THE UNIVERSITY OF ZAMBIA

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

A VARIABLE-SPEED DRIVE FOR LOCAL MANUFACTURE

by

HAILE-SELASSIE RAJAMANI

" Report submitted in fulfillment of the requirements for the Degree of Master of Engineering, University of Zambia"

February 1992

Declaration

I Haile - Selassie Rajamani do hereby solemnly declare that this dissertation represents my own work and that it has not been submitted for a degree at the University of Zambia or any other University.

Signature HSRajamani

Date 27-2-92

Acknowledgements

I would like to thank Prof. J.M. Mwenechanya and Prof. D. Whittaker for their supervision. Thanks also to all the members of staff that assisted me, especially Mr. C. Mubemba, Dr. J.S.J. Daka and Mr. Kelvin Maliti.

In addition, the assistance obtained through the UNZA/UMIST/UCS and the SIDA link schemes is acknowledged. The provision of a research grant from the University of Zambia and the facilities of the University are also acknowledged.

Abstract

Recent developments in the design of variable-speed drives have been aimed at the control of a standard squirrel cage induction motor because of its many attractive features including robustness, reliability and low maintenance requirements. This dissertation demonstrates the design of a drive package that meets specified requirements, while being sufficiently simple and economic to be attractive to local industry. The drive is designed for applications that do not require very precise speed control and fast dynamic response, such as water pumping and food processing.

In Chapter 1 an overview of variable speed drives is given. Chapter 2 describes the open loop control strategy used. Constant torque capability and acceleration of the motor are studied, illustrating the advantage of variable speed drive. A review of Pulse Width Modulation (PWM) techniques and their implementation is given in Chapter 3. Chapter 4 presents a study of asynchronous PWM technique using a high carrier frequency with particular reference to voltage utilization and harmonic distortion. An optimum waveform is suggested that ideally provides a pure sinusoidal output current. The effect of dead-time and harmonic losses are also considered.

Chapter 5 describes the inverter circuit built with MOSFET's. The following chapter describes the implementation of the control technique using a specialized integrated circuit, the MA818, and a computer. In Chapter 7 the implementation and testing of the drive system is detailed and the results presented. The optimum waveform suggested is shown to give good utilization of rail voltage. The current distortion is also shown to be low.

In Chapter 8, the practicability of the simple design is discussed. The final drive unit is shown to be very simple consisting of the inverter circuit, the MA818, a micro-controller chip and a low voltage power supply. The inverter circuit is also a simple design that avoids the use of transformers and opto-couplers. Possible modifications and suggestions for future work are given.

Table of Contents

List of Figures

List of Principal Symbols

f	Frequency (Hz)
f_s	Switching frequency (Hz)
f_c	Carrier frequency (Hz)
I_i	Current (A)
I_k	k^{th} harmonic current (A)
IHD	Current harmonic distortion (%)
J	Inertia
k	Order of harmonic
L_i	Inductance (H)
M	Modulation index or depth
m	Number of phases
P	Power (W)
p	Number of pole pairs
Q_i	Charge (C)
R	Resistance (Ω)
$R_{\phi sa}$, $R_{\phi jc}$, $R_{\phi cs}$	Thermal resistances (K/W)
s	Slip (p.u)
T_i	Torque (Nm)
T_j , T_a	Junction and Ambient temperatures.(K)
t	Time (s)
THD	Total harmonic distortion (%)
V_k	K^{th} harmonic voltage (V)
V_i	Voltage (V)
ΔV	Ripple voltage (V)

W	Energy (J)
X	Leakage reactances (Ω)
Z_i	Impedances (Ω)
ω	Angular velocity (rad/s)
ϕ_i	angle between i^{th} harmonic flux and fundamental.
σ	Distortion factor (%)
α_i	i^{th} switching angle (rad)
Φ	fundamental flux (p.u)

Chapter 1

Introduction

Electric motors have wide application in the Zambian industry and some of these applications would benefit from variation of the motor torque-speed characteristic. Recent developments in the design of variable-speed drives have been aimed at the control of a standard squirrel cage induction motor because of its many attractive features. Due to its simple construction, it is also feasible to envisage local production of the motor in the near future. However, the variable-speed drives existing in Zambia are imported and are generally application-specific and complex. Therefore, a suitable design of **a variable speed drive for local manufacture** will be needed in the near future.

In this research, the electrical engineering aspects of the design were of key academic interest whilst the equally important aspects of market considerations and manufacture process, that are essential to "local manufacture", were not covered due to limitations of time. The control of the motor using a high carrier frequency Pulse Width Modulation technique was studied in detail because of its many benefits. The design of an inverter was also done to give completeness to the drive design but was not studied in detail due to the many aspects that would have had to be covered. The tests, that were carried out, were primarily aimed at confirming some of the theoretical aspects of the PWM strategy used as well as to demonstrate a working first prototype. As a starting point, an overview of variable speed drive is first given in this chapter.

1.1 Users of Mechanical Power

The selection of the type of motor, its rating, and the form of its control depend on the requirements of the particular application, and details of the load torque-speed characteristics are basic information. The wide variety of the requirements of some typical mechanical loads are shown by the examples of torque-speed characteristics in figure 1.1. If the mechanical conditions change, then for each example "families" of curves may be drawn.

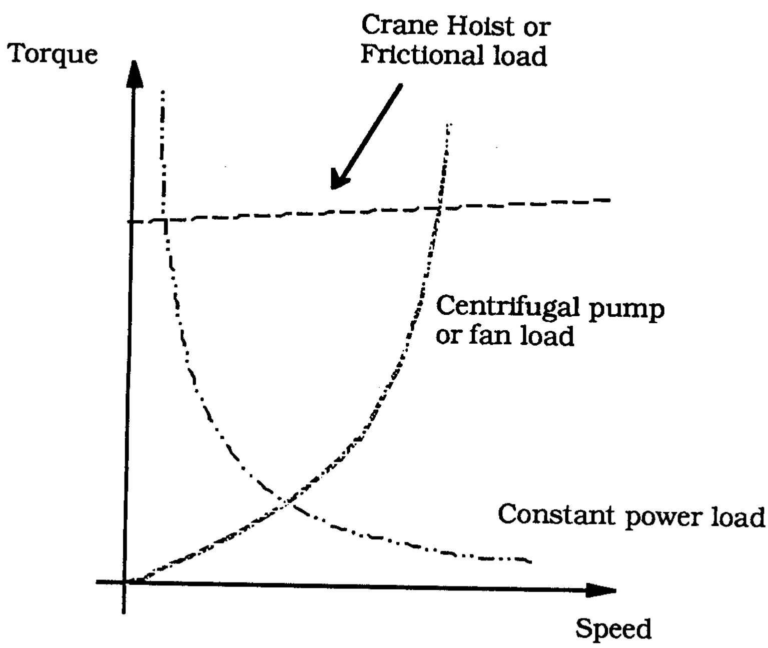


Fig 1.1 Load torque-speed Characteristics

An electric drive connected to a mechanical load comprises a system with various steady-state and transient operating features.

The steady-state operating conditions may be obtained by the intersection of the torque-speed characteristics of the motor and its load, a typical case of an induction motor with a fan load being shown in figure 1.2. The operating point and any shift, depend on the shapes of the torque-speed characteristics of both the motor and the load.

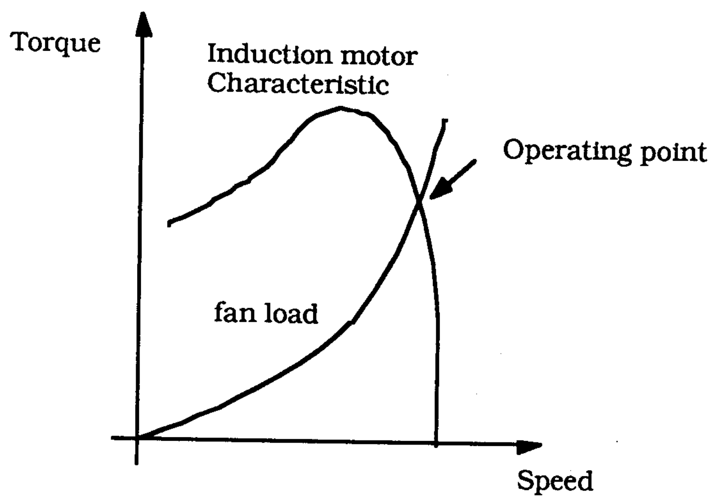


Fig 1.2 Interaction between motor and load

1.2. Motor Torque-Speed Characteristic

1.2.1. d.c. Motors

The inherent torque-speed characteristics of various d.c. motors are shown in figure 1.3. Families of these basic characteristic may be drawn to represent different electrical operating conditions. The characteristics are further modified by the type of control used. It is this flexibility that makes the d.c. motor suitable for variable speed control.

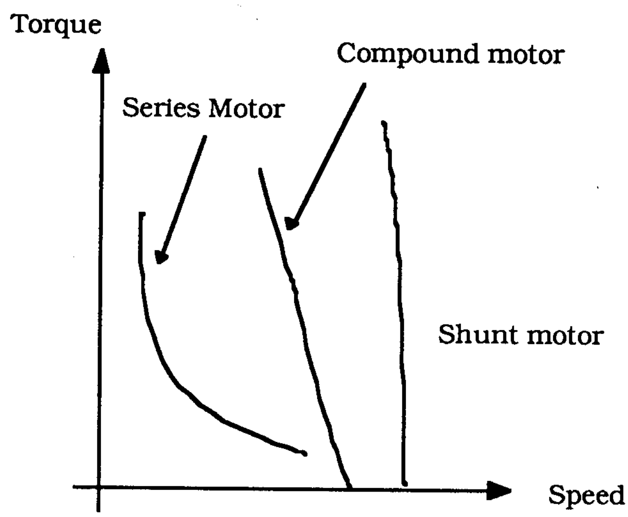


Fig 1.3 Torque-speed characteristics of d.c. motors

1.2.2. **Synchronous Motor**

The torque-speed characteristic of a synchronous motor is a straight line as shown in figure 1.4. The speed n is exactly the synchronous speed n_S, given by

$$n = n_S = \frac{f}{p} \qquad (1.1)$$

where f = motor supply frequency, p = pole pairs.

Smooth variations in the speed may be achieved by varying the motor supply frequency.

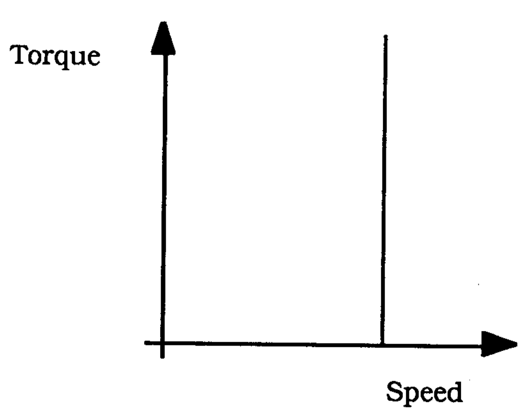


Fig 1.4 Torque-Speed characteristic of Synchronous motor

1.2.3. Induction Motor

The inherent torque-speed characteristic of an induction motor is shown in figure 1.5. Similar curves are obtained for different stator frequencies, voltages and pole pairs. The type of control also modifies the characteristics of the motor. The speed of the motor is conveniently given in terms of slip s, which is defined as the fractional drop of speed from synchronous speed n_s.

$$\text{Thus } s = \frac{n_s - n}{n_s} \text{, and so} \qquad n = \frac{f}{p}(1 - s) \qquad (1.1)$$

It can be seen that step changes in speed can be achieved by changing the the number of pole pairs p, but this can only be done in discrete steps. Another way of changing the speed is by variation of slip but there is a waste of energy unless methods to recover slip energy are used. The method becoming increasingly popular, due to developments in power electronics, is where the motor supply frequency is varied.

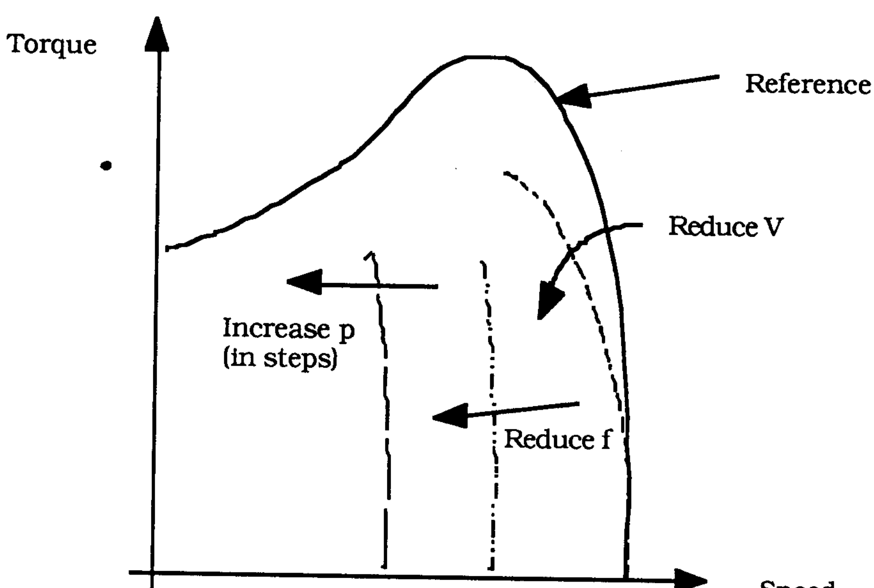


Fig 1.5 Torque-speed characteristics for induction motors

1.3 Variable Frequency Supply for an Induction Motor

A variable frequency supply for the motor may be derived from the constant mains supply frequency by (a) rectifying to d.c., followed by an inverter triggered at the new frequency, or by (b) switching a polyphase supply in such a way as to obtain the desired frequency output in a "cycloconverter".

1.3.1 Cycloconverter

In a cycloconverter the supply voltage is converted to the load frequency directly without any intermediate stage. The basic configuration of a cycloconverter is shown in figure 1.6. Each box is a three phase half bridge thyristor unit. By appropriate firing of the thyristors a new frequency is obtained that is lower than the supply frequency as shown in figure 1.7. However, the output has high harmonic content and the converter is mainly used in large drives.

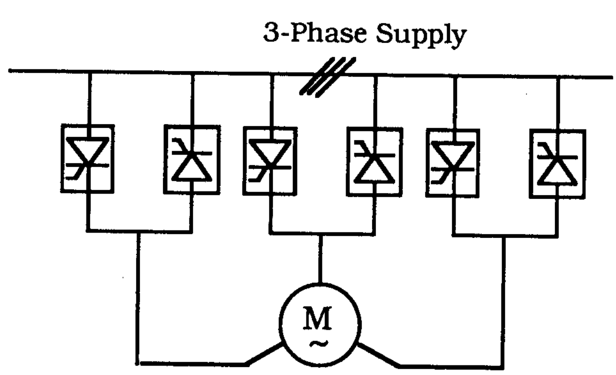


Fig 1.6 Schematic of the Cycloconverter

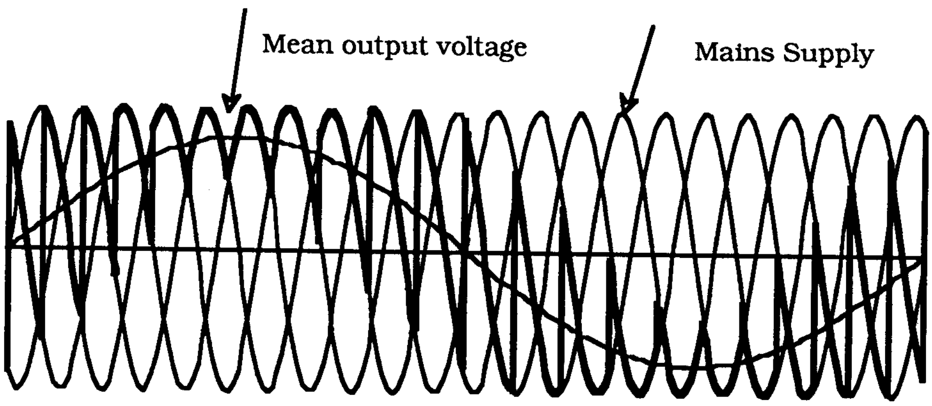


Fig 1.7. Voltage waveforms of Cycloconverter

1.3.2 Rectifier / Inverter

1.3.2.1 Voltage Source Inverter

The six-step voltage source inverter has a controlled three phase ac to dc converter that is used to vary the d.c. link voltage. The inverter circuit is used to invert the voltage to the new frequency.

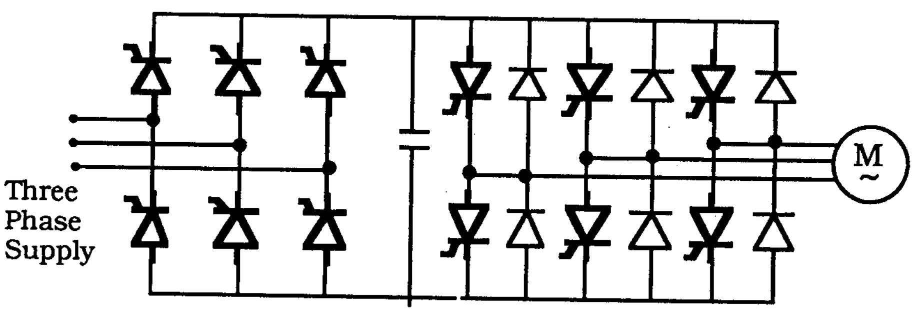
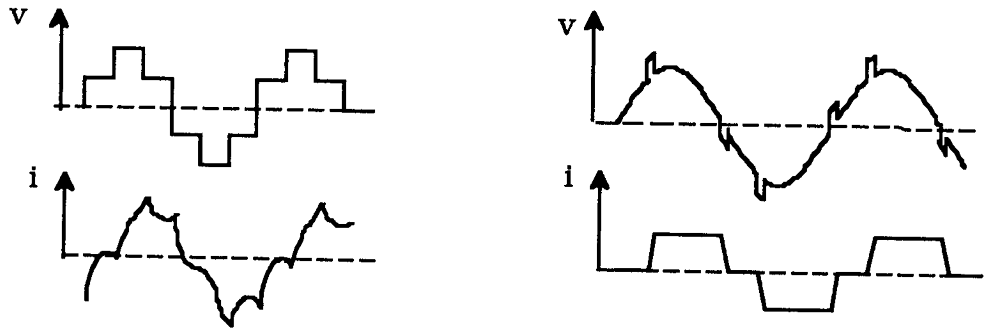


Fig 1.8 The six-step inverter

The voltage and current waveforms associated with the six-step inverter are shown in figure 1.9 (a).



(a) Voltage source inverter

(b) Current source inverter

Fig 1.9 Waveforms for Voltage and Current source inverters for one phase of a star-connected motor

1.3.2.2 Current Source Inverter

The current source inverter has a constant current supply and usually has a large d.c. link inductor. The corresponding voltage and current waveforms are shown in figure 1.9 (b).

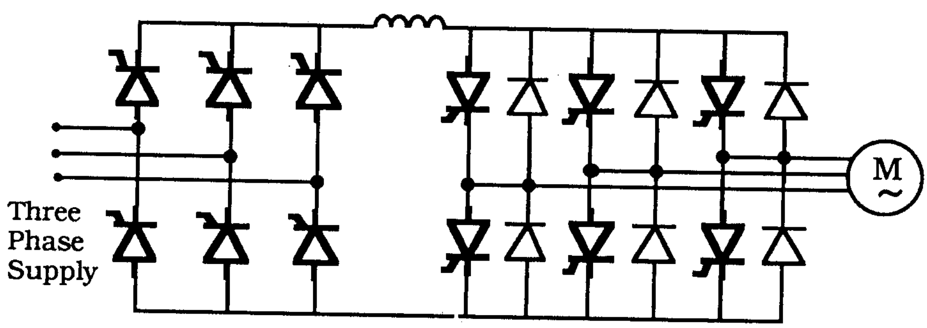


Fig 1.10 Current Source Inverter

1.3.2.3 The PWM Inverter

The PWM inverter is able to control both the output frequency and the output voltage and therefore the input is an uncontrolled rectifier as shown in figure 1.11. (devices are schematic only). The PWM output voltage consists of pulses of magnitude V_{dc} and whose widths are modified such that after low pass filtering only a fundamental component remains with very little harmonic content as shown in figure 1.12. The signals for each phase are 120 degrees apart whilst the signals in one arm are in anti-phase.

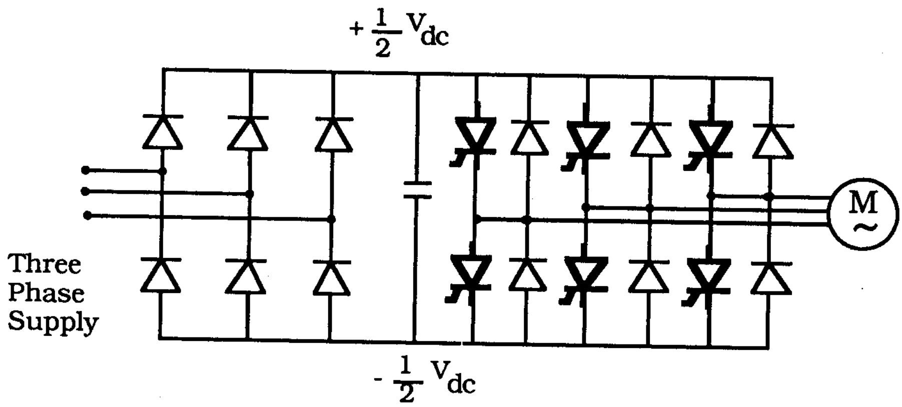


Fig 1.11 The PWM inverter

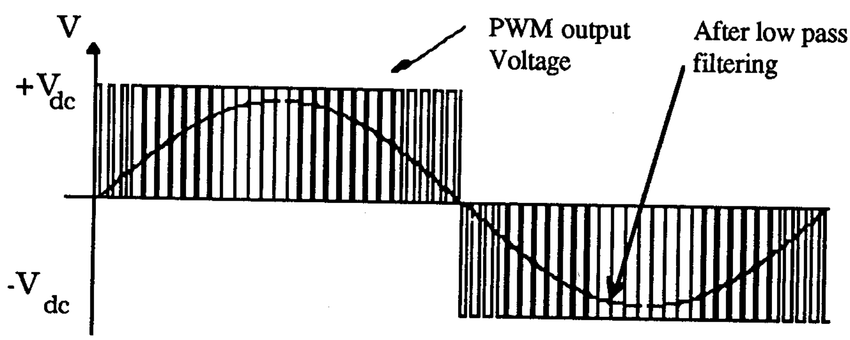


Fig 1.12 Typical PWM Motor Line Voltage

Details of the various PWM techniques available are further discussed in chapter 3, whilst the control aspects are discussed in chapter 2.

Chapter 2

Control Strategy of PWM Inverter

2.1 Control

Variable-speed control of induction motors is achieved by varying both the supply frequency and the supply voltage. Unlike d.c. motor control, induction motor control is complicated by the difficulty of determining the variables of the system and also because implementation of control is limited to only two control variables, the frequency and voltage. The control may be vector, scalar or open loop.

Vector methods are complex and necessitate the use of transducers that sense currents, flux, speed and voltages. Vector methods are suited for applications that demand quick dynamic response and accurate speed and torque control. However, the complexity of the system, due to the need for transducers, can deviate from the simplicity of the induction motor and other alternatives such as the reluctance motor may be better. Scalar methods of control, where the magnitude of the currents and voltages are used in a feedback loop, do not have the fast dynamic performance of vector methods but are simpler in design. Open loop control is the simplest and is suitable for a number of applications that require good steady state performance but not necessarily with fast dynamic performance. It is for this category of applications that the system described here is intended.

In an open loop control strategy for a voltage source inverter, the speed is varied by varying the supply frequency. The torque available may be set by the voltage and is adjusted such that the air-gap flux of the motor is constant. During changes in supply frequency the currents may be higher than rated and is limited by the rate of change of frequency, usually a ramp circuit. The open loop strategy is suitable for operating conditions where :

(i) the motor is operated under steady state conditions for long periods.

(ii) fast acceleration and deceleration are not required.

(iii) there is not much variation in load torque.

(iv) the dynamic performance of the motor is not crucial.

The main deficiencies of an open loop drive are :

(i). The shaft speed varies with slip frequency from full load to no-load. At lower speeds this can be rather large. This is reduced by maintaining constant flux operation over the whole speed range so that the slip remains almost constant.

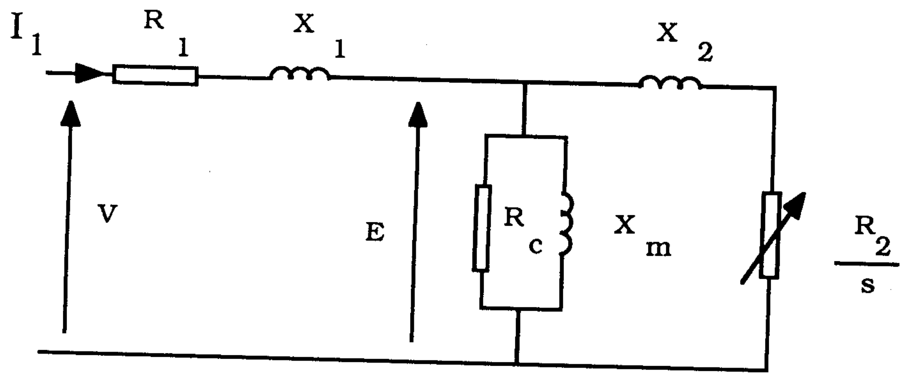
(ii) To compensate for the voltage drop due to stator resistance, voltage boosting is required at low frequencies. Voltage boosting at low frequencies, on no-load or light load, could lead to magnetizing currents larger than rated, leading to excessive heating if the motor is operated for long periods.

(iii) There is little indication of the torque that the motor is producing.

(iv) There is no direct control over the rise in currents during acceleration and deceleration of the motor. The current may be indirectly controlled by reducing acceleration.

2.2 Constant Flux Operation

Constant flux operation is only required for some applications to provide maximum torque capability throughout the entire speed range and also to keep the slip almost constant so that supply frequency is a reasonable estimate of the speed. For frequencies above 10 Hz, the V/f relationship, is normally kept constant. However, at lower frequencies the stator losses become significant and the voltage available to the air-gap is reduced.



Subscipt 1 - Stator
2 - Rotor (referred to stator)

Fig 2.1 Single Phase equivalent circuit of an 3-phase induction motor

For constant flux operation throughout the range of frequencies the ratio of voltage, E, to frequency, f, is kept constant. The amount of voltage boost, required to maintain E constant, may be obtained from the voltage equation:

$$\overline{V} = \frac{\overline{Z_1}\ \overline{E}}{\overline{Z}} + \overline{E} \qquad (2.1)$$

where $\overline{Z} = \dfrac{\overline{Z}_2 \overline{Z}_m}{\overline{Z}_2 + \overline{Z}_m}$, $\overline{Z}_2 = \dfrac{R_2}{s} + j X_2$

$\overline{Z}_1 = R_1 + j X_1$ and $\overline{Z}_m = \dfrac{j R_c X_m}{R_c + j X_m}$

The factor $\dfrac{\overline{Z_1}\ \overline{E}}{\overline{Z}}$ represents the additional voltage required to keep the flux constant.

At low frequencies the leakage reactance X_1 may be neglected as it is small compared with R_1.

$$\overline{V}_{boost} = \frac{ER_1}{\overline{Z}} \tag{2.2}$$

A good approximation may be obtained by further neglecting the magnetizing loop and the rotor leakage reactance [1].

$$\overline{V}_{boost} = \frac{\overline{E}\,R_1 s}{R_2} \tag{2.3}$$

However, the magnetizing current at low frequencies can be very high and care must be taken to keep the current below the rated current.

2.3 Acceleration of the Motor

When starting an a.c. motor from a variable frequency supply, the acceleration may be too fast leading to transient currents in excess of the ratings of the inverter. The starting currents are associated with rotor losses which are dissipated as heat. Thus, there is need to keep this temperature rise in the machine, which is determined by the thermal considerations, within permissible levels. An estimate of the permissible maximum energy may be obtained using the BS 4999 part 112 [2].

Assuming that for small motors the rotor inertia is negligible compared to the load inertia, the maximum permissible energy for two consecutive starts was calculated and this was compared with the energy dissipated with variable frequency starting. The p.u. energy dissipated is given by

$$W = \int_{\omega'}^{\omega''} \frac{J_n (\omega_s - \omega) T_e \, d\omega}{T_e - T_l} \tag{2.4}$$

and the time taken

$$t = \int_{\omega'}^{\omega''} \frac{J_n \, d\omega}{T_e - T_l} \tag{2.5}$$

(Appendix A.1. gives the detailed derivations of 2.4 and 2.5)

An indication of the currents during starting may be obtained from the approximation of differential energy as

$$\delta W = I_2^2 \; R_2 \, \delta t \tag{2.6}$$

A more accurate determination of the transient currents would require state-space methods because the approximate equivalent circuit does not cater for the transient conditions, which are represented by a set of differential equations. However, the approximation is sufficient for the purposes of this project because fast dynamic performance is not a primary criterion.

Using BS4999 as a guide, the permissible energy that can be dissipated in the rotor during starting, without overheating, is twice the energy dissipated by a direct on-line start with load torque proportional to speed and equal to rated torque at rated speed. The maximum allowable empirical value of the load inertia should be

$$J = 20.0475\ P^{0.9}\ p^{2.5} \qquad (2.7)$$

where P is rated power in Watts (W)

p is number of pole pairs

J is inertia (kg m^2).

Using equations 2.4, 2.5 and 2.6 a comparison between a direct on-line starting and variable frequency starting for the same load torque and inertia is shown in figure 2.2. (for conditions of R_2=0.1 p.u and X_2=0.2 p.u and p=2).

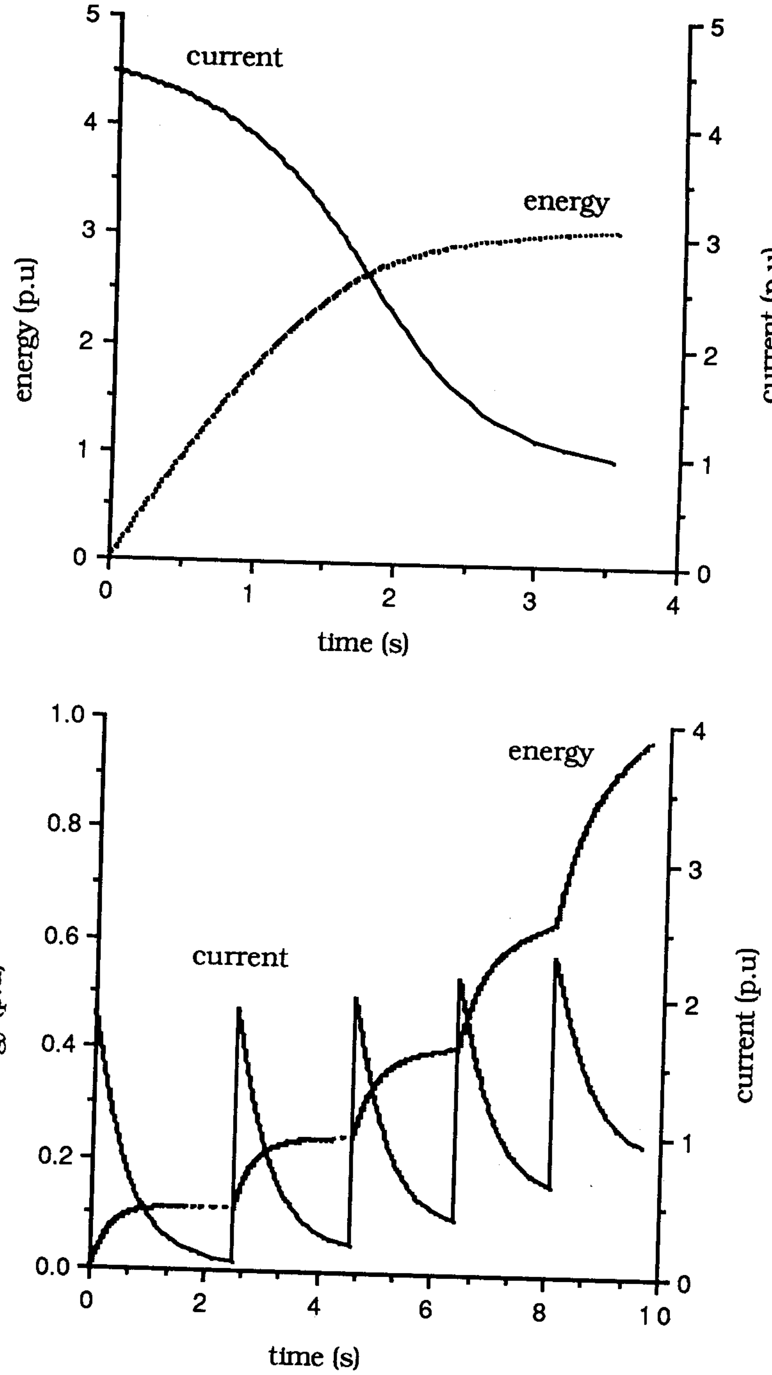


Fig 2.2 Comparison between direct on-line and step changes in frequency

The energy dissipated and the rotor current, and therefore the stator current, is lower with variable frequency than with direct on-line starting. A comparison of the effect of different frequency step sizes on energy dissipated and starting time is shown in figure 2.3 from which it will be observed that minimum dissipation of energy occurs when the frequency step is about 0.1 p.u. Shorter steps than this require a longer starting time and so the energy required increases. However, due to the longer time over which the energy is dissipated there would be no overheating of the rotor.

