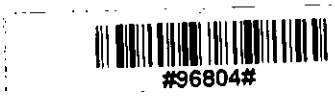
Analysis and Implementation of a Look Up Table Based Hybrid PWM Scheme for Voltage Source Inverters

by

Muhammad Quamruzzaman

MASTER OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING



Department of Electrical and Electronic Engineering
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

2002

The thesis titled "Analysis and Implementation of a Look Up Table Based Hybrid PWM Scheme for Voltage Source Inverters" Submitted by Muhammad Quamruzzaman Roll No: 9406116F Session 1993-94-95 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Electrical and Electronic Engineering on May 7, 2002.

BOARD OF EXAMINERS

1. _____

Dr. Kazi Mujibur Rahman — Chairman

Associate Professor
Department of EEE, BUET
Dhaka-1000, Bangladesh

2. _____

Dr. M. A. Choudhury — Member

Professor
Department of EEE, BUET
Dhaka-1000, Bangladesh

3. _____

Dr. M. M. Shahidul Hassan — Member (Ex-officio)

Professor and Head
Department of EEE, BUET
Dhaka-1000, Bangladesh

4. _____

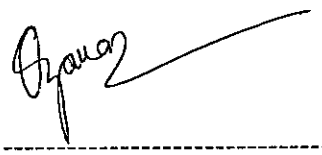
Dr. Farruk Ahmed — Member (External)

Professor
Department of Applied Physics and Electronics
Dhaka University
Bangladesh

Candidate's Declaration

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate



Muhammad Quamruzzaman

Contents

List of Figures

List of Abbreviations

ASIC	Application Specific Integrated Circuit
CPU	Central Processing Unit
DF	Distortion Factor
EPROM	Erasable Programmable Read Only Memory
LUT	Look Up Table
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
PIA	Peripheral interface Adapter
PPI	Programmable Peripheral Interface
PWM	Pulse Width Modulation
RAM	Random Access Memory
RC	Ripple Carry
ROM	Read Only Memory
RSHEPWM	Regular Sampled Harmonic Elimination Pulse Width Modulation
RSHMPWM	Regular Sampled Harmonic Minimized Pulse Width Modulation
RSPWM	Regular Sampled Pulse Width Modulation
SHEPWM	Selective Harmonic Elimination Pulse Width Modulation
SVPWM	Space Vector Pulse Width Modulation
THD	Total Harmonic Distortion
VSI	Voltage Source Inverter

Acknowledgements

At first, the author expresses his heartiest gratitude to the Almighty Allah, the most beneficent, the most merciful, for giving him 'toufique' to complete this work successfully.

The author expresses his gratitude to his supervisor Dr. Kazi Mujibur Rahman, Associate Professor, Department of EEE for his helpful suggestions, encouragement and invaluable assistance in the process of completing this work. His constructive suggestions and supply of research materials are gratefully acknowledged. Sincerest thanks to all of those who extended their hands, directly or indirectly, to complete this research work.

The author also pays his profound gratitude to his parents, relatives and friends for their inspiration towards the completion of this work.

Abstract

Different Pulse Width Modulation (PWM) techniques have been developed by researchers for the control of modern inverters. Many implementation schemes are available in the literature for the real time generation of PWM patterns. Real-time PWM patterns are generated either by on-line computation using Microprocessor and Programmable Timers or by look-up-table (LUT) based off-line computation. LUT based schemes require huge memory for storing PWM patterns for an inverter with wide operating range, which restricted the use of such schemes in industry. Also, other microcomputer-based schemes require high-speed processors for generating PWM patterns for wide operating range. In this research work, a hybrid technique is used where both processor and LUT are used that improve the overall operating range of a PWM scheme. In the first step, the equation of pulse width of PWM patterns is simplified so that the scheme can be implemented using normal microprocessor performing only integer type calculations. Using this simplified equation the PWM patterns are formulated using MATLAB program. The formulated byte patterns for three phases are stored in three 32k EPROMs as Look-up-Table (LUT). Sine values, generated using the MATLAB program, are used in the Assembly Language program for the microprocessor to calculate the duty cycles of the PWM pulses. The processor calculates the duty cycles of the PWM pulses for the three phases for a specified modulation index and frequency once in each carrier period. The duty cycles are then normalized on a scale of 128. The processor sends the three duty cycles (for three phases) to a Programmable Peripheral Interface (PPI) having three 8-bit ports. A 12-bit binary counter, connected to the lower 8-bit address lines (A0-A7) scans the 256 stored samples of the EPROM. The D0 and D1 line of the EPROM data bus present the real time PWM pulses for the upper and lower leg switches of a phase of the inverter. As the counter reaches its maximum count value (255), it sends a ripple carry (RC) output. This RC signal is used to prompt the microprocessor to send the duty cycles of the next carrier period to the respective EPROMs. The processor keeps track of the number of pulses sent. When one fundamental period is reached, the processor points to the first carrier cycle. The performance of the scheme is tested on a three-phase MOSFET inverter. Data of real time PWM voltages are collected using data acquisition card and both the patterns and frequency spectrums are plotted. The experimental plots are compared with simulated plots. Detail modeling, analysis and experimental results are presented.

Chapter 1

Introduction

Microprocessor control of static power inverters has become popular because of improved reliability, better performance and greater flexibility. Modern inverters are controlled by pulse width modulation (PWM) techniques. Microprocessor based PWM schemes are extensively used for efficient control of variable frequency inverters. PWM schemes are implemented using a microprocessor, I/O devices and programmable timers. For variable speed drives, multiple programmable timers are normally used in a microprocessor scheme [1]. Use of multiple timers and dedicated hardware make the scheme complex. Drive manufacturers normally use dedicated ASIC chips for PWM controllers to reduce the number of environmental interfacing circuits. There are different types of implementation schemes available for the control of PWM inverters.

1.1 Review of PWM Techniques

Different PWM techniques are available in the literature developed by the researchers for the control of voltage, frequency and spectral distribution of static inverters. These techniques include

- Regular Sampled PWM (RSPWM) [2],[3]
- Delta PWM [4],[5]
- Selective Harmonic Elimination PWM (SHEPWM) [6]
- Regular Sampled Harmonic Minimized PWM (RSHMPWM) [7],[1]
- Regular Sampled Harmonic Elimination PWM (RSHEPWM) [8]
- Space Vector PWM (SVPWM) [9]

The regular sampled PWM generates a stream of regularly spaced pulses whose widths are modulated by the amplitude of the sine-modulating wave. The spectral distribution of RSPWM depends on the carrier frequency. The harmonic spectrum contains the fundamental modulating frequency and the carrier. The intermediate harmonic contents are negligible by the nature of the modulation process. By choosing a relatively high carrier frequency, a small filter is sufficient to filter out all the harmonic contents to acceptable range. The Selective Harmonic Elimination PWM also generates a regularly sampled pulse stream and the switching edges are obtained analytically by eliminating certain specific harmonic contents. The SHEPWM technique is computation extensive and is generally calculated off line. This technique increases the order of

harmonics and reduces the size of output filter. The Harmonic Minimized Sampled PWM is similar to RSPWM but here third harmonic is added intentionally to some extent to the sine-modulating wave. This technique supports linear over modulation and is suitable for certain applications requiring high voltages. In Regular Sampled Harmonic Elimination PWM, the leading edge of a PWM pattern is calculated from the reference modulating signal and the trailing edge from the shifted reference signal. The resulting PWM pattern shows improved harmonic spectrum.

1.1.1 Regular Sampled PWM

Traditionally PWM control is accomplished using an analogue technique employing natural sampling strategy [10]. The basic circuit uses a comparator to detect the intersection of the triangular (carrier) wave with a sinusoidal modulating wave. The resultant output is a periodic pulse train that switches at points determined by the relative magnitude and frequency of its two inputs. Since the switching edge of the width modulated pulse is determined by the instantaneous intersection of the two waves, the resultant pulse width is proportional to the amplitude of the modulating wave at the instant where switching occurs. This natural sampling process is therefore "real-time" in nature providing instantaneous response to demand variations, such as a change in amplitude of modulating wave, but with minimal choice strategy. This has two important consequences: the first is that the centers of the pulses in the resultant PWM wave are not equidistant or uniformly spaced and, secondly, it is not possible to define the widths of the pulses using analytic expressions. Unfortunately, the equations describing the natural sampled switching angles are transcendental and therefore not suitable for software implementation and hence needs alternate solution [2].

The need to incorporate a PWM controller within a digital or discrete environment based on a microprocessor led to the development of a Regular Sampling technique in the early 1970's [2]. The fundamental difference is that the modulating sine wave is regularly sampled. This can be visualized as shown in Fig. 1.1 for symmetric regular sampled double edge PWM. For microprocessor implementation it is more informative to consider regular sampling as a digital process of sampling a sinusoidal modulating wave a, at regularly spaced intervals to produce sinusoidally weighted digital samples of the modulating wave, as presented by Fig. 1.2 (b).

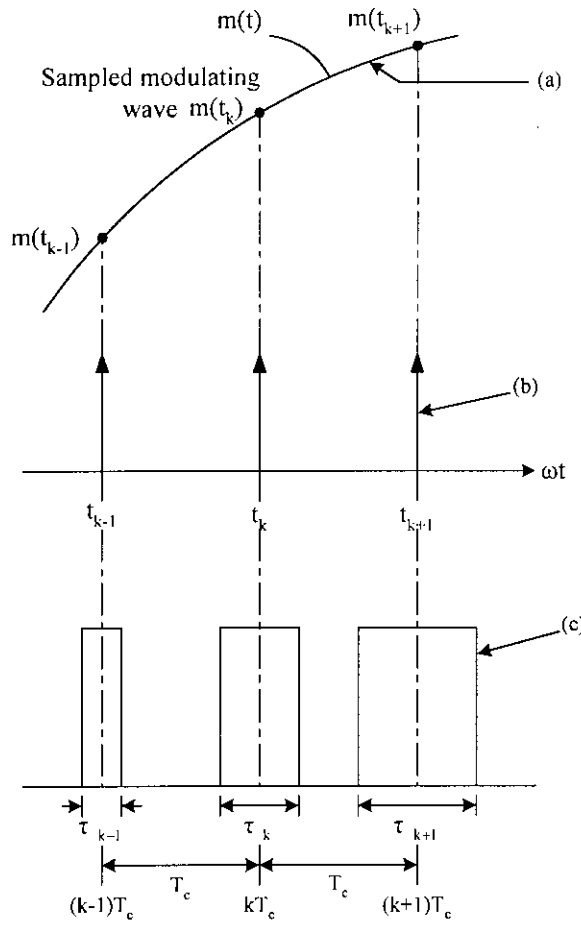


Fig. 1.1. Regular sampled symmetric double edge modulation, (a) Modulating wave (b) Regular-Sampled waveform (c) PWM waveform.

As shown in Fig.1.1, the process consists of taking samples $m(t_k)$ of a modulating wave m(t) at regularly spaced time intervals $t_k = kT_c$, where T_c corresponds to the sampling time period. These samples of the modulating wave $m(t_k)$ are subsequently used to modulate the widths, τ_k , of the PWM waveform. There are, in general, three ways in which the width of the pulse may be modulated by the samples $m(t_k)$ as defined in [2]. These can be categorized as either using "single-edge" or "double-edge" modulation. In double-edge modulation, both edges of the pulse are modulated according to $m(t_k)$, either "symmetrically" [2] or "asymmetrically" [3],[11-17]about the sampling positions $t_k = kT_c$.

In the modulation process shown in Fig.1.1 [18], the sampled modulating values $m(t_k)$ are used to modulate the PWM pulses symmetrically, with respect to the sampling instant $t_k=kT_c$, such that the pulsewidth τ_k is proportional to $\sin(\omega t_k)$ (for sinusoidal modulation), resulting in each edge of the pulse modulated equally; hence the term "symmetric double-edge" PWM.

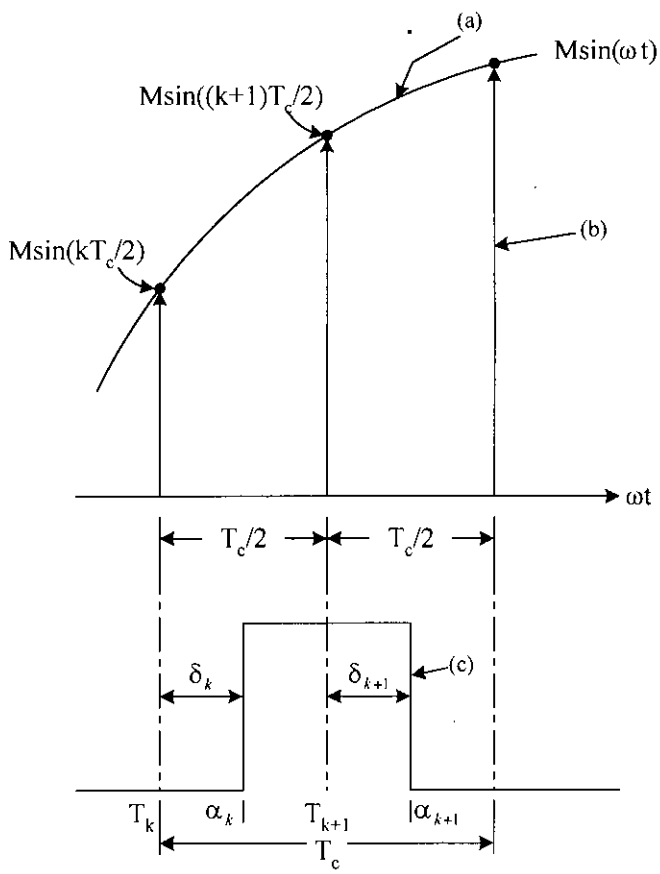


Fig. 1.2. Details of k^{th} pulse in two-level regular-sampled asymmetric PWM, (a) Modulating wave (b) Regular-Sampled wave (c) PWM voltage

As shown in Fig.1.2 two digital samples *b* per carrier cycle are generated. The first digital sample in the carrier cycle is used to sinusoidally modulate the "leading-edge" of the PWM pulse *c* , and the second digital sample is used to sinusoidally modulate the "trailing-edge" of the PWM pulse. Thus, each edge of the PWM pulse is modulated by a different amount with respect to the regularly spaced pulse centers. The equation used to calculate the PWM pulse-widths for a sinusoidal modulating wave takes the form [2], [12]

$$\tau_k = T/2 + (MT/4)[\sin(kT/2) + \sin(k+1)T/2] \tag{1.1}$$

The switching equations describing asymmetric regular-sampled PWM are given by,

$$\delta_k = T/4 - MT/4\sin(kT/2) \tag{1.2}$$

$$\delta_{k+1} = T/4 + MT/4\sin[(k+1)T/2] \tag{1.3}$$

The modulation process involved in the generation of symmetric regular sampled PWM waveform is shown in Fig.1.3. [19]

The expressions for the on-time pulse width and switching points are derived below:

Let us take one carrier cycle as shown in Fig. 1.4 for the calculation of pulse-width and switching points of PWM voltage waveform. From triangles BOA and CBD, ∠BOA= ∠CBD. Hence the tangents at angle O and B are same. This means that

$$\frac{BA}{OA} = \frac{CD}{BD} \tag{1.4}$$

$$\text{or,} \quad BD = CD \times \frac{OA}{BA} \tag{1.5}$$

$$\text{or,} \quad BD = \frac{T}{4} \times \frac{A_m \sin\{\omega_m t(i)\}}{A_c} \tag{1.6}$$

The total pulse width at the i[th] sample is 2 × (OA + BD) and is given by

$$t_w(i) = 2 \times \left[\frac{T}{4} + \frac{T}{4} \times \frac{A_m \sin\{\omega_m t(i)\}}{A_c} \right]$$

$$= 0.5T\left[1 + M \sin\{\omega_m t(i)\}\right] \tag{1.7}$$

where, M = modulation index

T = period of triangular carrier

ω_m = angular frequency of modulating signal

$t(i)$ = carrier sampling instant of the modulating wave = $T/2,\ 3T/2,\ 5T/2$.....

Now the leading edge of PWM pulse at the i[th] sample is given by

$$t_L(i) = t(i) - \frac{t_w(i)}{2}$$

$$= t(i) - 0.25T\ [1 + M \sin\{\omega_m t(i)\}] \tag{1.8}$$

The trailing edge of PWM pulse at the i[th] sample is given by

$$t_R(i) = t(i) + \frac{t_w(i)}{2}$$

$$= t(i) + 0.25T\ [1 + M \sin\{\omega_m t(i)\}] \tag{1.9}$$

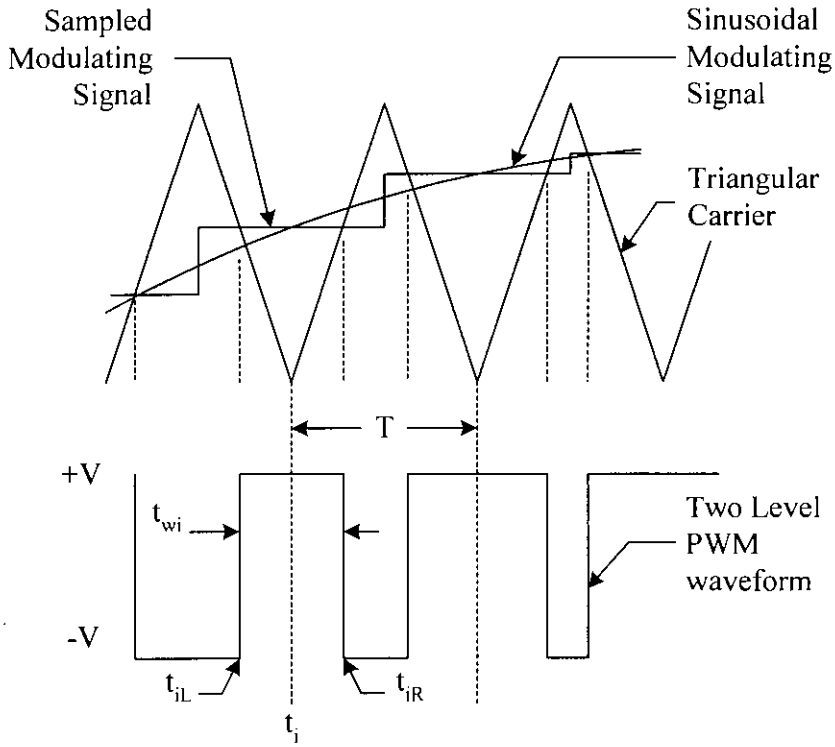


Fig.1.3. Regular sampled symmetric PWM

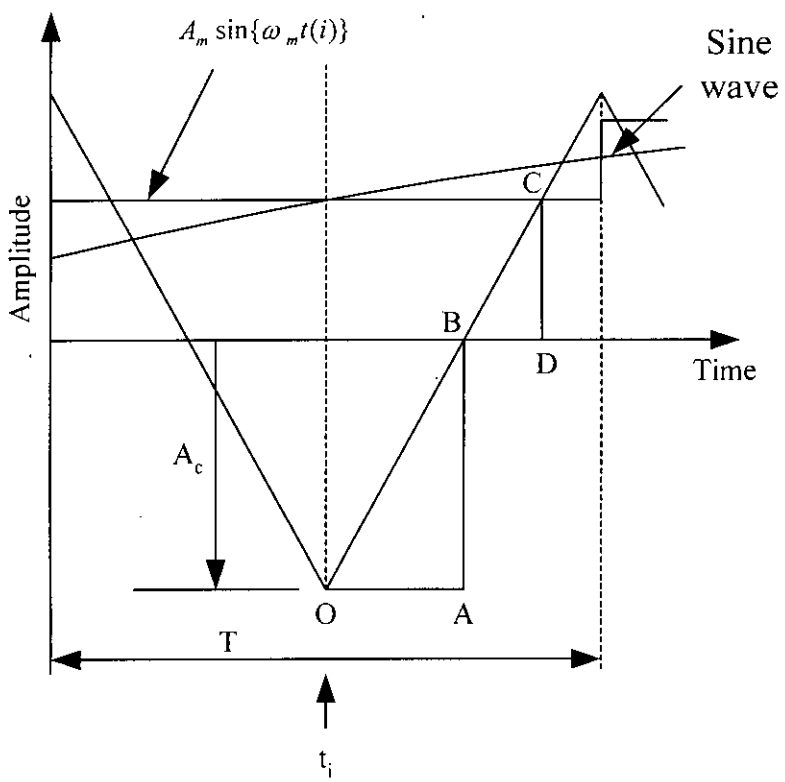


Fig. 1.4 Switching points calculation from regular sampled symmetric PWM waveform

The modulation process involved in the generation of asymmetric sampled PWM is shown in Fig.1.5. The expressions for the on time pulse width and switching points are derived below:

In this process, two samples are taken within a carrier period as shown in Fig. 1.6. The calculation is given for j^{th} sample and i^{th} sample.

From equation (1.5) it can be written

$$BD = \frac{T}{4} \times \frac{A_m \sin\{\omega_m t(i)\}}{A_c} \tag{1.10}$$

$$\text{and } B'D' = C'D' \times \frac{OA'}{B'A'} \tag{1.11}$$

$$\text{or, } B'D' = \frac{T}{4} \times \frac{A_m \sin\{\omega_m t(j)\}}{A_c} \tag{1.12}$$

Where, t(j) = t(i) - T/2.

Now the on-time pulse width is given by,

$$\begin{aligned} t_w(i) &= OA + BD + OA' + B'D' \\ &= \frac{T}{4} + \frac{T}{4} \times \frac{A_m \sin\{\omega_m t(i)\}}{A_c} + \frac{T}{4} + \frac{T}{4} \times \frac{A_m \sin\{\omega_m t(j)\}}{A_c} \\ &= 0.5T[1 + 0.5M\{\sin\{\omega_m t(i)\} + \sin\{\omega_m t(j)\}\}] \end{aligned} \tag{1.13}$$

The leading edge of the PWM pulse is given by,

$$\begin{aligned} t_L(i) &= t(i) - (OA' + B'D') \\ &= t(i) - \left[\frac{T}{4} + \frac{T}{4} \times \frac{A_m \sin\{\omega_m t(j)\}}{A_c}\right] \\ &= t(i) - 0.25T[1 + M \sin\{\omega_m t(j)\}] \end{aligned} \tag{1.14}$$

Similarly, the trailing edge of the PWM pulse can be derived as

$$t_R(i) = t(i) + 0.25T[1 + M \sin\{\omega_m t(i)\}] \tag{1.15}$$

Advantages of regular sampling process

As a direct result of regular-sampling process the width of a pulse is proportional to the amplitude of each sample m(t_k) and therefore the area of the pulse is proportional to the area of the modulating wave m(t) over a sampling (or carrier) period T_c [2].

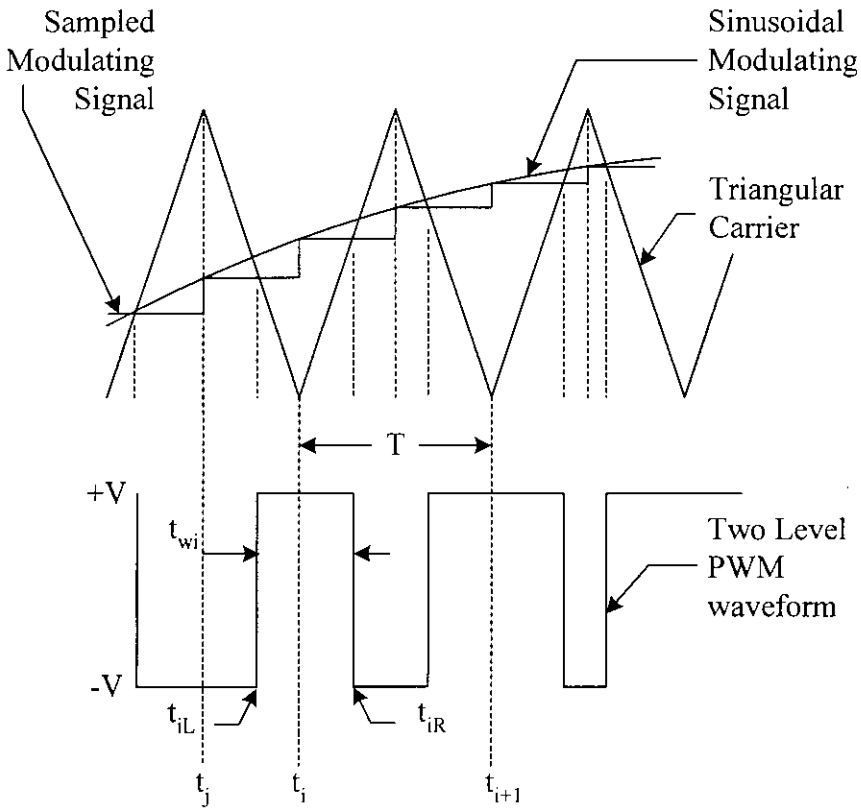


Fig. 1.5. Regular sampled asymmetric PWM

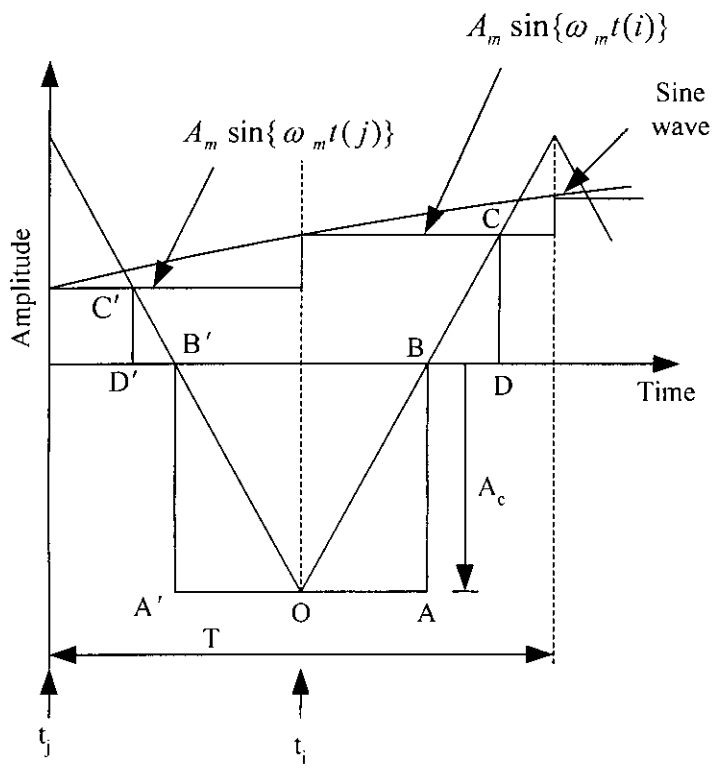


Fig. 1.6. Switching points calculation procedure from regular sampled asymmetric PWM waveform

Moreover, because of the regular sampling process the PWM pulse positions are regularly spaced and pulsewidths are precisely defined such that it is possible to derive a simple trigonometric equation to calculate the PWM pulsewidths [2]. Using this simple pulse width equation it is possible to generate regular sampled PWM directly in real time using an extremely efficient microprocessor based software algorithm. Alternatively, the simple pulsewidth equation can be used to model the regular sampled PWM process efficiently for computer-aided design purposes [12-14].

1.1.2 Delta PWM

In delta modulation [4], a triangular wave is allowed to oscillate within a defined window ΔV above and below the reference sine wave v_r. The inverter switching function, which is identical to the output voltage v_o is generated from the vertices of the triangular wave v_c as shown in Fig.1.5. It is also known as hysteresis modulation. If the frequency of the modulating wave is changed keeping the slope of the triangular wave constant, the number of pulses and pulse widths of the modulated wave would change. The fundamental output voltage can be up to $1V_s$ and is dependent on the peak amplitude A_r and frequency f_r of the reference voltage. The delta modulation can control the ratio of voltage to frequency, which is a desirable feature in ac motor control.

1.1.3 Selective Harmonic Elimination PWM (SHEPWM)

Through this technique, certain harmonics (e.g. third, fifth, ninth etc) can be eliminated from the output voltage of inverter.

The instantaneous output voltage of inverter [20],

$$v_o = \sum_{n=1,3,5}^{\infty} \frac{4V_s}{n\pi} \sin\frac{n\beta}{2} \cos n\left(\omega t - \frac{\beta}{2}\right) \tag{1.16}$$

where, β = delay or displacement angle. From the above equation it is seen that the nth harmonic can be eliminated by a proper choice of displacement angle β, if

$$\sin\frac{n\beta}{2} = 0$$

$$\text{or,} \qquad \beta = \frac{360^0}{n} \tag{1.17}$$

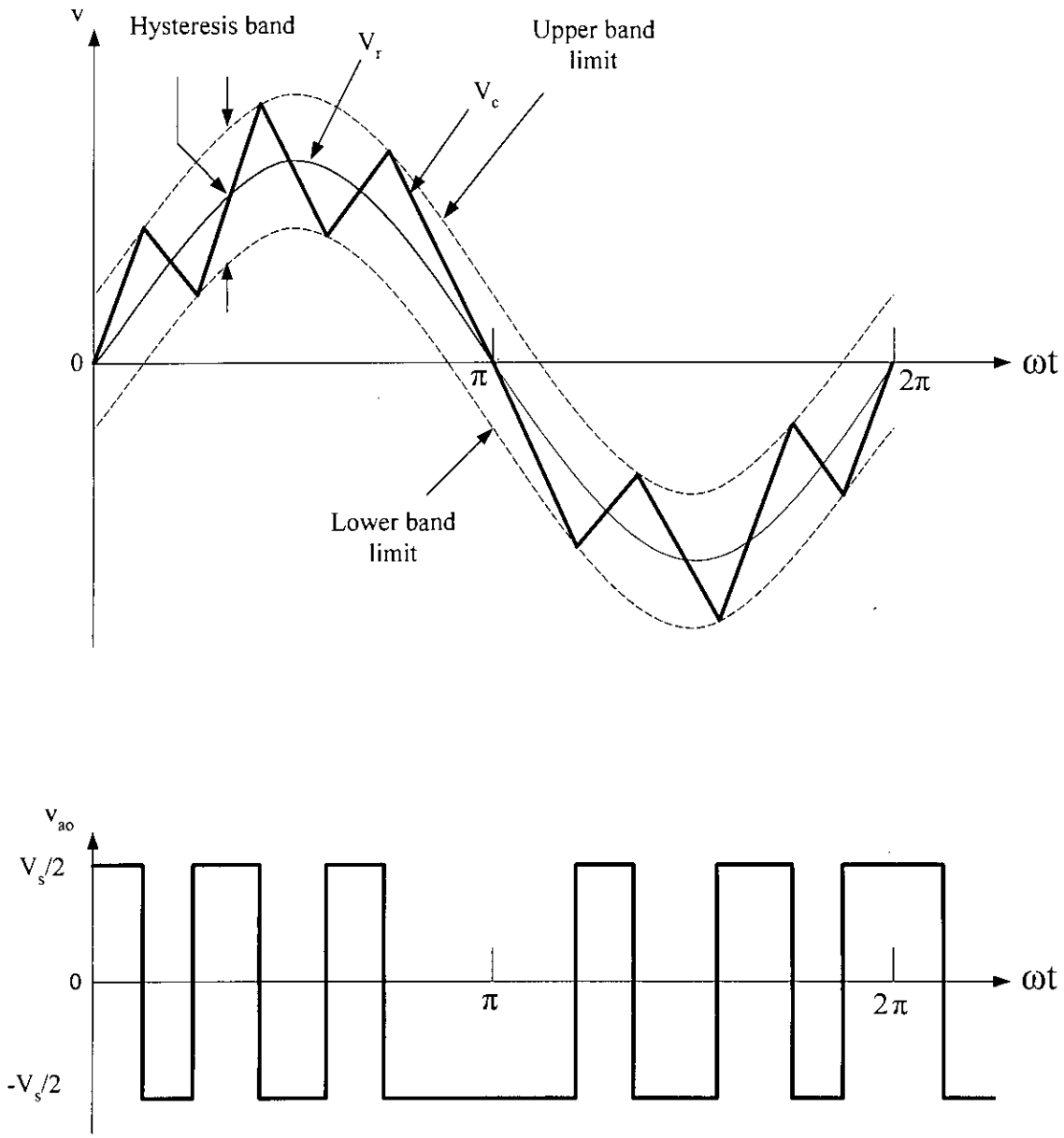


Fig.1.7. Delta Modulation

and the third harmonic will be eliminated if $\beta = 360^0/3 = 120^0$. Introducing a pair of symmetrically placed bipolar voltage notches as shown in Fig.1.8 can eliminate a pair of unwanted harmonics at the output of single-phase inverters.

The Fourier series of output voltage can be expressed as

$$v_o = \sum_{n=1,3,5.....}^{\infty} B_n \sin \omega t \tag{1.18}$$

where,

$$B_n = \frac{4V_s}{\pi}\left[\int_0^{\alpha_1} \sin n\omega t d(\omega t) - \int_{\alpha_1}^{\alpha_2} \sin n\omega t d(\omega t) + \int_{\alpha_2}^{\pi/2} \sin n\omega t d(\omega t)\right]$$

$$= \frac{4V_s}{\pi}\frac{1 - 2\cos n\alpha_1 + 2\cos n\alpha_2}{n} \tag{1.19}$$

Equation (1.19) can be extended to m notches per quarter wave

$$B_n = \frac{4V_s}{n\pi}(1 - 2\cos n\alpha_1 + 2\cos n\alpha_2 - 2\cos n\alpha_3 + 2\cos n\alpha_4 - .....) \tag{1.20}$$

The third and fifth harmonics would be eliminated if $B_3 = B_5 = 0$ and equation (1.13) gives the necessary equations to be solved.

$$1 - 2\cos 3\alpha_1 + 2\cos 3\alpha_2 = 0 \qquad or \qquad \alpha_2 = \frac{1}{3}\cos^{-1}(\cos 3\alpha_1 - 0.5) \tag{1.21}$$

$$1 - 2\cos 5\alpha_1 + 2\cos 5\alpha_2 = 0 \qquad or \qquad \alpha_1 = \frac{1}{5}\cos^{-1}(\cos 5\alpha_2 + 0.5) \tag{1.22}$$

Equations (1.21) and (1.22) can be solved for α_1 and α_2. The modified sinusoidal pulse width modulation (MSPWM) technique can be applied to generate the notches.

The output voltages of two or more inverters may be connected in series through a transformer as shown in Fig.1.10 to reduce or eliminate certain unwanted harmonics.

The output of first inverter,

$$v_{01} = A_1 \sin \omega t + A_3 \sin 3\omega t + A_5 \sin 5\omega t + ... \tag{1.23}$$

Since the output of the second inverter, v_{02} is delay by $\pi/3$,

$$v_{02} = A_1 \sin(\omega t - \pi/3) + A_3 \sin 3(\omega t - \pi/3) + A_5 \sin 5(\omega t - \pi/3) + ... \tag{1.24}$$

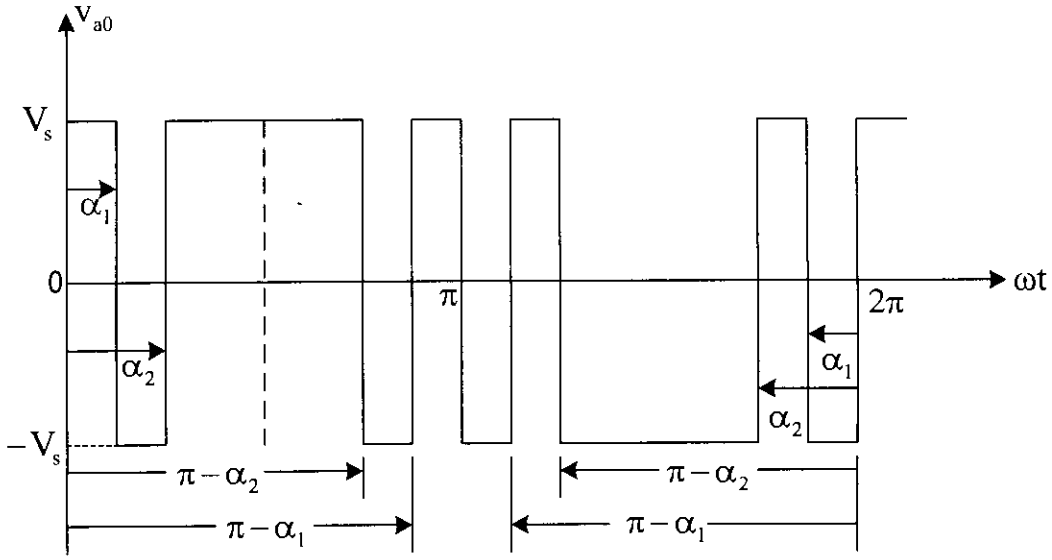


Fig. 1.8. Output voltage with two bipolar notches per half wave

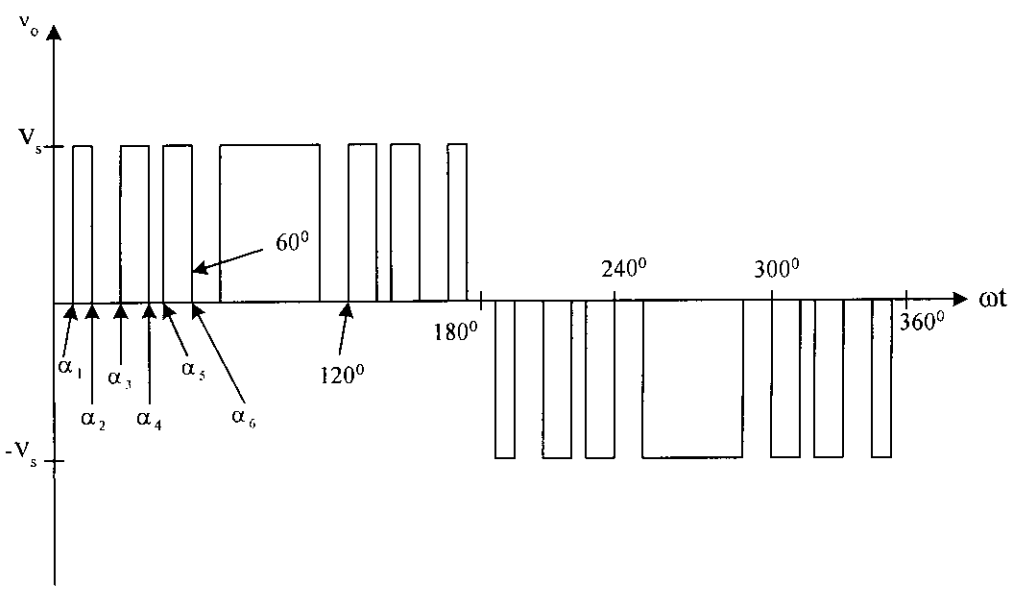


Fig. 1.9. Output voltage for MSPWM

The resultant voltage is,

$$v_0 = v_{01} + v_{02} = \sqrt{3}\left[A_1 \sin(\omega t - \pi/6) + A_5 \sin 5(\omega t + \pi/6) + ...\right] \quad (1.25)$$

Thus, a phase shifting of $\pi/3$ and combining voltages by transformer connection would eliminate third (and all triplen) harmonics. The harmonic elimination techniques, which are suitable only for fixed output voltage, increase the order of harmonics and reduce the sizes of output filter. However, this advantage should be weighed against the increased switching losses of power devices and increased iron losses in the transformer due to higher harmonic frequencies. The SHEPWM technique is computation intensive and is not suitable for on-line real time implementation. Rather the computations are done off line

1.1.4 Harmonic Minimized PWM

In a balanced three phase star connected load there is no path for the third harmonics and its integer multiples to flow. Hence, a PWM waveform containing triplen harmonic contents is not harmful for star connected load. It has been observed [1] that addition of 25% third harmonic component to the sine-modulating wave gives linear overmodulation facility up to 20%. It minimizes the THD, while maximizing the fundamental voltage. The injection of harmonic component to the modulating reference does not affect the total harmonic distortion, THD significantly. The modulating function is given by [1],

$$F(t) = M\left\{\sin(\omega_m t) + 0.25\sin(3\omega_m t)\right\} \quad (1.26)$$

The modulating signal with third and ninth harmonic injections is shown in Fig.1.11. The pulse widths and switching instants can be obtained as those of symmetric and asymmetric regular sampled PWM.

