
  This section of code includes all external functions/ library files required for linking and compiling the C-program. It also includes lookup tables used in functions within the main program and interrupt service routines.

## 4.1 DSP/Variable Initialisation

This section of code defines and initialises all the necessary program variables, defines the external functions and lookup tables, and sets up the system/peripheral registers to enable the desired functions on the DSP. Following is a description of the setup of these registers, detailing which functions have been enabled:

### 4.1.1 Initialisation of Timer Registers

The way in which this program has been structured is such that it requires only one timer. This timers main function is to provide a time base for the PWM switching, and thus its period determines the switching frequency. The timer provides a convenient way of making sure certain code is executed at fixed intervals, and is also used to ensure that the sampling of system parameters using the ADC (Analogue to Digital Converter), occurs at fixed intervals. In order to setup timer1 for this program the following registers have been initialised with the following functions:

- GPTCON (General Purpose Timer Control Register)
  *ADC start of conversion on timer1 underflow.* This sets up the sampling of the mains voltages to occur at the beginning of each cycle, such that the time between samples is constant and inverter phasor angle and switching times can be determined for the next cycle.
- T1CON (Timer1 Control Register)
  *Directional up/down count mode.* This is required for the generation of symmetrical PWM.
  *Disable Timer Operations.* This is done so that the timer is started only when PWM output is required.
- T1PER (Timer1 Period Register)
  *This register determines the period for Timer1 and consequently the PWM switching frequency.*
- T1CNT (Timer1 Count Register)
  *This register needs to be initialised to zero to ensure timer1 starts counting from zero.*

### 4.1.2 Initialisation of Analogue to Digital Converters (ADC)

The analogue to digital converters are used to sample voltages and currents needed to control the system. The ADCs have been setup such that a dual conversion is started on a Event Manager signal. For this to occur the following registers are initialised as follows:

- ADCTRL1 (ADC Control Register 1)
  *Enable both pseudo-ADC#1 and pseudo-ADC#2.* This enables a dual conversion on a request for analog to digital conversion.
  *No continuous conversion.* This is required to ensure that a conversion only occurs on a request.
  *Channel Select.* Pseudo-ADC#1&ADC#2 are initialised to read from channel 4 and channel 6. This prepares for a sampling of the incoming mains voltages.
- ADCTRL2 (ADC Control Register 2)

University of Cape Town

*Enable ADCEVSOC.* This enables an ADC start of conversion on an event manager signal, that is on a Timer1 underflow and Timer1 period interrupt.

- ADCFIFO1&2
  *Read from ADCFIFO registers twice.* It is necessary to read from these registers twice to ensure that the two level deep ADCFIFO stacks are clear, and ready for a conversion to take place.

### 4.1.3 Initialisation of PWM registers

In order to be able to generate a symmetric PWM output, the registers associated with PWM need to be setup as follows:

- ACTR (Compare Action Control Register)
  *CMPxACT set active high.* This sets the six full compare PWM output pins in an active high mode of operation.
- CMPRx (Compare Registers 1-6)
  *Initialised to zero.* The compare registers used in PWM operation are all initialised to zero to ensure that there is no PWM output for the first cycle.
  (NOTE: The compare registers are shadowed and are reloaded at the beginning of each switching cycle, this ensures symmetrical PWM)
- DBTCON (Dead Band Timer Control Register)
  *Disable deadband.* This is due to the fact that dead band generation has been hardware enabled on the IGBT driver board. Especially during the development stages it is a safer option to hardwire the dead band generation, rather than setting it through software.
- COMCON (Compare Control Register)
  *Timer compare operations are enabled.*
  *Compare Register CMPRx reload on underflow.*The compare Register CMPRx reload condition has been set to be when the value for Timer1 is zero (that is, on underflow). This ensures that the PWM generated is always symmetrical (complimentary compares for an output pole occur at the same value).
  *Compare Output Enable.* The PWM output pins are set not to be in high-impedance state, that is they are enabled.

### 4.1.4 Initialisation of Interrupts

In this program use of interrupts has been made to ensure execution of critical code. The DSP program has been written such that there are only two interrupts that occur and that one will not interfere with the other. Both Timer1 period and underflow interrupts have been enabled, all other interrupts have been disabled.

- EVIMRA (Event Manager Interrupt Mask Register)
  *Timer1 interrupts enabled.* Timer1 Period and Underflow Interrupts have been enabled.

## 4.2 Main Program Loop

The main program loop is necessary to ensure that the program has defined bounds. Any func-

University of Cape Town

tions that are not critical and are not required on a real time basis, can be executed within this loop, such as the displaying of system parameters on an LCD display. As of yet no functions have been implemented within this loop, as it was out of the scope of this project to provide these.

## 4.3 Interrupt Service Routines

In this program two interrupts have been defined, both occuring on the same interrupt level INT2. These interrupts have been chosen such that they occur at a fixed time during the switching cycle and thus do not interfere with eachother. The length of code for each interrupt has been written such that its execution completes within half a switching cycle time-frame.

The way these interrupts are handled can better be understood by referring to the flow chart in Fig.7-3. On a CPU interrupt, execution of the main loop (which is a background operation) is stopped, and the interrupt level is determined. Program execution then branches to the appropriate interrupt level (INT2), where the type of interrupt is determined. A context save of important registers (such as the accumulator and status registers) is then carried out, and the program execution then branches to the corresponding interrupt service routine.

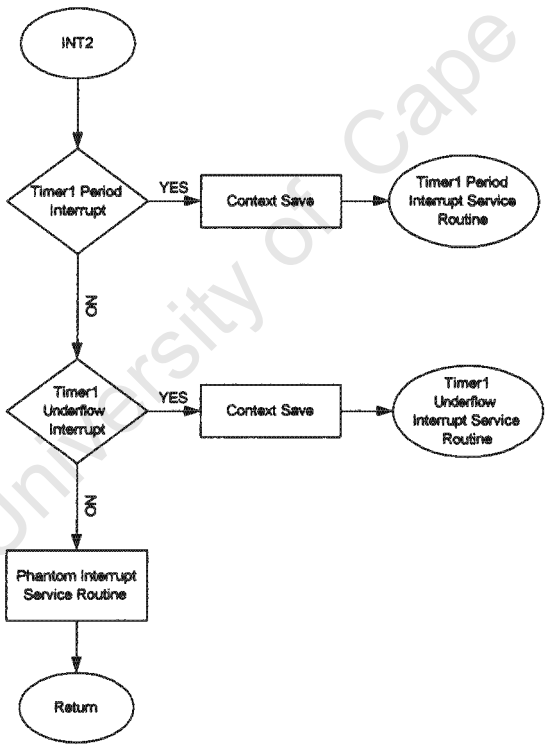


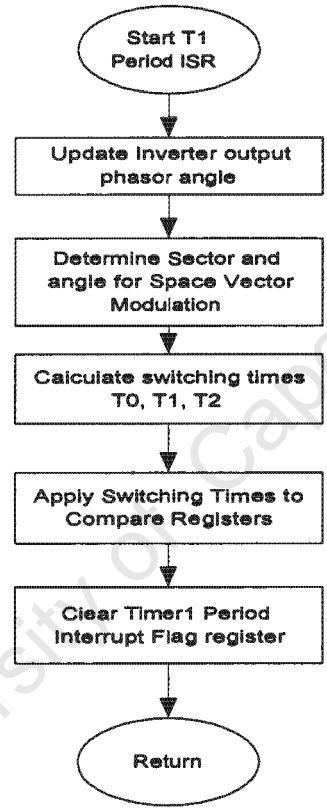

Figure 7-3.Interrupt Handling Flowchart

University of Cape Town

The two interrupt service routines are discussed further below:

### 4.3.1 Timer1 Period Interrupt Service Routine

When the value of Timer1 matches the value stored in its period register, an interrupt is generated and program execution branches to the Timer1 Period Interrupt Service Routine. This interrupt service routine has been written with the task of generating the Space Vector PWM switching times for the following switching cycle. The interrupt routine can be explained more clearly using the flow chart in Fig.7-4.




University of Cape Town

Figure 7-4.Timer1 Period Interrupt Flowchart

The interrupt service routine starts by determining the inverter output angle for the next switching cycle. The calculation of the output angle has been greatly simplified by the careful selection of the program structure, and thus all that is required is to add $3^o$ onto the previous inverter output angle. Next in order to be able to generate of Space Vector PWM it is necessary to determine in which sector and at what angle within the sector, the desired output is situated in. Once these values have been determined, it is then posible to calculate the PWM switching

times $T_0, T_1, T_2$, using standard Space Vector equations and a modified sine lookup table. These times are then written to the compare registers (shadowed), which are updated on a Timer1 underflow (at the beggining of the next switching cycle). This marks the end of the interrupt service routine, and program execution is returned to the main loop, after the enabling of the Timer1 period interrupt.

As much of this interrupt service routine has been written using integer variables, to ensure the fast execution times required to achieve the objectives of this project.

### 4.3.2 Timer1 Underflow Interrupt Service Routine

The function of this routine is to determine the DC Bus Voltage and the position of the mains input phasor, and from these values:

- Phase lock the inverter output to the input mains
- Control the charging of the battery set by sampling the dc bus voltage and using a PI controller to control the phase angle

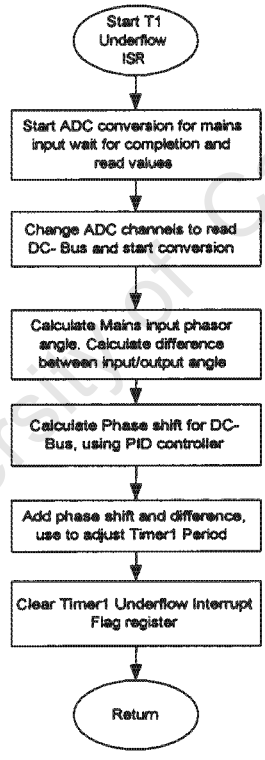



Figure 7-5.Timer1 Underflow Interrupt Flowchart

The Underflow Interrupt service routine is represented in flow diagram form in Fig.7-5. During this service routine, four separate signals need to be sampled using the analogue to digital

University of Cape Town

converter ADC. Due to the DSP architecture, only two conversions can take place simultaneously, and thus it is necessary to time multiplex the sampling of input signals.

Firstly the input mains is sampled, so that the position of the mains input phasor can be determined at the beggining of each switching cycle. Once the input mains samples have been read, the ADC channels are changed so as to be able to sample the DC-Bus. To maximise the use of processor time, while the conversion of the DC-Bus sampling is taking place ($1.7\mu s$), calculation of the input mains phasor position can be carried out.

On completion of the DC Bus sampling and once the position of the mains input phasor has been determined, the value of the DC-Bus is passed to the PI controller to determine a phase shift required to control the charging of the battery set.

The flow diagram representing the PI controller is shown in Fig.7-6. The controller used is a standard PI controller, having standard proportional $K_p$ and Integral $K_i$ constants, which have been modified to suit the application.

The DC-Bus voltage sample $vdcadc$ is compared with a reference value $vdcref$, and the voltage difference is then scaled by a factor $K_i$ (integral constant), and numerically integrated. It was found that it was necessary to limit the output of the integral unit in order to improve the dynamic response of the system, and then scale this output by a factor $K_{sc}$ before adding it to the voltage difference scaled by $K_p$ (proportional constant).

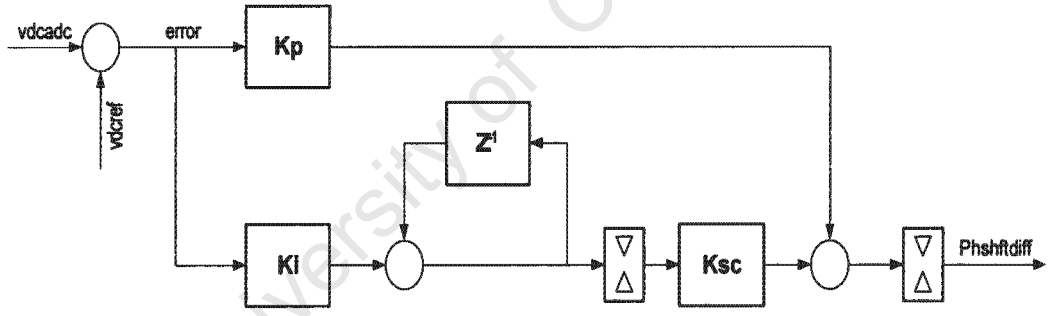



Figure 7-6.DC-Bus Numerical PI controller

The ouput variable Phshftdiff is then limited, such that the maximum phase shift is in the range of -30° to +30°so as to try and limit the incoming power factor within an acceptable range as described in Chap.5. The phase shift calculated by the PI controller is then added to the inverter output angle, and this new inverter angle is used for phase locking.

A simple proportional controller is then used to control the phase locking of this new inverter angle to the incoming mains phasor angle. The difference between these angles is found and scaled by a proportional constant. This value is then used to adjust the period of Timer1, which

University of Cape Town

results in a change in the inverter fundamental frequency, allowing for the inverter fundamental voltage output to be shifted relative to the incoming mains.

Finally the ADC channels are changed back, so as to be ready for sampling the incoming mains in the next cycle. The Timer1 underflow interrupt is enabled and program execution is returned to the main loop.

# 5 Implementation of Procedures

Since the programming of the DSP was carried out in the C-Programming language, a number of difficulties were encountered in the development of the software. The problems were brought about by the following limitations:

- The C-Compiler does not always generate compact assembler code, and it was found necessary to either optimise the C-Code or write some sections in the assembler language. In some cases this resulted in a vastly shorter program length while achieving the same goal, and with much improved execution time.
- The C-Compiler does not always generate assembler code as is expected and in such case it is necessary to debug the assembly code manually, and rewrite the C code appropriately.
- It was found that programming of the DSP using a high level language such as C, removes lower level control of the DSP making it difficult to take advantage of certain parts of the DSP architecture to improve execution time.
- Since the DSP is of fixed point architecture, it was found that use of floating point and signed integer variables greatly increased execution time. Thus it was necessary to write the majority of code using unsigned integer variables, and find innovative ways of achieving the same goals.

Following is an explanation of certain sections of code that took substantial time to develop, and that required use of innovative methods to achieve compact and efficient code:

## 5.1 Routine to generate quadrature mains input waveforms

The mains input waveforms supplied by the input transformers are two sinusoidal waveforms phase shifted by $60^o$. To determine the mains phasor angle, these waveforms first need to be first converted to two quadrature waveforms, in order to simplify the calculation of SVPWM. This can be achieved by applying the sampled input signals $v1adc$ and $v2adc$ to Eqns.:30 and 31[2] .

$$v1adc = v1adc \tag{30}$$

$$v2adc = \frac{-1}{\sqrt{3}}v1adc + \frac{2}{\sqrt{3}}v2adc \tag{31}$$

The implementation of a constant such as $\frac{1}{\sqrt{3}}$ on a fixed point DSP, cannot be directly implemented. To solve this an integer approximation of $\frac{37}{64}$ was used, which has the advantage of

University of Cape Town

multiplication by an integer and division by a multiple of two, which can be executed in a single clock cycle by appropriate use of the DSP architecture.

## 5.2 Determining Input Mains Phasor Angle

To be able to determine the position of the input mains phasor, the three mains phases are converted to two sinusoidal waveforms with sixty degrees phase shift, using two transformers as described in Chapter 6. These two sinusoidal waveforms are sampled by the ADC unit of the DSP at the beggining of each switching cycle, resulting in two values with ten bit resolution. Using a mathematical relationship the waveforms are converted to two quadrature sine waves, from which the phasor position can be more easily extracted. The phasor angle is then determined using a standard trigonometrical relationship, where one converted sample is divided by the other and an arctan operation is performed on the resulting value. Several problems were encountered in trying to achieve this in the minimum possible time, with sufficient resolution and using integer variables.

Firstly, since the quadrature sine wave samples are sampled and stored as unsigned integer variables, it is necessary to convert the values obtained into a format where one integer value can be divided by the other without loosing resolution.

University of Cape Town

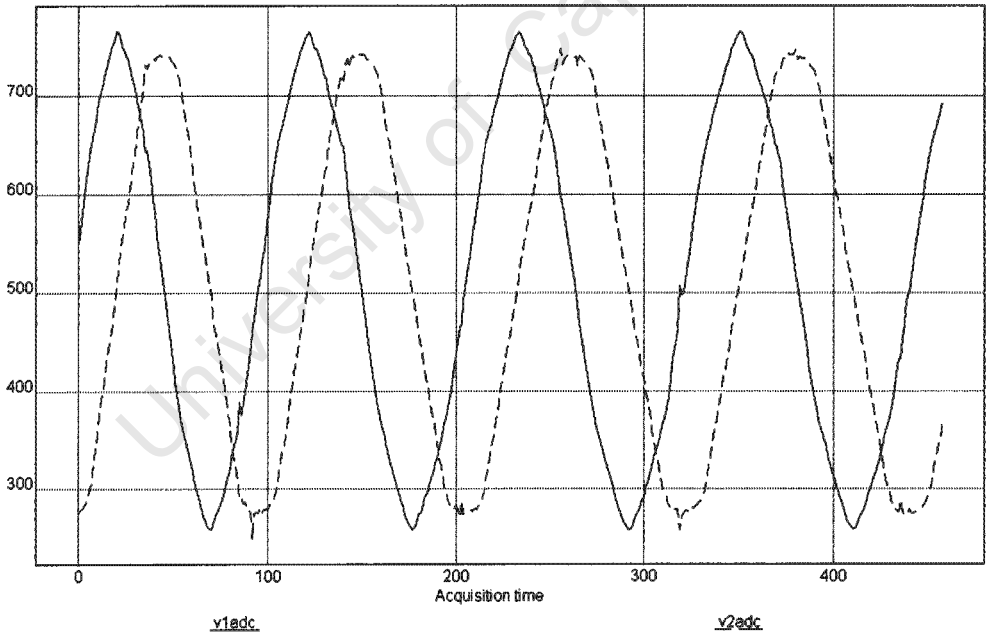
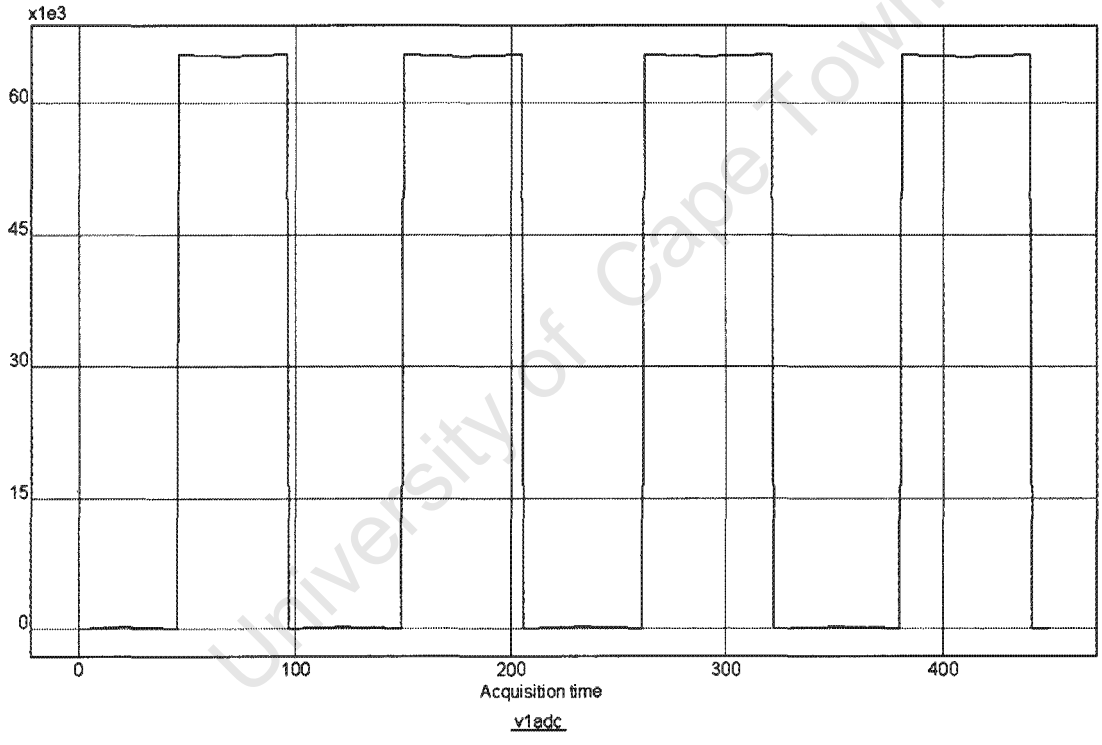



Figure 7-7.Mains Waveforms after 30deg Phase Shift

Shown in Fig.7-7, are the typical quadrature waveforms obtained during program execution.

Trace $v1adc$ is a waveform sampled directly from the input mains transformers, $v2adc$ is the waveform that has been mathematically generated from the input mains samples, so as to obtain two quadrature waveforms. It should be noted that the instantaneous values of both of these waveforms are unsigned integers and thus have a DC-offset (~512), which must be taken into account in subsequent calculations.