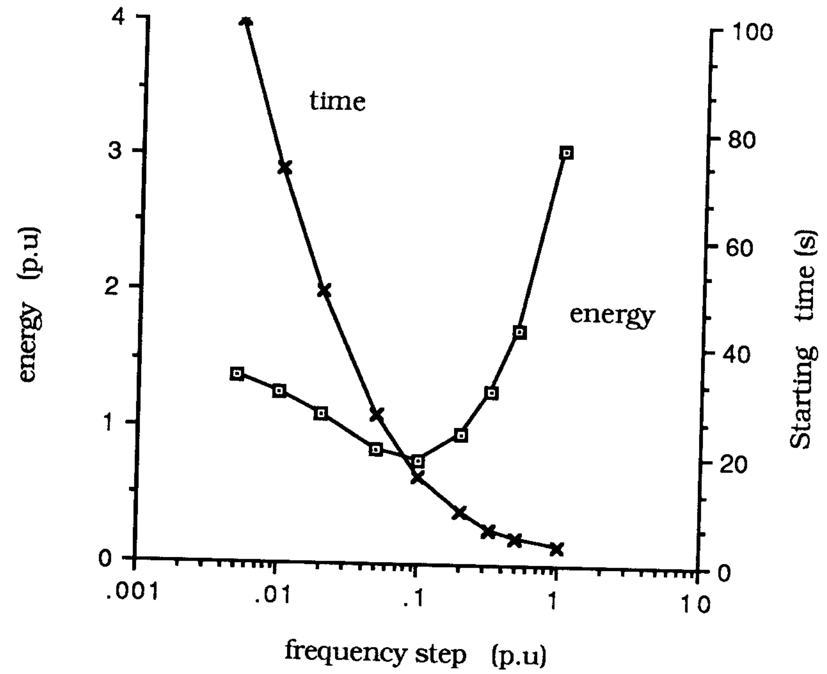
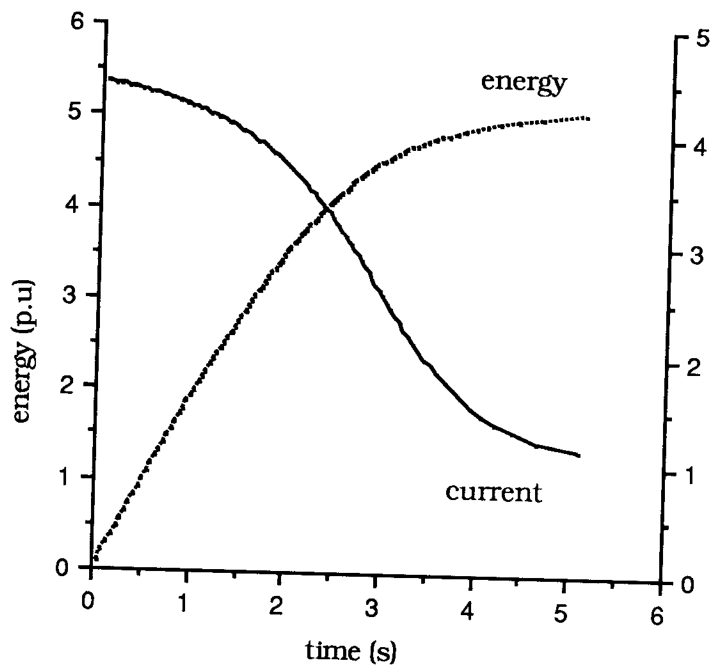


Fig 2.3 Comparison of step sizes

For a constant load at rated torque and with the same inertia as in figures 2.2 and 2.3 the comparisons are shown in figure 2.4.



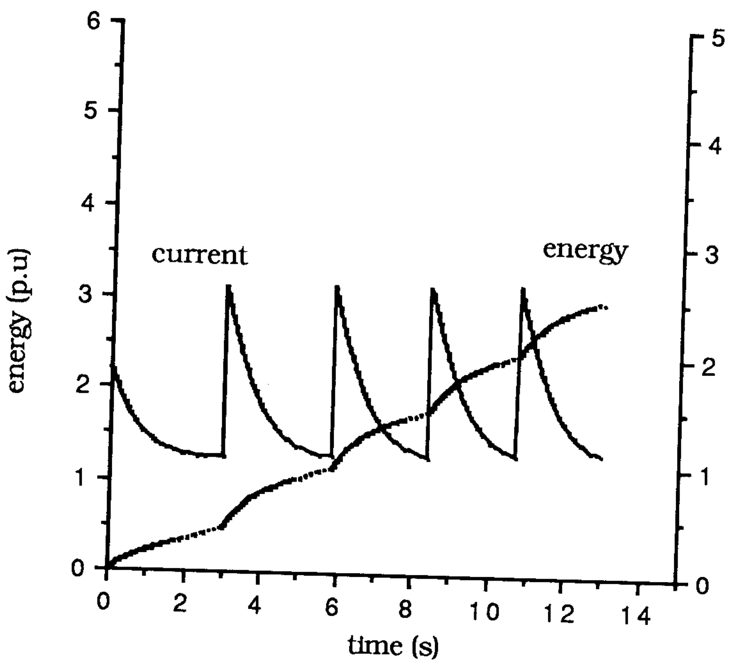


Fig 2.4 Comparison for constant load

It can be seen that if two consecutive starts are to be permitted the power dissipation is higher with direct starting and therefore variable frequency starting should be used.

During deceleration, the machine operates as an induction generator returning the energy to the d.c. rail through the inverter diodes. If this energy cannot be returned to the supply it will charge the capacitor, which is across the d.c. supply, causing the d.c. voltage to exceed the inverter voltage ratings. The energy may be dissipated by a resistor that is switched across the rails when the rail voltage exceeds the rated. "Plugging" the machine by a change of phase sequence thereby dissipating the energy within the rotor of the motor may instead be used to brake the motor. This latter method could lead to overheating of the rotor.

For the applications intended, rapid deceleration is not required and therefore no resistor is required as long as the capacitor voltage is kept within the maximum voltage ratings of the inverter.

2.4 Torque

The electromagnetic torque for constant flux operation is derived from figure 2.1 and is given by the equation:

$$T_e = pm \left[\frac{E}{\omega_r}\right]^2 \frac{(\omega_s - \omega)\, R_2}{R_2^2 + (\omega_s - \omega)^2\; L_2^2} \tag{2.8}$$

where: ω_r is the rated frequency,

ω_s the motor supply frequency and

ω the rotor frequency

The breakdown torque occurs at

$$\omega_s - \omega = \frac{R_2}{L_2} \tag{2.9}$$

and the breakdown torque is

$$T_b = pm \left[\frac{E}{\omega_r}\right]^2 \frac{1}{2\, L_2} \tag{2.10}$$

At low speeds, maximum torque capability may not be possible at low stator frequencies as shown in figure 2.5. If maximum torque is required at standstill, the minimum stator frequency is obtained from equation 2.9. with $\omega = 0$.

$$\omega_{min} = \frac{R_2}{L_2} \quad \text{or} \quad f_{min} = \frac{R_2}{2\,\pi\, L_2} \tag{2.11}$$

This, however does not take into account the losses due to windage and friction.

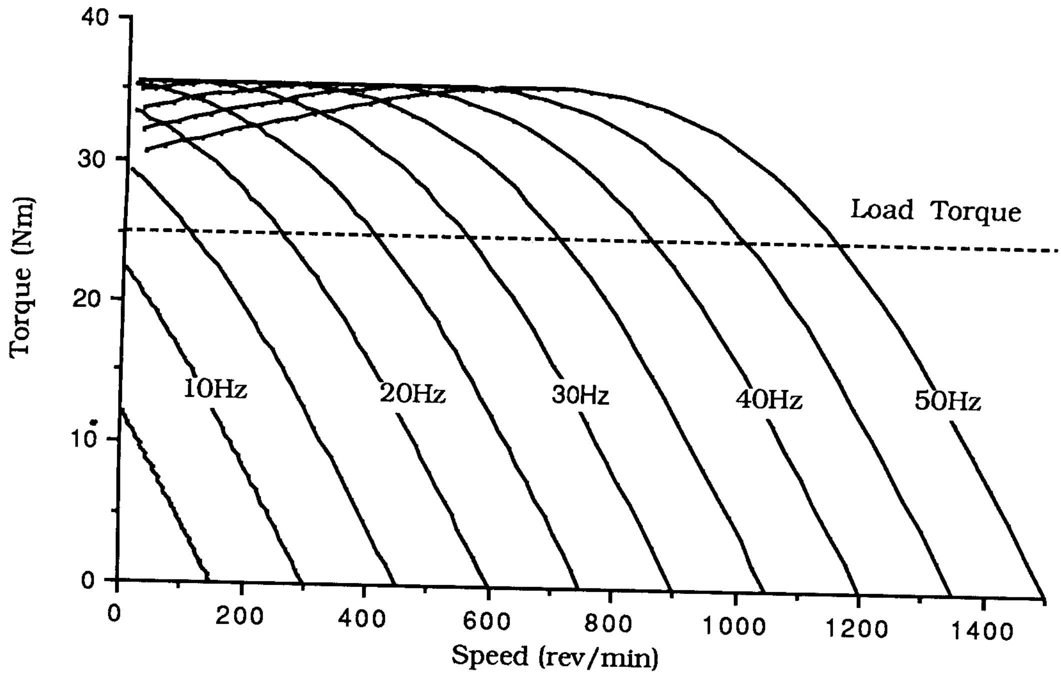


Fig 2.5 Ideal Torque-speed characteristics

Chapter 3

Pulse Width Modulation Techniques

Pulse width modulation (PWM) techniques are many and varied. They may be grouped into four basic techniques based on the way the switching angles are determined [3].

i). **Sampling Techniques :** Switching angles are determined by the points of intersection of two waveforms, one the carrier waveform (usually a triangular waveform) and the other the modulating waveform (sinusoidal, staircase, trapezoidal.etc.).

ii). **Harmonic Elimination Technique :** This is based on the square waveform but with notches added at specific points in order to eliminate certain harmonics.

iii). **Distortion Minimization Technique :** The same as (ii) except that a performance index or loss factor is defined and then the angles are computed to minimize the index.

iv) **Space Vector :** The method of representing the three phase waveforms by a single vector may be extended to six pulse inverters. Eight vectors are used to represent the various combinations of the inverter arms. By a combination of these vectors a sinusoidally rotating vector may be obtained.

3.1. Sampling Techniques

3.1.1 Sinusoid Derived PWM

In this method a sinusoid or an approximation to a sinusoid is used as a modulating function. The diagram below shows the main sinusoid derived PWM waveforms [4], namely natural, regular asymmetric, regular symmetric and modified regular PWM.

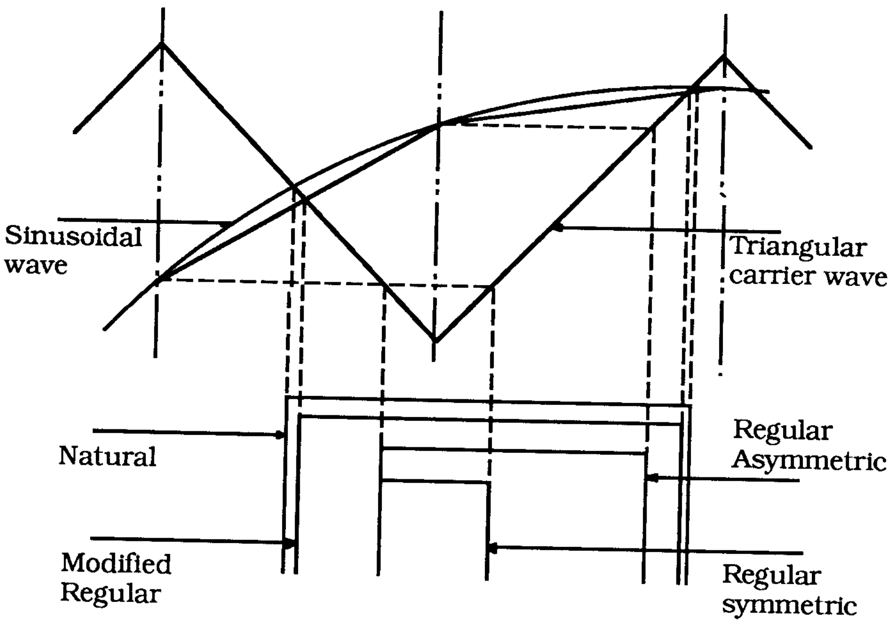


Fig 3.1 Sinusoid Derived PWM

The main sinusoid derived functions are as follows:

a) **Sinusoidal or Natural PWM :**

Here the angles are determined by the intersection between a triangular carrier wave and a sinusoidal modulating waveform [5]. The harmonics are the sidebands of the carrier frequency and its multiples of order $K = np + m$, the mth sideband of the nth carrier harmonic, and it is these that contribute significantly to the harmonic losses [6]. The carrier-to-modulator frequency ratio, p, usually 9 or 12, is chosen to be a multiple of 3 so that the carrier harmonics may be eliminated. For low values of p, gear changing is required to keep switching frequency within limits [7].

Analogue implementation is difficult because of the difficulty in generating a stable sinusoidal waveform due to d.c. offset and parameter drift. Digital implementation is easier and requires the use of look-up tables where switching angles have already been computed off-line. A microprocessor may be used to access the tables and to provide some form of control and monitoring.

b) **Regular Sampled symmetric PWM :**

Regular Sampled symmetric or Uniform Sampled PWM is a digital approach to sinusoidal PWM where the PWM waveform can be defined by an analytic expression [5].

$$t_{pw} = \frac{T}{2} \left\{ 1 + M \sin \left(\omega_m T_i \right) \right\} \qquad (3.1)$$

Where T is one period of the carrier waveform

T_i is the sample instant

$M \sin (\omega_m T_i)$ is the original sine wave

t_{pw} the pulse width.

The sine-wave is sampled at regular intervals corresponding to the positive peaks of the carrier triangular waveform and the resulting stepped waveform compared with the carrier to obtain the switching instants. Since the technique can be represented by an analytical expression, a microprocessor based generation technique using software based calculation is possible [5]. Specialized integrated circuits also exist which can produce this waveform.

c) **Regular sampled asymmetric PWM :**

This is the same as regular sampled symmetric PWM except that the sine wave is sampled at both peaks of the carrier. It is also expressed in the form of an analytic expression and, hence, a microprocessor based generation method may be used [8] [9].

3.1.2 Staircase PWM

A staircase waveform is used instead of a sinusoidal waveform [12]. The number of steps and step sizes of the waveform may be selected in such a way as to eliminate certain harmonics. A higher fundamental is achieved than sinusoidal PWM. According to Nystrom *et al* [13], torque pulsations are reduced because for each torque harmonic pulsation, the difference between the two current harmonics involved in producing it is low. Hardware implementation is possible using discrete logic. Three steps have been used and the carrier-to-modulator frequency ratio is 15 or 21. Higher ratios would require more steps.

3.1.3 Other Modulating Functions

A Trapezoidal modulating function has been suggested as an easy way of generating PWM suitable for microcomputer implementation [14]. Some non-linear functions have also been suggested for modulation depths exceeding unity [15]. These have also been done at low carrier-to-modulator ratios.

3.1.4 High Carrier - to - Modulator Ratio

A sampling technique with a high carrier frequency could also be used for generation of PWM. The maximum frequency is limited by the ratings of the switching devices and their associated losses. For a number of reasons a high carrier frequency is preferred [16] [17] [18].

a) The filter components are small, light, and inexpensive. Also the reactance of the motor stator coils attenuates the higher frequency components.

b) The unwanted carrier component and associated sidebands in the PWM waveform are well separated from the required output. Thus it would be possible to filter almost all the unwanted components.

c) The carrier frequency will be above the audio frequency leading to "noiseless" operation [19].

d) The drive operates with practically sinusoidal stator currents.

e) Radio frequency interference (RFI) from the cables can be eliminated.

f) The use of high carrier frequency increases the bandwidth of the output of the converter therefore addition of triplen harmonics would be possible.

3.2. Harmonic Elimination

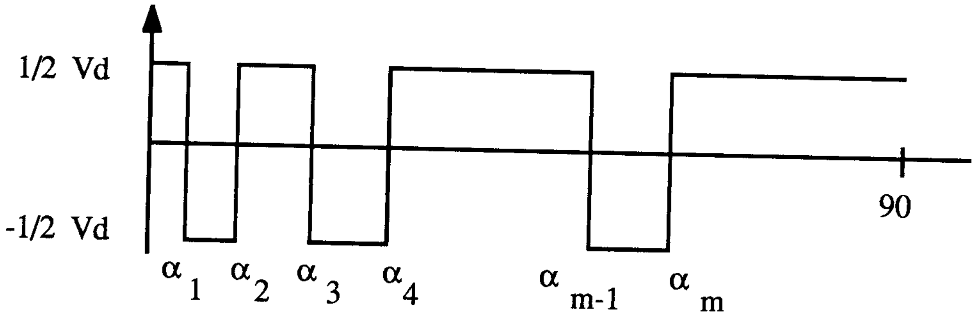


Fig 3.2 Harmonic Elimination

For any wave, with notches as shown, the kth harmonic voltage is given by :

$$V_k = \frac{2V_d}{k\pi}\left[1 + 2\sum_{i=1}^{i=m} (-1)^i \operatorname{Cos}(K\alpha_i)\right] \qquad (3.2)$$

for m switching instants per quarter cycle [20] [21].

By solving these equations, (m-1) harmonics can be eliminated and the fundamental controlled. Implementation of harmonic elimination requires a microprocessor and a look-up table using values obtained by an off-line solution of the equations to determine the switching angles [22, 23, 24]. Control of the fundamental and the suppression of the first four harmonics requires 5 switching instants per phase per quarter cycle or 22 per cycle corresponding to a carrier-to-modulator ratio of 22. This makes it possible to use a lower switching frequency and at the same time remove the low order harmonics. However, the amplitude of the higher order harmonics increases [25].

3.3. Minimization of Distortion

A performance index is chosen which represents the losses due to harmonics and then the switching angles are chosen in order to minimize the performance index. Various indices have been suggested mainly considering the harmonic currents [25, 26, 27]. Buja and Indri have defined a factor, the total harmonic voltage [25].

$$\sigma = \left[\sum_{k \neq 1} \frac{V_k}{k} \right]^{\frac{1}{2}} \qquad (3.3)$$

The factor, σ, is proportional to the total rms harmonic current neglecting skin effect. The square of this factor is proportional to the motor copper losses due to harmonics. The various loss factors that have been suggested have similar losses for different fundamental frequencies and just one loss factor as suggested by Buja and Indri may be used. The loss factor may be used to compare various PWM strategies [29]. Harmonic minimization does not eliminate the low order harmonics but rather attenuates them. The low order harmonics are the main cause of waveform distortion leading to reduction in the fundamental voltage component and also to increase in torque pulsations. Calculation of switching instants is done off-line and the values stored in look-up tables. A microprocessor is generally used to access the tables and some form of algorithm is used for flexibility.

3.4. Space Vector Modulation

A three-phase inverter may be represented by eight voltage vectors corresponding to the position of the switches. There will be two null vectors and six voltage vectors spatially displaced by 60° (figure 3.3). A rotating vector, V_s , which may be considered as representing the sum effect of the three phases on the fundamental rotating flux, is obtained by using the voltage vectors for direction and the null vectors for magnitude control.

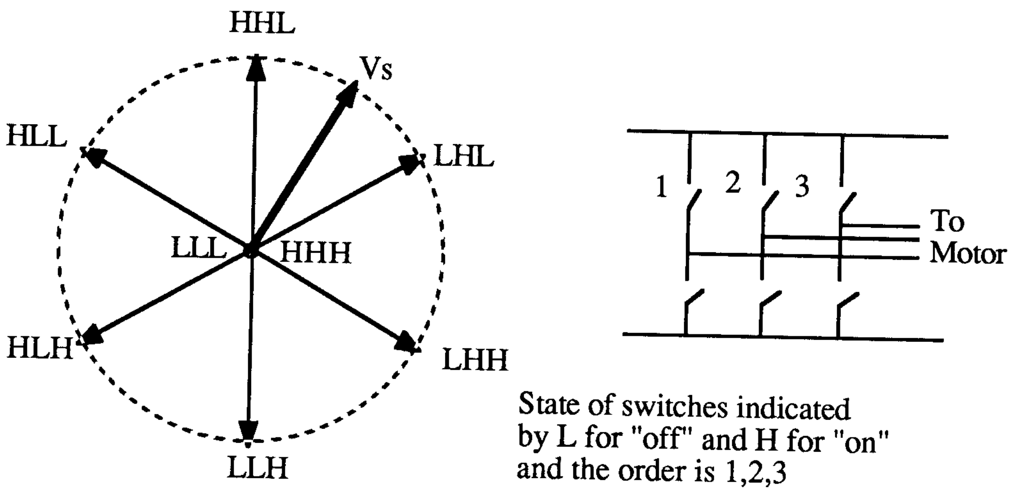


Fig 3.3 Space Vector Modulation

The space vector method is easy to implement by discrete implementation as well as by microprocessor. Some authors have described the space vector method as superior to the conventional sampling methods due to its ease of microprocessor implementation and the fact that waveforms are very nearly loss minimized [30] [31] [32].

3.5. Generation of PWM

The method of generating the PWM waveform is also important as it puts limits to the PWM technique used. The following are the main methods :

i) **Analogue generation :** This was common in the earlier forms of PWM but suffers from the effects of drift, dc offset, and temperature rises of analogue components, and is easily influenced by external sources of disturbances. It lacks flexibility and can be complex. It, however, has continual control of the output and there is no delay as in digital systems.

ii) **Discrete digital logic implementation :** Generally easier to implement but it lacks flexibility and the control is specific to the application. It also increases the component count.

iii) **Application Specific Integrated Circuits (ASIC) :** These are integrated circuits specially designed to generate PWM and thereby reduce the component count and complexity. Some ASIC's are used as microprocessor peripherals.

iii) **Microprocessor :** There are many varied implementations of microprocessors. The microprocessor may read the switching angles which have been pre-computed or it may compute them in real-time and output the PWM with or without some other external counters or timers. Another way would be to use a peripheral device to do the computation leaving the microprocessor to deal with the control aspects. Microprocessor implementations are more flexible than (i) and (ii) because they are controlled by software more than by the hardware.

The main features of microprocessor based generation of PWM has been reviewed by many authors. Hoft *et al* [33] have given a detailed review of microprocessor applications, over the period 1982-1984, and the various routines and factors involved. Bowes and Mount [34] outlined the various control functions of microprocessor implementation. Iwanciw [35] has described the user interface, diagnostics, links and application routines required. Lettner [36] has outlined the main routines for the interface of a portable PC with a microprocessor controlled drive with emphasis on the use of graphical user interfaces and communication. Sethuramen and Sagar [37] describe the main features of an expert system.

The use of ASIC's is expected to be the most common means of a.c. drive control especially at the lower power market. Component count and complexity of control is reduced [38]. Use as a microprocessor peripheral can lead to very sophisticated and efficient control algorithms being developed and tested but being implemented in an easy way [30] [40].

Chapter 4

Optimum Waveform Generation

4.1 Description of Ideal Waveform

A sinusoidal current applied to an induction motor, with appropriate phase shifts, is considered to give the best performance in terms of smoothness of rotation and losses. Current harmonics have been shown, by many authors [41, 42, 43, 44], to lead to torque pulsations and other losses. The typical PWM phase voltage, shown in figure 4.1, consists of a fundamental voltage and other voltage harmonics. The current will, however, have harmonics of smaller magnitudes, due to the motor reactance, thus approximating a sinusoidal current. Though voltage harmonics are generally not desirable, addition of certain harmonics, such as the triplen harmonics, may be an advantage.

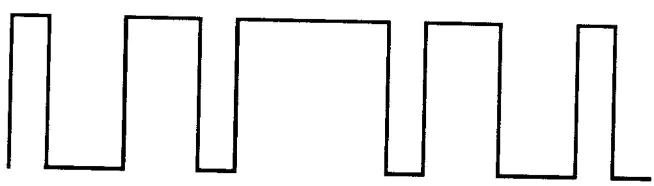


Fig 4.1 Typical PWM Voltage Waveform

An indication of the available voltage from the DC rail voltage may be obtained from the utilization factor defined as

$$\text{Utilization Factor} = \frac{\text{RMS of fundamental line voltage}}{\text{DC Rail voltage}} \qquad (4.1)$$

In a three-phase induction motor, connected in star, the line voltage is related to the phase voltage, V_p, by

$$V_L = \sqrt{3}\ V_p \tag{4.2}$$

If the dc supply to a 6-pulse inverter is ± 1 units then for a sinusoidal phase voltage of maximum magnitude

$$V_p = \sin ø \text{ units}$$

the line voltage will be

$$V_L = \sqrt{3} \sin ø \text{ units}$$

and $\quad$ utilization factor $= \dfrac{\sqrt{3}}{2\sqrt{2}} = 0.61.$

However, the maximum magnitude possible for a sinusoidal line voltage is 2 units, i.e.

$$V_L = 2 \sin ø \text{ units}$$

For this the phase voltage must be

$$V_p = (2/\sqrt{3}) \sin ø \text{ units}$$

and $\quad$ utilization factor $= \dfrac{2}{2\sqrt{2}} = 0.707$

which is not allowable because the magnitude of the instantaneous phase voltage can not exceed 1 unit. Since triplen harmonics do not appear in the line voltage, addition of these harmonics to the phase voltage would keep the magnitude of the instantaneous phase voltage below 1 unit. The problem is generalized to :

$$f(\omega t) = V_1 \sin(\omega t) + V_3 \sin(3\omega t) + V_9 \sin(9\omega t) + \ldots$$

and $\left| f(\omega t) \right| \leq 1 \quad$ for all t $\qquad$ (4.3)

where the coefficients V_1 , V_3 , V_9 ,..... are to be determined in order to maximize V_1 . The ideal waveform is one that utilizes the rail voltage most effectively and also produces the least harmonics.

4.2 Previous Work

Harmonic elimination and harmonic minimization seek to reduce the lower order harmonics to acceptable values and the waveforms generated are optimum in this respect. Their voltage utilization factor, however, is about 0.6. Bowes and Midoun [45] followed a procedure to optimize the total harmonic distortion THD (equation 4.4) in a search for a simple modulation process based on well-defined techniques that would produce the same results as optimized PWM techniques.

$$THD = \frac{1}{I_1}\left(\sum_{k=3}^{\infty} I_k^2\right)^{\frac{1}{2}} \tag{4.4}$$

where I_1 and I_k are the fundamental and harmonic rms currents.

Using a regular asymmetric technique, the addition of a third harmonic of 25% of the fundamental was found to be optimum in the reduction of THD factor. Comparison with the regular asymmetric and symmetric waveforms showed that this was superior in terms of THD especially at modulation depth near and above unity. These comparisons were done with low carrier-to-modulator frequency ratios.

Boys and Walton [46] have also shown that the harmonic losses may be optimized with the addition of about 25% to 28% third harmonic. The reason for this improvement was attributed to the change in harmonic distribution. The staircase waveform has been suggested by Nystrom *et al* [47] as it has smoother torque performance than regular asymmetric PWM. This was attributed to the fair distribution of harmonics. For space vector modulation technique and carrier ratio 9, addition of about 25% third harmonic has been shown by Handley and Boys [48] to have least THD. Other functions have been suggested but all of them deal with low carrier-to-modulator frequency ratios of between 9 and 25.

For the purpose of increasing the voltage utilization Trzynadlwski [49] gave the following harmonic modulating function for a dc rail voltage of ± 1/2 units :

$$f_{(3q)}(\omega t) = \begin{cases} f_{(3q)}(\omega t) & 0 \le \omega t < \pi \\ f_{(3q)}(2\pi - \omega t) & \pi \le \omega t < 2\pi \end{cases} \tag{4.5}$$

$$\text{where } f_{(3q)}(\omega t) = V_{(1)} \sin(\omega t) + \sum_{n=1}^{q} V_{(3n)} \sin(3n\omega t + \emptyset)$$

and obtained optimized values for q = 1,2,3 and 4. Thus, for q=4,

$$f_{(12)}(\omega t) = 1.200 \sin(\omega t) + 0.239 \sin(3\omega t) - 0.050 \cos(6\omega t) - 0.019 \sin(9\omega t) + 0.007 \cos(12\omega t) \tag{4.6}$$

This includes even harmonics of 6 and 12 whose net effect is zero due to asymmetry. Also the line voltage cannot support a fundamental phase voltage of 1.2 units without there being some distortion of the output. This and the discontinuities at π introduce more harmonics of higher order than shown by equation 4.6.

Taniguchi *et al* [50] has proposed a PWM technique where each arm is turned off for one third of a cycle whilst still achieving a sinusoidal waveform at the output (figure 4.2). It can be shown that this method reduces losses. The waveform is described by the equation

$$f(x) = \begin{cases} \sin(\omega t+\pi/3) & 0 < \omega t < 2\pi/3 \\ 0 & 2\pi/3 < \omega t < \pi \end{cases} \qquad (4.7)$$

The waveform, however, does not have quarter wave symmetry and existing ASIC's are designed for quarter wave symmetry.

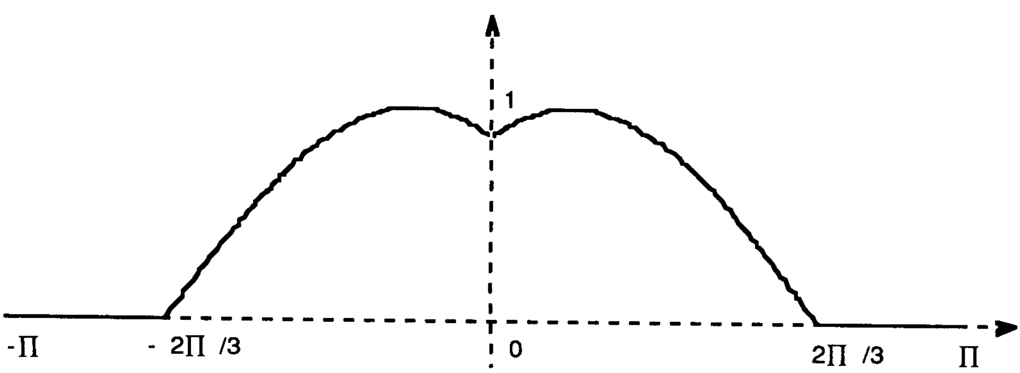


Fig 4.2 PWM modulating waveform

4.3 Proposed Waveform

In the previous work described above the ratios of carrier-to-modulator frequency were in the range 9 - 25. For the very high carrier-to-modulator ratios, considered in this work, the voltage harmonics at low frequency are reduced, and it can be expected that the addition of the third harmonic will also have a reduced effect on the losses.