1.1.5 Regular Sampled Harmonic Minimized PWM

Based on the sampled version of the modulating wave F(t) [7], the optimal PWM pulse-widths can be defined using [21]

$$F(T_k) = \sin(T_k) + \frac{1}{4}\sin(3T_k) \quad (1.27)$$

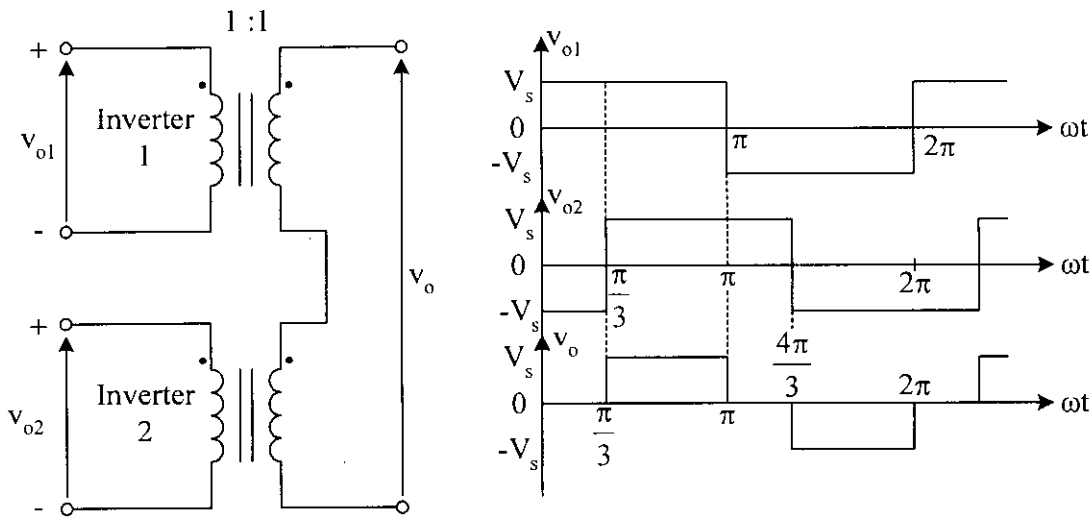


Fig. 1.10. Elimination of harmonics by transformer connection

(a). Circuit (b). Waveforms

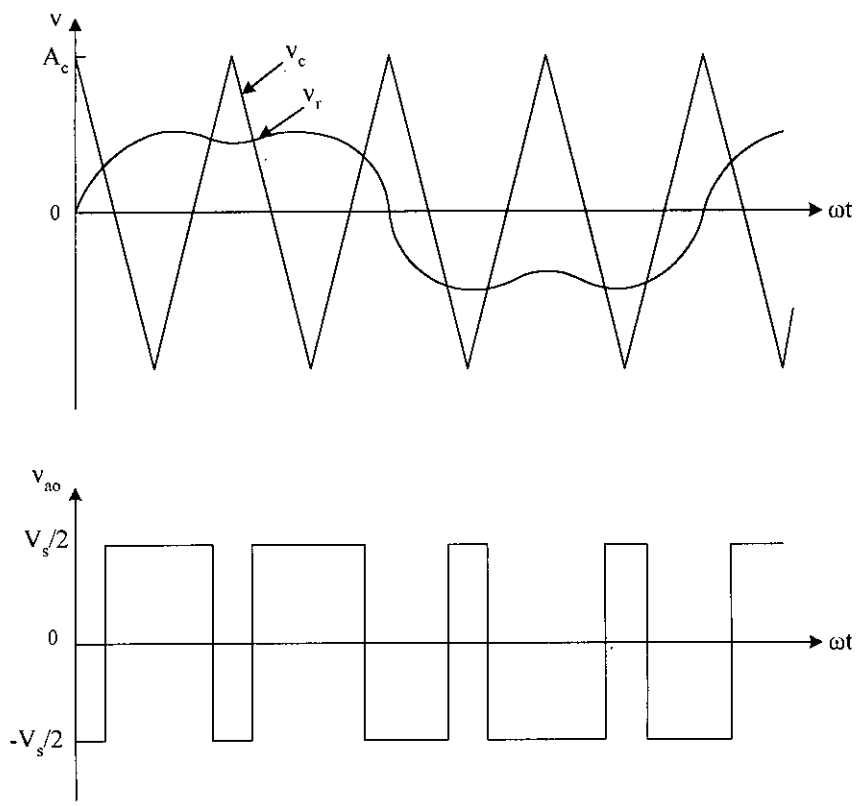


Fig. 1.11. Selected harmonic injection modulation (a). Gate signal generation (b). Output voltage

where, $F(T_k)$ represents the sampled values of $F(t)$ at regular sampling time instants $T_k=kT/2$. The addition of the 25% third harmonic minimizes the THD, while maximizing the fundamental voltage[16].

Based on (21) the optimal PWM pulse-widths can be defined as:[7]

Prepulse angle defining leading edge:

$$\delta_k = T/4 - (MT/4).F(T_k) \tag{1.28}$$

Postpulse angle defining trailing edge:

$$\delta_{k+1} = T/4 + (MT/4).F(T_{k+1}) \tag{1.29}$$

Equations (1.28) and (1.29) can be used as the basis for simple and efficient microprocessor software algorithm for real-time generation of optimal PWM.

1.1.6 Regular Sampled Harmonic Elimination PWM

The regular sampling approach eliminates many difficulties and problems encountered in harmonic elimination PWM. For example, in the past it has been necessary to employ significant off-line computing techniques and computing time to determine the relationships between the PWM switching angles and fundamental voltage. In addition, these switching angle-voltage characteristics were transcendental and therefore could not be solved on-line using a microprocessor-based PWM controller. Thus, extensive look-up tables (LUT) were necessary to implement these PWM strategies.

It has been observed that a PWM pattern close to Selective Harmonic Elimination PWM (SHEPWM) is possible with a regular sampling process[8]. In SHEPWM a PWM waveform is defined in terms of unknown switching angles and is solved numerically using a fast computer. For high power drives with thyristor switches this technique offers good performance at higher modulating frequencies. Huge calculations are involved in the computation process and hence these methods are not suitable for on-line control in digital implementation. In RSHEPWM scheme, two modulating references are used with a phase shift between them. The leading edge of a pulse is obtained from the original reference and trailing edge from the shifted reference. The shifted phase angle, β, is obtained from the carrier to modulating frequency ratio. The switching point calculations involve equations similar to symmetric and asymmetric sampled PWM.

1.2 Review of Microprocessor Based Implementation Schemes

Several types of implementation schemes are available in the literature for real-time implementation of PWM controllers. Such schemes use programmable timers and I/O devices. Real time PWM pattern can be generated using single timer, three timers or four timers.

1.2.1 Three-Timer Carrier Cycle PWM

In this scheme, the pulse width angles are calculated directly using the equation [3],

$$t_{pw} = \frac{T}{2}\{1 + M\sin(\omega_m T_1)\} \tag{1.30}$$

The 3-phase timer output these angles as shown in Fig 1.12. The use of this timer allows a more flexible and efficient program structure to be used. Thus, the computation time is minimized and the microprocessor is released to perform other operations.

The pulse-width angles are calculated and are then converted to a time period using a timer. Here an Intel 8253 timer has been used. This timer contains three 16-bit counters, each of which may be programmed to operate in one of six modes It is therefore possible to use this timer for 3-phase applications. To generate the width-modulated pulse in real time requires the timer counter to be loaded with a value from the microprocessor corresponding to the required pulsewidth. The rate of decrement of the counter is determined by its clock. When a zero value is reached, the counter output changes and generates an interrupt signal to inform the microprocessor that the time period has been completed. In order that the microprocessor can identify the source of the interrupt signal, and to avoid the ambiguity, which would arise if two signals were generated at the same time, interrupt priority logic circuitry is required.

The microprocessor responds by loading the next pulse width into the counter. Six signals are generated by the output port, corresponding to the six power devices in the inverter. Typical pulses for the three phases are illustrated in Fig. 1.13. The pulse widths for the three phases must be calculated in advance. At the start of the carrier period, times t_{a1}, t_{b1} and t_{c1} are loaded into the three counters. As each counter completes its period, the appropriate outputs are changed and the times corresponding to the pulses are loaded into the counters. In order to define the end of a

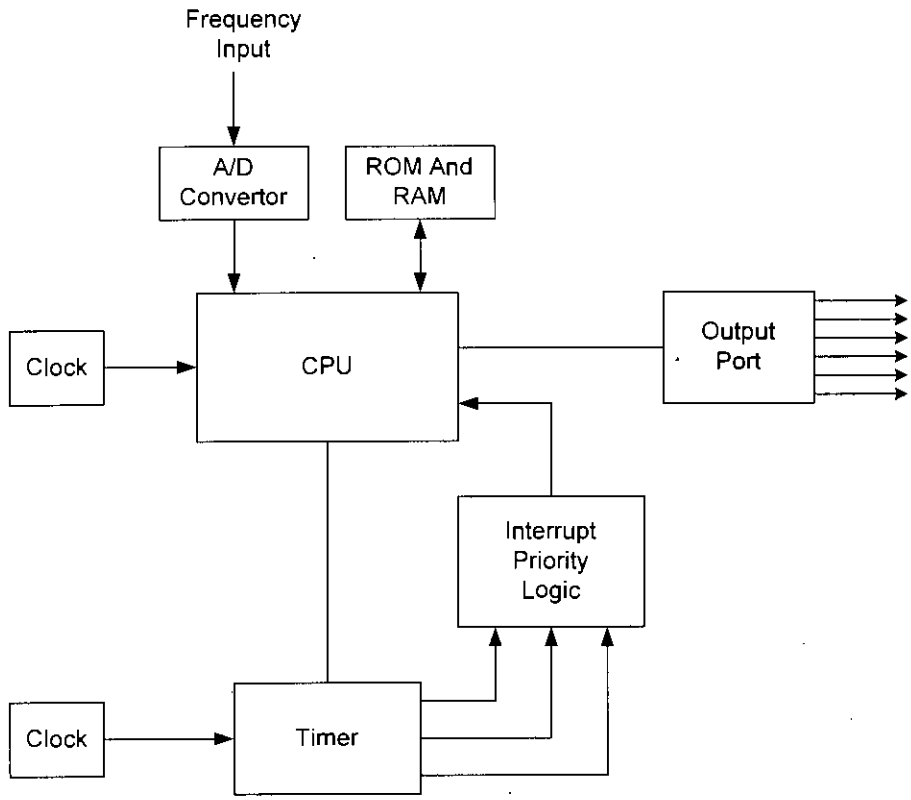


Fig. 1.12. Hardware for microprocessor-based regular sampled PWM

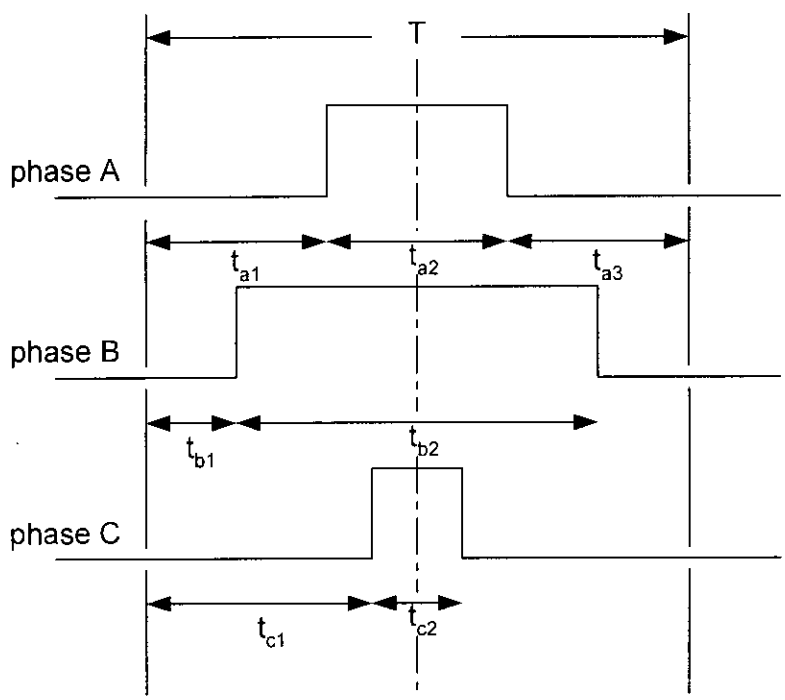


Fig. 1.13 Three-phase PWM waveform

carrier period, it is only necessary to load time t_{a3} into the first counter when the next interrupt is generated by that counter.

1.2.2 Four-Timer Carrier Cycle PWM

The process of PWM waveform generation using asymmetric regular sampling and a four-timer implementation is illustrated in Fig. 1.14. The pulse widths are calculated in the microprocessor using the following equations:[1]

$$\delta_k = T/4 - MT/4\sin(kT/2) \tag{1.31}$$

$$\delta_{k+1} = T/4 - MT/4\sin[(k+1)T/2] \tag{1.32}$$

The pulse widths are then converted into time intervals using four programmable timers, one for each phase, and the fourth one is used to generate interrupt signals to the microprocessor.

At the beginning of each carrier cycle the microprocessor loads the phase A, B and C timers with the prepulse times $\delta_A(k)$, $\delta_B(k)$ and $\delta_C(k)$ respectively, and the fourth with T/2. When the timers for each phase finish timing, their corresponding outputs clock three J-K flip-flops, which generate the PWM inverter gating signals. When the fourth timer completes timing out T/2, the microprocessor is interrupted and the three-phase timers are loaded with the post-pulse times $\delta_A(k+1)$, $\delta_B(k+1)$ and $\delta_C(k+1)$ and the fourth-timer times out T/2 to complete the carrier cycle. This process is repeated for the following carrier-cycles to generate the complete PWM wavelength.

1.2.3 Single-Timer Carrier Cycle PWM

The four-timer microprocessor implementation is relatively simple due to the extensive use of timers. If a single timer is used [1], then it is necessary to link the timer vector to the three phases that are switching. The single-timer version reduces the hardware requirements at the expense of increased calculation; although adopting some techniques these calculations can be significantly reduced.

Considering three phases together, two vectors are calculated as shown in Fig. 1.15. A timing vector holds the values t_0, t_A, t_B and t_7 and a switching vector defines the PWM output during

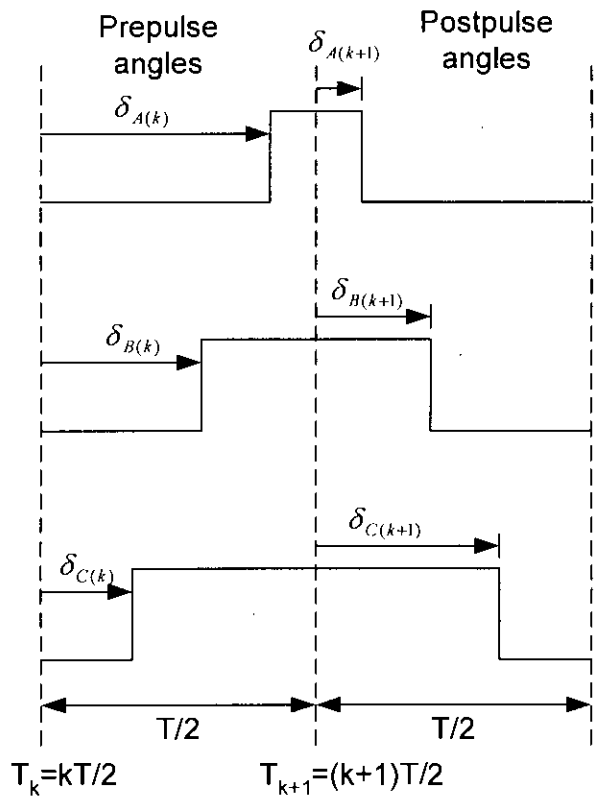


Fig. 1.14. Four-Timer PWM implementation

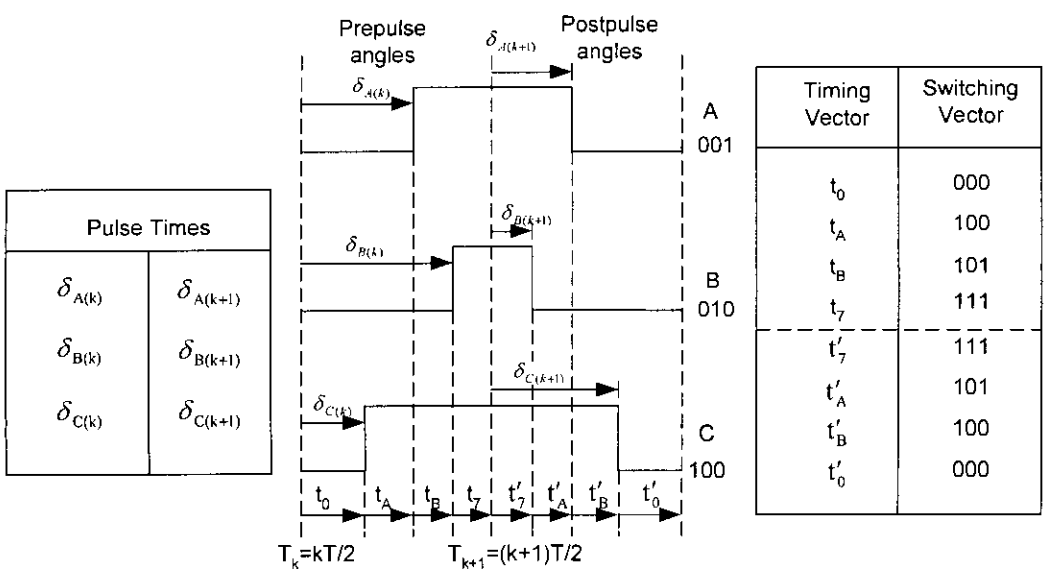


Fig. 1.15. Three-phase asymmetrical PWM single timer principles

these periods. The switching times at the beginning and end of the half carrier cycle are defined by times "t_0" and "t_7" respectively. "t_0" refers to a period when all three inverter phases are low (vector 000 = 0) and t_7 corresponds to all three inverter phases being high (111 = 7). These "null vectors" represent nonswitching or an inactive inverter state. The time period "t_A" and "t_B" correspond to one of the six possible inverter active states.

The output of the PWM gating signals to the inverter is via a memory-port mapped peripheral interface adapter (PIA), connected to the inverter gating circuitry [7].

The microprocessor implementation procedure loads the timer with a particular timing period and the PIA with the PWM output for that period. The timer interrupts the processor at the end of the period and the interrupt service routine loads the next time and PWM output to the respective devices.

1.3 Objectives of the Research Work

Different Pulse Width Modulation (PWM) techniques are used for the control of modern static power inverters. There are also different techniques for the implementation of these schemes. The PWM schemes are usually implemented with a microprocessor or a digital signal processor offering flexible control facilities with software without the need of changing the hardware. Of the PWM schemes, only the Harmonic Elimination PWM is implemented using Look Up tables (LUT). LUT based schemes offer fast response in on-line variable speed drive applications since no switching point calculations are involved in real time. Nevertheless, it was discarded for industrial use because of the huge memory involvement and cost. But this trend is changing because of the availability of low priced large memory modules and high-speed processors. Other PWM schemes generate PWM pattern using Microprocessor, I/O devices and Programmable Timers. They do not use LUT. Such schemes require high-speed processors for generating PWM patterns for wide operating range. For ac drives, the processor needs computation of other characteristic parameters and gets little time for the PWM pattern computation and hence limits the operating range.

The objective of this research is to analyze the PWM waveforms for three-phase inverter optimizing the pattern for minimum total harmonic distortion (THD) and to propose a LUT

based hybrid PWM scheme where both processor and LUT would be involved. The total operating range of the PWM scheme will be increased and less memory will be required since the same duty cycle can be shared for different frequencies and different modulation indices. The microprocessor will be used for computing duty cycles for the PWM pattern once in each carrier period. Thus it would get enough time upto the next carrier cycle for computing other parameters of the drive system. The scheme will be implemented in real time using microprocessor. The proposed scheme is expected to offer better performance than the existing techniques.

1.4 Organization of the Thesis

Chapter 2 describes the methods of determining switching points of PWM patterns. The equation of pulse width is developed.

Chapter 3 describes the process of obtaining the simplified equation of pulse width of PWM patterns so that the real time implementation of the scheme can be done using a normal microprocessor, which can perform only integer type calculations.

Chapter 4 describes the proposed LUT based PWM scheme with its block diagram. It also describes the formation of Look Up Table along with the algorithm of the MATLAB program used for this purpose. The simulated PWM patterns and corresponding frequency spectrums for different modulating frequencies and modulation indices are shown graphically and analyzed.

Chapter 5 describes the detailed circuit diagram and the process of real time implementation of the proposed scheme. The experimental PWM patterns and corresponding frequency spectrums for different modulating frequencies and modulation indices are shown graphically and analyzed. These patterns and spectrums are compared with the simulated patterns and spectrums.

Finally, chapter 6 concludes the thesis with some recommendations for future research.

Chapter 2

Analysis of Regular Sampled PWM

2.1 Introduction

The quality of a PWM pattern depends on its determination process. Two level regular sampled PWM is analyzed in this chapter and a simplified method is devised to reduce the computation time. It is observed that the simplification of the PWM process reduces the computation time of the switching points. For getting PWM voltage for the control of inverter, it is necessary to sample a modulating sinusoidal voltage signal at regular intervals.

2.2 Mathematical Modeling of the Modulation Process

It is necessary to apply sinusoidally weighted signal to the inverter for getting sinusoidal output voltage. Sinusoidally weighted signal can be obtained by sampling a sinusoidal modulating signal at regular intervals. However, variable voltage cannot be applied to the inverter. For this reason, PWM pulses are generated and applied to the inverter. The pulse widths of the PWM pulses can be calculated through different methods.. The switching edges of the pulses can be calculated after calculating the pulse widths. Two methods of pulse width calculation are discussed in the following two sections.

2.2.1 Determination of PWM Pulse Width: Method 1

For getting sinusoidally modulated pulses for the control of static power inverter, sampling of a sinusoidal modulating signal is done at regular intervals. If infinite number of sampling is done then the modulating signal would be very close to actual sine curve. However, real time implementation of PWM scheme with infinite number of sampling is not possible. Therefore, it is necessary to take finite number of sampling within a carrier period. The modulating signal voltage remains constant from the midpoint of an interval to the midpoint of next interval and is equal to the voltage at the sampling instant. Thus, the modulating signal is a stepped sinusoidal wave. If the modulating signal were pure sinusoidal wave, there would be very negligible harmonics in the spectrum of PWM pulses, which is a desirable feature. Since the modulating

signal is a stepped one, there will exist considerable harmonics in the spectrum of PWM pulses. PWM pulses are to be generated such that pulse widths vary in accordance with the height of the modulating signal at the sampling instant. The average value of the PWM pulse over a complete cycle varies in accordance with the width of the pulse. Hence, the pulse width is calculated considering that the average value of the pulse over a complete cycle is equal to the height of the modulating signal at the sampling instant. In Fig. 2.1 sampling of a sinusoidal modulating signal at regular intervals and the corresponding PWM pulse is shown. The value of PWM voltage varies between +V and –V.

In Fig. 2.1, Height of the modulating signal at n^{th} sampling instant,

$$V_h = V_m \sin(\omega\, nT_c) \tag{2.1}$$

Average value of PWM pulse over one complete cycle,

$$V_{avg} = \frac{t_1 * V - t_2 * V}{T_c} \tag{2.2}$$

where,

$v = V_m \sin(\omega t)$ = Modulating signal

nT_c= sampling instant (n=1,2,3.....)

t_1 = duration of voltage +V in a carrier cycle for n^{th} sampling

t_2 = duration of voltage -V in a carrier cycle for n^{th} sampling

T_c = sampling interval=$t_1 + t_2$

Considering that the average value of PWM pulse over one complete cycle is equal to the height of the modulating signal at a particular sampling instant, from (2.1) and (2.2)

$$V_h = V_{avg}$$

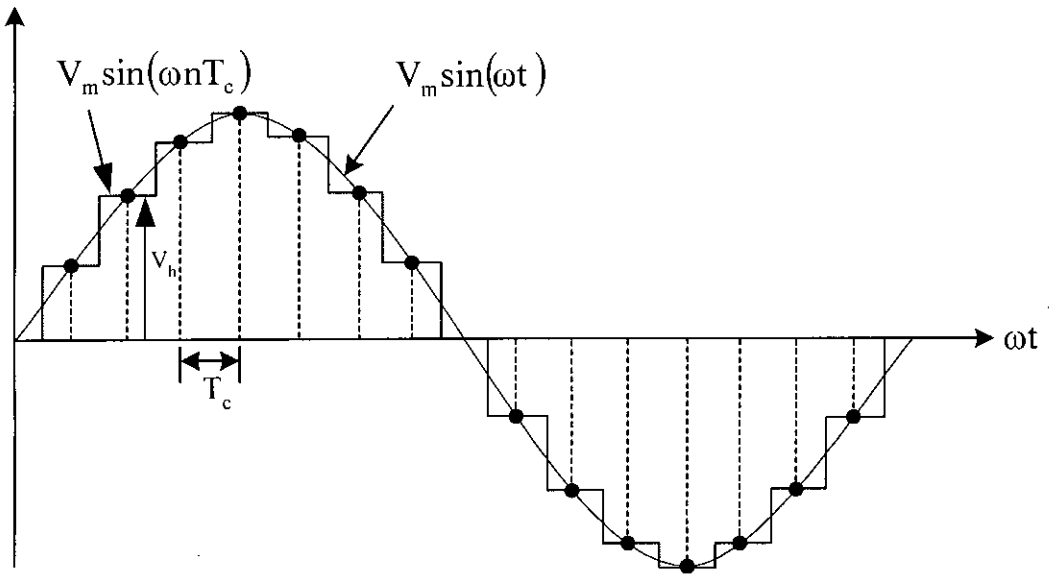
$$V_m \sin(\omega n T_c) = \frac{t_1 * V - t_2 * V}{T_c} \tag{2.3}$$

$$\text{or, } V_m \sin(\omega n T_c) = \frac{V\{t_1 - (T_c - t_1)\}}{T_c} \tag{2.4}$$

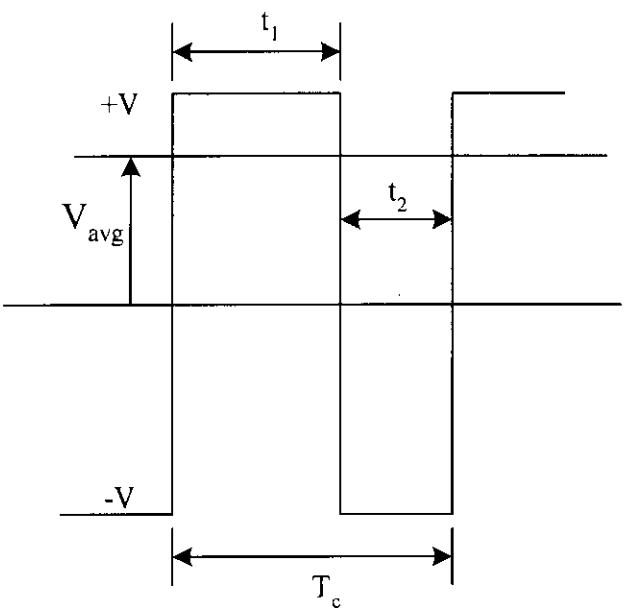
$$\text{or, } \frac{2t_1}{T_c} - 1 = \frac{V_m}{V} \sin(\omega n T_c) \tag{2.5}$$

$$\text{or, } \frac{2t_1}{T_c} = 1 + M \sin(\omega n T_c) \tag{2.6}$$

$$\text{or, } t_1 = \frac{T_c}{2}\left[1 + M \sin(\omega n T_c)\right] \tag{2.7}$$



(a)



(b)

Fig. 2.1 Sampling a sinusoidal modulating signal

(a) Stepped sinusoidal modulating signal (b)PWM pulse

Where, $\frac{V_m}{V}$ = M = Modulation Index

$\omega = 2\pi f$ = angular frequency of modulating signal

From (2.7) it is clear that width of a pulse varies in accordance with the sampling interval (T_c), modulation index (M) and the sampling instant (n).

2.2.2 Determination of PWM Pulse Width: Method 2

Switching points of the PWM pulses can be obtained from the modulation of a sinusoidal modulating signal with a high frequency triangular carrier signal. The modulating function is considered as a stepped sine wave having constant magnitude within a carrier period. As shown in Fig. 2.2, the sampling instant for the nth sample is $t_\delta + (n-1)T_c$, where t_δ is the delay time of

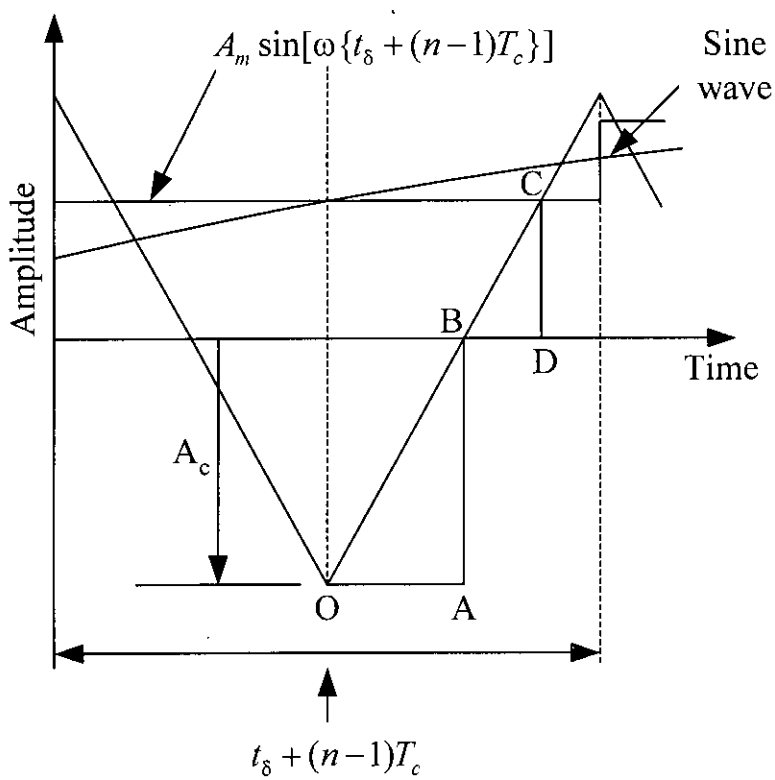


Fig. 2.2. Switching points calculation procedure from sampled waveform

the first sampling from the starting point of the modulating signal. At this instant, the modulating reference is $A_m \sin\omega\{t_\delta + (n-1)T_c\}$ and is represented by the length CD. From triangles BOA and CBD, ∠BOA = ∠CBD. Hence, the tangents at angle O and B are same. This means that

$$\frac{BA}{OA} = \frac{CD}{BD} \tag{2.8}$$

$$BD = OA \times \frac{CD}{BA} \tag{2.9}$$

$$BD = \frac{T_c}{4} \times \frac{A_m \sin \omega\{t_\delta + (n-1)T_c\}}{A_c} \tag{2.10}$$

The total pulse width at the n[th] sample is $2 \times (OA + BD)$ and is given by

$$t_w = 2 \times \left[\frac{T_c}{4} + \frac{T_c}{4} \times \frac{A_m \sin \omega\{t_\delta + (n-1)T_c\}}{A_c} \right]$$

$$\text{or, } t_w = \frac{T_c}{2} \left[1 + \frac{A_m}{A_c} \sin \omega\{t_\delta + (n-1)T_c\} \right] \tag{2.11}$$

$$\text{or, } t_w = 0.5T_c \left[1 + M \sin \omega\{t_\delta + (n-1)T_c\} \right] \tag{2.12}$$

where, $M = \frac{A_m}{A_c}$, is called the modulation index. In (2.7) M was defined as $M = \frac{V_m}{V}$. So it can be written that

$$M = \frac{A_m}{A_c} = \frac{V_m}{V} \tag{2.13}$$

For three phase applications, (2.12) represents the pulse width for phase A. For B and C phases, phase angles of $2\pi/3$ and $4\pi/3$ should be added to (2.12). Therefore, the expressions for the pulse widths for phases A, B and C take the form

$$t_{wa} = \frac{T_c}{2} \left[1 + M \sin \omega\{t_\delta + (n-1)T_c\} \right] \tag{2.14}$$

$$t_{wb} = \frac{T_c}{2} \left[1 + M \sin \omega \left\{ t_\delta + (n-1)T_c - \frac{2\pi}{3} \right\} \right] \tag{2.15}$$

$$t_{wc} = \frac{T_c}{2} \left[1 + M \sin \omega \left\{ t_\delta + (n-1)T_c - \frac{4\pi}{3} \right\} \right] \tag{2.16}$$

Replacing the angular frequency ω by $2\pi f$, the generalized form of the equation of pulse width for any phase would take the form

$$t_{wi}(M,f,n) = 0.5T_c\left[1 + M\sin 2\pi f\left\{t_\delta + (n-1)T_c - (i-1)\frac{2\pi}{3}\right\}\right] \quad (2.17)$$

where, i = 1, 2, 3 for phase A, phase B and phase C respectively. Equation (2.17) represents the width of n^{th} pulse for phase i at a given modulation index, M and modulating frequency, f. If the first sampling is done with no delay *i.e.* at the starting point of the modulating signal, then t_δ=0. From (2.17) we find that pulse width of PWM pulses is the function of sampling interval or carrier period (T_c), modulation index (M), modulating signal frequency (f) and the number of sampling (n). Sampling interval inversely varies with the carrier frequency. Hence, we can control the width of PWM pulses by varying the modulation index, the carrier frequency and the modulating signal frequency.

Chapter 3

Digital Representation of Sampled PWM

3.1 Introduction

In the generalized equation (2.17) derived in chapter 2 for the pulse widths of PWM pulses, some parameters are floating point variables. These parameters are T_c, M and sine function. Normal microprocessors do not possess built-in math coprocessor and cannot perform floating-point calculations. These processors are designed to perform integer type multiplications, divisions, additions, subtractions etc. MMX processors have built-in math coprocessor, which can perform floating-point calculations. However, it will not be cost effective to use MMX processor for the real time implementation of PWM scheme. For this reason, the simplified version of equation (2.17) is used where the data i.e. carrier period (T_c), modulation indices (M) and sine values are of integer type so that the scheme can be implemented using low cost processor.

3.2 Sampled Version of the Model for PWM Pattern Calculation

Recalling the expression for pulse width of PWM pulse (2.12) in time domain derived in chapter2

$$t_w = 0.5T_c\left[1 + M\sin\omega\{t_\delta + (n-1)T_c\}\right] \tag{3.1}$$

If the first sampling is done at the starting point of the positive half cycle of the sinusoidal modulating signal, $t_\delta = 0$. The equation of pulse width then takes the form

$$t_w = 0.5T_c\left[1 + M\sin\omega(n-1)T_c\right] \tag{3.2}$$

Let, N = number of carrier cycles in a fundamental period. We can write

$$\omega N T_c = 2\pi \tag{3.3}$$

$$\text{or, } \omega T_c = \frac{2\pi}{N} \tag{3.4}$$

Hence, the sampling is done at an interval $\omega T_c = \frac{2\pi}{N}$ radian. From equation (3.2) we get

$$t_w = 0.5T_c\left[1 + M\sin(n-1)\frac{2\pi}{N}\right] \tag{3.5}$$

Now, let the total number of samples in a carrier period = K_c. Therefore, in sample domain K_c represents the period of a carrier cycle. The value of K_c is taken as even number. Again, the modulation index is also converted to sampled form and normalized on a scale of K_c. So, if the value of K_c is 256, then the value of modulation tag of 256 represents 100% modulation, 128 represents 50% modulation and 0 represents 0% modulation. Again if the modulation index is normalized on a scale of $K_c/2$ then the value of modulation tag of 128 represents 100% modulation, 64 represents 50% modulation and 0 represents 0% modulation. In the present scheme, the value of K_c is taken as256 and M is normalized on a scale of $K_c/2$ i.e. 128.

Therefore, pulse width of the PWM pulse in sample domain is given by,

$$K_w = \frac{K_c}{2}\left[1 + \frac{M_s}{K_c/2}\sin(n-1)\frac{2\pi}{N}\right], \tag{3.6}$$

$$\text{or, } K_w = \frac{K_c}{2} + \frac{M_s}{K_c/2}\frac{K_c}{2}\sin(n-1)\frac{2\pi}{N}. \tag{3.7}$$

For three phase applications, (3.7) represents the pulse width for phase A. For B and C phases, phase angles of $2\pi/3$ and $4\pi/3$ should be incorporated in (3.7). Therefore, the expressions for the pulse widths for phases A, B and C take the form

$$K_{wa} = \frac{K_c}{2} + \frac{M_s}{K_c/2}\frac{K_c}{2}\sin(n-1)\frac{2\pi}{N} \tag{3.8}$$

$$K_{wb} = \frac{K_c}{2} + \frac{M_s}{K_c/2}\frac{K_c}{2}\sin\left[(n-1)\frac{2\pi}{N} - \frac{2\pi}{3}\right] \tag{3.9}$$

$$K_{wc} = \frac{K_c}{2} + \frac{M_s}{K_c/2}\frac{K_c}{2}\sin\left[(n-1)\frac{2\pi}{N} - \frac{4\pi}{3}\right] \tag{3.10}$$

Considering the three phases together, the generalized form of the equation of pulse width for any phase would take the form,

$$K_{wi} = \frac{K_c}{2} + \frac{M_s}{K_c/2}\frac{K_c}{2}\sin\left[(n-1)\frac{2\pi}{N} - (i-1)\frac{2\pi}{3}\right] \tag{3.11}$$

where, i = 1, 2, 3 for phase A, Phase B and phase C respectively.