In order to divide the instantaneous value of one waveform by the other it is necessary to remove this DC-Offset, before division takes place. Since the variables used are unsigned integers, it is not possible to simply subtract the DC-Offset as this would result in an undefined value for at least half of the waveform. This however proved crucial in finding a simple solution to the problem. Subtraction of the DC-Offset from these variables results in a waveform as shown in Fig.7-8 , where the negative part of the sine wave has been shifted to the top half of the 16bit integer range.




University of Cape Town

Figure 7-8.Waveform after Subtraction of DC-Offset

Since it is only necessary to know the sign and magnitude of each sample at any one instant, to be able to determine the input mains phasor position, the sign of each sample can be stored in a variable called $sign$ and the waveform rectified mathematically as shown in Fig.7-9. This waveform gives us the magnitude of each sample at any one instant in time, and it is now

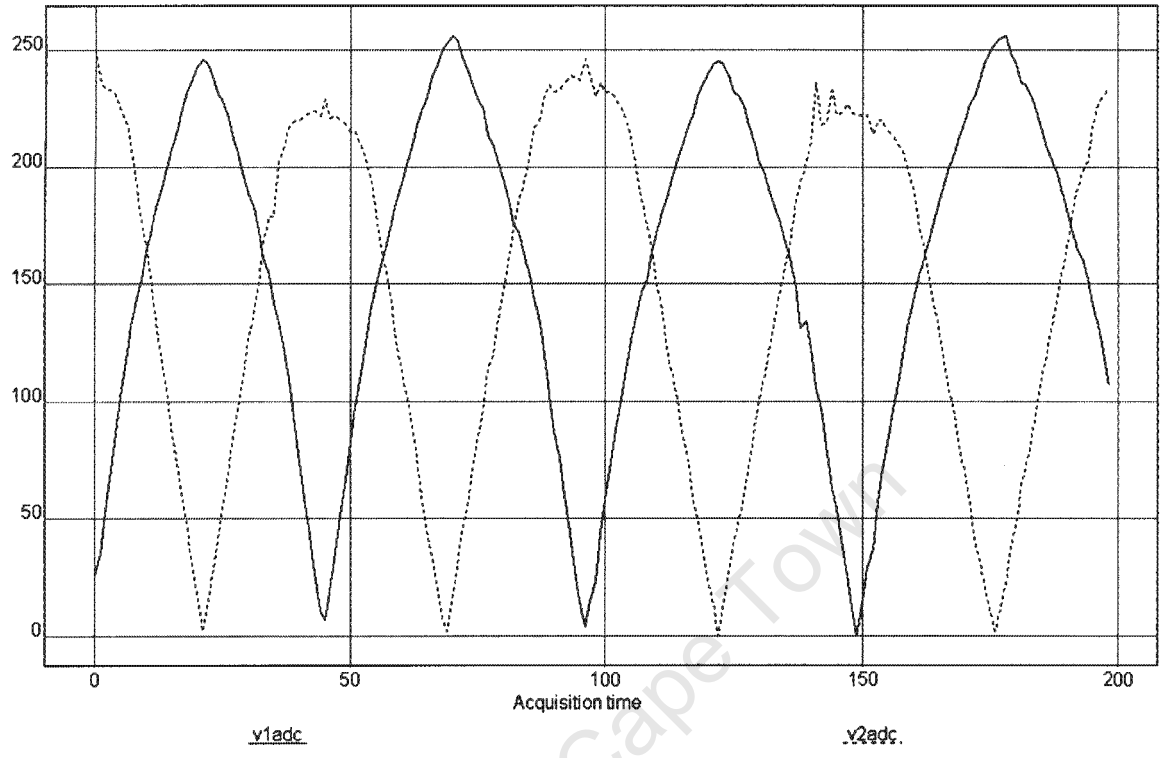



Figure 7-9.Resulting Rectified Waveform

possible to divide one value directly by the other. The next problem that arises from this is that an integer type division must be carried out in order to ensure fast execution, and this results in insufficient resolution.

In order to solve this problem an innovative solution was required, and by careful analysis of the problem, a simple solution was found. Since the samples from the ADC unit consist of ten significant bits, if the larger of the two samples is multiplied 64 (left shifted by six bits), no significant bits are lost. If this value is then divided by the smaller sample using integer division, it is possible to achieve an improved resolution by 1 in 64. The result of the division is then converted to an angle using a modified reverse lookup arctan table, to an accuracy of one degree.

## 5.3 Calculation of ArcTan

The process of an arctan reverse lookup was also found to be a challenging problem, as by the nature of a reverse lookup table, it is necessary to search for a solution. This means that for a reverse lookup in a table of 45 values, upto 45 compare operations need to be carried out

University of Cape Town

before a solution may be found. Since each compare operation requires several CPU cycles to complete, this can amount to several microseconds of execution time for a simple operation.

By simply starting the search half way through the lookup table, the maximum number of compare operations can be reduced by half as the search needs only to occur in one direction. This however was still found to result in an excessive execution time, and an alternative solution was necessary.

A more elegant solution was found by realising that for each switching cycle, the mains phasor angle will advance by approximately 3°. Assuming that this is true, and starting the search in the lookup table from the position of the last solution, it is possible to reduce the maximum number of search operations to 3. The resulting algorithm was found to greatly reduce the maximum execution time to an acceptable level.

## 5.4 Phase Locking Routine

The process of phase locking the inverter phasor angle to the mains phasor angle is not as simple as finding the difference between the angles and adjusting the inverter angle to match that of the mains angle. This is brought about by the fact that both of these angles are discontinuous functions at the 360° / 0° boundary. This can be seen in Fig.7-10, where the mains angle is shown leading the inverter angle. As can be seen during time period A, the difference between the angles is positive and in time period B negative, even though the phase difference is constant.

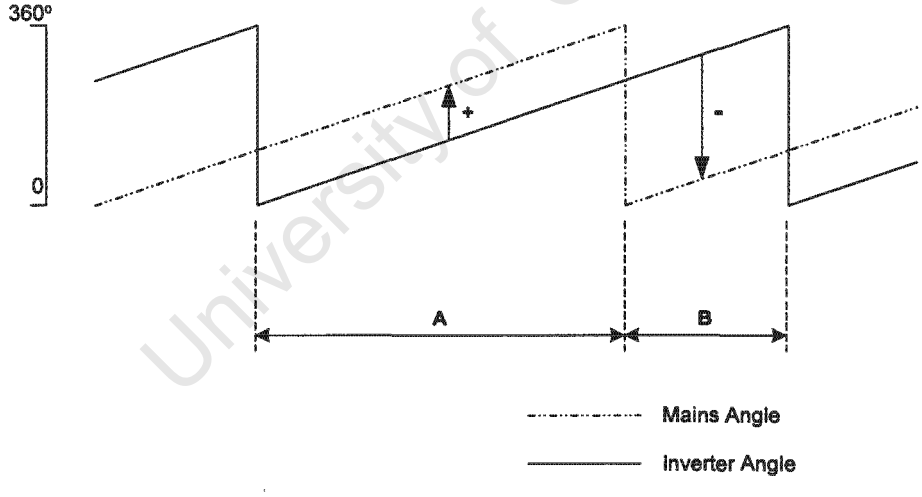



Figure 7-10.Mains / Inverter phase locking principle

To solve this problem using integer variables, the larger value is always subtracted from the smaller such that the result is always positive. In order to ensure the result of the difference does not jump at the dicontinuities, it is necessary to have a different strategy for time period A and

University of Cape Town

time period B. For instance if in time period A, the inverter angle is subtracted from the Mains angle, in time zone B it is necessary to subtract the mains angle from the inverter angle and then subtract this result fom $360^o$. The result of this is a difference range of $0^o$ to $360^o$, from which it can be determined whether to increase or decrease the inverter fundamental frequency in order to phase lock.

University of Cape Town

# Chapter 8
# Experimental Results

## 1 Introduction

As the one of the objectives of this project was to build a UPS system, a laboratory prototype UPS was built and tested. The results of these tests are shown and discussed in the sections that follow. The maximum power rating of the system was limited by the number of batteries available for the project, as this limits the maximum output voltage and consequently the power to the load. The maximum power transferred during these tests was 650W, however if a full battery set was available a power transfer of 10kW would have been possible using the rest of the system as is.

The battery set available was a set of 14 lead acid 12V 7AH batteries connected to the DC-Bus of the Inverter. The inverter used was a Semikron IGBT inverter, with a SKHI60 IGBT driver board and three IGBT modules (SKM200GB 123D) each rated at 200A, mounted on a fan cooled heat sink. The link inductors available for the project were three 14.8mH inductors. The loads used were a selectable set of 100W and 200W light bulbs in combination with 83Ω resistive loads.

Measurement of signals was carried out using a Hewlett Packard 54601B 100MHz 4-Channel digital oscilloscope and a Agilent 54622D 100Mhz 2-Channel digital oscilloscope. Both scopes were connected to a PC through an optically isolated cable (to ensure no ground loops), so as to allow trace waveforms to be saved. The software used to capture these waveforms was HP Benchlink Scope ver1.6. Current waveforms were captured using the above oscilloscopes connected to HTP25 LEM current modules.

Before carrying out full tests on the experimental setup, it is necessary to check that the software and hardware that has been built is operating as was intended.

## 2 Results of System Testing

To check that the resulting pwm output is as intended, is not so easy by just looking at the PWM switching output waveform, it is thus necessary to find an alternative method of checking this. This can be seen from Fig.8-1, which shows a trace of the phase to neutral output of the inverter, although the switching agrees with the theoretical values[8] , it is not clearly visible that the fundamental voltage output is correct.

University of Cape Town

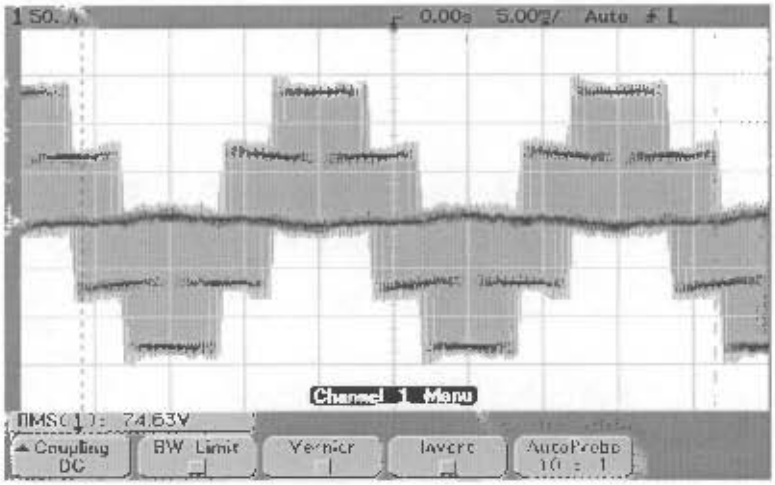



Figure 8-1.Phase to Neutral Unfiltered Inverter Output Voltage

In order to check for this the theoretical space vector fundamental modulating waveform was compared to the ones generated by software within the DSP and also to the filtered PWM output. Fig.8-2 shows the three compare register values captured by the DMCD Pro software during program execution which are identical to the modulating waveform expected in theory. These waveforms are also the fundamental line to neutral waveforms expected on the inverter output.

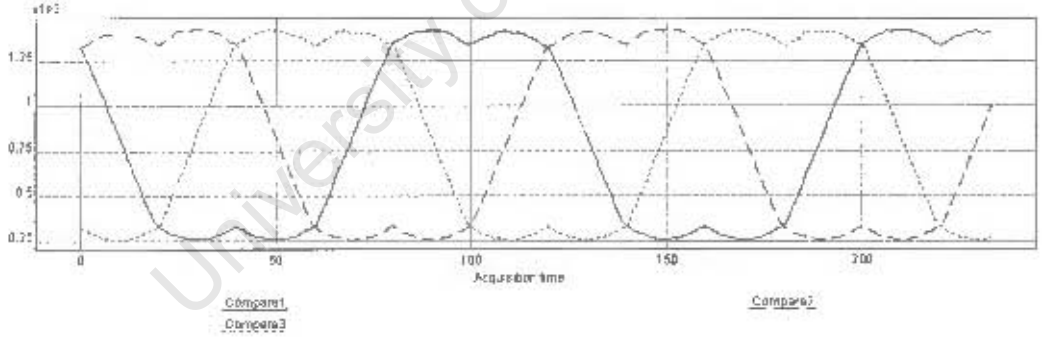



Figure 8-2.DSP Captured Compare Register Waveforms

Fig.8-3 shows the difference between two of these compare register waveforms captured by the DMCD Pro software during program execution, and as expected it is a sinusoidal wavform. We also expect to see this waveform if we look at the line to line output of the inverter.

University of Cape Town

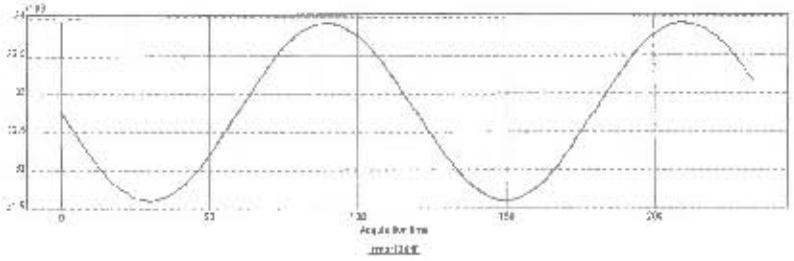


Figure 8-3.DSP Captured Compare Register Difference Waveform

A final check is to ensure that the inverter output is as expected, this was carried out using a set of three second order RC filters connected in star to the output of the inverter. The second order filters filter out the PWM switching frequency such that the output modulating waveform can be viewed. Fig.8-4 shows two of the line to neutral filtered inverter outputs in the top trace, with the mathematical difference of the waves in the bottom trace corresponding to the line to line filtered output. The two space vector line to neutral waveforms "consist of a desired sinusoidal fundamental component and a distorting triangular waveform at three times the fundamental frequency"[8] , however the difference between the signals eliminates this component and thus the inverter line to line voltages will be sinusoidal. As can be seen these waveforms correspond to the waveforms generated in the DSP and also to the theoretical expected waveforms.

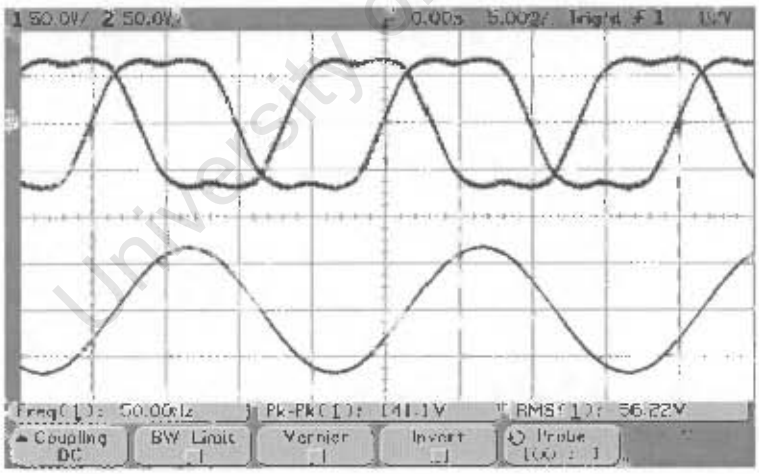



Figure 8-4.Inverter Output Line to Neutral and Line to Line Filtered Waveforms

University of Cape Town

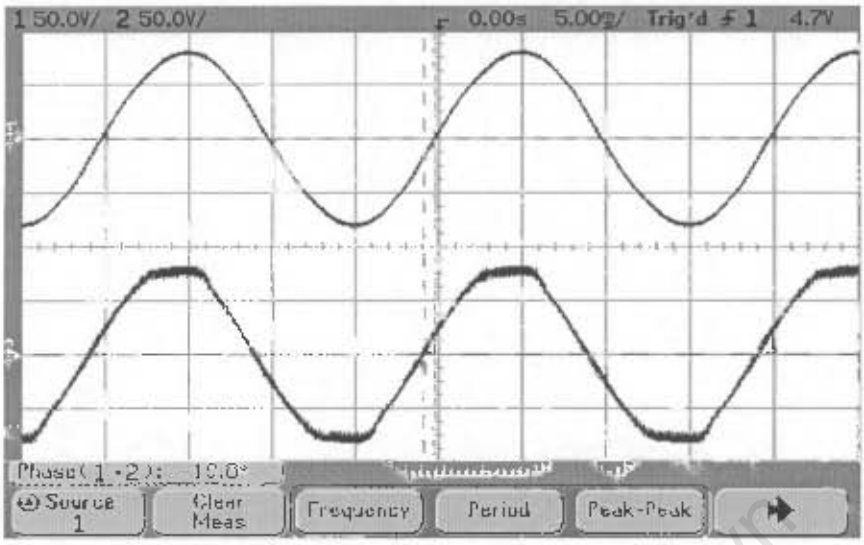



Figure 8-5.Filtered Inverter Output and Mains Input Voltage showing phase shift

## 3 Results of phase locking

Before connecting the inverter to the utility mains it was also necessary to confirm that the phase locking algorithm was functioning correctly. This was carried out by connecting the mains voltage control signals to the inverter such that the inverter could determine the utility mains phase. The three phase filtered output of the inverter was then compared to utility mains to ensure that it was in phase with and phase locked to it. Fig.8-5 shows a single phase of the inverter output on the top trace, compared to a single phase of the utility mains in the bottom trace, the $-10.8°$ degree phase shift between the waves being attributed to phase shift caused by the output filter.

Next the response of the phase locking algorithm was checked, to ensure that the inverter output waveform frequency does not get affected by sudden changes in the mains voltages such as voltage spikes and sags. Any change in utility mains frequency will be over a relatively long time period and thus the response of the phase locking controller can be such that phase locking occurs over several cycles. This ensures that on a mains failure the inverter output voltage does not change suddenly due to the sudden change of control signals. Fig.8-6 shows two traces of the angles of the utility mains space phasor and of the inverter output space phasor captured by the DMCD Pro software during program execution. These waveforms were captured during a system startup, and as can be seen it takes several cycles for the inverter output to phase lock onto the utility mains.

University of Cape Town

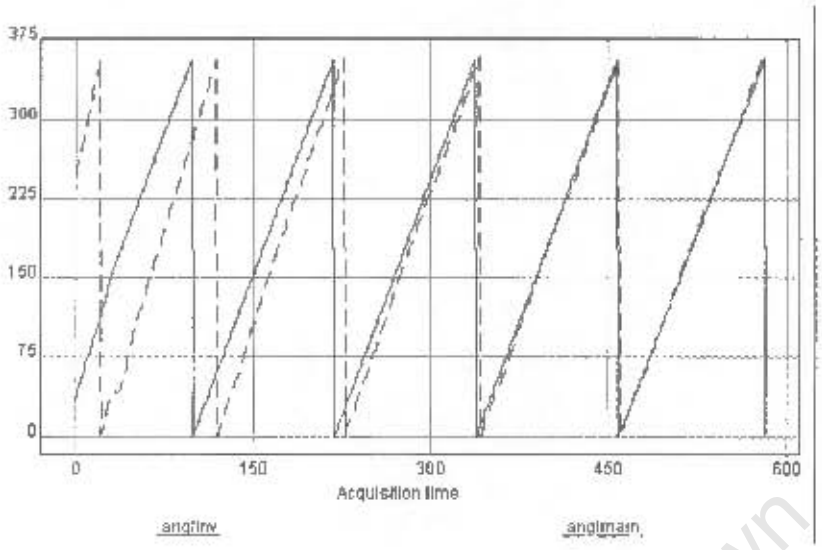



Figure 8-6.DSP Captured Inverter and Mains Phase angles during phase locking

# 4 Load Tests - During normal Operating Conditions

The next step in testing, once the correct functioning of the Hardware and Software had been established, was to test the system under load. For this a selectable bank of resistive loads was used to load the output of the system under normal operating conditions.

## 4.1 No Load Test

The results of the no load tests are shown in Fig.8-7 and Fig.8-8. Fig.8-7 shows one of the incoming mains voltages, and the filtered output of the same phase of the inverter during a no load test. As can be seen these are displaced from eachother by $640\mu s$, which corresponds to a $11.5^{\circ}$ phase shift caused by the filter. From this diagram it can also be seen that there is a significant improvement in the fundamental voltage waveform supplied to the load.

University of Cape Town

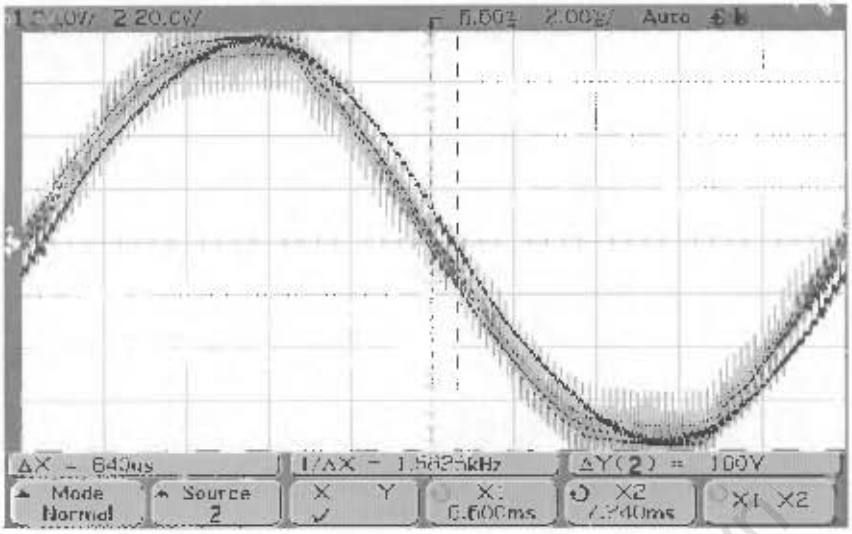



Figure 8-7.Input Voltage / Inverter Output Voltage Phase Shift at No Load condition

As expected in a no load condition, there is no effective phase shift between the input mains and the fundamental output of the inverter and the only power flowing from the mains will be to compensate for the switching losses, and imaginary power flowing due to slight differences in the inverter and mains phase voltages. This can be seen in Fig.8-8, which shows a single mains input voltage and current waveforms, where the input current is extremely small.

