

# STUDY AND ANALYSIS OF MICROPROCESSOR CONTROLLED PULSE-WIDTH MODULATED INVERTER



MOHAMMED YEASIN

Thesis  
Submitted to the Department of  
Computer science and Engineering  
in partial fulfilment of the requirements for the degree  
of  
Master of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY  
JULY, 1994 DHAKA, BANGLADESH

R  
623.81957  
1994  
YAS

**DEDICATED TO**

**MY PARENTS**

**AND**


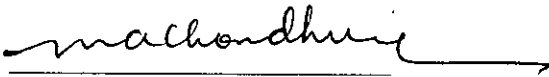
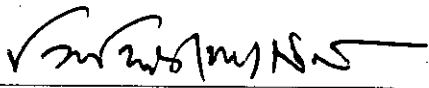
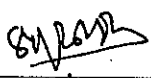
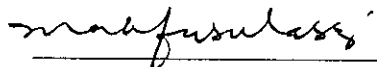
**TEACHERS**

The thesis titled; " STUDY AND ANALYSIS OF MICROPROCESSOR CONTROLLED PULSE-WIDTH MODULATED INVERTER ", submitted by Mohammed Yeasin, Roll No.: 891808 (F), Registration No: 87392 of M.Sc in Engineering has been accepted as satisfactory in partial fulfillment of the requirements for the degree of

**Master of Science in Computer Science and Engineering**

Examination held on: 25th July, 1994

Approved as to the style and contents by

1.   
(Dr. Md. Shamsul Alam) (Supervisor) Chairman  
Professor,  
Dept. of CSE, BUET
2.   
(Dr. M. A. Choudhury) (Co-supervisor) Member  
Associate professor  
Dept. of EEE, BUET
3.   
(Dr. A. B. M. Siddique Hossain) (Ex-Officio) Member  
Professor, <sup>of EEE</sup> and Head  
Dept. of CSE, BUET
4.   
(Dr. M. Kaykobad) Member  
Assistant Professor  
Dept. of CSE BUET
5.   
(Dr. S. Mahfuzul Aziz) Member (Ext.)  
Assistant Professor  
Dept. of EEE, BUET

## Acknowledgment

I wish to express my deep sense of gratitude to Dr. Md. Shamsul Alam, Professor, Department of Computer Science and Engineering and Dr. M. A. Choudhury, Associate Professor, Department of Electrical and Electronic Engineering, BUET, DHAKA for their guidance and kind supervision, constant encouragement and valuable suggestions in bringing this work to completion. Their encouragement specially during difficult times will be ever remembered. To them, I am forever, indebted.

It is but natural that due to the preoccupation with the research work, my parents, brothers and sisters were deprived of the privileges and prerogatives which either the society or nature has bestowed on them. To them, I am very much indebted.

I am extremely grateful to Dr. Siddique Hossain, Head, Department Of Computer Science and Engineering, BUET, DHAKA and Mr. Kazi Mujibue Rahman, Head, Department of EEE, BIT Chittagong for their kind help and cooperation during this work.

I would like to express my thanks to my close friends for their encouragement's, support and valuable advice. I would like to offer my thanks to the authorities of BIT Chittagong and my colleagues, for allowing me to complete this work. Finally, I would like to offer my thanks to all of my sincere well wishers specially all faculty members of the Department of Computer Science and Engineering and the Department of Electrical and Electronic Engineering, BUET, DHAKA.

## ABSTARCT

Microprocessor technology has opened a new frontier in the automation of industrial process controls. Interest has been growing in microprocessor based pulse width modulator ( PWM ) schemes in recent years. Among the several methods such as dedicated analog/digital, dedicated digital and microprocessor method of implementation, the last one offers several advantages. PWM inverters are used in a wide variety of industrial process such as uninterruptable power supplies (UPS), variable speed drives etc. The popularity is mainly due to the capability of these inverters permitting control of voltage, frequency and harmonic content in a single power stage. These inverters employ only one dc source with small filters at both ends of the inverter. The voltage and frequency of these converters can be changed according to need by varying gate or base drives of the static switches of a particular converter. The output waveforms are nonsinusoidal in nature and the harmonic components generated are detrimental to applications in terms of losses. These are also detrimental to the power supply utility in terms of line harmonics injection. Various methods have so far been used to minimize harmonics in inverters and one of the best methods is sine pulse width modulation (SPWM). In the era of microprocessor control and automation, there has been a constant effort by researchers to incorporate PWM switching in various static power converter/controller in order to obtain optimized operation. For real time implementation of PWM waveforms using microprocessor, there are different techniques of realization depending on the type of modulation process. The basic hardwares and softwares get modified depending on the technique of realization and the type of PWM waveforms. The techniques reported so far lack generality. The purpose of this research is to investigate the possible on-line computation and decision making for the generation of gating pulse to implement SPWM switching of an inverter.

# CONTENTS

	Pages
Acknowledgment	II
Abstract	III
Contents	IV
List of figures	VI
List of symbols	X
List of tables	XI
 <b>CHAPTER ONE: INTRODUCTION TO MICROPROCESSOR CONTROL</b>	
1-1 Power electronics and microprocessor	1
1-2 Advantage and limitations of microprocessor	3
1-3 Overview of microprocessor control strategy	5
1-4 Review of PWM techniques	6
1-5 Thesis Objectives and outlines	14
 <b>CHAPTER TWO: ANALYSIS OF SINUSOIDAL PULSE-WIDTH MODULATION</b>	
2-1 Introduction	16
2-2 Mathematical analysis of SPWM	19
2-3 Inverter waveform synthesis	21
2-4 Necessity of DFT analysis of PWM waveforms	21
2-5 Discrete Fourier Transform (DFT)	22
2-6 Discrete Fourier Transform of PWM waveforms	23
 <b>CHAPTER THREE: MICROPROCESSOR IMPLEMENTATION OF SINE PULSE-WIDTH MODULATION (SPWM)</b>	
3-1 Introduction	49
3-2 Formulation of the microprocessor implementation of PWM	49
3-3 Hardware schematic	52
3-4 Program description	56
3-5 Investigation for On-line implementation	63
3-6 Experimental results	66
 <b>CHAPTER FOUR: SUMMARY AND CONCLUSIONS</b>	
4-1 Conclusions	74
4-2 Recommendations for future work	75
 <b>REFERENCES</b>	 76
<b>APPENDIX - 1</b>	82
<b>APPENDIX - 2</b>	84
<b>APPENDIX - 3</b>	91
<b>APPENDIX - 4.1</b>	93
<b>APPENDIX - 4.2</b>	95

## LIST OF FIGURES

<u>Figure No</u>	<u>Title of the figure</u>	<u>Page</u>
Figure 1.1	Arrangement of an UPS system.	7
Figure 1.2	Single pulse-width modulation.	9
Figure 1.3	Multiple pulse-width modulation.	9
Figure 1.4	Principle of sine pulse-width modulation (SPWM).	11
Figure 1.5	Principle of delta modulation (DM).	11
Figure 1.6	Block diagram of analog SPWM implementation .	12
Figure 1.7	Block diagram of three-phase sine wave and Triangular wave generation.	13
Figure 2.1	Sinusoidal pulse-width modulation (SPWM).	16
Figure 2.2	Half-bridge inverter.	17
Figure 2.3	Expanded view of gating pulse in time scale.	19
Figure 2.4.1 (a)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 25$ and $m = .4$ .	25
Figure 2.4.1(b)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 25$ and $m = .6$ .	25
Figure 2.4.1(c)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 25$ and $m = .8$ .	25
Figure 2.4.2(a)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 15$ and $m = .4$ .	26
Figure 2.4.2(b)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 15$ and $m = .6$ .	26
Figure 2.4.2(c)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 15$ and $m = .8$ .	26
Figure 2.4.3(a)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 10$ and $m = .8$ .	27
Figure 2.4.3(b)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 20$ and $m = .8$ .	27
Figure 2.4.3(c)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 30$ and $m = .8$ .	27
Figure 2.4.4(a)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 10$ and $m = .6$ .	28
Figure 2.4.4(b)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 20$ and $m = .6$ .	28
Figure 2.4.4(c)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 30$ and $m = .6$ .	28
Figure 2.4.5(a)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 10$ and $m = .4$ .	29
Figure 2.4.5(b)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 20$ and $m = .4$ .	29
Figure 2.4.5(c)	Simulated PWM waveform at $f = 50\text{Hz}$ , $N = 30$ and $m = .4$ .	29
Figure 2.4.6(a)	Simulated PWM waveform at $f = 70\text{Hz}$ , $N = 15$ and $m = .4$ .	30
Figure 2.4.6(b)	Simulated PWM waveform at $f = 70\text{Hz}$ , $N = 15$ and $m = .9$ .	30
Figure 2.4.7(a)	Simulated PWM waveform at $f = 70\text{Hz}$ , $N = 7$ and $m = .4$ .	31
Figure 2.4.7(b)	Simulated PWM waveform at $f = 70\text{Hz}$ , $N = 7$ and $m = .9$ .	31
Figure 2.4.8(a)	Simulated PWM waveform at $f = 30\text{Hz}$ , $N = 15$ and $m = .4$ .	32
Figure 2.4.8(b)	Simulated PWM waveform at $f = 30\text{Hz}$ , $N = 15$ and $m = .9$ .	32
Figure 2.5.1(a)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 25$ and $m = .4$ .	33
Figure 2.5.1(b)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 25$ and $m = .6$ .	33
Figure 2.5.1(c)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 25$ and $m = .8$ .	33
Figure 2.5.2(a)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 15$ and $m = .4$ .	34
Figure 2.5.2(b)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 15$ and $m = .6$ .	34
Figure 2.5.2(c)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 15$ and $m = .8$ .	34
Figure 2.5.3(a)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 30$ and $m = .8$ .	35
Figure 2.5.3(b)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 20$ and $m = .8$ .	35
Figure 2.5.3(c)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 10$ and $m = .8$ .	35

<u>Figure No.</u>	<u>Title of the figure</u>	<u>Page</u>
Figure 2.5.4(a)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 10$ and $m = .4$ .	36
Figure 2.5.4(b)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 20$ and $m = .4$ .	36
Figure 2.5.4(c)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 30$ and $m = .4$ .	36
Figure 2.5.5(a)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 10$ and $m = .6$ .	37
Figure 2.5.5(b)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 20$ and $m = .6$ .	37
Figure 2.5.5(c)	Spectrum of PWM waveform at $f = 50\text{Hz}$ , $N = 30$ and $m = .6$ .	37
Figure 2.5.6(a)	Spectrum of PWM waveform at $f = 70\text{Hz}$ , $N = 15$ and $m = .4$ .	38
Figure 2.5.6 (b)	Spectrum of PWM waveform at $f = 70\text{Hz}$ , $N = 15$ and $m = .9$ .	38
Figure 2.5.7(a)	Spectrum of PWM waveform at $f = 70\text{Hz}$ , $N = 7$ and $m = .4$ .	39
Figure 2.5.7(b)	Spectrum of PWM waveform at $f = 70\text{Hz}$ , $N = 7$ and $m = .9$ .	39
Figure 2.5.8(a)	Spectrum of PWM waveform at $f = 30\text{Hz}$ , $N = 15$ and $m = .4$ .	40
Figure 2.5.8(b)	Spectrum of PWM waveform at $f = 30\text{Hz}$ , $N = 15$ and $m = .9$ .	40
Figure 2.6.1(a)	Three dimensional spectrum of SPWM waveform at $f = 30\text{ Hz}$ and $N = 20$ carrier wave per half cycle .	43
Figure 2.6.1(b)	Three dimensional spectrum of PWM waveform at $f = 30\text{ Hz}$ and $N = 15$ carrier wave per half cycle.	43
Figure 2.6.2(a)	Three dimensional spectrum of PWM waveform at $f = 50\text{ Hz}$ and $N = 30$ carrier wave per half cycle.	44
Figure 2.6.2(b)	Three dimensional spectrum of PWM waveform at $f = 50\text{ Hz}$ and $N = 25$ carrier wave per half cycle.	44
Figure 2.6.3(a)	Three dimensional spectrum of PWM waveform at $f = 50\text{ Hz}$ and $N = 20$ carrier wave per half cycle.	45
Figure 2.6.3(b)	Three dimensional spectrum of PWM waveform at $f = 50\text{ Hz}$ and $N = 15$ carrier wave per half cycle.	45
Figure 2.6.4(a)	Three dimensional spectrum of PWM waveform at $f = 70\text{ Hz}$ and $N = 10$ carrier wave per half cycle.	46
Figure 2.6.4(b)	Three dimensional spectrum of PWM waveform at $f = 70\text{ Hz}$ and $N = 5$ carrier wave per half cycle .	46
Figure 2.6.5(a)	Three dimensional spectrum of PWM waveform at $f = 70\text{ Hz}$ and $N = 15$ carrier wave per half cycle .	47
Figure 2.6.5(b)	Three dimensional spectrum of PWM waveform at $f = 70\text{ Hz}$ and $N = 20$ carrier wave per half cycle.	47
Figure 3.1.1	Typical sine pulse-width modulation (SPWM ).	50
Figure 3.1.2	Typical switching points of sine pulse-width modulation	51
Figure 3.2.1	Flow chart of the main program (software counter).	53
Figure 3.2.2	Flow chart of the main program ( 8253 timer).	53
Figure 3.3	Hardware schematic diagram.	54
Figure 3.4	Block diagram of the Experimental setup.	54
Figure 3.5	Half bridge inverter with commutation circuit.	55
Figure 3.6	Closed -loop control of D.C and induction motors	64
Figure 3.6 (a)	Schematic diagram of computer controlled four quadrant dc drive.	64