From (3.11) we can write

$$K_{wi} = K_c' + \frac{M_s}{K_c'}K_{si}. \tag{3.12}$$

Where, M_s = Modulation tag

$$K_c' = \frac{K_c}{2}$$

$$K_{si} = \frac{K_c}{2}\sin\left[(n-1)\frac{2\pi}{N} - (i-1)\frac{2\pi}{3}\right] \tag{3.13}$$

If the total number of patterns is K_c' then total number of duty cycles will be K_c'. In that case, the duty cycles of the patterns *Da*, *Db* and *Dc* for phases A, B and C respectively can be calculated as follows:

$$Da = \left(K_c' + \frac{M_s}{K_c'}.K_{sa}\right)/2 \tag{3.14}$$

$$Db = \left(K_c' + \frac{M_s}{K_c'}.K_{sb}\right)/2 \tag{3.15}$$

$$Dc = \left(K_c' + \frac{M_s}{K_c'}.K_{sc}\right)/2 \tag{3.16}$$

where, $$K_{sa} = \frac{K_c}{2}\sin(n-1)\frac{2\pi}{N} \tag{3.17}$$

$$K_{sb} = \frac{K_c}{2}\sin\left[(n-1)\frac{2\pi}{N} - \frac{2\pi}{3}\right] \tag{3.18}$$

and $$K_{sc} = \frac{K_c}{2}\sin\left[(n-1)\frac{2\pi}{N} - \frac{4\pi}{3}\right] \tag{3.19}$$

In the equations (3.14), (3.15) and (3.16), the modulation tag M_s is an integer variable. Again, since K_c is taken as an even number K_c' is an integer variable. The value of K_{sa}, K_{sb} and K_{sc} are generated through a MATLAB program using (3.13) and stored after round off. Therefore, K_{sa}

K_{sb} and K_{sc} are also integer variables. Hence, we see that all the parameters in (3.14), (3.15) and (3.16) are integer variables. Using these integer type data we can implement the PWM scheme using a low cost processor, which can perform integer type calculations only.

3.3 Formation of PWM Pulses in Digital Form

PWM pulses for three phases within one carrier cycle are shown in Fig. 3.1. For the formation of the pulses for phase A in digital form we can take small time interval Δt such that $K_c \Delta t = T_c$, where K_c is the number of samples taken over T_c. To get the pattern, the samples up to ($T_c/2$ - $t_1/2$) should be low and the rest of the samples up to $T_c/2$ should be high. Again, from $T_c/2$ to ($T_c/2 + t_1/2$) the samples should be high and the rest of the samples upto T_c should be low. In this way the PWM pattern can be formed. The patterns for other phases can be obtained in the same way.

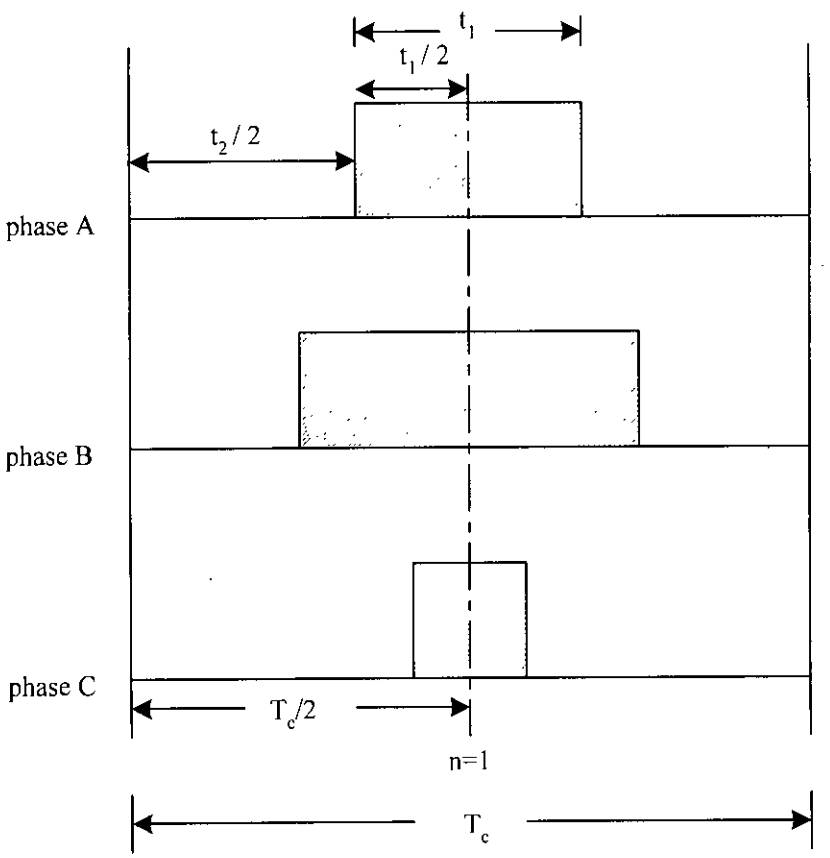


Fig. 3.1 PWM pulse for one carrier cycle for phase A, phase B and phase C

Algorithm for the formation of PWM pulses

The PWM patterns were generated using MATLAB program [Appendix: Program1]. The following algorithm was used in the program for this purpose:

1. Initialize modulation index M, modulating frequency f, carrier frequency f_c, total number of samples in a carrier period N_{max}, and total number of patterns q_{max}.
2. Initialize number of pattern, $q = 1$
3. If $q > q_{max}$ go to step 11, else go to step 4.
4. Initialize number of sample, $n = 1$
5. If $n > N_{max}$ go to step 10 , else go to step 6.
6. If $n < (N_{max}/2 - q + 1)$ or $n > (N_{max}/2 + q - 1)$ go to step 7, else go to step 8.
7. Store -1 in Pat(q,n), go to step 9 .
8. Store 1 in Pat(q,n), go to step 9
9. Increment n, go to step 5.
10. Increment q, go to step 3.
11. Compute number of pulses in a fundamental period, $N = \text{round}\left(\frac{f_c}{f}\right)$.
12. Initialize switching angle, $\theta_n = \frac{2\pi}{N}$.
13. If $\theta_n > 2\pi$ go to step 18, else go to step 14.
14. Compute the sine values by

$$Sna = \text{round}\left[\frac{N_{\max}}{2} \times \sin\theta_n\right] \tag{3.20}$$

$$Snb = \text{round}\left[\frac{N_{\max}}{2} \times \sin\left(\theta_n - \frac{2\pi}{3}\right)\right] \tag{3.21}$$

$$Snc = \text{round}\left[\frac{N_{\max}}{2} \times \sin\left(\theta_n - \frac{4\pi}{3}\right)\right] \tag{3.22}$$

15. Compute the duty cycles for three phases by

$$Da = \left(\frac{N_{\max}}{2} + Sna \times \frac{M_s}{\frac{N_{\max}}{2}}\right)/2 \tag{3.23}$$

$$Db = \left(\frac{N_{\max}}{2} + Snb \times \frac{M_s}{\frac{N_{\max}}{2}} \right) / 2 \tag{3.24}$$

$$Dc = \left(\frac{N_{\max}}{2} + Snc \times \frac{M_s}{\frac{N_{\max}}{2}} \right) / 2 \tag{3.25}$$

16. Store the patterns corresponding to the duty cycles *Da*, *Db* and *Dc* in PWMa, PWMb and PWMc.

17. Increment θ_n by $\frac{2\pi}{N}$, go to step 13.

18. Stop the program and return.

3.4 Summary

In this chapter, the sampled version of the model for PWM pattern calculation is developed and the PWM pulses are digitally formed. The algorithm for the formation of PWM pulses in digital form is presented. Since there are no floating-point type variables in the equation of the pulse width of PWM pulses it has become possible to implement the scheme in real time by using normal microprocessor that can perform only integer type calculations.

Chapter 4

Scheme for real time PWM waveform generation

4.1 Detailed Scheme

The basic functional blocks of the proposed scheme is shown in Fig. 4.1. In this scheme, three EPROMs are used each of 32k memory. In these EPROMs, PWM patterns for different duty cycles are stored as look up table. The processor calculates the duty cycles of the PWM pulses for the three phases for a specified modulation index and frequency once in each carrier period. The duty cycles are then normalized on a scale of 128. The processor then sends the three duty cycles (for three phases) to a Programmable Peripheral Interface (PPI) having three 8-bit ports (Each port of the PPI is connected to the respective EPROM of that phase). The counter used in the scheme is a 12-bit ripple binary counter. It is reset to the zero state by a logical "1" at the reset input independent of clock. In the proposed scheme, the counter is connected to the lower 8-bit address lines (A0-A7) of each EPROM and scans the 256 stored samples in the EPROM. The counter receives a logical "1" at the reset input after it reaches a count value of 255.The monostable timer is used to increase the width of the ripple carry (RC) pulse.

The frequency of the clock pulse generated by the clock pulse generator circuit is $f_{clock} = \frac{1}{3RC}$, where, R and C are the resistor and capacitor connected to the circuit externally.

4.2 Formation of the Look Up Table (LUT)

After deriving simplified form of the equation (3.12) for pulse width of PWM pulses, the PWM patterns are generated in digital form using MATLAB [Appendix:program1]. The patterns are then stored in three EPROMs as look up table. There are 15 bit address lines (A_0 - A_{14}) in the 32k EPROM (NM27C256). The upper seven bit lines (A_8-A_{14}) are used for pointing to the duty cycles and the lower eight bit lines (A_0-A_7) are used to scan the real time pattern corresponding to duties addressed by the upper lines.

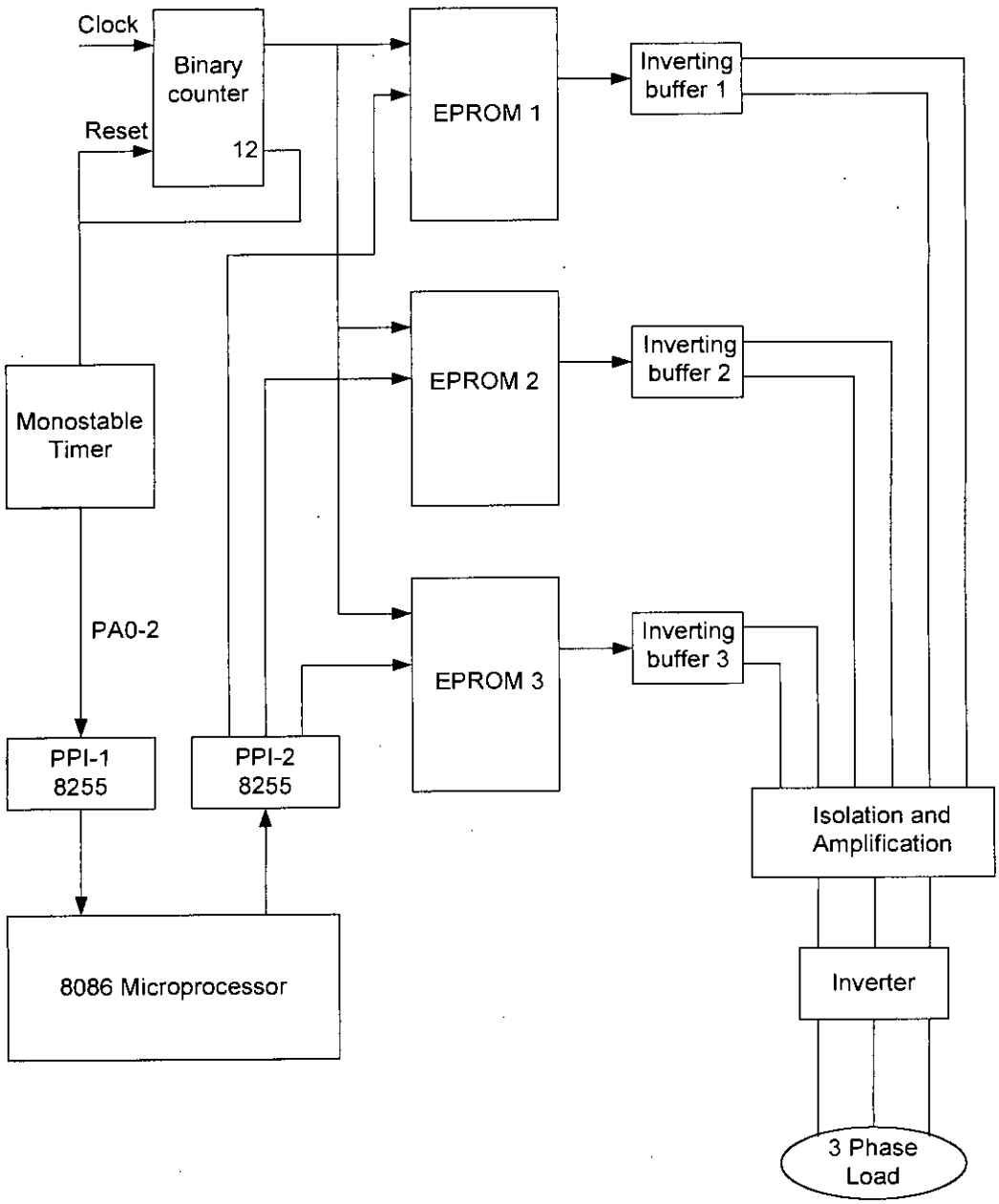


Fig. 4.1. Block schematic of the LUT based PWM scheme for three phase application

The following algorithm is used for formation of the Look Up table (LUT).

Algorithm for formation of LUT

1. Set total number of samples in a carrier period, N_{max} and total number of patterns, q_{max}.
2. Initialize number of pattern q (set $q = 1$).
3. If $q > q_{\max}$ go to step 11, else go to step 4.

4. Initialize number of sample n (set $n = 1$).
5. If $n > N_{max}$ go to step 10, else go to step 6.
6. If $n < (N_{max}/2 - q + 1)$ or $n > (N_{max}/2 + q - 1)$ go to step 7, else go to step 8.
7. Store 2 in Pat (q,n), go to step 9.
8. Store 1 in Pat (q,n), go to step 9
9. Increment n, go to step 5.
10. Increment q, go to step 3.
11. Stop the program and return.

4.3 Simulation of the Proposed PWM Scheme

The performance of the proposed PWM scheme is simulated for different frequencies and modulation indices that are shown in Fig.4.2-Fig.4.37. The carrier frequency (f_c) is kept constant at 1600 Hz. DC value, rms value of fundamental component and different harmonic components and Distortion Factor (DF) are obtained for different values of modulating frequencies (f) and modulation indices (M), which are listed in table I. The Distortion Factor (DF) is determined with the help of MATLAB by using the following equation:

$$DF = \frac{1}{V_1}\left[\sum_{n=2,3,4\ldots}^{\infty}\left(\frac{V_n}{n^2}\right)^2\right]^{\frac{1}{2}} \tag{4.1}$$

where, V_1 = rms value of fundamental component

and V_n = rms value of n^{th} harmonic component

From Table I we find the following features with some discrepancies:

- The dc values are negligible.
- The rms values of fundamental components and different harmonic components remain almost constant if f is decreased keeping M constant.
- The rms values of fundamental components and different harmonic components decrease if M is decreased keeping f constant.
- The Distortion Factor (DF) decreases if f is decreased keeping M constant.
- The Distortion Factor (DF) decreases if M is decreased keeping f constant.

Table I: List of values of dc component, fundamental and different harmonic components and Distortion Factors (DF) for different modulating frequencies and modulation indices

Modulation Index, M	Modulating Frequency, f (Hz)	DC value	Fundamental Component (rms)	2nd Harmonic Component (rms)	3rd Harmonic Component (rms)	Distortion Factor (DF)
0.94	100	9.84×10^{-4}	0.8008	0.0092	0.0013	0.0021
	90	0	0.7997	0.0056	8.056×10^{-17}	0.0015
	70	0.0014	0.8030	0.0024	0.0012	9.17×10^{-4}
	60	0	0.8032	0.0026	1.47×10^{-16}	7.12×10^{-4}
	50	4.92×10^{-4}	0.8049	0.0029	0.0013	5.97×10^{-4}
	40	7.87×10^{-4}	0.8051	7.88×10^{-4}	0.0022	3.4075×10^{-4}
0.6	100	0	0.5156	0.003	3.56×10^{-4}	0.0016
	90	0	0.5124	0.0023	5.17×10^{-17}	0.0012
	70	6.85×10^{-4}	0.5169	0.0028	0.0015	9.2277×10^{-4}
	60	0	0.5157	0.0029	9.44×10^{-17}	8.078×10^{-4}
	50	0	0.5179	7.496×10^{-4}	0.0011	4.299×10^{-4}
	40	0	0.5176	4.74×10^{-4}	0.0014	2.938×10^{-4}
0.4	100	0	0.3412	0.0013	0.0016	0.0015
	90	0	0.3443	0.001	3.51×10^{-17}	0.0012
	70	6.85×10^{-4}	0.3427	0.0022	0.0014	0.0012
	60	0	0.3421	0.0015	6.25×10^{-17}	7.3131×10^{-4}
	50	0	0.3440	3.26×10^{-4}	0.0011	4.2503×10^{-4}
	40	0	0.3429	2.10×10^{-4}	9.59×10^{-4}	3.7177×10^{-4}

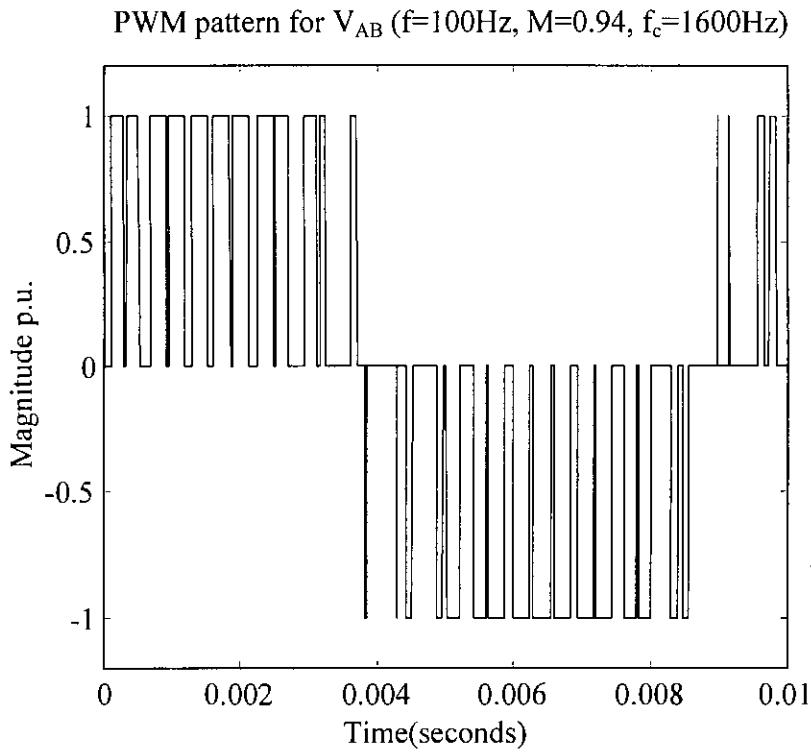


Fig. 4.2 Simulated PWM waveform pattern for V_{AB}
(f = 100 Hz, M = 0.94, f_c = 1600 Hz)

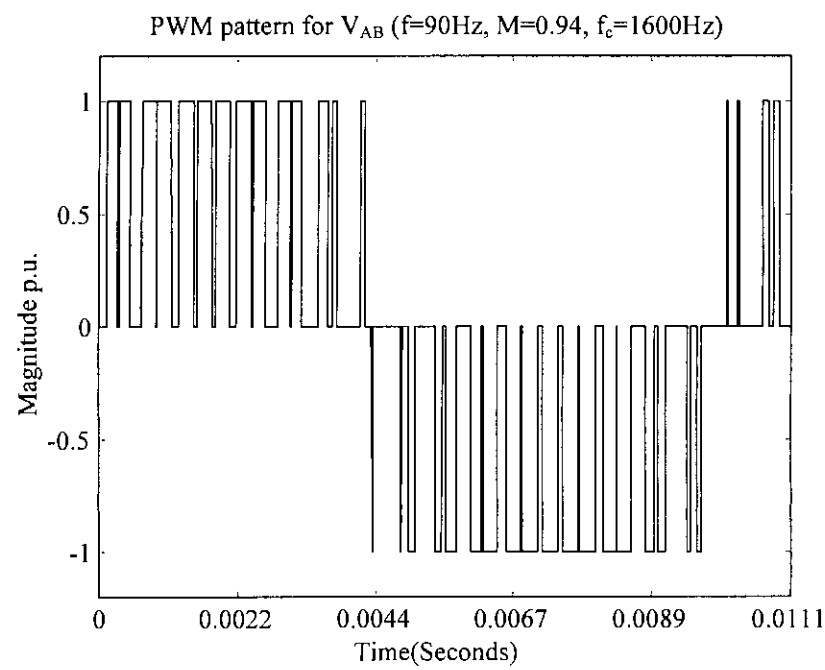


Fig. 4.3 Simulated PWM waveform pattern for V_{AB}
(f = 90 Hz, M = 0.94, f_c = 1600 Hz)

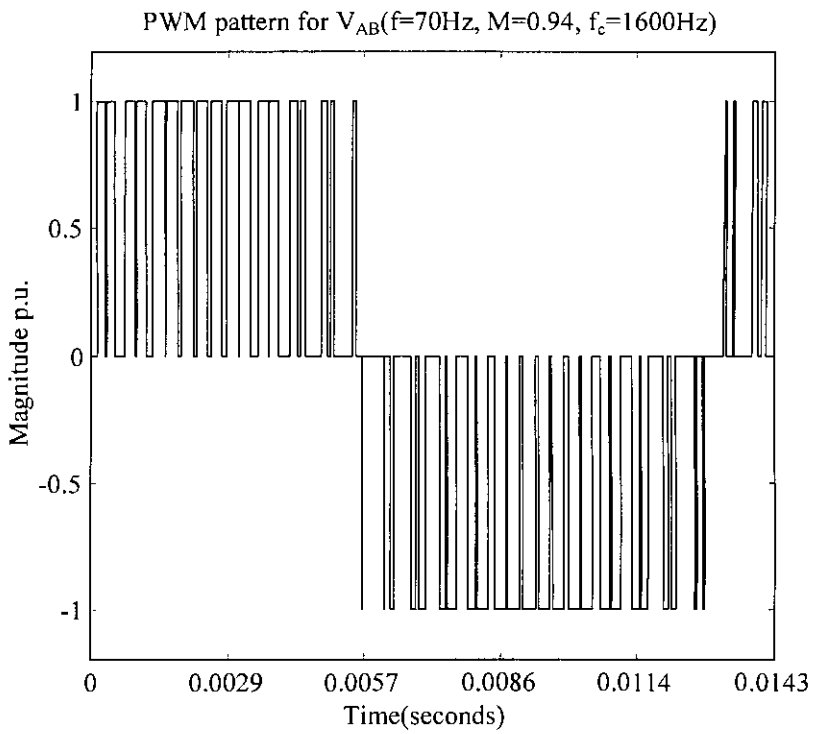


Fig. 4.4 Simulated PWM waveform pattern for V_{AB}
(f = 70 Hz, M = 0.94, f_c = 1600 Hz)

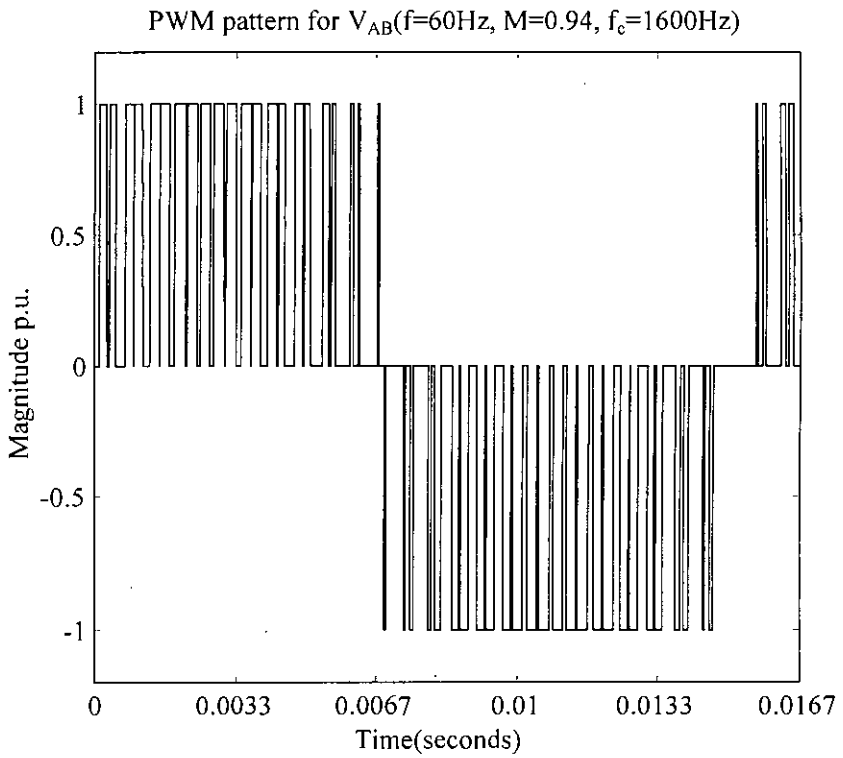


Fig. 4.5 Simulated PWM waveform pattern for V_{AB}
(f = 60 Hz, M = 0.94, f_c = 1600 Hz)

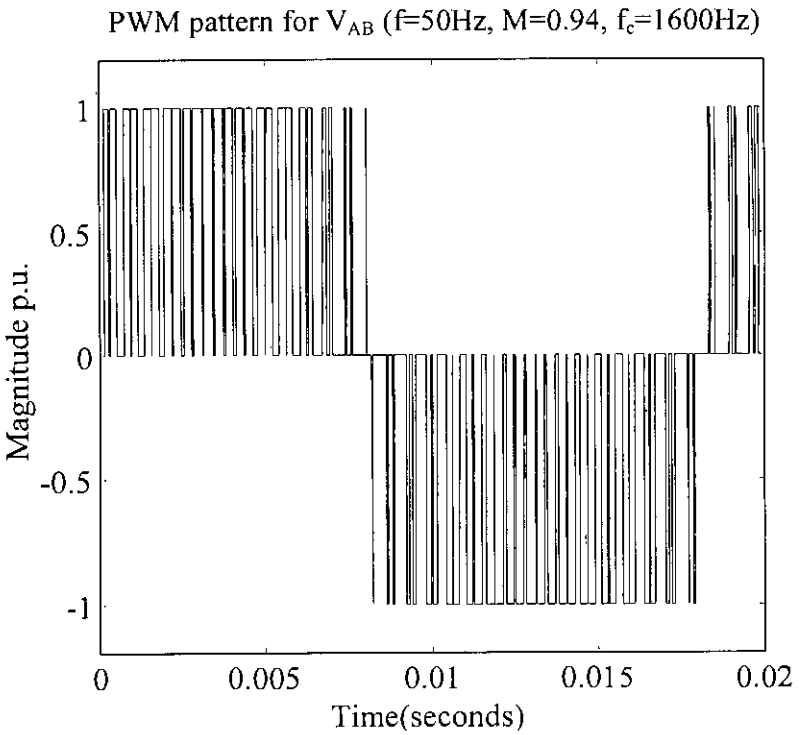


Fig. 4.6 Simulated PWM waveform pattern for V_{AB}

(f = 50 Hz, M = 0.94, f_c = 1600 Hz)

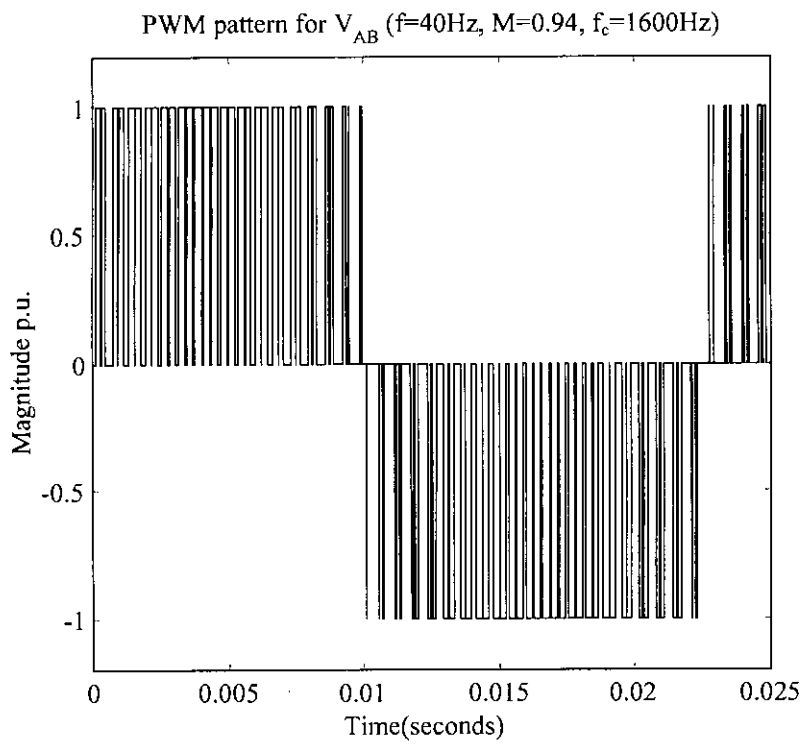


Fig. 4.7 Simulated PWM waveform pattern for V_{AB}

(f = 40 Hz, M = 0.94, f_c = 1600 Hz)

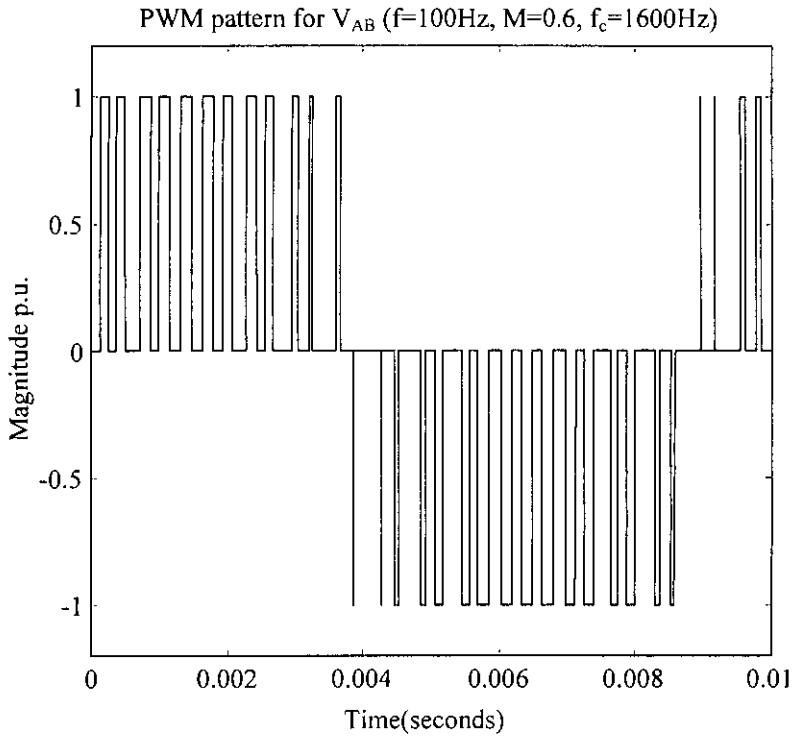


Fig. 4.8 Simulated PWM waveform pattern for V_{AB}
($f = 100$ Hz, $M = 0.6$, $f_c = 1600$ Hz)

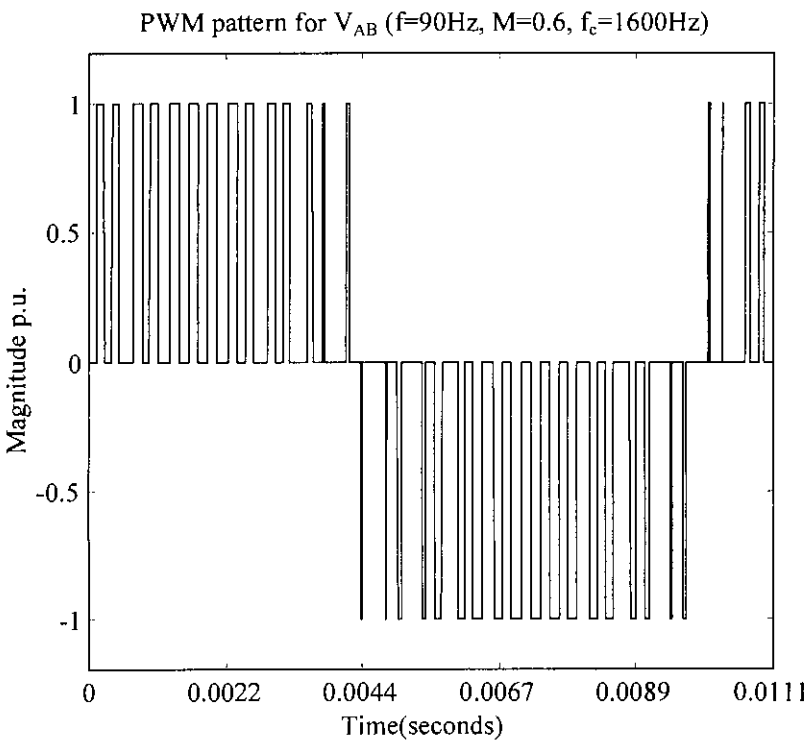


Fig. 4.9 Simulated PWM waveform pattern for V_{AB}
($f = 90$ Hz, $M = 0.6$, $f_c = 1600$ Hz)

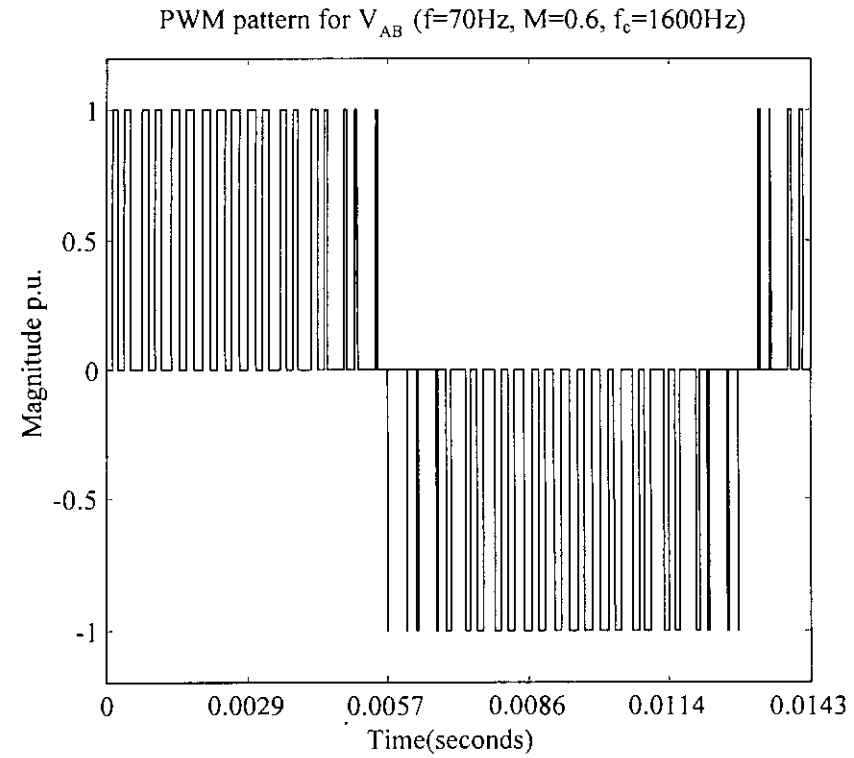


Fig. 4.10 Simulated PWM waveform pattern for V_{AB}
($f = 70$ Hz, $M = 0.6$, $f_c = 1600$ Hz

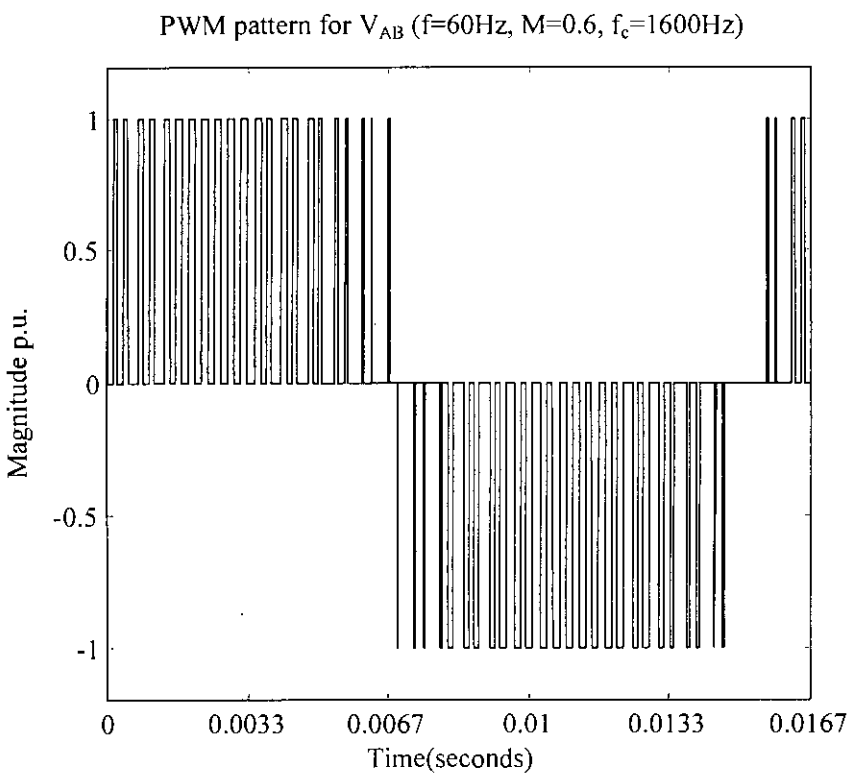


Fig. 4.11 Simulated PWM waveform pattern for V_{AB}
($f = 60$ Hz, $M = 0.6$, $f_c = 1600$ Hz

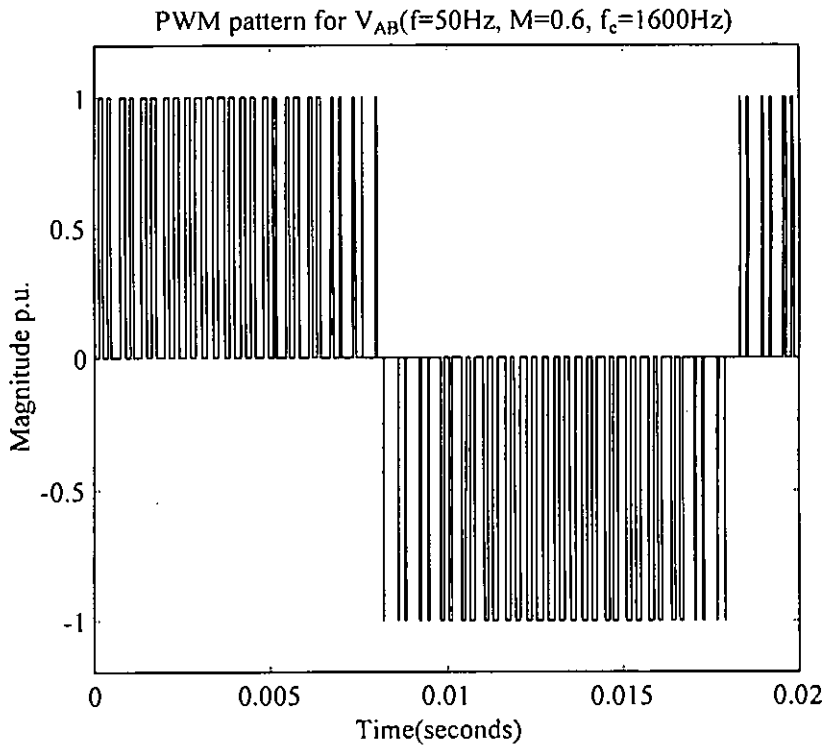


Fig. 4.12 Simulated PWM waveform pattern for V_{AB}
($f = 50$ Hz, $M = 0.6$, $f_c = 1600$ Hz