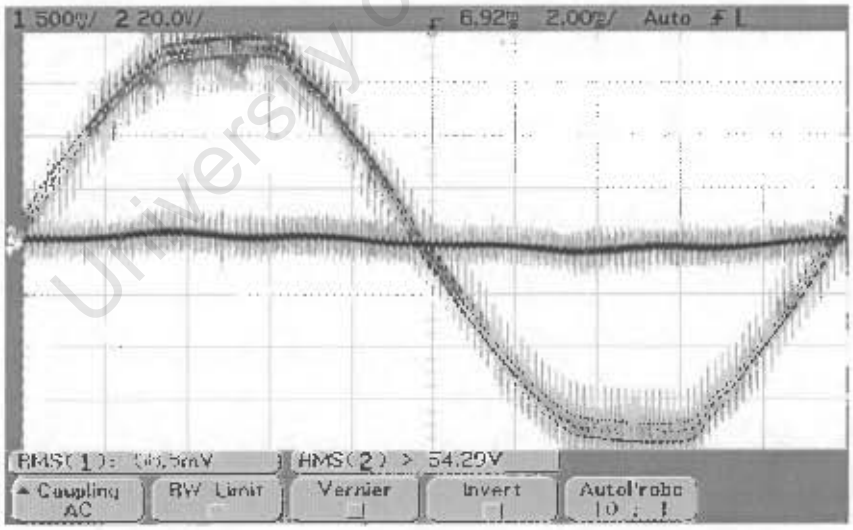



Figure 8-8.Input Voltage / Current Waveforms at No Load Condition

University of Cape Town

## 4.2 50% Load Test

The 50% load test consisted of connecting two 200$W$ light bulbs and one 83$\Omega$ resistive load in parallel on the output of the UPS. Fig.8-9 shows one of the mains phases input voltage and its respective current for this load condition, and from this we can see that the phase shift has been measured at 600$\mu s$ corresponding to a close to unity input power factor of 0.98. It can also be seen from this figure that the input current, being the smaller waveform is also near sinusoidal.

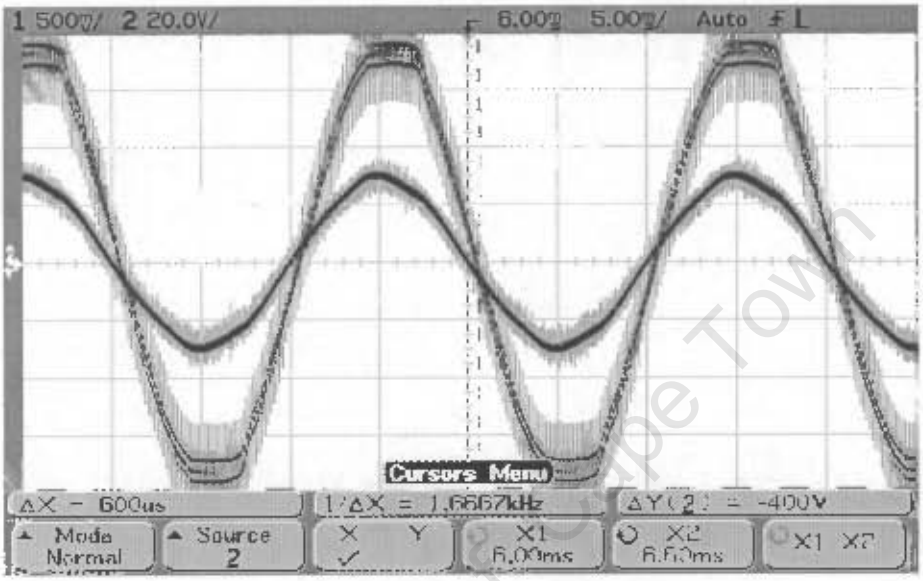



Figure 8-9.Input Voltage / Current Waveforms at 50% Load Condition

Fig.8-10 shows the phase shift between one phase of the utility mains and the same filtered phase of the inverter output, to be approximately 1.4$ms$ corresponding to a real phase shift of 14°, this being approximately half of the maximum possible phase shift.

University of Cape Town

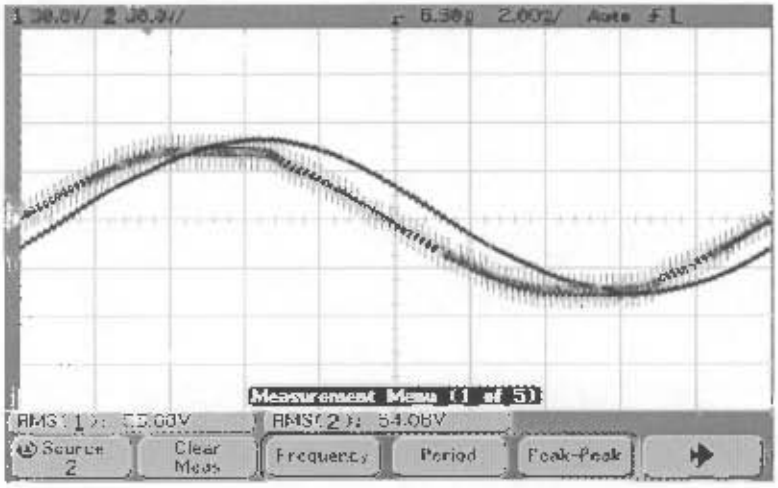



Figure 8-10.Input Voltage / Inverter Output Voltage Phase Shift at 50% Load Condition

The three phase input currents for the system shown in Fig.8-11, were captured using HTP25 current LEM modules configured with 220Ω resistors. As can be seen the input currents are near sinusoidal, equal in magnitude and equally displaced by 120°. Thus since the utility mains voltages are also equal in magnitude and displaced equally by 120°, we can assume that power drawn by the inverter is drawn equally from the three phases and supplied to the single phase load, as expected.

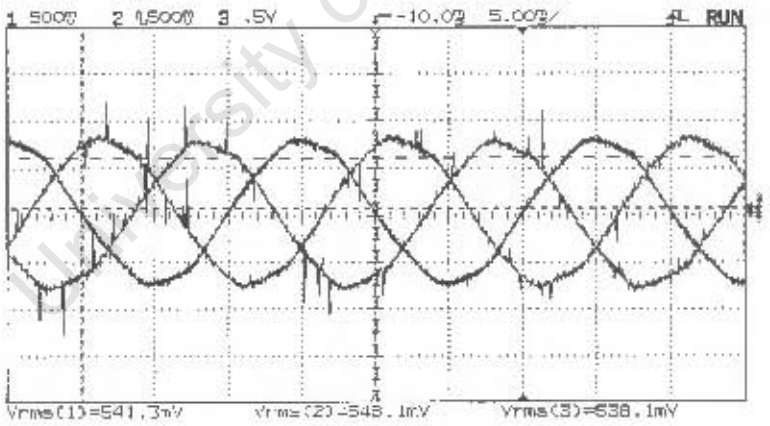



Figure 8-11.Three Phase Input Current Waveforms at 50% Load Condition

University of Cape Town

## 4.3 100%Load Test

The last of the load tests carried out, was the 100% load test, whereby the phase shift is at its maximum of 30°, this is shown in Fig.8-12 where the two waveforms are the mains input voltage and the filtered inverter output. The load resistance used was two 200$W$ bulbs, one 100$W$ bulb and two 84Ω resistances all in parallel.

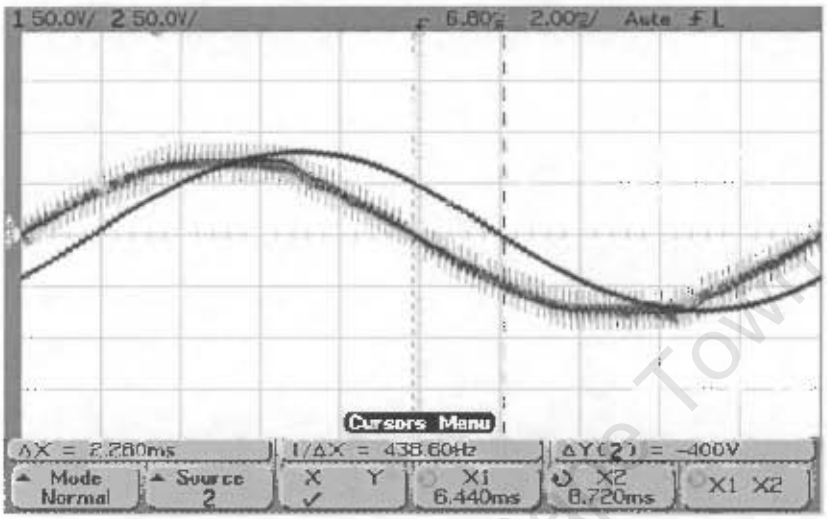



Figure 8-12.Input Voltage / Inverter Output Voltage Phase Shift at 100% Load Condition

Fig.8-13 shows the input voltage and its respective current waveform for the system at 100% load condition. As can be seen the power factor of the system is at unity powerfactor and has improved as the load has increased. This is contrary to what is expected as described in Chap.5, as this theory relies on the fact that the voltage vectors $V_{in}$ and $V_{out}$ are equal in magnitude. In real operating conditions these vectors are slightly different in magnitude and if $V_{out}$ is larger than $V_{in}$ the input power factor may improve as the phase shift angle is increased, as shown by our results.

University of Cape Town

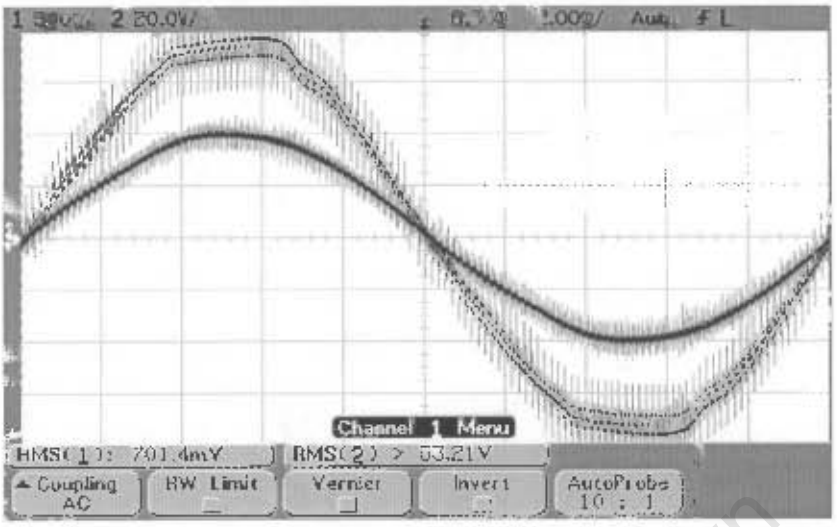



Figure 8-13.Input Voltage / Current Waveforms at 100% Load Condition

The three phase input currents for the system operating at 100% load in the normal operating condition, are shown in Fig.8-14. Since these waveforms are equally displace and of equal magnitude we can assume that the power drawn by the sytem is drawn equally from the the three phases even at higher loads.

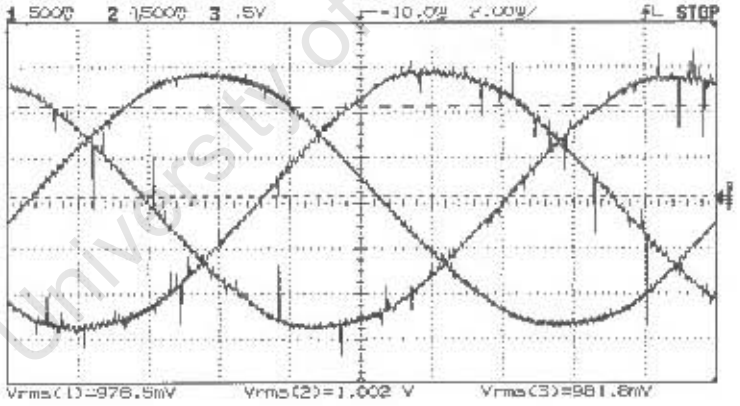



Figure 8-14.Three Phase Input Current Waveforms at 100% Load Condition

University of Cape Town

# Chapter 9
# Conclusions

The use of Space Vector modulation in this three to single phase UPS system proved to be advantageous. It both improved the DC-Bus Utilisation and provided a robust method of phase locking. The control of power flow and phase controlled battery charging, based on varying the phase angle was implemented successfully. Based on the results the following conclusions can be made:

- It is possible to implement the Space Vector Modulation technique on the TMS320F243 DSP. Space Vector PWM technique was successfully implemented in software to control a three to single phase system for an uninterruptible power supply.
- It is possible to implement a robust form of phase locking with the use of the Space Vector Modulation Technique using DSP technology. It was however found that close to the full processing capacity of the DSP was used to implement the phase locking and space vector modulation routines. An alternative solution needs to be found to reduce the processing requirements on the DSP.
- Power flow between the mains supply and the inverter can be controlled relatively easily using the phase control technique. By using this technique during normal operation of the UPS phase controlled battery charging can be implemented.
- Although it is possible to try and limit the input power factor, it is not possible to control it independently of the input power flow phase controlled battery charging. By careful design it was found possible to keep the input power factor close to unity during normal UPS operation.
- Using this topology it is possible to draw power equally from all three phases of the utility mains and supply it to any combination of single phase loads connected to the output in parallel. This would be an effective means of load balancing single phase loads onto a three phase system.

University of Cape Town

# Chapter 10
# Recommendations

The following recommendations can be made based on the results obtained in this project:

- Since testing was carried out at a low power level (600W) due to there being a limited number of batteries, the system should be tested at a higher DC-Bus voltage by installing more batteries and hence test the system at a higher power level in the order of 10kW.
- The phase locking implemented on this system was found to be extremely robust, due to the fact that the phase and magnitude of all three phases are used to achieve phase locking. It is recommended that this system be used instead of conventional phase locking in other converter topologies.
- Although the implemented system was found to operate successfully, it was found that the processing load imposed on the DSP was close to its limit. It is thus recommended that the processing to determine the mains space vector magnitude and phase be implemented external to the DSP, possibly in programable logic. Further optimisation of the program code would also be advisable, as the use of floating point variables in the program is inherently inefficient for a fixed point Digital Signal Processor.
- In this system the modulation index was kept constant at unity. It is recommended that the modulation index be made variable for better regulation of the output voltage.
- Since the output power factor of this system cannot be controlled independently of the DC-Bus voltage, it is recommended that a variable voltage DC-Bus system be investigated with regards to power factor improvement.
- The analysis of this system was carried out assuming that the three supply phase voltages are balanced and of equal magnitude. The effects of supply voltage unbalance on the system should be investigated further
- The system developed thus far only operates in the mode where the battery set is fully charged. A routine for the battery charging mode of operation needs to be developed, where the phase shift angle is controlled to limit the battery charging current.
- A UPS shutdown routine also needs to be implemented for when the battery power is exhaused on a mains failure. This routine would include a mains detection routine for establishing when the mains supply is back to normal so that the UPS can be reconnected to the grid and battery charging can occur.

University of Cape Town

# References


[1] MOHAN, N.,UNDELAND, T.M., ROBBINS, W.P.:'Power Electronics: Converters, Applications and Design', Wiley

[2] LAZAR,J.:'Converter Controlled Electrical Drives, Volume 1. Park-Vector Theory of Line Commutated Three-Phase Bridge Coverters', OMIKK'

[3] HO,W.-J.,and LIO,J.-B.,and Feng,W.-S.:'Economic UPS structure with phase-controlled battery charger and input-power-factor improvement', IEE Proc. Power Appl., Vol. 144, No. 4, July 1997

[4] TMS320F/C24X DSP Controllers - CPU and Instruction Set - Reference Guide - SPRU160C - June 1999

[5] TMS320F243/F241/C242 DSP Controllers - System and Peripherals - Reference Guide - SPRU 276C - January 2000

[6] Texas Instruments:'Space vector modulation hardware/software determined switching pattern', Application Report SPRA524

[7] Texas Instruments:'Field Orientated Control of 3-phase AC-Motors', Literature no.BPRA073

[8] ADI :'Space Vector Modulation', http://www.analog.com/industry/motor_control /seminars/mctech/ch2.html

[9] ADI :'Bus Clamping forms of SVM', http://www.analog.com/industry/motor_control /seminars/mctech/ch3.html

[10] VAN DER BROECK,H.W.,and SKUDELNY,H.-C.,STANKE,G.V.:'Analysis and Realisation of a Pulsewidth Modulator Based on Voltage Space Vectors',IEEE Transactions on Industry Applications, VOL.24,NO.1,January/February 1998

[11] HARAS,A.:'Space Vector Modulation in Orthogonal and Natural Frames Including the Overmodulation Range',EPE'97, pp337-342

[12] American Power Conversion:'Tech Note #T1 - The different types of UPS Systems', http://www.apcc.com


University of Cape Town

# APPENDIX A Program Code

University of Cape Town

### A.1. Main C-Code Program - Spcadc.c

```
/*;*****************************************************************************
; File Name: spcad.c
; Project: MSK243
; Originator: Daniele Beber
; Date: 21/03/2000
; Last update:
; Description: C file for outputing a space vector modulated sine wave to
; PWM outputs
;
;*****************************************************************************
;-----------------------------------------------------------------
; This program outputs a space vector modulated sine wave to the PWM compare
; outputs CMPR1-CMPR6 .Three phase output is generated. This is achieved by
; using timer1 to generate a 50Hz vector (continuous up/down counter) and
; converting this to switch the six PWM outputs in Space Vector Mode, using
; software generated switching. It also read in two sine waves 60deg to eachother
; from adc4 and adc6, and converts to two quadrature sine waves which are converted to
; to a phasor angle and magnitude so as to be able to generate phase locking
;
; GPT registers that need to be set up : OCRA
; GPTCON
; T1PER 7407h
; T1CNT 7405h
; T1CMP 7406h
; T1CON
;
; PWM registers that need to be set up ACTR 7413h
; DBTCON 7415h
; CMPR1 7417h
; CMPR2 7418h
; CMPR3 7419h
; COMCON 7411h
;
; ADC registers that need to be set up : ADCTRL1 7032h
; ADCTRL2 7034h
; ADCFIFO1 7036h
```

University of Cape Town

```
; ADCFIFO2 7038h
;
; Interrupts that need to be set up: tpint1vec (period interupt)
; tufint1vec (underflow interrupt)
;
;_____*/
/************************/
/*Define Constants&Arrays*/
/************************/
const int swfreq=0x0682; /*Setup 6kHz switching*/
const int angleincr=2; /*2==no of degrees per switching (3deg)*/
unsigned int loop, dummy, i, j, k, a;
unsigned int x, sector, alpha, output;
unsigned int time1, time2, taon, tbon, tcon, t1, t2, t3;
unsigned int v1adc, v2adc, v3adc, v4adc, vdcadc, centre, adcratio, sign;
unsigned int spcincr, anglinv, anglmain, value1, value2, adctmp;
unsigned int diff, dd, dump1, dump2;
signed int phshftint, phshft, phshftdiff, phshftact, vdcref;
signed int potref, vdcavg, tempp, vdc[10];
unsigned int angle;
float tper1, sinsectm[40];
unsigned int natam[91];
/************************/
/**********************/
/*Define include files*/
/**********************/
#include "F243regC.h"
#include "math.h"
#include "logger.h"
asm(" .include f243_a.h");
asm(" .include vect_243.h");
asm(" .include spcadc_a.asm");
#include "sinsecm.h"
#include "natam.h"
/**********************/
/*************************************************************************************************
/*************************************MAIN PROGRAM*********************************
```

University of Cape Town

```
/*****************************************************************************************/
void main ()
{
/*********************************/
/*Initialise Constants/Variables*/
/*********************************/
a = 0;
loop=1; /*Initialise loop variables*/
dummy=0;
sector=0;
alpha=0;
angle=0;
time1=0;
time2=0;
centre=516; /*Centre = 511 (added correction)*/
spcincr=23;
anglmain=0;
sign=0x0000; /*this variable is used to store the signs of various variables*?
/*bit0 = sign of Alpha component (of mains space vector)*/
/*bit1 = sign of Beta component (||) 1=+ve*/
/*bit2 = sign of |alpha|>|beta|*/
j = 0;
k = 0;
phshft = 0;
phshftint = 0;
phshftact = 0;
/*********************************/
/*********************************/
/*Initialise External Routines*/
/*********************************/
init_logger (); /*External Routine to log data*/
init_sinsectm(); /*External Modified sine lookup Table*/
init_natam(); /*External Modified arctan lookuptable*/
/*************************/
/*******************************************************/
/*Initialise Port Registers */
/*Setup ports to output values for debugging purposes */
```