<u>Figure No.</u>	<u>Title of the figure</u>	<u>Page</u>
Figure 3.6(b)	Closed loop control of induction motors stator voltage control	64
Figure 3.6(c)	Closed loop control of induction motors V/f control.	64
Figure 3.6(d)	Closed loop control of induction motors slip regulation control.	64
Figure 3.7	Block diagram for On-line gating pulse generation for inverter.	65
Figure 3.8.1(a)	Typical experimental oscillogram of PWM wave.	67
Figure 3.8.1(b)	Typical experimental PWM wave.	67
Figure 3.8.2(a)	Oscillogram of PWM wave at $f = 70$ Hz , $m = .9$ and $N = 7$ .	68
Figure 3.8.2(b)	Oscillogram of PWM wave at $f = 70$ Hz , $m = .4$ and $N = 7$ .	68
Figure 3.8.3(a)	Oscillogram of PWM wave at $f = 70$ Hz, $m = .4$ and $N = 15$ .	69
Figure 3.8.3(b)	Oscillogram of PWM wave at $f = 70$ Hz, $m = .9$ and $N = 15$ .	69
Figure 3.8.4(a)	Oscillogram of PWM wave at $f = 30$ Hz, $m = .9$ and $N = 15$ .	70
Figure 3.8.4(b)	Oscillogram of PWM wave at $f = 30$ Hz, $m = .4$ and $N = 15$ .	70
Figure 3.9.1(a)	Experimental output of the inverter at $f = 70$ Hz, $m = .9$ and $N = 7$ carrier wave per half cycle.	71
Figure 3.9.1(b)	Experimental output of the inverter at $f = 70$ Hz, $m = .4$ and $N = 7$ carrier wave per half cycle.	71
Figure 3.9.2(a)	Experimental output of the inverter at $f = 30$ Hz , $m = .9$ and $N = 15$ carrier wave per half cycle.	72
Figure 3.9.2(b)	Experimental output of the inverter at $f = 30$ Hz, $m = .4$ and $N = 15$ carrier wave per half cycle.	72
Figure 3.9.3	Oscillogram of PWM wave at 70 Hz $m = .9$ and $N = 7$ (using 8253 programmable timer)	73

## LIST OF SYMBOLS

$A_n$	Fourier coefficient.
$A_c$	Amplitude of the carrier wave.
$A_r$	Amplitude of the reference wave.
$B_n$	Fourier coefficient.
$C_n$	Fourier coefficient.
$e$	Modulating sine wave.
$f$	Frequency in Hz.
$f$	Output frequency in Hz.
$f_c$	Carrier frequency in Hz.
$m$	Modulation index.
$m_f$	Frequency modulation ration.
$N$	Number of carrier wave per half cycle.
$sw_1$	Switching function 1.
$sw_2$	Switching function 2.
$V_0$	Output voltage in volt.
$V$	Voltage in volt.
$S$	Slope of the triangular wave in volts/seconds or volts/radians.
$T$	Period of switching function $sw_1$ in second.
$\theta_i$	Mid position of $i$ th pulse.
$\delta_i$	Pulse width of $i$ th pulse.
$T_{ON}$	ON-time period of inverter.
$T_{OFF}$	OFF-time period of inverter.
$U(t, t_0)$	Unity step function starting at $t_0$ .
$g(t, t_0, t_1)$	Gate function starting at $t_0$ and ending at $t_1$ .
$wordcnt_{ON}$	Hexa decimal count value for the ON time of the static switches.
$wordcnt_{OFF}$	Hexa decimal count value for the OFF time of the static switches.

## LIST OF TABLES

<u>Table NO.</u>	<u>Title of the table</u>	<u>Page</u>
Table 2.1.	Result of spectral analysis of SPWM wave for $f = 50$ , $m = .4$ and $N = 10$ , and 30 respectively.	27
Table 2.2.	Result of spectral analysis of SPWM wave for $f = 50$ , $m = .6$ and $N = 10$ , and 30 respectively.	41
Table 2.3.	Result of spectral analysis of SPWM wave for $f = 50$ , $m = .8$ and $N = 10$ , and 30 respectively.	41
Table 2.4.	Result of spectral analysis of SPWM wave for $f = 50$ , $N = 15$ and $m = .4$ , $.6$ and $.8$ respectively.	42
Table 2.5.	Result of DFT analysis.	48
Table 3.1.	The switching instants and their hex values (software counter) and their decimal values for 8253 programmable timer at 70 Hz, modulation index $m = .9$ and $N = 15$ carrier pulse/ half cycle.	58
Table 3.2.	The switching instants and their hex values (software counter) and their decimal values for 8253 programmable timer at 70 Hz, modulation index $m = .4$ and $N = 15$ carrier pulse/ half cycle.	58
Table 3.3.	The switching instants and their hex values (software counter) and their decimal values for 8253 programmable timer at 70 Hz, modulation index $m = .9$ and $N = 7$ carrier pulse / half cycle.	59
Table 3.4.	The switching instants and their hex values Table No. Title of the table (software counter) and their decimal values for 8253 programmable timer at 30 Hz, modulation index $m = .4$ and $N = 15$ carrier pulse/ half cycle.	59
Table 3.5.	The switching instants and their hex values (software counter) and their decimal values for 8253 programmable timer at 70 Hz, modulation index $m = .4$ and $N = 7$ carrier pulse / half cycle.	60
Table 3.6.	The switching instants and their hex values (software counter) and their decimal values for 8253 programmable timer at 30 Hz, modulation index $m = .9$ and $N = 15$ carrier wave/half cycle.	60
Table 3.7.	The switching instants and their hex values (software counter) and their decimal values for 8253 programmable timer at 30 Hz, modulation index $m = .4$ and $N = 15$ carrier pulse/ half cycle.	61
Table 3.8.	The switching instants and their hex values (software counter) and their decimal values for 8253 programmable timer at 30 Hz, modulation index $m = .9$ and $N = 35$ carrier pulse/ half cycle.	62

---

CHAPTER  
ONE

---

INTRODUCTION TO MICROPROCESSOR CONTROL

# INTRODUCTION TO MICROPROCESSOR CONTROL

Microprocessors have profoundly influenced the technology of power electronics and drive systems. Their use brought simplification of hardware with improved reliability and permitted performance optimization combined with powerful diagnostic capability. The performance of microcomputers in terms of functional integration, improvements of architecture and speed is continually improving since their introduction at the beginning of 1970's. Super microcomputer featuring number of devices approaching millions on a chip [1] and having only few nanoseconds instruction cycle time are now available.

Microprocessors have been accepted in the control of power electronics and drive system and the spectrum of their application is growing continuously. The impact is significant in the context of industrial automation. Microcomputer is not only providing high level factory automation, but also playing a vital role in the control of lower level power electronics and motion control systems [2 - 7 ].

The first generation of microprocessors used 4-bit architecture which were quickly superseded by more widely used 8-bit microprocessors. Gradually 16-bit microprocessors appeared and now 32-bit microprocessors are available with a tremendous speed like 100 MHz [8]. In modern VLSI era, a severe competition is taking place between general purpose VLSI microcomputers and dedicated custom and semicustom design VLSI controllers.

## 1-1 POWER ELECTRONICS AND MICROPROCESSOR'S

In the beginning of the microprocessors era, microprocessors were essentially used for implementation of logic control functions. They functioned as a Boolean synthesizer for programmable logic control applications. As microprocessor improved in bit size, execution time, and hardware functional integration their application gradually expanded to general control systems. In power electronic system , the functions of microprocessor can in general be categorized as follows:

- Control of feedback loops.
- Gate firing control of phase-controlled converters.
- PWM or square wave signal generation for inverters.
- Optimal and adaptive control.

- Estimation of feedback signal.
- General sequencing control.
- Protection and fault overriding control.
- Signal monitoring and warning
- Data acquisition.
- Diagnostics
- Miscellaneous computation and control.

Some of the above functions are reviewed in the following section.

**Programmable time delay:-** A microprocessor can easily generate time delays by software upcounting or downcounting principles. In down counting, for example, a memory location can be preloaded with a digital number and decremented periodically until the location clears. Similar time delay can be generated by hardware timer located in the microcomputer. At clear, the timer can generate an interrupt signal which may execute a subroutine or start an external event. In multitasking operation, a microcomputer may generate several sampling time intervals.

**Pulse Width Modulation (PWM) :-** A microprocessor can generate a PWM signal for choppers or inverters ( DC to DC converter are known as chopper and DC to AC converters are known as inverter). In chopper control, programmable pulse widths can be generated proportional to digital words through a hardware or software downcounter. There may be several techniques for inverter PWM signal generation. Look-up tables of digital words corresponding to pulse and notch widths at different magnitudes of fundamental voltage can be stored in memory location and then converted to a PWM pulse train with the help of down counters. The counter clock frequency is directly related to the fundamental frequency so that time intervals match the angular intervals. The look-up tables may be based on a selected harmonic elimination or minimum rms ripple current for a specified load.

**Monitoring and Warning:-** Monitoring of software parameters by digital display is very convenient because the parameters are available in the microprocessor memory. A number of software system variables can be monitored through a diagnostic instrument during system debugging. If some variables tend to exceed the safe limits, audio visual signals can be generated. If the operator fails to take the appropriate action, the system may be shutdown automatically.

**Protection and Fault Overriding Control:-** The system fault protection, and in some cases , fault preventing control can be assigned to a microprocessor. If the protection is very time critical in nature, such as inhibiting base drive signals of transistors for overcurrent protection, it has to be designed with a dedicated hardware. A fault overriding control can be designed on the basis of detailed analysis of the system.

**Test and Diagnostics:-** A microprocessor can be used to perform various tests in a power electronic system. The data captured in data acquisition systems can be used to calculate the performance curves, such as torque, efficiency, power factor, etc., as a function of load and speed of the drive system. On-line parameter estimation can also be done on the system. At different steady state operating points of a nonlinear system, small perturbation signals can be injected, and the response can be analyzed to determine the structure, poles, zeroes, and gain of the systems. Powerful diagnostic software can be designed to identify fault in the microcomputer's hardware, software, and system components. Some amount of diagnostics is built into the application program, and is exercised in real time to identify fault during the running condition of the system. A separate microcomputer can be used if more sophisticated diagnostic tests are desired. Diagnostics can be designed for both On-line and OFF-line applications.

## 1-2 ADVANTAGES AND LIMITATIONS OF MICROPROCESSORS

Microprocessors, have some merits and limitations which can be summarized as follows,

**Hardware Cost:-** The simplification of control hardware and the corresponding reduction in cost is the principal advantage of microcomputer control when compared to analog control. This trend is evident as microcomputer speed is improving and more functions with increased volume of each function are being integrated. Custom or semicustom VLSI chips with the integration of total control hardware for specific applications in large volume production can be very economical. Smaller size and less weight with reduced power consumption's are additional advantages.

**Improved Reliability:-** System reliability or MTBF ( mean time between failures ) can be considerably improved by microprocessor control. The reliability of an LSI or VLSI chip is considerably higher than that of electronic circuit with a large number of electronic components. Past experience has shown that a microprocessor has a

much higher reliability than other components of power electronics. The reliability can be enhanced by temperature derating and by using military specification components at increased cost.

**EMI Problems:-** High level integration of microprocessor avoid coupling of large voltage and current transients in a power electronic system. Only nominal shielding of controller is sufficient. Noise coupling through power supply and signals can be minimized providing good analog and digital filtering.

**Drift and Parameter Variation:-** Digital signal processing eliminates signal drift and parameter variation effects which are prevalent in analog controllers. Digital computation is almost 100% accurate, and truncation and overflow problem can be avoided by appropriate scaling.

**Compatibility with Hierarchical Digital Control:-** Microprocessor control of local power electronic systems permits compatibility with higher level and lateral control systems. In automated industrial environment, for example, the bi-directional communication between local and host microcomputers, and among local microcomputers for control and diagnostics become important.

**Universal Hardware and Software:-** Universal hardware can be designed for certain class of systems where software can be modified to satisfy the subclass specifications, for example, all voltage-fed inverter drives can have the same controller hardware with different performance specifications incorporated by software modification. The additional advantage of software control is that it is flexible and can be easily upgraded as the performance requirement changes.

**Diagnostics:-** An additional advantage of microcomputer control is that powerful diagnostic software can be designed which can be used by a semiskilled technician, and system downtime can be reduced. In addition, data acquisition, signal monitoring, warning and displays can be provided.

**Sluggish computation:-** A microprocessor implementation of a function is slower than dedicated analog/ digital hardware control. The reason is that a microprocessor computes a function in a serial manner, whereas, dedicated hardware signals are processed in parallel paths. If a microprocessor handles multiple tasks, time has to be allocated for each task in a multiplexed manner. This slows down the execution speed. Large sampling delay causes deterioration of performance and may cause



stability problems in feedback loops. Microcomputers are being made available with faster and faster computation capability and the assistance of the dedicated hardware can considerably enhance execution speed.

**Quantization Error:-** The signals in a physical system are analog in nature, and when these are interfaced with a microcomputer by A/D and D/A converter there is a finite quantization error. The distortion of analog signals due to quantization and sampling delay may be evident if a sine wave voltage is sampled by an A/D converter and then displayed back through a D/A converter. The quantization error can be minimized by increasing the bit size of the microcomputer and signal converters.

**Lack of Access to Software Signals:-** Microprocessor control does not provide easy access to software signals for monitoring of a hardware controlled system. Diagnostic and debugging instruments may have to be custom designed for this purpose.

**Software Developments May Be Expensive:-** Software development require expert assistance and is time consuming, and thus expensive. The additional cost may be justified from the viewpoint of overall advantages. The incremental cost for high volume production items may not be very noticeable.

### **1-3 OVERVIEW OF MICROPROCESSOR CONTROL STRATEGY**

The microprocessor-based digital control technique has become indispensable with the growing requirements for enhanced supervising capability and better control performance in modern control systems. It has largely contributed to provide them with powerful RAS (Reliability, Availability and Serviceability). It can also contribute to the practical application of the advanced control techniques such as the ones based on the modern control theory. The trend is found not only in the rising industrial application in the form of robot control and factory automation and so forth but also in the replacement of plants in the established industries. The distinguished features obtained by microprocessor application are as follows:

#### **Standardization of hardware:**

The control system using a microprocessor can be standardized and modification or replacement can be done flexibly in software's to cope with the variation of the specifications.

### **Compact and light weight:**

The controller apparatus having microprocessor control is more compact and light weight than conventional apparatus because the programmed microprocessor can replace (the logic sequence of the power circuit in place of) the auxiliary relays or interlocks of line switches and power break changeover switches etc.

### **Low maintenance cost:**

Digital control system using microprocessor requires no adjustment. As a result, it is possible to reduce maintenance costs.

### **Self-check function:**

The microprocessor not only controls the control circuit but also tests itself and finds out wrong section through self-checking program. This function saves the time for trouble shooting.