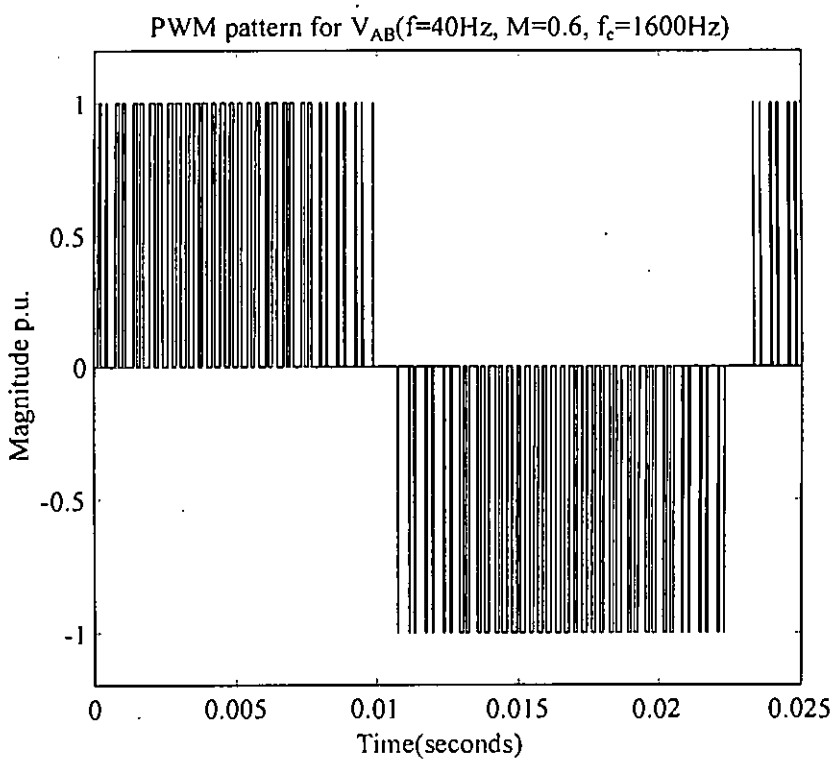


Fig. 4.13 Simulated PWM waveform pattern for V_{AB}
($f = 40$ Hz, $M = 0.6$, $f_c = 1600$ Hz

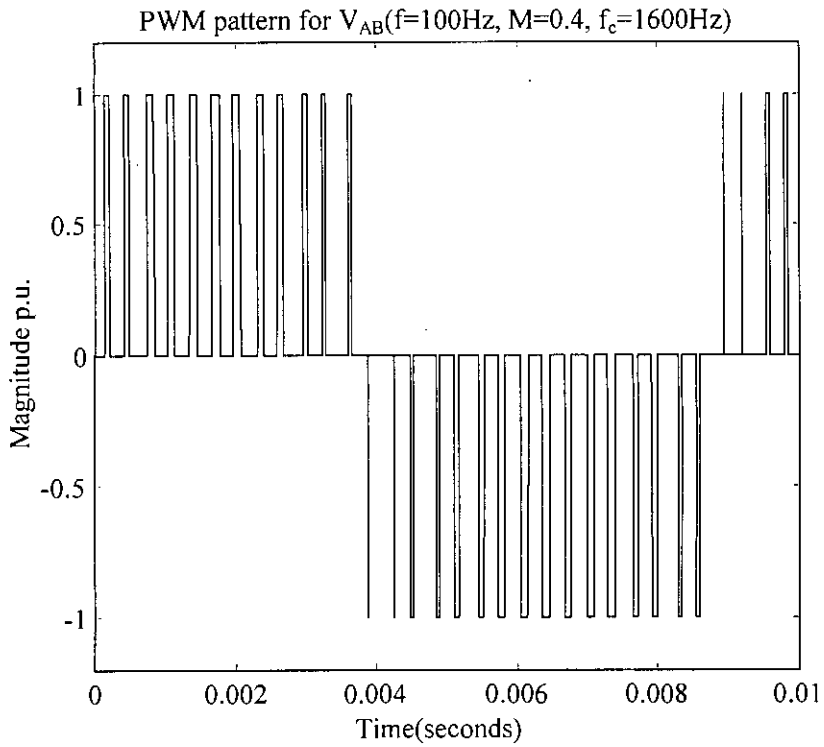


Fig. 4.14 Simulated PWM waveform pattern for V_{AB}
($f = 100$ Hz, $M = 0.4$, $f_c = 1600$ Hz

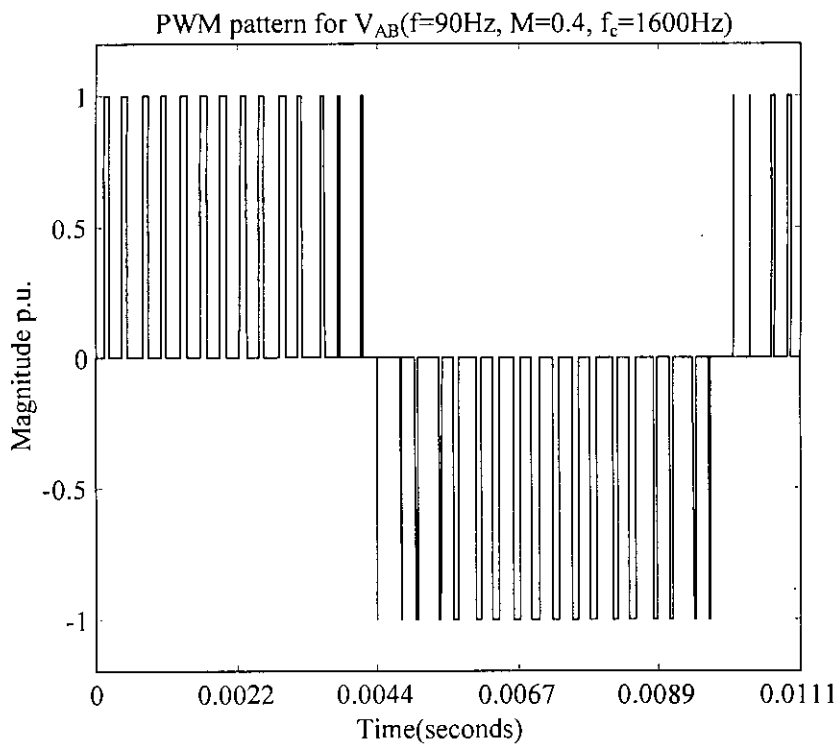


Fig. 4.15 Simulated PWM waveform pattern for V_{AB}
($f = 90$ Hz, $M = 0.4$, $f_c = 1600$ Hz

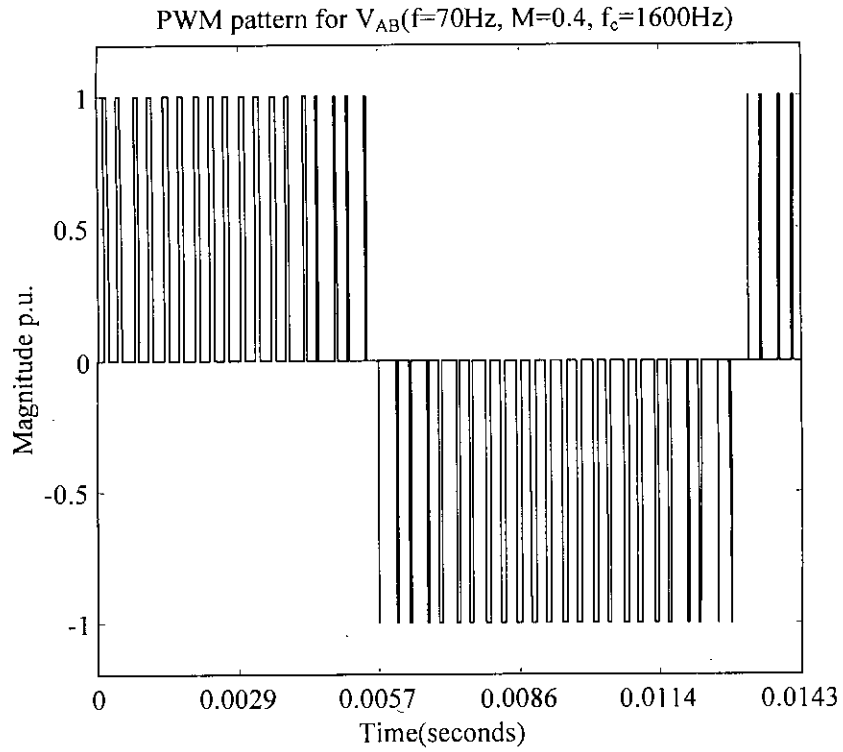


Fig. 4.16 Simulated PWM waveform pattern for V_{AB}

(f = 70 Hz, M = 0.4, f_c = 1600 Hz

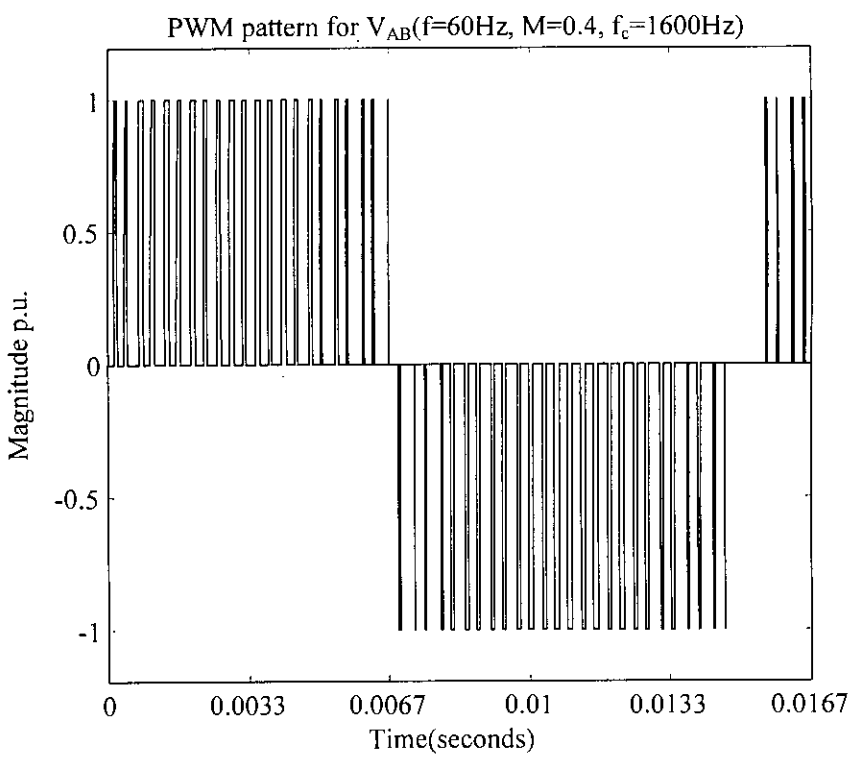


Fig. 4.17 Simulated PWM waveform pattern for V_{AB}

(f = 60 Hz, M = 0.4, f_c = 1600 Hz

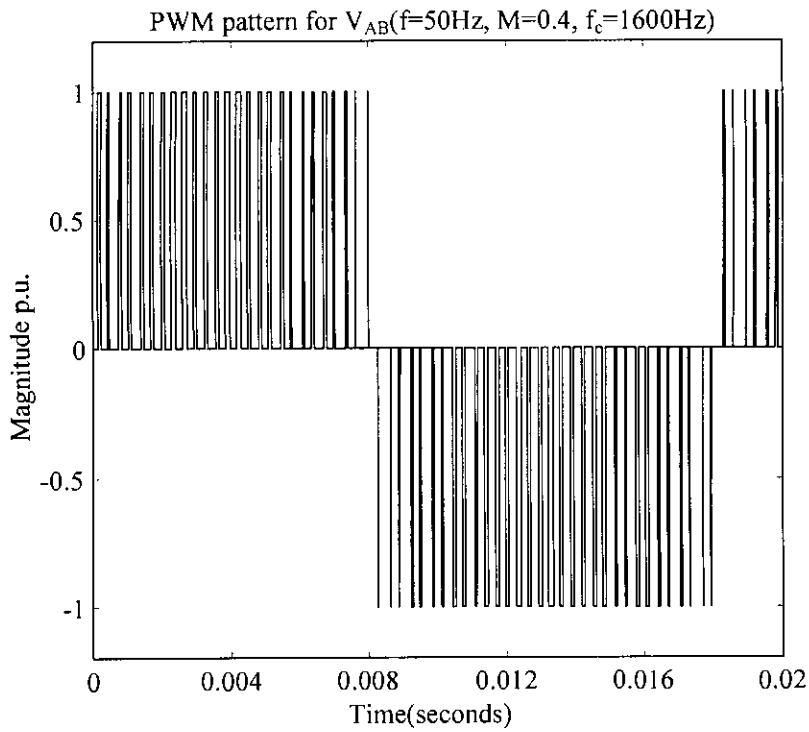


Fig. 4.18 Simulated PWM waveform pattern for V_{AB}

(f = 50 Hz, M = 0.4, f_c = 1600 Hz

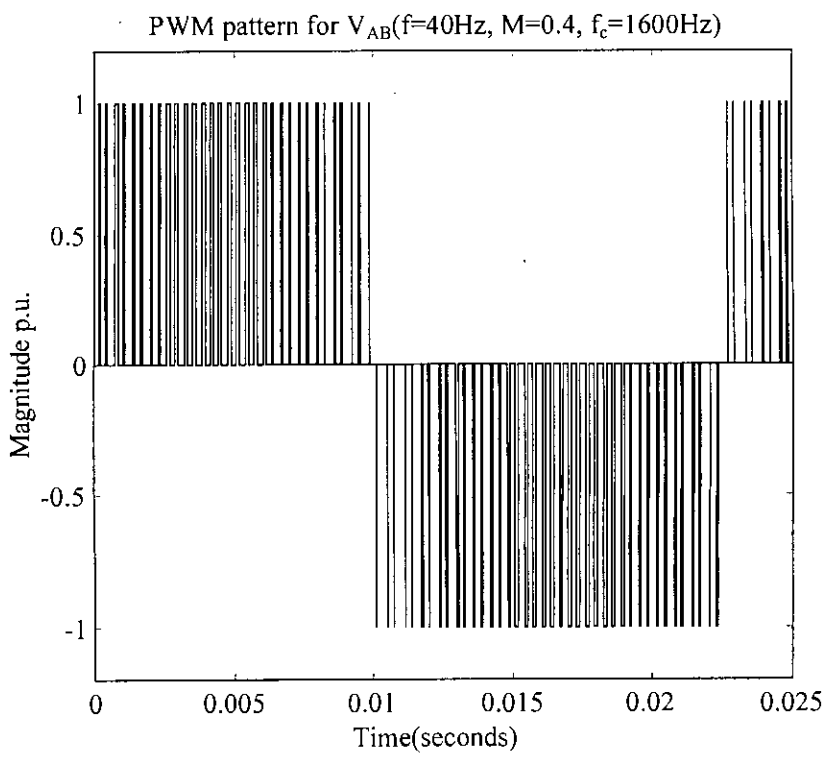


Fig. 4.19 Simulated PWM waveform pattern for V_{AB}

(f = 40 Hz, M = 0.4, f_c = 1600 Hz

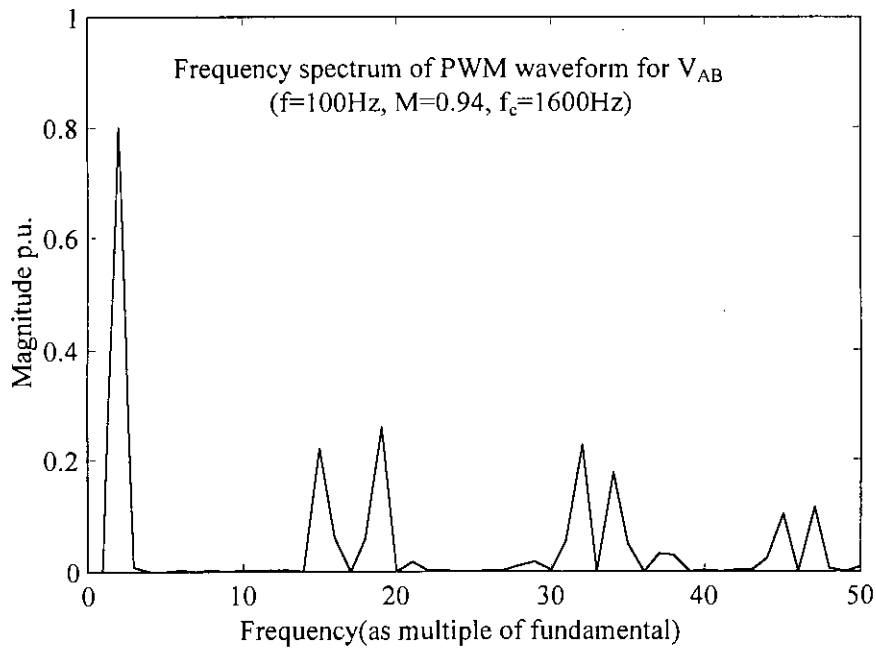


Fig. 4.20. Frequency spectrum of PWM waveform for V_{AB} (simulated)
(f = 100 Hz, M = 0.94, f_c = 1600 Hz)

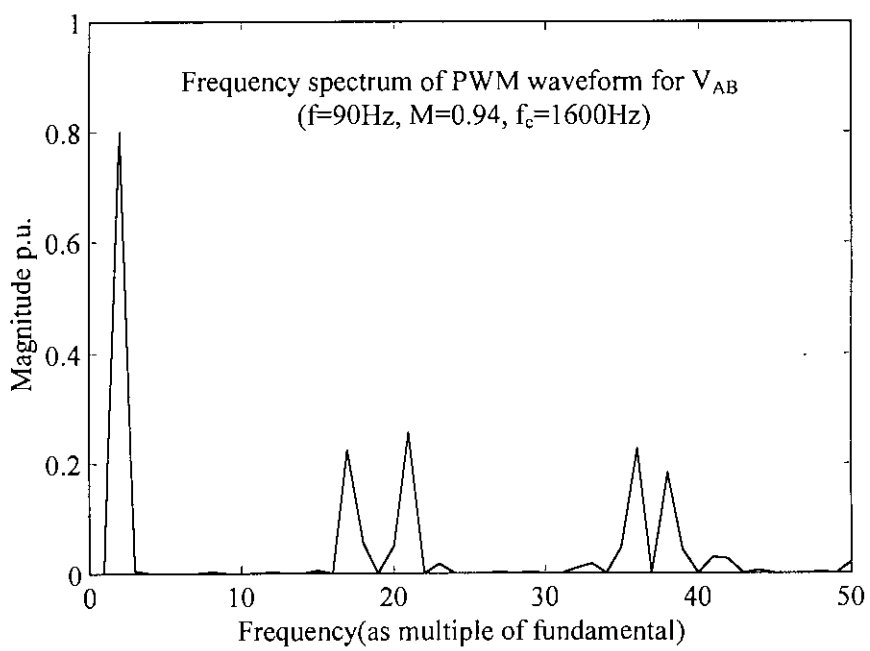


Fig. 4.21 Frequency spectrum of PWM waveform for V_{AB} (simulated)
(f = 90 Hz, M = 0.94, f_c = 1600 Hz)

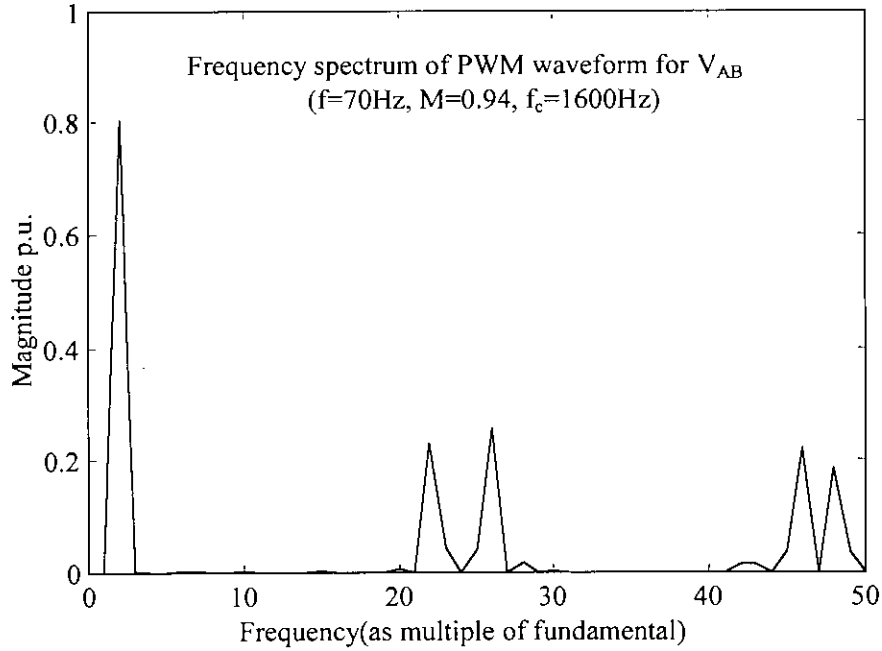


Fig. 4.22. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 70$ Hz, $M = 0.94$, $f_c = 1600$ Hz)

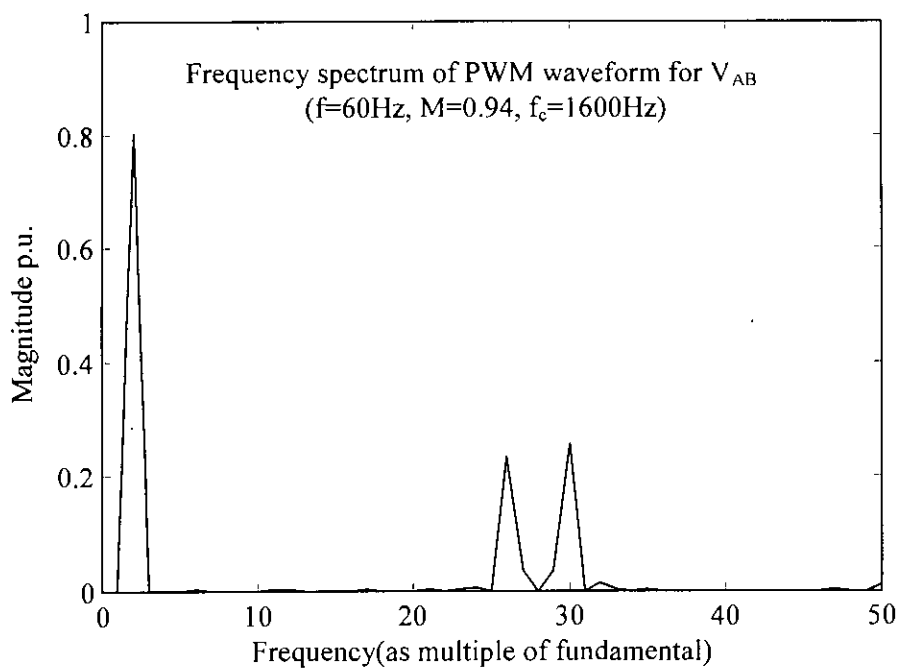


Fig. 4.23.Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 60$ Hz, $M = 0.94$, $f_c = 1600$ Hz)

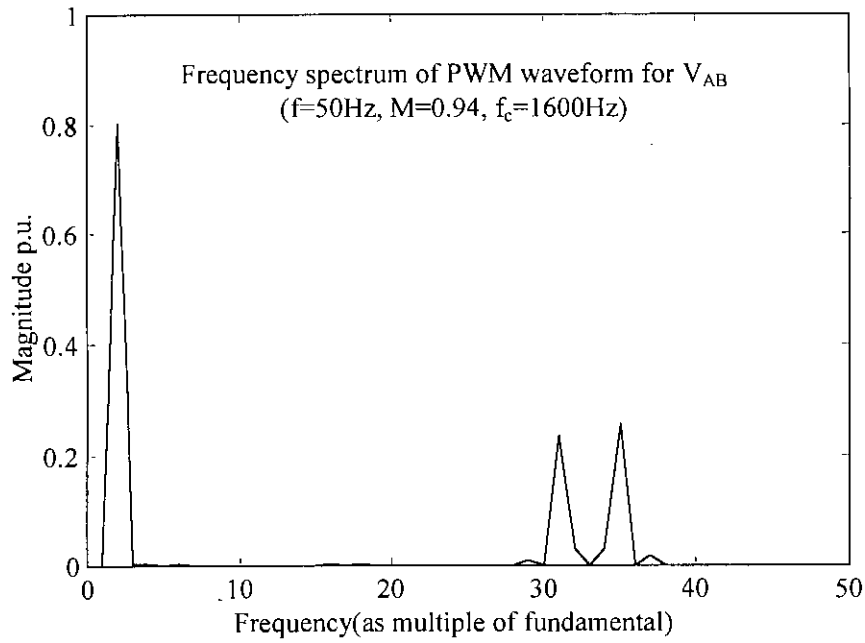


Fig. 4.24. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 50$ Hz, $M = 0.94$, $f_c = 1600$ Hz)

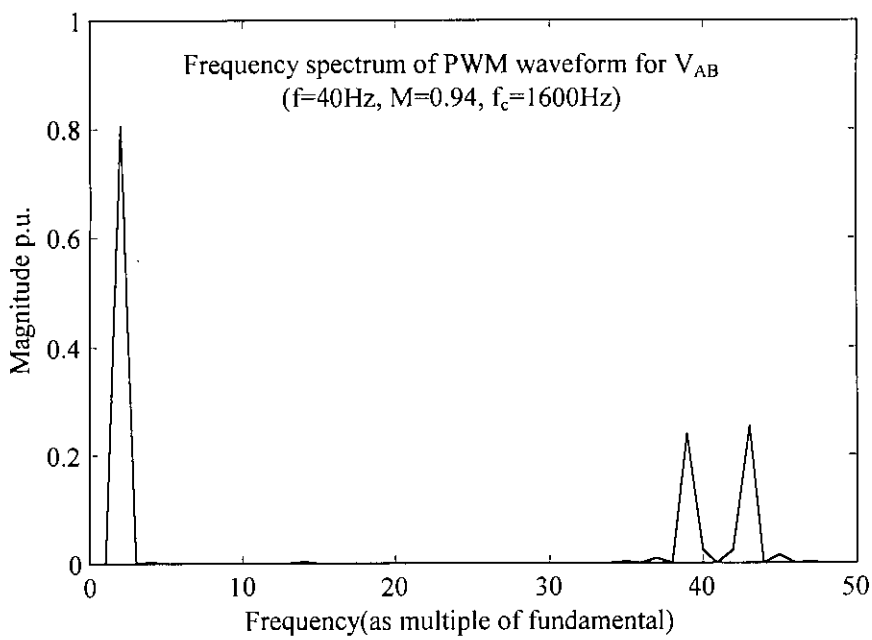


Fig. 4.25. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 40$ Hz, $M = 0.94$, $f_c = 1600$ Hz)

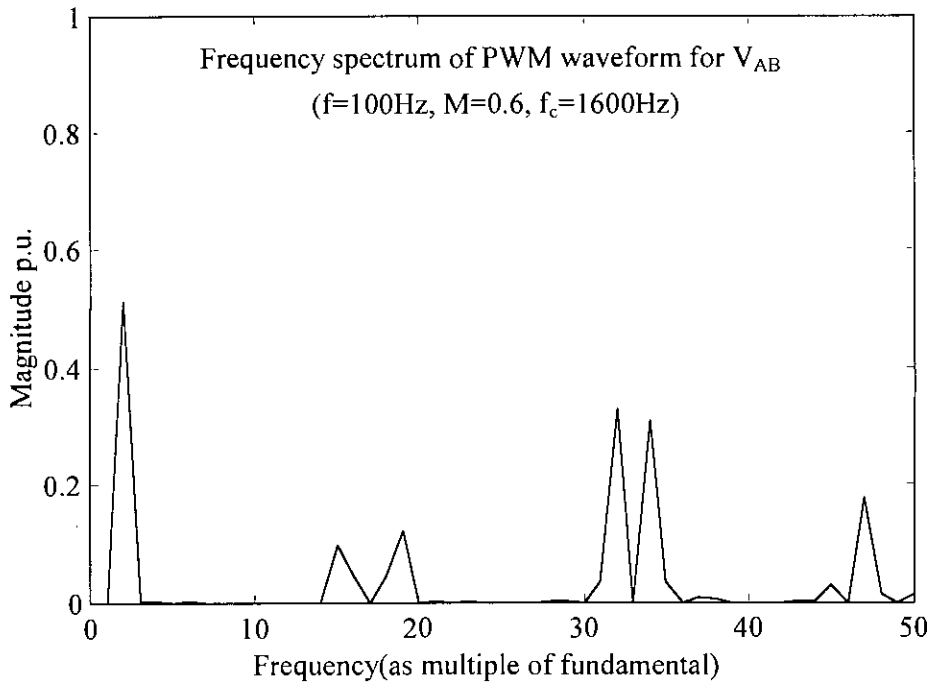


Fig. 4.26. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 100$ Hz, $M = 0.6$, $f_c = 1600$ Hz)

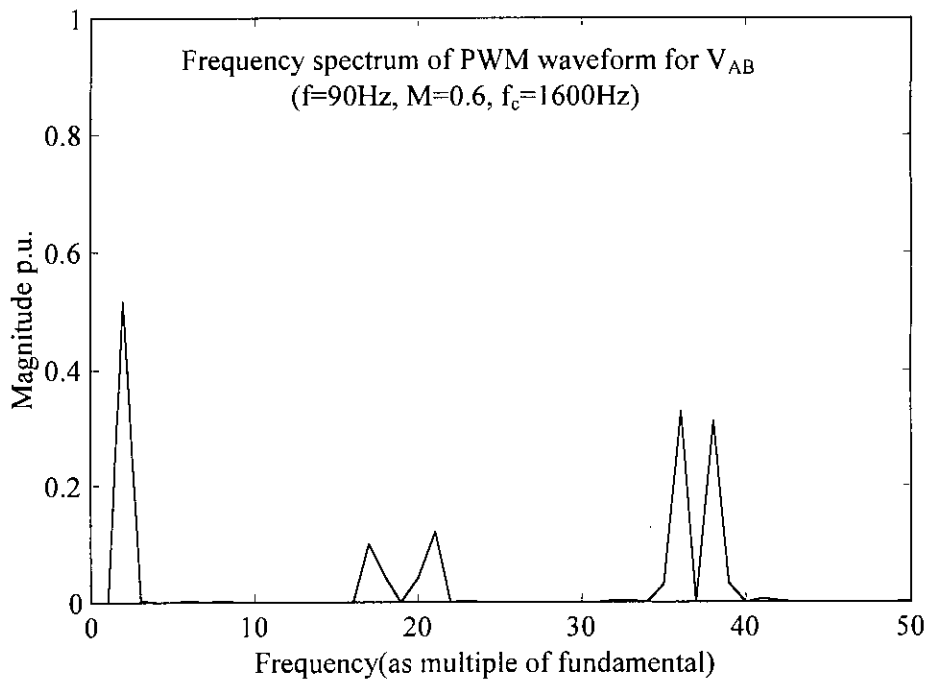


Fig. 4.27. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 90$ Hz, $M = 0.6$, $f_c = 1600$ Hz

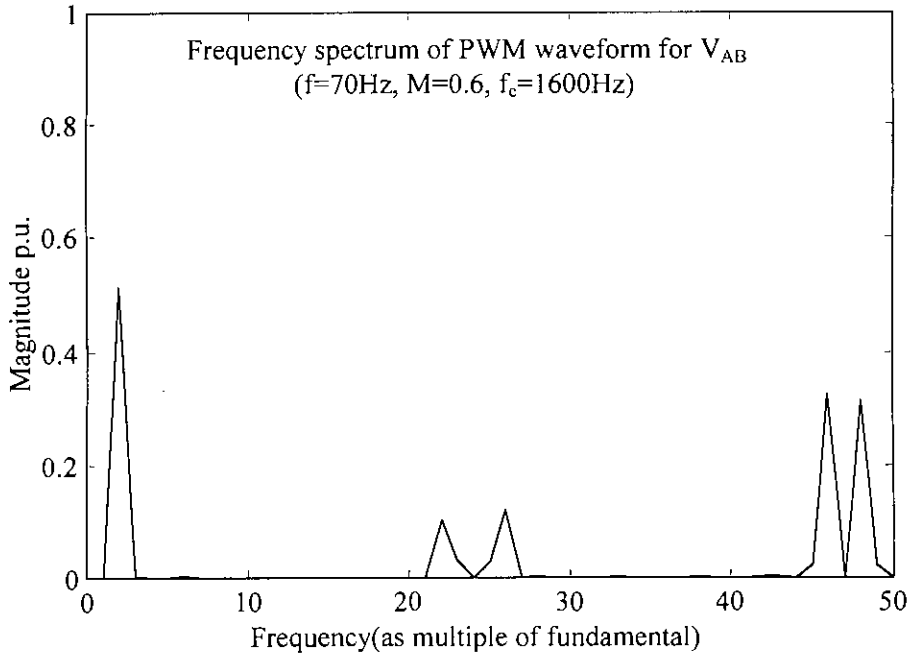


Fig. 4.28. Frequency spectrum of PWM waveform for V_{AB} (simulated)
(f = 70 Hz, M = 0.6, f_c = 1600 Hz)

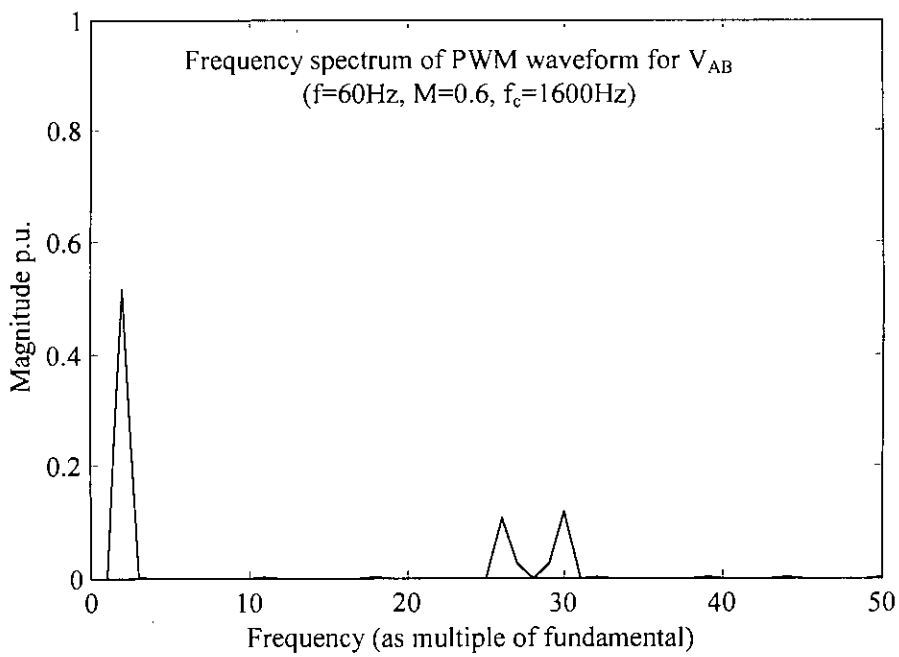


Fig. 2.4.29. Frequency spectrum of PWM waveform for V_{AB} (simulated)
(f = 60 Hz, M = 0.6, f_c = 1600 Hz)

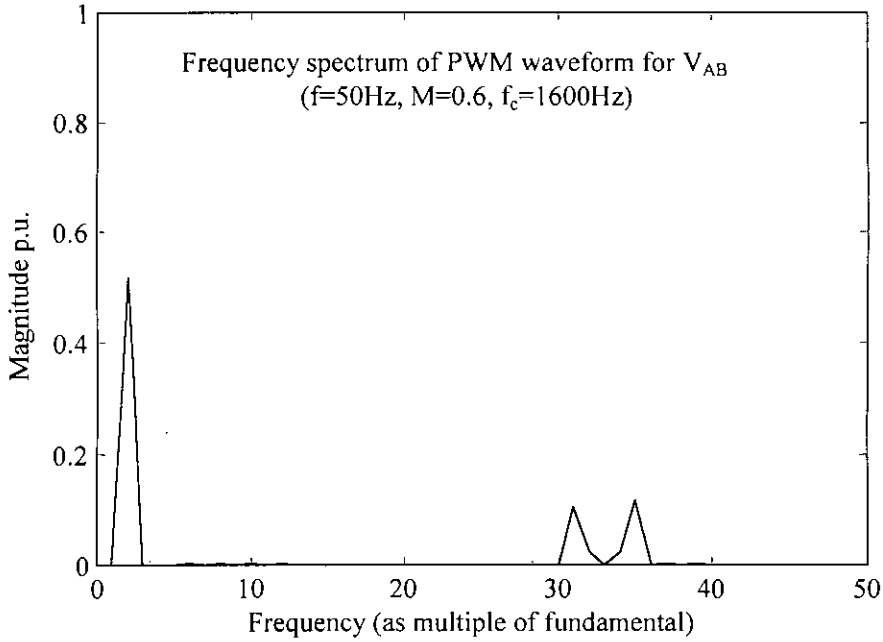


Fig. 4.30. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 50$ Hz, $M = 0.6$, $f_c = 1600$ Hz)

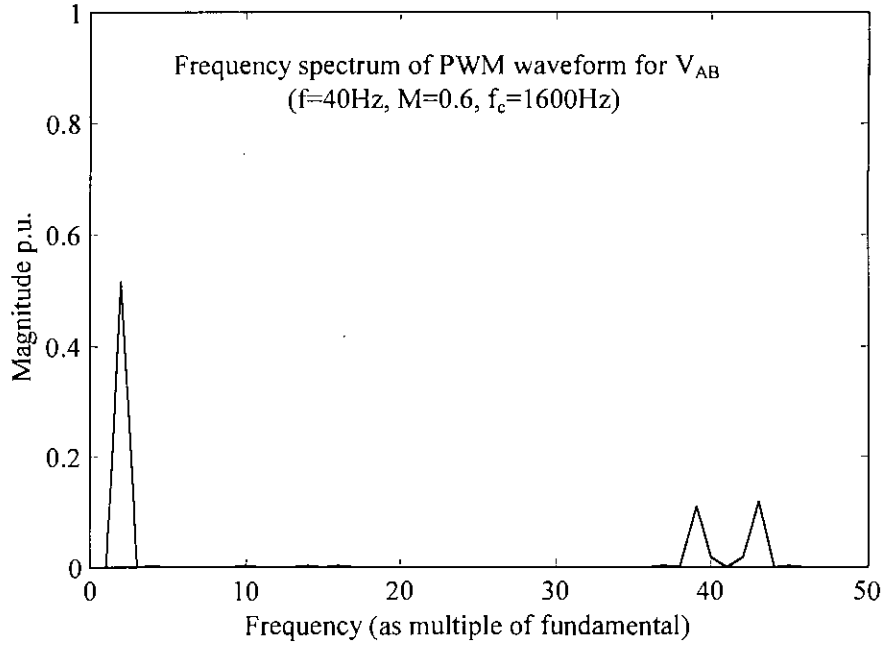


Fig. 4.31.Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 40$ Hz, $M = 0.6$, $f_c = 1600$ Hz)