4.3.1 High Carrier - to - Modulator Ratio

The harmonic currents are calculated using the simplified single phase equivalent model of the motor where the effect of the resistances and the magnetizing loop are considered negligible. Thus

$$I_k = \frac{V_k}{k\,X} \qquad (4.8)$$

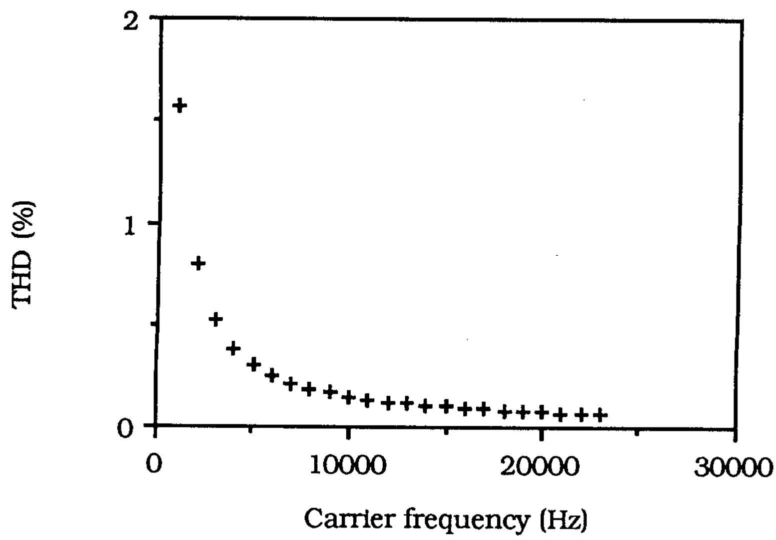
where I_k and V_k are the kth harmonic current and voltage respectively. $X = X_1 + X_2$ is the leakage reactance of the motor at fundamental frequency. For asynchronous PWM, the prominent voltage harmonics will be sidebands of the carrier frequency and for a high carrier frequency the corresponding current harmonics will be small. For example, for a carrier frequency of 23 kHz and fundamental 50 Hz (ratio of 460) , the current harmonic around the carrier is, in per-unit form,

$$I_{460} = \frac{V_{460}}{460\,X_{pu}} \times 50 = \frac{0.11\,V_{460}}{X_{pu}} \qquad (4.9)$$

V_{460} may be 0.2 to 0.3 of the fundamental voltage and therefore I_{460} is negligible. The harmonic distortion factor THD suggested by Buja is used for comparison and is defined as [51].

$$\mathrm{THD} = \frac{1}{V_1} \left(\sum_{k=5}^{\infty} \left(\frac{V_k}{k} \right)^2 \right)^{\frac{1}{2}} \quad \times \quad 100\,\% \qquad (4.10)$$

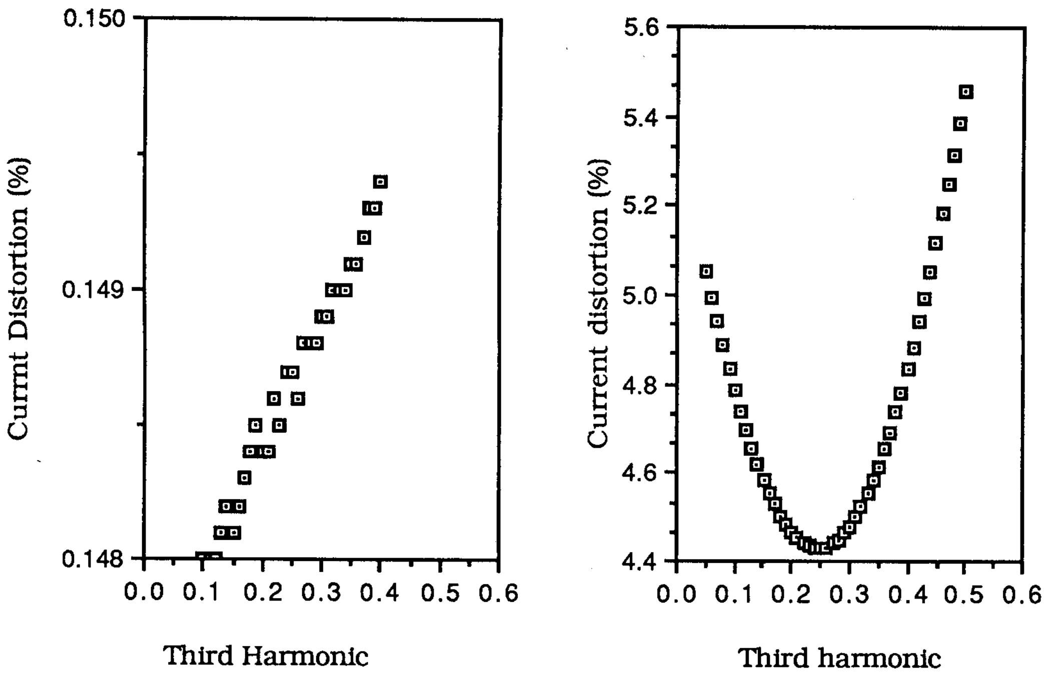
This factor is proportional to the total harmonic currents. A graph of the THD factor for various carrier frequencies with modulation depth of unity and modulator frequency of 50 Hz is shown in figure 4.3.



* The THD were obtained for carrier frequencies of multiples of 1000.1 Hz in order to avoid the synchronizing effect of the carrier being an integer multiple of the modulating waveform. Harmonics taken are up to sixty one.

<u>Fig 4.3 Variation of current distortion with carrier frequency</u>

The PWM current waveform may thus be considered to be free of harmonics and therefore the utilization factor will be more important. The addition of third harmonic has negligible effect on the current distortion when the carrier-to-modulator ratio is very high compared to when the ratio is low, as shown in figure 4.4. This is because the significant voltage harmonics are the sidebands of the carrier frequency and, therefore, their current contribution is negligible. For low carrier-to-modulator ratio the optimum, as expected, is around 25 % as shown in figure 4.4 b.



(a) High Ratio (b) Low Ratio

Fig 4.4 High Vs Low Carrier-to-Modulator ratio

4.3.2 MODULATING WAVEFORM

The waveform suggested by Taniquchi *et al* [50] is given by

$$f(\omega t) = \frac{2}{\sqrt{3}} \cos(\omega t) + \sum_{n=1}^{\infty} \frac{3}{\pi(1-9n^2)} \cos(3n\omega t) \qquad (4.11)$$

This has only triplen harmonics added to the fundamental. However, the presence of even harmonics is generally not acceptable as the effect cannot be beneficial due to asymmetry.

A waveform based on the above but without the even harmonics, would be the optimum waveform possible for modulation less than unity. The waveform is described by the equation 4.12 which has an unlimited number of terms.

$$f(\omega t) = \frac{2}{\sqrt{3}} \cos(\omega t) + \sum_{n=1\ (n\ \text{odd})}^{\infty} \frac{3}{\pi(1-9n^2)} \cos(3n\omega t) \qquad (4.12)$$

Any further increase in the fundamental can only be achieved by the distortion of the output with non-triplen harmonics. The square waveform is supposed to have the maximum available fundamental component of $4/\pi$ yielding a voltage utilization factor of 0.90. Deliberate distortion of the output so as to increase the fundamental voltage is possible by the addition of other harmonics. Trzynadloski [49] has suggested some non-linear functions for this purpose.

Since only the first few terms are assumed crucial, an optimum waveform based on equation 4.12 is

$$f(\omega t) = 1.1547 \sin(\omega t) + 0.2387 \sin(3\omega t) - 0.02387 \sin(9\omega t) + 0.00853 \sin(15\omega t) \qquad (4.13)$$

The waveform is shown in figure 4.5.

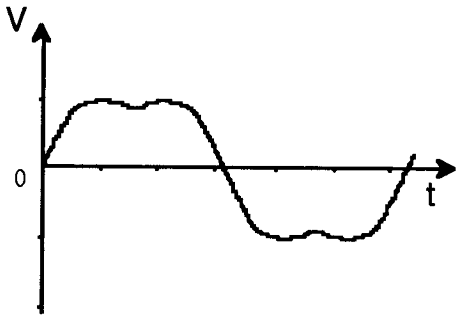


Fig 4.5 Waveform of optimum modulating signal

In Figures 4.6 to 4.8, four waveforms described by equations (a) to (d) below, are compared with respect to THD, utilization factor and torque distortion as functions of modulation depth.

(a) $f(\omega t) = 1.1547 \sin(\omega t) + 0.2387 \sin(3\omega t) - 0.02387 \sin(9\omega t) + 0.00853 \sin(15\omega t)$

(b) $f(\omega t) = 1.1547 \sin(\omega t) + 0.2387 \sin(3\omega t) - 0.05456 \cos(6\omega t) - 0.02387 \sin(9\omega t) - 0.01336 \cos(12\omega t) + 0.00853 \sin(15\omega t)$

(c) $f(\omega t) = \sin(\omega t)$

(d) $f(\omega t) = 1.1547 \sin(\omega t) + 0.2387 \sin(3\omega t)$

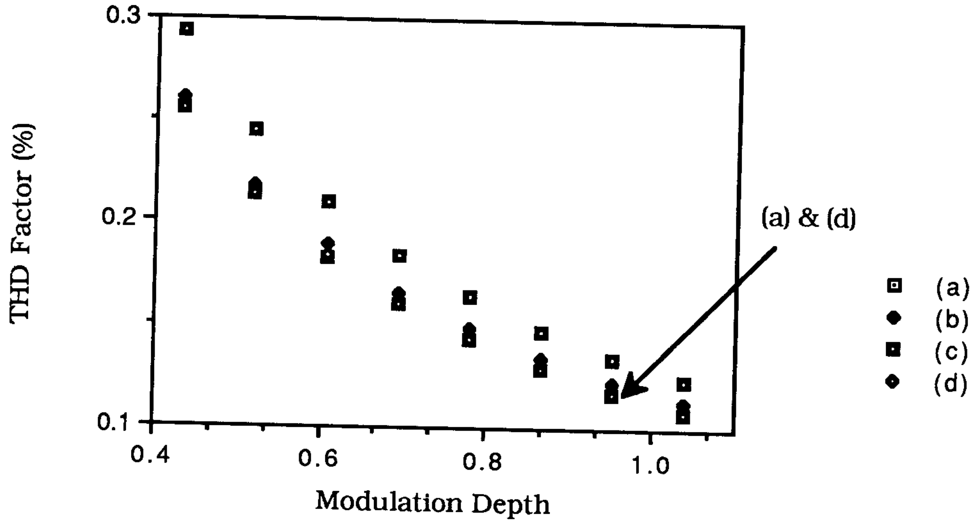


Fig 4.6 THD factor for various functions

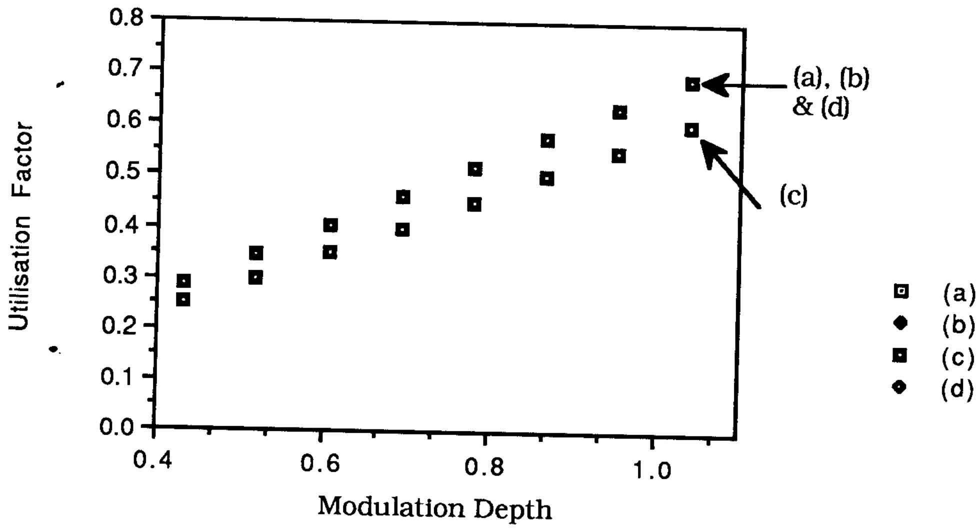
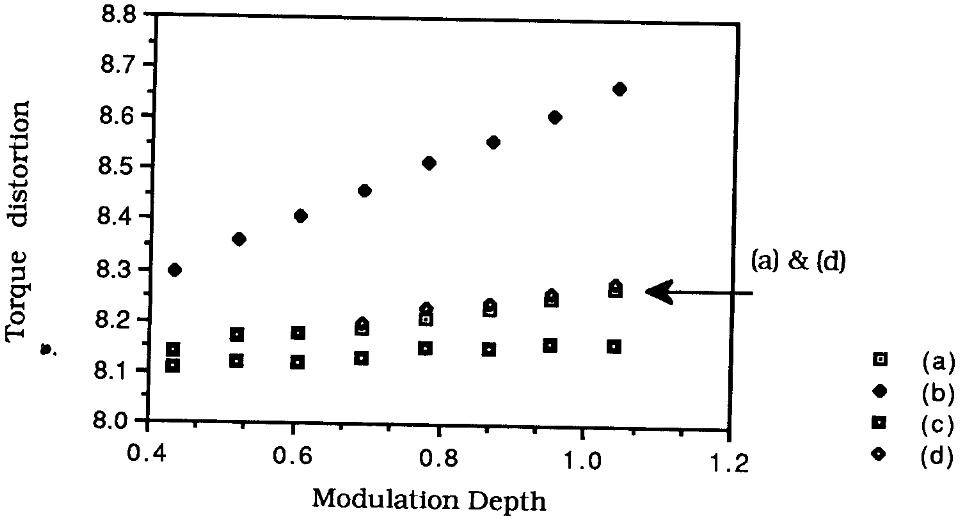


Fig 4.7 Voltage Utilization for various functions



* Rail voltage =400 V   * Rated motor voltage=230 V

Fig 4.8 Torque pulsation factor for various functions

The torque pulsation factor, t_f, is defined as

$$t_f = \sum_{n=6k} \left[\left(\frac{V_{n+1}}{n+1}\right)^2 + \left(\frac{V_{n-1}}{n-1}\right)^2 - 2\left(\frac{V_{n-1}}{n-1}\right)^2 \left(\frac{V_{n+1}}{n+1}\right)^2 \cos(\varnothing_{n-1} - \varnothing_{n+1}) \right]^{\frac{1}{2}} \quad (4.14)$$

where $\varnothing_n$ is the angle between per-unit voltages V_1 and V_{n-1}

This is obtained from the equation for torque pulsation given in per-unit form as [43] :

$$T_n = \Phi \times \left(I_{n+1}^2 + I_{n-1}^2 - 2\, I_{n-1}^2 I_{n+1}^2 \cos(\varnothing_{n-1} - \varnothing_{n+1}) \right)^{\frac{1}{2}}$$

where T_n is the $6k^{th}$ harmonic torque pulsation. (4.15)

It will be noted that the apparent lower THD for the functions with third harmonic is due to the higher fundamental voltage. For all the waveforms, it can be seen that addition of triplen harmonics has a negligible effect on the current distortion and the contribution to torque pulsations is also insignificant. However, as expected, there is a definite improvement in the utilization factor. Thus, the addition of about 20.7% third harmonic, as discussed earlier, would be the optimum.

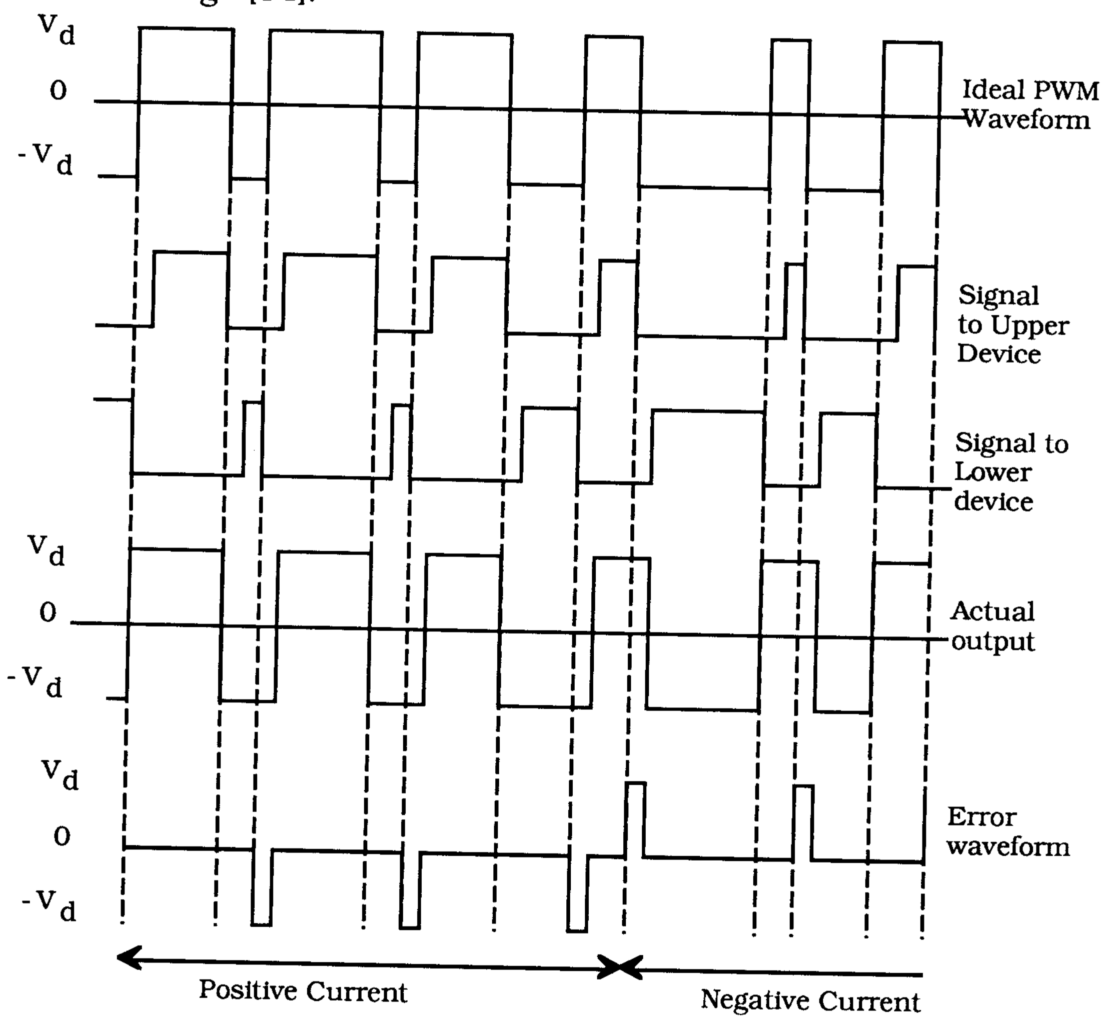
4.2 **Dead Time**

Dead-time, also known as lag-time or dwell-time, is the minimum time between turning a device in one leg 'off' and the other 'on' to safeguard against both devices turning on simultaneously. This feature introduces a distortion of the PWM waveform and has the effect of reducing the fundamental output voltage and introducing low order voltage harmonics not present in the ideal waveform and a consequent degradation of torque characteristics. Several authors have studied these effects and suggested compensation circuits [52, 53, 54]. Dodson *et al* [52] have described a general circuit using current feedback, whilst Murai *et al* [54] have described one with voltage feedback .

Dead-time adds a square voltage envelope, of the same frequency and phase as the fundamental output current, consisting of pulses whose widths are t_d, which is the dead-time, and height V_d, the rail voltage, to the ideal PWM wave (figure 4.9). After low pass filtering it may be considered as a square wave of height

$$V_e = f_c\, t_d\, V_d \qquad (4.16)$$

Where t_d is the dead-time, f_c is the carrier frequency and V_d the dc voltage [54].



* Current is positive when flowing out of leg to motor

Fig 4.9 Effect of Dead - time

The harmonics due to this are those of a square wave

$$V_k = \frac{4\,V_e}{k\,\pi} \tag{4.17}$$

and if the fundamental voltage component is given by

$$V_1 = M\,V_d \tag{4.18}$$

Using equations 4.9, 4.16 and 4.17

$$\mathrm{THD} = 100\% \times \frac{4\,f_c\,t_d}{\pi\,M}\left(\sum_{k=5}^{k=\infty}\frac{1}{k^4}\right)^{\frac{1}{2}}$$

$$\approx \frac{6\,f_c\,t_d}{M}\,\% \tag{4.19}$$

If t_d = delay/(512*f_c) where delay is an integer as will be used with the PWM generator

$$\mathrm{THD} \approx \frac{0.01\ \mathrm{delay}}{M}\,\%$$

for $M=1$ and delay $=10$, THD $\approx 0.1\%$

for $M=0.5$ and delay $=10$, THD $\approx 0.2\%$

If the current and voltage are in phase then the loss in fundamental rms voltage is given by

$$\dot{V}_{loss} = \frac{4\, V_e}{\sqrt{2}\, \pi} = \frac{4\, f_c\, t_d\, V_d}{\sqrt{2}\, \pi} \tag{4.20}$$

$$= \frac{\text{delay}\, V_d}{\sqrt{2}\; 128\, \pi}$$

for V_d = 400 Volts and delay = 5 , $V_{loss} \approx 3.5$ Volts

It will be noted that this does not change with the modulation depth and therefore it is best to operate at close to maximum modulation depth. Furthermore, the motor current is not in phase with the voltage and therefore V_{loss} is further reduced.

4.3 Harmonic Losses

If the losses associated with harmonics are excessive the motor may need to be de-rated to prevent overheating. For a squirrel cage induction motor the main harmonic losses may be separated into :

a) Stator copper winding losses

b) Rotor copper winding losses

c) Skew-leakage and end-leakage losses:

d) Stator & rotor laminations or core losses

An estimate of the rotor and stator copper harmonic losses may be obtained by neglecting the skin effect

$$P_{copper} = \sum_{k \neq 1} I_k^2 R \qquad (4.21)$$

where $R = R_1 + R_2$

Using equations 4.8 and 4.10

$$P_{copper} = \left(\frac{V_1}{X}\right)^2 \left(\frac{THD}{100}\right)^2 R \qquad (4.22)$$

For a more accurate determination of copper losses in the stator and rotor windings skin effect has to be considered. The influence of skin effect on losses may be approximated by making the stator and rotor resistance values proportional to the square root of the frequency. The harmonic copper losses for a fundamental freqeuncy of f_1 will therefore be

$$P_{copper} = \sum_{k \neq 1} I_k^2 R \sqrt{k f_1} \qquad (4.23)$$

where f_1 is per unit fundamental frequency.

Williamson [43] has given an approximate breakdown of the harmonic losses, for a 15 kW motor for a quasi-squarewave inverter, from the analyses of other authors, as

stator winding	14.2%
rotor winding	41.2%
end region	18.8%
skew flux	25.8%

These figures are an indication of the order of magnitude of the harmonic losses compared to the copper losses. These losses are approximately 20% of the total losses for quasi-square waveform.

For the test machine used, rated at 1.1 kW, 380 V, 2.8 A, 3-phase, 50 Hz, using equation 4.22, and X = 18.5 Ω, R = 13.1 Ω.

$$P_{copper} = 0.2 \ THD^2$$

For the PWM technique used, THD < 1% therefore

$$P_{copper} = 0.2 \ W$$

Using the above as a general guide, the harmonic losses, are negligible compared to the other losses.

snubber circuits that are needed increase losses and therefore other devices are preferred.

The gate turn-off thyristor is an improvement on the silicon controlled thyristor, because it can be turned off from the gate by a negative current and therefore does not require the interruption of the load current. GTO's almost match thyristors in terms of ratings but are more expensive GTO's have a maximum operating frequency of 1 kHz and can be used for low carrier frequency PWM. New GTO's are becoming available with higher frequency ratings in the range of 20 kHz. GTO's require snubbers and have a high drive power requirement and hence have generally higher losses than transistors. They have become popular in applications where thyristors have been used and, being similar to thyristors, have been used for retro-fits [55, 56].

Power bipolar transistors are current controlled, minority carrier devices. Devices of 2000 A, 400 V ratings and 400 A, 1200 V ratings are available. Their high switching frequency and low on-state losses (0.2 V $V_{ce\ sat}$) make them very popular for PWM inverters. As the power transistor is a current controlled device, the drive is, ideally, a current source and therefore there are higher drive losses as compared to the drive requirements of MOSFET's, which are voltage controlled. The drive circuits can also be quite complex in comparison to that of MOSFET's. Power Darlingtons are now being used for high power applications at high frequencies [57]. Other power devices such as the static induction devices and IGBT's are being developed, which seek to combine the switching advantages of the MOSFET and the power handling capabilities of a transistors and thyristors, and so may become more popular [58].

Power metal oxide field effect transistors (MOSFET's) are relatively new and ratings are continually improving. Devices of up to 1000 V,10 A and 500 V, 50 A are available. They have excellent switching characteristics because, unlike ordinary transistors, they are majority carriers. Switching on/off times can be lower than 50 ns and so switching losses are low. They can be used for very high frequency applications up to 100 kHz. The drive requirements are minimal because they are voltage controlled devices with very high input impedance and so one can have a compact drive. The main disadvantage is the high on-resistance which leads to much higher conduction losses than ordinary transistors. They are also more expensive and have lower ratings than any of the other power devices. However, paralleling of MOSFET's can increase their power handling capabilities. MOSFET's have a positive temperature coefficient which means they do not suffer from the thermal runaway phenomenon of ordinary transistors. The ease of switching and the high frequency capabilities have made the MOSFET very popular for low power ultrasonic applications, including PWM [59].

Since a high carrier or switching frequency was being used in the PWM generation an inverter with devices capable of high frequency switching was essential. Therefore, the SCR thyristors were not suitable despite their high power capabilities. The bipolar transistor could have been used but the complexity of the drive requirements and design made it unsuitable. In view of the requirement for simplicity MOSFET's were chosen because of their excellent switching capabilities and their simple drive design. However, the power rating of the inverter would be limited to only a few kilowatts as MOSFET ratings are not very high. This is sufficient for the intended applications such as water pumping. IGBT's may be used in future if IGBT's with higher frequency rating become available.

5.2 Voltage Clamp Design

An equivalent circuit of a MOSFET is shown in figure 5.1.

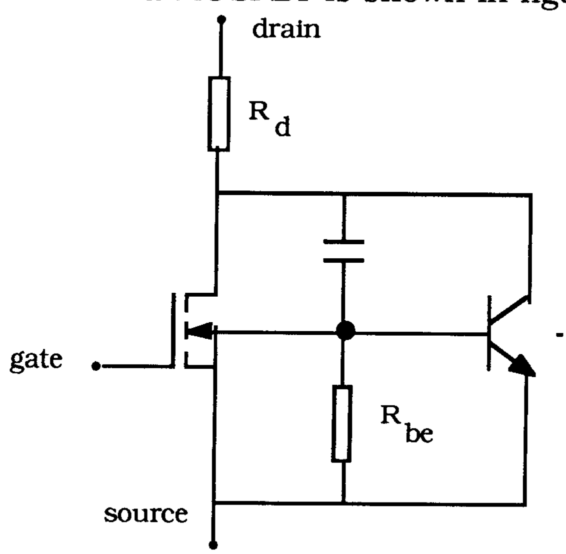


Fig 5.1 Model of MOSFET

If a high external dv/dt is applied between the drain and source enough displacement current could flow through the capacitor to generate a voltage drop across the resistor R_{be} sufficient to turn on the transistor causing device failure. In order to avoid this the body-drain diode is kept out of conduction. Furthermore, the body drain diode is generally too slow to be used as the free-wheel diode and so is kept out of conduction by inserting a diode in series with the MOSFET as shown in figure 5.2.

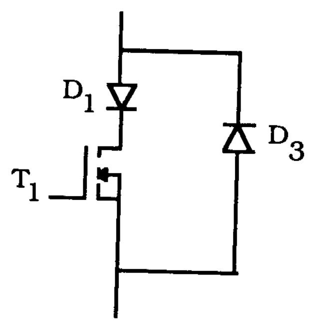


Fig 5.2 Free wheel diode connection

Since MOSFET's cannot handle voltage spikes above the maximum ratings it is necessary to include voltage clamping circuits especially when operating near the maximum ratings. This increases the reliability of the inverter though it also increases the losses. These losses are , however, relatively small since the snubber circuits are only for voltage clamping. A full discussion on the three major voltage clamping circuits for power MOSFET PWM inverters is given by Nair and Sen [60], who suggested the circuit shown in figure 5.3 which produces minimal losses. The RC series networks, across the supply, ensure that the voltage across the capacitors do not fall below V_{dc}. When T_1 is turned off the load current through the stray inductance L_1 does not fall to zero and so takes a path through D_4 and C_3.

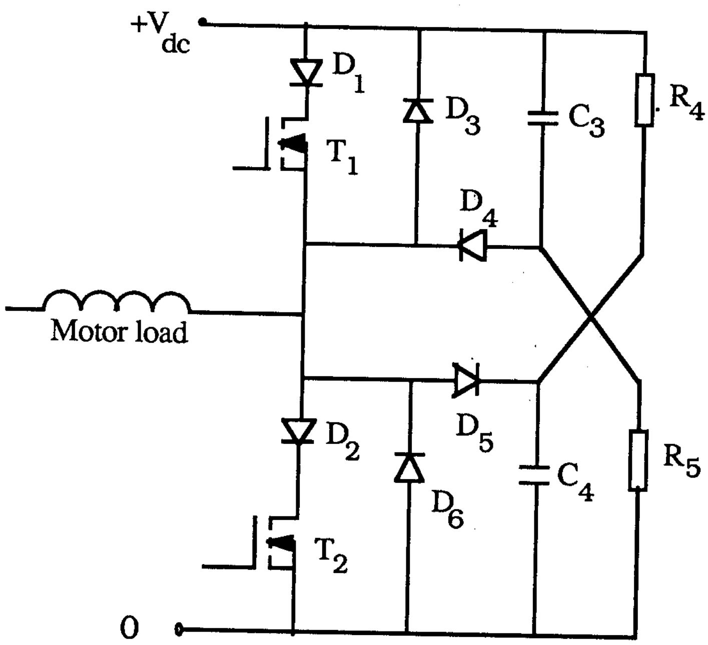


Fig 5.3 Voltage Clamping Circuit

Current through C_3 overcharges it and so turns D_6 on which then starts sharing the load current. The circuit may be seen as in figure 5.4.

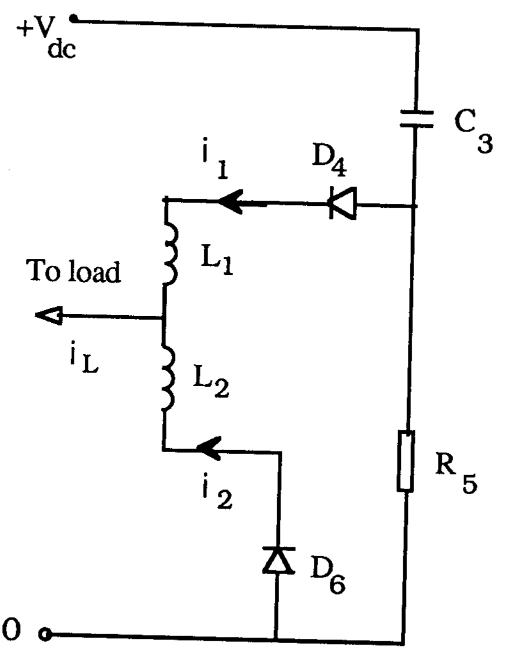


Fig 5.4 Stray inductance in inverter circuit

The voltage equation is

$$V_{dc} = \frac{1}{C} \int i_1 \, dt) + L_1 \frac{di_1}{dt} - L_2 \frac{di_2}{dt} \qquad (5.1)$$

where $C = C_3 = C_4$

Taking Laplace transforms and using the initial conditions $i_1(0) = i_L$ and $i_2(0)=0$,

$$i_1 = i_L \cos(\omega t) \qquad (5.2)$$

$$i_2 = i_L - i_1 \qquad (5.3)$$

where $\omega = 1 / \sqrt{(L1+L2)\, C} \approx 1 / \sqrt{2LC}$ and $L = L_1 + L_2$

Voltage drops across the inductances are given by

$$L_1 \frac{di_1}{dt} = -\omega L_1 i_L \sin(\omega t) \tag{5.4}$$

$$L_2 \frac{di_2}{dt} = -\omega L_2 i_L \sin(\omega t) \tag{5.5}$$

$$\begin{aligned} V_c(t) &= V_{dc} + \omega(L_1+L_2)\, i_L \sin(\omega t) \\ &= V_{dc} + \sqrt{\frac{2L}{C}}\; i_L \sin(\omega t) \end{aligned} \tag{5.6}$$

The maximum peak voltage across C_1 is given by

$$V_{c\,(max)} = V_{dc} + \sqrt{\frac{2L}{C}}\; I_{L(max)} \tag{5.7}$$

$$\Delta V = V_{c\,(max)} - V_{dc} = \sqrt{\frac{2L}{C}}\; I_{L(max)} \tag{5.8}$$

$$C = 2\,L \left(\frac{I_L}{\Delta V}\right)^2 \tag{5.9}$$

The capacitor then discharges through the resistor R to V_{dc}. If T is the switching period then the capacitor has to discharge within time T/2.

$$\Delta V\; e^{\frac{-T}{RC}} = 0.01\, V_{dc} \tag{5.10}$$

$$\Rightarrow \qquad \frac{T}{RC} = -\ln\left[0.01\,\frac{V_{dc}}{\Delta V}\right] = Z$$

$$\Rightarrow \qquad R \le \frac{T}{2ZC} \qquad (5.11)$$

The energy dissipated in the clamp circuit is the energy that the capacitor discharges and is given for one cycle as;

$$W = 0.5\ C\ \Delta V^2 \qquad (5.12)$$

This, it may be shown, can also be expressed as

$$W = L\ I_L^{\ 2} \qquad (5.13)$$

For the inverter designed the following values were used.
$L = 10\ \mu H$. $I_L=10\ A$. $V_{dc}= 400\ V$

$\Delta V = 10\ V$. $T = 1/23000 = 43\ \mu s$

$$\Rightarrow \qquad C = 20\ \mu F \text{ and } R \le 1.2\ \Omega$$

The values used were $C=1\ \mu F$ and $R=1\ \Omega$ which yields

$\Delta V = 45\ V$ and $T =4.8\ \mu s$

This was considered sufficient as the maximum voltage across the MOSFETs would be 445 V which is within the rated maximum of 500 V.

5.3 Description of Device Drivers

For the device drivers, the IR2110 high voltage bridge drivers integrated circuit was used. The IC uses a bootstrap method to provide the power required for the device in the upper arm which has to be "isolated" from the other arm. The lower arms do not require this. The connections for one totem pole arm is shown in figure 5.5. The driver is capable of sourcing 2 A and switching at speeds of up to 25 ns. It can be controlled directly by TTL or CMOS logic.