### **Monitoring function:**

The programmed microprocessor observes the condition of control circuit. Observed data may be stored in memory repeatedly and used to find out the cause of the trouble in the service. The microprocessor can also be used to inform the operator immediately what to do in case of trouble.

## **1-4 REVIEW OF PWM TECHNIQUES**

In many industrial applications, it is often required to control the output voltage of inverters (1) to cope with the variations of dc input voltage, (2) for voltage regulations of inverters, and (3) for constant volts /frequency control requirements [9-14]. PWM inverters are used in a wide variety of industrial processes such as uninterruptable power supplies, static frequency changers and variable speed drives. A practical circuit of an inverter is shown in figure 1.1. This arrangement is an UPS system, which consists of a dc source, inverter and bridge rectifier. In case of power failure the battery supplies the inverter ( if the gating signals are available thyristor T1 and T4 conducts for negative half cycle of the load current and T2 and T3 conducts for the positive half cycle of the load current) which ensures uninterrupted power supply at load. When the main ac supply is available, the bridge rectifier (diode D1 and D4 conducts during positive half cycle of the ac input and diode D2 and D3 conducts during negative half cycle of the ac input and capacitor C acts as a filter to smooth the output of bridge rectifier) charges the battery  $V_s$  and the

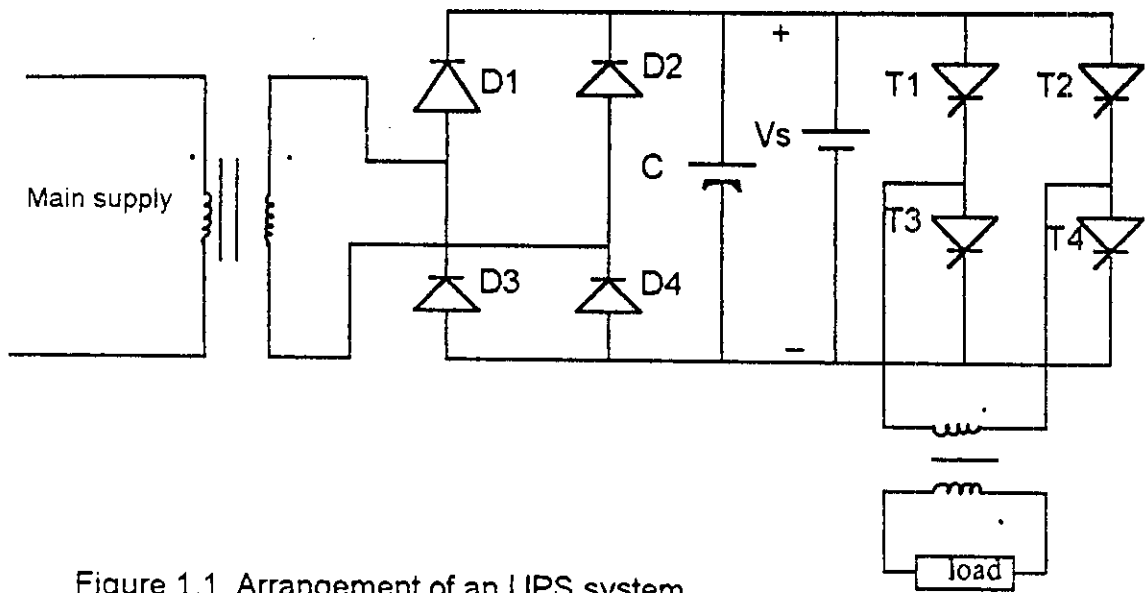


Figure 1.1 Arrangement of an UPS system

inverter takes power from from main supply through bridge rectifier. In this arrangement, the inverter has to operate at the fundamental output frequency. The outputs of normal inverters are non sinusoidal( square wave, semi square wave) in nature and contains harmonics. The harmonic contents are detrimental to the applications in terms of losses. They are also detrimental to the power supply utility in-terms of line harmonic injection and requires large output filter to eliminate low order harmonics. This can be improved by employing a PWM inverter to reduce the output transformer size. in PWM switching the output voltage and frequency can be changed according to the need by varying the gating pulse of a particular converter. Switches are operated at higher frequencies. High frequency switching causes the dominant harmonic to occur at higher frequency which is odd integer multiple of the fundamental frequency and reduces the output filter size. PWM inverters are capable of controlling voltage, frequency, and harmonic contents in a single power stage [15- 17, 2-6, 12].

In the pulse width modulation technique the switches of the power converters are operated at higher frequencies according to a particular pattern so as to produce pulses of varying widths at the output of the inverter. The earliest modulation technique applied to inverter operation are single pulse modulation and the multiple pulse modulation [ 18-22, 6,17]. In single pulse-modulation control, there is only one pulse per half cycle of the reference wave and the width of the pulse is varied to control the inverter output voltage or power. Figure 1.2 shows the

generation of gating signals and output voltage of single phase full bridge inverter. The gating signals are generated by comparing a rectangular reference wave, of amplitude  $A_r$  with a triangular carrier wave of amplitude  $A_c$ . The frequency of the reference signal determines the fundamental frequency of the output voltage. By varying  $A_r$  from 0 to  $A_c$  the pulse width  $\delta$  can be varied from 0 to 180 degree. The dominant harmonic is the third, and the distortion factor increases significantly at low output voltage [23-24,5, 7]. The harmonic content can be reduced by using several pulses in each half cycle of output voltage. The generation of gating signals of multiple pulse PWM for turning ON and OFF the inverter switches are shown in figure 1.3. The frequency of the reference signal sets the output frequency  $f_c$  and the carrier frequency  $f_0$  determines the number of pulses per half cycle,  $N$ . The order of harmonics is the same as that of single pulse modulation. The distortion factor is reduced significantly compared to that of single-pulse modulation. However, due to large number of switching per half cycle the switching loss would increase. With larger values of  $N$  the amplitudes of the lower harmonics would be lower but the amplitudes of the higher order harmonics would increase. These techniques are capable of providing inverter output voltages with low harmonic contents.

Among several PWM techniques, sinusoidal pulse width modulation [SPWM] is common. At the beginning two different types, namely the synchronous and asynchronous sine pulse width modulation scheme were used for switching power converters [25- 31, 12,16, 24]. In sine pulse width modulation, instead of maintaining the width of each pulse same (as in the case of multiple-pulse modulation) the width of each pulse is varied in proportion to the amplitude of a sine wave evaluated at the center of the same pulse. The distortion factor and lower order harmonics are reduced significantly [32- 34,6-12,23]. The gating signals as shown in figure 1.4, are generated by comparing an isosceles triangular carrier wave with the sine wave signal. The crossover points determine the points of commutation. Except at low frequency range the carrier is synchronized with the modulating signal, and an odd integral (multiple of three, five etc.) ratio is maintained to improve the harmonic content. The fundamental output voltage can be varied by variation of the modulation index [35-36,12,22-34]. It can be shown that if the modulation index is less than unity, only carrier frequency harmonic with the fundamental frequency related side bands appear at the output. The voltage of inverter can be increased by changing modulation index until maximum voltage is obtained in square wave mode. The distortion factor is significantly less compared to that of multiple-pulse modulation. This type of modulation eliminates all the harmonics less than or equal to  $2(N - 1)$ . The output voltage of an inverter contains

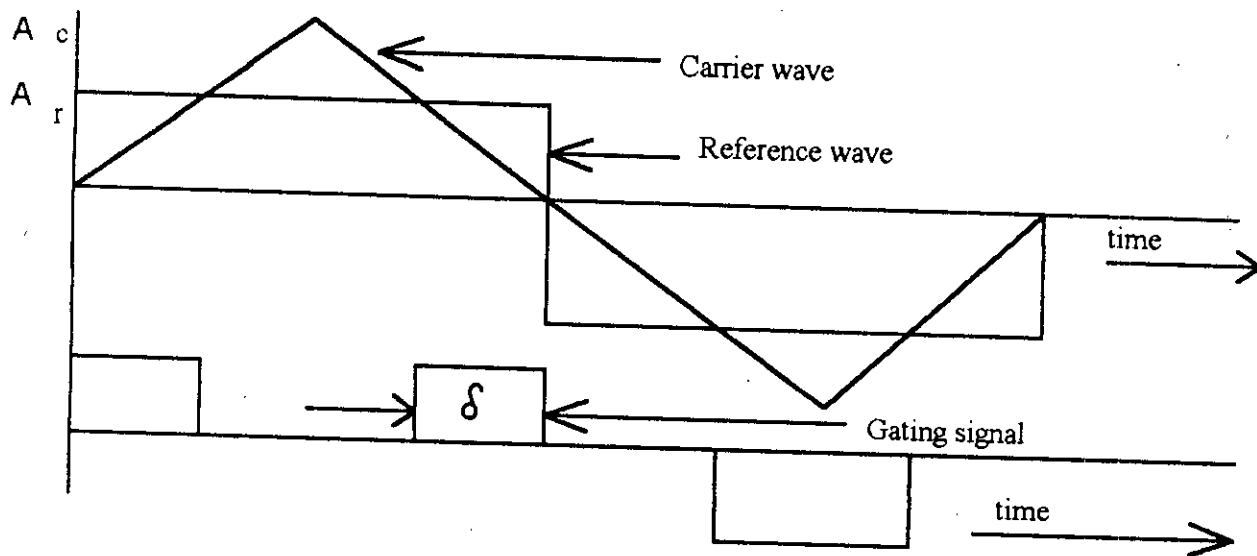


Figure 1.2 Single pulse-width modulation

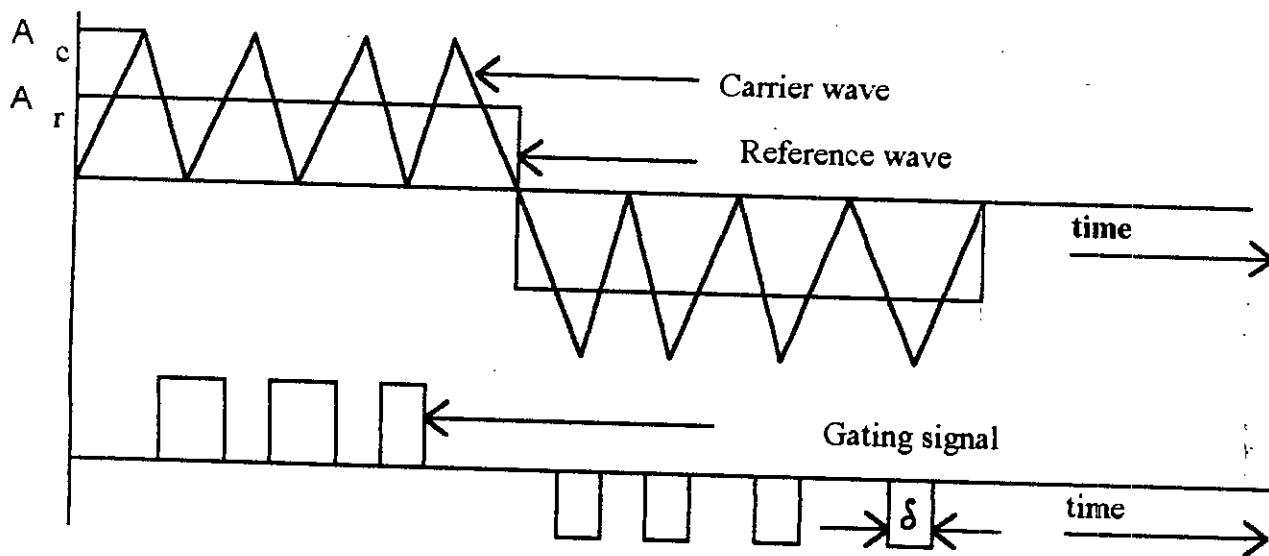
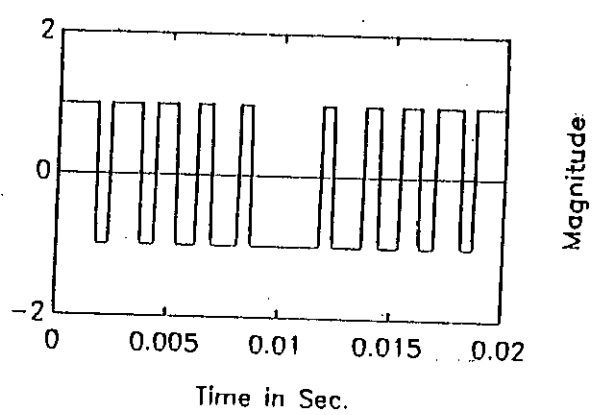
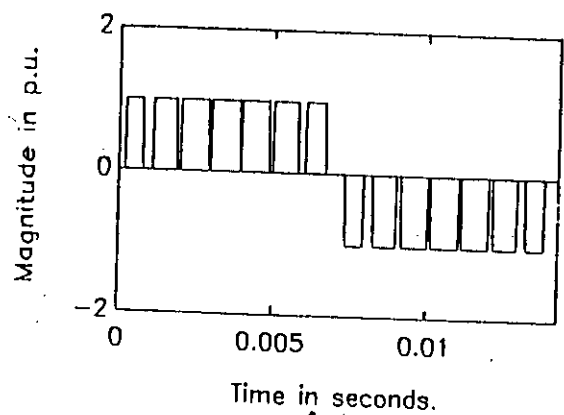
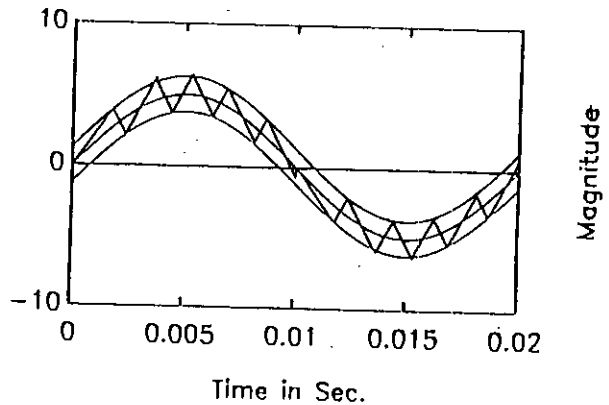
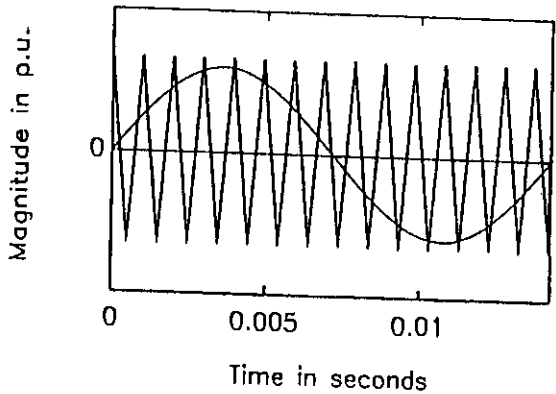


Figure 1.3, Multiple pulse-width modulation

harmonics. The PWM pushes the harmonics into a higher frequency range around the frequency  $f_c$  and its multiples, that is, around harmonics  $m_f$ ,  $2m_f$ ,  $3m_f$  and so on. For drive applications, the fixed frequency modulation was found to be problematic at different frequencies. Over modulation is normally avoided in applications requiring low distortion [i.e. uninterruptable power supplies (UPSs)].