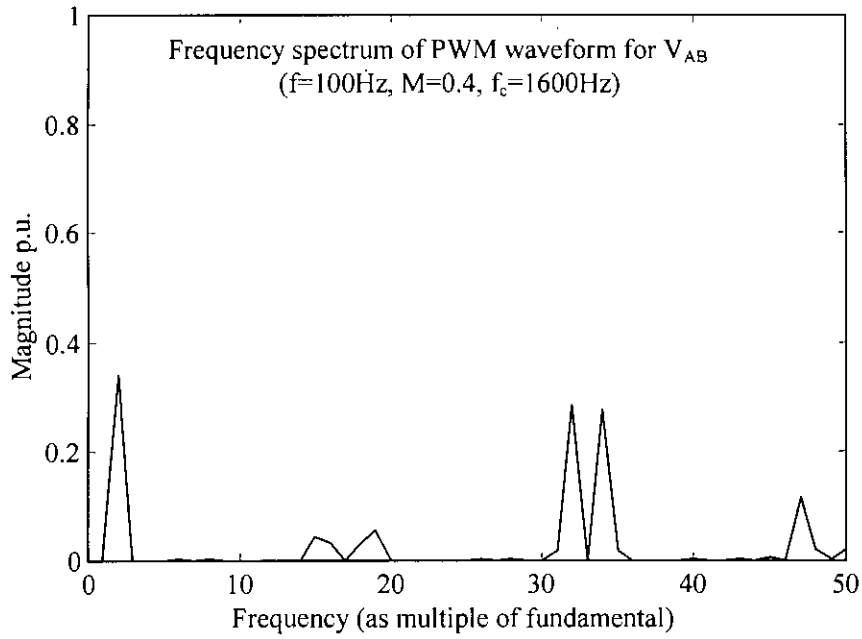


Fig. 4.32. Frequency spectrum of PWM waveform for V_{AB} (simulated)
(f = 100 Hz, M = 0.4, f_c = 1600 Hz)

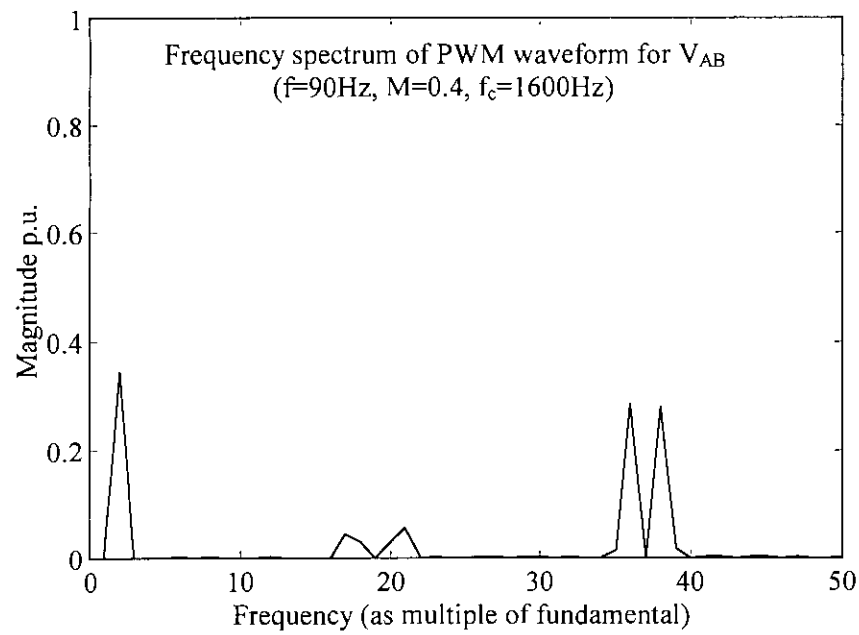


Fig. 4.33. Frequency spectrum of PWM waveform for V_{AB} (simulated)
(f = 90 Hz, M = 0.4, f_c = 1600 Hz)

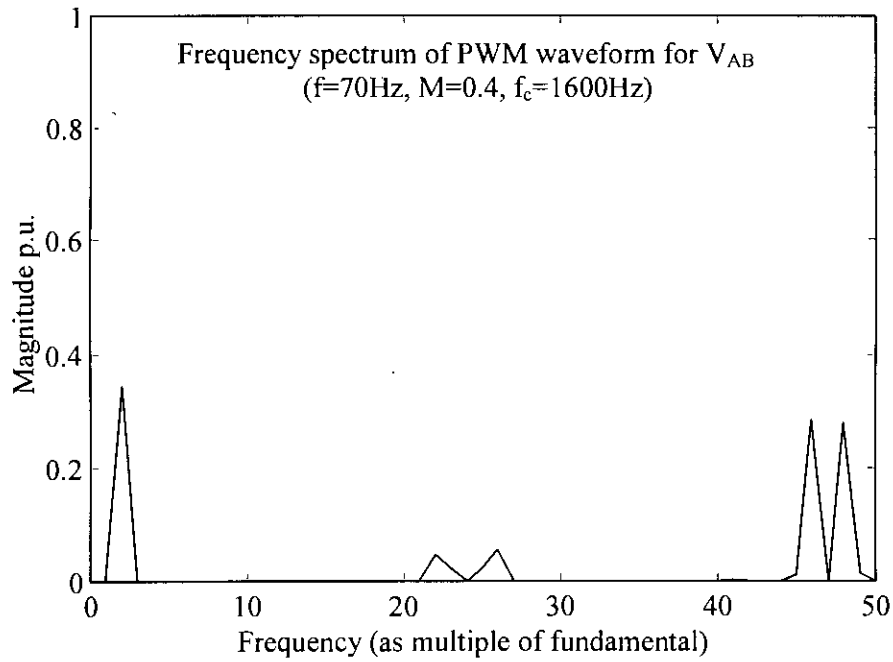


Fig: 4.34. Frequency spectrum of PWM waveform for V_{AB} (simulated)

(f = 70 Hz, M = 0.4, f_c = 1600 Hz)

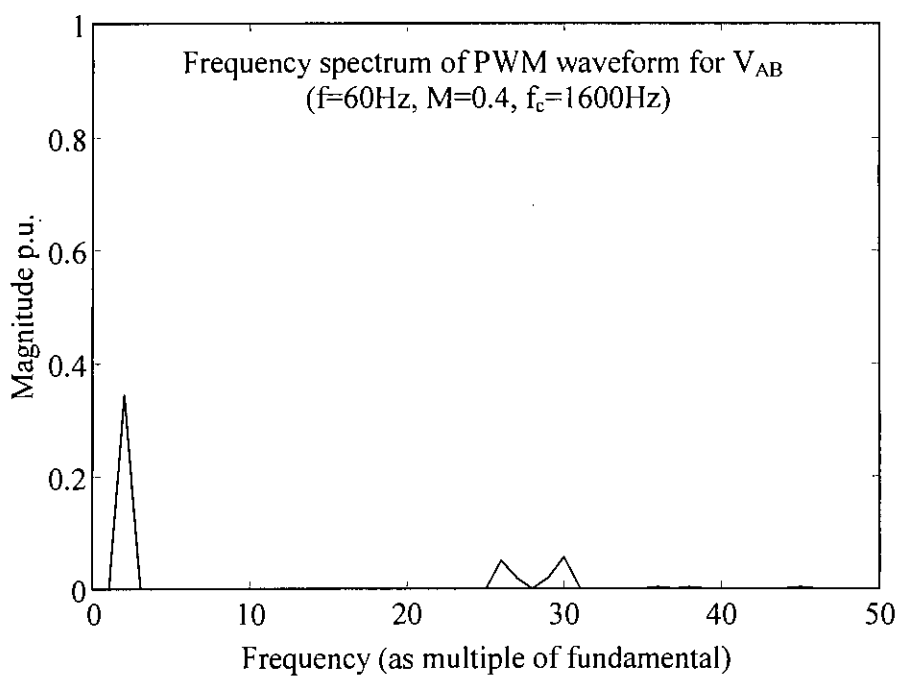


Fig. 4.35. Frequency spectrum of PWM waveform for V_{AB} (simulated)

(f = 60 Hz, M = 0.4, f_c = 1600 Hz)

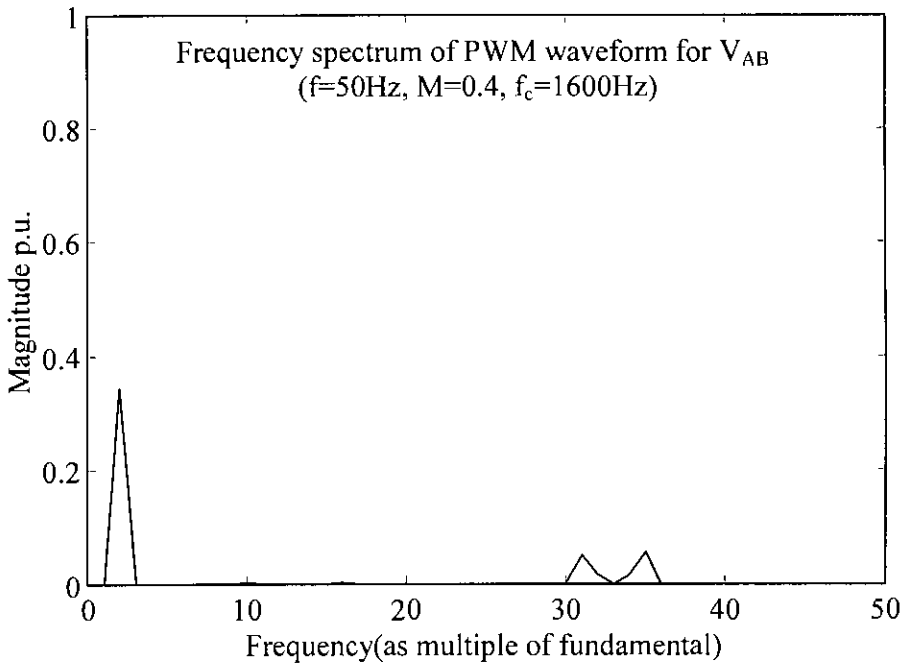


Fig. 4.36. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 50$ Hz, $M = 0.4$, $f_c = 1600$ Hz)

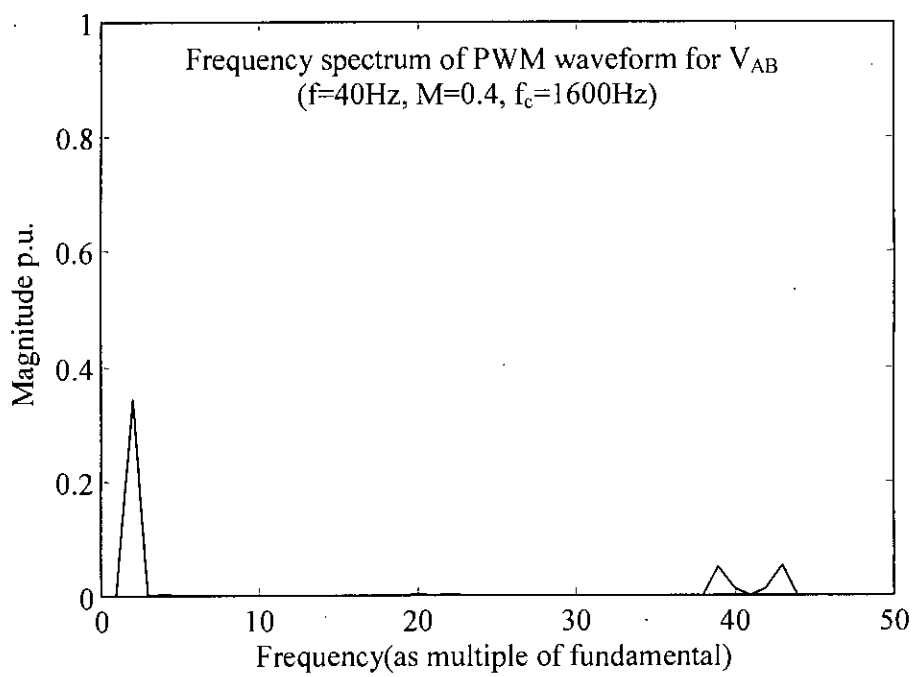


Fig. 4.37. Frequency spectrum of PWM waveform for V_{AB} (simulated)
($f = 40$ Hz, $M = 0.4$, $f_c = 1600$ Hz)

4.4 Real Time PWM Waveform Generation Algorithm

The program [Appendix: Program2] for the real time implementation of the LUT based PWM scheme was developed in Assembly Language used for microprocessor. The algorithm of the program is given below:

1. Store values of $126\sin\theta$, $126\sin(\theta-120^0)$ and $126\sin(\theta-240^0)$ at 1^0 step.
2. Configure port A, port B and port C of PPI 8255-1 as output port and port C of PPI 8255-2 as input port.
3. Set start frequency , f and modulation index, M.
4. Set pulse counter maximum value, $N\left(N=\frac{f_c}{f}\right)$.
5. Initialize counter n=0,
6. Inilitialize CYCLE=0.
7. Compute duty cycles Da, Db and Dc for the n^{th} pulse for phase A, phase B and phase C using equations (3.14), (3.15) and (3.16) respectively.
8. Wait for terminal count (Ripple Carry) from port C of PPI 8255-2.
9. If Ripple Carry, RC=0, go to step 8 else go to step 10.
10. Send Da, Db and Dc to port A, B and C of 8255-1.
11. Increment n.
12. if $n < N$, go to step 7 else go to step 13
13. Increment CYCLE.
14. If CYCLE < 10, go to step 6 else go to step 15.
15. Read frequency and modulation index adjust status pins and set frequency and modulation index value.
16. Go to step 4.

4.5 Summary

The proposed Look Up Table (LUT) based PWM scheme is presented in this chapter together with its block schematic and details. The PWM patterns are generated theoretically using MATLAB program. The look up table of the PWM patterns is then formed for the real time implementation of the scheme. Theoretical patterns and spectrums are presented. The dc

components, different harmonic components and distortion factors of the simulated patterns and spectrums are presented in Table I. Observing the figures of theoretical patterns and spectrums and the values of Table I it can be said that the theoretical performance of the scheme is well. The algorithm for the real time implementation of the scheme is presented finally.

Chapter 5

Experimental Results

5.1 Implementation of the Scheme and Generation of Real Time PWM Pulses

The LUT based PWM scheme is implemented in the laboratory using 8086 Microprocessor. The circuit diagram of the scheme is shown in Fig. 5.1 in details. Other devices that were used to implement the scheme is listed below:

a. Three 32k EPROMs (NM27C256)
b. Two Programmable Peripheral Interfaces (PPI)-8255-1 and 8255-2.
c. One 12-stage Ripple Binary Counter-CD 4040BE
d. One Dual Monostable Timer-74LS123
e. Three CMOS Inverters-4049
f. Two Inverters -74LS04.
g. One Multifunction data acquisition card: 8-Channel Analog and Digital I/O card with 20 pin cable; Model: PCL-711S-A

The scheme is implemented in a hybrid manner. The processor calculates the duty cycles of the PWM pulses for the three phases for a specified modulation index and frequency once in each carrier period. The duty cycles are then normalized on a scale of 128. The processor then sends the three duty cycles (for three phases) to a Programmable Peripheral Interface (PPI) having three 8-bit ports (Each port of the PPI is connected to the respective EPROM of that phase). A 12-bit binary counter (CD 4040) is connected to the lower 8-bit address lines (A_0 - A_7) that scan the 256 stored samples of the EPROM. The clock pulse is generated using two inverters (74LS04) along with external R-C components. The clock is buffered with an inverter and then connected to the clock input of the binary counter. As the counter reaches its maximum count value (255), it sends a ripple carry output (RC) at line D_8 (pin 12) of the counter to reset it and to prompt the microprocessor to send the duty cycles of the next carrier period to the respective EPROMs. When one fundamental period is reached, the processor points to the first carrier

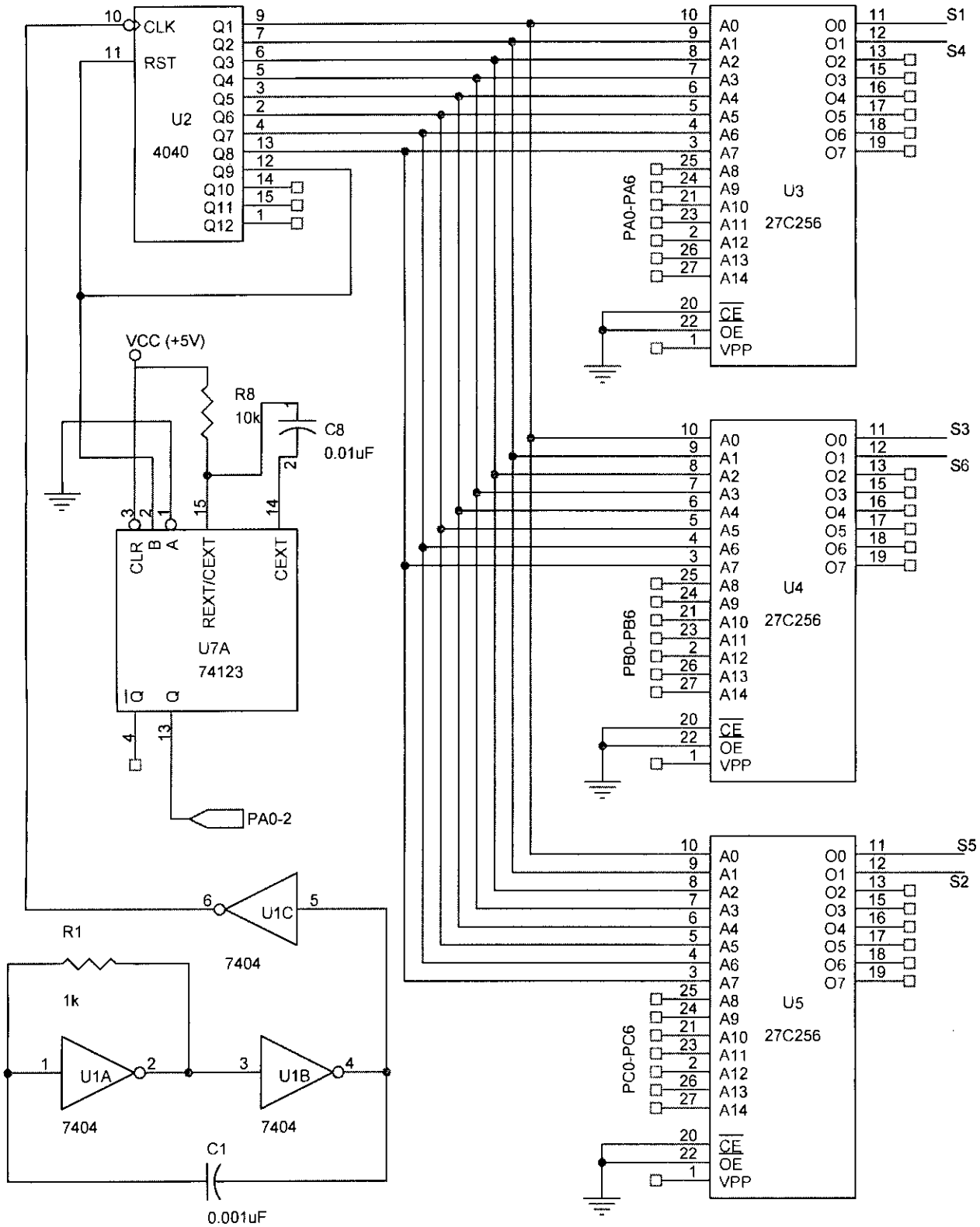


Fig 5.1 Detailed circuit diagram of the proposed LUT based PWM scheme

cycle. For resetting, line D_8 (pin 12) of the counter is connected to its reset input (pin 11). When D_8 is HIGH (logical "1") then the counter is reset. The counter possesses edge-triggered resetting. Therefore, it will be reset just for a change in the level of voltage at the reset input from LOW to HIGH whatever the width of the RC pulse may be. The RC pulse is sent to the processor through the Programmable Peripheral Interface (8255-1). The processor scans at the port of PPI (8255-1) for ripple carry pulse at regular time intervals that is of microsecond range. Now, if the width of the RC pulse is of nanosecond range then the processor may miss to scan all the RC pulses sent. Hence, the width of the RC pulse needs to be increased. Since, resetting the counter is done at the positive edge of the RC pulse, a positive edge triggered dual monostable timer (74LS123) with external R-C components is used for increasing the width of the RC pulse. The width of the RC pulse is increased to a minimum value that must be greater than the time interval of microprocessor to scan the port of PPI for RC pulse. The D_0 and D_1 lines of the EPROM data bus presents the real time PWM pulses for the upper and lower leg switches of a phase of the inverter.

The PWM patterns obtained at S_1, S_4 of EPROM1, S_3, S_6 of EPROM2 and S_5, S_2 of EPROM3 are used for triggering the gates a, a′, b, b′, c and c′of the transistors of a three phase full bridge voltage source inverters as shown in fig.5.2. V_a, V_b and V_c are the output voltages of the inverter. Q_1 through Q_6 are six power transistors that shape the output which are controlled by the triggering signals at the gates a, a′, b, b′, c and c′. The transistors are numbered in the sequence of gating the transistors (e.g., 123, 234, 345, 456, 561,612). When an upper transistor is switched on (i.e. when a, b, or c is 1), the corresponding lower transistor is switched off (i.e. the corresponding a′, b′ or c′ is 0). The upper switches of the inverter are labeled with odd numbers whereas the lower switches are labeled with even numbers. The switches Q_1, Q4 are assigned for phase A, Q_3, Q6 for phase B and Q_5, Q2 for phase C respectively. In the present scheme, the output S_1, S_4 of EPROM1 are the gating signals for transistors Q_1,Q_4, the output S_3, S_6 are gating signals for transistors Q_3,Q_6 and output S_5, S_2 of EPROM3 are the gating signals for transistors Q_5,Q_2.

5.2 Determination of the Carrier Frequency

Some real time PWM waveforms were generated for different modulating frequencies and modulation indices. The carrier frequency was kept constant. The desired carrier frequency can be obtained by the appropriate choice of clock frequency. The clock frequency is determined by

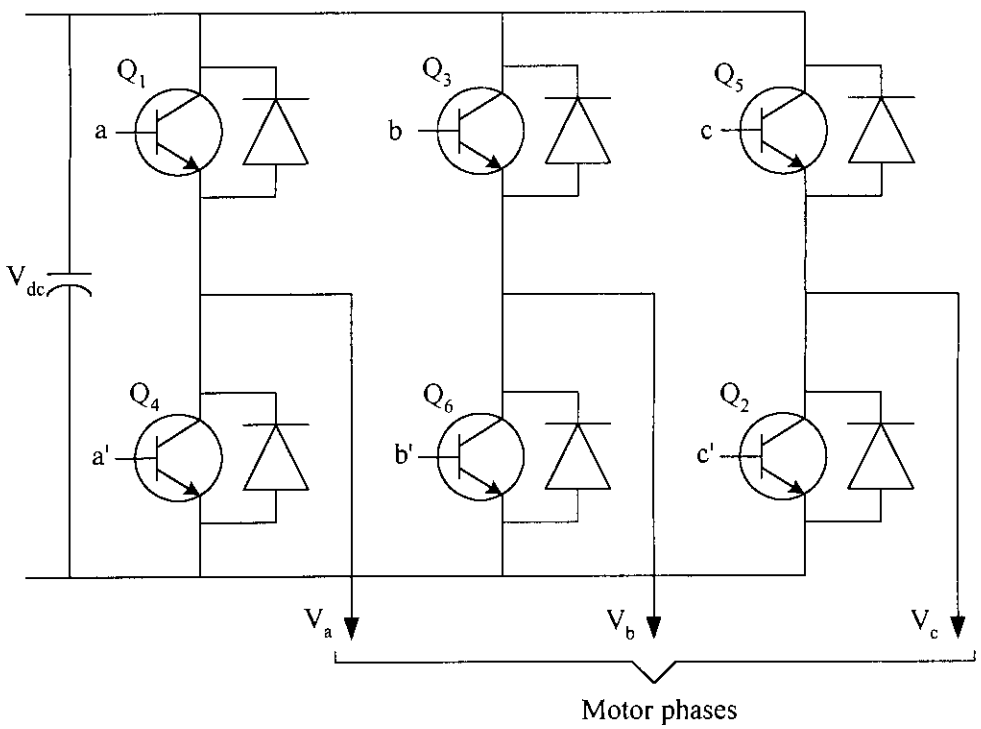


Fig. 5.2 Three phase VSI diagram

the relation

$$f_{clock} = \frac{1}{3RC} \tag{5.1}$$

where, R and C are the resistor and capacitor connected externally to the clock pulse generator(7404).

In the proposed scheme, the carrier frequency was held at f_c = 1600 Hz and to obtain this frequency the value of R and C was chosen in the following way:

Total number samples in a carrier cycle = 256.

So, for getting f_c = 1600 Hz the clock frequency needed

f_{clock} = 1600 $\times$256 = 409600 Hz

Let us take C = 0.001 μF

So, from (5.1)

$$409600 = \frac{1}{3 \times R \times 0.001 \times 10^{-6}}$$

$\Rightarrow$ R = 813.8 Ω

Thus, for getting carrier frequency f_c = 1600 Hz the value of external resistor and capacitor in the clock pulse generator circuit is chosen as

R = 813.8 Ω

C= 0.001 μF.

5.3 Experimental PWM Patterns and Corresponding Spectrums for Different Frequencies and Modulation Indices

The real time PWM patterns were obtained through the real time implementation of the proposed scheme. The data corresponding to the PWM waveforms was acquired through a multifunction data acquisition card (PCL-711S-A). Using these data the experimental PWM waveforms and corresponding spectrums were plotted. Some experimental PWM waveforms and corresponding spectrums for different modulation indices and modulating frequencies are shown in Figs. 5.3 – 5.32.

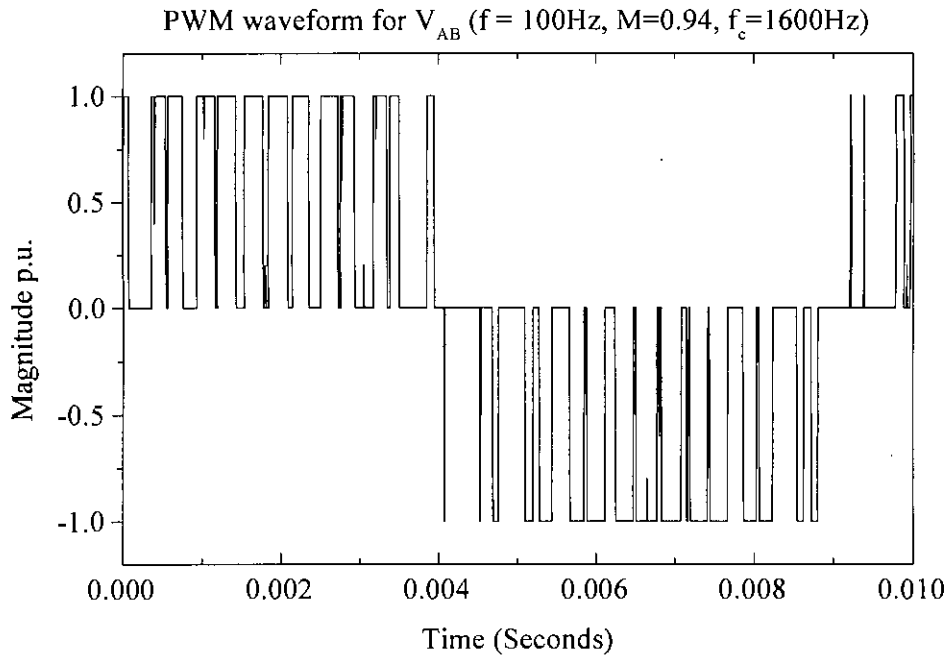


Fig. 5.3. PWM waveform pattern of V_{AB} (experimental)
(f =100 Hz, M = 0.94, f_c = 1600Hz)

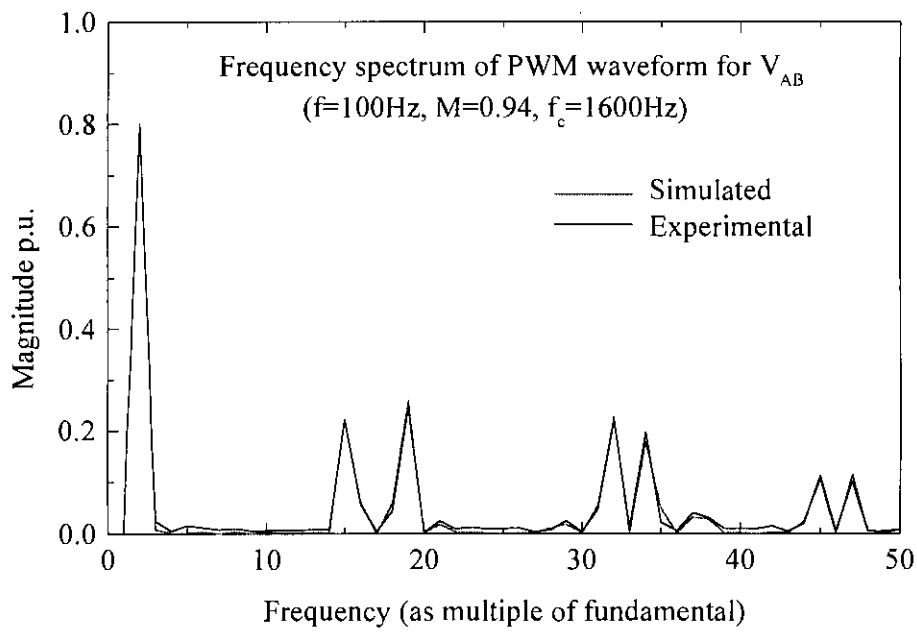


Fig. 5.4. Frequency spectrum of PWM waveform for V_{AB}
(f =100 Hz, M = 0.94, f_c = 1600Hz)

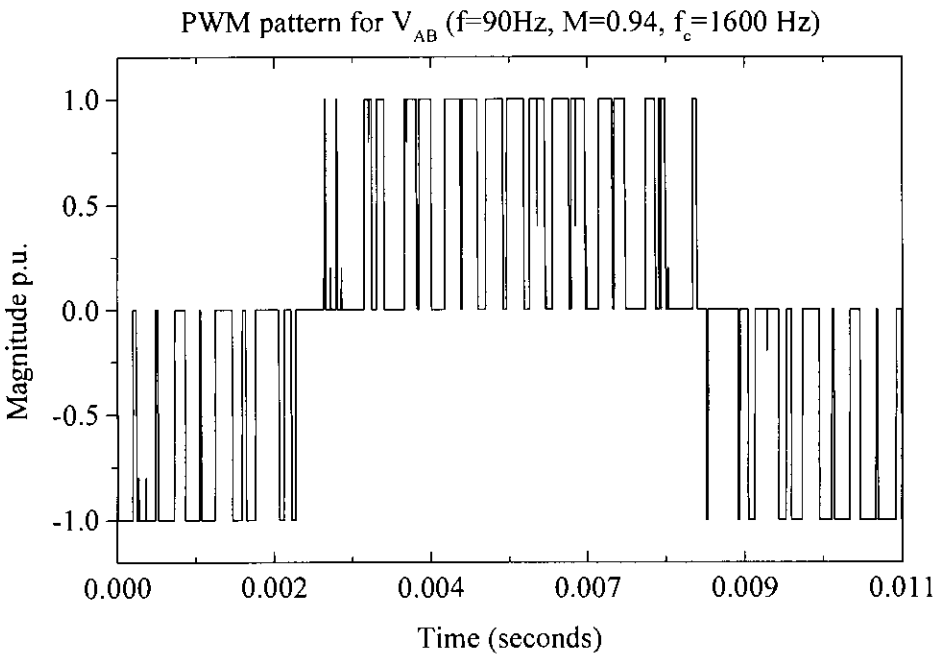


Fig. 5.5. PWM waveform pattern for V_{AB} (experimental)
(f = 90Hz, M = 0.94, f_c = 1600Hz)

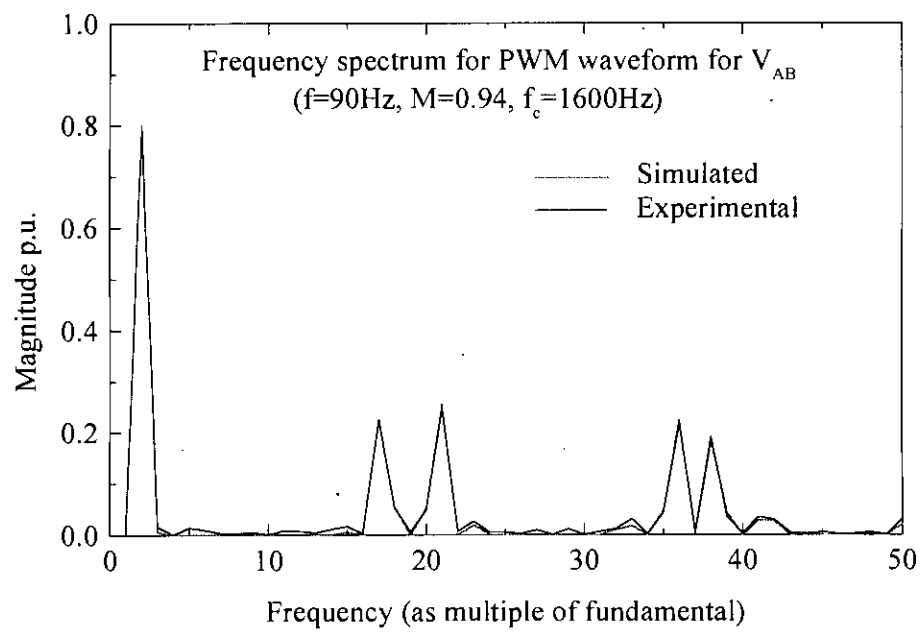


Fig. 5.6. Frequency spectrum of PWM waveform for V_{AB}
(f = 90 Hz, M = 0.94, f_c = 1600 Hz)

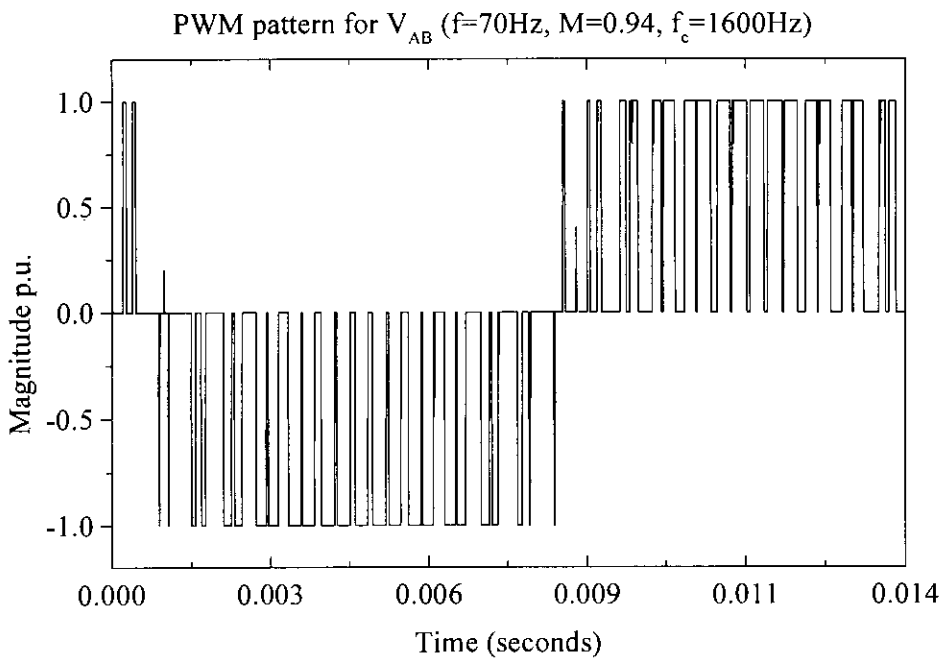


Fig. 5.7. PWM waveform pattern for V_{AB} (experimental)
(f = 70 Hz, M = 0.94, f_c = 1600 Hz)

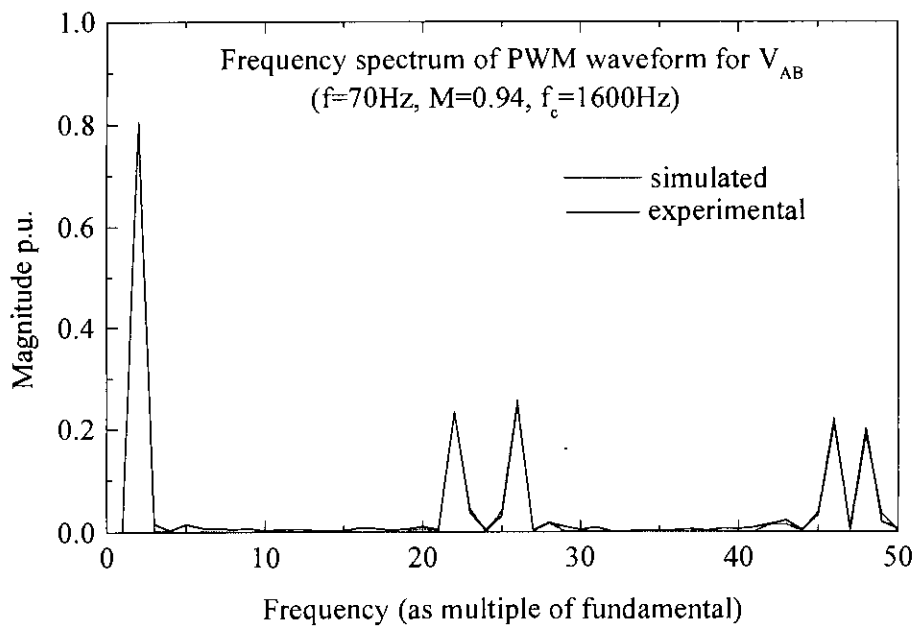


Fig. 5.8. Frequency spectrum of PWM waveform for V_{AB}
(f = 70 Hz, M = 0.94, f_c = 1600 Hz)