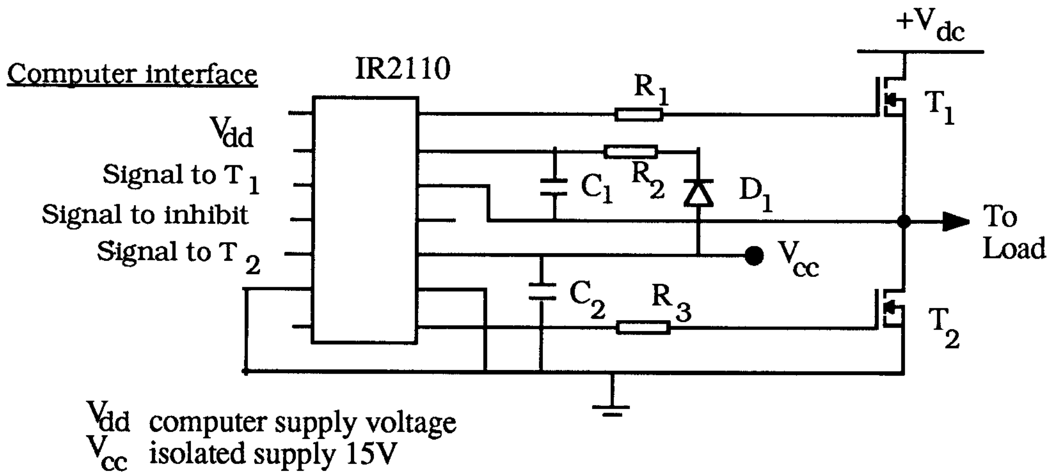


Fig 5.5 : One complete phase of the inverter

The charging diode D_1 must have a voltage withstand capability higher than the peak rail voltage. The resistor R_2 and capacitor C_1 determine the startup transient and are selected to suit high frequency switching.

$$5 R_1 C_{in} \le T_{ch} \quad \Rightarrow \quad R_1 \le \frac{T_{ch}}{5 C_{in}} \tag{5.17}$$

For the inverter designed

Q_g =130 nC, I_q = 500 μA, ΔV=2 V, C_{in} = 2.7 nF,

T_{ch} = 100 ns, f_m = 10 Hz,

$$\Rightarrow \quad C_1 \ge 175\ \mu F, \quad R_2 = 18\ \Omega, \quad R_1 \le 7.4\ \Omega$$

The values used were

C_1 = 370 μF, R_2 =8 Ω and R_1 = 4.7 Ω

5.4 Losses in MOSFET's

The major components of power loss in the inverter are

a) the conduction loss due to the on-resistance

b) the switching loss due to the rise and fall times

The conduction losses may be calculated from

$$P_{on} = I_{ds}^2 R_{ds} D \tag{5.18}$$

where I_{ds} is the load current

R_{ds} on resistance of the MOSFET

D is the duty ratio

For the three phase inverter the current in a phase is shared between the upper and lower arms and so D = 0.5

The switching loss may be calculated by approximating the rise and fall as straight lines and so

$$P_s = V_{ds} I_{ds} t_s f_c \qquad (5.19)$$

where : V_{ds} the on state voltage across the MOSFET

t_s sum of the rise and fall times for one cycle

f_c the switching frequency

Total power dissipated in one MOSFET

$$P_t = P_{on} + P_s$$

$$P_t = I_{ds}^2 R_{ds} D + V_{ds} I_{ds} t_s f_c \qquad (5.20)$$

with I_{ds} = 10 A, R_{ds} = 0.7 Ω, V_{ds} = 400 V, t_s = 150 ns and f_c = 23 kHz

$$P_t = P_{on} + P_s$$

$$= 35 + 13.8 \approx \underline{48.8\ \mathrm{W}}$$

The electrical equivalent of the thermal circuit is shown in figure 5.6 and the thermal impedance of the heat sink is calculated from the following equation

$$R_{øsa} = \frac{T_j - T_a}{P_t} - (R_{øjc} + R_{øcs}) \qquad (5.21)$$

Where $R_{øsa}$, $R_{øjc}$, $R_{øcs}$ are the thermal resistances and T_j , T_a are the temperatures.

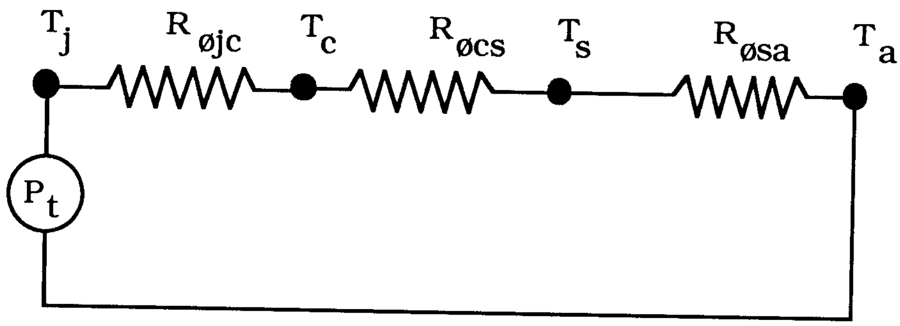


Fig 5.6 Thermal Equivalent Circuit

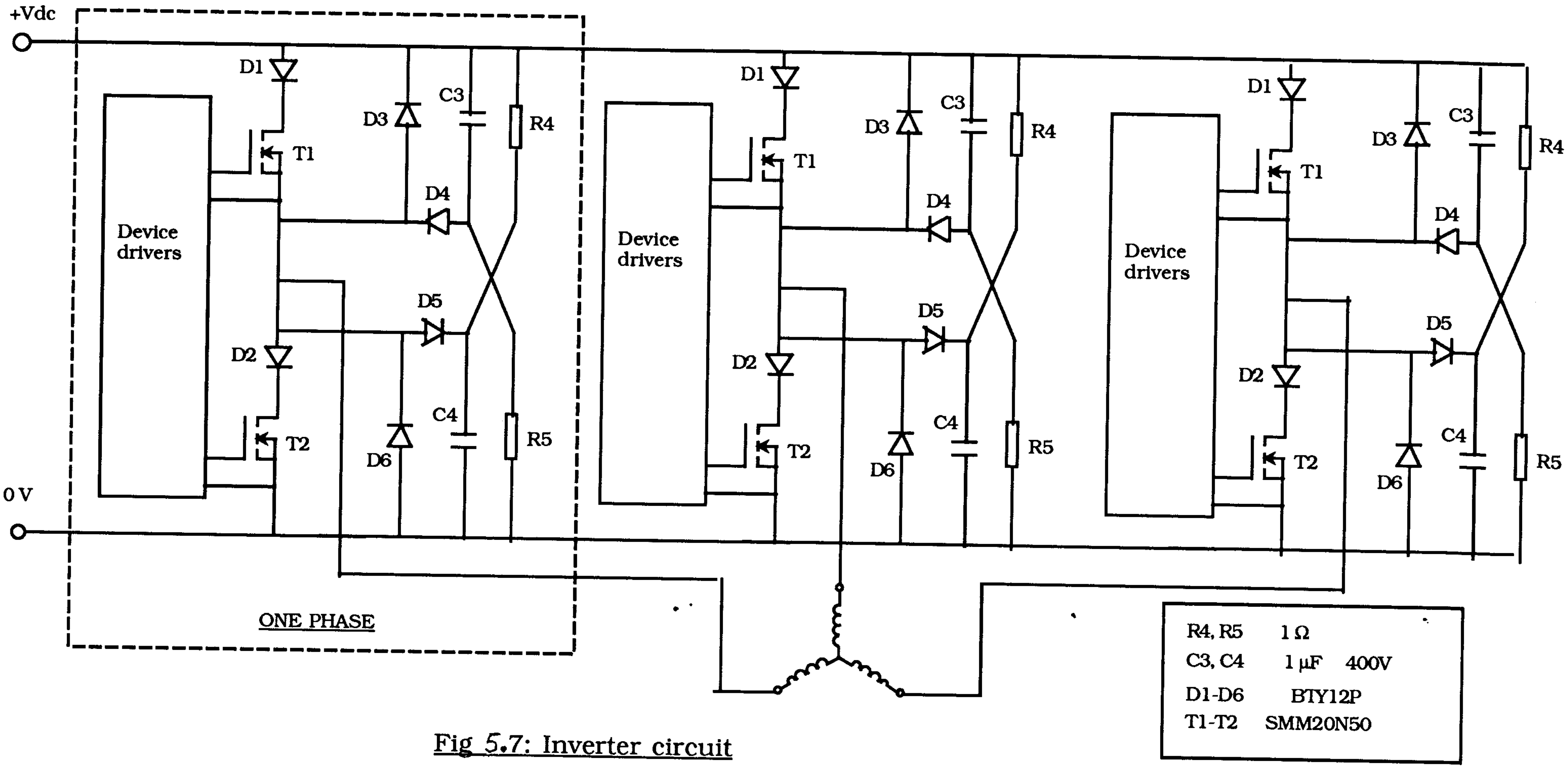
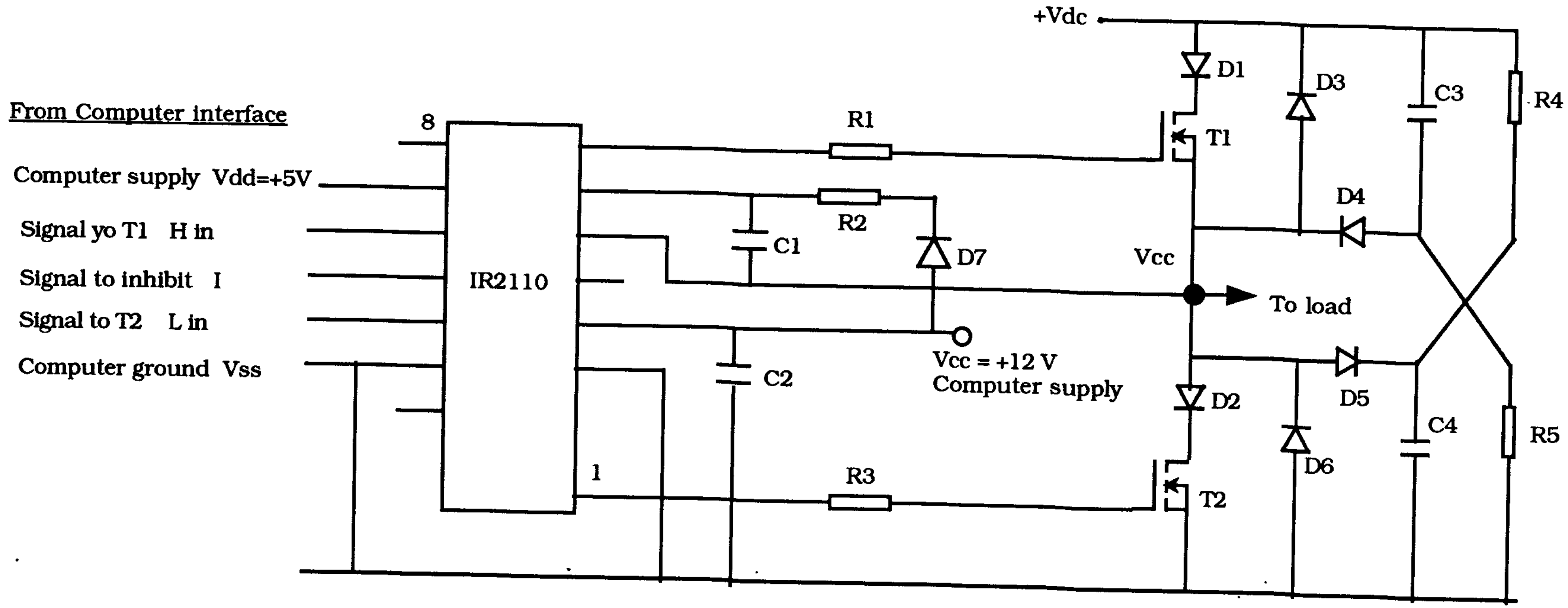


Fig 5.7: Inverter circuit



R1, R3	4.7 Ω	C1	370 μF	100V	D1-D6	BTY12P
R2	8 Ω	C2	47 nF	250V	D7	IN4004,800V
R4, R5	1 Ω	C3, C4	1 μF	400V		

Fig 5.8: Driver Circuit for one phase of inverter

Chapter 6

Design of Control Circuit

6.1 Description of Three Phase PWM Generator

The MA818 is a digital three phase PWM waveform generator that can be controlled by a microprocessor via the MOTEL interface. It may be operated as a stand alone or as a microprocessor peripheral where the microprocessor is used to set the operating parameters only. It uses an asynchronous regular sampling method to generate PWM. The modulating waveform is read from an external EPROM allowing variation in the modulating function. The carrier frequency may be selectable up to 24 kHz and power frequencies up to 4 kHz are obtainable. The device generates all the pulses required for a 6 - pulse inverter and also adjusts the pulse delay and the minimum pulse width thus reducing on external circuitry. The output phase sequence can also be changed to provide direction control. Separate control of the amplitude and frequency enables the user to implement variable-speed control of AC machines with user defined V/f characteristics.

The MA818 has two 24 bit internal registers for initialization and control data that are loaded via the MOTEL interface. The initialization register is loaded before the motor is operated and deals with data not normally changed during motor operation. The control register deals with operations that are frequently changed such as power frequency. The MOTEL interface is suited to standard INTEL and MOTOROLLA microprocessors and is 8 bit wide. Data to the registers is transferred by writing to three 8 bit temporary registers and then writing to a "dummy register" that transfers the data to the 24 bit registers.

The initialization register sets the following :

Carrier frequency
Power frequency range: sets maximum power frequency
Pulse delay time: also known as minimum dwell time.
Pulse deletion time: Minimum pulse width of PWM.
Counter Reset: Gives 50% duty cycle (Not used)

The control register controls the following parameters :

Power frequency: Up to 12 bit resolution.
Forward / reverse: Phase sequence adjustment.
Amplitude: Sets the amplitude (modulation depth).
Over-modulation: Used to get modulation of above unity.
Output inhibit: Places all the outputs to low.

The registers are written to by first filling three temporary registers Reg0, Reg1, and Reg2. Transfer of data to either the control or initialization registers is by writing to "dummy registers" Reg3 and Reg4 as shown in table 6.1.

AD0	AD1	AD2	Function
0	0	0	Reg0
0	0	1	Reg1
0	1	0	Reg2
0	1	1	Reg3 control data
1	0	0	Reg4 initialization data

Table 6.1 MA818 registers

The waveform is stored,in a 2K x 8 bit standard EPROM, as seven hundred and sixty eight 8-bit samples covering 0° to 180° giving an angular resolution of 0.23° degrees. The MA818 constructs the remaining 180° to 360° by assigning negative values to the same samples. The waveform is assumed to have quarter wave symmetry. The 8-bit samples are stored as two separate nibbles as shown in the memory map of the EPROM, figure 6.1. Thus, only four data lines need be used by the MA818 to access the samples. A typical configuration of the MA818 is shown in figure 6.2. An 8-bit microprocessor loads the MA818 with initialization and control data. The MA818 then functions independently of the microprocessor, accessing the EPROM and generating the PWM outputs.

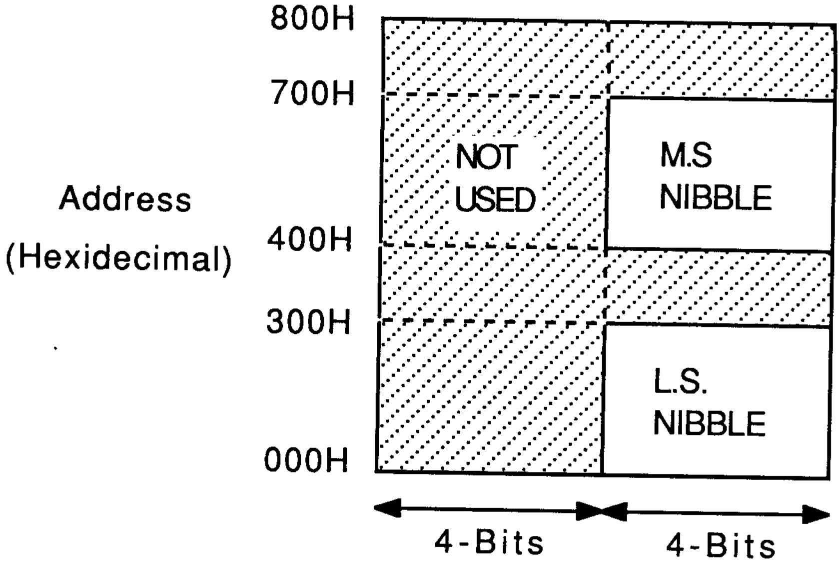


Fig 6.1 Memory Map of EPROM

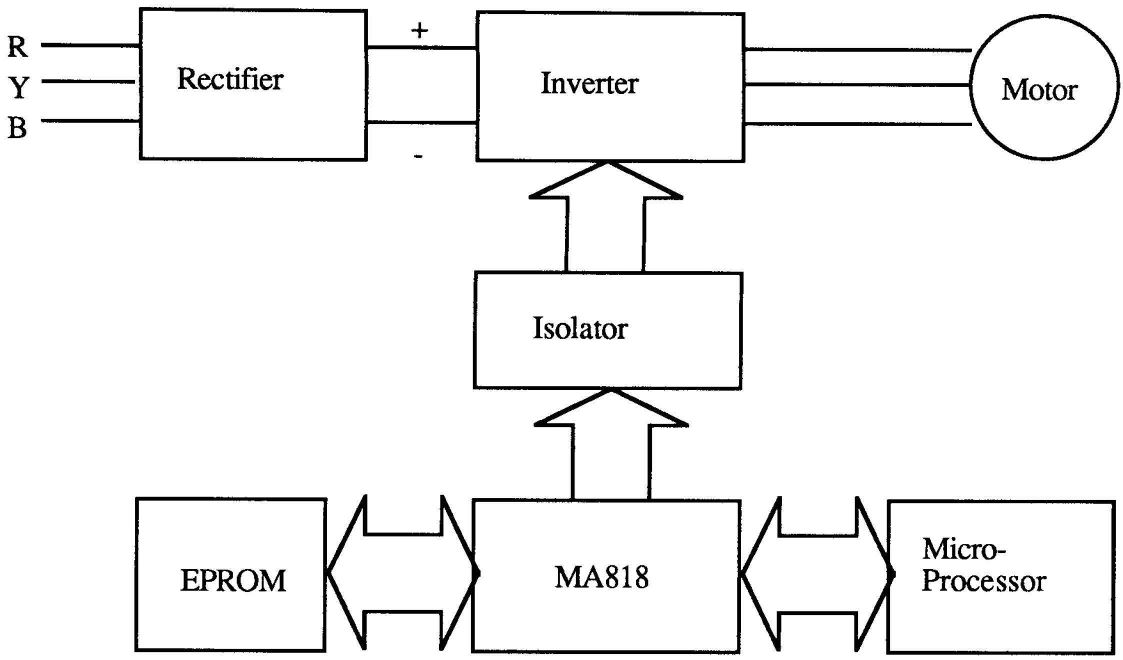


Fig 6.2 Typical System Configuration using the MA818

6.2 Circuit Description

For purposes of study and production of the prototype, the MA818 was interfaced to a PC. The ROM or EPROM storing the modulating waveform was replaced by static RAM that is controlled by the PC. This would enable testing of various waveforms. A block diagram of the interface card containing the MA818 is shown in figure 6.3 and the circuit diagram is in Appendix A. The MA818 is connected to the computer bus via the address / data multiplexer and buffers. The RAM is selected by either the computer or the MA818 using the data and address selectors. The PWM output is buffered and the signals carried over a cable to the inverter circuit.

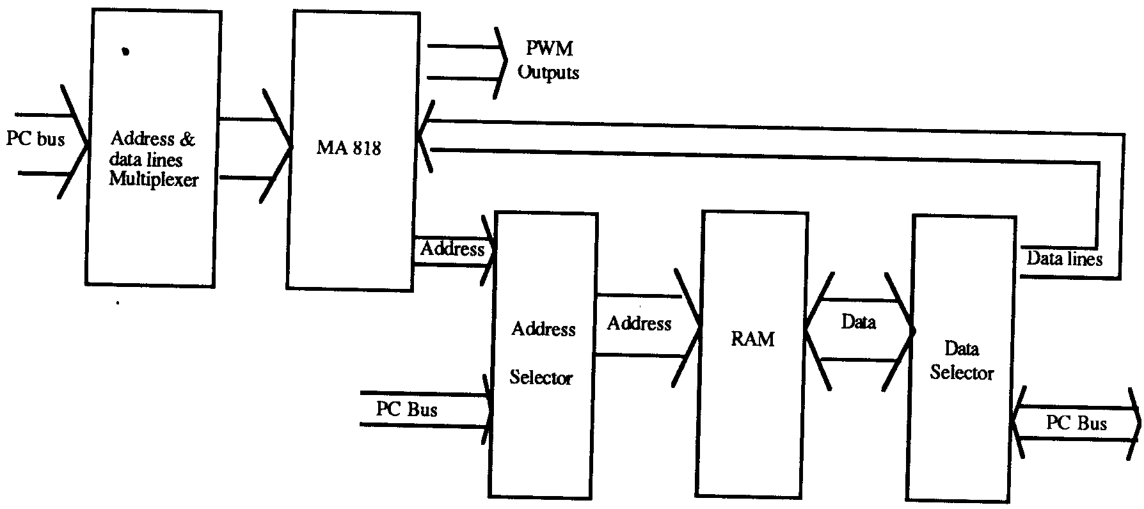


Fig 6.3. Block diagram of PC controlled MA 818

6.2.1 Register Communication

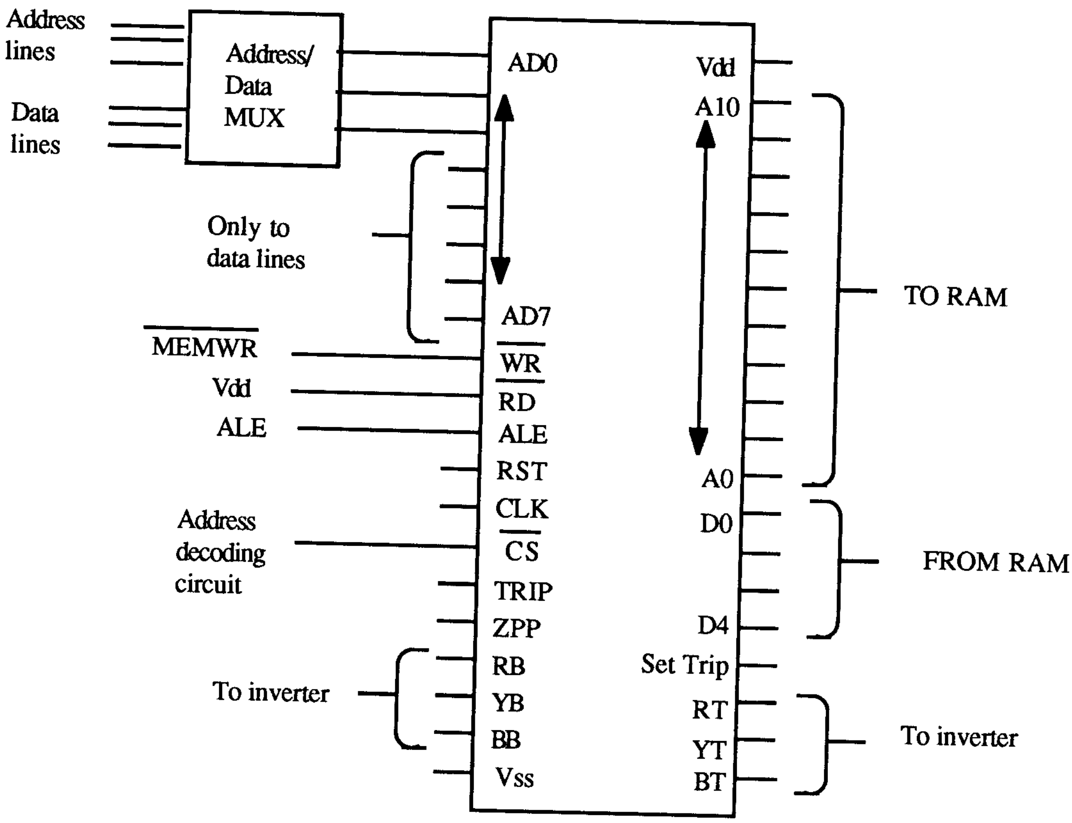


Fig 6.4 Connections to the MA818

The COPAM computer that was used has an 80286 microprocessor and an AT bus. Since the address and data lines are separate, it was necessary to multiplex the lines in order to be compatible with the MOTEL interface of the MA818. There are five registers to be addressed and so only three address lines had to be multiplexed. The remaining data lines were connected directly to the MA818. The WR input of the MA818 was connected to the MEMW (memory write) signal so that the registers could be addressed as memory locations.. The RD line was connected to high since the registers are not readable. A chip-select was generated from an address decoding circuit to select the MA818.

6.2.2 RAM Communication

Since only 4-bits of an address location were being used to store the waveform, two 1K x 4 bit static RAM modules were used. The MEMR and MEMW from the computer were used so that the RAM could be referenced as memory by the computer.

6.3 Interface Software

For the control program to communicate with the interface board some accompanying low-level software interface routines were written that could be called by the program. The programmer, therefore, communicates only with the software interface and does not have to deal with the hardware. The routines were written in C-Language as it has both the attributes of a high level language and of a low level language. The routines could, however, be written in assembly language for speed sensitive applications and be called by any high level language program as object code. A C-language implementation of these low-level routines is described below.

6.3.1 Communication with MA818 Registers

The initialization and control registers are addressed by the computer as specific memory locations. The address locations in memory are chosen from the available address space given by the memory map of the computer. The address decoding circuit on the interface card is then set to the address using hardware switches. Writing to memory using C language is easily achieved using pointers. The physical address is in Hexi- decimal format.

<u>C-program for Addressing MA818 registers</u>

```
void writing()
{
unsigned char far        *pt;
        pt=0xd0000800;          /* Address of register Reg0          */
        *pt= 1;                 /* writing value 1 to the register   */
                                /* Reg0 on interface board using a pointer */
}
```

In order to facilitate the easy manipulation of data, bit-fields and unions are used as shown below

Bit-fields for easy access of Register

```
union reg1{
    struct{                                              /* bit fields          */
        unsigned int    car_freq:     3;  /* Bits for carrier   */
        unsigned int    dontcare:     2;  /* Unused bits        */
        unsigned int    freq_range:  3;  /* Bits for Range     */
        unsigned int    zeros:        8;  /* Unused bits        */
    }p;
    struct{
        unsigned char  i;                 /* Accessing  first 8-bits   */
        unsigned char  d;                 /* Accessing  second 8-bits*/
    }r;
}reg1;
```

The variables reg1.r and reg1.p refer to the same memory location, that is they contain the same data but differ only in the way they are accessed by the software. The carrier frequency bits and frequency range bits are accessed by the software as unsigned integer variables reg1.p.car_freq and reg1.p.freq_range respectively. The data to the MA818 is, however, in the form of a byte or char(acter) type and so data can be sent to the MA818 as the char variable reg1.r.i which contains the information in reg1.p.car_freq and reg1.p.freq_range.

Routine for writing data to MA818 Register

```
void  writing()
{
unsigned char far         *pt;
      pt=0xd0000801;           /* Address of register reg1         */
      *pt=reg1.r.i;            /* writing  contents of reg1.r,i to */
}                              /* the MA818 register Reg1          */
```

6.3.2 **Communication with the RAM**

The waveform is stored in RAM and is addressed as memory locations by the computer, using pointers and C- program memory handling routines. As the nibbles are stored in different memory locations the routine must be able to separate them. The waveform may be modified by varying the contents of the matrix ROM[780]. The RAM can be read in a similar manner.

Routine to write to the RAM

```
void  write_mem()
{
unsigned char far  *pt1, *pt2;
unsigned char      value;
float              ROM[780];        /* data to be sent to  MA818
     */
int                j;
      pt1=0xd0000000;        /* address of low order nibbles      */
      pt2=0xd0000400;        /*  address of high order nibbles    */
      for(j=0;j<=767;j++){
            value = (unsigned char)ROM[j];
            *pt1=value;        /* writing low order nibble         */
            *pt2=(value>>4);  /* writing high order nibble        */
            pt1++;   pt2++;    /* next addresses                   */
      }
      memset(pt1,15,255);  /* filling remaining locations      */
      memset(pt2,15,255);  /*    with FF                       */
}
```

6.4 Algorithms

The control program consists of five main sections, the Initialization, Control, RAM waveform, file routines and user interface routines. The functional diagram of the initialization routine is shown in figure 6.5 and mainly deals with the fixed parameters of the inverter and the MA818 that will not be under the control of the user.

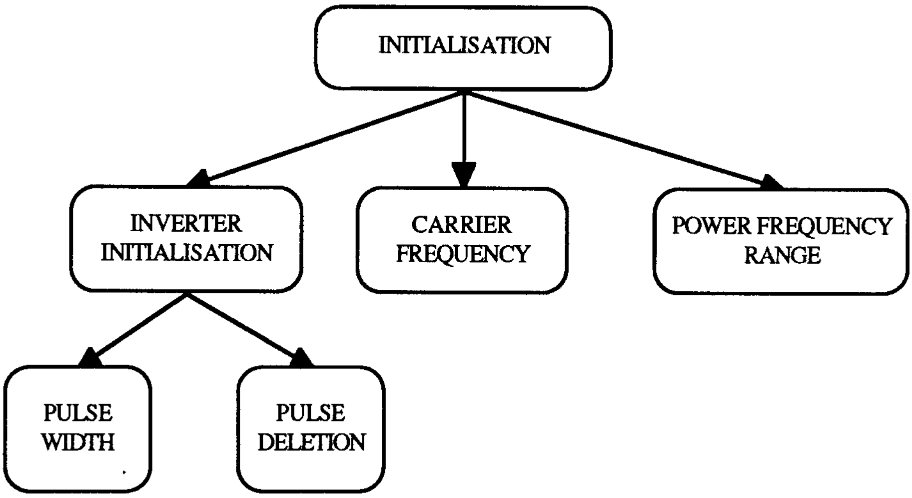


Fig 6.5 Functional diagram of the initialization routines

The control routines, figure 6.6, deal with the running of the motor. These routines control the V/f characteristic, the acceleration and deceleration of the motor.

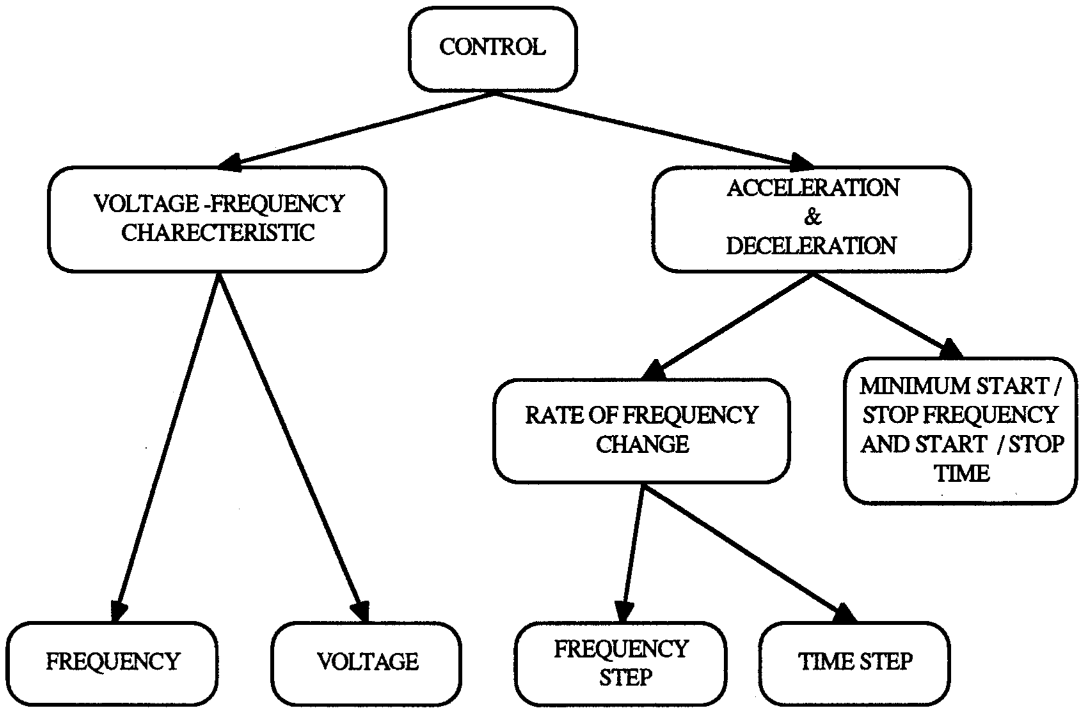


Fig 6.6 Functional diagram of the control routines

The routine for generating the waveform calculates the values of the modulating waveform and stores them in the matrix ROM[768] which is then sent to the interface software routine. The file routines are required to keep record of the status of the MA818 so that when the program is exited and then re-entered the status can be loaded. This facility may be extended to keep a record of events that some applications may require such as the number of starts over a given period.