University of Cape Town

```
/*******************************************************/
dummy = OCRA;
dummy |= 0x0FC3; /*last value 0x2FC4 - Setup portB4-7*/
OCRA = dummy; /*Set Pins as PWM CMPx*/
PDDATDIR=0x3C00; /*Set PortD pins as output*/
/*******************************************************/
/*********************************************/
/*Initialise Timer Registers */
/*Setup timers appropriately for PWM output */
/*********************************************/
GPTCON |= 0x0080; /*Setting of underflow interrupt GPT1 starts ADC*/
T1CON |= 0x8802; /*Continuous up/down count mode*/
T1PER = swfreq; /*Set timer1 period*/
tper1 = T1PER;
T1CNT = 0x0001; /*Initialise T1 Counter set at 1 to avoid 1st interrupt*/
T1CMP = 0x0000;
T2CON = 0x9382; /*Continuous up count, x/8,Use T1ENABLE,
Enable Timer Compare Operation */
T2CMP = 0x0000;
IMRA |= 0x0280; /*T1 period & uf interrupt enable*/
IMRB |= 0x0000; /*T2 interrupts disable*/
IMRC |= 0x0000; /*Compare interrupts disable*/
/*********************************************/
/*****************************************/
/*initilise ADC registers */
/*Setup ADCs ready for sampling */
/*****************************************/
ADCTRL2 = 0x0400; /*Interrupt on two words in FIFO,
High priority interrupt, start conv.
on event manager signal, ADC prescaler 1*/
ADCTRL1 = 0x984C; /*Setup for ADCTRL1 - Enable ADC#1,ADC#2, ADCIN4, ADCIN6*/
/************************/
/*Clear ADCFIFO registers*/
/************************/
asm(" RPT #1");
asm(" LACL ADCFIFO1");
asm(" RPT #1");
```

University of Cape Town

```
asm(" LACL ADCFIFO2");
/*****************************************/
/*******************/
/*Enable interrupts*/
/*******************/
IMR |= 0x0002; /*Enable INT2 (for T1PINT & T1UFINT)*/
/*******************/
/***********************************************/
/*Define Interrupt SubRoutines */
/*Define memory locaion of Subroutines */
/***********************************************/
/*;Load ISR addresses to interrupt vector in on-chip block B2*/
asm(" LACC #t1per_ISR_assembler");
asm(" LDP #0");
asm(" SACL tpint1vec");
/*;Load ISR addresses to interrupt vector in on-chip block B2*/
asm(" LACC #t1uf_ISR_assembler");
asm(" LDP #0");
asm(" SACL tufint1vec");
/***********************************************/
/***********************************************/
/*Setup PWM Registers */
/*Initialise PWM output */
/***********************************************/
ACTR = 0x0666; /*Initialise Active high/low PWM outputs, CCW Space Vector
; Bits 15-12 not used, no space vector
; PWM compare actions
; PWM6/PWM5 - Active Low/Active High
; PWM4/PWM3 - Active Low/Active High
; PWM2/PWM1 - Active Low/Active High*/
CMPR1 = 0x0000; /*Initialise PWM Compare Registers*/
CMPR2 = 0x0000;
CMPR3 = 0x0000;
DBTCON = 0x0000; /*Initialise Dead Band Timer Register*/
COMCON = 0x8200; /*Enable compare operation. Update registers on underflow
;Enable compare and full compare output*/
/***********************************************/
```

University of Cape Town

```
/*********************************/
/*Start Timer1 */
/*Start PWM output */
/*********************************/
T1CON |= 0x0040;
/*********************************/
/************************************/
/*Mains Program Loop */
/************************************/
while(loop!=2) /*Program stays within this loop unless interrupted*/
{
}
/************************************/
}
/**********************Interrupt Service Routines****************/
/*****************************************************************/
/*****Calculating Switching Times and updating registers*********/
/*****************************************************************/
void t1per_ISR(void) /*Timer 1 Interrupt*/
{
dd = 1;
/*Section to determine sector of operation and angle alpha (angle within sector)*/
sector = 0;
tper1 = T1PER; /*Update value of variable tper1*/
angle = angle + angleincr; /*Sequence of these two equations affect length*/
/*of assembler generated due to the code that */
/*follows...*/
if(angle>=240) /*Prevent angle being 360deg ie. reset to zero*/
{angle=0;
}
alpha = angle;
while(alpha>=40) /*Determine alpha and sector*/
{sector = sector+1; /*Sequence of these two equations*/
alpha = alpha-40; /*affects length of assembler*/
}
/*****************************************************************************/
/*****************************************************************/
```

University of Cape Town

```
/*Section to determine switching times T0, T1, T2 */
/*NB! removed modindx */
/*************************************************************************/
t1 = sinsectm[39-alpha] * tper1;
t2 = sinsectm[alpha] * tper1;
switch(sector)
{
case 0: {time1 = t1;
time2 = t2;
break;
}
case 1: {time2 = t1;
time1 = t2;
break;
}
case 2: {time1 = t1;
time2 = t2;
break;
}
case 3: {time2 = t1;
time1 = t2;
break;
}
case 4: {time1 = t1;
time2 = t2;
break;
}
case 5: {time2 = t1;
time1 = t2;
break;
}
default: {COMCON = 0x0000; /*Puts PWM output in high impedance state*/
break; /*in the case of an execution error*/
}
}
/*Determine on times for inverter legs*/
t3 = tper1; /*This line reduces code length*/
```

University of Cape Town

```
taon = (t3 - (time1 + time2))/2;
tbon = taon + time1;
tcon = tbon + time2;
/*************************************/
/*Assign switching times taon,tbon,tcon to correct PWM Channel*/
switch(sector)
{
case 0: {CMPR2 = taon;
CMPR3 = tbon;
CMPR1 = tcon;
break;
}
case 1: {CMPR2 = tbon;
CMPR3 = taon;
CMPR1 = tcon;
break;
}
case 2: {CMPR2 = tcon;
CMPR3 = taon;
CMPR1 = tbon;
break;
}
case 3: {CMPR2 = tcon;
CMPR3 = tbon;
CMPR1 = taon;
break;
}
case 4: {CMPR2 = tbon;
CMPR3 = tcon;
CMPR1 = taon;
break;
}
case 5: {CMPR2 = taon;
CMPR3 = tcon;
CMPR1 = tbon;
break;
}
```

University of Cape Town

```
default: {COMCON = 0x0000; /*Put PWM output in high impedance state*/
break; /*in the case of an execution error*/
}
}
/*logger(); /*log data for debugging purposes*/
/*****************************************************************/
IFRA |= 0x0080; /*Clear Period interrupt flag register*/
/*End of interrupt service routine*/
}
/*****************************************/
/***Phase Locking Sub-Routine************/
/*****************************************/
void t1uf_ISR(void) /*Timer1 Underflow interrupt service routine*/
{
while(ADCTRL1 & 0x0080) /*Wait for ADC conversion to complete*/
{
}
/*****************************************/
/*Routine to read two mains input voltages */
/*****************************************/
v1adc = ADCFIFO2>>6; /*First phase taken as is*/
dump1 = ADCFIFO1; /*Read FIFO second time to clear it*/
v2adc = ADCFIFO1>>6; /*Second phase to be calculated by*/
dump2 = ADCFIFO2; /*adding two readings and manipulating*/
/*****************************************/
v3adc = v2adc; /*Store ADCFIFO value at this instant*/
v4adc = v1adc; /*for capturing later in program for */
/*debugging purposes*/
/*****************************************/
/*Routine to convert two phases at 60deg to */
/*two phases at 90 deg. */
/*****************************************/
asm(" SPM 3"); /*This causes a division by 64 after a
multiply operation, all in one
instruction cycle*/
v2adc = ((1024 + (v2adc<<1) - v1adc)*37); /*add 1024 to make sure doesn't go negative*/
v2adc = v2adc - 376; /*multiply by 37/64 to approx sqrt(3)
```

University of Cape Town

```
add waves to shift v2adc by 30 deg
subtract 376 to get right offset*/
/*If this eqn written in one line, it does
not give the correct result*/
asm(" SPM 0"); /*This reverts the state of the DSP to before
this routine
/*****************************************/
/*****************************************/
/*Routine to change ADC input channel */
/*to be able to sample the DC-Bus */
/*****************************************/
adctmp = ADCTRL1; /*Change ADC channel1 to ADCIN5 (Occurs after EOC)*/
adctmp &= 0xFF80; /*and ADC channel 2 to ADCIN3*/
adctmp |= 0x003A;
ADCTRL1 = adctmp;
ADCTRL1 |= 0x0001; /*Start ADC conversion of DC-Bus signal*/
/*****************************************/
/*******************************************************/
/*Rectify Quadrature Sine Voltage Waveforms and store */
/*sign of waveform */
/*******************************************************/
v1adc = (v1adc) - centre; /*v1adc>>6 Capture ADC reading and convert*/
if (v1adc >= (centre+1))
{v1adc = 0xFFFF - v1adc;
sign &= 0xFFFE;} /*FFFE Sign of alpha component is negative bit0=0*/
else
{sign |= 0x0001;} /*0001 else sign is positive*/
v2adc = (v2adc) - centre; /*Capture ADC reading and convert*/
if (v2adc >= (centre+1))
{v2adc = 0xFFFF - v2adc;
sign &= 0xFFFD;} /*FFFD Sign of beta component is negative*/
else
{sign |= 0x0002;} /*0002 else sign is positive*/
/*******************************************************/
/*************************************************/
/*Determine which rectified instantaneous voltage */
/*waveform is larger */
```

University of Cape Town

```
/*****************************************************/
if(v1adc > v2adc) /*Determine sign of is alpha>beta*/
{sign |= 0x0004; /*and store it*/
}
else
{sign &= 0xFFFB;
}
/*****************************************************/
/*************************************************************/
/* Routine to find ratio between alpha and beta components */
/* such that it's always greater than unity, to avoid using */
/* floating point operations the larger is shifted left six */
/* bits (multiplied by 64) and divided by the smaller value */
/* This gives a resolution of 1 in 64 which is sufficient */
/*************************************************************/
if(v1adc>v2adc) /*Determine larger component (alpha or beta)*/
{
v1adc = v1adc<<6; /*Left Shift larger value by six bits*/
if(v2adc>0) /*Check so as no to divide by zero*/
{
adcratio = v1adc/v2adc; /*Find ratio between components)*/
}
else
{adcratio = 0xFFC0;} /*If denominator zero, define ratio*/
}
if(v1adc<=v2adc) /*Determine larger component (alpha or beta)*/
{
v2adc = v2adc<<6; /*Left Shift larger value by six bits*/
if(v1adc>0) /*Check so as no to divide by zero*/
{
adcratio = v2adc/v1adc; /*Find ratio between components)*/
}
else
{adcratio = 0xFFC0;} /*If denominator zero, define ratio*/
}
if(adcratio>0xFFC0) /*Define largest ratio so as not to*/
{adcratio = 0xFFC0; /*exceed limits of lookup table*/
```

University of Cape Town

```
}
/*************************************************************************/
/*************************************************/
/* Routine to ArcTan ratio between alpha and beta */
/* Output is in the range 0..45deg */
/*************************************************/
while((adcratio > natam[spcincr])) /*Decrement position in look up table*/
{spcincr = spcincr - 1; /*if ratio > value at that position*/
}
while((adcratio < natam[spcincr]) & (spcincr < 45)) /*Increment position
inlook up table*/
{spcincr = spcincr + 1; /*if ratio < value at
that position*/
}
/*************************************************/
/*************************************************/
/*Depending on the instantaneous values/sign of the */
/*voltage waveforms, determine the mains phasor */
/*sector and thus angle */
/*************************************************/
switch(sign)
{
case 0: {anglmain = 270 - spcincr;
break;
}
case 1: {anglmain = 270 + spcincr;
break;
}
case 2: {anglmain = 90 + spcincr;
break;
}
case 3: {anglmain = 90 - spcincr;
break;
}
case 4: {anglmain = 180 + spcincr;
break;
}
```

University of Cape Town

```
case 5: {if(spcincr > 0)
{anglmain = 360 - spcincr;
}
else
{anglmain = 0;
}
break;
}
case 6: {anglmain = 180 - spcincr;
break;
}
case 7: {anglmain = 0 + spcincr;
break;
}
default: {COMCON = 0x0000; /*Put PWM output in high impedance state?*/
break; /*in the case of an execution error*/
}
}
/*****************************************************/
/*****************************************************/
/* Scale Inverter angle into the 0 to 360 degree range */
/*****************************************************/
anglinv = (angle) * 3; /*Current inverter angle (no need to compensate)*/
anglinv = anglinv>>1;
/*****************************************************/
/*************************************************************************/
/*Correct inverter angle to adjust for 30deg difference between line to*/
/*line space vector and line to neutral space vector */
/*************************************************************************/
anglinv = anglinv + 30;
/*************************************************************************/
/* logger();*/
/***************************************/
/*Sample DC Bus & Potentiometer reference*/
/***************************************/
vdcadc = ADCFIFO1>>6; /*Read ADC1 (DC Current)ADCIN5*/
potref = ADCFIFO2>>6; /*Read ADC2 (pot ref)ADCIN3*/
```

University of Cape Town

```
dump1 = ADCFIFO1;
dump2 = ADCFIFO2;
/*****************************************/
vdcref = 302; /*Set DC-Bus reference value*/
/*=81.5V= =109.2V=350 4.7k,2.2k *** 415, 1k,1k*/
/*************************************************************/
/*Calculate phase shift from dc-bus PID controller */
/*************************************************************/
/*****************************************/
/*Integrate DC-Bus value to reduce noise */
/*****************************************/
vdc[j] = vdcadc;
if(j >= 9)
{j = 0;
}
else
{j = j + 1;
}
tempp = 0;
k = 0;
while(k < 10)
{tempp = tempp + vdc[k];
k = k + 1;
}
tempp = tempp /10;
vdcavg = tempp; /*temp line*/
/*****************************************/
/*****************************************/
/*Implement PI Controller */
/*****************************************/
phshft = (tempp - vdcref);
phshftint = (31*phshftint + 2*phshft)/32; /*phshftint = (phshftint + 2*phshft/5);*/
if(phshftint > 500) /*Limit values of Integral*/
{phshftint = 500;
}
if(phshftint < -500) /*Limit values of Integral*/
{phshftint = -500;
```

University of Cape Town

```
}
phshftdiff = (phshftint/3 + phshft/4); /*Output of PI Controller*/
if((phshftdiff < 30) & (phshftdiff > -30)) /*Limit Values of PI controller output*/
{anglinv = anglinv - phshftdiff;
phshftact = phshftdiff;
}
if(phshftdiff >= 30)
{anglinv = anglinv - 30;
phshftact = 30;
}
if(phshftdiff <= -30)
{anglinv = anglinv + 30;
phshftact = -30;
}
/************************************/
/*****************************************************************/
/*****************************************************************/
/*Output phshftact to DAC to see locking for debugging */
/*****************************************************************/
value1 = (phshftact) & 0x00F0; /*Select B4-B7 of inverter angle/3*/
value1 |= 0xF000; /*Keep B4-B7 as outputs*/
PBDATDIR = value1; /*Output to bits 4-7 of Portb*/
value2 = (phshftact<<2) & 0x003C;
value2 |= 0x3C00; /*Keep D2-D5 as outputs*/
PDDATDIR = value2; /*Output to D2-D5 on PortD*/
/*****************************************************************/
/*****************************************/
/*Limit inverter angle to 0-360deg */
/*****************************************/
if(anglinv>=360) /*Make sure angle is in range 0-360*/
{anglinv = anglinv - 360;
}
/*****************************************/
logger(); /*Log selected Variables*/
/*****************************************/
/*Proportional Contoller for phase locking */
/*****************************************/
```

University of Cape Town

```
if(anglmain > anglinv)
{diff = (anglmain - anglinv);
}
if(anglmain < anglinv)
{diff = 360 - (anglinv - anglmain);
}
/*********************************************************************/
/*Change period of output 50 Hz wave to lock onto incoming */
/*********************************************************************/
if(diff>=180) /*mains lagging - decr. inv period*/
{T1PER = 0x682 + (360 - diff)*5;
}
else /*mains leading - incr. inv period*/
{T1PER = 0x682 - diff*5;
}
/*********************************************************************/
/*****************************************/
/*Routine to return ADC channel 1 to ADCIN6 */
/*and channel 2 to ADCIN4 */
/*****************************************/
adctmp = ADCTRL1; /*Change ADC channel1 to ADCIN6 (Occurs after EOC)*/
adctmp &= 0xFF80; /*and channel 2 to ADCIN4*/
adctmp |= 0x004C;
ADCTRL1 = adctmp;
/*****************************************/
IFRA |= 0x0200; /*Clear T1UF interrupt flag register*/
/*End of Underflow Interrupt Service
Routine*/
}
/**********************Define Entry Point************************/
void c_int0(void)
{
main();
}
/*********END OF PROGRAM CODE*****/
/*****************************************************************/
/*Spare Code */
```

University of Cape Town

```
/*This is code developed for the development/debugging stages */
/*of this project and is no longer used in the program however */
/*is still useful for debugging purposes */
/*****************************************************************/
/*****************************************/
/*Calculate phase shift from reading pot */
/*****************************************
tempp = vdcref/64;
if((tempp) >= 0x01FF)
{phshft = (tempp - 0x01FF)/32;
anglinv = anglinv + phshft;
}
else
{phshft = (0x01FF - tempp)/32;
anglinv = anglinv - phshft;
}
/************************************/
/*****************************************/
/*Calculate phase shift from dc-bus */
/*****************************************
tempp = vdcadc/64; /*Shift 10bits of AtoD reading 6bits to right*
if((tempp) >= 120)
{phshft = (tempp - 120);
if(phshft < 20)
{
anglinv = anglinv - phshft;
}
}
else
{phshft = (120 - tempp);
if(phshft < 20)
{anglinv = anglinv + phshft;
}
}
/************************************/
/*****************************************/
/*Calculate phase shift from dc-bus */
```

University of Cape Town

```
/*****************************************
tempp = vdcadc>>6; /*Shift 10bits of AtoD reading 6bits to right*
phshft = (tempp -100);
phshftint = (phshftint*2 + phshft)/2;
if((phshftint < 20) & (phshftint > -20))
{anglinv = anglinv - phshftint;
}
if(phshftint > 20)
{anglinv = anglinv - 20;
}
if(phshft > 20)
{anglinv = anglinv + 20;
}
/*********************************/
/*************************************************************/
/*Output inverter angle to DAC to see locking */
/*************************************************************
value1 = (angle>>1) & 0x00F0; /*Select B4-B7 of inverter angle/3*
value1 |= 0xF000; /*Keep B4-B7 as outputs*
PBDATDIR = value1; /*Output to bits 4-7 of Portb*
value2 = (angle<<1) & 0x003C;
value2 |= 0x3C00; /*Keep D2-D5 as outputs*
PDDATDIR = value2; /*Output to D2-D5 on PortD*
/*************************************************************/
/*
v1 = ADCFIFO1 - 32704;
v2 = ADCFIFO2 - 32704;
value1adc = v1;
value2adc = v2;
value3adc = v1/v2;
*/
/*****************************************************/
/* Routine to find absolute values of alpha and beta */
/* components of mains space vector and to record */
/* whether they are +ve/-ve at these instants */
/*****************************************************/
/*
```

University of Cape Town

```
alph[j] = ADCFIFO1>>6;
bet[j] = ADCFIFO2>>6;
j = j + 1;
if(j == 16)
{j = 0;
}
v2adc = 0;
v1adc = 0;
while(k<16)
{v2adc = v2adc + alph[k];
v1adc = v1adc + bet[k];
k = k + 1;
}
k = 0;
v1adc = v1adc/16 - 9;
v2adc = v2adc/16 -3;
*/
/*************************************************************/
/*Output DCRef to DAC to see setpoint */
/*************************************************************
value1 = (vdcref>>8) & 0x00F0; /*Select B4-B7 of Pot Reading*
value1 |= 0xF000; /*Keep B4-B7 as outputs*
PBDATDIR = value1; /*Output to bits 4-7 of Portb*
value2 = (vdcref>>6) & 0x003C;
value2 |= 0x3C00; /*Keep D2-D5 as outputs*
PDDATDIR = value2; /*Output to D2-D5 on PortD*/
/*************************************************************/
/*Tried to interpolate angle
if(anglinv == 0)
{tmp1 = adcratio - natam[spcincr];
tmp2 = natam[spcincr-1] - natam[spcincr];
fracangl = tmp1/tmp2; /* + anglmain;*
logger();
}
*/
/*********************************************************************/
/*Work out inverter angle relative to mains where 360=mains period*/
```