In order to overcome the drawbacks of ordinary sine pulse width modulation in drive applications, variable ratio PWM schemes were introduced. Three distinct sinusoidal pulse width modulation schemes are in use for inverters [37-41,2-4, 6-19]. They are the 1) Natural sampled, 2) Symmetric regular sampled and 3) Asymmetric regular sampled PWM. In regular sampling method sine wave was replaced by a sampled or stepped sine wave. The stepped sine wave is not a sampled approximation to the sine wave. It is divided in specified intervals, say  $20^\circ$ , each interval being controlled individually to control the magnitude of the fundamental component and to eliminate specific harmonics. This type of control gives low distortion but higher fundamental amplitude compared to that of normal PWM control. This method is very popular in microcomputer implementation [42-43,3-9, 37-39]. On the other hand the optimized PWM wave form do not follow well defined modulation process [44,39-43]. This PWM approach is based on certain performance criteria [45,30,32,35]. As a result of the developments of microprocessor technology in recent years, the implementation of optimized pulse width modulation for switching inverters has become feasible [46,41,4]. The technique of selected harmonic elimination PWM has received wide attention recently. In this method the notches are created at predetermined angles of the square wave which permits voltage control as well as elimination of selected harmonics. The notch angle can be programmed so that the rms ripple current for a specified load condition is minimum. The microcomputer is specially adaptable to this type of PWM, where look-up table of the angles are stored in the ROM.

In current PWM scheme the feedback current tracks the reference current wave within a hysteresis band. Two types of PWM strategies have been reported recently for inverter operation. They are bang-bang sample PWM technique and the delta modulation ( DM ) technique. The principal of bang-bang sampling is based on the motor output current hysteresis comparison with a reference wave form to generate the modulated wave form [47, 41-46]. In recent years delta modulation has been finding its way for generating switching wave forms for converters [48,39,28]. In delta modulation a triangular wave is allowed to oscillate within a defined window above and below the reference sine wave  $A_r$ . The inverter switching function



(14)

(15)

Figure 1.4

Principle of sine pulse-width modulation (SPWM.)

Figure 1.5

Principle of Delta modulation (DM).

which is identical to the output voltage  $V_o$  is generated from the vertices of the triangular wave  $A_c$  as shown in figure 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 the pulse width of the modulated wave would change. The fundamental output voltage can be upto  $1.27V_s$  [49,6] 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 of an inverter which is a desirable feature in ac motor control. The analysis and the applications of the PWM inverters in drives are also important areas of research [50,21,27,36]. However, due to the complexity of modulation process, a general approach for such studies have not been developed.

**Typical analog SPWM circuit requirements:-** The following functional blocks are essential for the implementation of a analog SPWM circuit.

- Carrier wave generation
- Reference sine wave generation
- Modulator
- Logic circuit

The basic principle can be explained using the block diagram presented in figure 1.6.

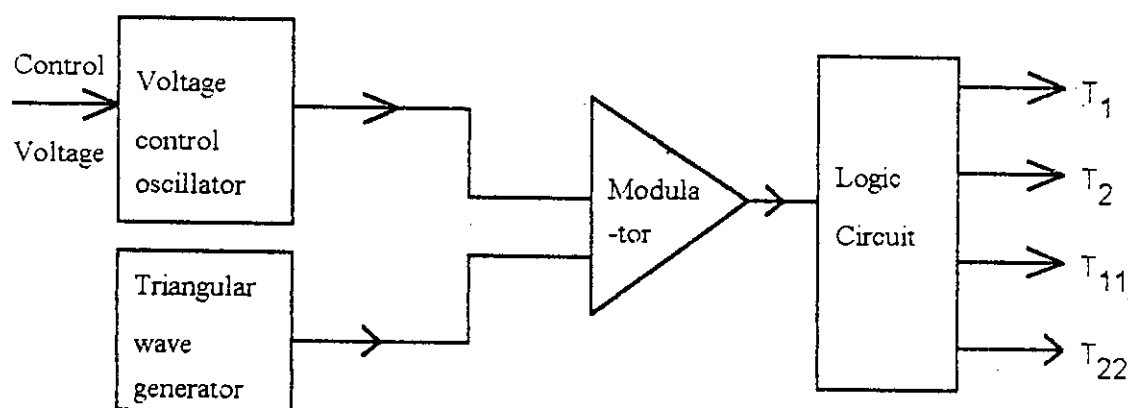


Figure 1.6, Block diagram of analog SPWM gating pulse generation

Applying control voltage to the input of voltage control oscillator sine wave can be generated. Similarly from function generator we can get triangular wave of required frequency. These two signals are then compared using a modulator (



Usually an OP-AMP). The hardware required for generation of the reference sine wave and carrier triangular wave are available in [51-56]. The process of generation of gating pulse  $T_1$ ,  $T_2$ ,  $T_{11}$ , and  $T_{22}$  were described in chapter three. It may appear simple but the synchronization of reference wave and carrier wave is difficult to implement. This problem is more acute for three-phase implementation. The generation of three-phase sine wave and triangular can be explained by the block diagram presented in figure 1.7.

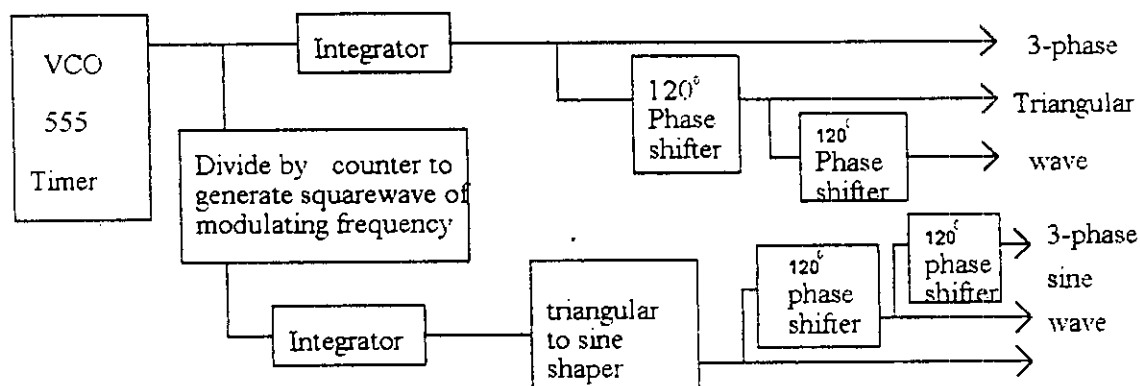


Figure 1.7 Block diagram of a 3-phase triangular and sine wave generation.

To mitigate the synchronization problem reference wave is usually generated from carrier wave, employing divide by counter to scale down the carrier frequency to reference frequency. For three phase generation 120 degree phase shift networks are needed for both carrier and reference wave signals. Triangular to sine shaper is needed to generate sine wave. The problem at this stage is that 120 degree phase shift network offer poor performance for variable frequency operation. Additional compensation and adjustment techniques are needed for this purpose. Three phase variable frequency reference wave of 5-200 Hz are usually essential in ac motor controllers. The low variable frequency sinusoidal signal is used in the control circuitry of cycloinverters and pulse-width modulated inverter. A number of techniques have been reported in literature's. These can be classified as: a) analog b) digital c) hybrid and d) analog computer techniques method. The main disadvantage of these methods is that the control circuitry for low frequency sine wave generation are very complex. The sine wave generated using the principle

described above are not totally distortionless. Moreover, sine wave generated by sine shaping of the triangular wave have unsymmetrical rising and falling edge slope. For three-phase implementation the mentioned problem are to be solved and the required adjustments and compensation are to be provided. Additionally a logic circuit is needed to obtain the gating pulses for the static switches. The circuit and the timing diagram is also available in [56]. For three-phase implementation three identical units are needed and became costly and poor in respect of performance for variable frequency operation.

**Need for microprocessor implementation:-** An analog control scheme has several disadvantages: Non linearity of speed sensor, Temperature dependency, drift and offset. Once a control circuit is built to meet certain performance criteria, it may require major changes in the hardwired logic circuits to meet other performance requirements. On the other hand microprocessor implementation of this is straight forward. A microcomputer control reduces the size and cost of hardware, improves reliability and control. The control scheme is implemented in software and is flexible to change the control strategy to meet different performance characteristics or to add extra control features. A microcomputer can also perform various desirable functions: on/off of the main power supply, start/stop of the control system, speed control, current control, monitoring the control variables, protection and trip circuit, diagnostics for built in fault finding and communication with a supervisory computer.

## 1-5 THESIS OBJECTIVES AND OUTLINES

For real time implementation of wave forms using microprocessors, there are different techniques of realization depending on the type of modulation process [57,2,22-34]. The basic hardware and software get modified depending upon the technique of realization and the type of PWM wave form. Hence the PWM technique though useful lacks generality. Further, at higher frequencies, the PWM wave form resulting from the techniques described in "Review of PWM techniques" suffer from the harmonic distortion due to shifting of edges of PWM pulses [57,14, 41-49]. The objective of this study is the "Microprocessor implementation of SPWM wave forms generation to control an inverter".

- a) A technique is studied to find the switching points of a sine pulse-width modulated wave. This technique will enable probable On-line implementation of the modulation scheme in variable frequency operation of inverters.

- b) Knowledge of the switching point is used to study harmonics of inverter outputs. The harmonic analysis is carried out by Fast Fourier Transform .
- c) A selection was done by observing three dimensional spectral analysis for carrying out the experiment of inverter wave forms for a particular fundamental frequency based on number of carrier wave per half cycle, the harmonic contents of the modulated wave forms at different carrier frequency and at different modulation index to make judicious selection for fundamental voltage available at the inverter output.
- d) Experimental implementation and verification of the result.

A novel method of finding switching points of a SPWM inverter wave forms has been proposed during this research. For the wave form analysis discrete Fourier transform has been used on the sampled wave forms. This type of wave form analysis is also suitable for on-line spectral estimation and sub harmonic detection of PWM wave forms [57,54]. The On-line spectral estimation will enable judicious selection of carrier frequency based on spectral contents and will be useful for close loop implementation of SPWM. This part of the study was not done during this research. Study and analysis of the proposed technique is described in chapter two. Implementation details and the experimental results are summarized in Chapter Three. Chapter Four concludes the thesis with statements on achievements and recommendation for future work.

---

**CHAPTER**

**TWO**

---

**ANALYSIS OF SINE PULSE-WIDTH MODULATION**

## ANALYSIS OF SINE PULSE-WIDTH MODULATION

In Sinusoidal pulse-width modulation ( SPWM ) the width of each pulse is varied in proportion to the amplitude of a sine wave evaluated at the center of the same pulse. The gating signals as shown in figure 2.1 are generated by comparing an isosceles triangular carrier wave with a sine wave where the crossover points determine the instances of commutation. Except at low frequency the carrier is synchronized with the signal, and an odd integer (multiple of three, five etc.) ratio is maintained to maintain defined harmonic content. The fundamental output voltage can be varied by variation of the modulation index. It can be shown that if the modulation index is less than unity only carrier frequency harmonic with the fundamental frequency related side bands appear at the output [ 3,7,13-21]. It is observed that the area of each pulse corresponds approximately to the area under the sine wave between the adjacent midpoints of OFF periods. If  $\delta_i$  is the width of the  $i$ th pulse, then the output RMS voltage can be calculated by,

$$V_0 = V_s \sqrt{\sum_{i=1}^N \frac{\delta_i}{\pi}} \quad (2.1)$$

The fourier co-efficient of the output voltage can be determined according to the following equation.

$$A_n = 0$$

$$B_n = \sum_{i=1}^N \frac{2V_s}{n\pi} \sin \frac{n\delta_i}{2} \left[ \sin n \left( \alpha_i + \frac{\delta_i}{2} \right) - \sin n \left( \pi + \alpha_i + \frac{\delta_i}{2} \right) \right] \quad (2.2)$$

$$C_n = \sqrt{A_n^2 + B_n^2}$$

for  $n = 1, 3, 5, \dots$  etc

### 2-1 INTRODUCTION:

Dc-to-ac converters are inverters. The function of an inverter is to change a dc input voltage to a symmetrical ac output voltage of desired magnitude and frequency. The principle of a single phase inverter can be explained with figure 2.2. When only thyristor  $T_1$  is turned ON for a period  $T/2$ , the instantaneous voltage  $V_0$  across the

load is  $V_s/2$ . If the thyristor  $T_2$  is turned for a time  $T/2$ ,  $-V_s/2$  appears at the load. the logic circuit should be designed such that  $T_1$  and  $T_2$  are not turned ON at the same time. A variable output voltage can be obtained by varying the input dc voltage. On the other hand, if the dc input voltage is fixed and it is not controllable,

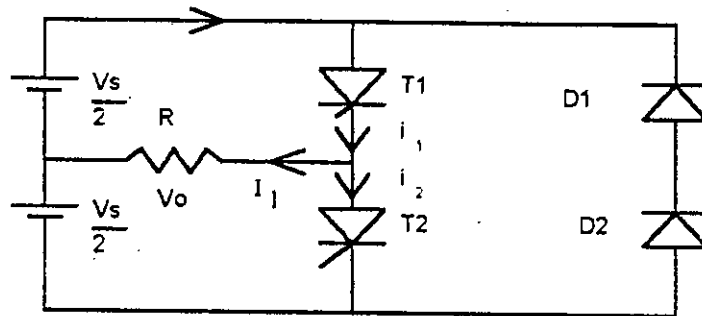


Figure 2.2 Half bridge inverter

a variable dc output voltage can be obtained by varying the gain of the inverter, which is normally accomplished by pulse-width modulation (PWM) within the inverter. This chapter describes the analysis of SPWM inverter waveforms. Modulation of static converter waveform is a common technique to minimize harmonics and to achieve simultaneous voltage and frequency control. High frequency switching reduces filter size of converters. Sine pulse-width modulation is a popular technique of obtaining required gating pulse for different converter configuration. Trapezoidal, Triangular and Delta modulation are other modulation processes which are used in converter control. A number of methods has so far been reported on the solution of switching points of SPWM inverter waveforms and their subsequent use in studies like harmonic determination and performance predictions of applications. However, most of the methods are mathematically involved and computationally complex. A new technique for the solution of switching points of SPWM inverter waveforms has been investigated in this research. This method involves solution of a simple algebraic equation. Spectral analysis has been carried out for single phase inverter waveforms. Proper choice of modulation parameters like modulation index, carrier frequency can be selected from the harmonic study presented in this chapter. On the basis of analysis features of SPWM technique as applied to the inverters are summarized.