The user interface handles the communication between the user and the control program. This may be a complex graphical user interface GUI for users with video display unit, or simple keyboard or potentiometer type. This routine may be written according to the type of user interface available and for the purposes of this study a very basic keyboard command interface was used. The main control routines are the acceleration / deceleration and V/f routines described below

The routines are written in the C-language due to its modular approach and ease of manipulating bit-fields. The modular approach allows easy maintenance of software in that program modules can be changed virtually independently of each other. A main program calls the routines in turn depending on the desired control action.

6.4.1 Software Implementation of Flux Control

The frequency of the output is represented by a 12 bit binary number and the maximum frequency, which is the frequency range set by the initialization register, corresponds to 4095. The amplitude of the output voltage is also represented by a 9-bit binary number with a modulation depth of unity corresponding to an amplitude count of 255.

$$\text{modulation depth} = \frac{\text{Output line voltage (RMS)}}{\text{DC rail voltage * Utilisation factor}}$$

For a pure sinusoidal modulating function, the inverter output line-line rms voltage resolution corresponding to amplitude count of one is

$$\text{Voltage resolution} = \frac{\text{DC rail voltage * Utilisation}}{\sqrt{2} \times 255}$$

If the V/f ratio is not boosted it is sufficient to multiply the frequency count by a constant to obtain the amplitude required. The routine below uses the constant freq-to-amp to obtain the amplitude count which is then separated to appropriate registers by bit manipulation.

Routine to set output voltage

```
void set amplitude()
{
unsigned int        i;
float               freq-to-amp;        /* conversion factor */
    freq-to-amp=0.06;
    /* modulation = power frequency  x  conversion factor */
    /* the result is converted to unsigned integer format */
amplitude = ( unsigned int) ceil( freq-to-amp * c.p.power_freq);
    /* writing  to the appropriate bits */
    ct.p.amp=amplitude & (unsigned int)255;
    c.p.overmod=(amplitude & (unsigned int)511)>>8;
}
```

For inclusion of the voltage boost only this segment need be changed by addition of a constant term, V_{boost}, corresponding to the boost voltage. This flexibility may be extended, as indicated earlier, to more complex V/f characteristics involving the machine parameters. The program developed computes the ideal V/f characteristic for the motor from motor parameters, based on the model shown in figure 2.1, and converts them to unsigned integer format and stores them in a matrix. For each frequency count, a corresponding amplitude count is read. However, it must be noted that fine control of the flux is limited by the voltage resolution of about 1-2 volts.

6.4.2 Software Implementation of Acceleration

The software that has been developed enables the user to control the rate of acceleration by varying the frequency step size and the delay between changes in frequency. A flow chart of the routines dealing with acceleration and deceleration is shown in figure 6.7. The minimum frequency step size is determined by the frequency resolution of the MA818 which is given by

$$\Delta f = \text{Frequency range} / 4096 \qquad (6.4)$$

$$\approx 0.015 \text{ Hz} \qquad \text{for 0-60 Hz range}$$

The frequency is given by the product of the frequency count (an integer up to 4096) and frequency resolution.

$$f = \text{frequency count} \times \Delta f$$

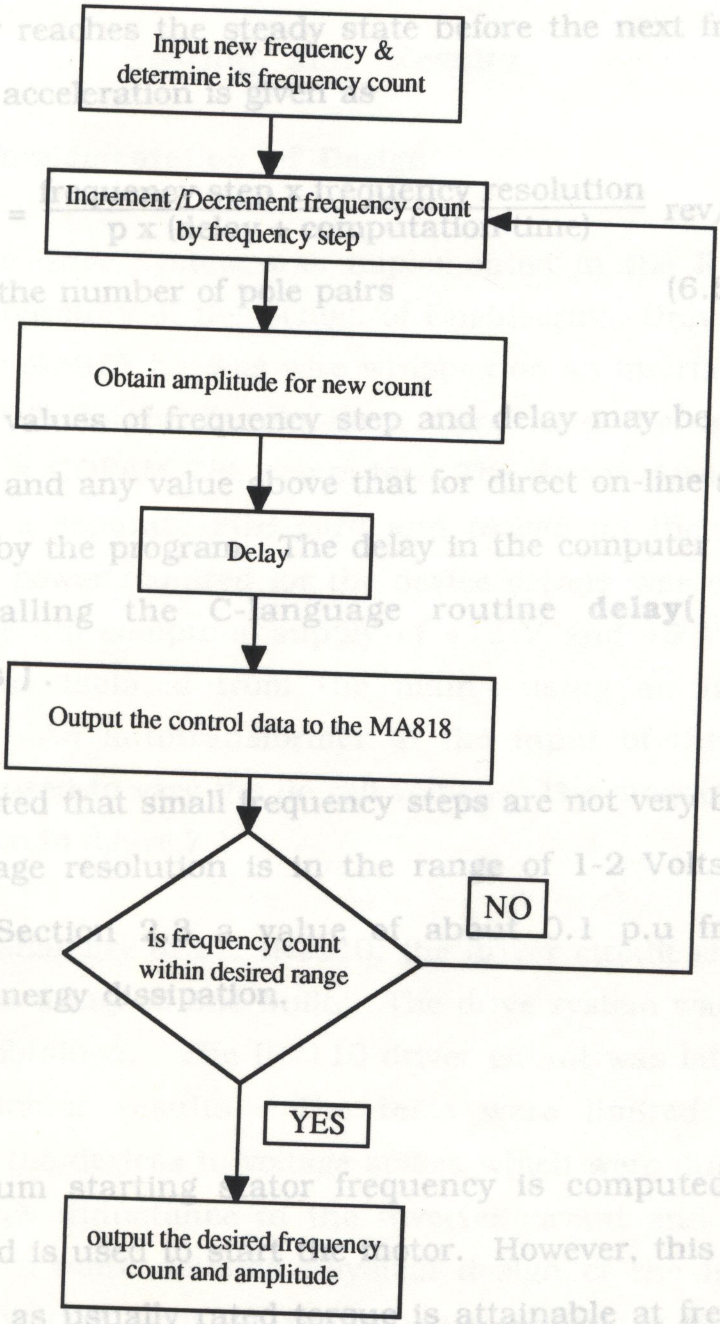


Fig 6.7 Flow chart of the acceleration routine

If the motor reaches the steady state before the next frequency change, the acceleration is given as

$$\text{acceleration} = \frac{\text{frequency step x frequency resolution}}{\text{p x (delay + computation time)}} \ \text{rev/s}^2$$

where p is the number of pole pairs (6.5)

The default values of frequency step and delay may be changed by the user and any value above that for direct on-line start will be rejected by the program. The delay in the computer is set by software calling the C-language routine **delay(** integer milliseconds **)** .

It will be noted that small frequency steps are not very beneficial as the voltage resolution is in the range of 1-2 Volts and as shown in Section 2.3 a value of about 0.1 p.u frequency minimizes energy dissipation.

The minimum starting stator frequency is computed by the program and is used to start the motor. However, this may not be required as usually rated torque is attainable at frequencies much lower than the starting frequency and the user may have control over this up to the minimum starting frequency.

Chapter 7

Testing and Results

7.1. Implementation of Design

The complete drive system was implemented in the Electrical Machines Laboratory of the School of Engineering, University of Zambia. The MA818 IC was wire wrapped on an interface card, the PCL-750 Prototype Development Card, and the card was plugged into a COPAM 286 computer. The device drivers were mounted on a separate PCB card and placed on the inverter circuit. The power required for the device drivers was obtained from the internal computer supply of +12 V and +5 V. The computer was isolated from the mains using an isolation transformer. An autotransformer at the input of the bridge rectifier was used to vary the dc rail voltage. The diagram of the setup is shown in figure.7.1.

Due to unavailability of the IR2110, the driver circuit shown in figure 7.2 was designed and built. The drive system was tested and results obtained. The IR2110 driver circuit was later built and gave similar results. The tests were limited by the sensitivity of the devices to voltage spikes which were due to the excessive stray inductance in the inverter circuit and can be corrected by a more detailed physical design of the inverter. Also, the dc voltage from the single phase rectifier is limited to 342 V, limiting the maximum inverter output voltage to 242 V and therefore the inverter frequency to 32 Hz for constant flux operation. The tests were, however, sufficient for the purposes of the research. The measurement of torque pulsations would have been very useful but unfortunately was not done due to non-availability of an accelerometer.

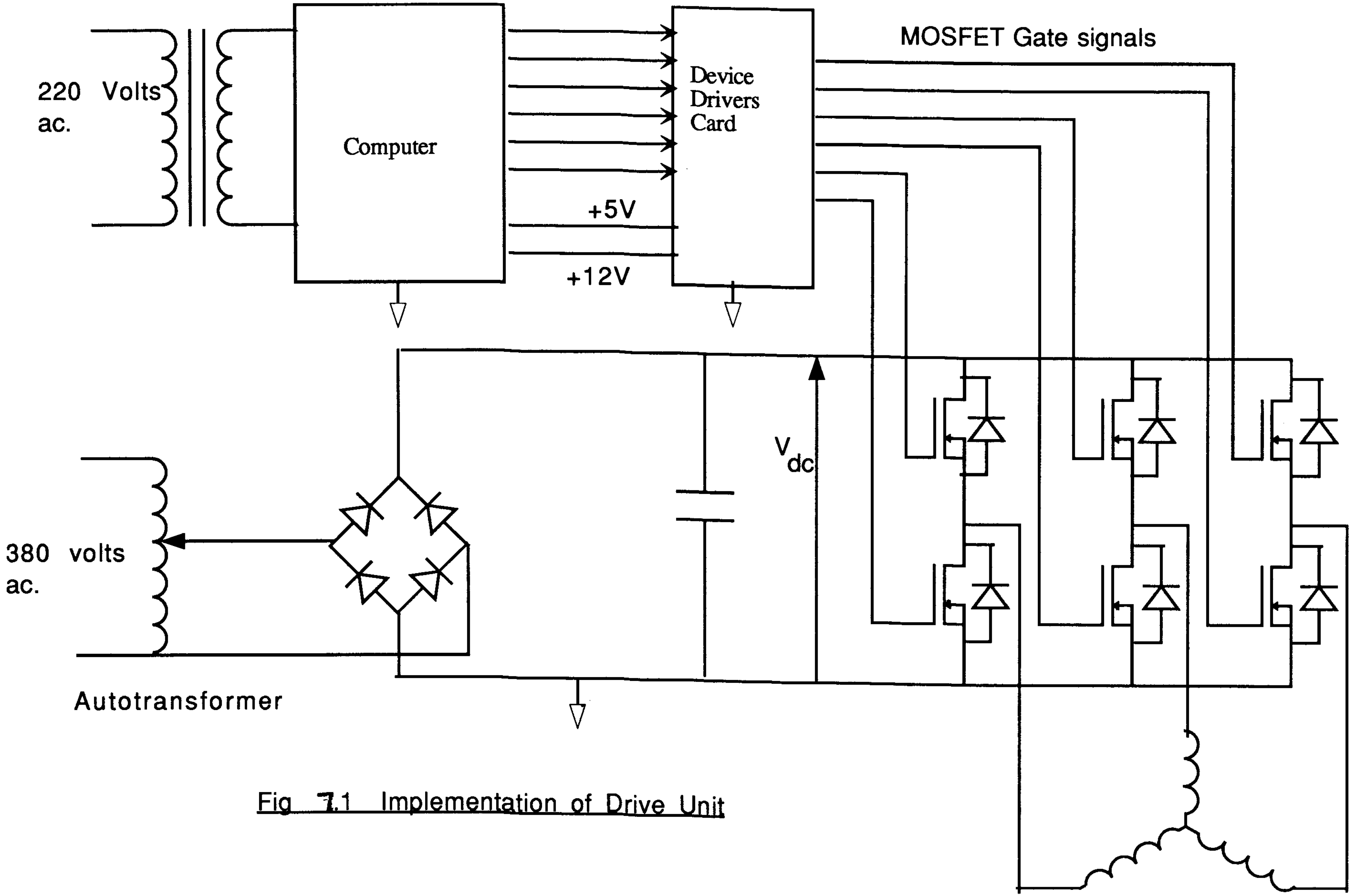


Fig 7.1 Implementation of Drive Unit

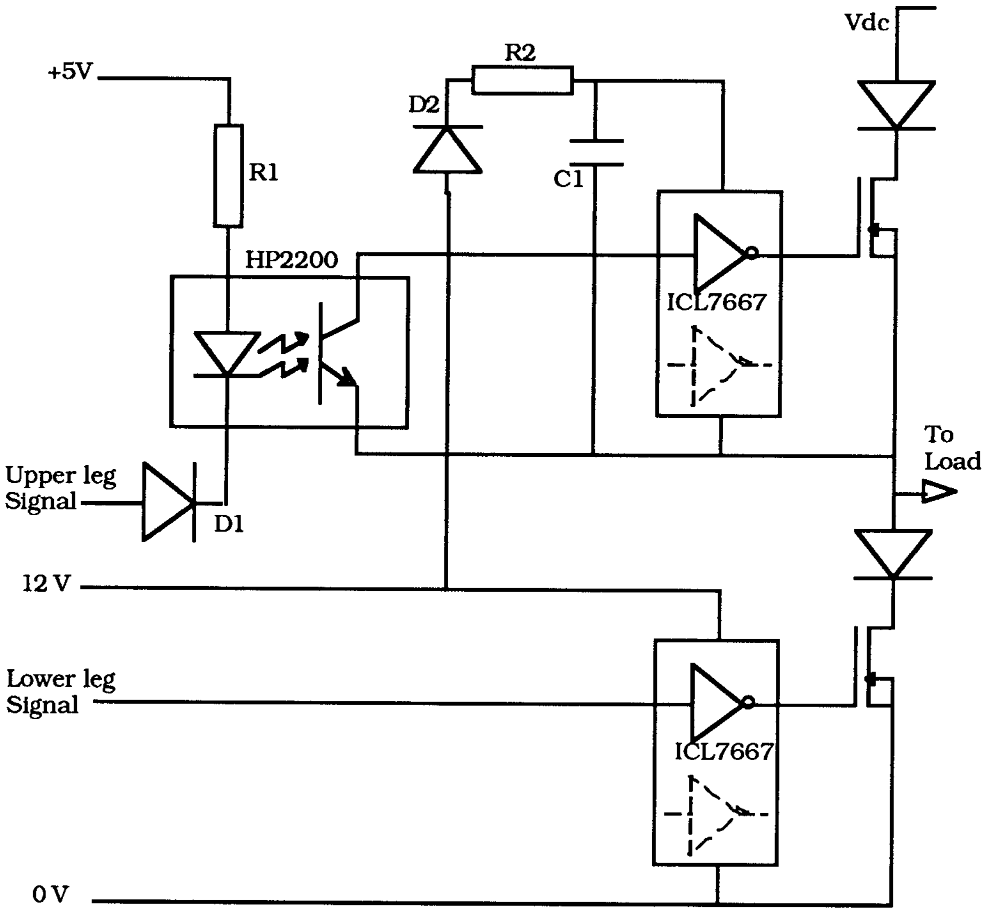


Fig 7.2 Driver circuit used for drive tests

7.1.1 Motor Ratings

The motor used in the tests was a 3-phase squirrel cage induction motor rated as follows:

1.1 kW,	1410 rev/min,	50 Hz
Volt Δ/Y 220/380 V	Starting Current 29/17 A	
Current 4.9/2.8 A	Efficiency 76%	cos ø 0.78
Class B	44 kg	

7.1.2 Equivalent Circuit Parameters of the Motor

The single-phase equivalent circuit shown in figure 2.1 was used to determine the voltage and frequency applied to the motor. The parameters of the motor were determined from the No-Load and Locked rotor tests with the assumption that they are constant with frequency. The ratio between X_1 and X_2 is chosen based on experience and ratios commonly used are 4/6 and 3/7. A ratio of 3/7 was used. [61]. The following are the values used.

$R_m = 1210\ \Omega$	$R_1 = 5.8\ \Omega$	$R_2 = 7.27\ \Omega$
$X_m = 121.5\ \Omega$	$X_1 = 5.56\ \Omega$	$X_2 = 13\ \Omega$

7.2 Measurement Technique

The current spectra were obtained using a Hewlet Packard Spectrum analyzer, 3588A. The line current was measured across a shunt resistor of 0.1 Ω with a 1 MΩ oscilloscope probe. The accuracy of the spectrum analyzer below 10 Hz is, however, not guaranteed by the manufacturers .

The fundamental component of voltage was also measured using the spectrum analyzer. Since the maximum input voltage of the spectrum analyzer is 230 mVrms, a 100:1 resistive potential divider was used followed by a 10:1 using an oscilloscope probe.

The spectrum analyzer was isolated from the mains using an isolation transformer. Use of other means of isolation, namely opto-couplers, differential amplifiers, and current transformers, were found to be inadequate due to the high frequency components in the currents and voltages.

The current harmonic distortion, IHD, was calculated using the formula

$$\mathrm{IHD} = \frac{1}{I_1}\left(\sum_{k\neq 1}^{\infty} I_k^2\right)^{\frac{1}{2}} \times 100\% \qquad (7.1)$$

where I_1 is the fundamental current and I_k is the rms currents of higher frequency than the fundamental, including non-integer harmonics, below the 21st. The IHD is different from the THD which is calculated on the basis of the ideal voltage waveform and simplified model of the motor.

7.3 Resistive and Inductive Loads Test

The drive system was initially tested with resistive and inductive loads. At a rail voltage of 20 V and dc line current of 0.46 A, figures 7.3 and 7.4 show typical voltage waveforms for a resistive load and figure 7.5 and 7.6 show typical current and voltage waveform for an inductive load.

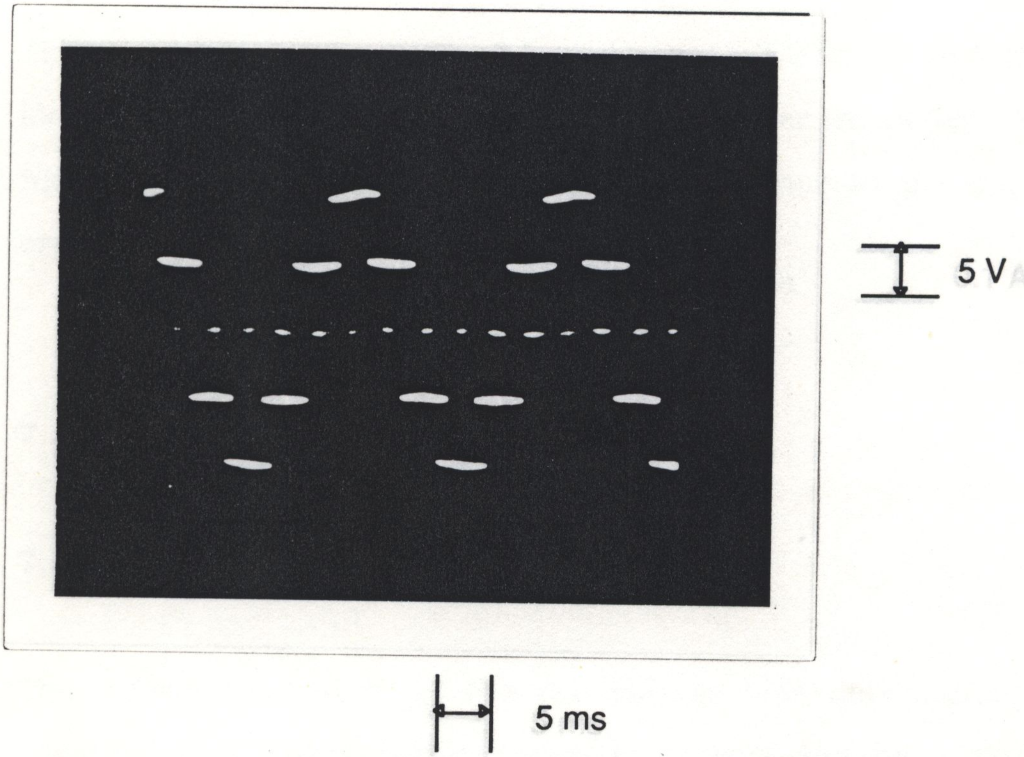


Fig 7.3 Phase Voltage for Resistive Load

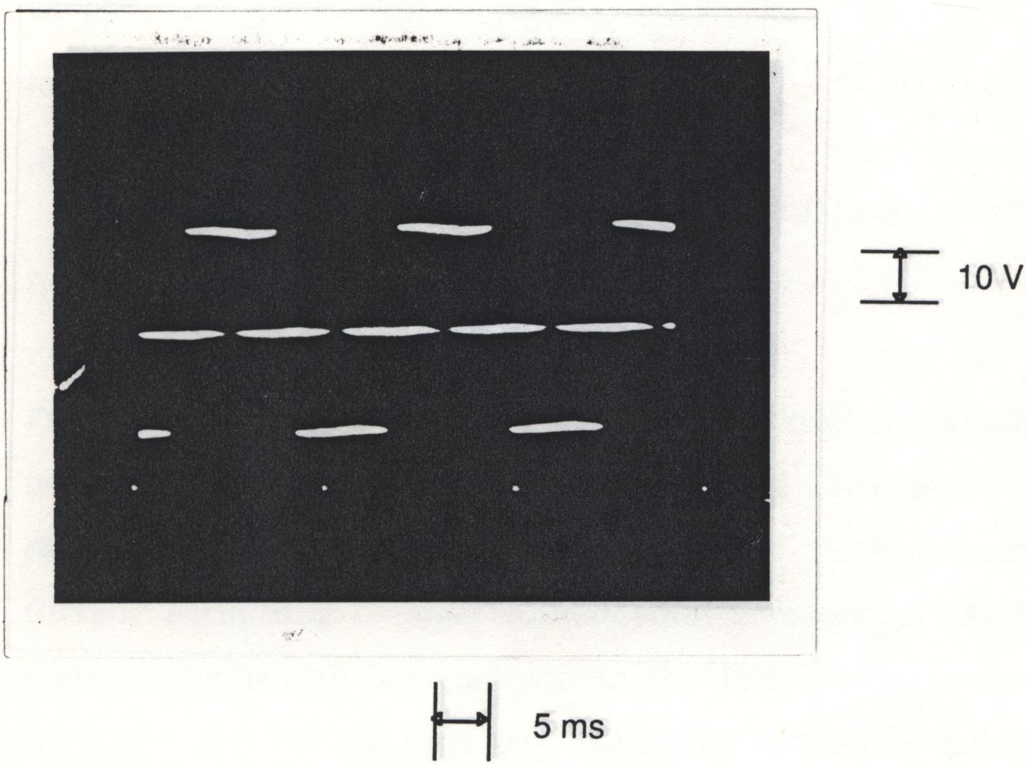


Fig 7.4 Line Voltage for Resistive Load

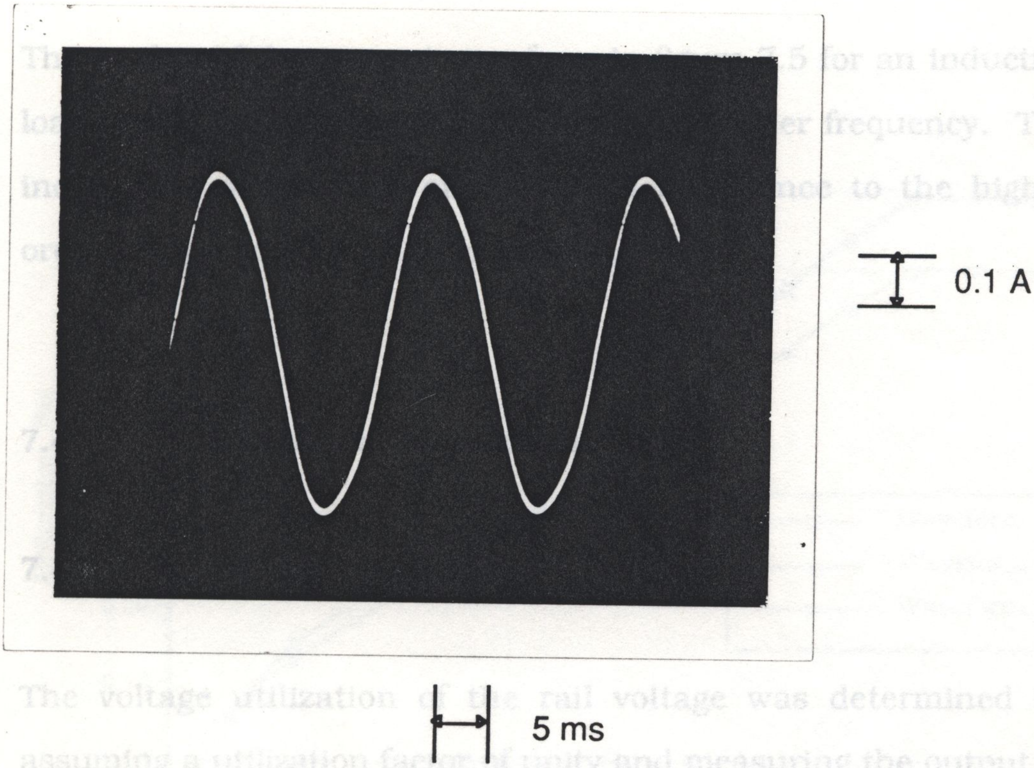


Fig 7.5 Line Current for Inductive Load

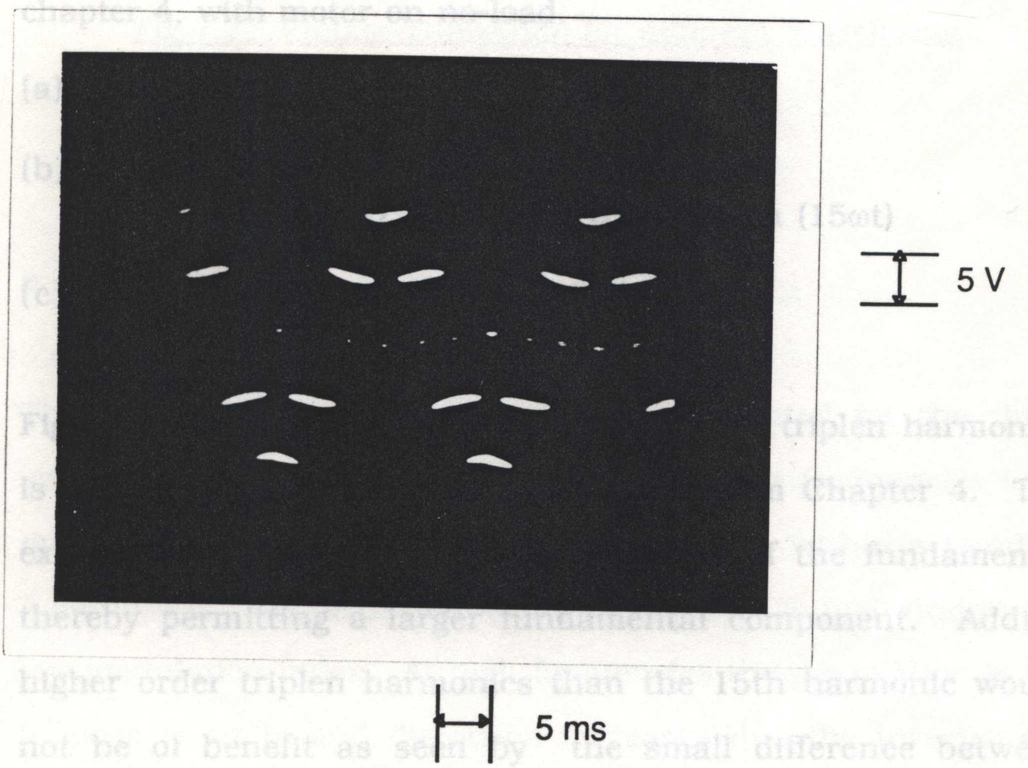


Fig 7.6 Line Voltage for Inductive Load

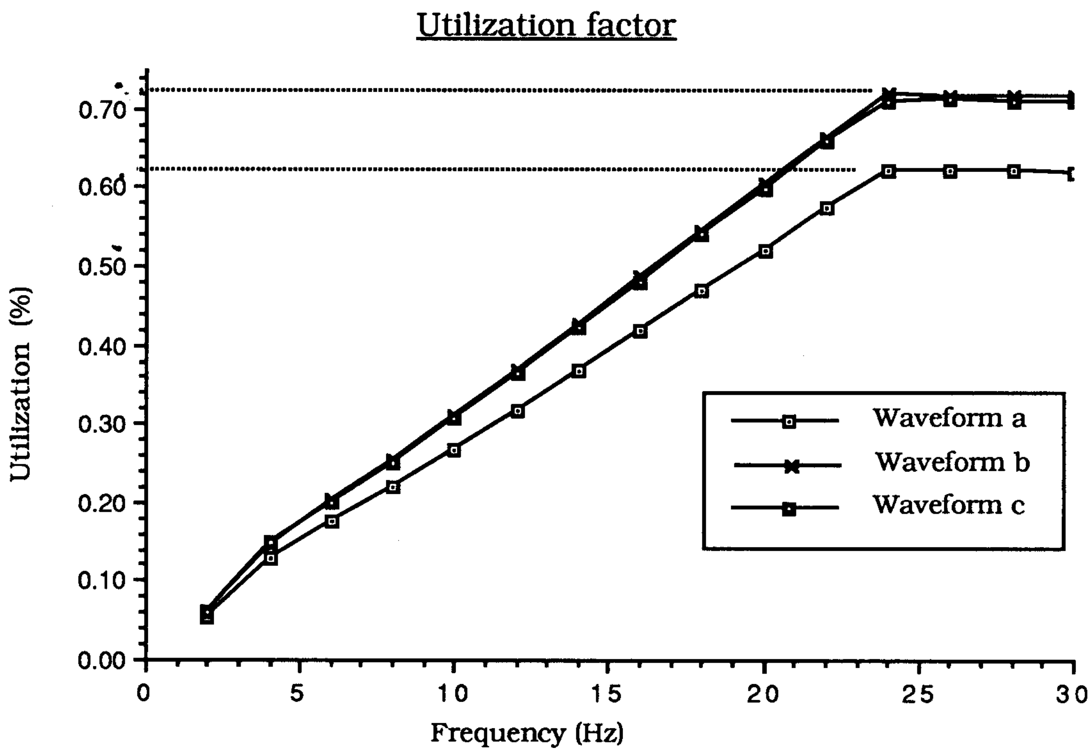


Fig 7.7. Voltage Utilization for different Waveforms

7.4.2 Current Distortion

The total current distortion was also computed for the three waveforms from measured spectra at different frequencies with motor on no-load. The current distortion lies within a band of 1.5% to 4%. The distortion at low frequencies is high due to the low modulation index. At high frequencies the modulation index is close to unity and distortion increases due the inverter not being able to reproduce the small pulses.

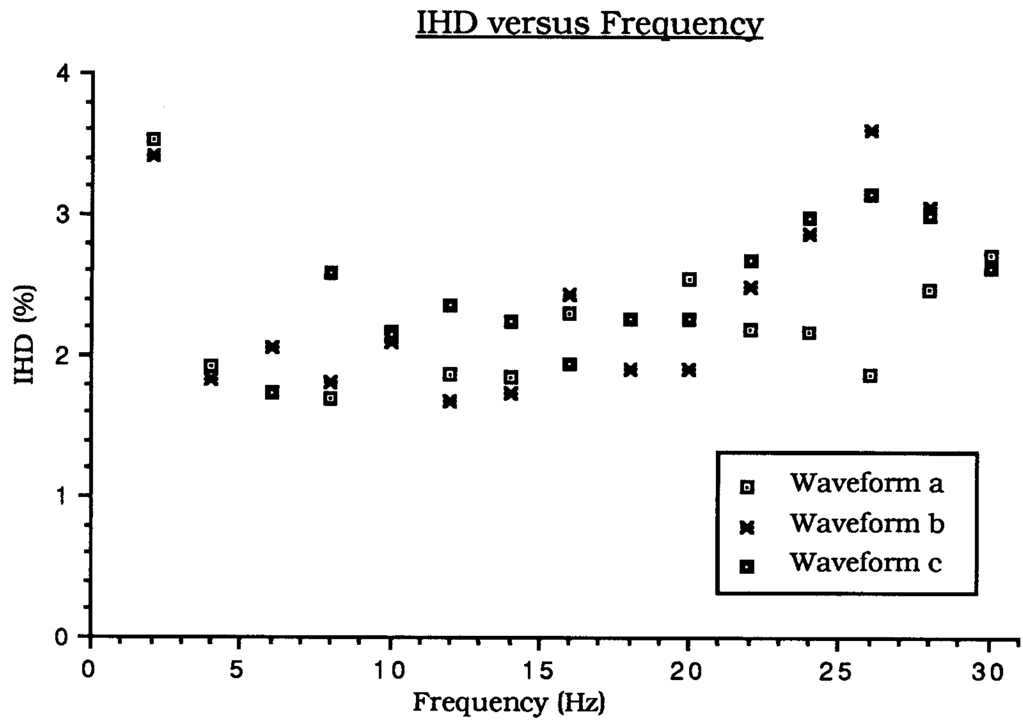


Fig 7.8 Current distortion on no-load for different waveforms

7.5 Voltage to Frequency Relationship

The voltage - frequency relationship is computed from the parameters of the motor determined from the no-load and blocked rotor tests. Figure 7.9 shows the v/f characteristic used for the motor.