University of Cape Town

```
/*********************************************************************/
/*
if(anglmain <= 3)
{if(zz == 0)
{measold = anglinv;
zz = 1;
}
if(anglinv>measold) /*inverter period>mains period*
{measdiff = anglinv - measold;
if(measdiff>36) /*inverter period<mains period 36 instead of 72???*
{measdiff = 360 - measdiff;
measper = 360 - measdiff; /*obsolete line*
}
else
{measper = 360 + measdiff;
}
}
else /*inverter period<mains period*
{measdiff = measold - anglinv;
if(measdiff>36) /*inverter period>mains period*
{measdiff = 360 - measdiff;
measper = 360 + measdiff; /*obsolete line*
}
else
{measper = 360 - measdiff;
}
}
measold = anglinv;
/* mainsper = 360;
mainsper = T1PER * mainsper/measper;
mainsperu = mainsper;
measper= T1PER + measdiff;*
}
*/
/*
if(anglmain > anglinv)
{diff = (anglmain - anglinv);
```

University of Cape Town

```
}
else
{diff = (360 - (anglinv - anglmain));
}
*/
/*
output = 0x0020;
output = output&0x003C;
output = output|0x3C00;
PDDATDIR = output;
mainsper = measper;
bbb = 360;
ccc = mainsper/bbb;
*/
/*
output = 0x0000;
output = output&0x003C;
output = output|0x3C00;
PDDATDIR = output;
*/
/* value1 = (value1adc>>8) & 0x00F0; /*Select B9-B6 of ADC reading*
value1 |= 0xF000; /*Keep B4-B7 as outputs*
PBDATDIR = value1; /*Output to bits 4-7 of Portb*
value1 = (value1adc>>6) & 0x003C;
value1 |= 0x3C00; /*Keep D2-D5 as outputs*
PDDATDIR = value1; /*Output to D2-D5 on PortD*
IFRA |= 0x0100; /*Clear Compare interrupt flag register*/
/*****************************************/
/*Calculate phase shift from dc-bus */
/*****************************************
tempp = vdcadc>>6; /*Shift 10bits of AtoD reading 6bits to right*
phshft = (tempp -150);
phshftint = (phshftint + phshft)*3/4;
phshftdiff = phshftint + phshft;
if((phshftdiff < 20) & (phshftdiff > -20))
{anglinv = anglinv - phshftdiff;
phshftact = phshftdiff;
```

University of Cape Town

```
}
if(phshftdiff > 20)
{anglinv = anglinv - 20;
phshftact = -20;
}
if(phshftdiff < -20)
{anglinv = anglinv + 20;
phshftact = 20;
}
/***********************************/
/***********************************/
/*Add adjustment to inverter angle*/
/*just for test purposes */
/***********************************
if(vdcref > 511)
{tempint = (vdcref-511)>>4;
anglinv = anglinv + tempint; /*+1 increases power*
}
else
{tempint = (511 - vdcref)>>4;
anglinv = anglinv - tempint; /******Had problems with compiler******
}
/***********************************/
/***************************************/
/*Calculate phase shift from dc current */
/***************************************
tempp = idcadc;
phshft = (tempp - 511);
phshftint = (phshftint + phshft)*7/8;
phshftdiff = phshftint + phshft;
/* if((phshftdiff < 20) & (phshftdiff > -20))
{anglinv = anglinv + phshftdiff;
phshftact = phshftdiff;
}
if(phshftdiff > 20)
{anglinv = anglinv - 20;
phshftact = -20;
```

University of Cape Town

```
}
if(phshftdiff < -20)
{anglinv = anglinv + 20;
phshftact = 20;
}
/***********************************/
/*sector = angle/60; /*NB! Fractional values always rounded down*/
/* alpha = (256*(angle - sector*60))/60;*/
/*if(spcincr>45)
{spcincr=44;
}
*/
/*idcadc = idcadc + curroff;*/
/*vdcadc = vdcavg;*/
/*vdcavg = (vdcavg*7 + vdcadc)/8;*/
/* asm(" ROR");
asm(" ROR");
asm(" ROR"); /*Last three line to divide accumulator by 8
asm(" SACL _vdcavg, 0000h"); Store Accumulator Low in variable*/
/*
if((phshft < 5) & (phshft > -5))
{phshftint = 15 * phshftint / 16;
}
*/
/*phshftact = phshftact *5;*/
/*Tried to integrate difference "diff" by create variable "intdiff" and performing
intdiff = (intdiff *119 +diff)/120 where intdiff = float, problem- still have
oscillation */
```

University of Cape Town

### A.2. Interrupt Service Routines Assembler Definition - Spcadc_a.asm

```
;*****************************************************************************
; File Name: spc2_a.asm
; Copyright © 2000 Maria Rosa inc.
;*****************************************************************************
; Include Files
;-----------------------------------------------------------
.include F243_a.h
.include demos_a.h
.include vect_243.h
;=============================================================================
.global _t1per_ISR
.global _t1uf_ISR
MON243 .set 0e3h
;=============================================================================
.text
; -----------------------------------------------------------
; load ISR addresses to Interrupt Vector in on-chip block B2
LACC #t1per_ISR_assembler
LDP #0
SACL tpint1vec ; load _ISR_assembler address into corresponding
; interrupt vector
LACC #t1uf_ISR_assembler
LDP #0
SACL tufint1vec
t1per_ISR_assembler:
; entry to this ISR is done by the on-chip monitor which has made the
; following context saving:
;
; larp AR1 ; ARP=1, ARB=ARP before int.
; mar *+ ; skip one value on the stack AR1=TOS+1
; sst #1, *+ ; save ST1, AR1=TOS+2
; sst #0, *+ ; save ST0, AR1=TOS+3
; sach *+ ; save ACCH, AR1=TOS+4
; sacl *+ ; save ACCL, AR1=TOS+5
;
; call to ISR function in C:
```

University of Cape Town

```
;
CALL _t1per_ISR
;
; context restauration from an ISR:
;
;restore context save done by MON243
MAR *,AR1 ;ARP=1, AR1=TOS+5
MAR *- ;AR1–, AR1=TOS+4
LACL *- ;Restore ACCL, AR1=TOS+3
ADD *-,16 ;Restore ACCH, AR1=TOS+2
LST #0, *- ;Restore STO, AR1=TOS+1
LST #1, *- ;Restore ST1, AR1=TOS
;
clrc INTM
;
RET
; ————————————————————————————————————
t1uf_ISR_assembler:
; entry to this ISR is done by the on-chip monitor which has made the
; following context saving:
;
; larp AR1 ; ARP=1, ARB=ARP before int.
; mar *+ ; skip one value on the stack AR1=TOS+1
; sst #1, *+ ; save ST1, AR1=TOS+2
; sst #0, *+ ; save ST0, AR1=TOS+3
; sach *+ ; save ACCH, AR1=TOS+4
; sacl *+ ; save ACCL, AR1=TOS+5
;
; call to ISR function in C:
;
CALL _t1uf_ISR
;
; context restauration from an ISR:
;
;restore context save done by MON243
MAR *,AR1 ;ARP=1, AR1=TOS+5
MAR *- ;AR1–, AR1=TOS+4
```

University of Cape Town

```
LACL *- ;Restore ACCL, AR1=TOS+3
ADD *-,16 ;Restore ACCH, AR1=TOS+2
LST #0, *- ;Restore STO, AR1=TOS+1
LST #1, *- ;Restore ST1, AR1=TOS
;
clrc INTM
;
RET
;---------------------------------------------------------------
;---------------------------------------------------------------
```

University of Cape Town

# APPENDIX B HTTP References

University of Cape Town

# Space Vector Modulation

## 2.1 Introduction

Traditionally, pulsewidth modulation for ac drives had been implemented using analogue techniques. Ordinarily, the pulsewidth modulation signals are obtained by direct comparison of a triangular carrier waveform with some desired modulating function. For a three-phase machine, three independent pulsewidth modulation stages are required; one per phase. The three modulating functions applied to these PWM stages are proportional to the desired stator phase voltages. Most often these reference voltages are obtained as the outputs of analogue regulating current controllers.

However, the current control of ac drives is now moving towards a digital implementation. The generation of the pulsewidth modulation signals is an inherently digital problem. Consequently, it makes little sense to convert the current controller outputs to analogue in order to generate the digital PWM signals. Indeed, many modern processors include a digital PWM generation function on the same integrated circuit. Therefore, in a modern all-digital drive it is inevitable that some form of digital PWM algorithm is required.

It is possible to implement a direct digital equivalent of the comparison between a sawtooth carrier and a desired modulating function. These techniques are known as **regularly sampled** as the PWM signals are derived by comparison of a regularly sampled version of the modulating function with the carrier. This notation distinguishes these schemes from the analogue **naturally sampled PWM**. However, such schemes are simply close approximations to the traditional analogue sine-triangle pulsewidth modulation strategy.

In recent years, a new modulation technique known as **Space Vector Modulation (SVM)** has been developed. While offering a new, inherently digital computation method, this technique produces identical PWM signals to those that would be obtained from comparison of a triangular carrier waveform with a suitably defined modulating waveform. The main advantages of this modulation technique are:

- Simple, inherently digital calculation of the switching times
- A 15% increase in dc link voltage utilisation compared with sine-triangle techniques
- Lower harmonic content, particularly at high modulation indices, compared with sine-triangle techniques

## 2.2 Realisable inverter states

For a three-phase voltage source inverter (VSI) each pole voltage may assume one of two values depending on whether the upper or lower switch is turned on. Consequently, there are only eight possible operating states for the three-phase voltage source inverter. These different states are illustrated in Figure 1.

University of Cape Town

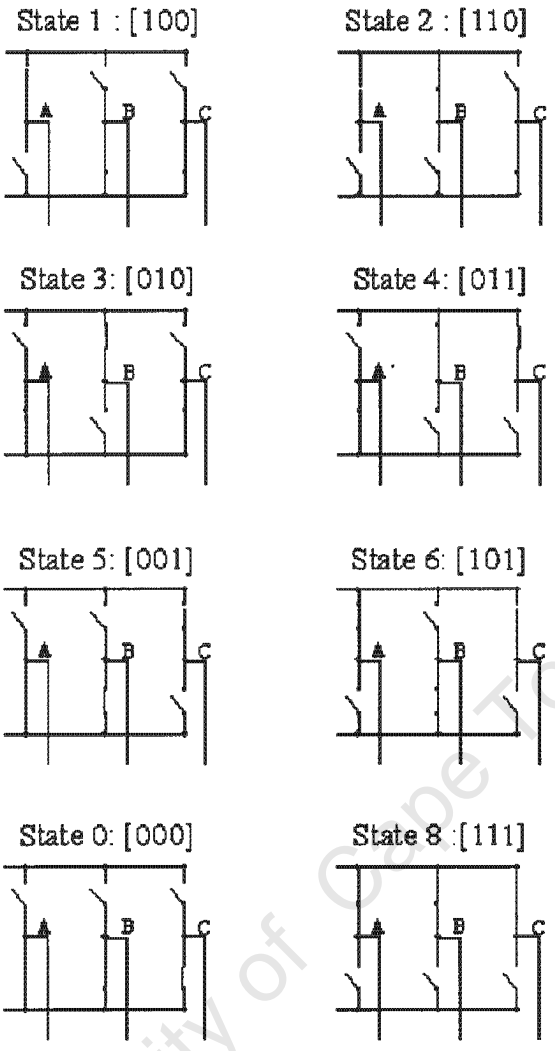



**Figure 1:** The 8 possible operating states for a three-phase VSI.

University of Cape Town

It emerges that the eight inverter states comprise six active states, State 1 to State 6, and two zero states, State 0 and 8. The six active states occur when either one upper and two lower or two lower and one upper inverter devices conduct simultaneously. The zero states occur when either the three upper or the three lower devices are turned on. These two states are often referred to as freewheeling states, since all motor currents are freewheeling during operation in these configurations. For each of the eight states it is possible to calculate the resultant voltage space vector using (2).

For example, when the inverter operates in State 1, corresponding to having the upper switch of inverter pole A and the lower switches of inverter poles B and C turned on simultaneously, the resultant inverter phase voltages are:

$$V_{AN_1} = \frac{2}{3}V_{dc}$$

$$V_{BN_1} = V_{CN_1} = -\frac{1}{3}V_{dc} \quad (1)$$

Substituting into (2), it is possible to show that the voltage space vector due to State 1 is given by:

$$v_{s1} = \begin{bmatrix} v_{s\alpha_1} \\ v_{s\beta_1} \end{bmatrix} = \frac{2}{3}V_{dc}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2)$$

The state, vs1, is shown in the complex space vector plane of Figure 2 as a vector of length $2V_{dc}/3$ lying along the real axis. Similarly, it is possible to compute the voltage space vectors for the other seven inverter states which are also illustrated in Figure 2. The six active voltage space vectors are of equal magnitude and mutually phase displaced by $\pi/3$. The general expression for the eight realisable voltage vectors is:

$$v_{s,k} = \begin{cases} \frac{2}{3} V_{dc} \exp\left(\frac{(k-1)\pi}{3}\right) & k = 1,\Lambda\ ,6 \\ 0 & k = 0,8. \end{cases} \qquad (3)$$

The six active vectors subdivide the space vector plane into six equal sectors.

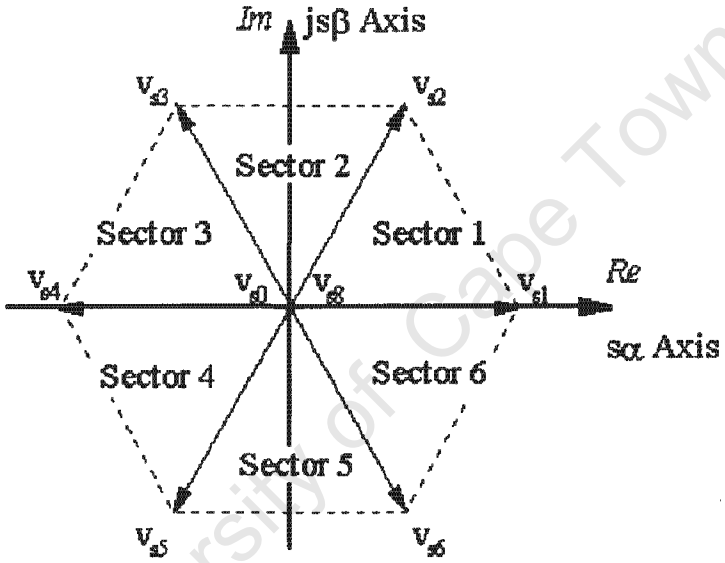



**Figure 2:** Realisable voltage space vectors for a three-phase voltage source inverter.

## 2.3 Inverter capability and reference voltage definition

The hexagon of Figure 2 represents the range of realisable voltage space vectors. Using the space vector modulation process it is possible to realise any arbitrary voltage space vector that lies within this hexagon. The maximum fundamental phase voltage that may be produced by the inverter for a given dc link voltage occurs under **six-step operation.** The resultant phase voltage developed by the inverter is shown in Figure 3, where the six different voltage levels, corresponding to operation at each of the active inverter states, are clearly seen.

University of Cape Town

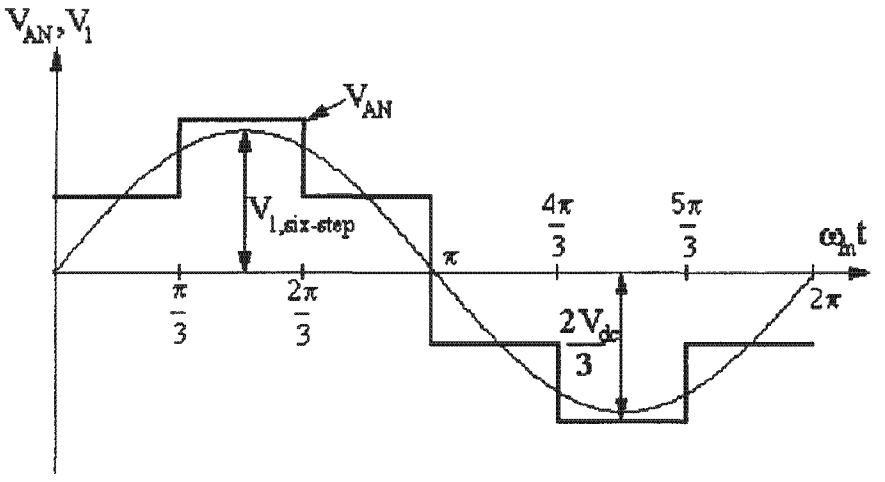



**Figure 3:** Resultant inverter phase voltage and corresponding fundamental component for six-step operation.

The fundamental component of the six-step voltage waveform is also illustrated in Figure 3. From the Fourier analysis, the fundamental voltage magnitude is given by:

$$V_{1,six-step} = \frac{4}{\pi}\frac{V_{dc}}{2} \quad (4)$$

This voltage level is achieved only at the expense of significant low frequency distortion. Nevertheless, it can be shown that for conventional sinusoidal modulation the maximum achievable fundamental voltage is:

$$V_{1,sin-pwm} = \frac{V_{dc}}{2} \quad (5)$$

so that only 78.5% of the available inverter capacity is used.

If the space vector modulator is required to produce a balanced three-phase system of voltages of magnitude $V_1$ and frequency $\omega_m$, given by:

$$\begin{aligned} V_{AN}^* &= V_1 \cos(\omega_m t) \\ V_{BN}^* &= V_1 \cos(\omega_m t + \gamma) \\ V_{CN}^* &= V_1 \cos(\omega_m t + 2\gamma) \end{aligned} \quad (6)$$

the corresponding reference voltage space vector is given by:

$$\begin{aligned} v_s^* &= V_1\left[\cos(\omega_m t) - j\sin(\omega_m t)\right] \\ &= V_1 e^{-j\omega_m t} \\ &= M\frac{V_{dc}}{2} e^{-j\omega_m t} \end{aligned} \quad (7)$$

The modulation index is defined as the ratio of the desired peak fundamental magnitude to half the dc link voltage:

University of Cape Town

$$M = \frac{V_1}{\left(\frac{V_{dc}}{2}\right)} \quad (8)$$

Therefore, the reference space vector describes a circular trajectory of radius $V_1$ at an angular velocity $\omega_m$ in the complex plane. Two reference voltage trajectories are shown in Figure 4 for modulation indices of 0.58 and 1.15, respectively.

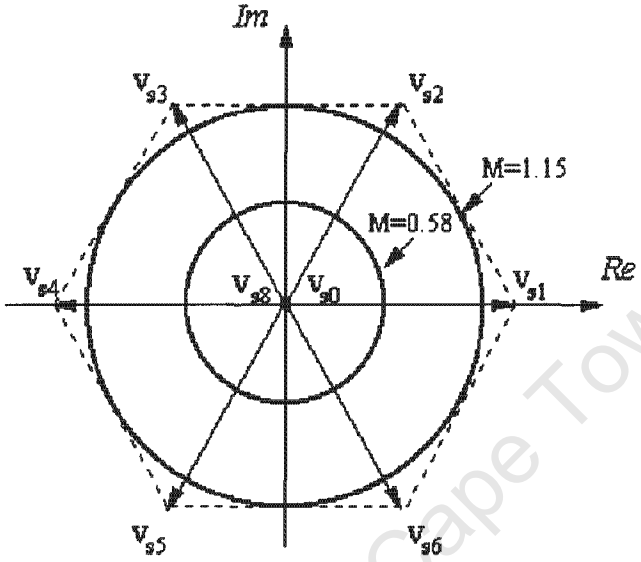



**Figure 4:** Reference voltage space vector trajectories for M = 0.58 and M = 1.15.

Clearly, the largest possible voltage magnitude that may be achieved using the space vector modulation strategy corresponds to the radius of the largest circle that can be inscribed within the hexagon of Figure 4. This circle is tangential to the midpoints of the lines connecting the ends of the active state vectors. The maximum fundamental phase voltage that may be achieved is:

$$V_{1,max} = \frac{V_{dc}}{\sqrt{3}} \quad (9)$$

which corresponds to a maximum modulation index $M_{max} \approx 1.15$. As a result, the maximum peak fundamental magnitude that may be obtained with the SVM technique is about 90.6 % of the inverter capacity. This represents a 15% increase in the maximum voltage compared with conventional sinusoidal modulation.

## 2.4 Determination of sector of operation

The first step in the space vector modulation algorithm is the determination of the sector in which the desired voltage space vector lies. Obviously this may be simply evaluated from the argument, $\xi$, of the reference vector given by:

$$\xi = \tan^{-1}\left(\frac{v^*_{s\beta}}{v^*_{s\alpha}}\right), \quad 0 \le \xi \le 2\pi \quad (10)$$

The sector, m, is then simply determined from:

University of Cape Town

$$m = \begin{cases} 1 & \text{if } 0 \le \xi \le \frac{\pi}{3} \\ \vdots & \vdots \\ 6 & \text{if } \frac{5\pi}{3} \le \xi \le 2\pi \end{cases} \quad (11)$$

However, this approach requires the computation of an **arc tangent** function in each modulation cycle. This can be a time consuming calculation or require a large look up table.

An alternative approach is possible by examination of the polarities and relative magnitudes of the real and imaginary parts of this reference vector. Using such logical determinations, it is possible to develop a simple algorithm to determine the sector of operation. This is outlined in the following pseudo-code:

```
if (v*sβ > 0) then " m = 1,2 or 3"
if (|v*sα| > √3 |v*sβ|) then " m = 1 or 3 "
    if (v*sα > 0) then
        m = 1
    else
        m = 3
else
    m = 2
else " m = 4,5 or 6"
if (|v*sα| > √3 |v*sβ|) then " m = 4 or 6 "
    if (v*sα > 0) then
        m = 4
    else
        m = 6
else
    m = 5
```

University of Cape Town

While this algorithm may seem more complicated than the first one, it can easily and quickly be evaluated by most modern processors using simple logical and conditional expressions.