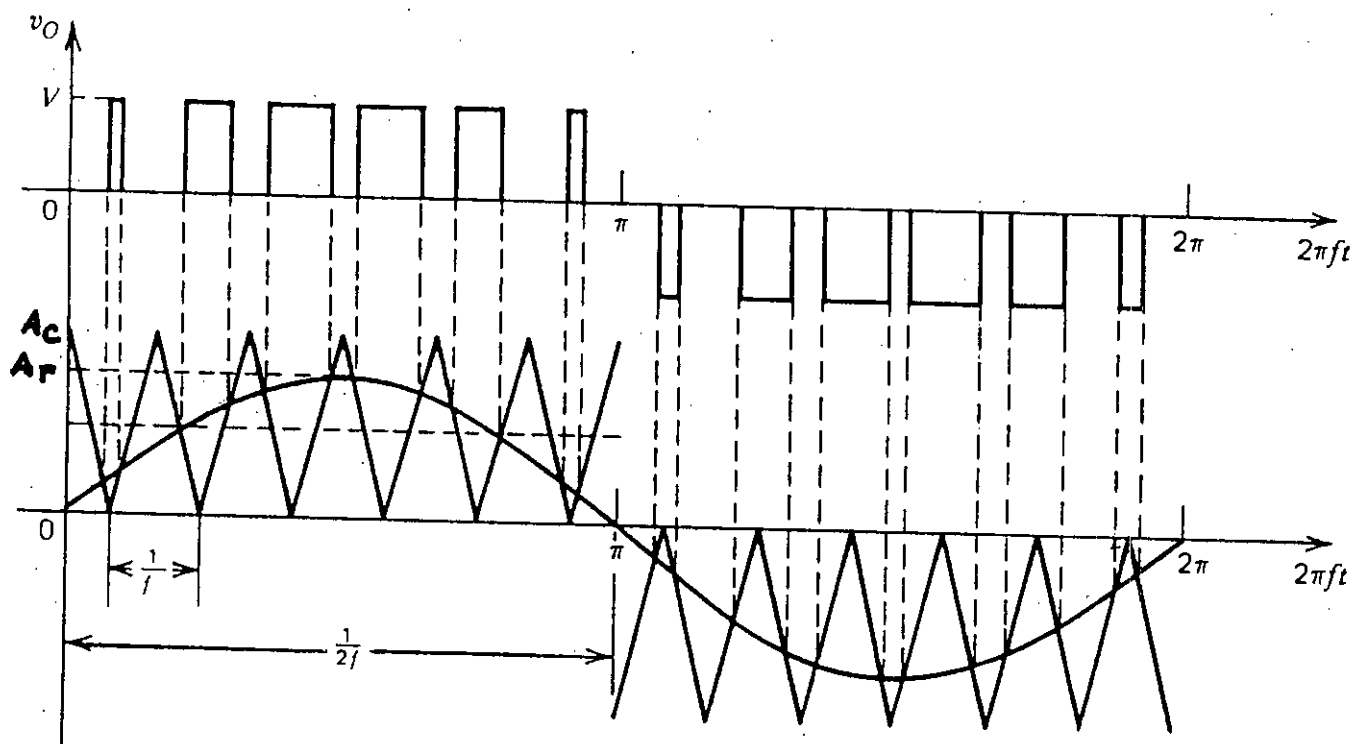


Fig. 21

Sinusoidal-pulse modulation ( $N=6$ ).





$$A_m \sin x_2 = S \left( x_2 - \frac{\pi i}{N} \right) \quad (2.4)$$

for falling edge slope we can write,

$$A_m \sin x_1 = -S \left( x_1 - \frac{\pi i}{N} \right) \quad (2.5)$$

Where  $S$  = Slope of the carrier wave and is given by,

$$S = \frac{A_c}{\frac{\pi}{2N}} = \frac{2NA_c}{\pi} \quad (2.6)$$

Adding equation (2.4) and (2.5), We get,

$$A_m (\sin x_1 + \sin x_2) = S (x_2 - x_1) \quad (2.7)$$

But  $x_2 - x_1 = \delta_i$  and  $\delta_i = k_i m \sin \theta_i$ . Comparing equation (2.7) with equation (2.3) We can write,

$$x_2 - x_1 = k_i m \sin \theta_i \quad (2.8)$$

from equation (2.7) and (2.8) we get,

$$K_i = \frac{A_r}{S} (\sin x_2 + \sin x_1) \quad (2.9)$$

Assuming  $\sin x_1 = \sin x_2 = \sin \theta_i$ , which is absolutely correct for regular sampled PWM, hence we can write,

$$\sin x_1 + \sin x_2 = 2 \sin \theta_i$$

By simplifying equations (2.6)- (2.9) we can write ,

$$K_i = \frac{\pi i}{N} \quad (2.10)$$

Putting the value of  $K_i$  the width of the pulses of the SPWM waveforms can be expressed as,

$$\delta_i = \frac{\pi}{N} m \sin \theta_i \quad (2.11)$$

The switching instance can be expressed as,

$$t_{i+1} = \frac{\pi i}{N} + \frac{\delta_i}{2} \quad (2.12)$$

$$t_{i-1} = \frac{\pi i}{N} - \frac{\delta_i}{2} \quad (2.13)$$

### 2-3 INVERTER WAVEFORM SYNTHESIS:

The study of the performance of the inverter and their applications requires knowledge of harmonics of the modulated waveforms. Waveform generated by sine pulse-width modulation schemes has been analyzed by discrete Fourier transform to find harmonics of inverter waveforms. The formulation of this method of simulation and results of waveforms of single phase inverter and their spectrum are presented. It is a well known fact that pulse width modulated waveforms contain sub harmonics due to the concurrent presence of modulating and carrier wave frequencies in the modulated waves [ 30-45,47,49]. The conventional Fourier series analysis of such waveforms determine only the frequency components which are multiples of the modulating frequency and ignore the presence of sub harmonics. Fourier transform method takes into account continuous spectrum and is not dependent on the period of the waveform under study. As a result transform method would give sub harmonics of PWM waveforms in its continuous spectrum. The analysis is based on representing PWM waves by gating function and subsequent FFT on the waveforms. In this approach, the pulse width modulated waveforms are defined and simulated by gate functions and the switching points of the waveforms. The switching points are obtained from the solution of a equation (2.12) and (2.13) developed in the previous section. The simulated waveforms are then sampled for discrete Fourier transform analysis. DFT approach also has advantage of simplicity and adaptation in on-line harmonic determination of the modulated waveforms by microcomputer. The method for On-line harmonic estimation was not studied in this work.

### 2-4 NECESSITY OF DFT ANALYSIS OF PWM WAVEFORMS:

Theoretical works on pulse width modulated inverter waveform synthesis are based on the ordinary Fourier series method [17-20, 26-41]. Spectral information of these converter waveform is necessary for their design and performance study. Very few papers, however, dealt with computer processing of actual waveforms for validation

of simulated waveforms. Also harmonic analysis by the Fourier series does not account for the sub harmonics present in the pulse width modulated waveforms. These sub harmonics are inherent in all pulse width modulated waveforms because they have components of modulating waveform and carrier waveform [11-16]. Therefore in order to study the nature of the modulated waveforms an approach is taken towards the development of an workable method of spectral analysis based on discrete Fourier transform (DFT).

## 2-5 DISCRETE FOURIER TRANSFORM (DFT) [ 57 ]:

The Fourier transform pair for continuous signal can be written as,

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (2.14)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (2.15)$$

for  $-\infty < f < \infty$  and  $-\infty < t < \infty$ ; and  $j = \sqrt{-1}$

The analogous discrete Fourier transform pair that apply to the sampled versions of these function can be written as,

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{j2\pi mn/N} \quad (2.16)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(K) e^{-j2\pi mn/N} \quad (2.17)$$

Both  $X(K)$  and  $x(n)$  are complex series. When the expression  $e^{j2\pi/N}$  is replaced by the term  $W_N$ , the DFT pairs takes the form

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad (2.18)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(K) W_N^{-kn} \quad (2.19)$$

Equation 2.18 and equation (2.19 ) can be denoted by transform pair

$$X(m) = D [ x(n)] \tag{2.20}$$

$$x(n) = D^{-1} [ X (m)] \tag{2.21}$$

With the subscripts omitted from  $WN$  , the equation (2.18 ) can be written as,

$$\begin{aligned} X (0) &= x (0) W^0 + x (1) W^0 + \dots + x (N-1) W^0 \\ X (1) &= x (0) W^0 + x (1) W^1 + \dots + x (N-1) W^{N-1} \\ &\dots \tag{2.22} \\ &\dots \end{aligned}$$

$$X (N-1) = x (0) W^0 + x (1) W^{N-1} + \dots + x (N-1) W^{N-1^2}$$

Where  $X(0), X(1), \dots, X(N - 1)$  are the sample values of the signal  $x(t)$  at sampling instances. The evaluation of the above form of DFT can be done either by a direct method which is computationally time consuming or by Fast Fourier Transforms (FFT) available to evaluate DFT [35,41-49].

**2-6 DISCRETE FOURIER TRANSFORM OF PWM WAVE FORMS:**

Discrete Fourier transform has been carried out on the sampled simulated wave forms of the PWM inverter. The discrete Fourier transform pair representing the continuous transform has been used in the detection of all frequency including the sub harmonic of the modulated signal. To perform DFT analysis it is necessary to define the output waveform interms of switching function (defined in gating functions). Gating function can be defined as,

$$g (t, t_0, t_1) = U ( t, t_0 ) - U( t, t_1 )$$

Where,

$g (t, t_0, t_1)$  gate function starting at  $t_0$  and ending at  $t_1$

$U ( t, t_0 )$  Unit step function starting at  $t_0$ .

$U(t, t_1)$  Unit step function starting at  $t_1$   
 $= 1$  for  $t \geq t_1$   
 $= 0$  elsewhere.

Different waveforms thus obtained can be expressed as follows

$$sw1 = g(t, a(i), b(i)); \quad (2.23)$$

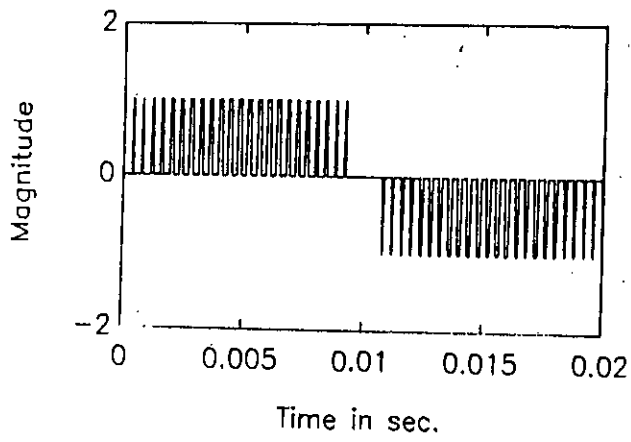
$$sw2 = -g(t, a(i) + T/2, b(i) + T/2). \quad (2.24)$$

where,  $a(i) = \theta(i) - \frac{t(i)}{2}$  and  $b(i) = \theta(i) + \frac{t(i)}{2}$

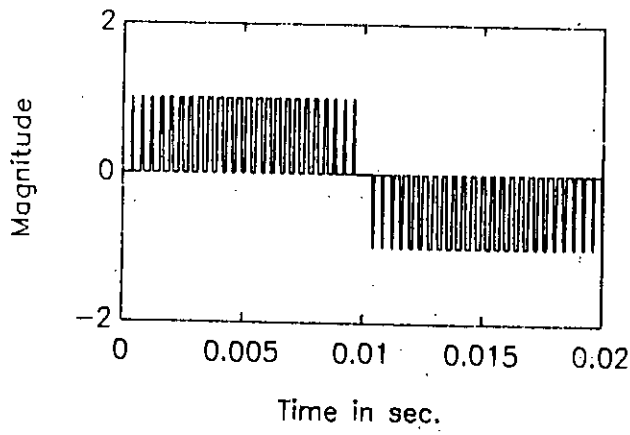
The first step in evaluating spectral content of the modulated waveforms is to simulate the waveforms at different frequencies, modulation index and number of carrier waves per half cycle using the switching points obtained by solving the equation developed in the previous section. The simulation was done using MATLAB software and the program to simulate the waveforms are included in Appendix-1 and Appendix-2. Typical PWM waveforms thus obtained are shown in figure 2.4.1 to figure 2.4.8 respectively. The magnitude spectra obtained for waveforms of figure 2.4.1 to figure 2.4.8 are presented in figures 2.5.1 to 2.5.8 respectively. The summary of the result of the spectral analysis are presented in table 2.1 to 2.4.