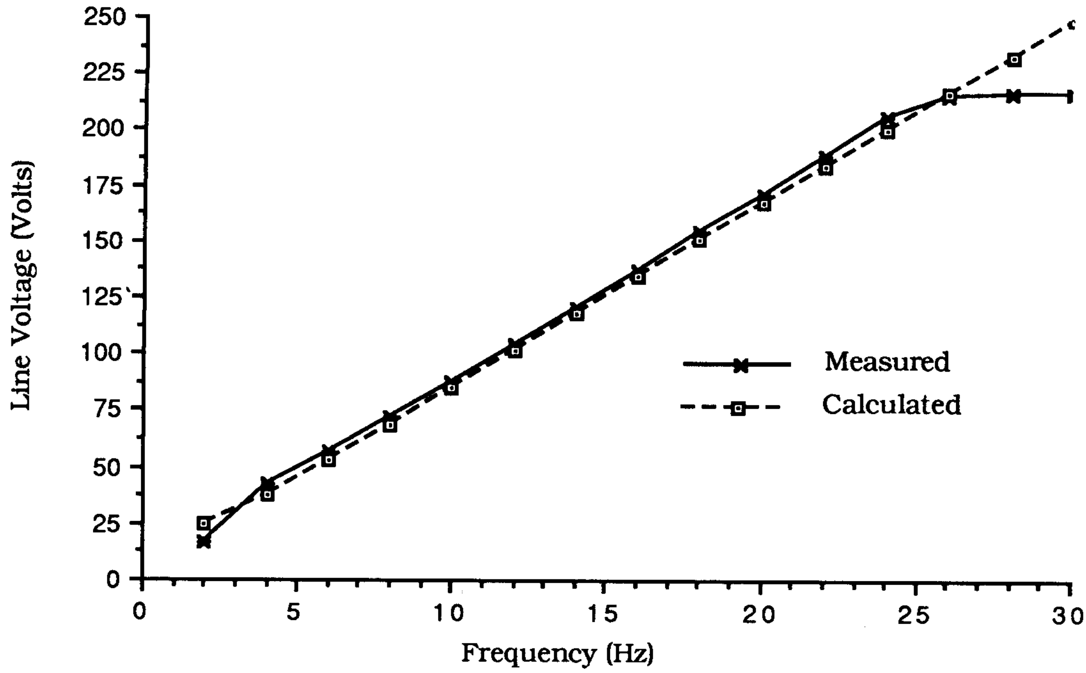


Fig 7.9 Voltage frequency relationship

Below 5 Hz the performance of the inverter deteriorates. This is due to the low modulation index and the pulse deletion and delay circuits. Above 30 Hz the operation of the motor is in constant voltage region due to the limited supply voltage. The graph shows good correspondence between expected and measured values. The losses due to the pulse deletion and delay circuits are therefore negligible as expected and there is no need for compensation circuits.

7.6 Motor Load

The current distortion for different loads is shown in figure 7.10.

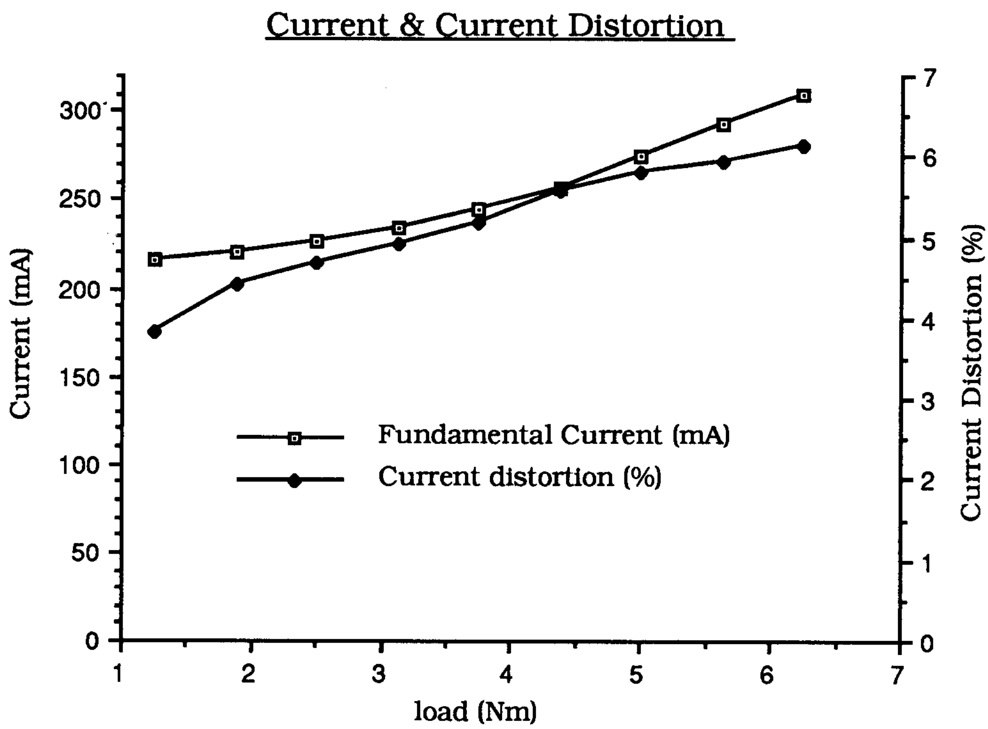


Fig 7.10 Current Distortion & Current Vs load

The current distortion (IHD) of 3-4% is acceptable for inverter drives. The calculated THD given in chapter 3, does not agree with this because the derivation of THD is based on the ideal voltage harmonics and the simplified model of the motor. The IHD obtained, however, compares with the experimental result obtained by Bowler *et al* [62], who used a carrier of 1220 Hz and fundamental of 48 Hz and obtained IHD of 10.92%.

The spectra of the motor current shows unexpected even harmonics (figure 7.11, 7.12). These may be due to the the modulation of the DC ripple voltage, by the inverter, as well as motor and load construction harmonics [63]. The spectra of line voltages on load (figure 7.13) and no load (figure 7.14) shows the increase in the voltage harmonics and shows the effect of the rail voltage on the inverter output as the load is increased - the fundamental component remains almost constant. As the load is increased, the relative magnitude of the harmonics increases, 3% to 4%, due to the effect of increased load current on the DC link voltage. Their magnitudes are, however, negligible. The harmonics due to the modulation of the ripple voltage are given by the expression [63] :

$$f_{nh} = p f_1 \pm f_o (3n \pm 1) \qquad (7.2)$$

where

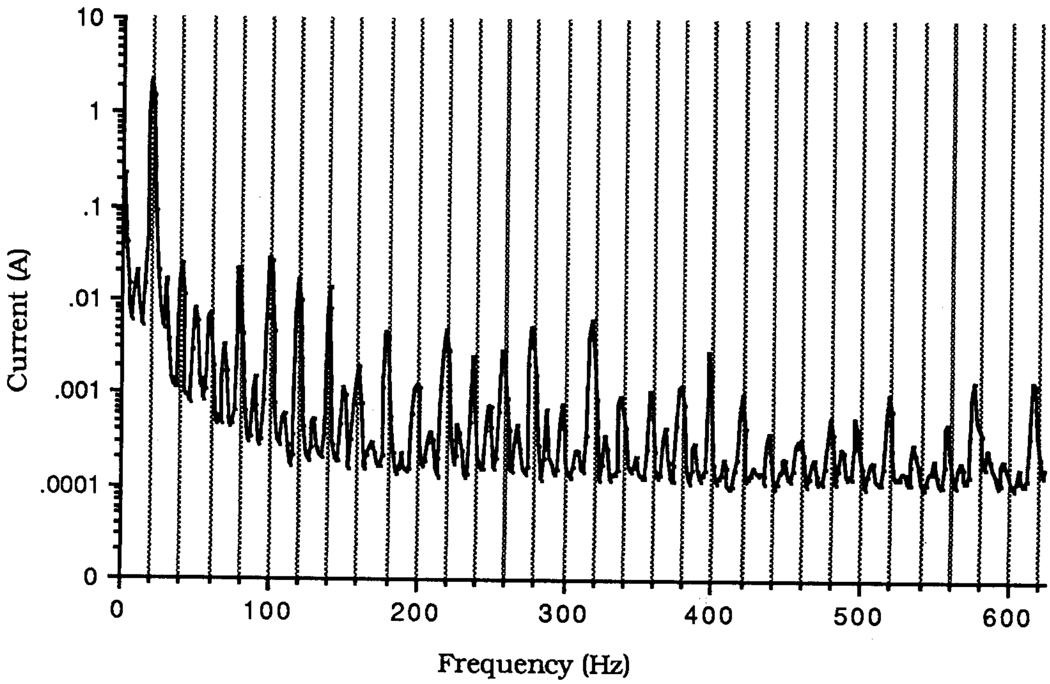
f_o is inverter supply frequency, f_{nh} is ripple modulation frequency

f_1 is supply frequency, p is inverter pulse number and

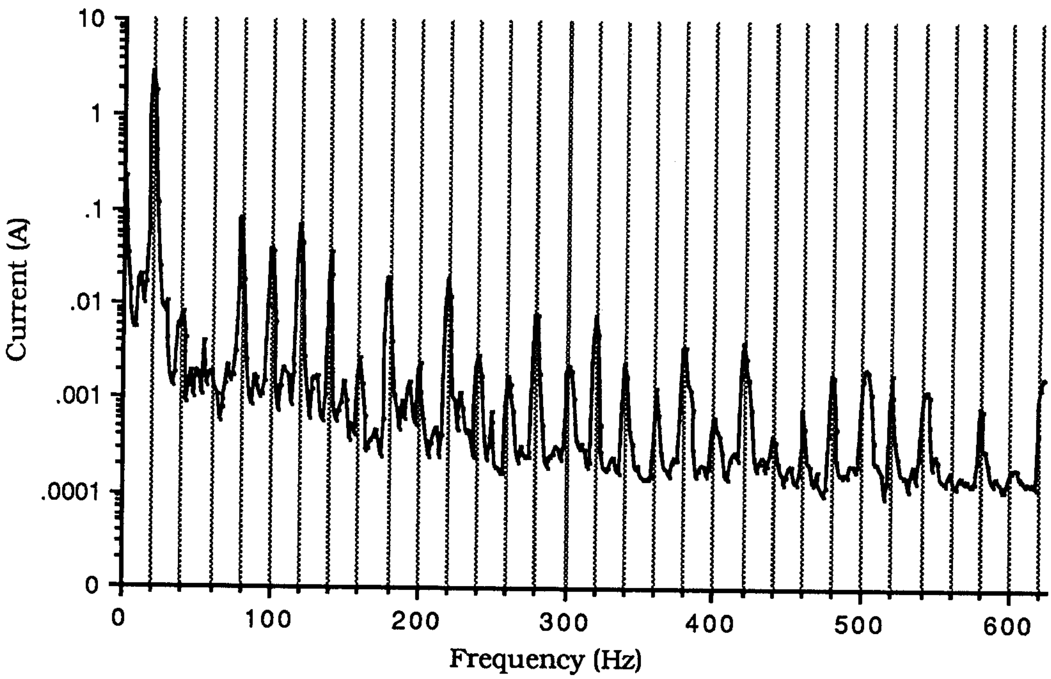
n is a positive integer or zero

7.7 DC - Link Current

The dc link current spectra, figures 7.15 to 7.18, show that the the major current harmonics occur near the carrier frequency. The DC current is not shown as the spectrum analyzer filters DC. The significant harmonics are around the carrier frequency and can be filtered from the supply with small filter components.



<u>Fig 7.11   No-load Current Spectrum</u>



<u>Fig 7.12 Full Load Current Spectrum</u>

*   *Each vertical line represents an integer harmonic.*

Fundamental frequency = 20 Hz

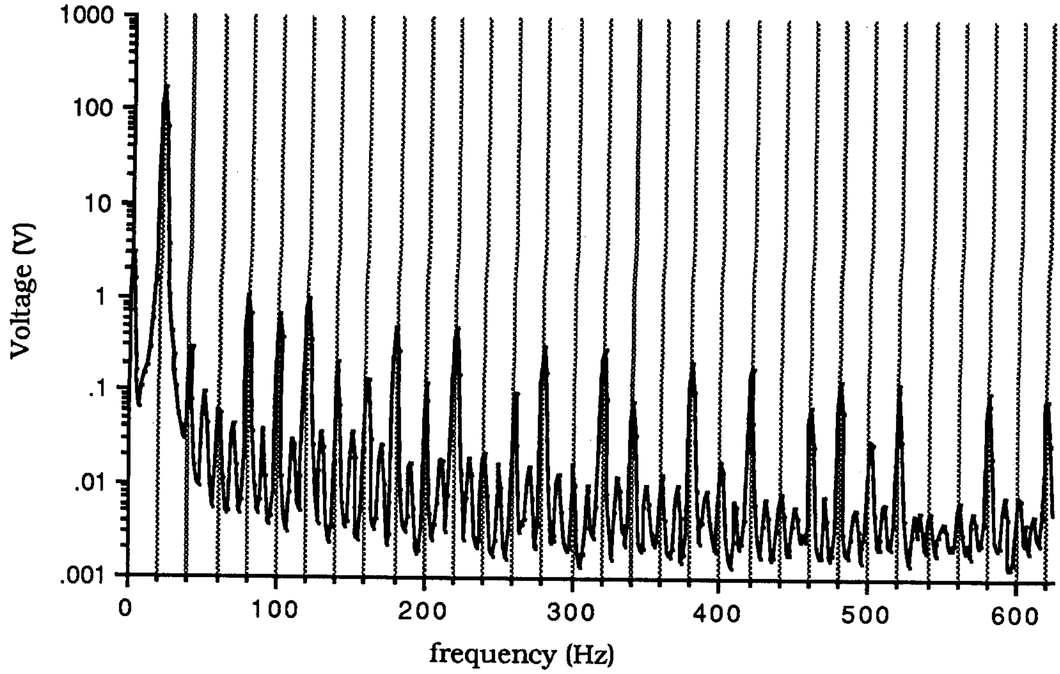


Fig 7.13 No-load Line Voltage Spectrum

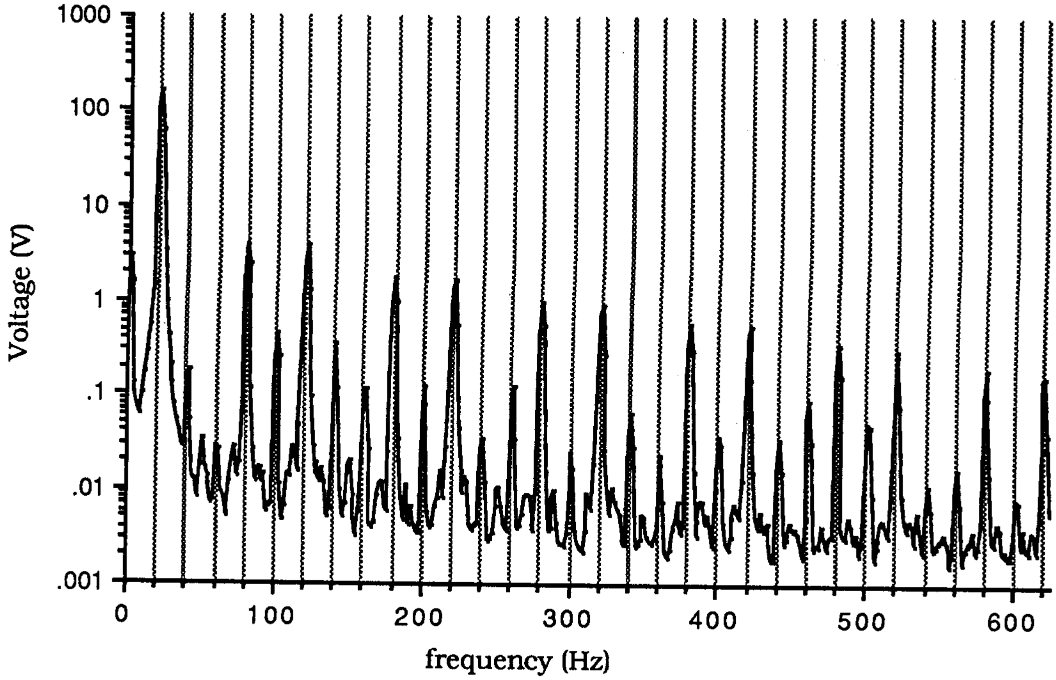


Fig 7.14 Full Load Line Voltage Spectrum

* *Each vertical line represents an integer harmonic.*

Fundamental frequency = 20 Hz

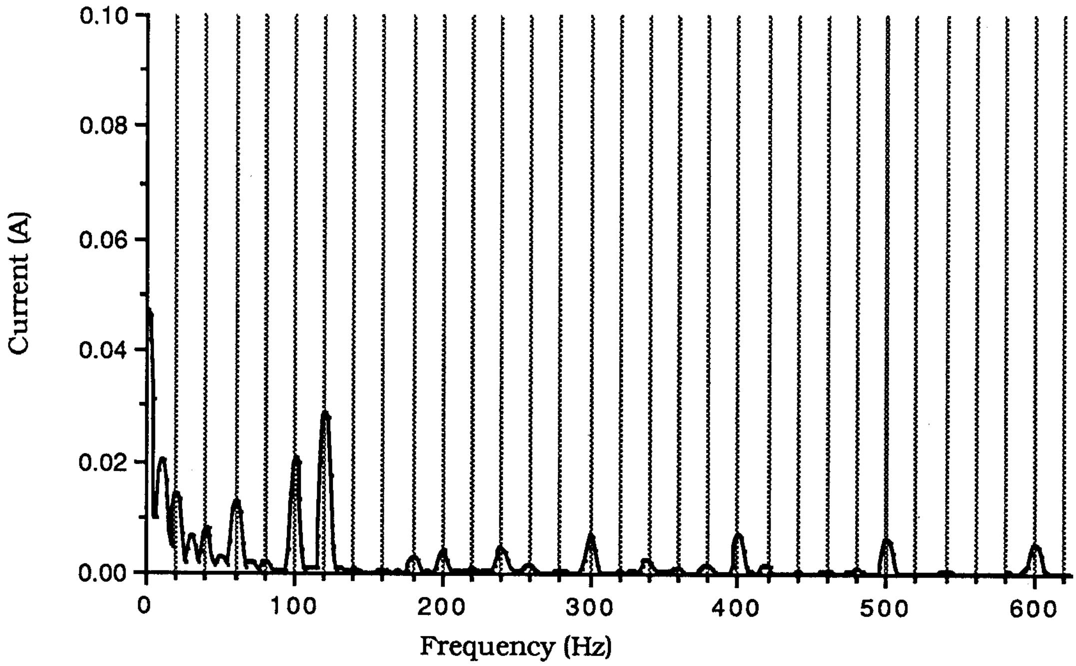


Fig 7.15 Low frequency Spectrum of DC Current on No-load

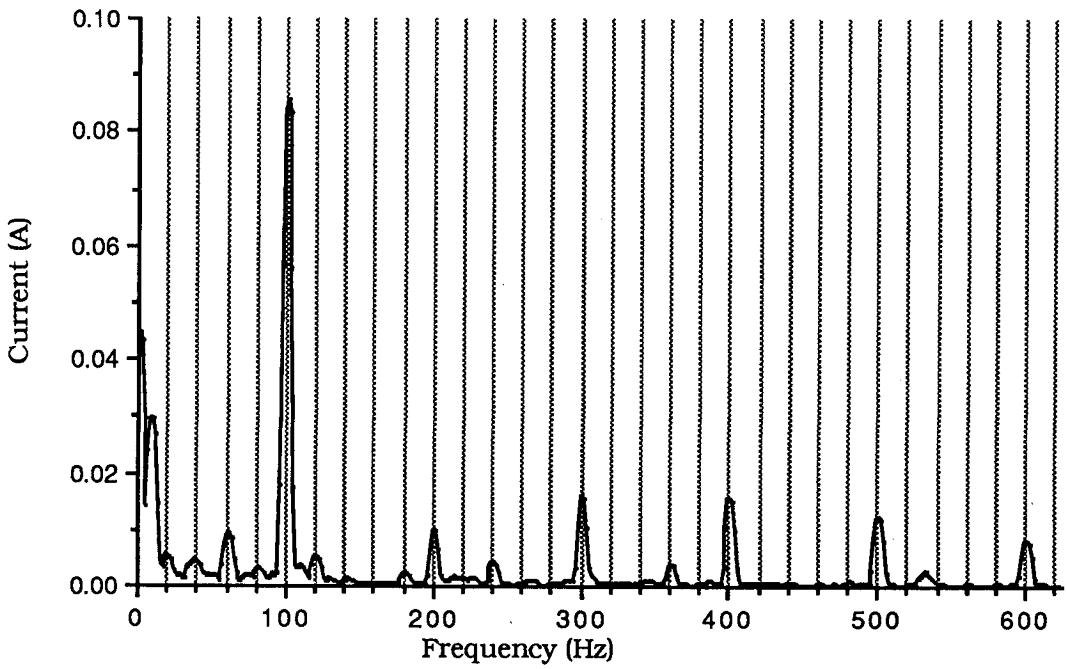


Fig 7.16 Low frequency spectrum of DC current on Full -load

* *Each vertical line represents an integer harmonic.*

Fundamental frequency = 20 Hz

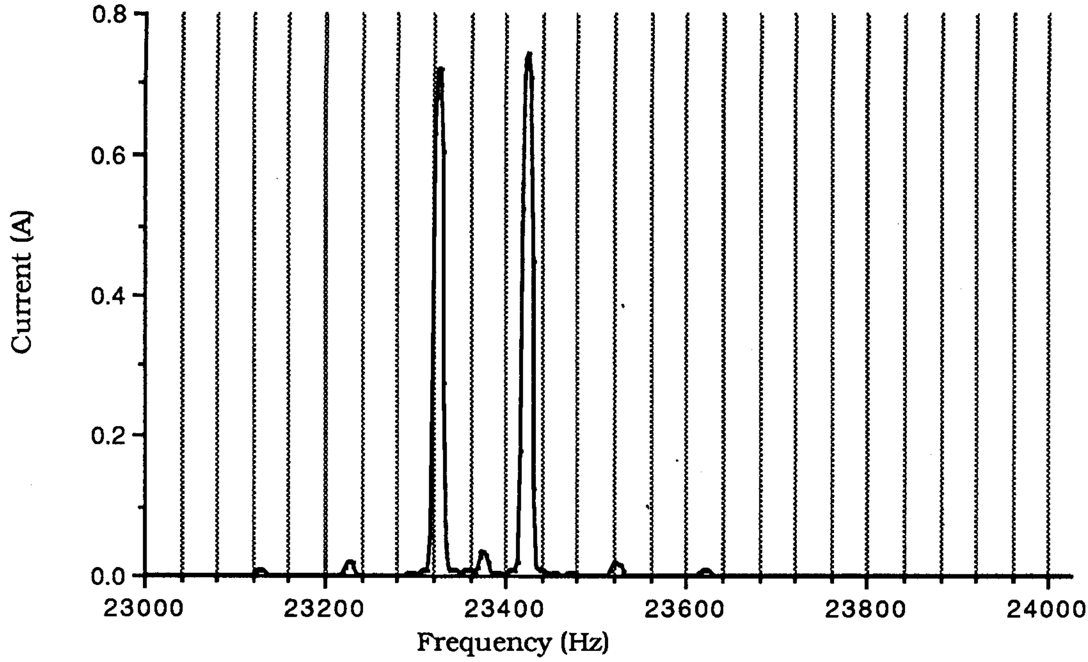


Fig 7.17 High frequency spectrum of DC current on No-load

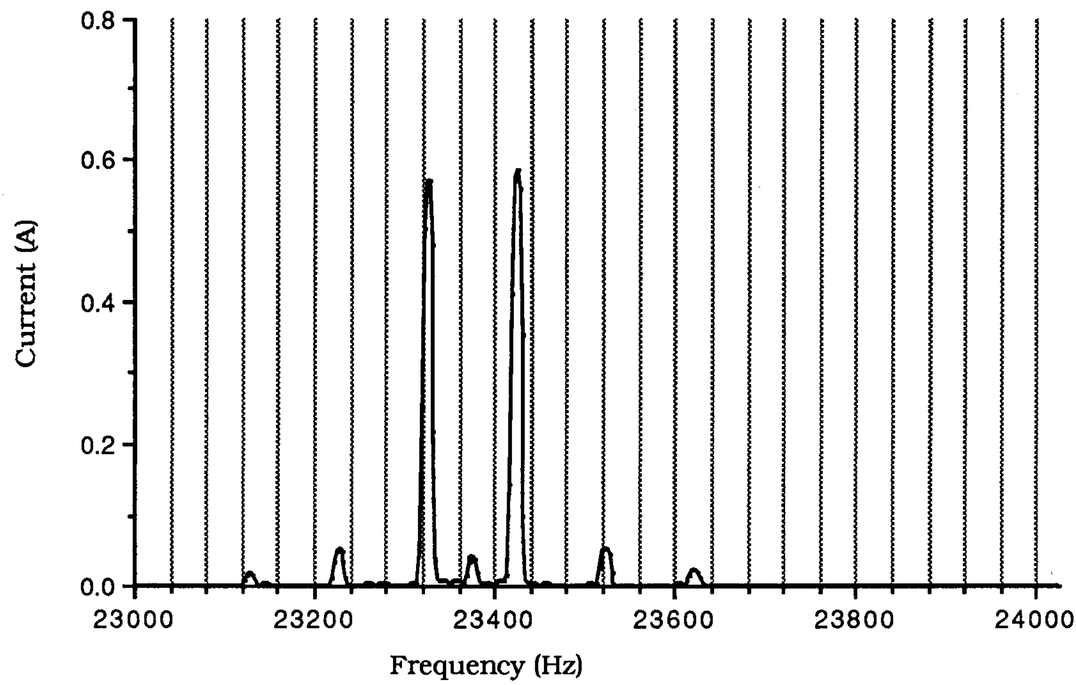


Fig 7.18 High frequency spectrum of DC current on Full-load

* *Each vertical line represents an integer harmonic.*

Fundamental frequency = 20 Hz.

7.8 Torque

For constant flux operation the electromagnetic torque is given by equation 2.12. Assuming, that the losses may be represented, for all frequencies, by an equivalent constant torque at rated frequency, the torque-speed characteristics expected for various frequencies are shown in figure 7.19. The breakdown torque for the machine is shown in figure 7.20. This is contrary to the theoretical prediction, based on the equivalent circuit model, that the breakdown torque is substantially constant. The main reason is the inadequate motor model for low frequencies such that the compensation for I^2R losses is not sufficient. The equivalent circuit is based on the assumption that the parameters do not change with frequency and current. The stator leakage reactance decreases at low frequencies. Windage and friction losses taken together are assumed to be constant and this is only a crude approximation which for a small machine is a significant part of the total load. As the desired constant torque capability of the motor is not attainable at low stator frequencies, the starting of the induction motor is affected.

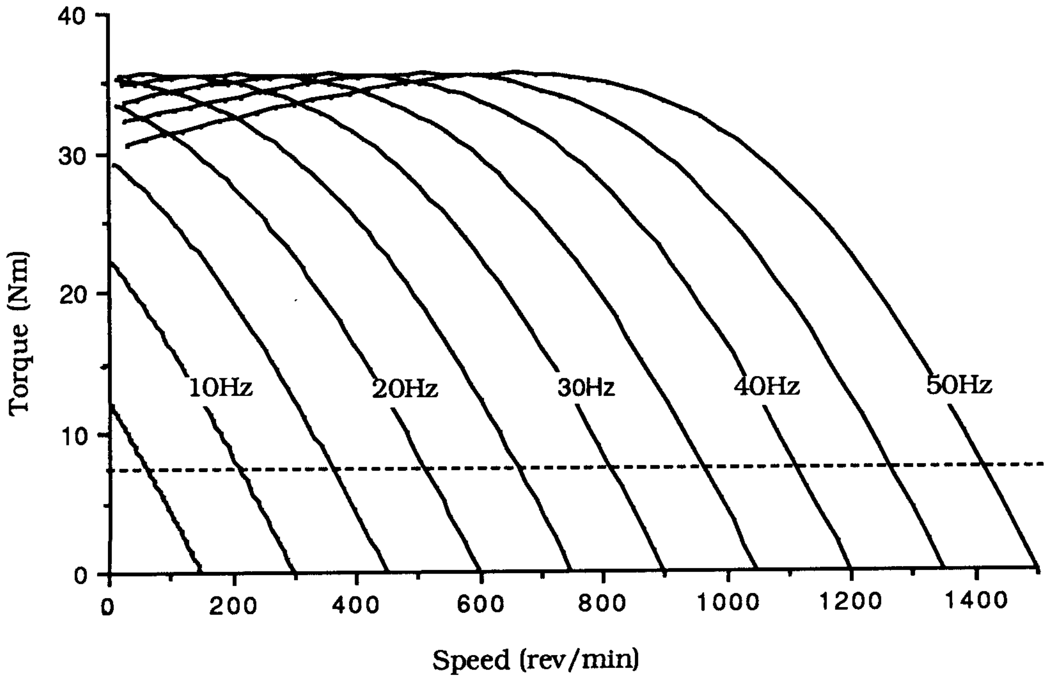


Fig 7.19 Torque speed Characteristics for variable frequency

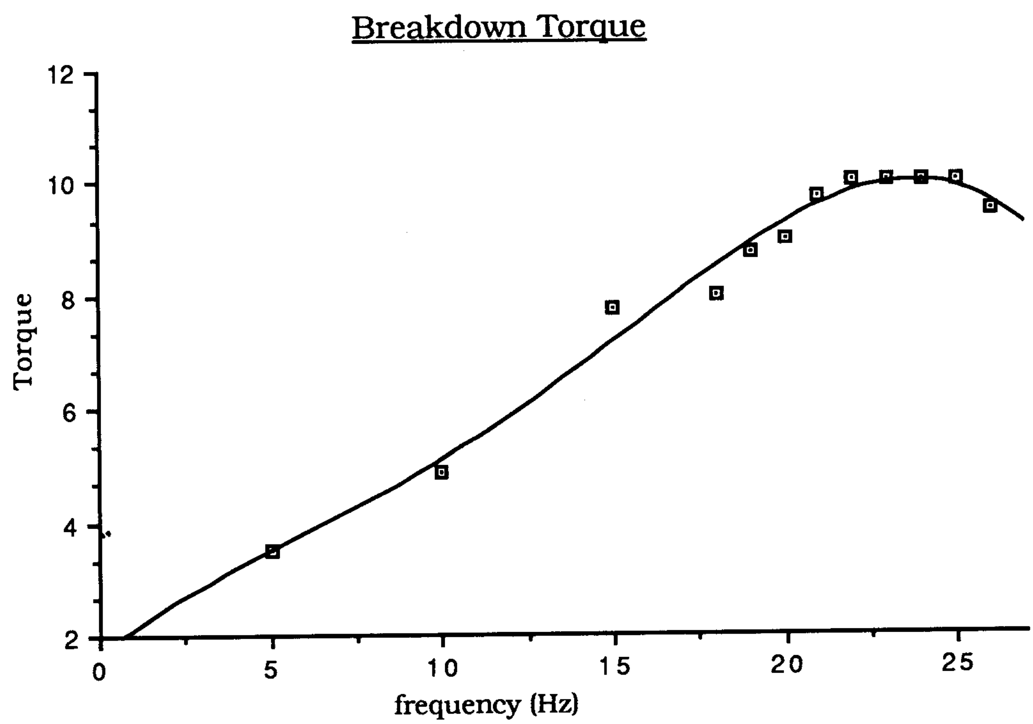


Fig 7.20 Breakdown torque for machine

A comparison was made between a direct supply of a variable frequency sinusoidal supply from an alternator and the PWM inverter on a delta connected squirrel-cage induction motor with the following ratings:

2.2 kW   1440 rev/min   50 Hz   400 V   4.9 A

3-phase   4 pole

One comparative result was obtained at a frequency of 20 Hz and is shown in table 7.1. The efficiency of the drive was computed from the measured values of input and output power. The measurement of input electrical power was considered an approximation due to the limited bandwidth of the wattmeters, especially in the case of the inverter supply. The power factor computed is the ratio of input power to input V_{rms} I_{rms}, and is also affected by the non-sinusoidal nature of the waveforms measured.

20 Hz Supply	Torque (Nm)	Input Power (W)	Output Power (W)	Speed (Rev/Min)	Efficie (%)
Direct	31.3	592	473	577	80
Inverter	31.5	700	475	576	68

Table 7.1 Comparison between inverter and direct supply

The difference of 100 W is lost in the rectifier, the MOSFET's and the clamp circuits. For a dc voltage of 270 V, load current of 3 A and other parameters as in chapter 5, the total losses in the MOSFET's from equation 5. is:

$$P_t = 36 \text{ W}$$

The total clamp circuit losses from equation 5.14 is given as

$$P_c = 3 L I_L^2 f_c = 6.4 \text{ W}$$

The remaining 50 W losses may be due to the losses in the rectifier and also due to instrument error.