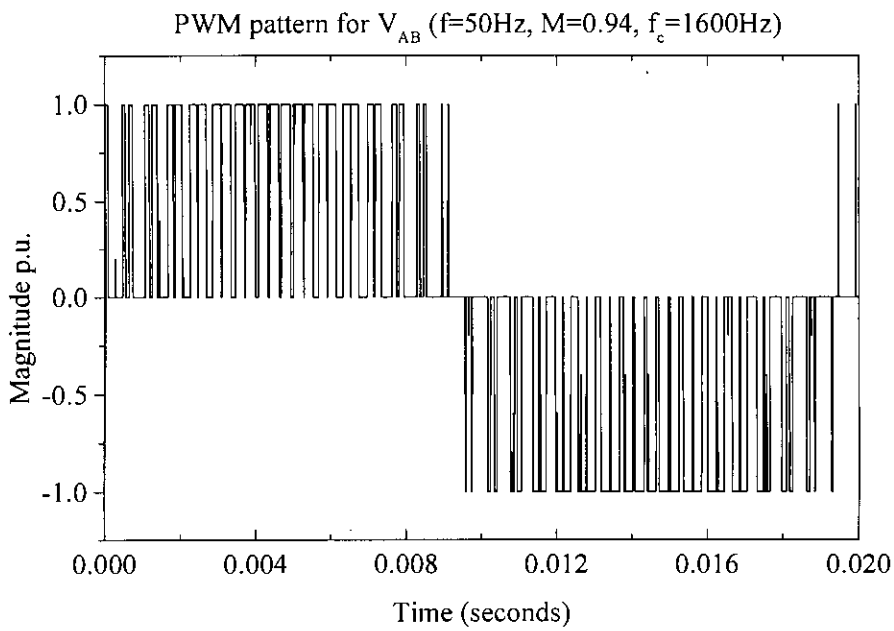


Fig. 5.9. PWM waveform pattern for V_{AB} (experimental)
(f = 50 Hz, M = 0.94, f_c = 1600 Hz)

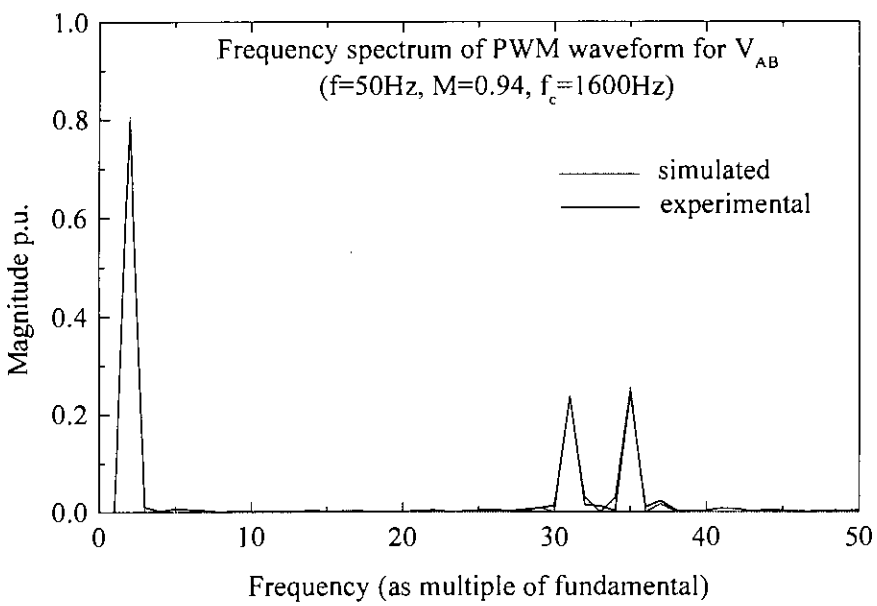


Fig. 5.10. Frequency spectrum of PWM waveform for V_{AB}
(f = 50 Hz, M = 0.94, f_c = 1600 Hz)

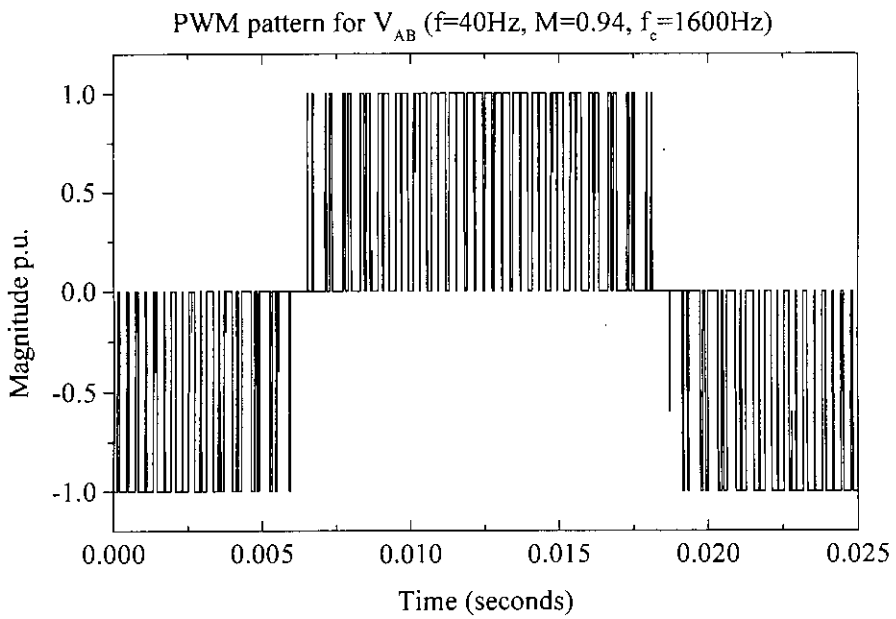


Fig. 5.11. PWM waveform pattern for V_{AB} (experimental)
(f = 40 Hz, M = 0.94, f_c = 1600 Hz)

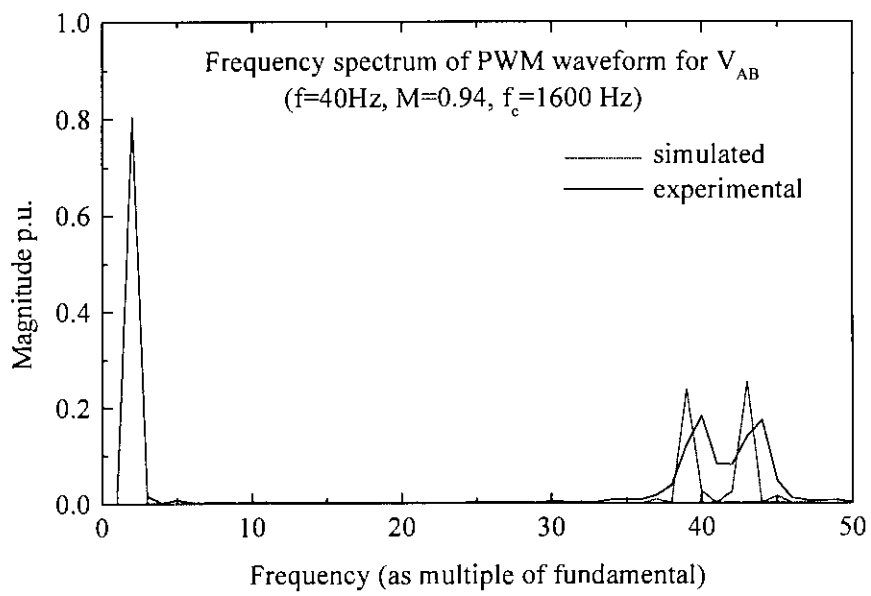


Fig. 5.12. Frequency spectrum of PWM waveform for V_{AB}
(f = 40 Hz, M = 0.94, f_c = 1600 Hz)

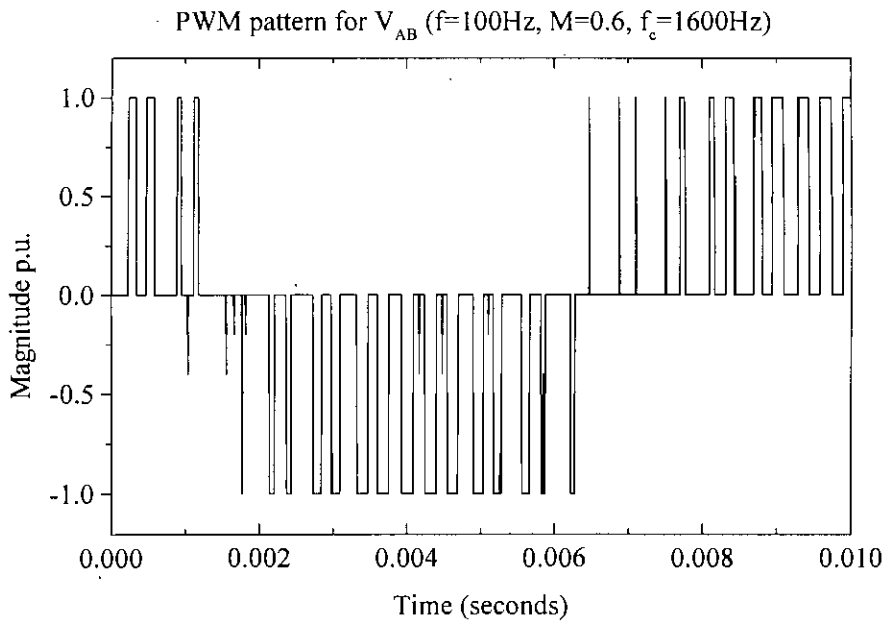


Fig. 5.13. PWM waveform pattern for V_{AB} (experimental)
(f = 100 Hz, M = 0.6, f_c = 1600 Hz)

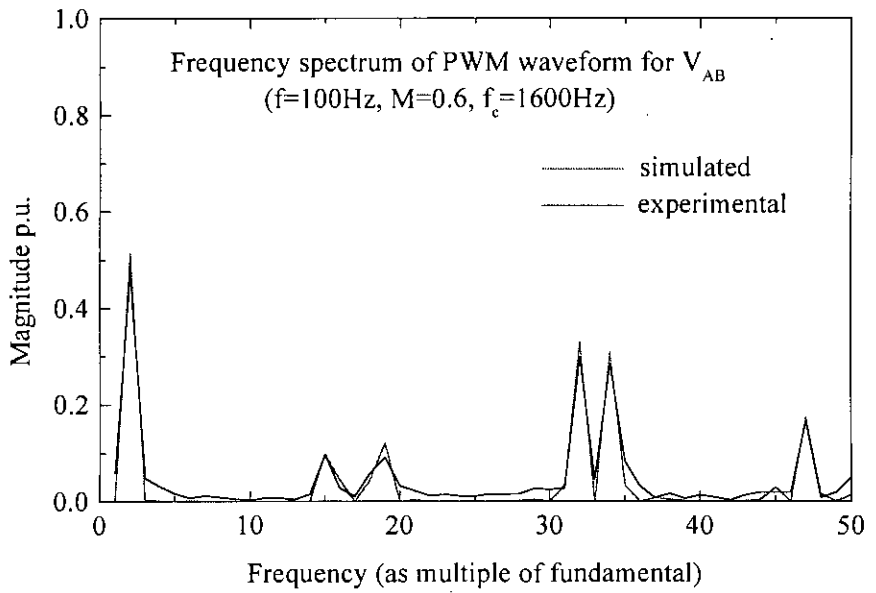


Fig. 5.14. Frequency spectrum PWM waveform for V_{AB}
(f = 100 Hz, M = 0.6, f_c = 1600 Hz)

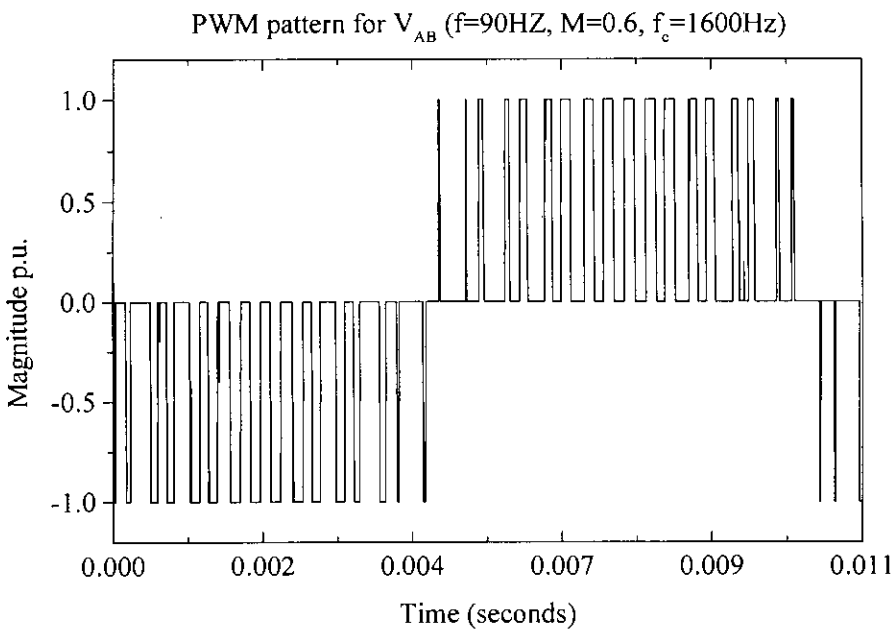


Fig. 5.15. PWM waveform pattern for V_{AB} (experimental)
(f = 90 Hz, M = 0.6, f_c = 1600 Hz)

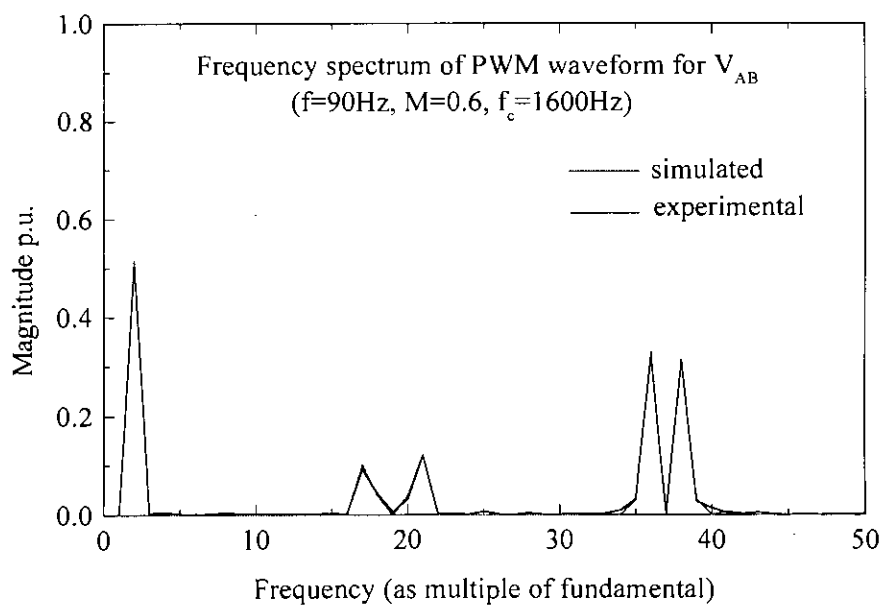


Fig. 5.16. Frequency spectrum of PWM waveform for V_{AB}
(f = 90 Hz, M = 0.6, f_c = 1600 Hz)

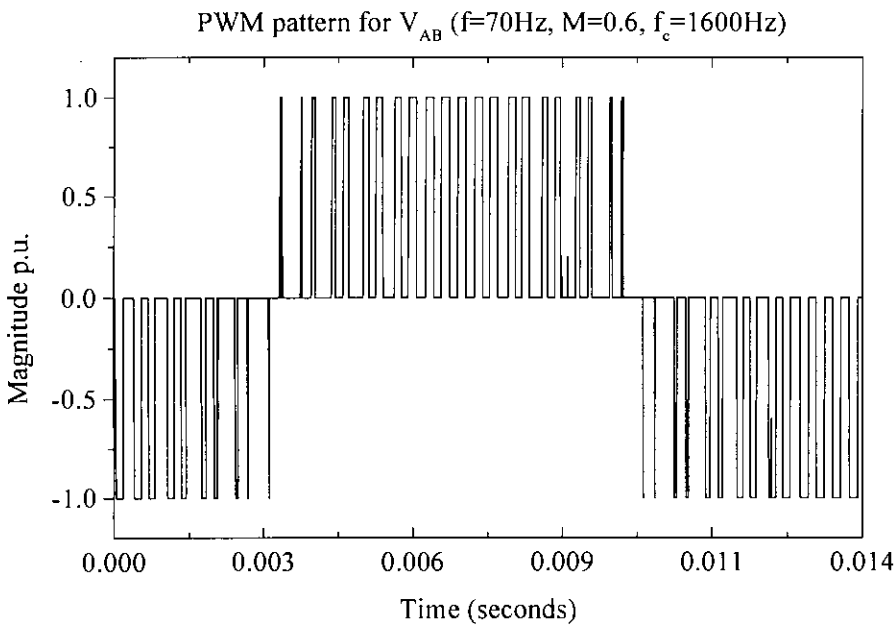


Fig. 5.17. PWM waveform pattern for V_{AB} (experimental)
(f = 70 Hz, M = 0.6, f_c = 1600 Hz)

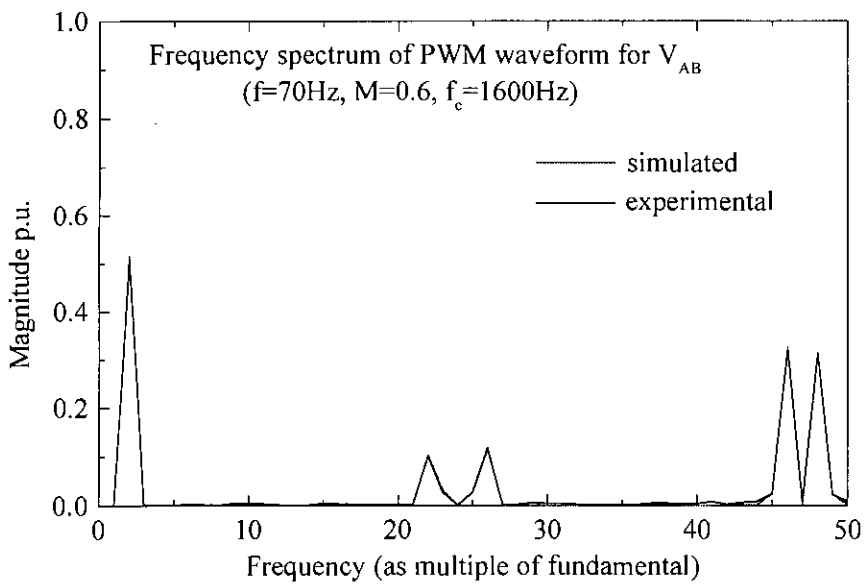


Fig. 5.18. Frequency spectrum of PWM waveform for V_{AB}
(f = 70 Hz, M = 0.6, f_c = 1600 Hz)

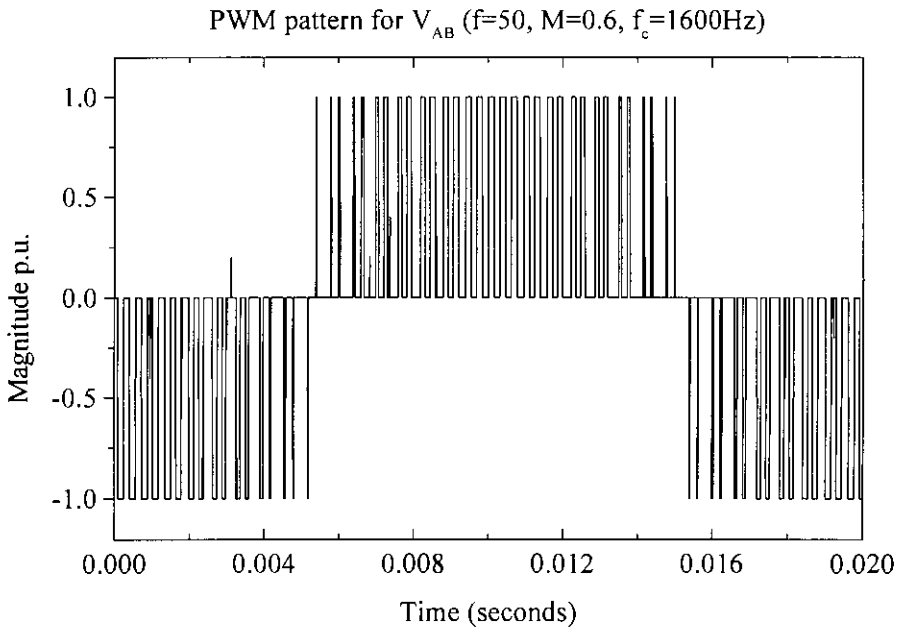


Fig. 5.19. PWM waveform pattern for V_{AB} (experimental)
(f = 50 Hz, M = 0.6, f_c = 1600 Hz)

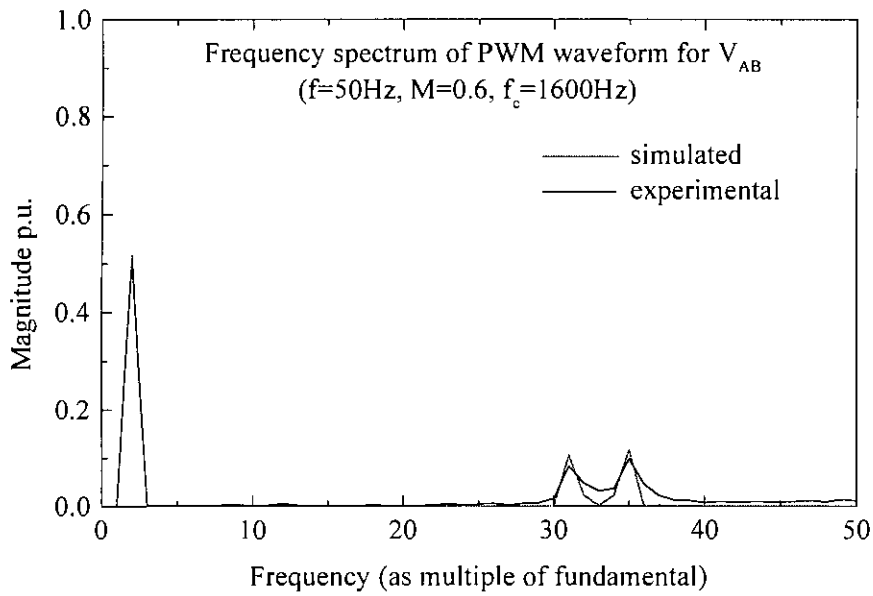


Fig. 5.20. Frequency spectrum of PWM waveform for V_{AB}
(f = 50 Hz, M = 0.6, f_c = 1600 Hz)

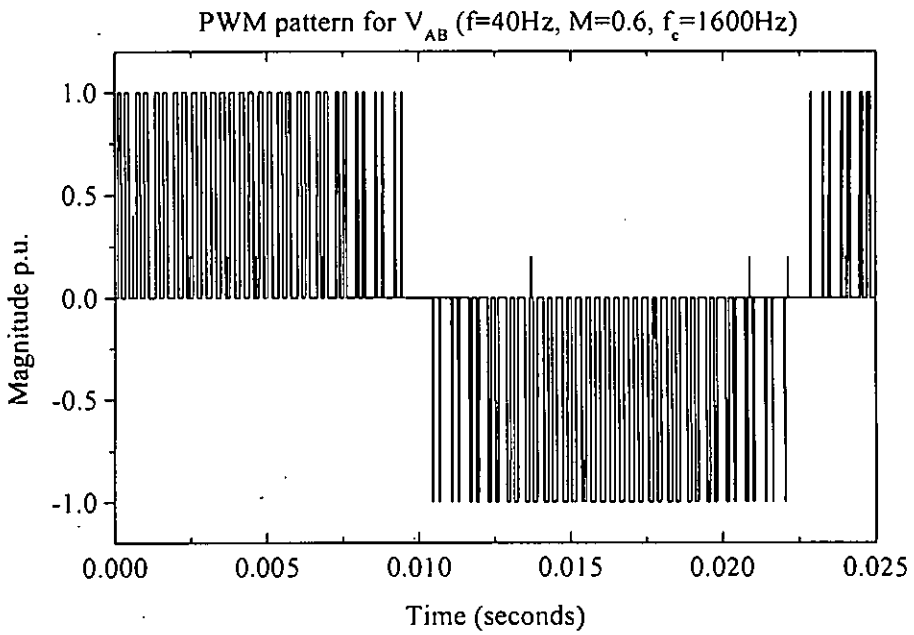


Fig. 5.21. PWM waveform pattern for V_{AB} (experimental)
(f = 40 Hz, M = 0.6, f_c = 1600 Hz)

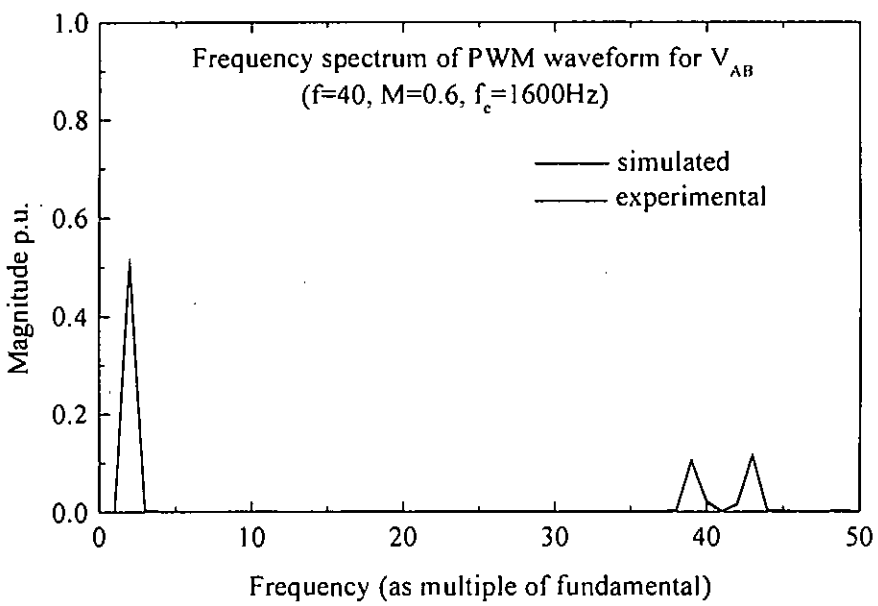


Fig. 5.22. Frequency spectrum of PWM waveform for V_{AB}
(f = 40 Hz, M = 0.6, f_c = 1600 Hz)

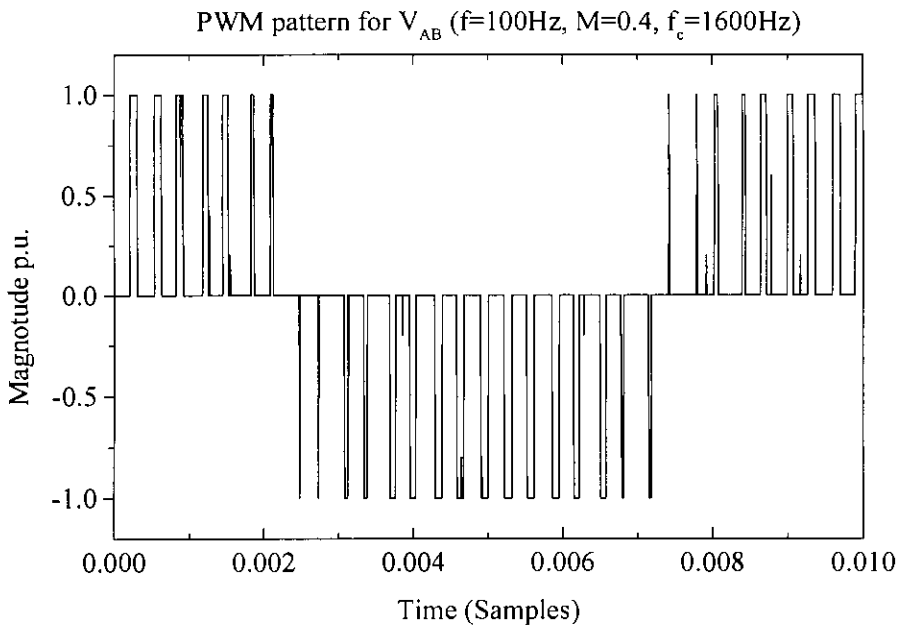


Fig. 5.23. PWM waveform pattern for V_{AB} (experimental)
(f = 100 Hz, M = 0.4, f_c = 1600 Hz)

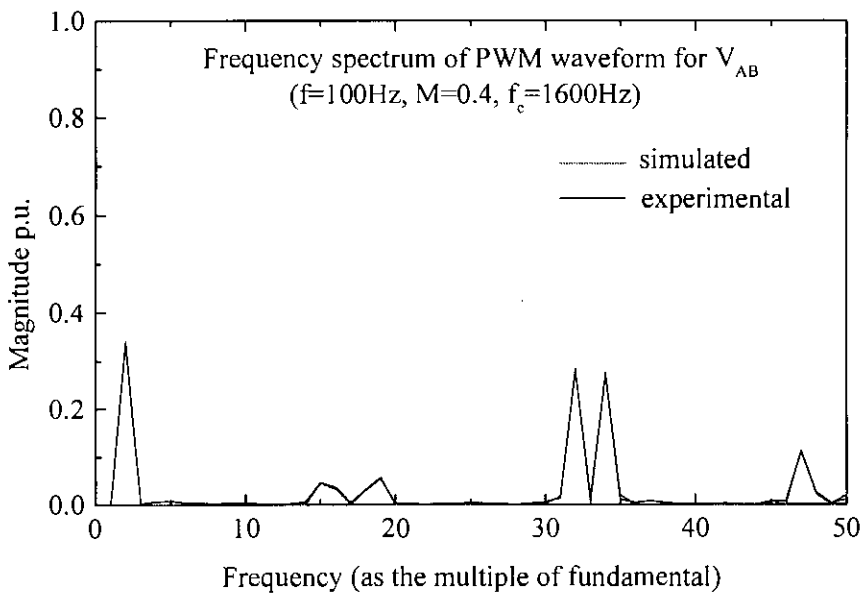


Fig. 5.24. Frequency spectrum of PWM waveform for V_{AB}
(f = 100 Hz, M = 0.4, f_c = 1600 Hz)

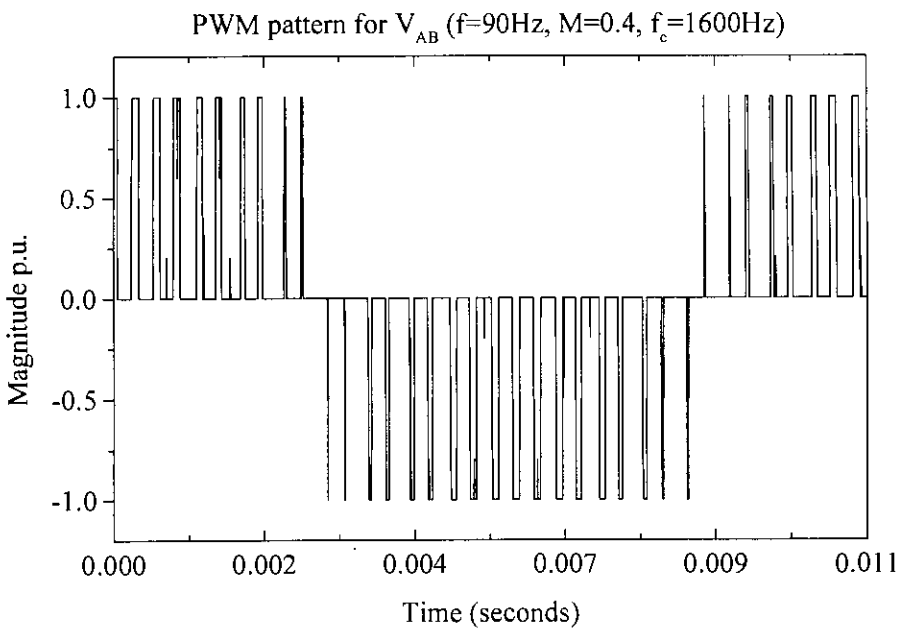


Fig. 5.25. PWM waveform pattern for V_{AB} (experimental)
(f = 90 Hz, M = 0.4, f_c = 1600 Hz)

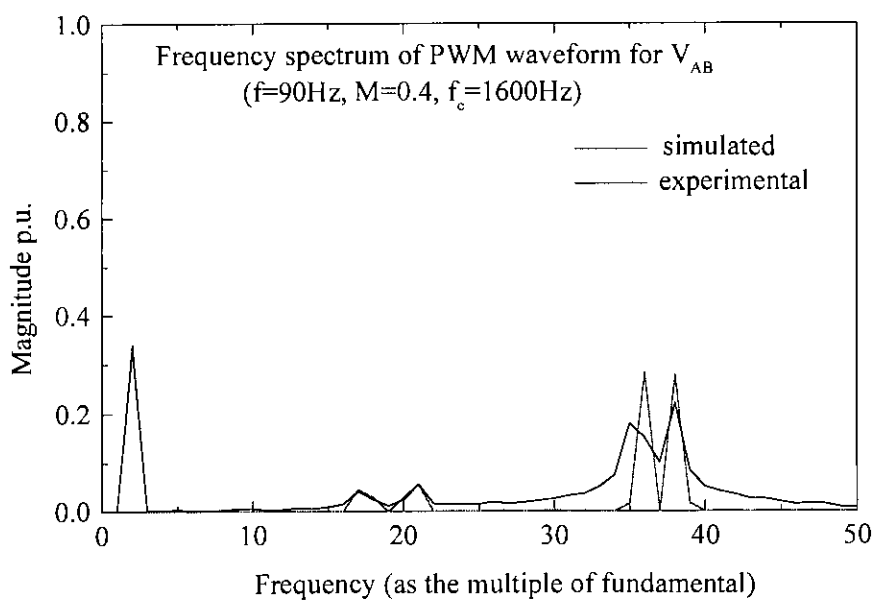


Fig. 5.26. Frequency spectrum of PWM waveform for V_{AB}
(f = 90 Hz, M = 0.4, f_c = 1600 Hz)

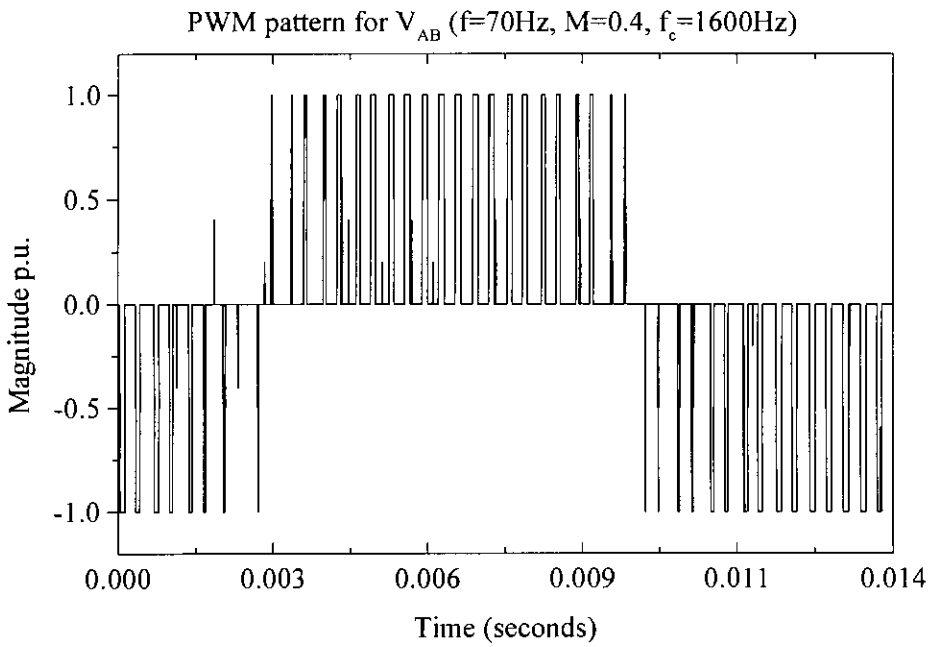


Fig. 5.27. PWM waveform pattern for V_{AB} (experimental)
(f = 70 Hz, M = 0.4, f_c = 1600 Hz)

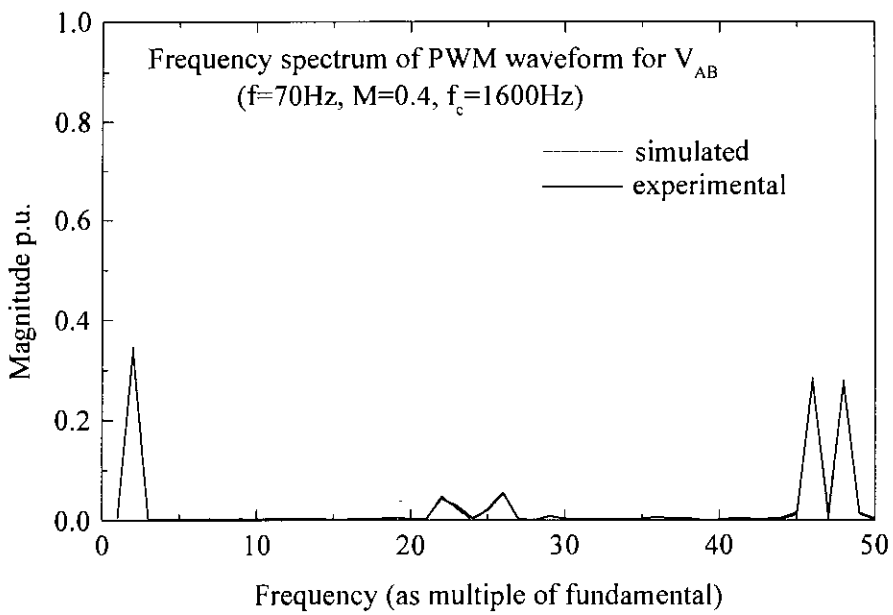


Fig. 5.28. Frequency spectrum of PWM waveform for V_{AB}
(f = 70 Hz, M = 0.4, f_c = 1600 Hz)

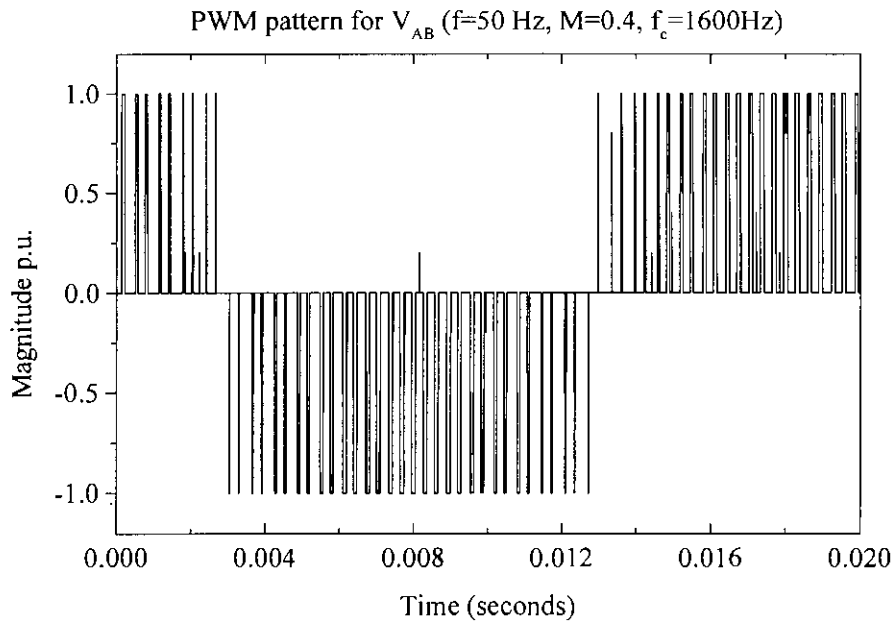


Fig. 5.29. PWM waveform pattern for V_{AB} (experimental)
(f = 50 Hz, M = 0.4, f_c = 1600 Hz)