No. of carrier pulse per half cycle.	Harmonics occurs at frequency in Hz	Magnitude in P.U			
10	1000	.05	.36	.36	.05
10	2000	.08	.17	.18	.07
10	3000	0.0	.905	.10	0.0
20	1000	0.0	.05	.07	0.0
20	2000	0.0	.375	.35	0.0
20	3000	0.0	.05	.05	0.0
30	1000	0.0	.05	.07	0.0
30	2000	0.0	.07	.06	0.0
30	3000	0.0	.35	.36	0.0

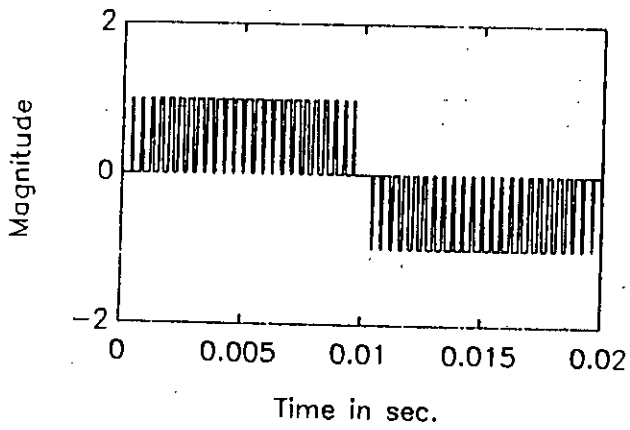
Table 2.1. Result of spectral analysis of SPWM wave for  $f = 50$ ,  $m = .4$  and  $N = 10, 20$  and  $30$  respectively.



(a)

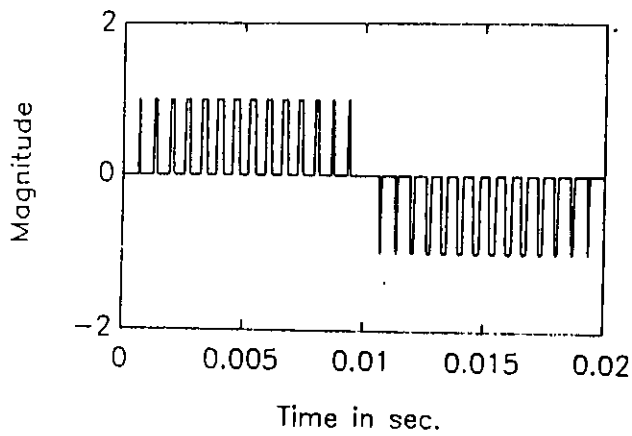


(b)

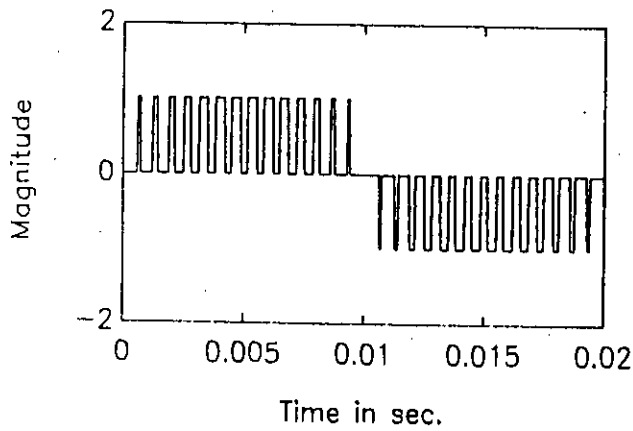


(c)

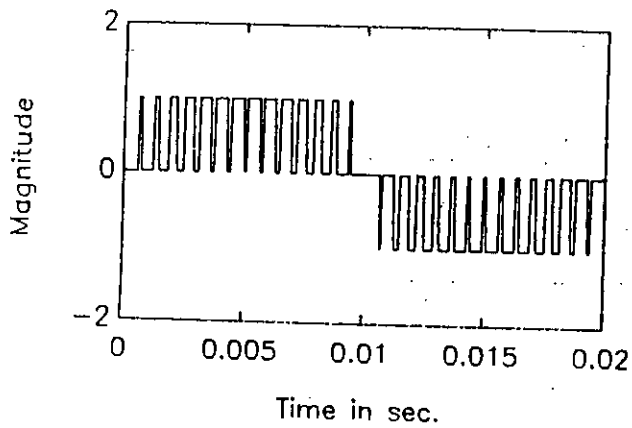
Figure 2.4.1(a) Simulated PWM waveform at  $f = 50\text{Hz}$ ,  $N = 25$  and  $m = .4$   
 (b) Simulated PWM waveform at  $f = 50\text{Hz}$ ,  $N = 25$  and  $m = .6$   
 (c) Simulated PWM waveform at  $f = 50\text{Hz}$ ,  $N = 25$  and  $m = .8$



(a)

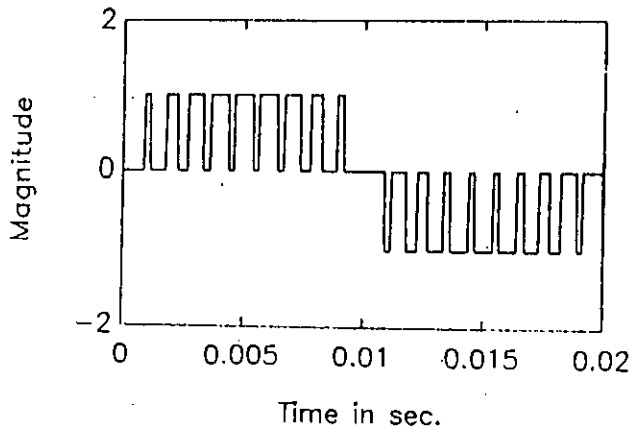


(b)

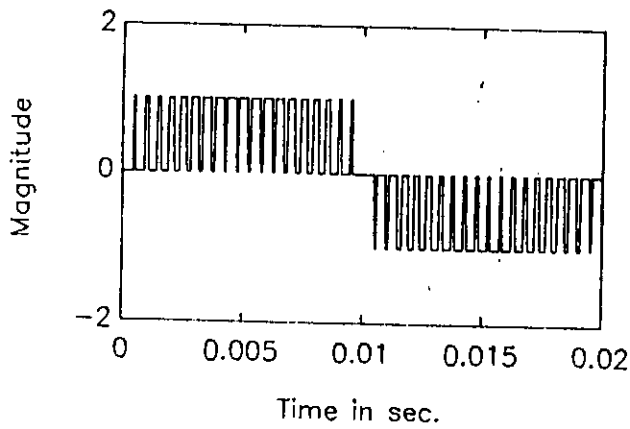


(c)

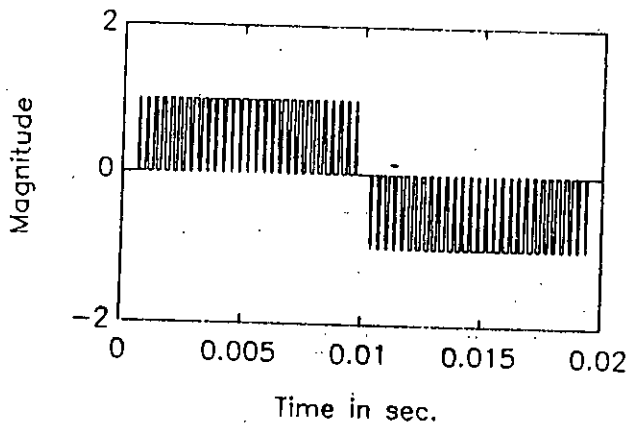
- Figure 2.4.2(a) Simulated PWM waveform at  $f = 50\text{Hz}$ ,  $N = 15$  and  $m = .4$   
 (b) Simulated PWM waveform at  $f = 50\text{Hz}$ ,  $N = 15$  and  $m = .6$   
 (c) Simulated PWM waveform at  $f = 50\text{Hz}$ ,  $N = 15$  and  $m = .8$



(a)



(b)

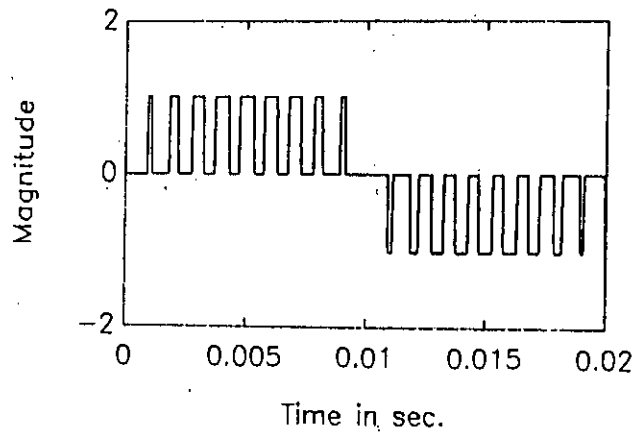


(c)

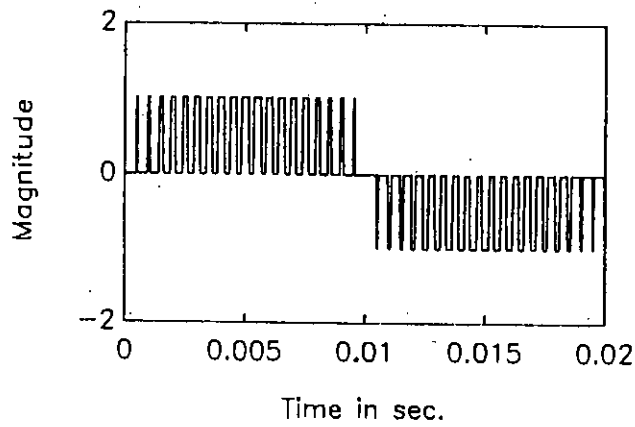
Figure 2.4.3

- (a) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 10$  and  $m = .8$
- (b) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 20$  and  $m = .8$
- (c) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 30$  and  $m = .8$

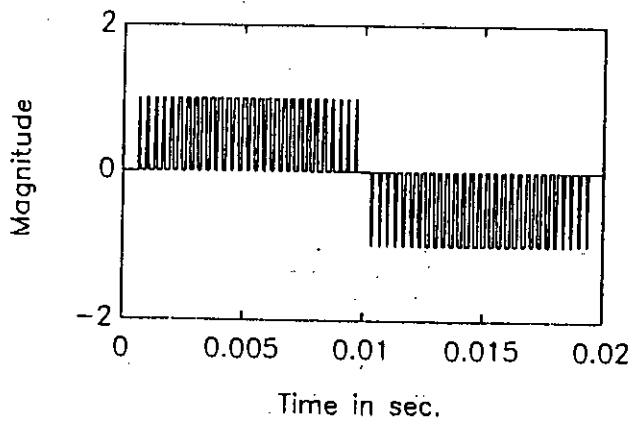




(a)



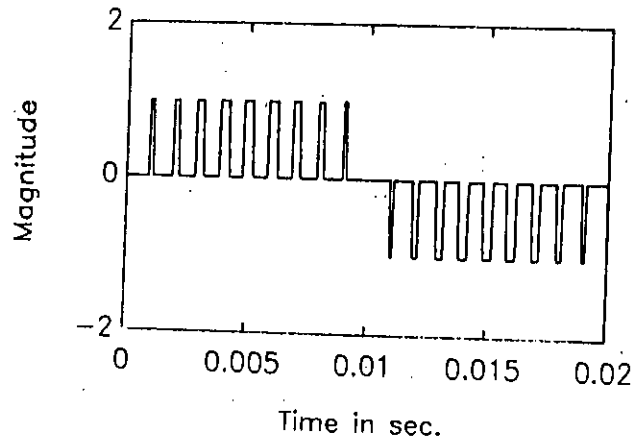
(b)



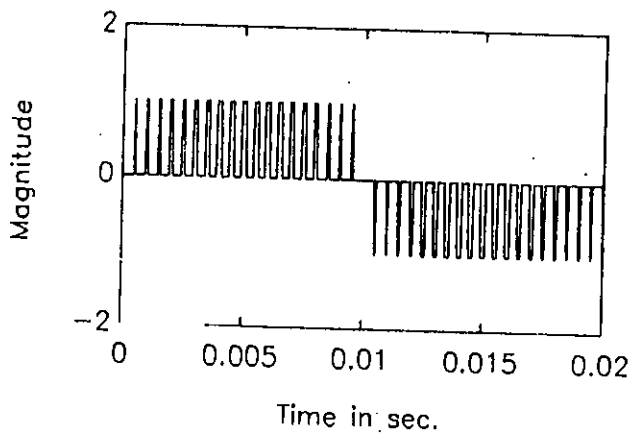
(c)

Figure 2.4.4

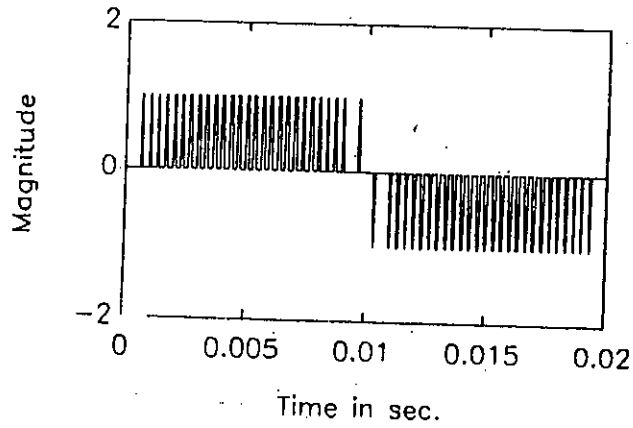
- (a) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 10$  and  $m = .6$
- (b) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 20$  and  $m = .6$
- (c) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 30$  and  $m = .6$



(a)



(b)



(c)

Figure 2.4.5

- (a) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 10$  and  $m = .4$
- (b) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 20$  and  $m = .4$
- (c) Simulated pwm waveform at  $f = 50\text{Hz}$ ,  $N = 30$  and  $m = .4$