Graphs of speed, efficiency and power factor against output power for a supply frequency of 20 Hz are shown in figures 7.21, 7.22 and 7.23 respectively. As the load is increased the dc link voltage drops and this has a significant effect on the speed.

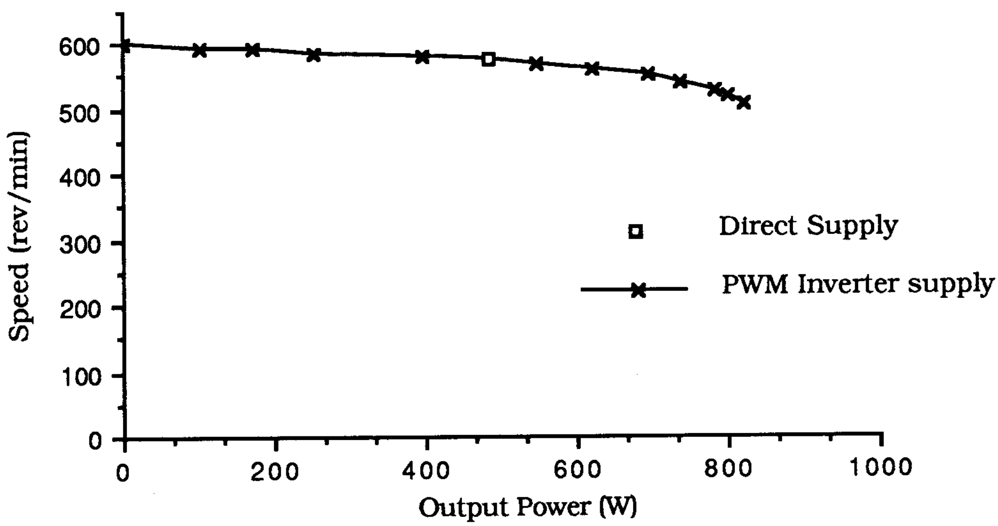


Fig 7.21 Graph of speed against output power

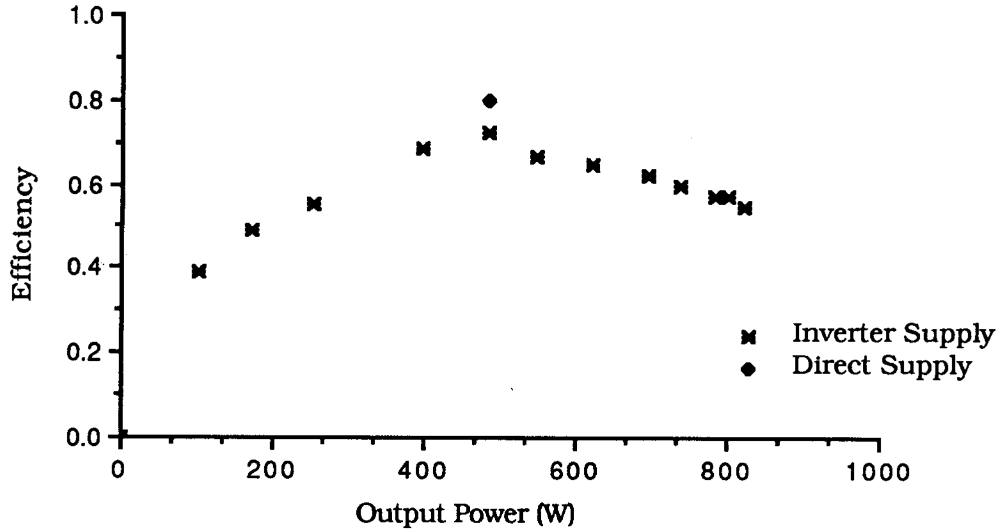


Fig 7.22 Graph of efficiency against output power

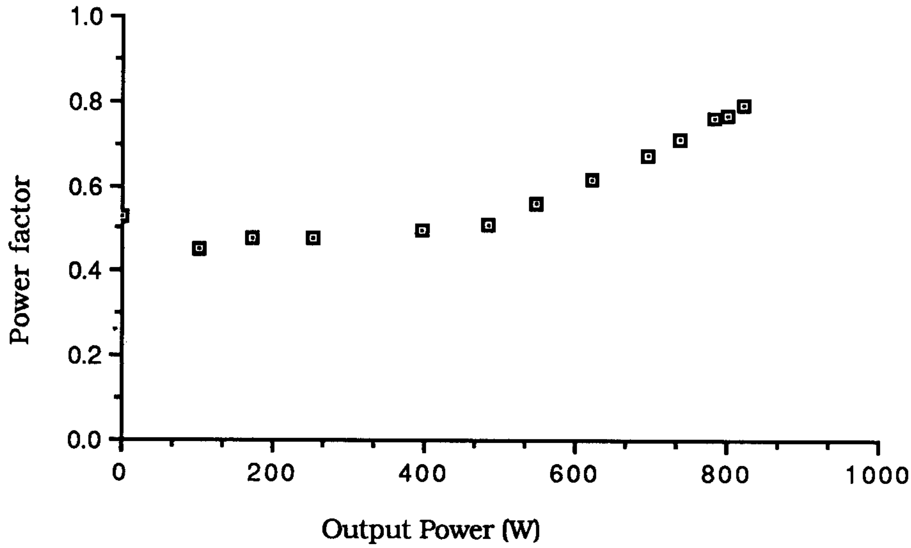


Fig 7.23 Graph of power factor against output power

7.9 Starting

The starting currents with variable frequency starting is much reduced. The starting current, with frequency step of about 0.8 Hz and step time of 0.5 seconds with motor on no load is shown in figure 7.24 and the deceleration in figure 7.25. Against rated load, the motor does not start immediately and the starting current is twice as much, figure 7.26. If the starting frequency is sufficieht to give rated torque, the starting current does not persist for long, as shown in figure 7.27. The small peaks indicate the portion where the motor has reached stable operating points and the step changes in frequency are smooth.

For a minimum starting frequency of 18 Hz, the effect of step time on the peaks is shown in figure 7.28. If the time between frequency steps is too long, the speed response is sluggish without any significant benefit in peak current.

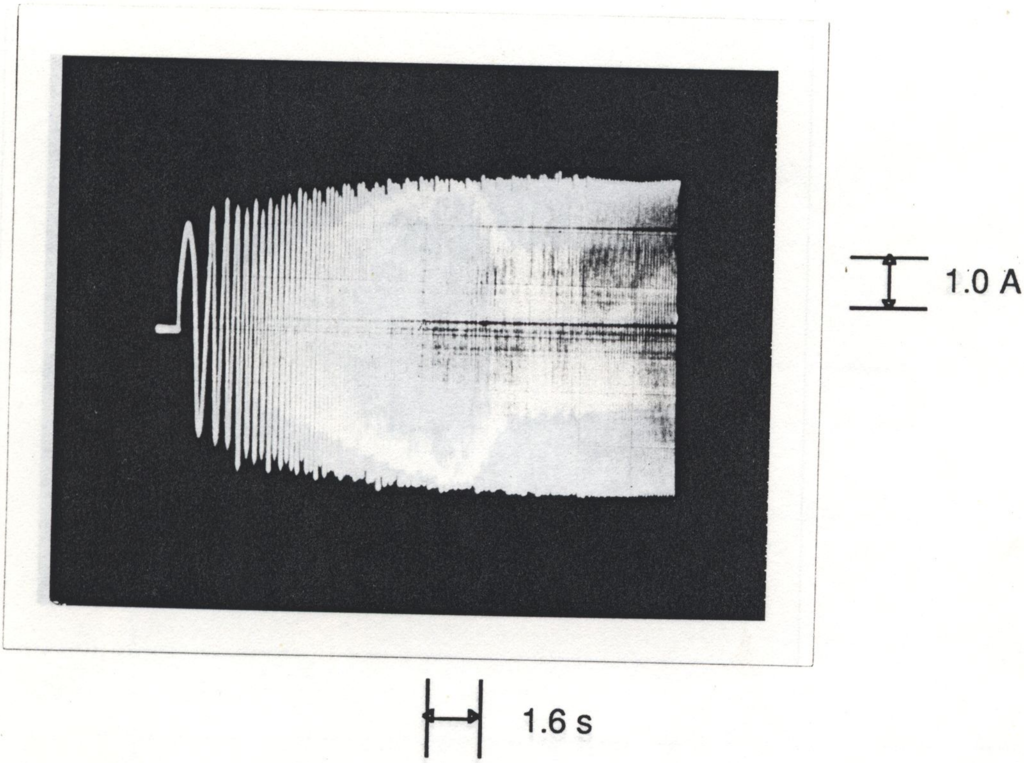


Fig 7.24 Starting Current on Motor on No-Load

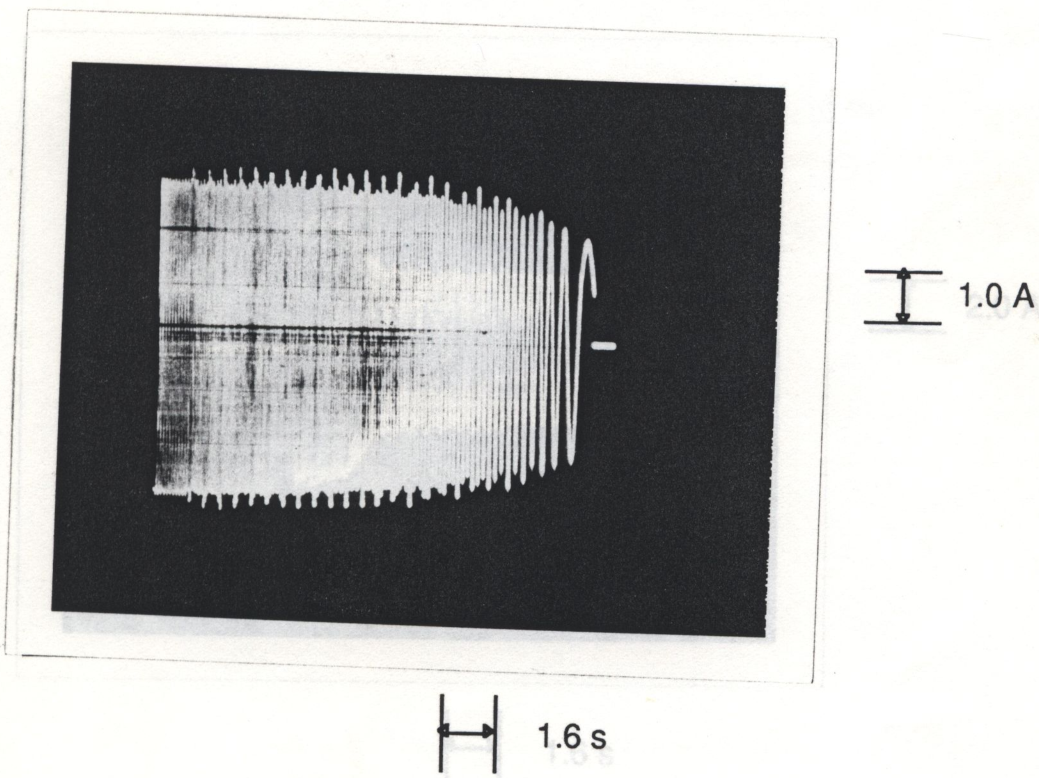


Fig 7.25 Current during deceleration of motor on No-load

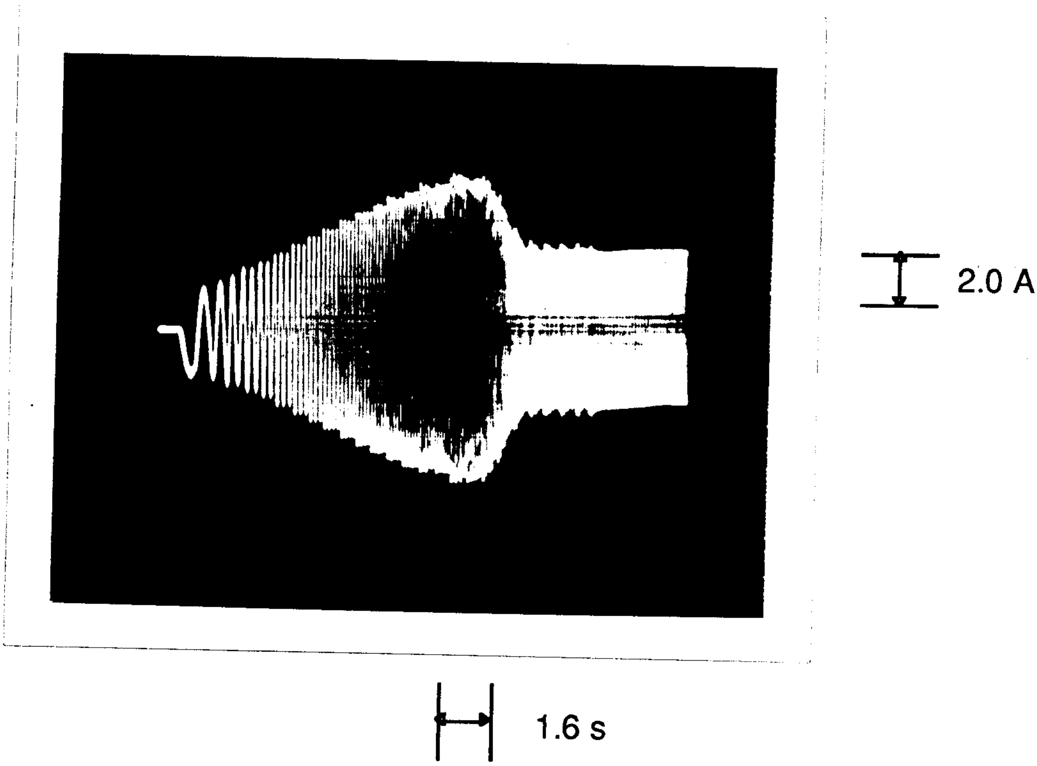


Fig 7.26 Starting current of motor on full load

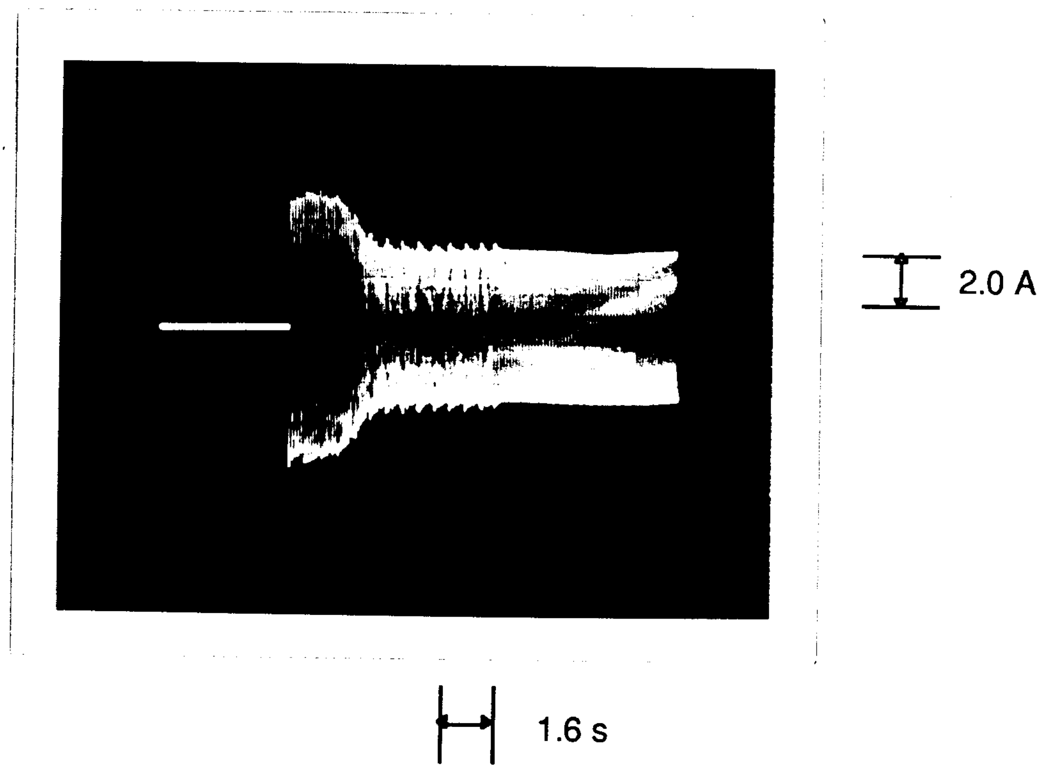


Fig 7.27 Starting with minimum starting frequency

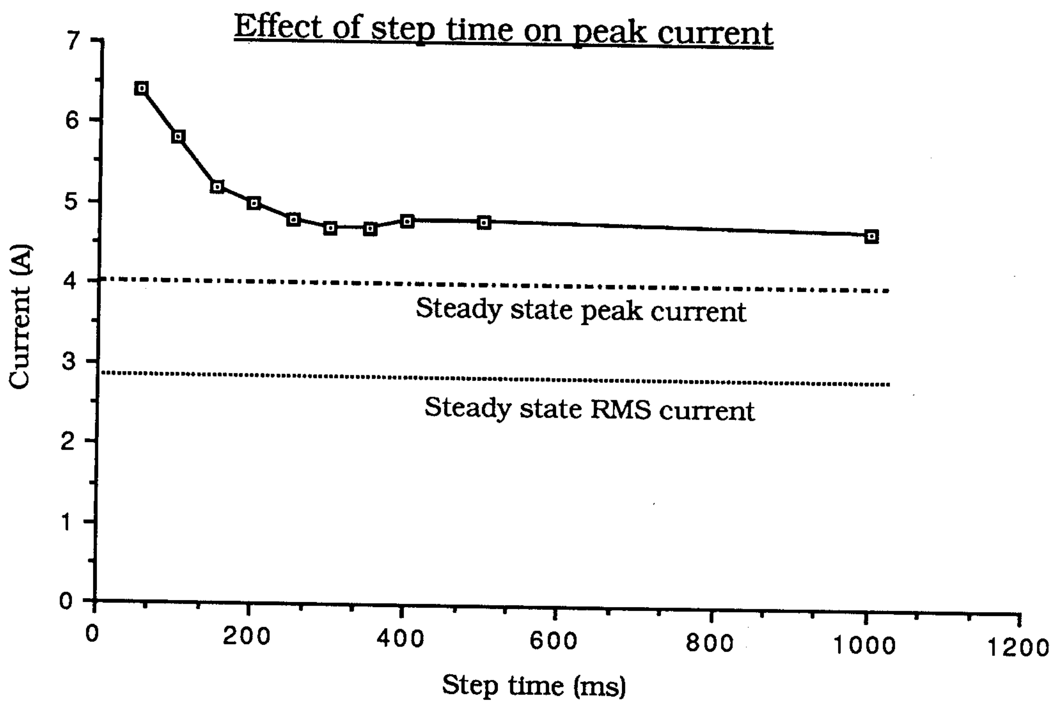


Fig 7.28 Effect of Step time on peaks

Chapter 8

Conclusion

A complete variable-speed drive has been designed and shown to work satisfactorily. It has been shown that a simple control strategy and design is suitable for variable - speed control of an induction motor. The drive was based on a ASIC, the MA818, and a 286 CPU computer. It has been shown that an inverter design based on MOSFET's and the IR2110 device drivers reduces the complexity of the inverter, eliminating transformers and opto-couplers.

The unit was tested on a 1.1 kW induction motor from no-load to full load at different speeds and useful results were obtained. The operation was smooth with no noticeable torque pulsations. Above 15 Hz the variation of speed produced very small current transients.

The use of a high carrier frequency has been shown to provide very low voltage harmonic distortion in the low frequency region. The inductance of the motor proved to be sufficient to attenuate the high order harmonics so that no output filters had to be used. The current distortion was in the range of 3-4% and is mainly caused by the modulation of the dc-ripple voltage and the switching of the devices.

It has been proved that the optimum waveform suggested in chapter 3 does provide maximum utilization of dc rail-voltage without distortion of the output line voltage waveform. An increase of about 15% of the fundamental is possible with no additional increase in harmonic distortion. It has also been proved that the addition of triplen harmonics has negligible effect on current distortion at high carrier frequencies.

Rated flux was, however, not attainable in the low frequency range. The rated torque of 7 Nm was not attainable below 15 Hz without extra voltage boosting. The cause for the low frequency deterioration was not obvious from the test results. It is possible that the equivalent circuit model is inadequate for low frequencies.

In the final prototype a micro-controller, preferably an 8-bit CPU, will be used to control the MA818. The routines will either be pre-programmed into the micro-controller's memory or written to an external memory. One card will be printed with the MA818, the micro-controller, the memory chip and the three IR2110 chips together with their associated passive components. A small power supply will be required to provide the 5 V and 12 V derived from either the dc-rail or the mains.

The MA818 was found to be very sensitive to noise and accidental shut down occurs at high rail voltage. Good construction of the inverter and control circuit, with noise reduction techniques used, would reduce the noise. The input ac/dc converter has to be designed and its effect on the inverter considered. It is suggested that due to the motor accidentally stalling the input supply circuit be designed to handle the energy fed back. The inverter circuit could be improved with the use of IGBT's. IGBT's are now becoming available in three phase modules with ratings of 1000 V, 25 A and with switching frequencies suitable for PWM. The similarity of the drive requirements to that of MOSFET's but with better current and voltage ratings makes them a better choice for manufacture. The reliability would be improved and the ratings would be higher.

References

1 Dodson, R.C and Evans, P.D: "Application of Dead-time compensation to variable speed drives", 4 th Int. Conf. on Electrical Machines and Drives, 1989, pp 369-373.

2 BS4999 part 112 , 1987.

3 Bowes, S.R. and Midoun, A: "Suboptimal switching strategies for microprocessor-controlled PWM inverter drives", IEE Proc. Vol. 132, Pt B, No.3, May 1985. pp 133 - 148.

4 Sethuramen. S.K. and Waheed. M.A.: "A single chip microcontroller based real-time PWM converter", IEE Conf. Pub. No. 291, 1988, pp 390-393.

5 Bowes, S.R : " New sinusoidal pulse width modulated inverter", IEE Proc. Vol. 122, Pt B, No 11, Nov 1975, pp 1279-1285.

6 Hamman, J. and Van der Merve, F : " Voltage Harmonics generated by voltage fed inverters using PWM natural sampling, " IEEE Trans. on Power Electronics, Vol. 3, No. 3, July 1988.

7 Grant D.A and Siedner R. : "Ratio changing in pulse-width-modulated inverters", IEE Proc. Vol. 128, Pt B, No.5, Sept 1981, pp 243-248.

8 Bowes.S.R and Mount M.J.: "Microprocessor control of PWM inverters", IEE Proc. Pt B, Vol. 128, No.6, Nov 1981, pp 293-305.

9 Addoweesh, K.E, Sheperd, W, and Hulley, L.N.: "Induction motor speed control using a microprocessor based PWM inverter", IEEE Trans. on Power Electronics, Vol. 36, No. 4, Nov 1989.

10 Acharya,G.N, Shakharat,S.S, Sheperd,W,M, Rao,U.M and Yaun, M.N: "Microprocessor based PWM inverter using modified regular sampling techniques", IEEE Trans. on Power Electronics, Vol. IA-22, No. 2, March/April 1986, pp 286-291.

11 Grant, D and Seidner, R : "Technique for pulse elimination in pulse width inverters with no waveform discontinuity", IEE Proc Vol. 129, Pt. B, No. 4, July 1982, pp 205-210.

12 Thorberg, K. and Nystrom, A.: " Staircase PWM : An uncomplicated and Efficient modulation Technique for AC Motor Drives", IEEE Trans on Power Electronics, Vol. 3, No. 4, Oct 1988. pp 391-397.

13 Nystrom, A., Hylander, K., Thorberg, K : "Harmonic currents and torque pulsations with pulse width modulation methods in ac motor drives", IEE. Conf. Pub. No.291, 1988, pp 378.-381.

14 Taniguchi K. and Irie.H : "Trapezoidal modulating signal for three-phase PWM inverter". IEEE Trans Ind. Elect. , Vol. 1E-33, No 2, May 1986, pp 193-200.

15 Trzynadlowski, A : "Non-sinusoidal Modulating functions for 3-phase inverters', IEEE Trans on Power Electronics, Vol. 4, No. 3, July 1989., pp 331-338.

15 Grant, D.A., and Houldsworth, J.A. : "PWM AC motor drive employing ultrasonic carrier", IEE Conf. Pub. No. 234, 1984, pp.237-240.

17 Holtz,J, Lammert, P and Lotzkat, W : " High speed drive system with ultrasonic MOSFET PWM inverter and single-chip microprocessor control," IEEE Trans. Ind. Appl., Vol. IA-23, No. 6, Nov/dec 1987, pp 1010-1015.

18 Grant,D.A, Houldsworth, J.A and Lower, K.N : " A new high quality PWM AC drive", IEEE Trans. Ind. Appl., Vol. IA-19, No. 2, March/April 1983, pp 211-216.

19 Nishimura, T., Nakaoka M. and Maruhashi, T.: "Reduction of vibration and acoustic noise in induction motor driven by three-phase PWM AC chopper using static induction thyristors", IEEE Trans. Power Elect., Vol. 4, No. 3, 1989, pp 313-318.

20 Patel, H.S and Hoft, R.G : " Generalized techniques of Harmonic Elimination and voltage control in thyristor inverters, Part 1 - harmonic Elimination", IEEE Trans. Ind. Appl., 1A-9, 3, May/June 1973, pp 310-317.

21 Patel, H.S and Hoft, R.G : " Generalized techniques of Harmonic Elimination and voltage control in thyristor inverters, Part 2 - Voltage control techniques", IEEE Trans. Ind. Appl., 1A-13, 5, Sept/Oct 1974, pp 666-673.

22 Thomas, G. and Lim, K.M.: "Recent developments in microprocessor control of variable speed inverter fed AC drives", IEE Conf. Pub. No. 234, 1984, pp 241-244.

23 Loh. W.K and Renfrew A.C.: " Optimized Microelectronic control of an AC drive for an electric vehicle", IEE Conf. Pub. No 234, 1984, pp 245-248.

24 Taufig.J.A. Prof Mellitt. B. and Goodman.C.J.: "Novel algorithm for generating near optimal PWM waveforms for AC traction drives:, IEE Proc. Pt B. ,Vol. 133, No. 2, March 1986, pp 85-94.

25 Buja. G.S and Indri. G.B.: "Optimal pulse width modulation for feeding ac motors", IEEE Trans., 1977, vol. 1A - 13, No. 1, pp.38-44.

26 Bellini. A. and Figalli. G.: "On the selection of commutation instants in induction motor drives", IEEE Trans. Ind. Appl. Vol. IA-16, 5 Sept/ Oct 1979.

27 Buja. G.S: "Optimum output waveforms in PWM inverters", IEEE Trans Ind Appl, Vol. IA-16, No.6, Nov/dec 1980, pp 830-836.

28 Buck. De. D. Gistelinck P . and De Backer. D.: "Loss-optimal PWM waveforms for variable-speed induction-motor drives", IEE Proc. Vol. 130, Pt B, No 5, Sept 1983, pp 310-320.

29 Murphy. J.M.D. and Egan. M.G : " A comparison of PWM strategies for inverter fed induction motors", IEEE Trans. Ind. Appl. Vol. IA-19, May/June 1983, pp 363-369.

30 Boys.J.T and Handley. P.G: " Harmonic analysis of space vector modulated PWM waveforms", IEE Proc. Vol. 137, Pt B, No 4, July 1990, pp 197-204.

31 Murai. Y. , Ohashi. K. and Hosono. I.: " New PWM method for fully digitized inverters", IEEE Trans. Ind. Appl., Vol. 1A-23, No. 5, Sept/Oct 1987, pp 887-893.

32 Holtz. J. , Lammert. P. and Lotzkat. W.: "High-Speed drive system with Ultrasonic MOSFET PWM Inverter and single-chip microprocessor control", IEEE Trans. Ind. Appl., Vol. 1A-23, No. 6., Nov/Dec 1987. pp 1010- 1015

33. Hoft. R.G., McLaren. R.W., Pimmel. R.L. and Gokhale. K.P.:" The impact of Microelectronics and Microprocessors on Power Electronics", IEE Conf. Pub. No. 234, May 1-4, 1984, pp 191-198.

34 Bowes.S.R and Mount M.J.: "Microprocessor control of PWM inverters", IEE proc, Pt B,Vol. 128, No.6, Nov 1981,pp 293-305.]

35 Iwanciew.P.: "Applications of microprocessor controlled inverter drives ", IEE Conf. Pub. No.291, 1988, pp 202-205.

36 Lettner, E.D: "Interfacing a portable PC to a microprocessor controlled drive for development work, commissioning and trouble-shooting", IEE Conf. Pub. No. 324, 17-19 July 1990, pp 523-528.

37 Sethuramen S.K. and Sagar. P.: "Knowledge based system model for ac drive control", IEE Conf. Pub. No. 324, 17-19 July 1990, pp 394-399.

38 Mirkqazemi-Moud. M. ,Green. T.C., Williams. B.W.: "Use of ASIC Technology in the design of two novel PWM generators", IEE Conf. Pub. No. 324, 17-19 July 1990, pp 347-352.

39 Tez,E.S : " Motonic chip set : High performance intelligent controller for industrial variable speed A.C drives, IEE Conf. Pub. No.291, 1988, pp 382-385.

40 MA818 - Three phase PWM waveform generator (application notes), Marconi Electronic Devices, C10402FDS Issue 1 August 1989.

41 Klingshirn. E.A., and Jordan. H.E.: "Polyphase induction motor performance on non-sinusoidal voltage sources", IEEE Trans. Power Appar. Syst., PAS-87, 3, March. 1968, pp 624-631.

42 Chalmers. B.J., and Sarkar. B.R: "Induction motor losses due to non-sinusoidal supply waveforms", IEE Proc. Vol.115, No. 12, Dec. 1968, pp 1777-1782.

43 Williamson. A.C.: "The effects of system harmonics upon machines, Int. J. Electr. Eng. Educ. ,19, 2, Apr. 1982, pp. 145-155.

44 Cummings. P.G.: "Estimating effect of system harmonics on losses and temperature rise of squirrel-cage motors", IEEE Trans. Ind. Appl., Vol. 1A-22, No 6, Nov/Dec 1986, pp 1121-1126.

45 Bowes.S.R. and Midoun, A: "Suboptimal switching strategies for microprocessor-controlled PWM inverter drives", IEE Proc. Vol. 132, Pt B, No.3, May 1985, pp 133 - 148.

46 Boys, J.T and Walton, B.E: "A loss minimized sinusoidal PWM inverter", IEE proceedings. Vol. 132, Pt B, No.5, September 1985 .

47 Nystrom, A., Hylander, K., Thorberg, K. : "Harmonic currents and torque pulsations with pulse width modulation methods in ac motor drives", IEE. Conf. Pub. No.291, 1988, pp 378.-381.

48 Handley, P.G and Boys,J.T: " Space vector modulation: An Engineering review", IEE Conf. Pub. No. 324 , July 1990, pp 87-91.

49 Trzynadlwski, A.M: "Non-Sinusoidal Modulating Functions for Three phase Inverters", IEEE transactions on Power Electronics, Vol. 4, No.3, July 1989, 331 - 338 .

50 Taniguchi, K, Yasamasa, O and Hisaichi, I: "PWM Technique for Power MOSFET Inverter", IEEE transactions on Power Electronics, Vol. 3, No.3, July 1988, 329 - 334.

51 Buja,G.S. and Indri, G.B.: "Optimal pulsewidth modulation for feeding AC motors", IEEE Trans. Ind. Appl., IA-13, 1, Jan/Feb 1977, pp 38-44

52 Dodson, R.C., Evans, P.d. , Yazdi, H.T and Harley, S.C : " Compensating for dead-time degradation of PWM waveforms ", IEE Proc. Vol. 137, Pt B, No. 2, March 1990. pp 73-81.

53 Dodson, R.C and Evans , P.D : "Application of Dead-time compensation to variable speed drives", IEE Conf. Pub. 1989, pp 369-373.

54 Murai, Y.M, Watanabe, T., and Iwasaki, H : " Waveform distortion and correction circuit for PWM inverters with switching lag times." IEEE Trans. Ind. Appl. Vol. IA-23, No. 5 Sept/Oct 1987, pp 881-886.

55 Paice, D.A and Mattern, K.E. : "Gate-Turn-Off thyristors and their application", IEE Conf. Pub. No. 234,1984, pp 7-10.

56 Bassett, R.J , Jones, S.R, and Taylor, P.D.: " The practical limitations of high power GTO's and very high power bipolar transistors", IEE Conf. Pub. No.291, 1988, pp 95-98.

57 Van Wyk, J.D., Swanepoel, P.M and Schoeman, J.J : "Megahertz base drives for high power converters with high voltage bipolar transistors", IEE Conf. Pub. No.234, 1984, pp187-190.

58 Nishizawa, J. and Tamamushi, T.: "Recent developments and future potential of the power static induction (SI) devices",IEE Conf. Pub. No.291, 1988, pp 21-24.

59 Van Wyk, J.D :. " Electronic control of power flow : Present possibilities and some expected trends in applications ", IEE Conf. Pub. No.291, 1988, pp 1-12.