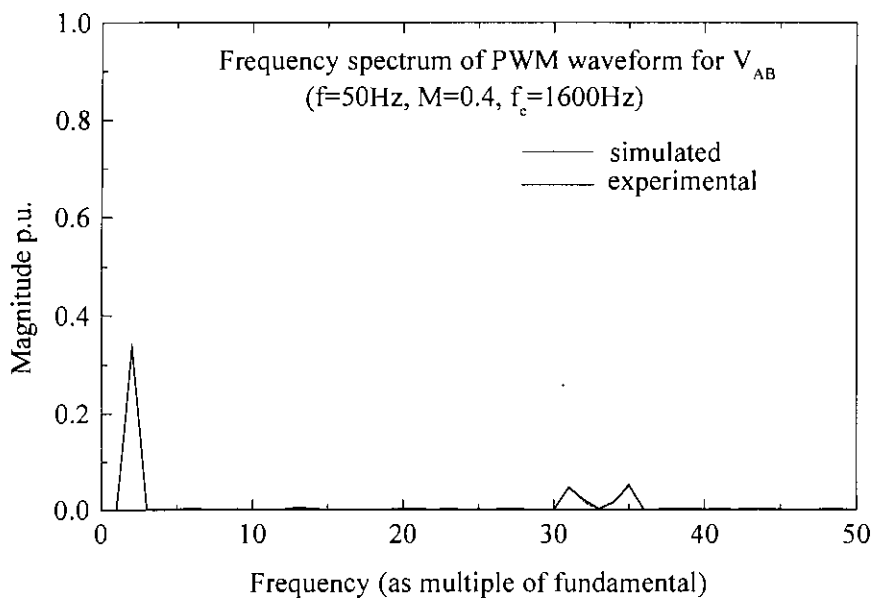


Fig. 5.30. Frequency spectrum of PWM waveform for V_{AB}
(f = 50 Hz, M = 0.4, f_c = 1600 Hz)

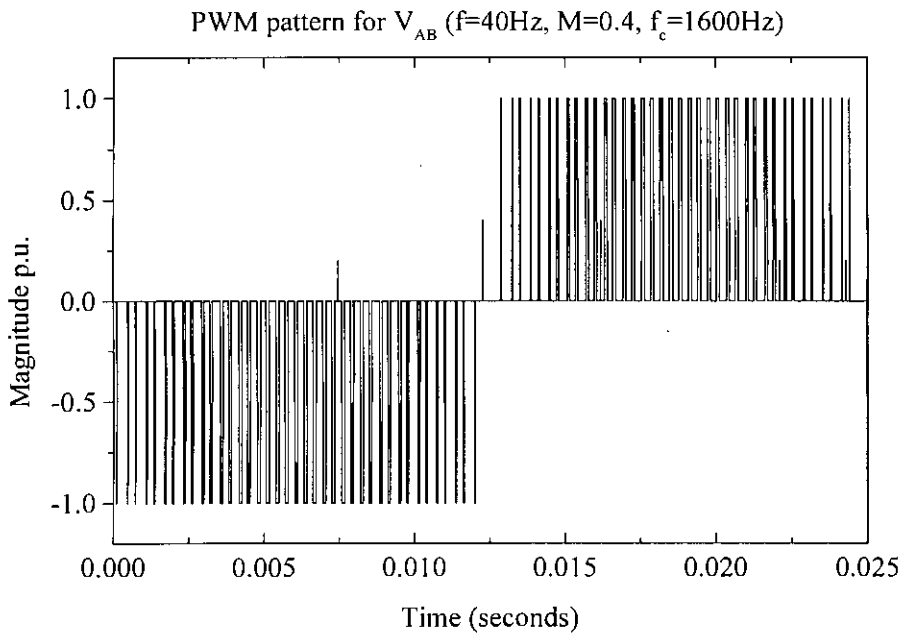


Fig. 5.31. PWM waveform pattern for V_{AB} (experimental)
(f = 40 Hz, M = 0.4, f_c = 1600 Hz)

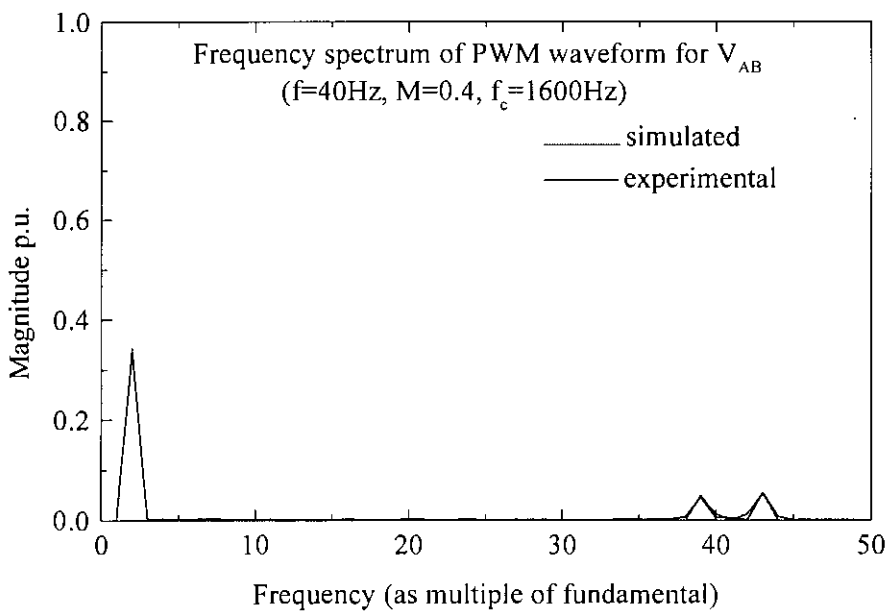


Fig. 5.32. Frequency spectrum of PWM waveform for V_{AB}
(f = 40 Hz, M = 0.4, f_c = 1600 Hz)

In Figs.5.3 - 5.32, we see that, the plots of both simulated and experimental frequency spectrum for a specific modulating frequency and modulation index overlap each other with some negligible discrepancies. Since it is difficult to compare the PWM patterns on the same plot therefore the experimental PWM patterns are shown on separate plots. Comparing these patterns with the simulated patterns shown in chapter 4 (Figs. 4.2 – 4.37) we see that for specific modulating frequency and modulation index the patterns are nearly similar.

5.4 Testing of the Scheme

The scheme is tested on a three-phase prototype inverter with IRF540 MOSFET switches. The inverter is run from a 60V regulated dc power supply. A cage type induction motor is connected to the inverter as three phase load. The PWM patterns for upper switch (s_1) of phase A and the waveform of phase current for phase A for different modulating frequencies are obtained from the oscilloscope. The waveforms of phase voltages for phase A for different frequencies are also obtained from the oscilloscope. The modulation index is kept constant (M = 0.7) throughout the test. The testing can also be carried out for different modulation indices. The photographs of the PWM patterns, waveforms of the phase current of phase A and the phase voltage of phase A are taken with a camera and are shown in Figs. 5.33 – 5.46. The pictures of the experimental set up in the laboratory are shown in Figs. 5.47 – 5.52.

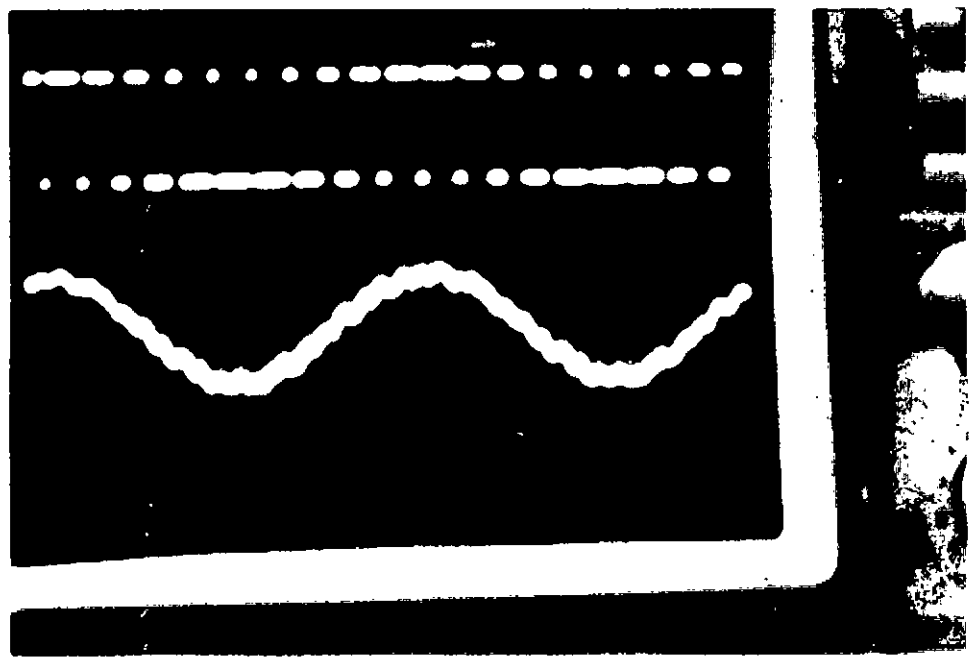


Fig. 5.33 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 100 Hz, f_c=1000Hz and M = 0.7

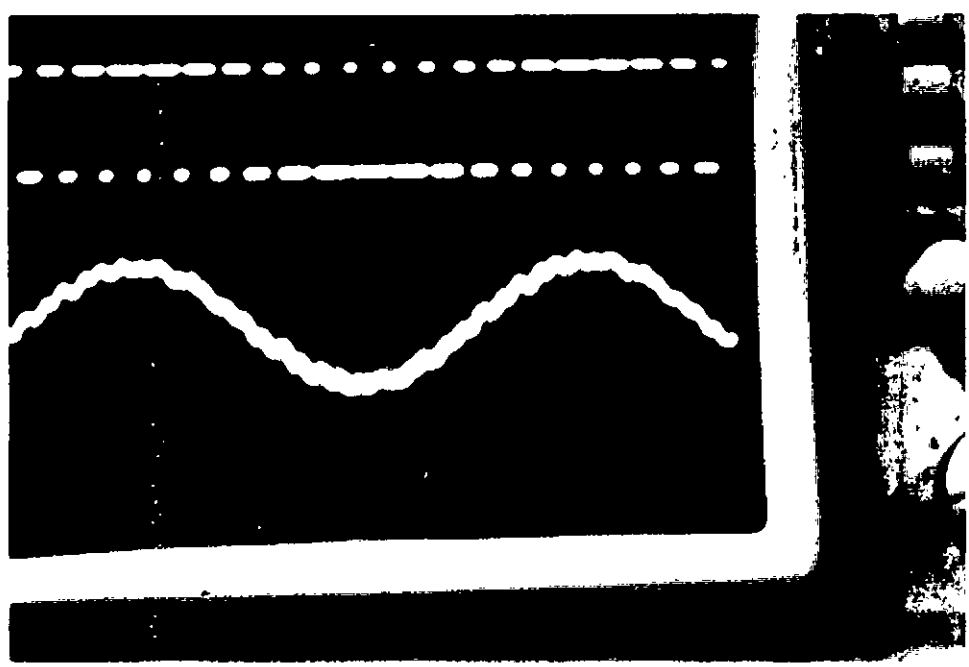


Fig. 5.34 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 83 Hz, f_c=1000Hz and M = 0.7

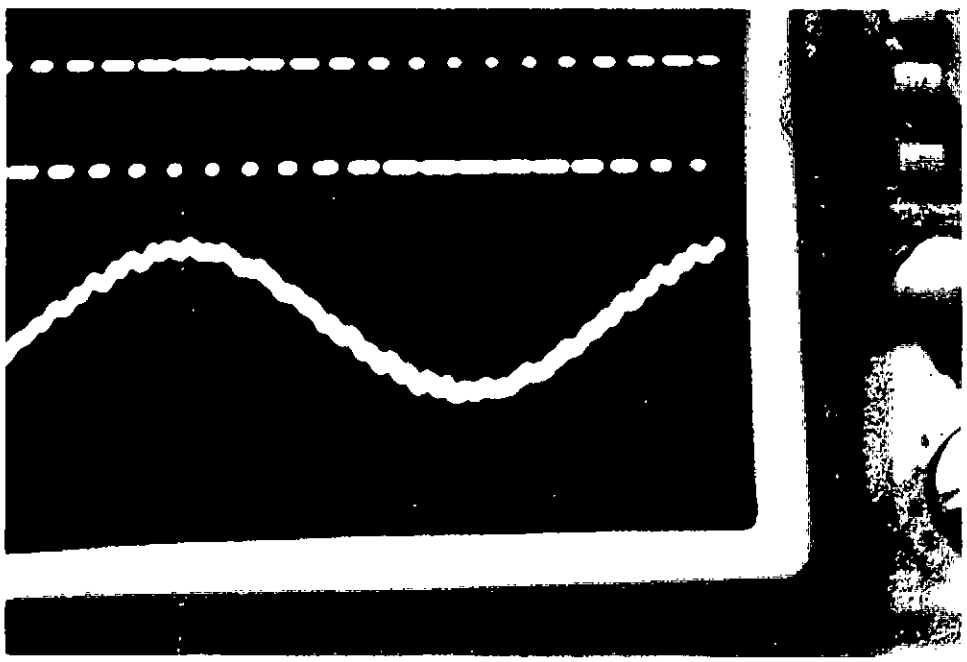


Fig. 5.35 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 67 Hz, f_c=1000Hz and M = 0.7

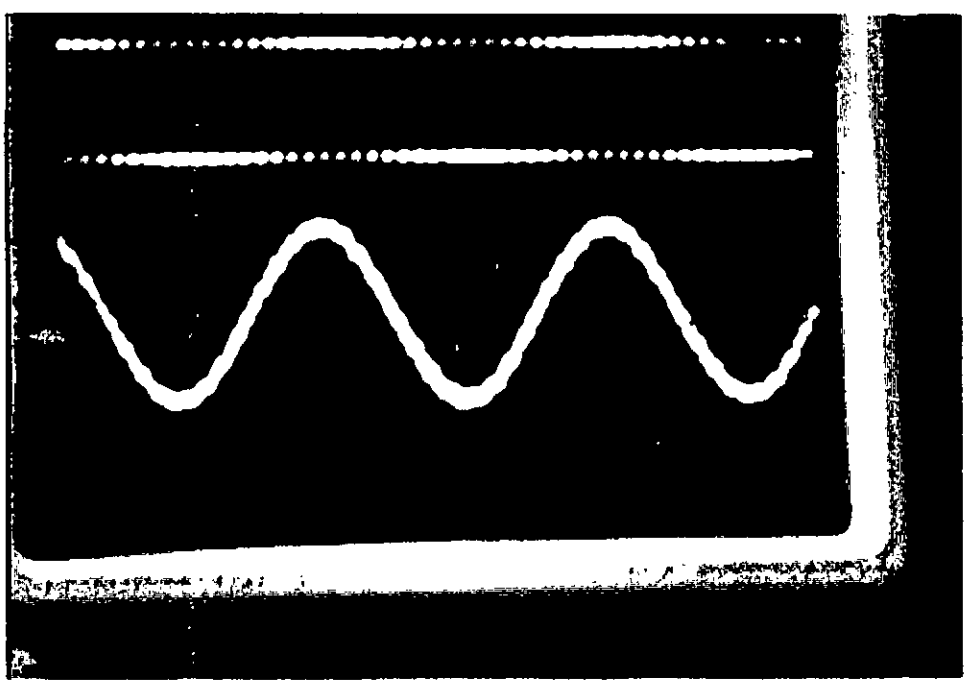


Fig. 5.36 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 56 Hz, f_c=1000Hz and M = 0.7

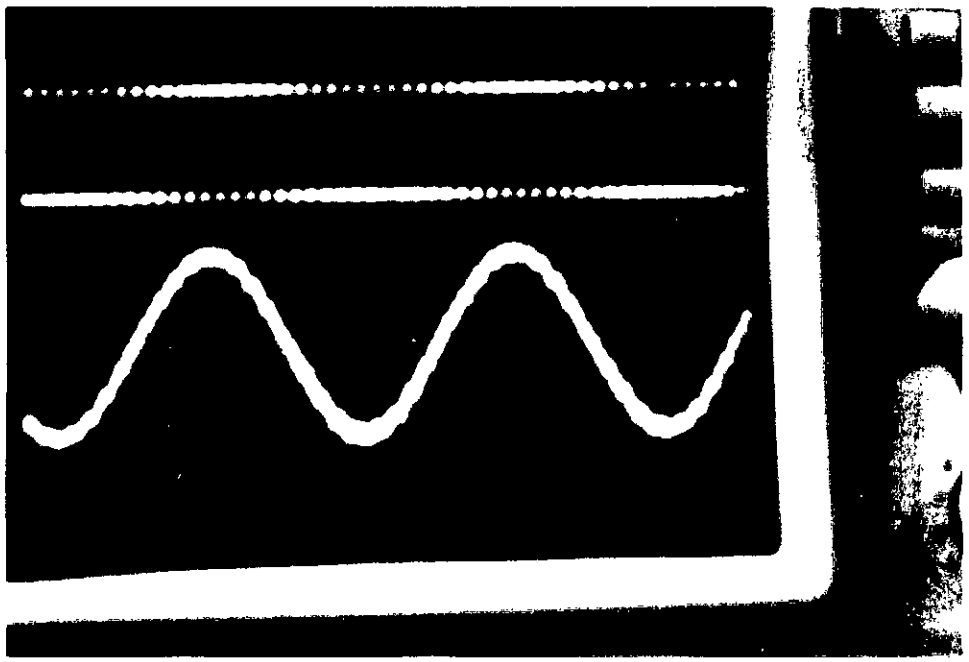


Fig. 5.37 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 50 Hz, f_c=1000Hz and M = 0.7

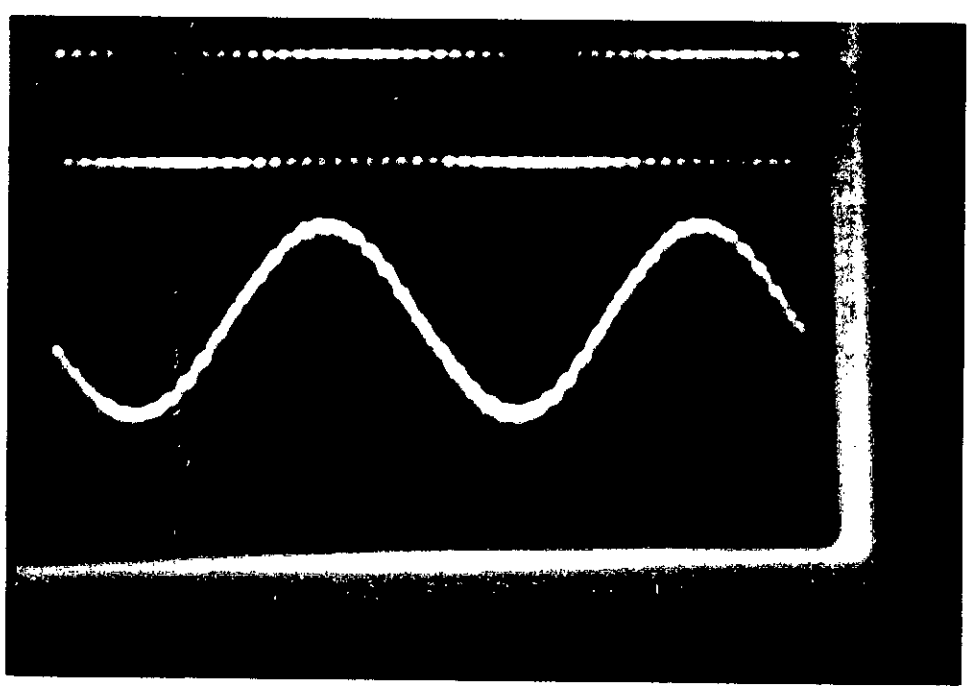


Fig. 5.38 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 42 Hz, f_c=1000Hz and M = 0.7

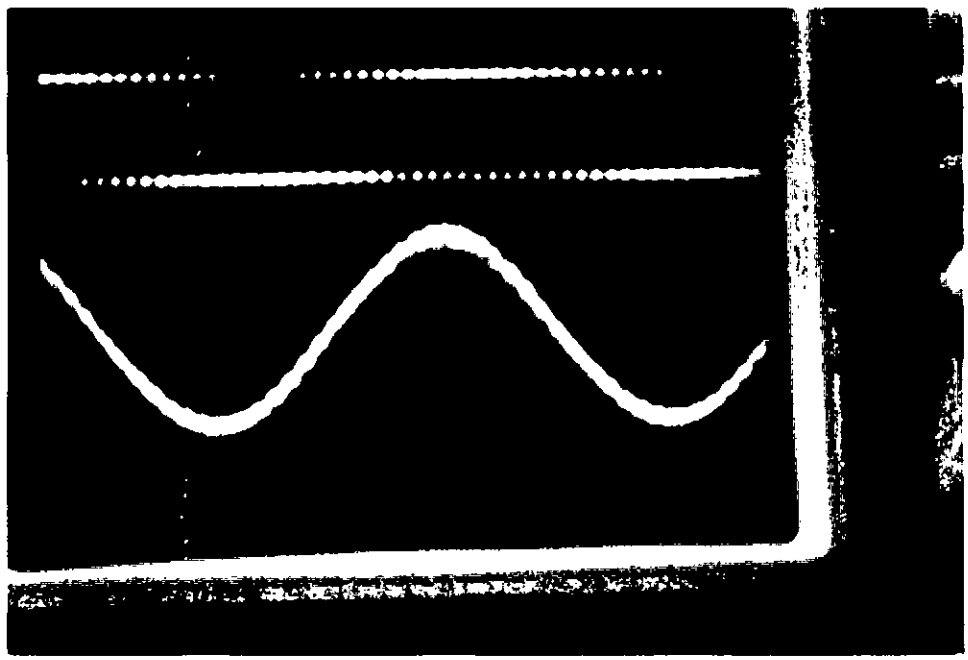


Fig. 5.39 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 33 Hz, f_c=1000Hz and M = 0.7

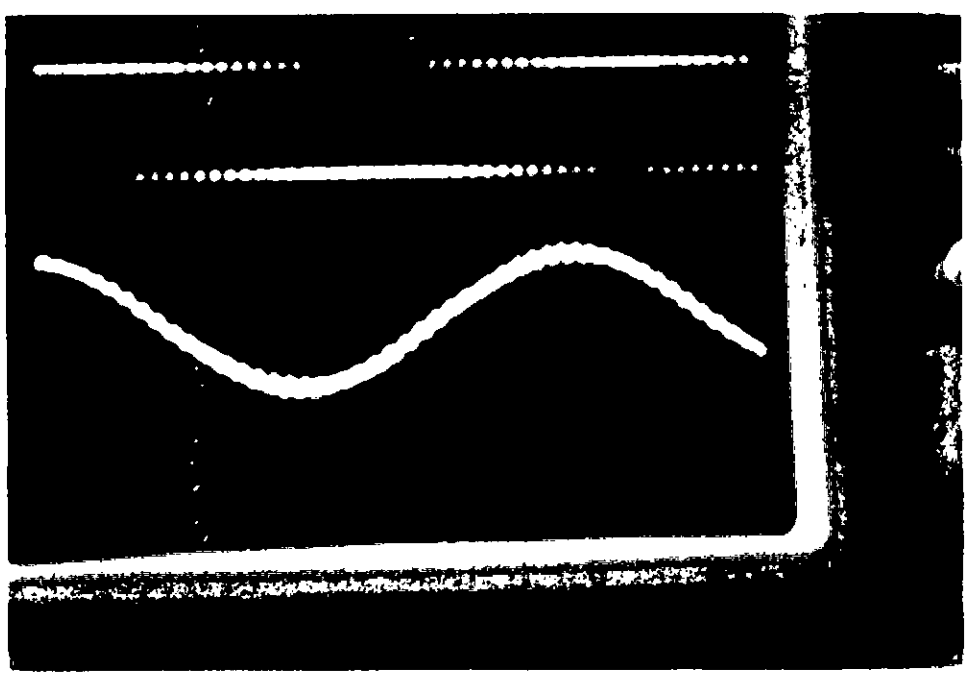


Fig. 5.40 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 28 Hz, f_c=1000Hz and M = 0.7

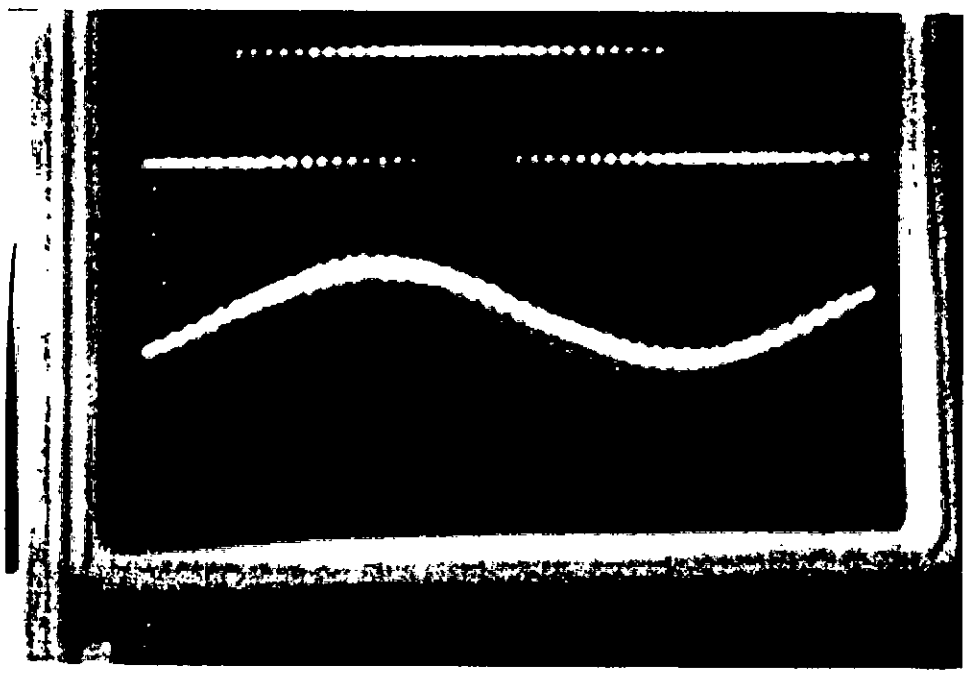


Fig. 5.41 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 25 Hz, f_c=1000Hz and M = 0.7

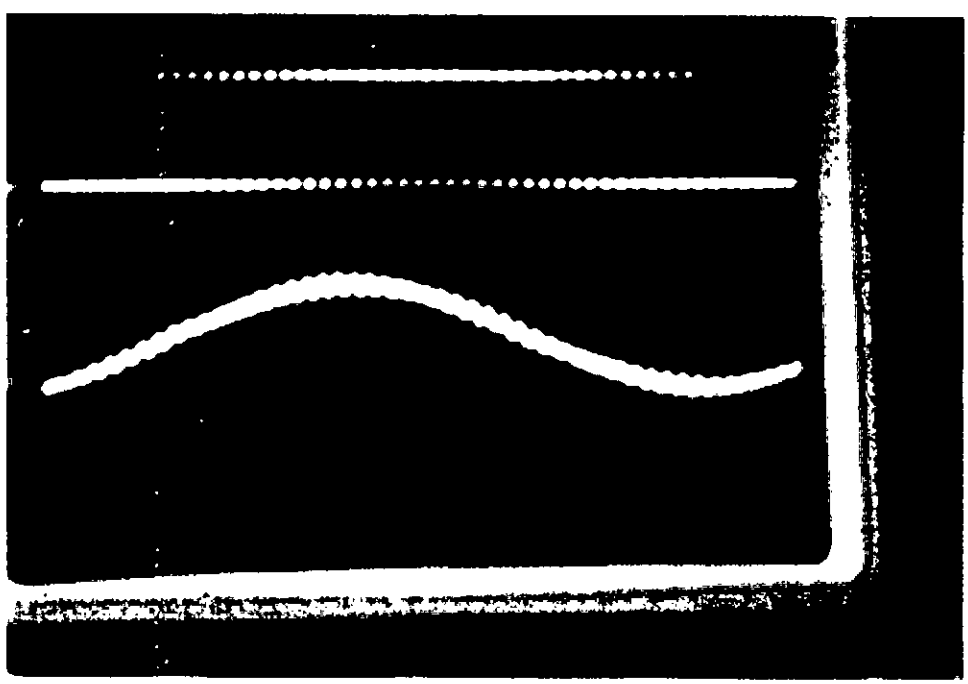


Fig. 5.42 PWM patterns for switch S_1 of phase A and phase current of phase A for f = 22 Hz, f_c=1000Hz and M = 0.7

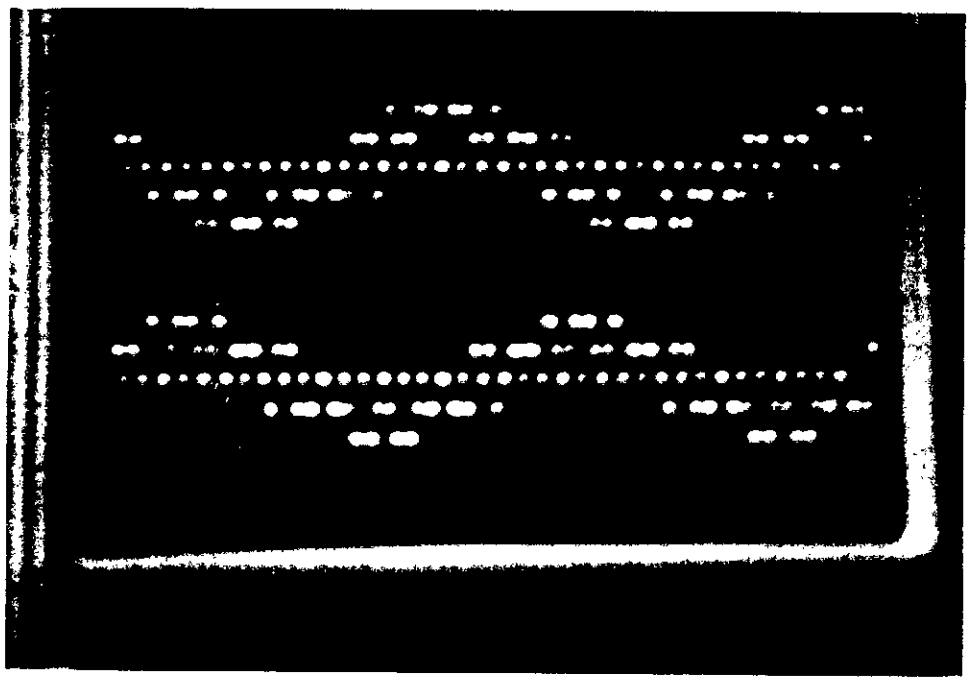


Fig. 5.43 Phase voltage waveform of phase A for f = 100 Hz, f_c=1000Hz and M = 0.7

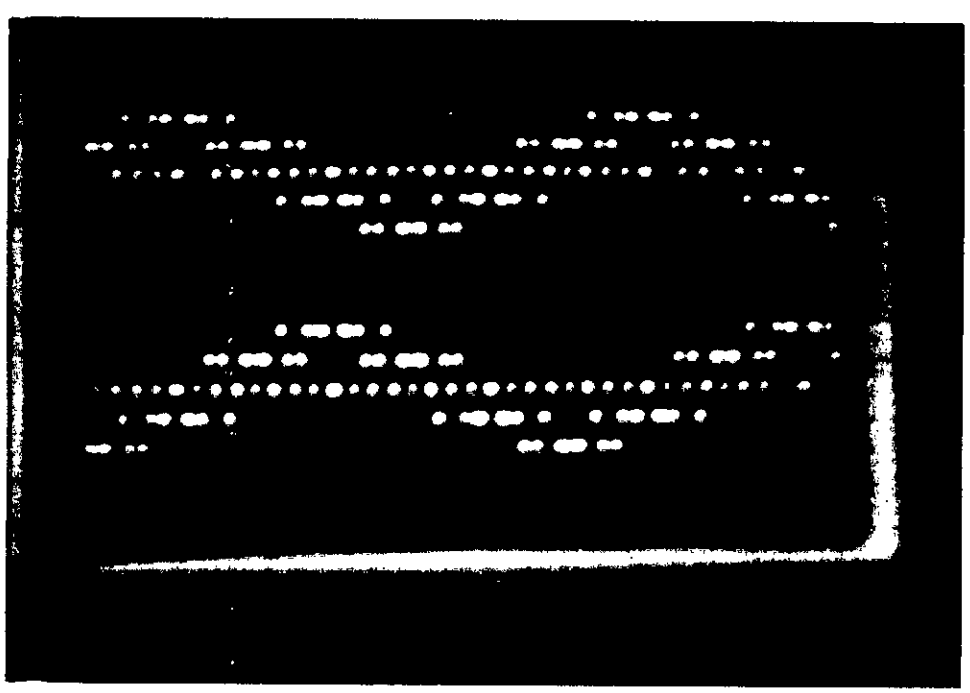


Fig. 5.44 Phase voltage waveform for phase A for f = 83 Hz, f_c=1000Hz and M = 0.7

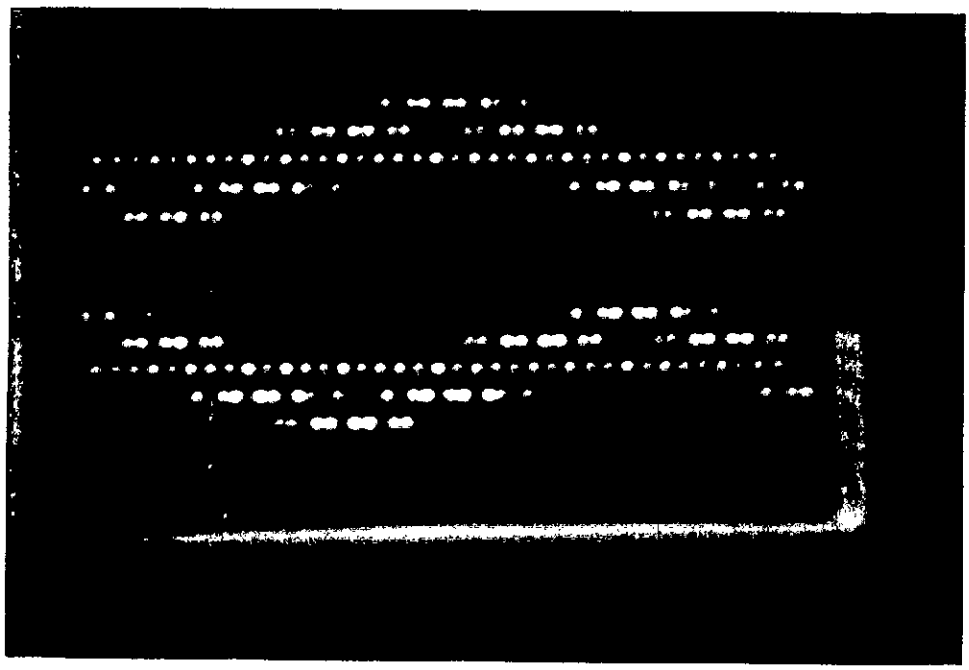


Fig. 5.45 Phase voltage waveform for phase A for f = 67 Hz, f_c=1000Hz and M = 0.7

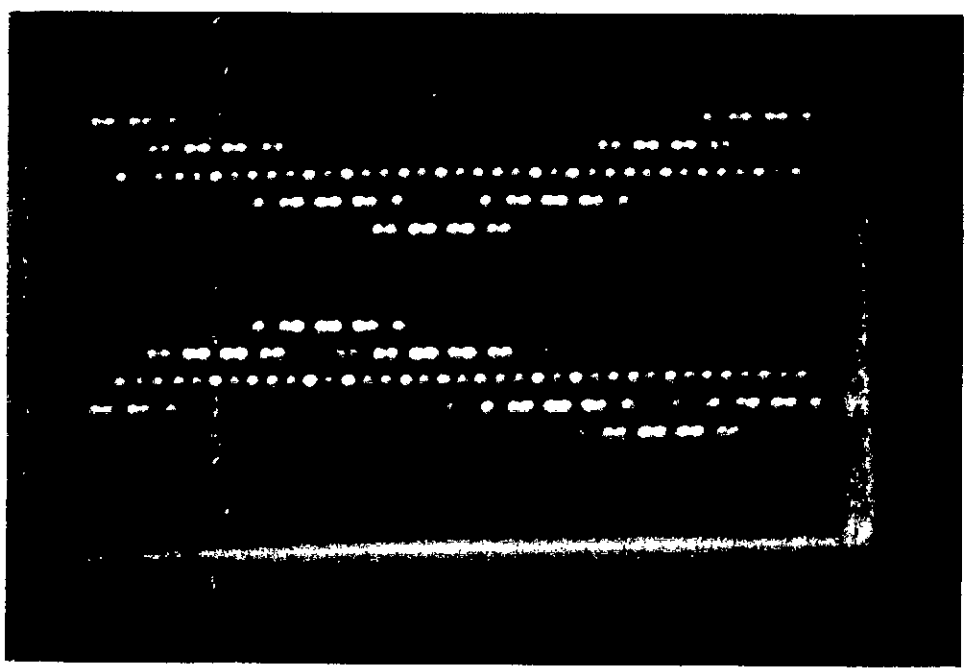


Fig. 5.46 Phase voltage waveform for phase A for f = 56 Hz, f_c=1000Hz and M = 0.7

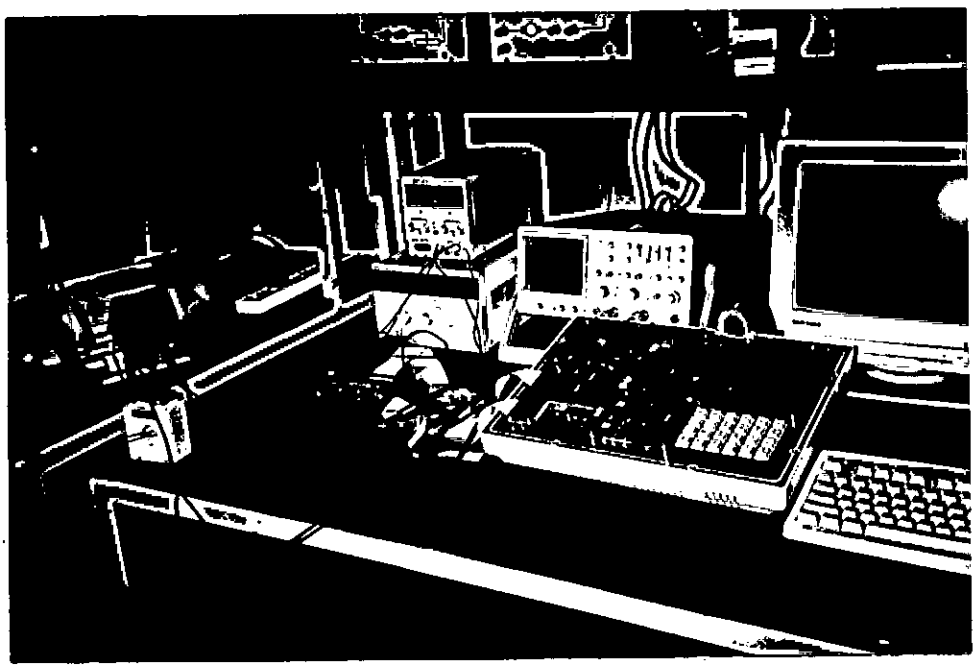


Fig. 5.47 Experimental set up (VIEW1)