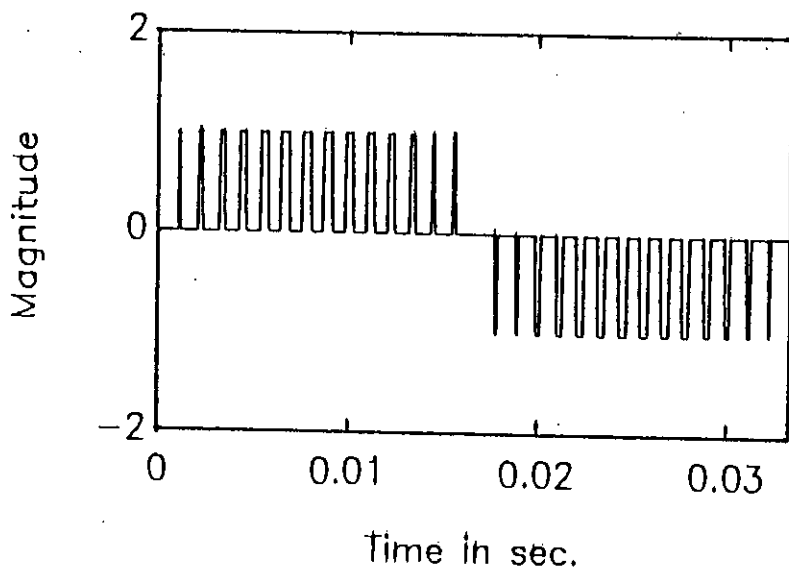


Figure 2.4.6(a) Simulated PWM waveform at  $f = 70\text{Hz}$ ,  $N = 15$  and  $m = .4$

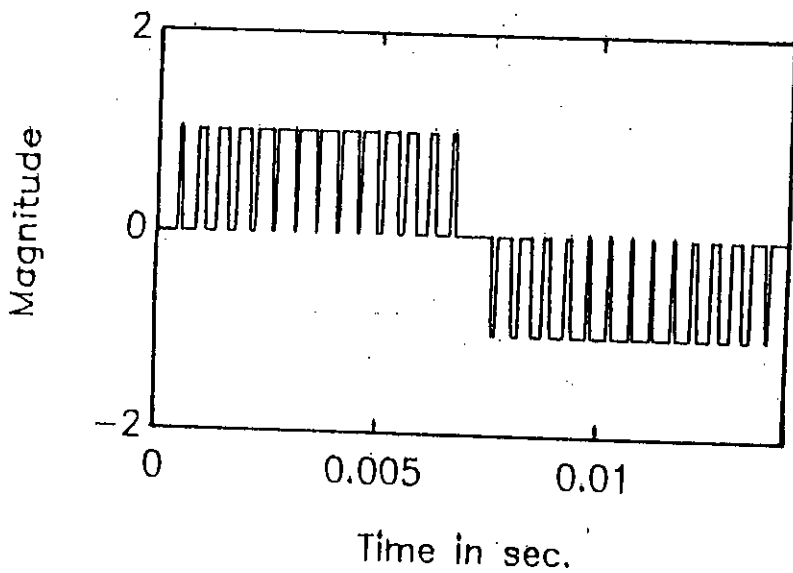


Figure 2.4.6(b) Simulated PWM waveform at  $f = 70\text{Hz}$ ,  $N = 15$  and  $m = .9$

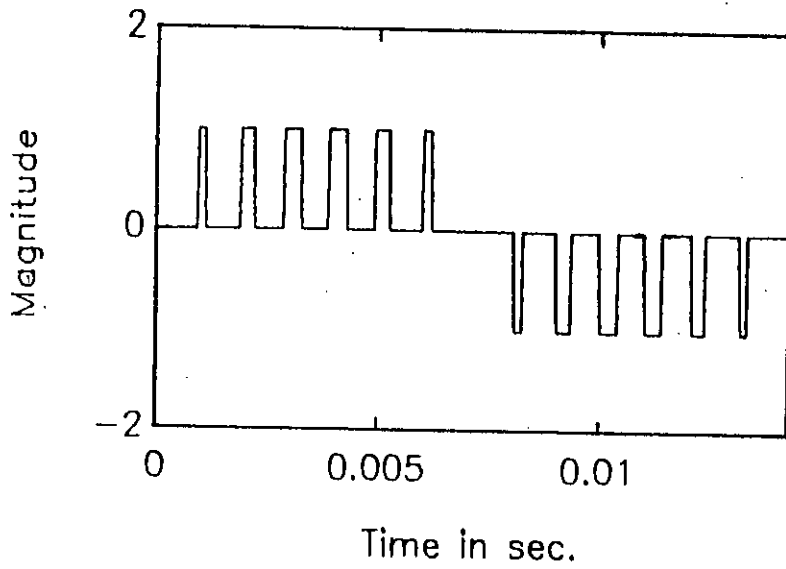


Figure 2.4.7(a) Simulated PWM waveform at  $f = 70\text{Hz}$ ,  $N = 7$  and  $m = .4$

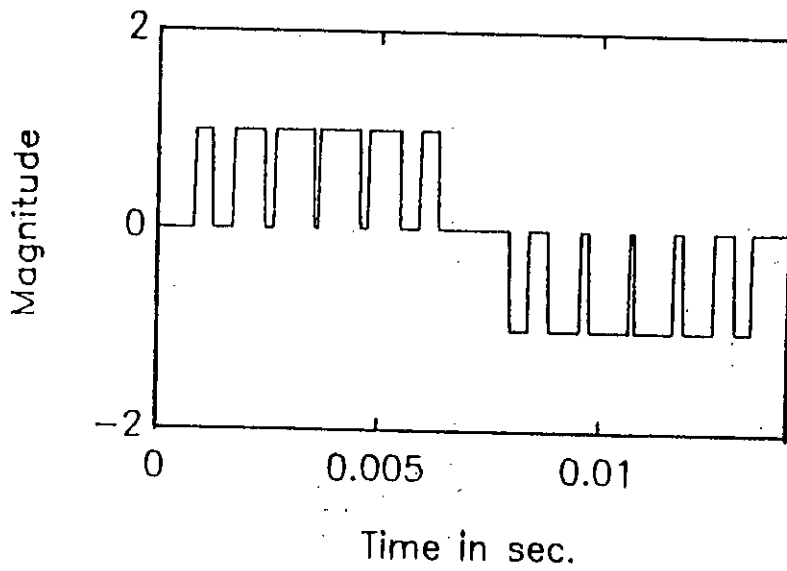


Figure 2.4.7(b) Simulated PWM waveform at  $f = 70\text{Hz}$ ,  $N = 7$  and  $m = .9$

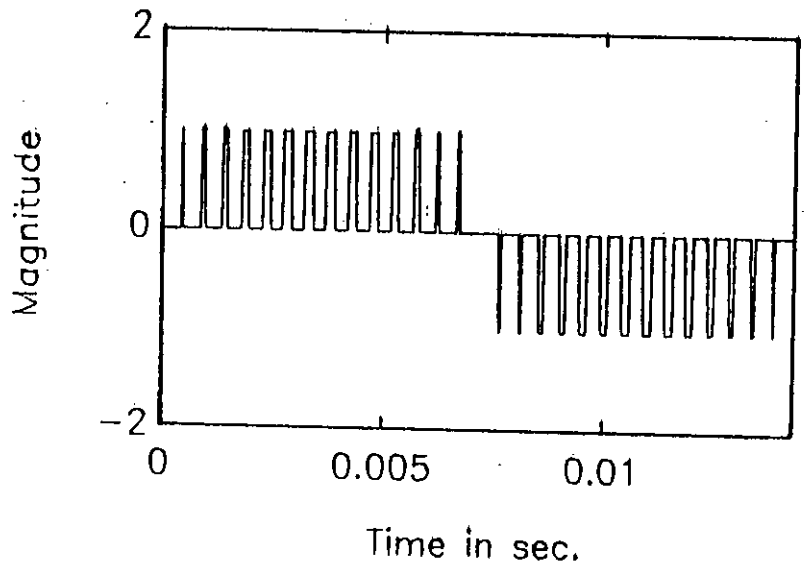


Figure 2.4.8(a) Simulated PWM waveform at  $f = 30\text{Hz}$ ,  $N = 15$  and  $m = .4$

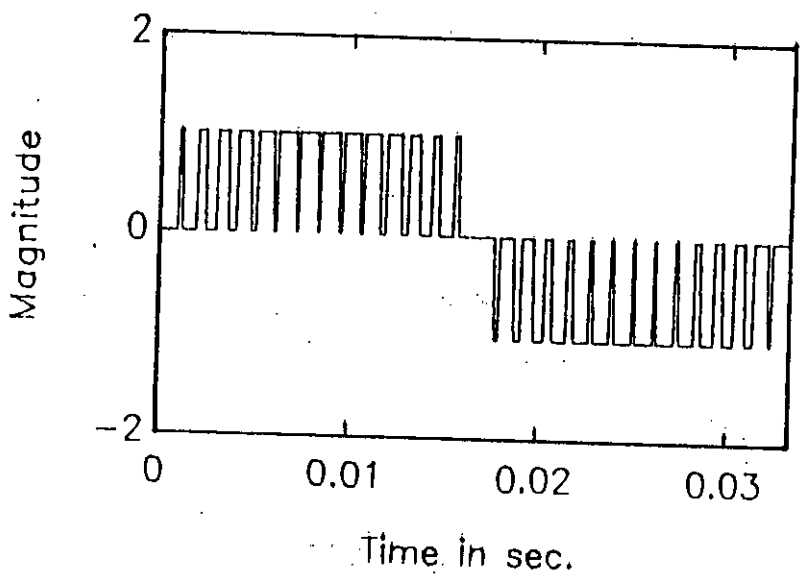
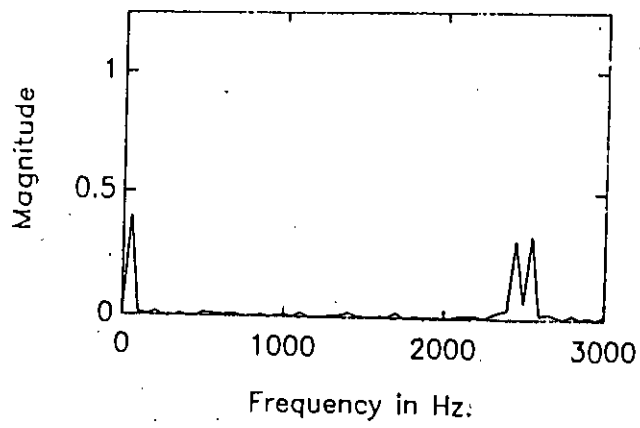
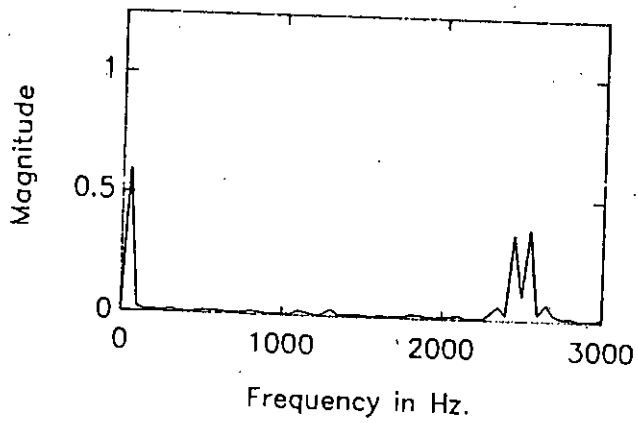


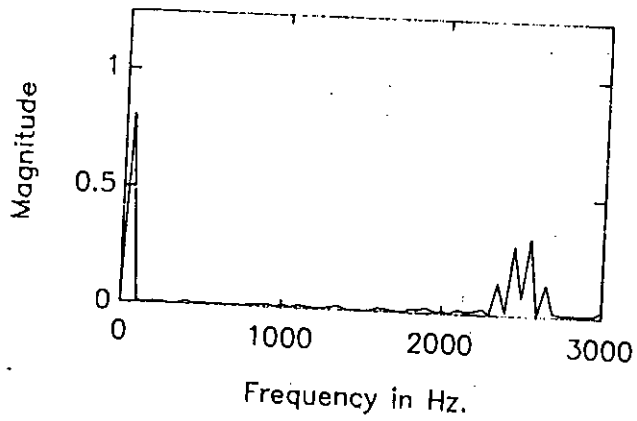
Figure 2.4.8(b) Simulated PWM waveform at  $f = 30\text{Hz}$ ,  $N = 15$  and  $m = .9$



(a)



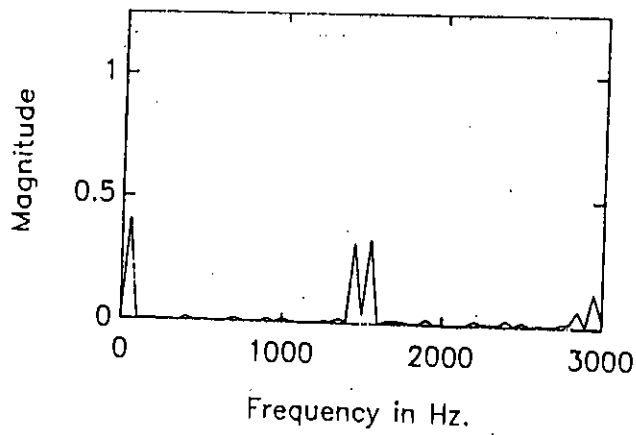
(b)



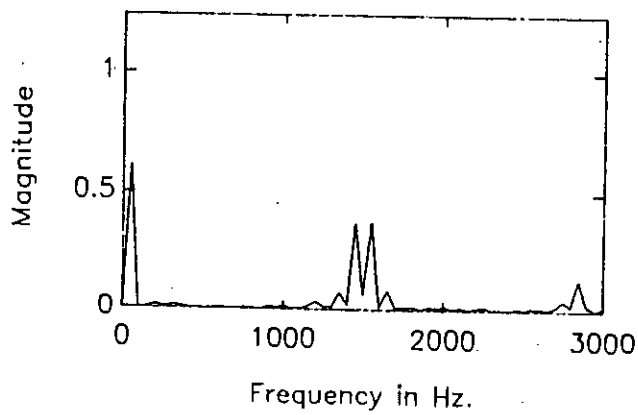
(c)

Figure 2.5.1

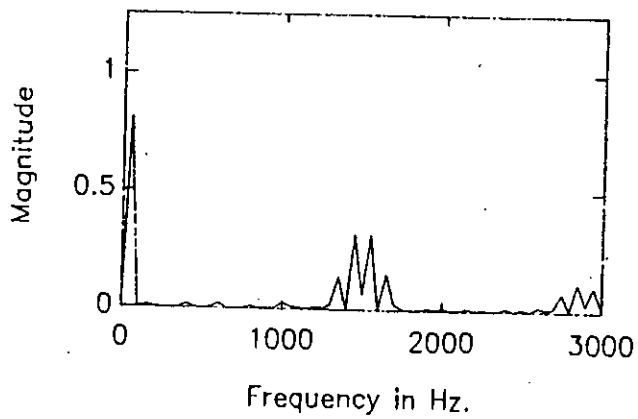
- (a) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 25$  and  $m = .4$
- (b) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 25$  and  $m = .6$
- (c) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 25$  and  $m = .8$