## 2.5 Computation of active state times

Although often required to synthesise a circular locus, the space vector modulation strategy is equally applicable to any arbitrary voltage trajectory that lies within the inverter's capability. In general, the desired voltage space vector at any particular instant may be written in Cartesian co-ordinates as:

$$\bar{v}_s^* = v_{s\alpha}^* + jv_{s\beta}^* \quad (12)$$

Consider the example depicted in Figure 5, in which the desired voltage is found to lie in Sector 1. Although, the inverter cannot produce this voltage directly, it can be seen in the figure that it is possible to decompose it into two vectors, $v_x$ and $v_y$, that lie on the two active inverter vectors on either side of the reference vector. **Although it is not necessary to use the two adjacent inverter states in the synthesis**

**of the output voltage, it can be shown that superior harmonic performance is obtained when this condition is satisfied.**

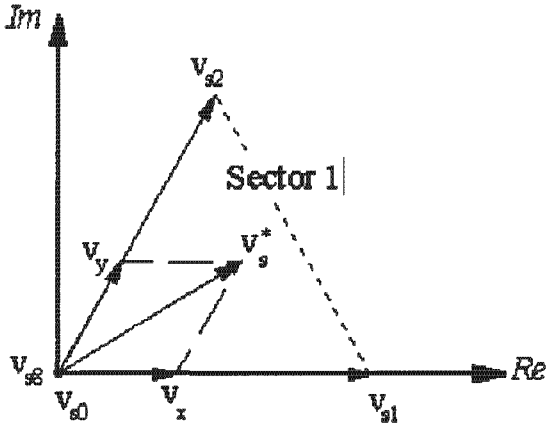



**Figure 5:** Synthesis of desired voltage space vector using realisable voltage vectors.

Therefore, in space vector notation:

$$\mathbf{v}_s^* = \mathbf{v}_x + \mathbf{v}_y \quad (13)$$

where the vectors, $v_x$ and $v_y$, are obtained by operating at the relevant inverter states, $v_{s1}$ and $v_{s2}$, for suitable portions of the switching period, $T_s$. In general, when operating in sector m, the reference vector may be decomposed according to:

$$\mathbf{v}_s^* = \frac{T_m}{T_s}\mathbf{v}_{s,m} + \frac{T_{m+1}}{T_s}\mathbf{v}_{s,m+1} \quad (14)$$

where $T_m$ and $T_{m+1}$ are the times spent at adjacent active inverter states, $v_{s,m}$ and $v_{s,m+1}$. Substituting in (14) for $\mathbf{v}_s^*$ from (12) and for $v_{s,m}$ and $v_{s,m+1}$ from (3) and considering real and imaginary parts, it is possible to write expressions for the active state times that are valid for any arbitrary voltage space vector and all possible sectors of operation. The relevant expressions are:

$$T_m = \frac{\sqrt{3}T_s}{V_{dc}}\left[v_{s\alpha}^* \sin\left(\frac{m\pi}{3}\right) - v_{s\beta}^* \cos\left(\frac{m\pi}{3}\right)\right]$$

$$T_{m+1} = \frac{\sqrt{3}T_s}{V_{dc}}\left[v_{s\beta}^* \cos\left(\frac{(m-1)\pi}{3}\right) - v_{s\alpha}^* \sin\left(\frac{(m-1)\pi}{3}\right)\right] \quad (15)$$

As the sine and cosine terms above assume only a limited number of values, these equations are particularly amenable to digital implementation. The remainder of the modulation cycle is subdivided between the zero states:

$$T_{zero} = T_0 + T_6 = T_s - T_m - T_{m+1}. \quad (16)$$

The way in which the zero states are utilised, in particular the apportionment of the time $T_{zero}$ between the two zero states, presents a degree of freedom to the process that may be used to change the characteristics of the modulation, as will be demonstrated later. However, equal portioning between the two zero states produces conventional SVM waveforms.

## 2.6 Production of inverter switching signals

University of Cape Town

Having computed the active and zero state times for a particular modulation cycle, it is possible to produce the switching signals, $S_A^*$, $S_B^*$, and $S_C^*$, to be applied to the inverter. Although, there are numerous possible sequences in which the various inverter states may be applied, **minimum inverter switching frequency is obtained if the transition from one inverter state to another is obtained by switching only one inverter pole**. In addition, since the zero states are common to all sectors, it proves convenient to begin and end each modulation cycle in one or other zero state. For convenience, the total zero time is most often divided equally between the two zero states. It is possible to satisfy all of the above restrictions by the use of symmetrical pulses as shown in Figure 6. The choice of which zero state to begin each cycle with is completely arbitrary. Here, the cycle begins in State 0, *i.e.* [000], with each inverter pole being successively toggled until state 8, [111], is obtained. The pattern is then reversed in order to complete the modulation cycle. This approach to the production of the inverter signals is by far the most prevalent due to its inherent harmonic advantages and this technique is termed **double edge space vector modulation**.

Figure 6 shows the times from the start of each modulation cycle at which the inverter poles are toggled, $T_{Aon}$, $T_{Bon}$ and $T_{Con}$, respectively. Taking the variations from one sector to another into consideration, it is possible to tabulate these times as functions of both the active and zero state times. The results are tabulated in Table 1. The time at which pole A is toggled back to its initial state is:

$$T_{Aoff} = T_s - T_{Aon} \quad (17)$$

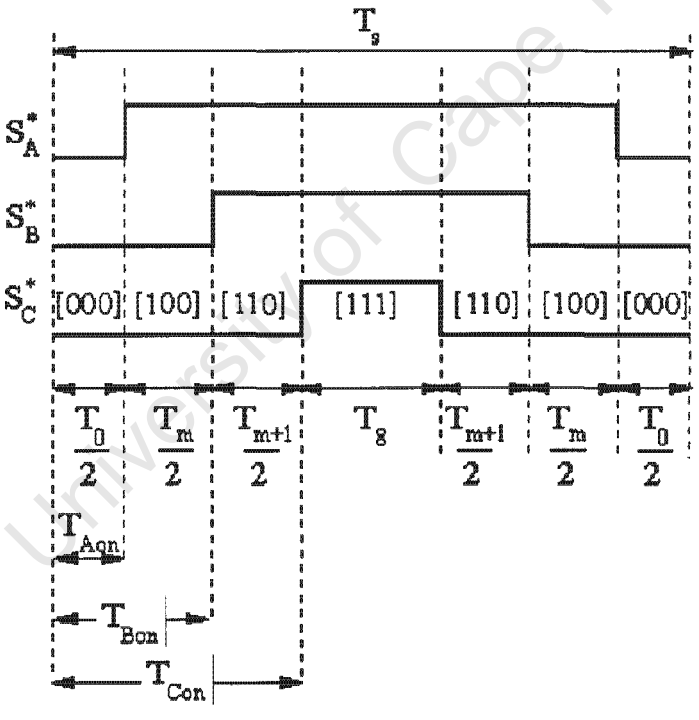



**Figure 6:** Typical inverter switching signals for double edge SVM in Sector 1.

| SECTOR | $T_{Aon}$ | $T_{Bon}$ | $T_{Con}$ |
|---|---|---|---|
| 1 | $\frac{T_0}{2}$ | $\frac{T_0}{2} + \frac{T_m}{2}$ | $\frac{T_0}{2} + \frac{T_m}{2} + \frac{T_{m+1}}{2}$ |
| 2 | $\frac{T_0}{2} + \frac{T_{m+1}}{2}$ | $\frac{T_0}{2}$ | $\frac{T_0}{2} + \frac{T_m}{2} + \frac{T_{m+1}}{2}$ |

University of Cape Town

| | | | |
|---|---|---|---|
| 3 | $\frac{T_0}{2}+\frac{T_m}{2}+\frac{T_{m+1}}{2}$ | $\frac{T_0}{2}$ | $\frac{T_0}{2}+\frac{T_m}{2}$ |
| 4 | $\frac{T_0}{2}+\frac{T_m}{2}+\frac{T_{m+1}}{2}$ | $\frac{T_0}{2}+\frac{T_{m+1}}{2}$ | $\frac{T_0}{2}$ |
| 5 | $\frac{T_0}{2}+\frac{T_m}{2}$ | $\frac{T_0}{2}+\frac{T_m}{2}+\frac{T_{m+1}}{2}$ | $\frac{T_0}{2}$ |
| 6 | $\frac{T_0}{2}$ | $\frac{T_0}{2}+\frac{T_m}{2}+\frac{T_{m+1}}{2}$ | $\frac{T_0}{2}+\frac{T_{m+1}}{2}$ |

**Table 1:** Switching times of the three inverter poles as a function of the sector of operation for double edge space vector modulation.

## 2.7 Limiting the applied voltage vector

If the space vector modulation algorithm is used inside a digital current control loop, a voltage vector that exceeds the inverter capability may be demanded. This is particularly true during a demanding torque transient, where an excessively large voltage vector may be demanded. It is necessary to detect and appropriately limit such a vector in a properly implemented space vector modulation algorithm. There are two basic limiting strategies. Either the voltage vector may be limited to the maximum circular locus of Figure 4 or to the hexagonal limit of the inverter capability.

### 2.7.1 Circular Limit

This method is applied if the magnitude of the reference voltage vector is found to exceed that of the maximum circular locus of Figure 4 so that:

$$\sqrt{\left(v^*_{s\alpha}\right)^2+\left(v^*_{s\beta}\right)^2} > V_{1,svm} \quad (18)$$

where V1,svm is given by (9). In this case a modified voltage vector:

$$v^*_s = v^*_{s\alpha} + jv^*_{s\beta} \quad (19)$$

is applied where:

$$v^*_{s\alpha} = \frac{V_{1,svm}}{\sqrt{\left(v^*_{s\alpha}\right)^2+\left(v^*_{s\beta}\right)^2}} \cdot v^*_{s\alpha}$$

$$v^*_{s\beta} = \frac{V_{1,svm}}{\sqrt{\left(v^*_{s\alpha}\right)^2+\left(v^*_{s\beta}\right)^2}} \cdot v^*_{s\beta} \quad (20)$$

This limiting method applies the largest voltage vector that is oriented in the same direction as the reference voltage vector and remains within the largest circular locus of Figure 4. This approach ensures continuous modulation so that the average line voltages remain sinusoidal at all times. However, the algorithm does require the calculation of a square root function and a division. In practice, if it is desired to eliminate pulse dropping, it may be necessary to limit the voltage vector to a somewhat smaller circular locus so that the pulse widths do not become excessively small.

### 2.7.2 Hexagonal Limit

Alternatively, it is possible to detect an unrealisable voltage vector if:

University of Cape Town

$$T_m + T_{m+1} > T_s \quad (21)$$

In this case the calculated times are rescaled as:

$$T_m'' = \frac{T_s}{T_m + T_{m+1}} . T_m$$
$$T_{m+1}'' = \frac{T_s}{T_m + T_{m+1}} . T_{m+1} \quad (22)$$

to produce a physically realisable voltage vector. The resultant vector is again oriented in the same direction as the reference voltage vector but now lies along the hexagonal inverter limit of Figure 4. This limiting algorithm is somewhat simpler to implement in a real-time microprocessor environment but the resultant line voltages can contain some low frequency distortion and pulse-dropping can occur. However, slightly larger voltage vectors can be obtained. The effective applied voltage vectors for the two different limiting strategies are shown in Figure 7.

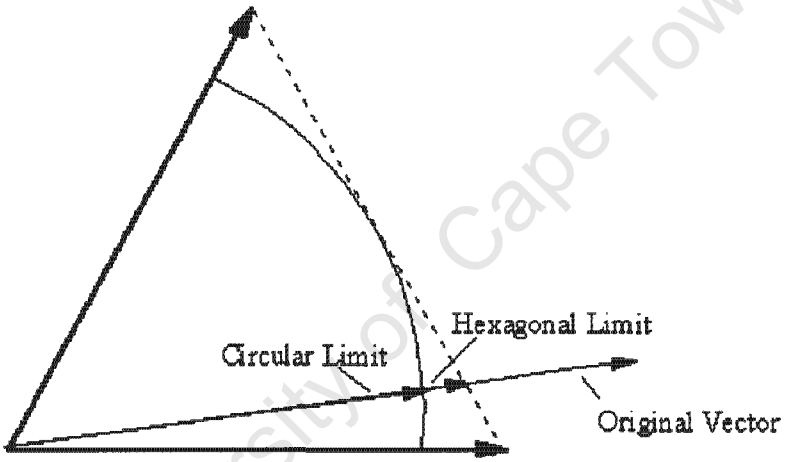



**Figure 7:** Resultant applied voltage vector for different limiting strategies.

University of Cape Town

## 2.8 Space vector modulating function

The space vector modulation process would seem to differ radically from the more conventional sinusoidal pulsewidth modulation schemes. However, the analysis and computational steps have been presented to illustrate its suitability to microprocessor implementation. However, it can be shown that space vector modulation is analogous to regular sampled pulsewidth modulation with a modified modulating waveform. The resultant modulating function, $v_{svm}$, is illustrated in Figure 8 and may be written:

$$v_{svm}(\omega_m t) = \frac{\sqrt{3}M}{2} \begin{cases} \text{Cos}\left(\omega_m t - \frac{\pi}{6}\right) & 0 \le \omega_m t < \frac{\pi}{3} \\ \sqrt{3}\,\text{Cos}(\omega_m t) & \frac{\pi}{3} \le \omega_m t < \frac{\pi}{2} \end{cases} \quad (23)$$

with half-wave and quarter-wave symmetry used to define the remainder of the waveform. This function also equals the averaged pole voltage normalised to $V_{dc}/2$. The maximum value of this function is 1.0 which occurs at $\omega_m t = \pi/6$ for the largest modulation index, M = 1.15.

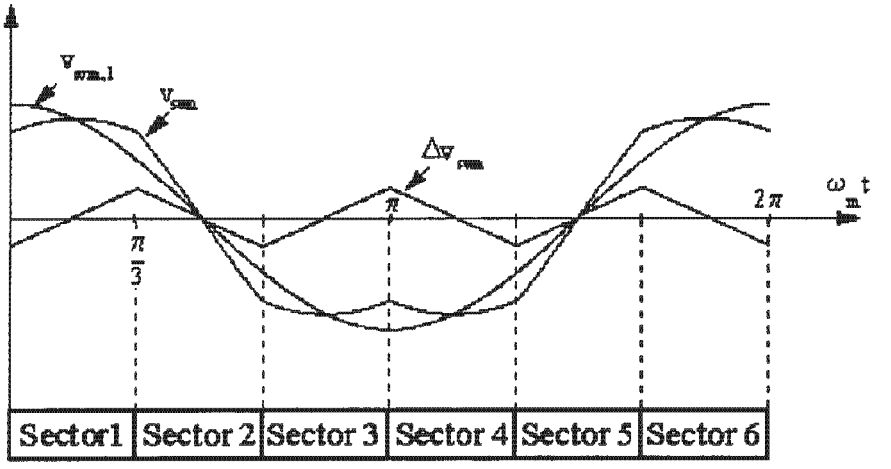



**Figure 8:** Normalised space vector modulating function and its constituent components; the desired fundamental, $v_{svm,1}$, and the added triplen waveform, $\Delta v_{svm}$.

As shown in Figure 8, the space vector modulating function consists of a desired sinusoidal fundamental component, $v_{svm,1}$, and a distorting triangular waveform, $\Delta v_{svm}$, at three times the fundamental frequency, so that:

$$v_{svm}(\omega_m t) = v_{svm,1}(\omega_m t) + \Delta v_{svm}(\omega_m t) \quad (24)$$

Using Fourier analysis, the harmonic series of the modulating function is:

$$v_{svm}(\omega_m t) = M\left\{\mathrm{Cos}(\omega_m t) - \frac{3\sqrt{3}}{4\pi}\sum_{r=0}^{\infty}\frac{1}{(3r+1)(3r+2)}\mathrm{Cos}\left\{3[2r+1]\omega_m t\right\}\right\} \quad (25)$$

The distorting waveform contains components at odd integer multiples of three times the fundamental frequency. **Since $\Delta v_{svm}$ consists of only triplen harmonics, these pole voltages form zero sequence terms that do not exist as phase voltages or produce currents in a three-phase star connected load provided there is no connection to the neutral point of the load.** Instead the triplen harmonic components of the pole voltage exist as a potential difference between the star point of the load and the fictitious mid-point of the dc supply. Indeed, their effect is identical to the introduction of third harmonic voltages common in sinusoidal modulation schemes. The triplen voltages act to flatten the peaks of the modulating function allowing the higher maximum voltages to be achieved.

## 2.9 Space vector modulation algorithm

The SVM algorithm can be illustrated in the flow diagram of Figure 9. In general the reference voltages are determined from the output of the digital current controller. The generation of the pulse sequences may be implemented without undue difficulty using the intrinsic PWM generation circuitry of modern microprocessors.

University of Cape Town

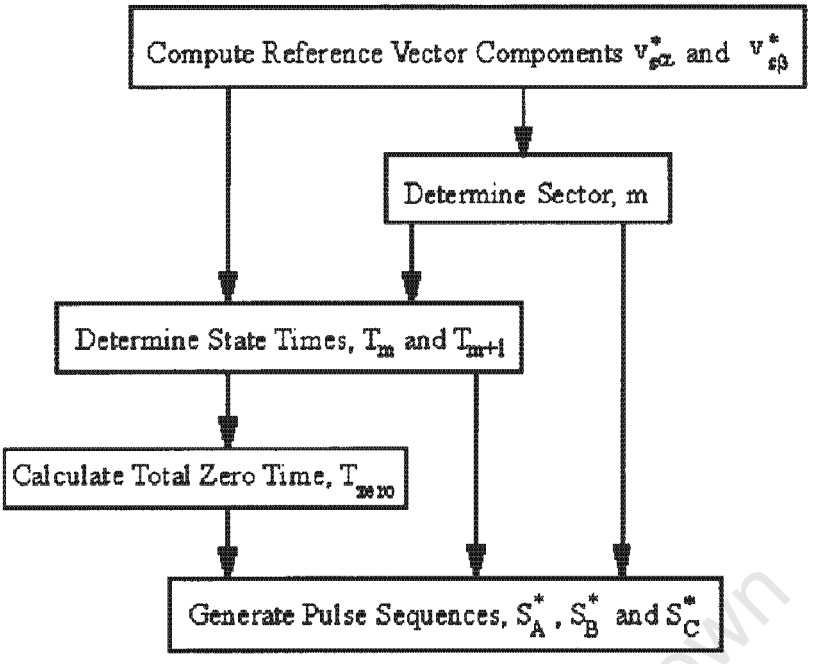



**Figure 9:** Space vector modulation algorithm

The exact implementation of this function differs between processors offering this facility. However, the net effect is the ability to generate accurate PWM signals of high resolution without processor intervention. The availability of this feature is a significant advantage when determining a suitable processor for the complete digital current control and modulation of ac drives. However, if this feature is not available in the chosen processors, additional digital hardware is necessary (perhaps in the form of a FPGA or PAL) to implement the same function. In either case it is essential that once the switching times have been computed by the control algorithm, there should be no processor overhead in the generation of the PWM sequences.

*Copyright 1995- 2001 Analog Devices, Inc. All rights reserved.*

University of Cape Town

# Bus Clamping forms of SVM

## 3.1 Introduction

The space vector modulation strategy presented in the previous sections may be viewed as the classical approach. However, many different variations of this techniques are possible. In the classical approach the total zero state time is divided equally between the two inverter zero states. However, since the effect of both zero states is identical, there is a degree of freedom in how each zero state is used in every modulation cycle. It is this degree of freedom that can be exploited to give alternative forms of the space vector modulation pulse sequences.

## 3.2 Bus clamped space vector modulation

In efforts to reduce the effective inverter switching frequency, various modulation strategies have been proposed that use only one zero state in each switching period. These techniques, although often reported under different titles, are denoted generically as **bus clamped modulation schemes**. All of the methods maintain one inverter pole clamped to either the positive or negative dc link for a number of modulation cycles while the other two poles are switched using space vector techniques. Alternative approaches exist depending on the amount of time each inverter pole remains clamped and to which of the dc link rails it is connected. However, three main classifications may be identified, 30° clamped, 60° clamped and 120° clamped, depending on the size of the contiguous clamped blocks of the corresponding modulating functions.