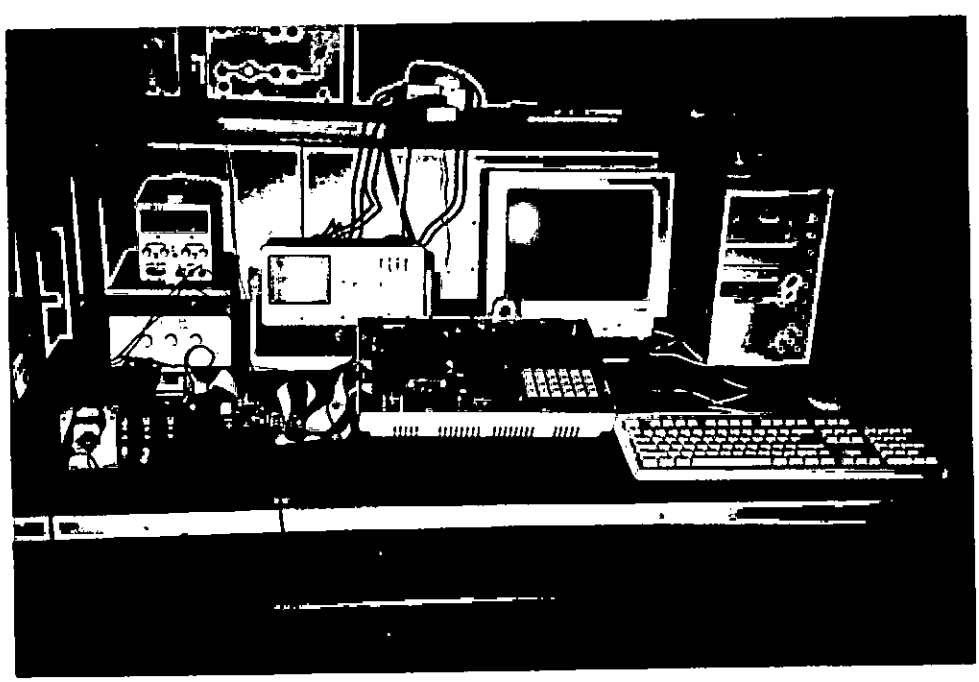


Fig. 5.48 Experimental set up (VIEW2)

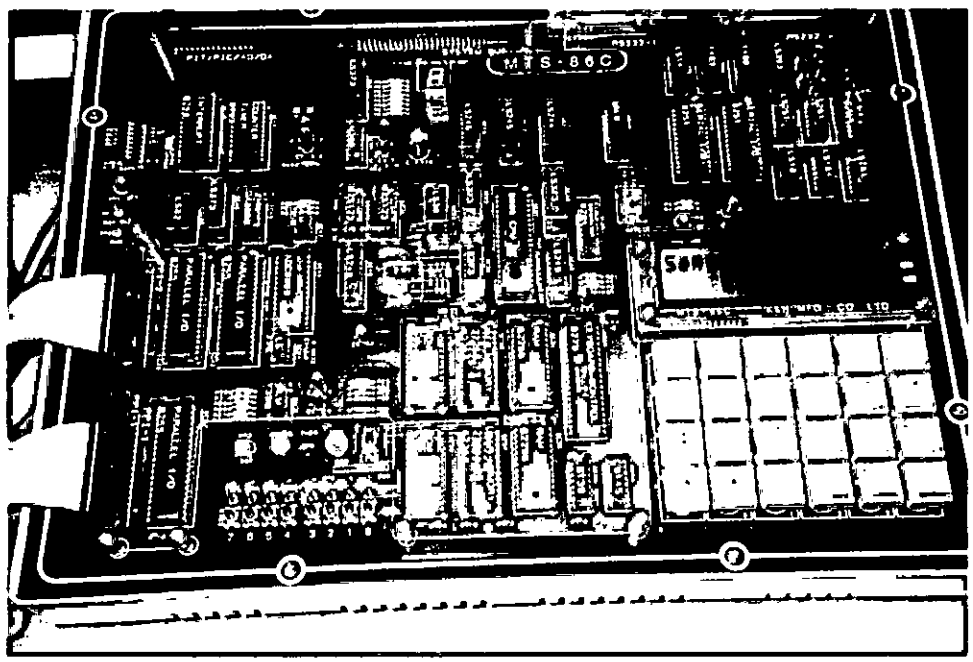


Fig. 5.49 Microprocessor kit (MTS-86C)

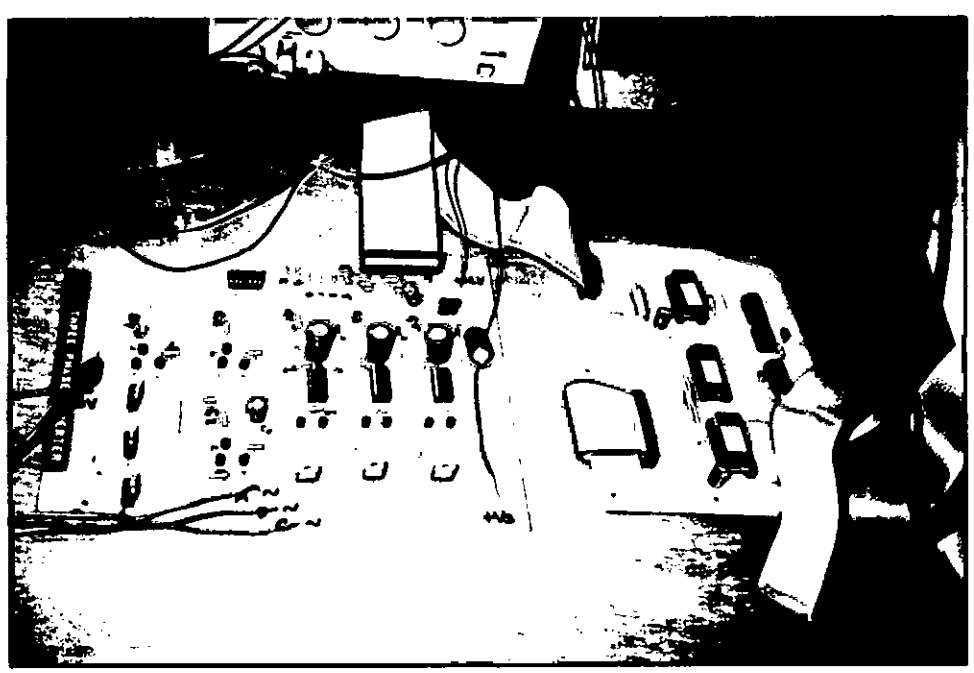


Fig. 5.50 PWM scheme circuit board and three phase inverter

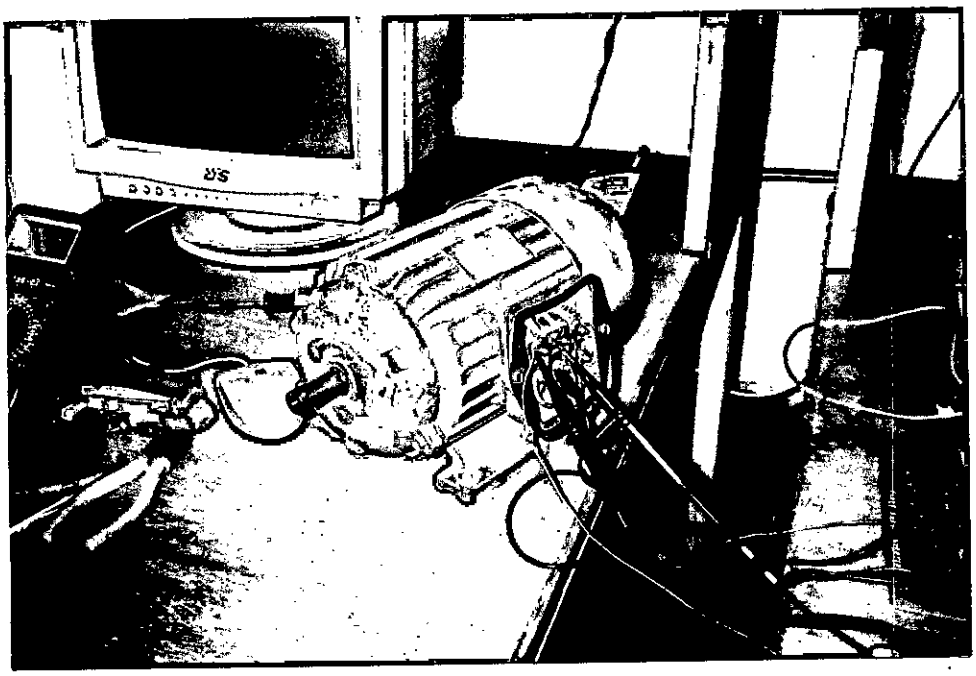


Fig. 5.51 Induction motor with external connections (VIEW1)

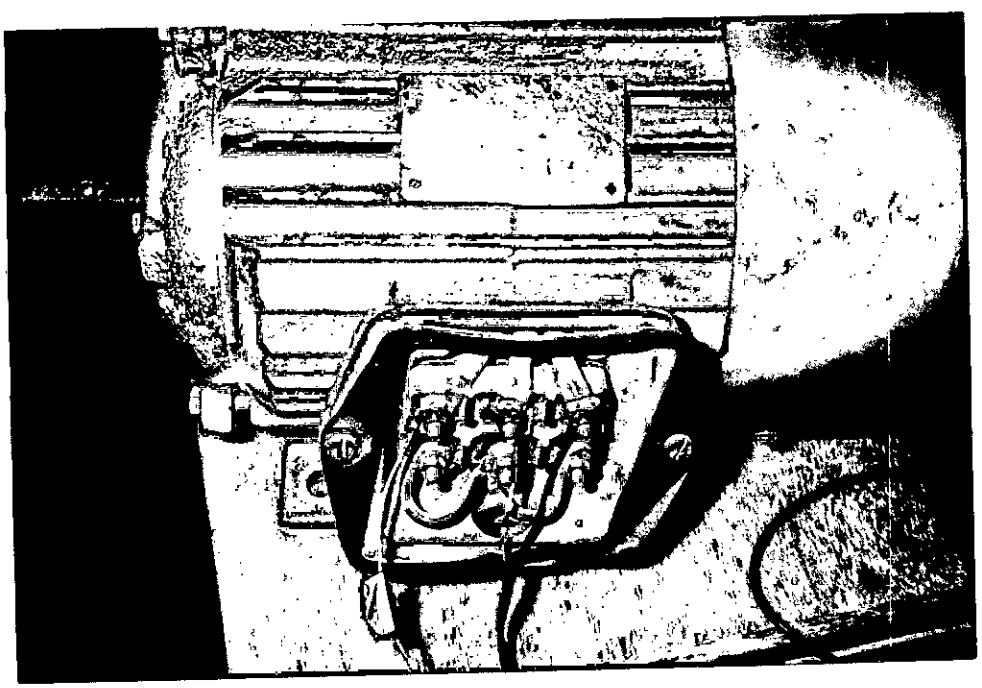


Fig. 5.52 Induction motor with external connections (VIEW2)

In Figs.5.33 – 5.42, it is seen that the phase current is sinusoidal in shape and some harmonics are present in it. The harmonics are of negligible amount. In Figs.5.43 – 5.46, the phase voltages of phase A for different modulating frequencies are shown. It is mentionable that the variation of modulating frequency and modulation index can be done on line, i.e., keeping the motor in running condition. It is observed that the performance of the scheme is not hampered if the variation of the frequency and modulation index is carried out on line.

Chapter 6

Conclusions

6.1 Conclusions

A new Look Up Table (LUT) based hybrid PWM scheme for voltage source inverters is presented in this thesis. Other microcomputer-based PWM schemes generate PWM patterns using microprocessor and Programmable Timers. Such schemes require high-speed processors for generating PWM patterns for wide operating range. Embedded schemes cost much as they use high-speed processors and huge memories. For ac drives, the processor is also used for computation of other characteristic parameters. The processor gets little time for the PWM patterns computation and hence limits the operating range. Since, both processor and LUT have been involved in the hybrid technique, the overall operating range of the PWM scheme is improved. The microprocessor is used for computing duty cycles of the PWM pattern once in each carrier period. Thus, it gets enough time up to the next carrier cycle for computing other parameters of the drive system. In this way the overall operating range of the PWM scheme is increased which is greater than the existing techniques.

Although LUT based schemes require large memory for storing PWM patterns for an inverter operating in wide frequency/modulation range, the memory requirement is reduced in this hybrid scheme since the same duty cycle is shared for different frequencies and different modulation indices.

If the equation of pulse width contains floating-point type variables then MMX processors having built in math coprocessor are needed for the real time implementation of the PWM scheme. However, using such processors make the scheme costly. In this thesis, the sampled version of the model for PWM pattern calculation is developed. The equation of pulse width of PWM pulses is simplified so that the equation contain only integer type variables. Thus, it has become possible to implement the scheme in real time with normal low cost processors that can perform only integer type calculations. In this way, the scheme has become cost effective.

The PWM patterns are generated theoretically using MATLAB program. The corresponding frequency spectrums are also obtained. It is found that the width of the simulated PWM patterns vary sinusoidally, which is a desirable feature. The dc value in the frequency

spectrum is negligible. Different harmonic components, especially the second harmonic components are of negligible value, which is a desirable feature.

The Look Up Table (LUT) for the patterns is also formed using MATLAB program and stored in the EPROMs. Three 32k EPROMs are used for three phases. The EPROMs are of low cost. Thus, the cost of the scheme is reduced.

The simulated PWM patterns and the real time PWM patterns are nearly similar except that some glitches appeared in the real time patterns. The effect of these glitches is very negligible. Comparing the spectrums of the simulated patterns and real time patterns we find that both set of spectrums overlaps each other with some discrepancies at higher frequencies. We can neglect these little discrepancies. In the spectrums of the real time patterns, very negligible dc values in the fundamental and negligible second harmonic components were found.

The scheme is tested on a three-phase prototype inverter. A cage type induction motor is connected to the inverter as three phase load. The phase currents of phase A for different frequencies are obtained directly from the oscilloscope that are presented in the thesis. The phase currents are of sinusoidal shape containing negligible harmonics. The phase voltages of phase A for different frequencies have also been obtained from the oscilloscope and found to be of desirable shape. The currents and voltages can also be controlled by varying the modulation index. The variation of the frequency and modulation index is done from the microprocessor using assembly language program. It is of great advantage that the variation of frequency and modulation index can be done on line, that is, keeping the motor in running condition. The motor need not be switched off to change frequency or modulation index.

6.2 Future Works

Having reviewed the contributions made in the thesis, there is scope of extending this work in future to meet other goals. In this scheme, the variation of modulation index can be done continuously whereas the modulation frequency can be varied discretely. For smooth variation of speed of a drive, the frequency may need continuous adjustments in some applications. Since the proposed scheme can generate high-resolution PWM patterns, further investigation may be made in future to apply it for continuous frequency variation that will increase its application area.

References

[1] S. R. Bowes, "Advanced Regular Sampled PWM Control Techniques for Drives and Static Power Converters," *IEEE Transactions on Industrial Electronics,* Vol. 42, No. 4, August 1995, pp. 367-373.

[2] S. R. Bowes, "New sinusoidal pulsewidth-modulated inverter," *IEE Proc.*, Vol. 122, No. 11, pp. 1279-1285, 1975.

[3] S. R. Bowes and M. J. Mount, "Microprocessor control of PWM inverters," *Proc. IEE, B-Elect. Power Appl.,* vol. 128, no. 6, pp. 293-305, 1981.

[4] P. D. Ziogas, "The delta modulation techniques in static PWM inverters," *IEEE Transactions on Industrial Applications,* March/April, 1981, pp. 199-204.

[5] K. M. Rahman, M. A. Choudhury, M. R. Khan and M. A. Rahman, "Dual slope integrator type delta modulator for high performance voltage source inverters," Power Conversion Conference (PCC)-Nagaoka'97, Japan, pp. 451-455, 1997.

[6] H.S patel and R.G.Hoft, "Generalized techniques of harmonic elimination and voltage control in thyristor inverters," *IEEE Transactions on Industry Applications,* pp. 310-317, 1973.

[7] S. R. Bowes, "Novel Real-Time Harmonic Minimized PWM Control for Drives and Static Power Converters," *IEEE Transactions on Power Electronics*, Vol. 9, No. 3, May 1994, pp 256-362.

[8] S. R. Bowes, "Regular Sampled harmonic Elimination PWM control of Inverter Drives," *IEEE Transactions on Power Electronics,* Vol. 10, No. 5, September 1995, pp. 521-531.

[9] K.S. Rajashekara, Joseph Vithayathil, "Microprocessor based Sinusoidal PWM Inverter by DMA transfer," *IEEE Transactions on Industry Applications,* vol. IA-19, no. 2, pp. 187-194, March/April 1983.

[10] Schonung, A., and Stemmler, H.: "Static frequency changers with subharmonic control in conjunction with reversible variable speed AC drives", *Brown Boveri Rev.*, 1964, 51, pp. 555-577

[11] S. R. Bowes, "Pulsewidth modulation," British patent application no. 10753/75, Mar. 1975.

[12] S. R. Bowes and R. R. Clements, "Computer aided design of PWM inverter systems," *Proc. IEE, B-Elect. Power Appl.,* vol. 129, no. 1, pp. 1-17, 1982.

[13] S. R. Bowes and R. R. Clemenrs, "Digital computer simulation of variable speed PWM inverter-machine drives," *Proc. IEE, B-Elect. Power Appl.,* vol. 130, no. 3, pp. 149-160, 1983.

[14] S. R. Bowes and J. C. Clare, "Steady state performance of PWM inverter drives," *Proc. IEE, B-Elect. Power Appl.,* vol. 130, no. 4, pp. 229-244, 1983.

[15] S. R. Bowes and T. Davies, "Microprocessor-based development system for PWM variable-speed drives," *Proc. IEE, B-Elect. Power Appl.,* vol. 132, no. 1, pp. 18-45, 1985.

[16] S. R. Bowes and A. Midoun, "Suboptimal switching strategies for microprocessor-controlled inverter drives," *Proc. IEE, B-Elect. Power Appl.,* vol. 132, no. 3, pp. 133-148, 1985.

[17] S. R. Bowes and A. Midoun, "A new PWM switching strategy for microprocessor-controlled inverter drives," *Proc. IEE, B-Elect. Power Appl.,* vol. 133, no. 4, pp. 237-254, 1986.

[18] S. R. Bowes, "An Algebraic Algorithm for Microcomputer-Based (Direct) Inverter Pulsewidth Modulation," *IEEE Transactions on Industry Applications,* Vol. 24, No. 6, Nov/Dec. 1988.

[19] K. M. Rahman, "PC based Regular Sampled Pulse Width Modulator For Inverters And AC Drives," *Proceedings of the International Conference on Computer and Information Technology,* (ICCIT'99), SUST, 3-5 Dec. '99.

[20] M. H. Rashid, "Power Electronics Circuits, Devices and Applications" pp. 381-390.

[21] S. R. Bowes and R. J. Bullough, "Harmonic minimization in microprocessor controlled PWM current-fed inverter drives," *Proc. IEE, B – Elect. Power Appl.,* vol. 134, no. 2, pp – 1 – 17, 1987

Appendix

Program 1: Matlab Program to Form LUT and Generate the Theoretical PWM Patterns and Spectrums.

```
%Patterns for New Hybrid PWM using LUT
echo off all;
clear all;
Nmax=254;
for q=1:127,
   for n=1:Nmax,
      if n < (Nmax/2-q+1) | n > (Nmax/2+q-1),
         Pat(q,n) = 2;
      else
         Pat(q,n) = 1;
      end
   end
end

%%%%%%%%%
f=100;
fc=1600;
fb=50.0;
Vs=800; %Input dc voltage
M=120;
N=round(fc/f);
n=1:N;
%phase=3*(2.0*pi/N)/4;
phase=0;
theta_n=2*pi/N:2*pi/N:2*pi;
Sna=round(126*sin(theta_n-phase));
```

```
Snb=round(126*sin(theta_n-2*pi/3-phase));
Snc=round(126*sin(theta_n-4*pi/3-phase));

Duty_a=128+ Sna*M/128;
Duty_b=128 + Snb*M/128;
Duty_c=128+ Snc*M/128;
Da=round(Duty_a/2);
Db=round(Duty_b/2);
Dc=round(Duty_c/2);
PWMa=Pat(Da,:);
PWMb=Pat(Db,:);
PWMc=Pat(Dc,:);
r=0;
for k=1:length(Da),
   for n=1:Nmax,
      r=r+1;
      PWM(r)=PWMa(k,n);
      if PWM(r)==2,
         PWM_A(r)=-1;
      else
         PWM_A(r)=PWM(r);
      end
   end
end
r=0;
for k=1:length(Db),
   for n=1:Nmax,
      r=r+1;
      PWM(r)=PWMb(k,n);
      if PWM(r)==2,
         PWM_B(r)=-1;
      else
         PWM_B(r)=PWM(r);
      end
```

```
   end
end
r=0;
for k=1:length(Dc),
   for n=1:Nmax,
      r=r+1;
      PWM(r)=PWMc(k,n);
      if PWM(r)==2,
         PWM_C(r)=-1;
      else
         PWM_C(r)=PWM(r);
      end
   end
end
r=0;
p=0;
for k=1:length(Da)*Nmax,
      r=r+1;
van(r)=PWM_A(r)/3-(PWM_B(r)+PWM_C(r))/6;
vbn(r)=PWM_B(r)/3-(PWM_A(r)+PWM_C(r))/6;
vcn(r)=PWM_C(r)/3-(PWM_B(r)+PWM_A(r))/6;
p=p+1;
vx(p)=van(r);
end

Vload = abs(2.0*Vs*fft(vx)/p);

sum=0;
 for n=3:p/2,
    sum=sum+Vload(n)*Vload(n)/(n*n*n*n);
 end
 DF=sqrt(sum)/(Vload(2));

r=0;
```

```
tau=1/(length(Da)*Nmax*f);
%for k=1:length(Da),
 %  for n=1:Nmax,
  %     r=r+1;
%       time=r*tau;
   %    PWM_AB(r)=(PWM_A(r)-PWM_B(r))/2;
   %end
   %end
   r=1:length(Da)*Nmax;
   PWM_AB(r)=(PWM_A(r)-PWM_B(r))/2;
   time=r*tau;
spectrumAB=abs(2.0*fft(PWM_AB)/length(PWM_AB));

%plot(time,PWM_AB);
%axis([0 (1/f) -1.2 1.2]);
%title('PWM pattern for V_{AB}(f=100Hz, M=0.94, f_(c)=1600Hz)');
%xlabel('Time(seconds)');
%ylabel('Magnitude p.u.');

plot(spectrumAB);
axis([0 50 0 1]);
title('Frequency spectrum of PWM waveform for V_{AB}(f=100Hz,
M=0.94, f_{c}=1600Hz)');
xlabel('Frequency(as multiple of fundamental)');
ylabel('Magnitude p.u.');

%plot(PWM_A);
%axis([0 length(PWM_A) -1.2 1.2]);
%title('PWM pattern for V_{AB}(f=100Hz, M=0.4, f_{c}=1600Hz)');
%xlabel('Time(seconds)');
%ylabel('Magnitude p.u.');

%spectrumA=abs(2.0*fft(PWM_A)/length(PWM_A));
%subplot(2,1,1),plot(spectrumA);
```

```
%axis([0 2*N 0 1]);
%subplot(2,1,2),plot(PWM_A);
%axis([0 length(PWM_A) -1.2 1.2]);

%spectrumB=abs(2.0*fft(PWM_B)/length(PWM_B));
%subplot(2,1,1),plot(spectrumB);
%axis([0 2*N 0 1]);
%subplot(2,1,2),plot(PWM_B);
%axis([0 length(PWM_B) -1.2 1.2]);

%spectrumC=abs(2.0*fft(PWM_C)/length(PWM_C));
%subplot(2,1,1),plot(spectrumC);
%axis([0 2*N 0 1]);
%subplot(2,1,2),plot(PWM_C);
%axis([0 length(PWM_C) -1.2 1.2]);

%subplot(3,1,1),plot(PWM_A);
%axis([0 length(PWM_A) -1.2 1.2]);
%subplot(3,1,2),plot(PWM_B);
%axis([0 length(PWM_B) -1.2 1.2]);
%subplot(3,1,3),plot(PWM_C);
%axis([0 length(PWM_C) -1.2 1.2]);
%plot(vcn);
%axis([0 length(van) -1 1]);

Da1=Da';
Db1=Db';
Dc1=Dc';
y=[Da1,Db1,Dc1];
y1=[Sna',Snb',Snc'];
fid=fopen('pwm.txt','w');
%fid1=fopen('sine.txt','w');
%fid=fopen('spectrum5e2.txt','w');
```

```
for n=1:length(Da),
   fprintf(fid,'%2d \t %2d \t %2d\n',y(n,:));
    %fprintf(fid1,'%2d \t %2d \t %2d\n',y1(n,:));
end
fclose(fid);
fid1 = fopen('c:\aaqz\pwm.txt','r');
A=fscanf(fid1,'%5d \t');
fclose(fid1);

%for n=1:length(spectrumAB),
%  fprintf(fid,'%2d \n',spectrumAB(n));
%end
%fclose(fid);
%%% Set break point to see PWM pattern and spectrum
```

PROGRAM 2: Assembly Language Program for Real Time Implementation of Hybrid LUT PWM Scheme.

```
; ASSEMBLY LANGUAGE PROGRAM FOR HYBRID LUT PWM

PORT1A   EQU     0FFF9H
PORT1B   EQU     0FFFBH
PORT1C   EQU     0FFFDH
CNTL1    EQU     0FFFFH

PORT2A   EQU     0FFF8H
PORT2B   EQU     0FFFAH
PORT2C   EQU     0FFFCH
CNTL2    EQU     0FFFEH

PORT3A   EQU     3FD0H
```

```
PORT3B  EQU     3FD2H
PORT3C  EQU     3FD4H
CNTL3   EQU     3FD6H
FND     EQU     3FF0H


CODE    SEGMENT
        ASSUME  CS:CODE, DS:CODE, SS:CODE, ES:CODE
        ORG     0H
        MOV     AX,CS
        MOV     DS,AX

        MOV     AL,90H    ;CONFIGURE PORT3 A INPUT B,C = OUTPUT
        MOV     DX,CNTL3
        OUT     DX,AL

        MOV     AL,80H    ;CONFIGURE PORT2 A,B,C = OUTPUT
        MOV     DX,CNTL2
        OUT     DX,AL

        MOV     AL,90H    ; CONFIGURE PORT1 A=INPUT
        MOV     DX,CNTL1
        OUT     DX,AL
        MOV     CL,00H
        MOV     AX,11
        MOV     SI,AX

J1:     MOV     BL,PULSES
        MOV     AX,360
        DIV     BL
        MOV     STEP,AL
        DEC     AL
        MOV     AH,0
        MOV     DI,AX    ; INITIAL TAG step
```

```
J2:     MOV     DX,PORT1A ; RIPPLE CARRY DETECTOR
        IN      AL,DX
        AND     AL,1
        CMP     AL,1
        JNZ     J2

        MOV     DX,PORT2A
        MOV     AL,DATA1
        OUT     DX,AL

        MOV     DX,PORT2B
        MOV     AL,DATA2
        OUT     DX,AL

        MOV     DX,PORT2C
        MOV     AL,DATA3
        OUT     DX,AL

        ;;;;;;;;;;;;; SCAN MODULATION INCREASE OR DECREASE

        CMP     BL,1
        JNZ     NOCHANGE
        MOV     AL,COUNTS
        CMP     AL,10
        JNZ     NOCHANGE
        MOV     AL,0
        MOV     COUNTS,AL    ; RESET COUNTS = 0
        MOV     DX,PORT3A
        IN      AL,DX
        MOV     CL,AL
        AND     AL,01
        CMP     AL,01
        JZ     NEXT
```

```
        MOV     DL,M
        INC     DL      ;INCREASE M
        MOV     M,DL

NEXT:   MOV     AL,CL
        AND     AL,02
        CMP     AL,02
        JZ      PULSEADJ
        MOV     DL,M
        DEC     DL      ;DECREASE M
        MOV     M,DL

;NEW LINES FOR FREQUENCY

PULSEADJ:       MOV     AL,CL
        AND     AL,04
        CMP     AL,04
        JZ      NEXTFREQ
        INC     SI
        CMP     SI,14
        JNZ     NOSETSI
        MOV     SI,13

NEXTFREQ: MOV   AL,CL
        AND     AL,08
        CMP     AL,08
        JZ      NOCHANGE
        DEC     SI
        CMP     SI,0
        JNZ     NOSETSI
        MOV     SI,1

NOSETSI: MOV     AL,PULSENO[SI]
        MOV     PULSES,AL
```

```
;;; CODES FOR DISPLAYING FREQUENCY NO.
        MOV     CL,BL
        MOV     BX,OFFSET FONT
        MOV     AX,SI
        XLAT
        MOV     DX,FND
        OUT     DX,AL
        MOV     BL,CL

NOCHANGE:  MOV     DX,0
        ;;;;;;;;;;;
        MOV     AL,SINEA[DI]
        MOV     AH,0
        MOV     CL,M
        IMUL    CL
        MOV     CL,80H
        IDIV    CL
        CBW
        ADD     AX,80H
        MOV     CL,02H
        DIV     CL
        MOV     DATA1,AL
        ;;;;;;;;;;;;;;;;

        MOV     AL,SINEB[DI]
        MOV     AH,0
        MOV     CL,M
        IMUL    CL
        MOV     CL,80H
        IDIV    CL
        CBW
        ADD     AX,80H
        MOV     CL,02H
```

```
        DIV     CL
        MOV     DATA2,AL
        ;;;;;;;;;;;;;;;;

        MOV     AL,SINEC[DI]
        MOV     AH,0
        MOV     CL,M
        IMUL    CL
        MOV     CL,80H
        IDIV    CL
        CBW
        ADD     AX,80H
        MOV     CL,02H
        DIV     CL
        MOV     DATA3,AL
        ;;;;;;;;;;;;;;;;
        XOR     AX,AX
        MOV     AL,STEP
        ADD     DI,AX
        DEC     BL
        JNZ     J3

        ; END OF ONE FUNDAMENTAL CYCLE

        MOV     AL,COUNTS
        INC     AL
        MOV     COUNTS,AL
        JMP     J1
J3:     JMP     J2

 ;; STARTING 1 DEGREE, INCREMENT 1 DEGREE, TOTAL 360 ANGLES
PULSENO  DB  0, 90, 72, 60, 45, 40, 36, 30, 24, 20, 18, 15, 12,
10 ; 14 FREQS.
```

```
SINEA   DB   2,4,7,9,11,13,15,18,20,22,24,26,28,30,33,35,37,39
        DB   41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,70
        DB   72,74,76,78,79,81,83,84,86,88,89,91,92,94,95,97
        DB   98,99,101,102,103,104,106,107,108,109,110,111,112
        DB   113,114,115,116,117,118,118,119,120,120,121,122
        DB   122,123,123,124,124,124,125,125,125,126,126,126
        DB   126,126,126,126,126,126,126,126,125,125,125,124
        DB   124,124,123,123,122,122,121,120,120,119,118,118
        DB   117,116,115,114,113,112,111,110,109,108,107,106
        DB   104,103,102,101,99,98,97,95,94,92,91,89,88,86,84
        DB   83,81,79,78,76,74,72,70,69,67,65,63,61,59,57,55
        DB   53,51,49,47,45,43,41,39,37,35,33,30,28,26,24,22
        DB   20,18,15,13,11,9,7,4,2,0,-2,-4,-7,-9,-11,-13,-15
        DB   -18,-20,-22,-24,-26,-28,-30,-33,-35,-37,-39,-41
        DB   -43,-45,-47,-49,-51,-53,-55,-57,-59,-61,-63,-65
        DB   -67,-69,-70,-72,-74,-76,-78,-79,-81,-83,-84,-86
        DB   -88,-89,-91,-92,-94,-95,-97,-98,- 99,-101,-102
        DB   -103,-104,-106,-107,-108,-109,-110,-111,-112,-113
        DB   -114,-115,-116,-117,-118,-118,-119,-120,-120,-121
        DB   -122,-122,-123,-123,-124,-124, -124,-125,-125
        DB   -125,-126,-126,-126,-126,-126,-126,-126,-126,-126
        DB   -126,-126,-125,-125,-125,-124,-124,-124,-123,-123
        DB   -122,-122,-121,-120,-120,-119,-118,-118,-117,-116
        DB   -115,-114,-113,-112,-111,-110,-109,-108,-107,-106
        DB   -104,-103,-102,-101,-99,-98,-97,-95,-94,-92,-91
        DB   -89,-88,-86,-84,-83,-81,-79,-78,-76,-74,-72,-70
        DB   -69,-67,-65,-63,-61,-59,-57,-55,-53,-51,-49,-47
        DB   -45,-43,-41,-39,-37,-35,-33,-30,-28,-26,-24,-22
        DB   -20,-18,-15,-13,-11,-9,-7,-4,-2,0

SINEB   DB   -110,-111,-112,-113,-114,-115,-116,-117,-118,-118
        DB   -119,-120,-120,-121,-122,-122,-123,-123,-124,-124
        DB   -124,-125,-125,-125,-126,-126,-126,-126,-126,-126
        DB   -126,-126,-126,-126,-126,-125,-125,-125,-124,-124
```

```
        DB   -124,-123,-123,-122,-122,-121,-120,-120,-119,-118
        DB   -118,-117,-116,-115,-114,-113,-112,-111,-110,-109
        DB   -108,-107,-106,-104,-103,-102,-101,-99,-98,-97
        DB   -95,-94,-92,-91,-89,-88,-86,-84,-83,-81,-79,-78
        DB   -76,-74,-72,-70,-69,-67,-65,-63,-61,-59,-57,-55
        DB   -53,-51,-49,-47,-45,-43,-41,-39,-37,-35,-33,-30
        DB   -28,-26,-24,-22,-20,-18,-15,-13,-11,-9,-7,-4,-2,0
        DB   2,4,7,9,11,13,15,18,20,22,24,26,28,30,33,35,37,39
        DB   41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,70,7
        DB   2,74,76,78,79,81,83,84,86,88,89,91,92,94,95,97,98
        DB   99,101,102,103,104,106,107,108,109,110,111,112,11
        DB   3,114,115,116,117,118,118,119,120,120,121,122,122
        DB   123,123,124,124,124,125,125,125,126,126,126,126
        DB   126,126,126,126,126,126,126,125,125,125,124,124
        DB   124,123,123,122,122,121,120,120,119,118,118,117
        DB   116,115,114,113,112,111,110,109,108,107,106,104
        DB   103,102,101,99,98,97,95,94,92,91,89,88,86,84,83
        DB   81,79,78,76,74,72,70,69,67,65,63,61,59,57,55,53
        DB   51,49,47,45,43,41,39,37,35,33,30,28,26,24,22,20
        DB   18,15,13,11,9,7,4,2,0,-2,-4,-7,-9,-11,-13,-15,-18
        DB   -20,-22,-24,-26,-28,-30,-33,-35,-37,-39,-41,-43
        DB   -45,-47,-49,-51,-53,-55,-57,-59,-61,-63,-65,-67
        DB   -69,-70,-72,-74,-76,-78,-79,-81,-83,-84,-86,-88
        DB   -89,-91,-92,-94,-95,-97,-98,-99,-101,-102,-103
        DB   -104,-106,-107,-108,-109

SINEC   DB   108,107,106,104,103,102,101,99,98,97,95,94,92,91
        DB   89,88,86,84,83,81,79,78,76,74,72,70,69,67,65,63
        DB   61,59,57,55,53,51,49,47,45,43,41,39,37,35,33,30
        DB   28,26,24,22,20,18,15,13,11,9,7,4,2,0,-2,-4,-7,-9
        DB   -11,-13,-15,-18,-20,-22,-24,-26,-28,-30,-33,-35
        DB   -37,-39,-41,-43,-45,-47,-49,-51,-53,-55,-57,-59
        DB   -61,-63,-65,-67,-69,-70,-72,-74,-76,-78,-79,-81
        DB   -83,-84,-86,-88,-89,-91,-92,-94,-95,-97,-98,-99
```

```
        DB   -101,-102,-103,-104,-106,-107,-108,-109,-110,-111
        DB   -112,-113,-114,-115,-116,-117,-118,-118,-119,-120
        DB   -120,-121,-122,-122,-123,-123,-124,-124,-124,-125
        DB   -125,-125,-126,-126,-126,-126,-126,-126,-126,-126
        DB   -126,-126,-126,-125,-125,-125,-124,-124,-124,-123
        DB   -123,-122,-122,-121,-120,-120,-119,-118,-118,-117
        DB   -116,-115,-114,-113,-112,-111,-110,-109,-108,-107
        DB   -106,-104,-103,-102,-101,-99,-98,-97,-95,-94,-92
        DB   -91,-89,-88,-86,-84,-83,-81,-79,-78,-76,-74,-72
        DB   -70,-69,-67,-65,-63,-61,-59,-57,-55,-53,-51,-49
        DB   -47,-45,-43,-41,-39,-37,-35,-33,-30,-28,-26,-24
        DB   -22,-20,-18,-15,-13,-11,-9,-7,-4,-2,0,2,4,7,9,11
        DB   13,15,18,20,22,24,26,28,30,33,35,37,39,41,43,45
        DB   47,49,51,53,55,57,59,61,63,65,67,69,70,72,74,76
        DB   78,79,81,83,84,86,88,89,91,92,94,95,97,98,99,101
        DB   102,103,104,106,107,108,109,110,111,112,113,114
        DB   115,116,117,118,118,119,120,120,121,122,122,123
        DB   123,124,124,124,125,125,125,126,126,126,126,126
        DB   126,126,126,126,126,126,125,125,125,124,124,124
        DB   123,123,122,122,121,120,120,119,118,118,117,116
        DB   115,114,113,112,111,110,109

FONT    DB      11000000B ; FONT FOR DATA = 0
        DB      11111001B ; 1
        DB      10100100B ; 2
        DB      10110000B ; 3
        DB      10011001B ; 4
        DB      10010010B ; 5
        DB      10000010B ; 6
        DB      11011000B ; 7
        DB      10000000B ; 8
        DB      10010000B ; 9
        DB      10001000B ; A
        DB      10000011B ; B
```

```
        DB      11000110B ; C
        DB      10100001B ; D
        DB      10000110B ; E
        DB      11000000B ; F
DATA1   DB      ?
DATA2   DB      ?
DATA3   DB      ?
PULSES  DB      20
STEP    DB      18
COUNTS  DB      0
M       DB      90  ; for 64 M=50%
CODE    ENDS
        END
```