60 Nair.B.R. and Sen.P.C : " Voltage clamp circuits for a power MOSFET PWM inverter", IEEE Trans. Ind. Appl. Vol. 1A-23, No.5, Sept/Oct 1987, pp 911-920.

61 Power semiconductor Drives, Dewan.S,B,, Slemon. G.R. and Straughan. A., chapter 6.5.1. Distribution of leakage reactance in squirrel cage induction motors, pp 180-181, John Willey & Sons, 1984.

62 Bowler.P., Crater.R.C. and Jones. D.M: "The development and performance of a digital subharmonic modulator for ac motors", IEE Conf. Pub. 1988, No. 291, pp 403-407.

63 Grieve. D.W and McShane. I.E.: "Design engineering of inverter-fed motors and their shaft systems", GEC Technical Review, May 1990 No.2 , pp43-58.

Appendix A1: Derivation of Equation for Energy Dissipated

The equation of motion may be written as

$$J\frac{d\omega}{dt} + T_l = T_e \qquad \text{(A1.1)}$$

where J is the total inertia of load and rotor, and T_l and T_e are the load and electromagnetic torque respectively.

In order to convert to per unit form, the following conversion is used where the term J_n is defined, in seconds, with ω_s as the synchronous angular velocity in radians per second, as

$$J_n = \frac{\omega_s\, J}{T_n} \qquad \text{(A1.2)}$$

Substituting A1.2 in A1.1 and dividing both sides by base torque

$$J_n \frac{d(\omega/\omega_s)}{dt} + \frac{T_l}{T_n} = \frac{T_e}{T_n} \qquad \text{(A1.3)}$$

converting to per unit form

$$J_n\frac{d\omega}{dt} + T_l = T_e \qquad \text{(A1.4)}$$

$$\Rightarrow \quad dt = J_n \frac{d\omega}{T_e - T_l} \qquad \text{(A1.5)}$$

The energy lost in the rotor circuit of the motor in p.u

$$W = \int_{t'}^{t''} I_2^2 R_2 \, dt \qquad \text{(A1.6)}$$

where t' and t" are the start and end times, respectively.

T_e is given by the the expression

$$T_e = \frac{I_2^2 R_2}{s\omega_s}$$

$$= \frac{I_2^2 R_2}{(\omega_s - \omega)} \qquad \text{(A1.7)}$$

where ω is the per unit instantaneous angular velocity.

Using A1.6 and A1.7 gives

$$W = \int_{\omega'}^{\omega''} (\omega_s - \omega)\, T_e \, dt \qquad \text{(A1.8)}$$

and substituting A1.2 in A1.8

$$W = \int_{\omega'}^{\omega''} \frac{J_n\,(\omega_s - \omega)\; T_e \; d\omega}{(T_e - T_l)} \qquad \text{(A1.9)}$$

The time taken is given by

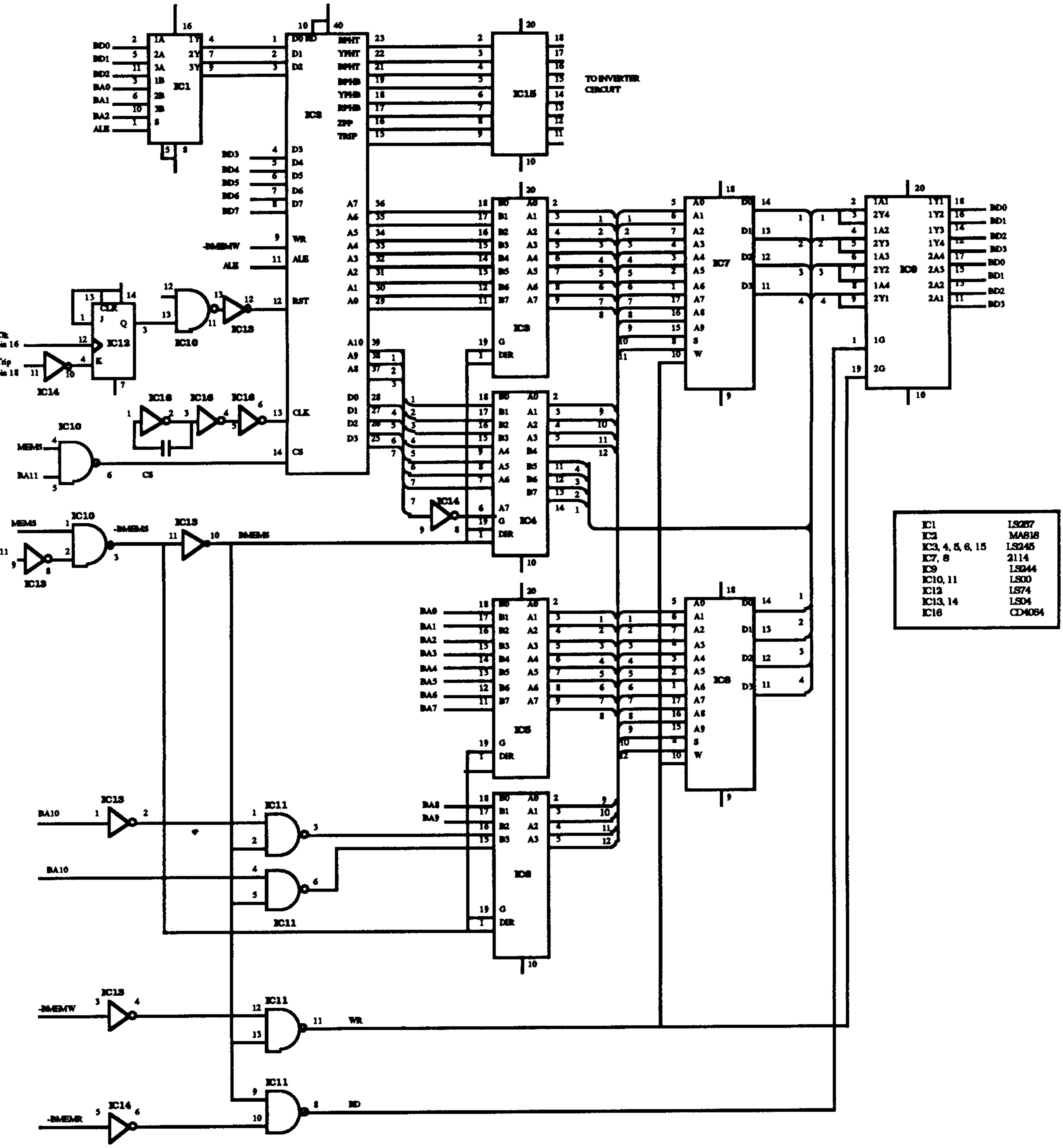
$$t = \int_{\omega'}^{\omega''} dt$$

$$\Rightarrow \qquad t = \int_{\omega'}^{\omega''} \frac{J_n \; d\omega}{(T_e - T_l)} \qquad \text{(A1.10)}$$

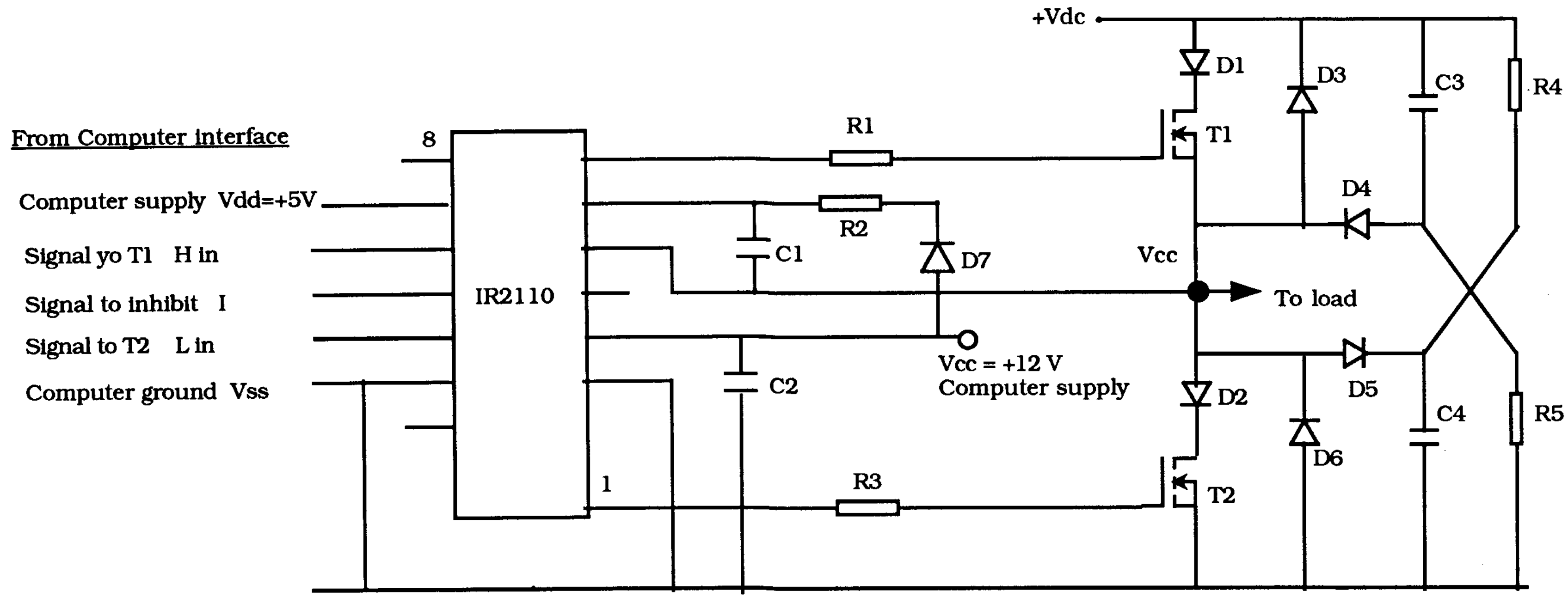
The energy is obtained in natural units by multiplying the per unit energy with the apparent power of the motor. An indication of the currents during startup may be obtained from the approximation of differential energy.

$$\delta W = I_2^2 R_2 \,\delta t \qquad \text{(A1.11)}$$

ppendix A2 : CIRCUIT DIAGRAM OF INTERFACE CARD



IC1	LS257
IC2	MA818
IC3, 4, 5, 6, 15	LS245
IC7, 8	2114
IC9	LS244
IC10, 11	LS00
IC12	LS74
IC13, 14	LS04
IC16	CD4084



R1, R3	4.7 Ω	C1	370 μF 100V	D1-D6	BTY12P
R2	8 Ω	C2	47 nF 250V	D7	IN4004,800V
R4, R5	1 Ω	C3, C4	1 μF 400V		

Driver Circuit for one phase of inverter

APPENDIX A3: Program Listing

```
#define         CLOCK           12000000.
#define         INHIBIT         0
#define         UNINHIBIT       1
#define         OVERMODULATION  1
#define         MODULATION      0
#define         FORWARD         0
#define         REVERSE         1
#define         RESET           0
#define         SET             1
#define         ONE             0x01

/*              initialisation registers                        */
/*                                                              */
/*      register R0             */
/*      ===========             */

typedef union Reg_init0 {
     struct {
          unsigned int   pdt:       7;
          unsigned int   counter:   1;
          unsigned int   zeros:     8;
     }x:
     struct {
          unsigned char  high;
          unsigned char  low;
     }y;
Reg_init0;

*       register R1             */
*       ===========             */

ypedef union Reg_init1{
    struct{
          unsigned int   car_freq:      3;
          unsigned int   dontcare:      2;
          unsigned int   freq_range:    3;
          unsigned int   zeros:         8;
    }x;
    struct{
          unsigned char  high;
          unsigned char  low;
    }y:
eg_init1;

        register R2             */
        ===========             */

pedef union Reg_init2 {
    struct {
          unsigned int   pdy:       6;
          unsigned int   zeros:     10;
    }x:
    struct {
          unsigned char  high;
          unsigned char  low;
```

```
/*              control registers                              */
/*                                                             */

/*         register R0 & R1      */
/*         ================      */


typedef union Reg_cont01{
     struct {
          unsigned int   power_freq:     12;
          unsigned int   dontcare:        1;
          unsigned int   outinhibit:      1;
          unsigned int   overmod:         1;
          unsigned int   for_rev:         1;
     }x;
     struct {
          unsigned char  high;
          unsigned char  low;
     }y;
}Reg_cont01;


/*         register R2           */
/*         ===========           */

typedef union Reg_cont2 {
     struct {
          unsigned int   amp:       8;
          unsigned int   zeros:     8;
     }x;
     struct {
          unsigned char  high;
          unsigned char  low;
     }y;
}Reg_cont2;


/* registers defined in main.c */

extern     Reg_init0       init_reg0;
extern     Reg_init1       init_reg1;
extern     Reg_init2       init_reg2;
extern     Reg_cont01      cont_reg01;
extern     Reg_cont2       cont_reg2;
```

```
/* Routine definitions */

float      setcarrier(int carrier_bits);
float      setrange(int range_bit);
float      setdelay(int pdy_bits);
float      setdeletion(int pdt_bits);
void       setcounter(int counter_bit);
void       setamplitude();
void       setdirection(int direction_bit);
void       setoutput(int inh_bit);
void       inhibit();
void       uninhibit();
void       initialise();
void       control();
void       status();
void       setfreq(int freq,unsigned int acc,
                unsigned int dec, unsigned int rate);
void       test(int freq);
void       forward();
void       reverse();
void       counter_set();
void       counter_reset();
unsigned int getfreq(int freq);
void       init(int pdy,int pdt);
int        save_status();
int        load_status();
void       write_mem();
void       read_mem();
int        test_mem();
void       volt_freq();
```

```
#include  <stdio.h>
#include  <fcntl.h>
#include  <stdlib.h>
#include  <dos.h>
#include  <conio.h>
#include  <time.h>
#include  "head.h"
#include  <graphics.h>
#include  <math.h>

Reg_init0       init_reg0;
Reg_init1       init_reg1;
Reg_init2       init_reg2;
Reg_cont01      cont_reg01;
Reg_cont2       cont_reg2;

unsigned int vf[4097],a,mini_freq;
float      Rail_Volt,util,min_freq;

float      xscale, yscale,xMax,yMax;     /*The scaling factors */
extern int  MaxX,MaxY;

main()
{
int  index,freq,i;
unsigned int dec,acc,rate;

     slip= 0.06; X1= 5.56; X2=13.; XM=121.5;
     R1=5.8;   R2=7.27; Rail_Volt=200.; min_freq=0.; a=0;
     init_reg0.x.zeros=init_reg1.x.zeros=init_reg1.x.dontcare=0;
     init_reg2.x.zeros=cont_reg2.x.zeros=cont_reg01.x.dontcare=0
     rate=500;acc=dec=50;
     mini_freq=getfreq(min_freq);
     if(load_status()==NULL){
          write_mem();
          volt_freq();
          init(5,5);
          initialise();
     }
     else{
          if(test_mem()==NULL){
               write_mem();
               volt_freq();
               initialise();
          }
     }
     index=1;
     do{
          status();
          switch(getch()){
/*             case 'c': scanf("%d",&i);      */
/*                       setcarrier(i);       */
/*                       initialise();        */
/*                       break;               */
/*             case 'f': scanf("%d",&i);      */
/*                       setrange(i);         */
/*                       initialise();        */
/*                       break;               */
```

```
case 'l': printf("delay = %u\n",init_reg2.x.pdy);
          printf("delay = ?");
          if((i=getint())==-1)break;
          setdelay(i);initialise();break;
case 't': printf("deletion= %u\n",init_reg0.x.pdt);
          printf("deletion = ?");
          if((i=getint())==-1)break;
          setdeletion(i);
          printf("test");
          initialise();
          break;
case 's': if(getyes()==1){
                counter_set();
                initialise();
          }
          break;
case 'r': if(getyes()==1){
                counter_reset();
                initialise();
          }
          break;
case 'd': printf("direction = ?");
          if((i=getint())==-1)break;
                if(cont_reg01.x.for_rev==i)break;
                setfreq(0,acc,dec,rate);
                setdirection(i);
                control();
                setfreq(freq,acc,dec,rate);
                break;
case 'f': printf(" Start freq \t%f\n",min_freq);
          if(getyes()==1){
                scanf("%f",&min_freq);
                mini_freq=getfreq(min_freq);
          }
          break;
case 'i': inhibit();control();break;
case 'u': uninhibit();control();break;
case 'm': write_mem();break;
case 'n': read_mem();break;
case 'z': printf("acc = %u\n",acc);
          printf("\tacc = ?\t");
          if((i=getint())==-1)break;
          acc=(unsigned int)i;
          break;
case 'x': printf("dec = %u\n",dec);
          printf("\tdec = ?\t");
          if((i=getint())==-1)break;
          dec=(unsigned int)i;break;
case 'v': printf("rate = %u\n",rate);
          printf("\trate = ?\t");
          if((i=getint())==-1)break;
          rate=(unsigned int)i;
          break;
case 'k': volt_freq();break;
case 'p': print_spec();break;
case 0x1b:      if(save_status()){
                index=0;
                printf("status file saved\n");
```

```
                              }
                              break;
                  default   : printf("Power frequency = ? ");
                              if((i=getint())==-1)break;
                              if(i<(60*acc/4096)){
                                   printf("minimum required\n");
                                   break;
                              }
                              if((i<min_freq)&(i!=0)){
                                   printf("motor may not start agai
                                             load\n");
                              }
                              freq=i;
                              setfreq(freq,acc,dec,rate);break;
            }
      }while(index==1);
}


/*    ==============================================    */
/*    Routine to set the default initialising data      */
/*    ==============================================    */

void init(int pdy,int pdt)
{
     setcarrier(0);
     setrange(0);
     setdelay(pdy);
     setdeletion(pdt);
     setcounter(SET);


/*    ==================================================    */
/*    Routine to set the default control info of the MA818  */
/*    ==================================================    */

     cont_reg01.x.power_freq=(unsigned int)0;
     setoutput(INHIBIT);
     setdirection(1);
     cont_reg2.x.amp=(unsigned int)0;
     cont_reg01.x.overmod=0;
}
```

```
/*    ================================================    */
/*    Routine to set the voltage frequency relationship   */
/*    ================================================    */

void volt_freq()
{
unsigned int i;
float     t9,t1,t2,t3,t4,t5,range;
float     L1,L2,LM,w,R22,X1,X2,XM,R1,R2,slip;

     printf("motor parameters  values ");
     if(getyes()==1){
          printf("\nX1 = %f\t",X1);
          if(getyes()==1)scanf("%f",&X1);
          printf("\nX2 = %f\t",X2);if(getyes()==1)scanf("%f",&X2
          printf("\nXM = %f\t",XM);if(getyes()==1)scanf("%f",&XM
          printf("\nR1 = %f\t",R1);if(getyes()==1)scanf("%f",&R1
          printf("\nR2 = %f\t",R2);if(getyes()==1)scanf("%f",&R2
          printf("\ns  = %f\t",slip);
          if(getyes()==1)scanf("%f",&slip);
          printf("\nRail Voltage = %f\t",Rail_Volt);
          if(getyes()==1)scanf("%f",&Rail_Volt);
          printf("minimum start freq\t",min_freq);
          if(getyes()==1){
               scanf("%f",&min_freq);
               mini_freq=getfreq(min_freq);
          }
     }
     R22=R2/slip;
     L1=X1/314.159;
     L2=X2/314.159;
     LM=XM/314.159;
     range=setrange(0);
     t3=2.0*M_PI*range/4096.;
     t4=(256.*4.)/(M_PI*util*(Rail_Volt));
     t9=20.56*256./(util*Rail_Volt);
     for(i=1;i<4096;i++){
          w=2.0*i*M_PI*range/4096.;
          t5=R22*R22+w*w*L2*L2;
          t1=1+L1/LM+(R1*R22+w*w*L1*L2)/t5;
          t1*=t1;
          t2=(w*L1*R22-w*L2*R1)/t5-R1/(w*LM);
          t2*=t2;
          vf[i]=(unsigned int)ceil(t9+t4*w*sqrt(t1+t2));
     }
     for(i=0;i<4096;i++)if(vf[i]>254)vf[i]=254;
     vf[0]=vf[1];
     print_spec();
}
```

```
/*   ========================================      */
/*   Routine to set the speed of the motor          */
/*   ========================================      */
void setfreq(int freq,unsigned int acc,unsigned int dec,unsigned in
{
int  expected_freq,diff;

     if(cont_reg01.x.power_freq==0)uninhibit();
     expected_freq=getfreq(freq);
     if(cont_reg01.x.power_freq!=expected_freq){
          if(cont_reg01.x.power_freq>expected_freq){
               while(cont_reg01.x.power_freq>
                                        (expected_freq+dec-1)){
                    cont_reg01.x.power_freq-=dec;
                    setamplitude();
                    delay(rate_d);
                    control();
               }
          }
          else{
               while(cont_reg01.x.power_freq<
                                        (expected_freq-acc+1)){
                    cont_reg01.x.power_freq+=acc;
                    if(cont_reg01.x.power_freq<mini_freq){
                         printf("motor will not start\n");a=0;
                    }
                    else{
                         if(a==0){
                              setamplitude();
                              control();
                              delay(rate_s); a=1;
                         }
                         setamplitude();
                         delay(rate_a);
                         control();
                    }
               }
               if(a==0)cont_reg01.x.power_freq=0;
          }
     }
     if(a!=0){
          if(cont_reg01.x.power_freq!=expected_freq){
               if(cont_reg01.x.power_freq>expected_freq){
                    diff=cont_reg01.x.power_freq-expected_freq;
                    diff=rate_d*diff/dec;
               }
               else{
                    diff=expected_freq-cont_reg01.x.power_freq;
                    diff=rate_a*diff/acc;
               }
               cont_reg01.x.power_freq=expected_freq;
               setamplitude();
               delay(diff);
               control();
          }
     }
     if(cont_reg01.x.power_freq==0){inhibit();control();a=0;}
```

```
/*    ==========================================          */
/*    Routine to set the output voltage                   */
/*    ==========================================          */

void setamplitude()
{
unsigned int i;

      i=vf[cont_reg01.x.power_freq];
      cont_reg2.x.amp=i & (unsigned int)255;
      cont_reg01.x.overmod=(i & (unsigned int)511)>>8;
}


/*    ==========================================          */
/*    Routine to get the frequency of the PWM             */
/*    ==========================================          */

unsigned int getfreq(int freq)
{
float x;

      x=(196608.*4096.*freq*(1<<init_reg1.x.car_freq))/
                            (CLOCK*(1<<init_reg1.x.freq_range)
      if(x>=4096.){
            printf("frequency out of allowable range\n");
            return(cont_reg01.x.power_freq);
      }
      return((unsigned int)ceil(x));
}


/*    ============================================        */
/*    Routine to ensure only the correct values are input */
/*    ============================================        */

int getint(){
int  i,k;

      scanf("%d",&i);
      printf("retype your answer\t ");
      scanf("%d",&k);
      if(k!=i){
            printf("wrong: stop motor  \n" );
            return(-1);
      }
      return(i);
}
```

```
/*   -----------------------------------------------   */
/*   Routine to for yes or no tes                      */
/*   ===============================================   */

int getyes(){
int i;
char c;
     printf("\nDo you want to change it ?  ");
     c=getche();
     if(c=='y'){
          printf("\nenter new value\t");
          return(1);
     }
     return(0);
}

/*   ===============================================   */
/*   Routine to plot   graphics                        */
/*   ===============================================   */

print_spec(){
int  i,length;
int  xvalue,yvalue;
float     max,min,t1;

     max=vf[0];min=0.0;
     length=4000;
     for(i=1;i<length;i++){
          if(max<vf[i])max=(float)vf[i];
          else if(min>vf[i])min=(float)vf[i];
     }
     t1=Rail_Volt*util/256.;
     max=t1*max;
     xvalue=length/10;
     yvalue=(int)ceil(max/10.0);
     max=yvalue*10;
     length=xvalue*10;
     Initialize();        /* Set system into Graphics mode   */
     setviewport(0,0,MaxX,MaxY,1);
     scale((float)length,(float)(max-min));
     setbkcolor(WHITE);setcolor(BLUE);
     setlinestyle(SOLID_LINE,0,NORM_WIDTH);
     plot(10,10,xvalue,yvalue);
     setlinestyle(SOLID_LINE,0,NORM_WIDTH);
     setcolor(BLUE);
     for(i=0;i<=length;i++)glineto((float)i,t1*vf[i]);
     Pause();
     closegraph();        /* Return the system to text mode  */
     for(i=1;i<=length;i+=200)
          printf("%d\t%f\t%u\n",i,t1*vf[i],vf[i]);
     return(0);
}
```

```
/*    =========================================================
/*    Routine to save the values of the information sent to the
/*    registers before program is exited. The info is appended
/*    =========================================================

int   save_status()
{
FILE *fp;
      fp=fopen("status.dat","ab+");
      if(!fp){
           printf("status file could not be created\n");
           return(NULL);
      }
      fwrite(&init_reg0,sizeof(int),1,fp);
      fwrite(&init_reg1,sizeof(int),1,fp);
      fwrite(&init_reg2,sizeof(int),1,fp);
      fwrite(&cont_reg01,sizeof(int),1,fp);
      fwrite(&cont_reg2,sizeof(int),1,fp);
      fclose(fp);
      return(1);
}

/*    ======================================== */
/*    Routine to load the status of the MA818 */
/*    ======================================== */

int   load_status()
{
FILE *fp;
      fp=fopen("status.dat","rb");
      if(!fp){
           printf("error in opening status file\n");
           return(NULL);
      }
      if(fseek(fp,(long)-10,SEEK_END)){
           printf("error in seeking\n");
           fclose(fp);
           return(NULL);
      }
      fread(&init_reg0,sizeof(int),1,fp);
      fread(&init_reg1,sizeof(int),1,fp);
      fread(&init_reg2,sizeof(int),1,fp);
      fread(&cont_reg01,sizeof(int),1,fp);
      fread(&cont_reg2,sizeof(int),1,fp);
      fclose(fp);
      return(1);
}
```

```
/*    ==================================================        */
/*    Routine to print the information currently in MA818      */
/*    ==================================================        */

void status()
{
     printf("\ncarrier : %u\trange : %u\t",
          init_reg1.x.car_freq,init_reg1.x.freq_range);
     printf("delay 1  : %u\twidth t : %u\n",
          init_reg2.x.pdy,init_reg0.x.pdt);
     printf("counter : %u\t",init_reg0.x.counter);
     printf("Direction d : %u\t\t\t",cont_reg01.x.for_rev);
     printf("inhibit u(1),i(0) : %u\n",cont_reg01.x.outinhibit)
     printf("Frequency  : %u\t\t",cont_reg01.x.power_freq);
     printf("Amplitude  : %u  %u\n",
          cont_reg01.x.overmod,cont_reg2.x.amp);
     printf("m: write mem\tn: read mem\t");
     printf("z: acc\tx: dec\tv: rate\n");
     printf("g: load_status\tk: volt_freq\tESC: save\n\n");
     printf("MA818 >\t");
}
```

```
/*   -----------------------------------------          */
/*   SENDING INITIALISING DATA TO THE MA818 IC          */
/*   -----------------------------------------          */
void initialise()
{
unsigned char far   *pt;

     pt=0xd0000800;                      /* address of register r0 */
     outportb(0x3e6,(char)NULL);         /* rst = logic low        */
     *pt=init_reg0.y.high;
     pt++;
     *pt=init_reg1.y.high;
     pt++;
     *pt=init_reg2.y.high;
     pt++; pt++;
     *pt=(unsigned char)NULL;            /* writing to register r4 */
     control();            /* writing the control information     */
     outportb(0x3e5,(char)NULL);         /*  rst = logic high      */
}

/*          ---------------------------------          */
/*          SENDING CONTROL DATA TO THE MA818          */
/*          ---------------------------------          */

void control()
{
unsigned char far   *pt;

     pt=0xd0000800;              /* address of register r0 */
     *pt=cont_reg01.y.high;
     pt++;
     *pt=cont_reg01.y.low;
     pt++;
     *pt=cont_reg2.y.high;
     pt++;
     *pt=(char)0x03;             /*  writing to register r3 */
}
/*   ==================================================      */
/*   Routine to read the RAM where the wavefrom is stored    */
/*   ==================================================      */

void read_mem()
{
unsigned char far  *pt1, *pt2;
unsigned char  value;
int        i;

     pt1=0xd0000000;
     pt2=0xd0000400;
     for(i=0;i<1024;i++){
          value=*pt2;
          value=value<<4;
          value=value|(*pt1&0x0f);
          pt1++;
          pt2++;
     }
}
```

```
/*    =====================================================    */
/*    Routine to test whether the memory contains the data    */
/*    =====================================================    */
int test_mem()
{
unsigned char far  *pt1, *pt2;
unsigned char  value,test;

     pt1=0xd0000305;
     pt2=0xd0000705;
     test=(unsigned char)(10);
     value=*pt2;
     value=value<<4;
     value=value|(*pt1&0x0f);
     if(value!=test)return(NULL);
     return(1);
}

/*    ==============================================    */
/*    routine to write to the RAM the waveform    */
/*    ==============================================    */
void write_mem()
{
unsigned char far  *pt1, *pt2;
unsigned char  value,t1;
float     max,rom[780],am[30];
int       i,j,no[30],c;
     pt1=0xd0000000;
     pt2=0xd0000400;
     printf("\nnumber of components = \t");scanf("%d",&c);
     printf("No.\tAmp\n");printf("------------\n");
     for(i=0;i<c;i++)scanf("%d%f",&no[i],&am[i]);}
     printf("\nUtilisation factor = \t");scanf("%f",&util);
     max=0.0;
     for(j=0;j<=767;j++){
          rom[j]=0;
          for(i=0;i<c;i++)
               rom[j]+=am[i]*sin(no[i]*M_PI*(float)j/767.);
          if(max<rom[j])max=rom[j];
     }
     for(j=0;j<=767;j++){
          if(max==0){ value=0;}
          else{
               value = (unsigned char)((255./max)*rom[j]);
          }
          *pt1=value;
          *pt2=(value>>4);
          pt1++;
          pt2++;
     }
     memset(pt1,15,250);          /* filling the rest with FF  */
     memset(pt2,15,250);
     pt1=0xd0000305;              /* writing a test code   */
     pt2=0xd0000705;              /* to indicate data is valid  */
     value = (unsigned char)(10);      /* used by test_mem()    */
     *pt1=value;
     *pt2=(value>>4);
}
```

```c
/*    ==========================================    */
/*    Routine to set the carrier frequency of the PWM    */
/*    ==========================================    */

float setcarrier(int carrier_bits)
{
     if(carrier_bits>5)return(NULL);
     init_reg1.x.car_freq=(unsigned int)carrier_bits;
     return(CLOCK/(512*(ONE<<init_reg1.x.car_freq)));
}

/*    ==========================================    */
/*    Routine to set the frequency range of the MA818    */
/*    ==========================================    */

float setrange(int range_bit)
{
     if(range_bit>6)return(NULL);
     init_reg1.x.freq_range=(unsigned int)range_bit;
     return(((CLOCK*(ONE<<init_reg1.x.freq_range))/
                    (196608.*(ONE<<init_reg1.x.car_freq))));
}

/*    ========================================= */
/*    Routine to set the pulse delay            */
/*    ========================================= */

float setdelay(int pdy_bits)
{
     if(pdy_bits>64)return(NULL);
     init_reg2.x.pdy=(unsigned int)(64-pdy_bits);
     return((1000000.*(ONE<<init_reg1.x.car_freq)*pdy_bits)/CLOC
}

/*    ========================================= */
/*    Routine to set pulse deletion             */
/*    ========================================= */

float setdeletion(int pdt_bits)
{
     if(pdt_bits>128)return(NULL);
     init_reg0.x.pdt=(unsigned int)(128-pdt_bits);
     return((1000000.*(ONE<<init_reg1.x.car_freq)*pdt_bits)/CLOC
}

/*    ========================================= */
/*    Routine to set the counter bit of the MA818  */
/*    ========================================= */

void setcounter(int counter_bit)
{
     if(counter_bit)     counter_set();
     else counter_reset();
}
```

```
/*   ========================================   */
/*   Routine to set the MA818                   */
/*   ========================================   */

void counter_set()
{
     init_reg0.x.counter=1;
}

/*   ========================================   */
/*   Routine to reset the MA818 the MA818       */
/*   ========================================   */

void counter_reset()
{
     init_reg0.x.counter=0;
}

/*   ============================================   */
/*   Routine to set the direction bit of the MA818  */
/*   ============================================   */

void setdirection(int direction_bit)
{
     if(direction_bit)reverse();
     else forward();
}

/*   ============================================ */
/*   Routine to set the direction bit to forward  */
/*   ============================================ */

void forward()
{
     cont_reg01.x.for_rev=0;
}

/*   ============================================ */
/*   Routine to set the direction bit to reverse  */
/*   ============================================ */

void reverse()
{
     cont_reg01.x.for_rev=1;
}

/*   ============================================ */
/*   Routine to set the inhibit bit of the MA818  */
/*   ============================================ */

void setoutput(int inh_bit)
{
     if(inh_bit) uninhibit();
     else inhibit();
}
```