Consider the case where the desired voltage vector lies in Sector 1. This vector is synthesised in conventional space vector modulation using the two adjacent active states and both zero states. However, it is equally possible to generate the required voltage using the two active states and only one zero state. Consider first the case where State 8 or [111] is used in odd numbered sectors. As a result, this scheme is termed **odd 60° clamped space vector modulation**. The sector, inverter states used and clamped inverter device are tabulated in Table 1 for operation with this strategy. Clearly, for operation in Sector 1, inverter pole A does not switch and its upper switch remains on, connecting the pole to the positive dc rail over the entire 60° period.

| Angle | Sector | States used | Clamp Device |
|---|---|---|---|
| $0 \le \omega_m t \le \frac{\pi}{3}$ | 1 | [111],[110],[100] | Upper Switch A |
| $\frac{\pi}{3} \le \omega_m t \le \frac{2\pi}{3}$ | 2 | [000],[010],[110] | Lower Switch C |
| $\frac{2\pi}{3} \le \omega_m t \le \pi$ | 3 | [111],[011],[010] | Upper Switch B |
| $\pi \le \omega_m t \le \frac{4\pi}{3}$ | 4 | [000],[001],[011] | Lower Switch A |

University of Cape Town

| | | | |
|---|---|---|---|
| $\frac{4\pi}{3} \le \omega_m t \le \frac{5\pi}{3}$ | 5 | [111],[101],[001] | Upper Switch C |
| $\frac{5\pi}{3} \le \omega_m t \le 2\pi$ | 6 | [000],[100],[101] | Lower Switch B |

**Table 1:** Inverter states used and clamped device for operation with odd 60° clamped space vector modulation.

Typical inverter switching signals for operation with this strategy are depicted in Figure 1 for the case of the reference vector lying in Sector 1. Obviously, when operation transitions to Sector 2, the modulation cycle begins with the other zero state [000]. Therefore, it is necessary to toggle all three inverter states simultaneously when there is a transition into another sector.

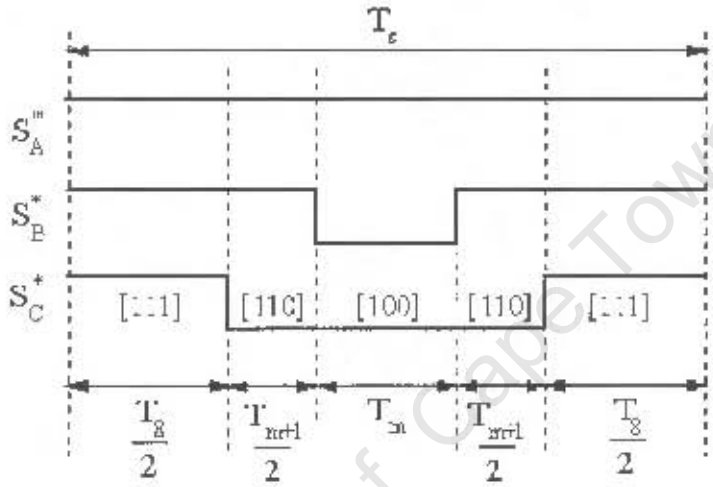



**Figure 1:** Inverter switching signals for operation in Sector 1 with odd 60° clamped SVM.

A variation on this technique results when the zero state [111] is used in even numbered sector. This method is termed **even 60° clamped space vector modulation**. Operation of the inverter with this scheme is described fully by Table 2 in which, for example, inverter pole A is now clamped to the positive dc rail during operation in Sector 6.

| Angle | Sector | States used | Clamp Device |
|---|---|---|---|
| $0 \le \omega_m t \le \frac{\pi}{3}$ | 1 | [000],[100],[110] | Lower Switch C |
| $\frac{\pi}{3} \le \omega_m t \le \frac{2\pi}{3}$ | 2 | [111],[110],[010] | Upper Switch B |
| $\frac{2\pi}{3} \le \omega_m t \le \pi$ | 3 | [000],[010],[011] | Lower Switch A |
| $\pi \le \omega_m t \le \frac{4\pi}{3}$ | 4 | [111],[011],[001] | Upper Switch C |
| $\frac{4\pi}{3} \le \omega_m t \le \frac{5\pi}{3}$ | 5 | [000],[001],[101] | Lower Switch B |
| $\frac{5\pi}{3} \le \omega_m t \le 2\pi$ | 6 | [111],[101],[100] | Upper Switch A |

**Table 2:** Inverter states and clamped device for operation for even 60° clamped SVM.

University of Cape Town

A third form of 60° bus clamped SVM is a combination of the above two methods. Instead of insisting that only one of the zero states be used to synthesise the voltage in any particular cycle, this technique chooses which zero state to use based on the position of the reference vector within the sector. If the reference vector is located in Sector 1 and lies between 0 and π /6, the odd clamped technique is chosen. However, when the vector lies between π /6 and π /3, the even clamped scheme is used. Here, this technique is termed **split clamped modulation**. The technique clamps inverter poles A, B and C to the positive dc rail in the 60° regions about vectors [100], [010], and [001] respectively and to the negative rail about [011], [101], and [110] respectively. A more complete picture of operation with this scheme may be obtained by combining the information of Tables 1 and 2.

An insightful way of comparing the three different 60° bus clamped space vector modulation methods is to look at the corresponding modulating functions. The modulating function for odd 60° clamped space vector modulation may be written:

$$v_{60°odd}(\omega_m t) = \begin{cases} 1 & 0 \le \omega_m t \le \frac{\pi}{3} \\ \sqrt{3}M\cos\left(\omega_m t - \frac{\pi}{6}\right) - 1 & \frac{\pi}{3} \le \omega_m t \le \frac{2\pi}{3} \\ \sqrt{3}M\cos\left(\omega_m t + \frac{\pi}{6}\right) + 1 & \frac{2\pi}{3} \le \omega_m t \le \pi \\ -1 & \pi \le \omega_m t \le \frac{4\pi}{3} \\ \sqrt{3}M\cos\left(\omega_m t - \frac{\pi}{6}\right) + 1 & \frac{4\pi}{3} \le \omega_m t \le \frac{5\pi}{3} \\ \sqrt{3}M\cos\left(\omega_m t + \frac{\pi}{6}\right) - 1 & \frac{5\pi}{3} \le \omega_m t \le 2\pi \end{cases} \quad (1)$$

The parallel expression for the even 60° clamped scheme is:

$$v_{60°even}(\omega_m t) = \begin{cases} \sqrt{3}M\cos\left(\omega_m t - \frac{\pi}{6}\right) - 1 & 0 \le \omega_m t \le \frac{\pi}{3} \\ \sqrt{3}M\cos\left(\omega_m t + \frac{\pi}{6}\right) + 1 & \frac{\pi}{3} \le \omega_m t \le \frac{2\pi}{3} \\ -1 & \frac{2\pi}{3} \le \omega_m t \le \pi \\ \sqrt{3}M\cos\left(\omega_m t - \frac{\pi}{6}\right) + 1 & \pi \le \omega_m t \le \frac{4\pi}{3} \\ \sqrt{3}M\cos\left(\omega_m t + \frac{\pi}{6}\right) - 1 & \frac{4\pi}{3} \le \omega_m t \le \frac{5\pi}{3} \\ 1 & \frac{5\pi}{3} \le \omega_m t \le 2\pi \end{cases} \quad (2)$$

The expression for the split clamped scheme is obtained by extracting the relevant parts of (1) and (2). The modulating functions for the three 60° bus clamped modulation schemes are shown in Figure 2 for M = 0.9. **Although, not obviously sinusoidal in shape, the distortion is made up entirely of triplen harmonics so that these modulation functions do produce the desired sinusoidal line-to-line voltage waveforms**. It should be noted that the form of these modulating functions can vary quite dramatically as the modulation index is changed. This phenomenon is illustrated in Figure 3 in which the split clamped modulating waveform is shown for three different modulation depths, M = 0.1, M = 0.5 and M = 1.15.

University of Cape Town

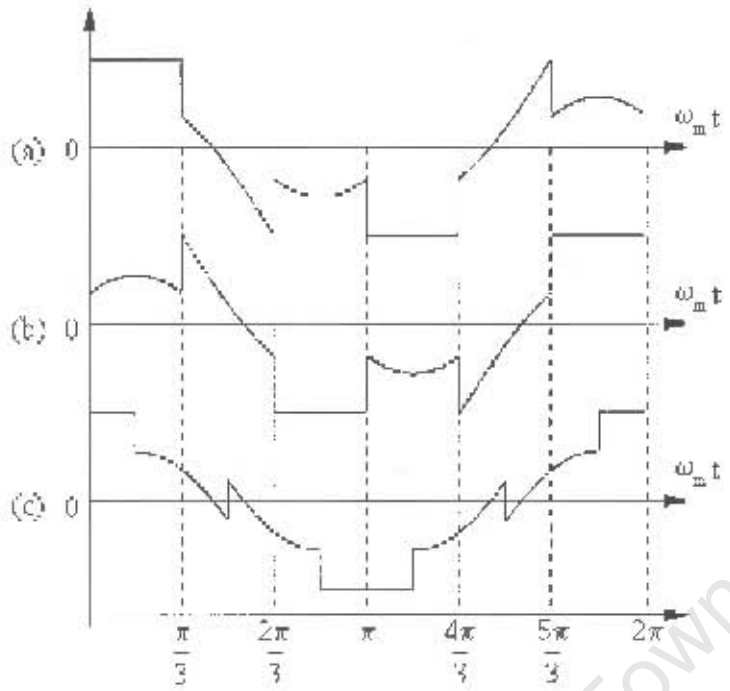



**Figure 2:** Effective modulating functions of 60° bus clamped space vector modulation with M = 0.9 (a) Odd clamped (b) Even clamped (c) Split clamped.

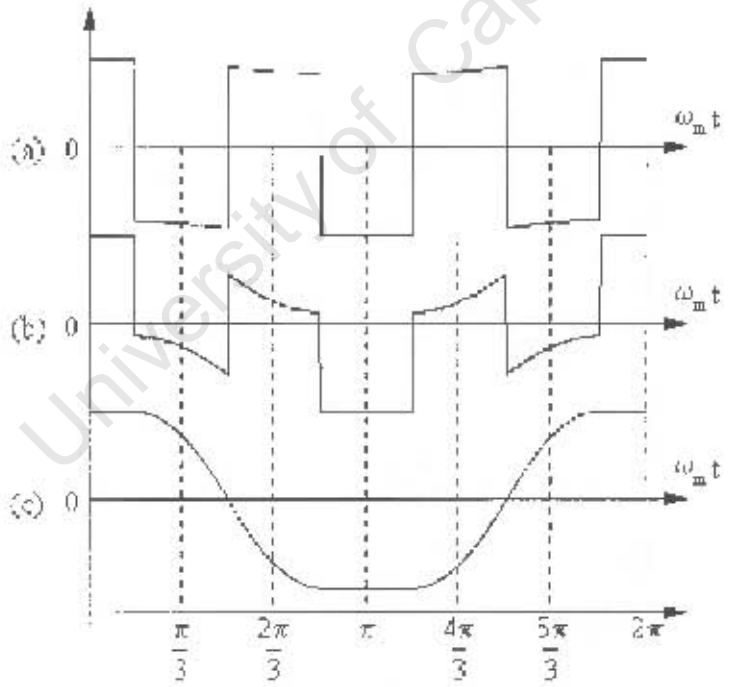



**Figure 3:** Variation of modulating function of split clamped space vector modulation for (a) M = 0.1 (b) M = 0.5 and (c) M = 1.15.

An alternative discontinuous modulation strategy results when the clamped 60° blocks of the modulating waveforms of Figure 3 are each replaced with two 30° intervals. Characteristic modulating waveforms for M = 0.5, 0.9 and 1.15 are shown in Figure 4.

University of Cape Town

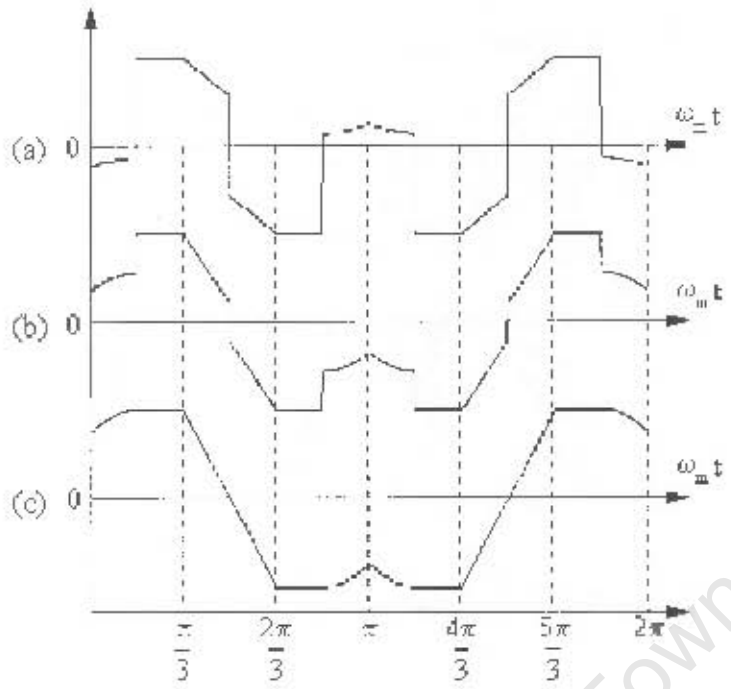



**Figure 4:** Characteristic modulating functions for 30° clamped space vector modulation with (a) M = 0.5 (b) M = 0.9 (c) M = 1.15.

This scheme is a variation of the odd and even 60° clamped strategies. In formulating the split clamped scheme, the modulating function is made up of alternate 30° segments of the basic odd and even clamped waveforms. If the order in which the two basic strategies are used is reversed, the result is a modulation method in which four bus clamping intervals of 30° appear. Therefore, operation in sector 1 follows the even clamped formulation in the region 0 to $\pi/6$ and then reverts to the odd clamped scheme from $\pi/6$ to $\pi/3$.

It is possible to have a modulating function where the total 120° clamping interval is applied as one block. Consequently, each inverter pole is clamped to either the positive or negative dc rail for two adjacent sectors. Two distinct possibilities exist: negative 120° bus clamping in which the state [111] is never used and positive 120° bus clamping where the state [000] is never applied. A complete representation of the operation of both schemes is given in Table 3. The corresponding modulating functions for both techniques are illustrated in Figure 5 for a modulation depth of M = 0.9.

| Positive 120° Bus Clamping | | | Negative 120° Bus Clamping |
|---|---|---|---|
| **Sectors** | **Clamped Device** | **Sectors** | **Clamped Device** |
| 6,1 | Upper Switch A | 1,2 | Lower Switch C |
| 2,3 | Upper Switch B | 3,4 | Lower Switch A |
| 4,5 | Upper Switch C | 5,6 | Lower Switch B |

**Table 3:** Inverter operation with positive and negative 120° clamped intervals.

University of Cape Town

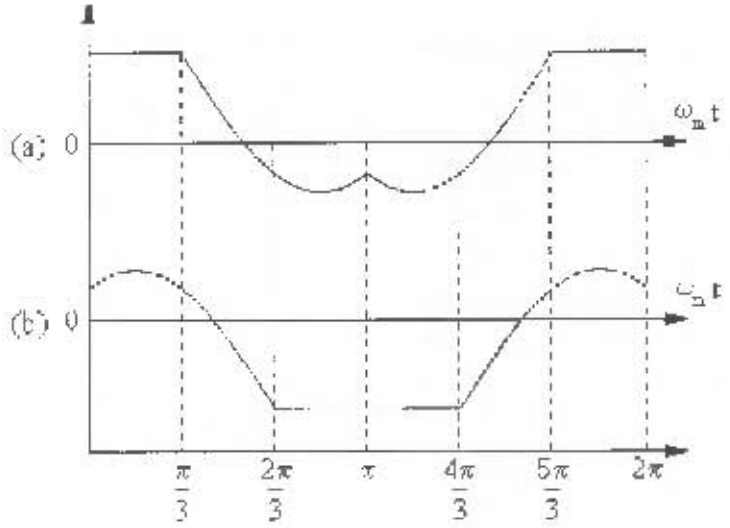



**Figure 5:** Modulating functions for 120° clamped space vector modulation with M = 0.9 (a) Positive clamped and (b) Negative clamped.

Modulating functions for both the positive and negative 120° bus clamped schemes are:

$$v_{120°,pos}(\omega_m t) = \begin{cases} 1 & -\frac{\pi}{3} \le \omega_m t \le \frac{\pi}{3} \\ \sqrt{3}M\,Cos\left(\omega_m t + \frac{\pi}{6}\right) + 1 & \frac{\pi}{3} \le \omega_m t \le \pi \\ \sqrt{3}M\,Cos\left(\omega_m t - \frac{\pi}{6}\right) + 1 & \pi \le \omega_m t \le \frac{5\pi}{3} \end{cases} \quad (3)$$

and:

$$v_{120°,neg}(\omega_m t) = \begin{cases} \sqrt{3}M\,Cos\left(\omega_m t - \frac{\pi}{6}\right) - 1 & 0 \le \omega_m t \le \frac{2\pi}{3} \\ -1 & \frac{2\pi}{3} \le \omega_m t \le \frac{4\pi}{3} \\ \sqrt{3}M\,Cos\left(\omega_m t - \frac{\pi}{6}\right) - 1 & \frac{4\pi}{3} \le \omega_m t < 2\pi \end{cases} \quad (4)$$

## 3.3 Comparison of various SVM techniques

1. All of the bus clamping schemes reduce the effective inverter switching frequency. In all of these schemes, each inverter pole is not switched during operation in two sectors. As a results the effective switching frequency is two thirds that of the conventional approaches.

2. Such a reduction in the switching frequency produces a corresponding reduction in the switching losses of the inverter. The reduction in the total switching loss is also a function of the motor current and consequently of the power factor of the load. The largest reduction is achieved when using the 60° bus clamped strategy where a reduction to 40% is achieved at unity power factor. This high reduction is preserved if the 60° non-switching blocks can be maintained in phase with the load current. Unfortunately, this may prove excessively difficult to implement in practice so that the switching losses increase with power factor. However, even at a power factor of zero, the 60° bus clamped scheme has only 66% of the switching losses of the more conventional space vector modulation schemes. The switching losses are reduced to about 53% at unity power factor

University of Cape Town

with the 120° bus clamping schemes. Similar to the 60° strategy this increases to 66% at zero power factor. In addition, the 30° bus clamped strategy provides a reduction in the switching losses to 66% regardless of power factor.

3. The total inverter power loss consists of conduction losses as well as the switching losses. The conduction losses can be the dominant component of the total power loss. The conduction losses are due to the finite on-state voltage drop of the inverter power devices, which can generally be divided into a constant and a current dependent term. It can be shown that the portion of the conduction loss due to the constant on-state voltage is not affected by the choice of modulation method. In addition, the portion due to the current dependent on-state losses is only slightly affected, so that the bus clamped modulation schemes present no significant advantage in relation to the total inverter conduction losses.

4. The inverter loss reduction of the bus clamping schemes is obtained only at the expense of increased harmonic distortion of the load currents. A fairer comparison of the different switching strategies should be made on the basis of identical switching losses. This improved harmonic performance of the bus clamped schemes is illustrated by comparing the rms harmonic currents of the different modulation schemes as a function of the modulation index. For the same calculation time, the conventional SVM scheme produces the lowest rms ripple current. However, if the switching frequency of the bus clamped schemes is increased by 50%, the bus clamped schemes produce the lowest ripple for M>0.7. Indeed, compromise modulation techniques have been reported in which the switching strategy transitions to a bus clamped approach at higher modulation depths.

5. At similar levels of total inverter switching losses, the dominant harmonic current components of the bus clamping schemes are located at frequencies approximately 50% higher than those of standard space vector modulation. Consequently, the audible noise level emanating from drives operating with a bus clamped modulation scheme may be significantly less than when operated with conventional space vector modulation.

6. One inherent disadvantage of the 120° strategies when compared to either the 30° or 60° schemes, is that the modulating waveforms are no longer symmetrical about zero. This is due to the effective addition of a constant dc offset to each modulating function. This symmetry about zero can be desirable in reducing the effects of any offset or quantization in the actual computation of the modulation process. As a result some undesirable imbalance can be introduced to the phase currents when this method is adopted.

7. Also, since each modulation cycle of the bus clamped strategy is defined by four, rather than six, switching instants, its implementation can be slightly simpler. However, the complexity of selecting the appropriate inverter states based on the location of the reference vector may, in certain implementations, outweigh this advantage. This added computational burden is particularly true of the 30° clamping scheme.

8. One attractive feature of all the space vector modulation strategies presented here is that the instantaneous line currents at the start of each cycle reflect the average value over the past cycle. This proves extremely advantageous in the development of digital current control schemes for ac drives In conventional analogue current controllers, the motor line currents are often low-pass filtered with a cut-off at approximately half the inverter switching frequency in order to remove the ripple current component. Such filtering introduces an unavoidable phase shift into the current control loops and reduces the system stability limits. However, in a digital environment, there is no need to filter the currents provided they are sampled at the beginning of each period. Consequently there is no phase shift and the controller gains may be increased giving improved dynamic performance.

*Copyright 1995- 2001 Analog Devices, Inc. All rights reserved.*

University of Cape Town

# Tech Note #T1 - The Different Types of UPS Systems

APC

- Set your screen size to 800x600 for optimal viewing -

## Introduction To The UPS

This Technical Note explains the different types of UPS systems and their characteristics. Block diagrams are provided.

## Background Of The UPS

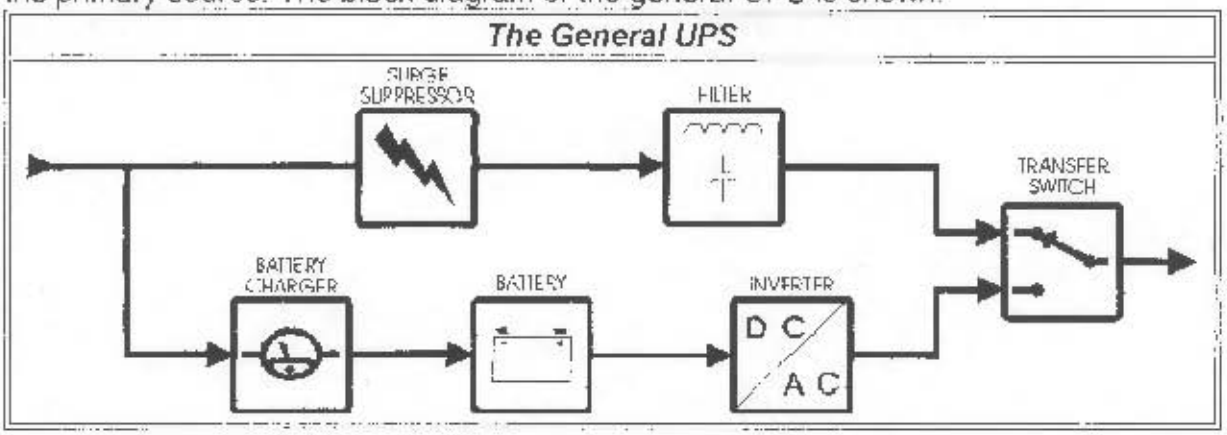
It is widely believed that there are only two types of UPS systems, namely standby type UPS and on-line type UPS. These two terms are, as commonly understood, not correctly applied to many UPS systems on the market today. Many misunderstandings about UPS systems are cleared up when the different types of UPS topologies are properly identified.

## Classical UPS Definitions

UPS systems are intended to improve the quality of AC power in order to provide uninterrupted operation of AC powered equipment. To accomplish this function, a UPS takes in normal quality utility AC power and provides two enhancements:

- Power Quality Improvement.
- A Redundant (Back Up) Power Source.

Power quality defects which may be improved by the UPS include surges, noise, or sags. A UPS system provides redundant power by supplying the load with a primary power source and then providing a back up power source in the case of the failure of the primary source. The block diagram of the general UPS is shown:

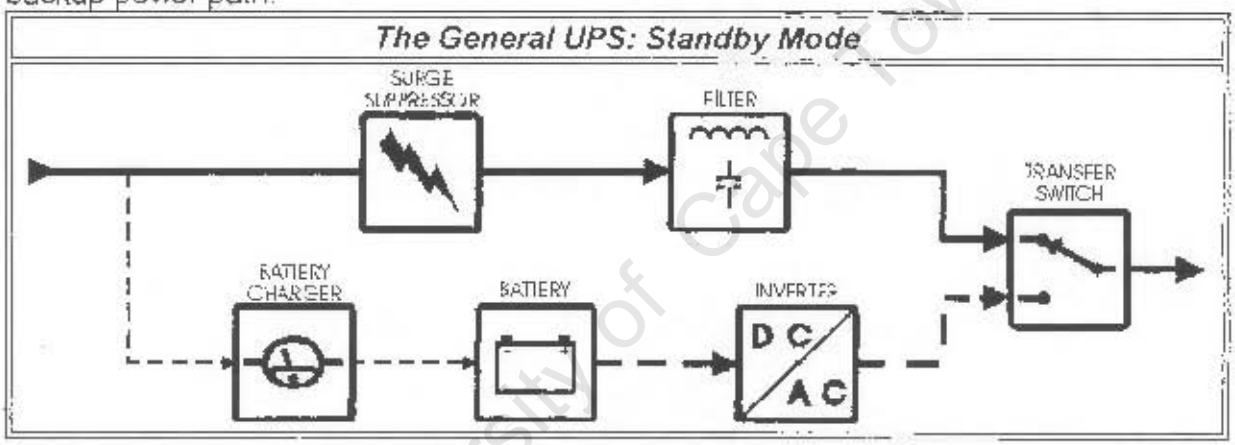


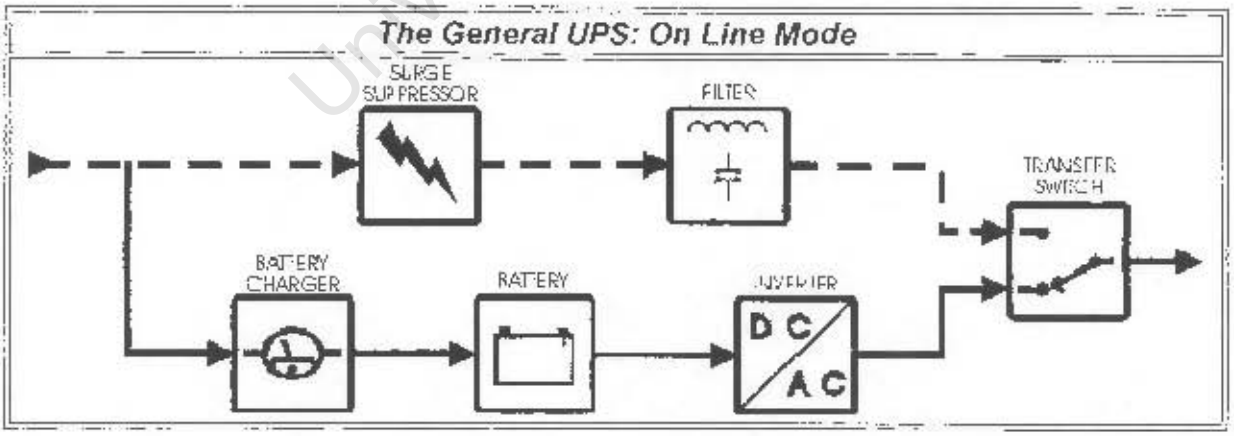

The general UPS may be operated as either a standby type UPS or an on-line type UPS. The main difference is which power path is chosen to be the primary power path.

University of Cape Town

In the diagrams, the solid power path is the primary power path, and the dashed power path is the backup power path. For standby UPS operation, the transfer switch is set to choose the filtered AC input as the primary power source, and switches to the battery / inverter as the backup source in case of the failure of the primary source (AC). For on line operation, the transfer switch is set to choose the battery / inverter as the primary source, and switches to the input AC as the backup source in case of the failure of the primary source (battery / inverter). This distinction between on line and standby UPS operation is very simple, but it gives rise to some important differences in operation. One interesting difference between standby and on line operation of the general UPS is the operation during an input AC power failure. In the case of standby UPS operation, the transfer switch must operate to switch over to the battery / inverter backup power source. However, in the case of on line operation, failure of the input AC does not cause activation of the transfer switch, because the input AC is NOT the primary source, but is rather the backup source. Therefore, during an input AC power failure, on line operation results in no transfer time.

In the figures below, the solid line is the primary power path, and the dotted line is the backup power path.




University of Cape Town

The on line mode of operation exhibits a transfer time when the power from the primary battery charger / battery / inverter power path fails. This can occur when any of the blocks in this power path fail. The power can also drop out briefly, causing a transfer, if the inverter is subjected to sudden changes in the load, or if the inverter experiences an internal control "glitch". Actual on line UPS systems do exhibit a transfer time, and in actual installations may transfer as frequently as standby type UPS systems: however

on line UPS transfers are not related to AC input power failures as they are in a standby UPS.

The size of the battery charger is greatly affected by the choice of standby vs. on line operation of the general UPS. When used in the on line mode, the battery charger must be large enough to handle all of the output power in order to prevent the battery from discharging. When used in the standby mode, the battery need only supply the comparatively small battery recharging power.

The heat generated by the UPS is much larger when the general UPS is operated in the on-line mode. The flow of power through the battery charger and inverter causes a power loss of 25 to 30 percent. This power loss generates heat which shortens the lifetime of the electrical components in the UPS and drastically reduces the life of the battery (the negative effect on battery life is eliminated if the batteries are in a separate cabinet). When operated in the standby mode, the power loss of the filter and surge suppresser are an insignificant 1 to 2 percent. Over the lifetime of the UPS, the cost of the extra wasted electricity required when the UPS is operated in the on-line mode will be a significant fraction of the original cost of the UPS itself.

The only UPS systems that exhibit the classical on-line UPS topology are high power designs of over 10kVA. Examples of units which follow the classical standby approach include the APC Back-UPS, and TrippeLite models.

## The Other UPS Topologies

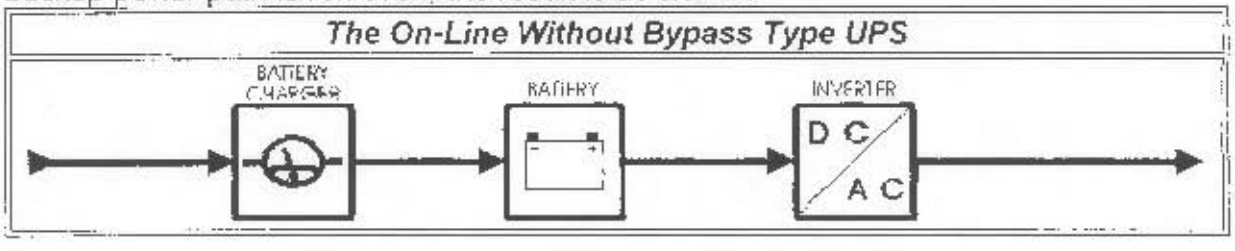
The vast majority of UPS systems available today are not of the classical standby or on-line type. These UPS systems use a variety of different approaches which include:

- On-Line Without Bypass
- Standby On-Line Hybrid
- Standby-Ferro
- Line-Interactive

Many of these UPS types are improperly classified as either a standby or on-line design. Such improper classification leads to mistaken beliefs on the part of users regarding the levels of protection provided by the UPS.

## The On-Line Without Bypass Topology

In this topology, the general UPS is set up to operate in the on-line mode but the entire backup power path is removed; the result is as shown:



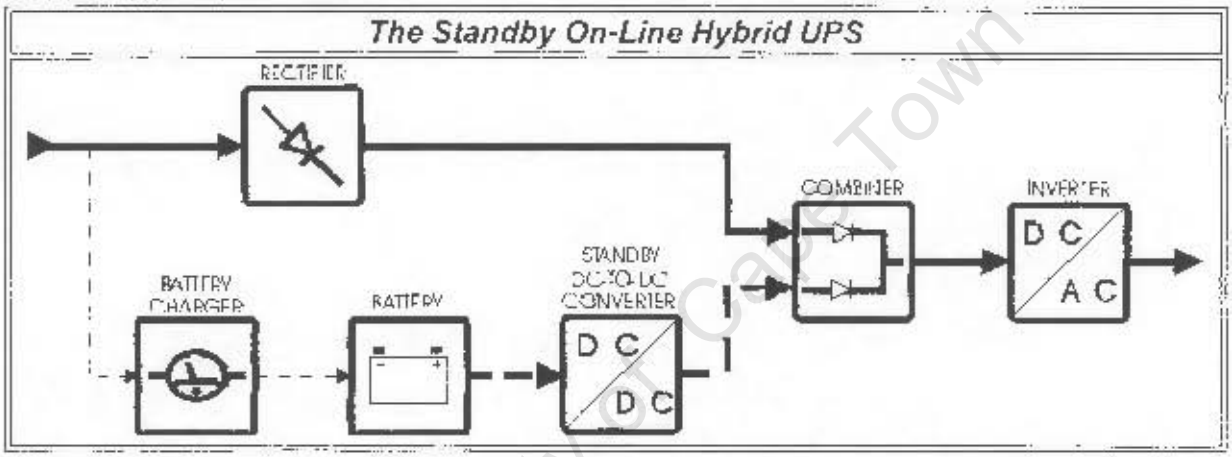

When an on-line type UPS does not have a bypass, the UPS does not provide a backup power source in case of the failure or glitch of the primary source (inverter). Therefore,

University of Cape Town

one important characteristic of a UPS, namely redundancy, is not achieved by this UPS type. This UPS does not exhibit a transfer time during a power failure and is for that reason frequently portrayed as an on-line UPS. Large UPS systems for minicomputers and mainframes are never of this type, but the lack of the back up power path (often referred to as a "bypass") is not frequently recognized in the less experienced PC marketplace and therefore this type of UPS is sometimes sold. The standby/on-line hybrid design is a derivation of this design.

## The Standby On-Line Hybrid Topology

The standby/on-line hybrid UPS is a modification of the "on-line without bypass" design. In this case, the battery charger and battery connections are modified, and a standby DC/DC converter is added:

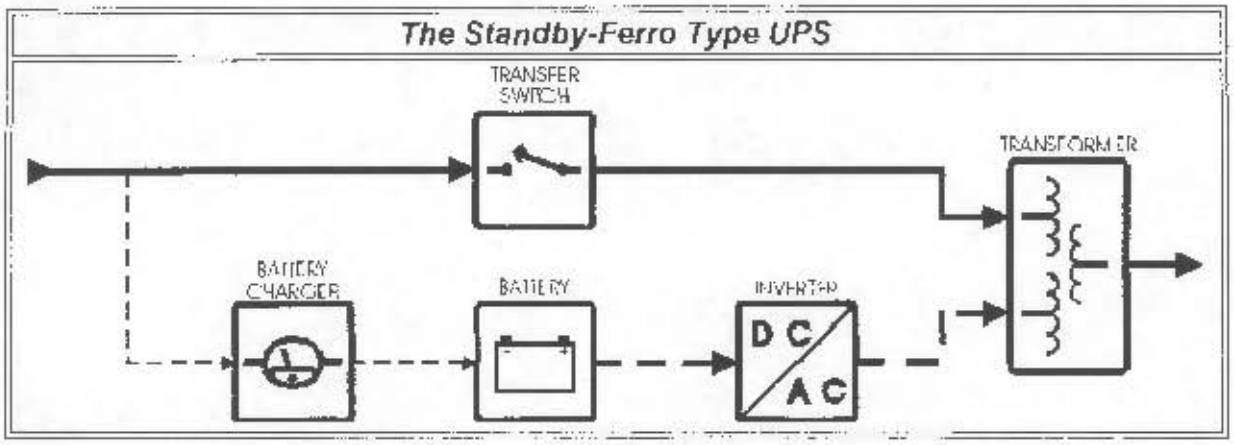



University of Cape Town

The standby converter from the battery is switched on when an AC power failure is detected, just like in a standby UPS. The battery charger is small, just like in a standby type UPS. This UPS will exhibit no transfer time during an AC power failure. However, like the "on-line without bypass" type of UPS, this unit has an inverter which is a possible single point failure for which there is no backup power path. The most misunderstood part about this topology is the belief that the primary power path is always "on-line" when in fact, the power path from the battery to the output is only half "on-line" (the inverter), while the other half (the dc-dc converter) is operated in the standby mode. Note that in this design, unlike either the classical standby or on-line designs, there is no backup power path provided in the case of the failure of the primary power
path. This topology is used in UPS systems such as the Unison Unipower, and Exide Personal Powerware.

## The Standby-Ferro Topology

This design depends on a special transformer that has three windings (power connections). The primary power path is from AC input, through a transfer switch, through the transformer, and to the output. In the case of a power failure, the transfer switch is opened, and the inverter picks up the output load.



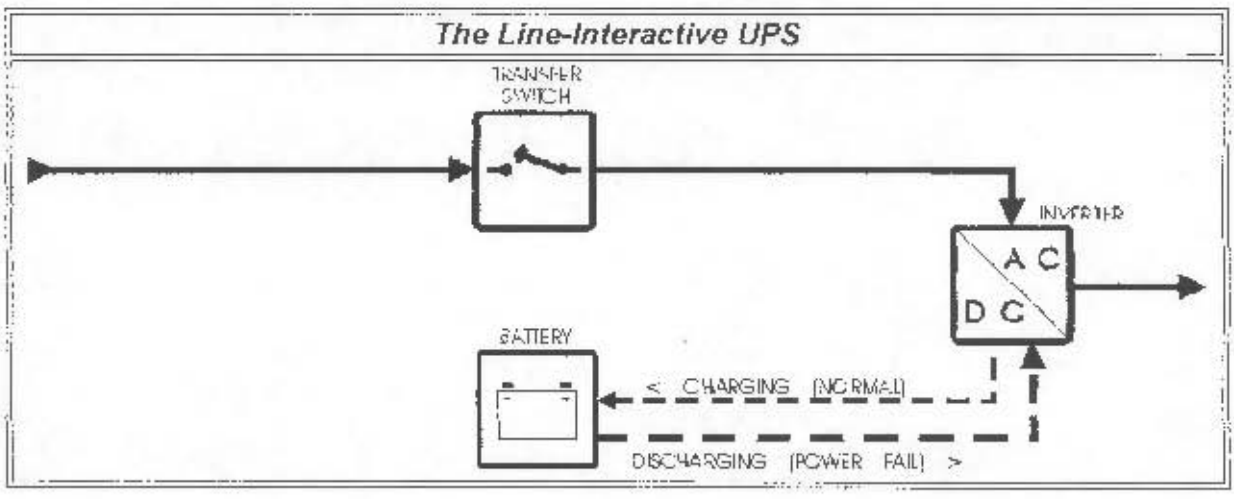

In the Standby-Ferro design, the inverter is in the standby mode, and is energized when the input power fails and the transfer switch is opened. The transformer has a special "Ferro-resonant" capability which provides limited regulation and output waveform "shaping". The isolation from AC power transients provided by the ferro transformer is as good or better than any filter available, but the ferro transformer itself creates severe output voltage distortion and transients which can be worse than a poor AC connection (see APC Technical Note T12). Even though it is inherently a standby UPS, the Standby-Ferro generates a great deal of heat because the Ferro-resonant transformer is inherently inefficient. The best known example of this type of UPS is the BEST Ferrups.

Standby-Ferro UPS systems are frequently represented as on-line units, even though they have a transfer switch, the inverter operates in the standby mode. and they exhibit a transfer characteristic during an AC power failure.

University of Cape Town

## The Line-Interactive Topology

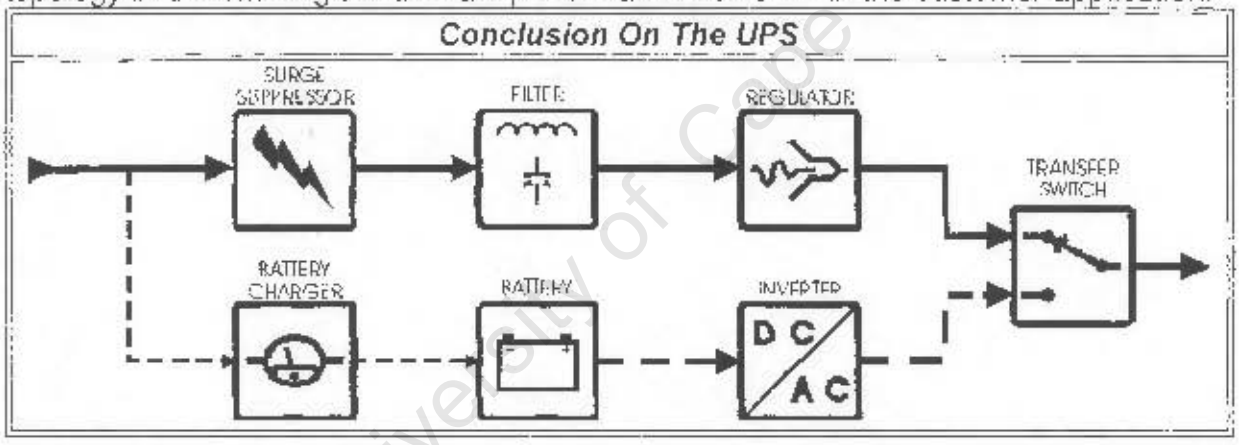
In this design, the battery-to-ac power converter (inverter) is always connected to the output of the UPS. Battery charging is provided by operating the inverter in reverse during times when the input AC power is normal. When the input power fails the transfer switch opens and the power flow is from battery to the UPS output. The fact that the inverter is always connected to the output provides additional filtering and reduced switching transients when compared with the standby type UPS.



The inverter also provides regulation, operating to correct brownout conditions which would otherwise force the UPS to switch to battery operation. This allows the UPS to operate at sites with very poor power. The inverter can be designed such that it's failure will still permit power flow from the AC input to the output, which eliminates the potential of single point failure and effectively provides for two independent power paths. This topology is inherently very efficient which leads to high reliability while at the same time providing superior power protection.
The APC Smart-UPS is of this type. Other line interactive designs, like the Detec PowerRite and BEST Fortress, are similar except that the inverter is not running backwards during charging. This has the disadvantage that the inverter is in the standby mode before a power failure.

## Conclusion On The UPS

The commonly used terms "on-line" and "standby" do not correctly describe the majority of UPS systems available. There are significant differences in UPS topologies between available products on the market, with theoretical advantages for different approaches. Nevertheless, the basic quality of design and construction is more dominant than topology in determining the ultimate performance achieved in the customer application.




**If you have further questions, please write back to apcinfo@apccorp.apcc.com.**

• All Rights Reserved. • All original web site material: animations, graphics, logos, photographs, text, HTML, JavaScript, CGI, META, and all Promotional Codings are Copyrighted © 1997 by J. Miller.

University of Cape Town