(a)



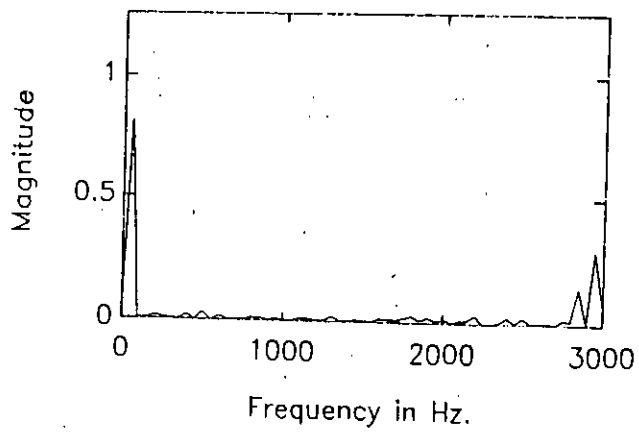
(b)



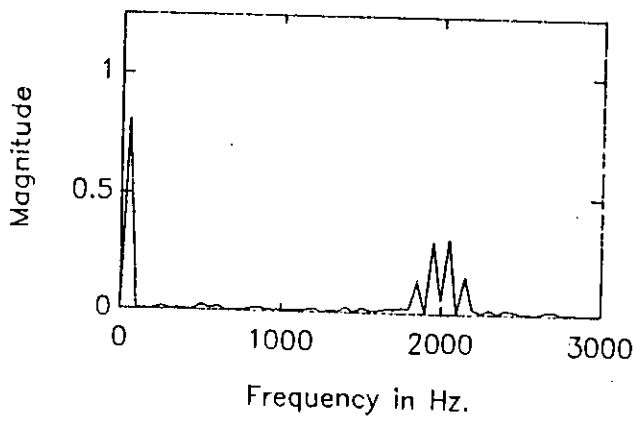
(c)

**Figure 2.5.2**

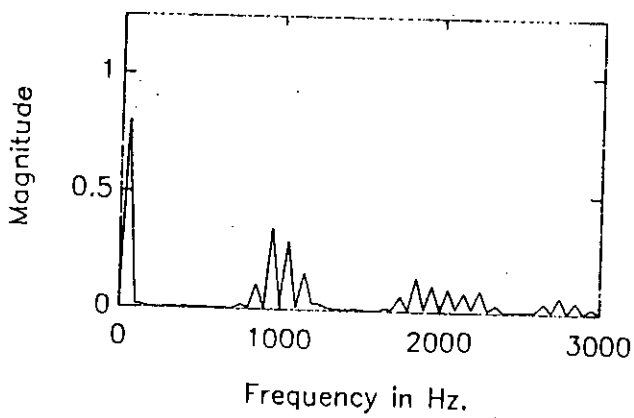
- (a) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 15$  and  $m = .4$
- (b) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 15$  and  $m = .6$
- (c) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 15$  and  $m = .8$



(a)



(b)

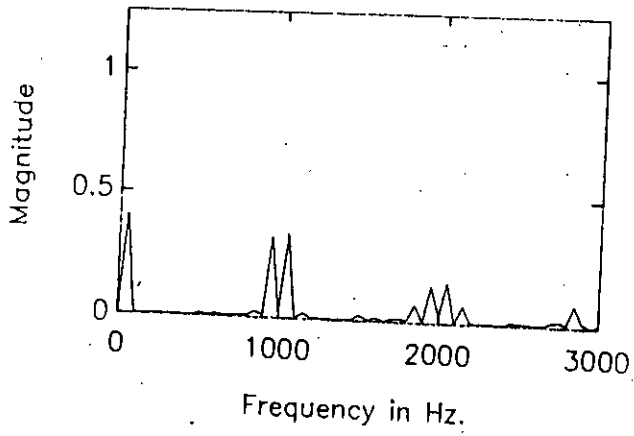


(c)

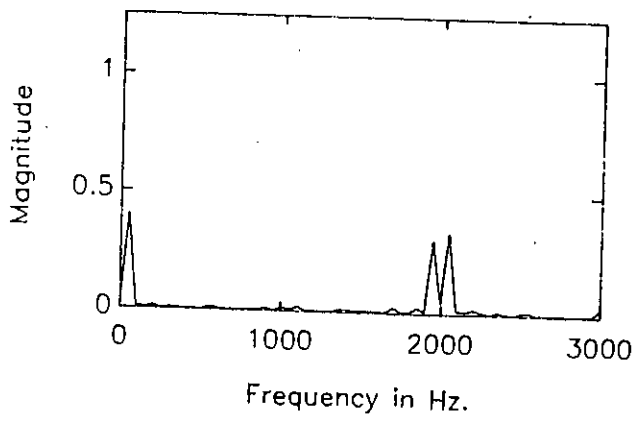
Figure 2.5.3

- (a) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 30$  and  $m = .8$
- (b) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 20$  and  $m = .8$
- (c) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 10$  and  $m = .8$

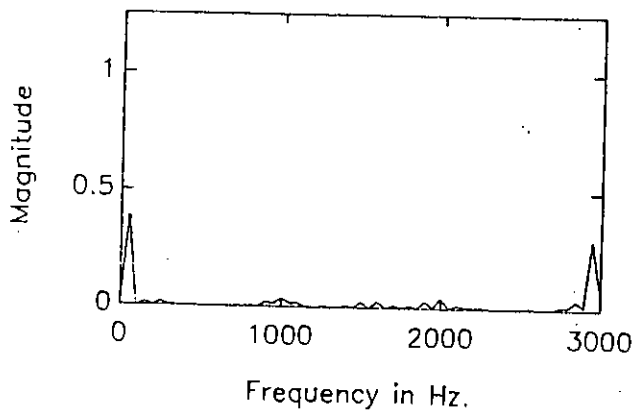




(a)



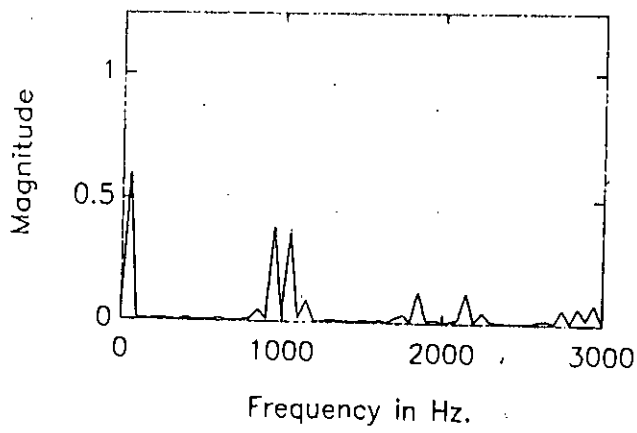
(b)



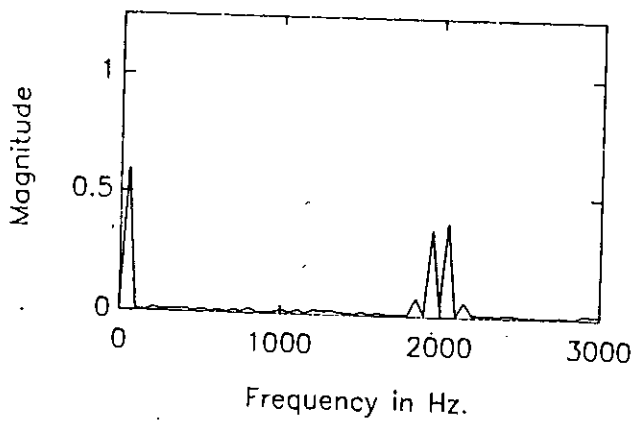
(c)

Figure 2.5.4

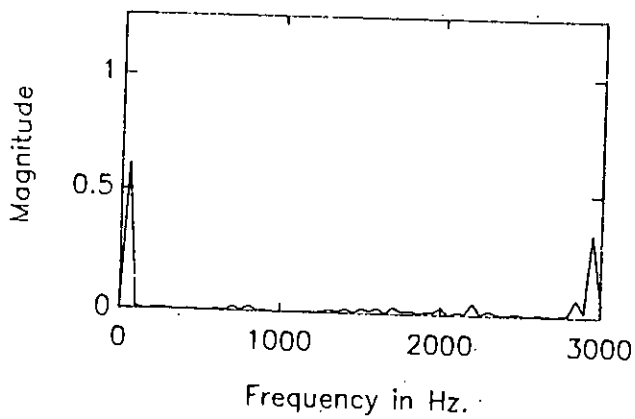
- (a) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 10$  and  $m = .4$
- (b) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 20$  and  $m = .4$
- (c) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 30$  and  $m = .4$



(a)



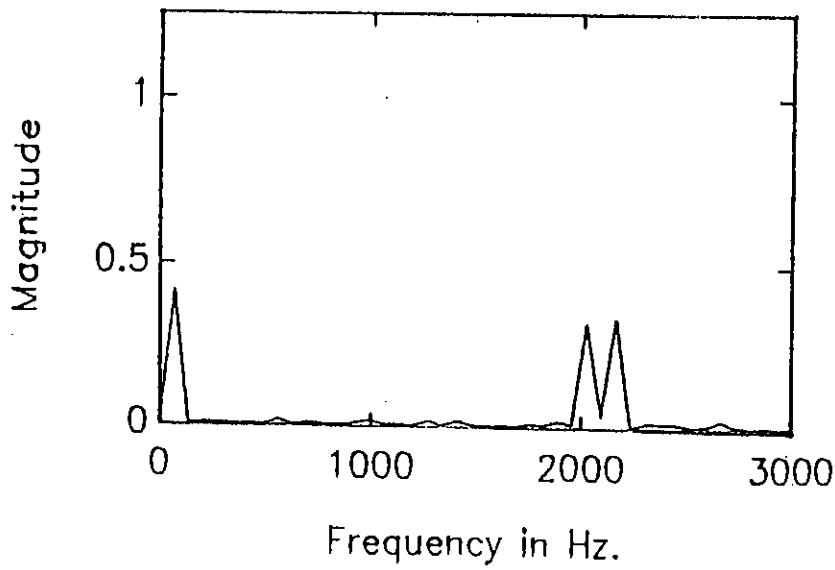
(b)



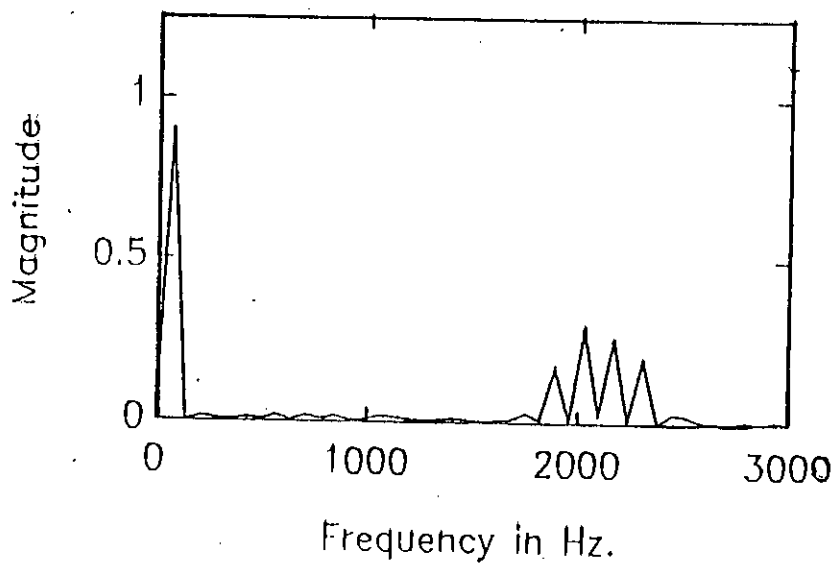
(c)

Figure 2.5.5

- (a) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 10$  and  $m = .6$
- (b) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 20$  and  $m = .6$
- (c) Spectrum of pwm waveform at  $f = 50\text{Hz}$ ,  $N = 30$  and  $m = .6$



(a)



(b)

Figure 2.5.6(a) Spectrum of PWM waveform at  $f = 70\text{Hz}$ ,  $N = 15$  and  $m = .4$   
 (b) Spectrum of PWM waveform at  $f = 70\text{Hz}$ ,  $N = 15$  and  $m = .9$

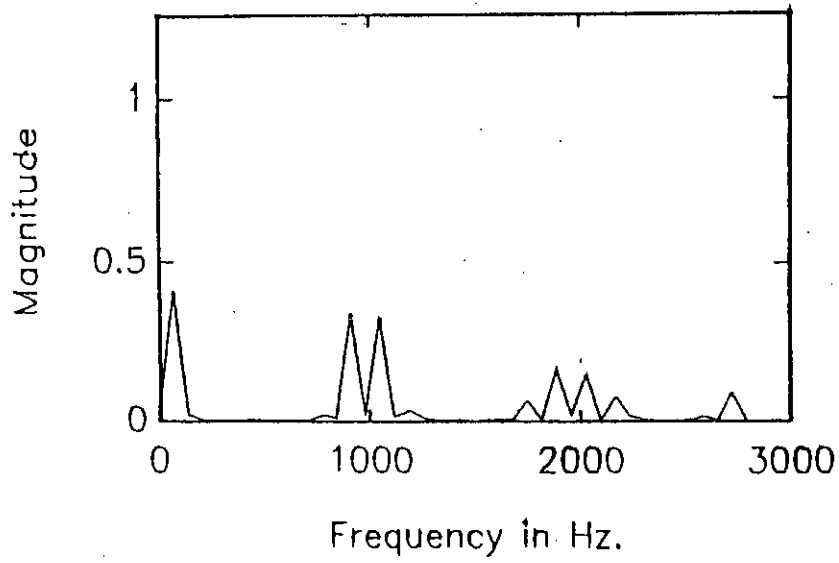


Figure 2.5.7(a) Spectrum of PWM waveform at  $f = 70\text{Hz}$ ,  $N = 7$  and  $m = .4$

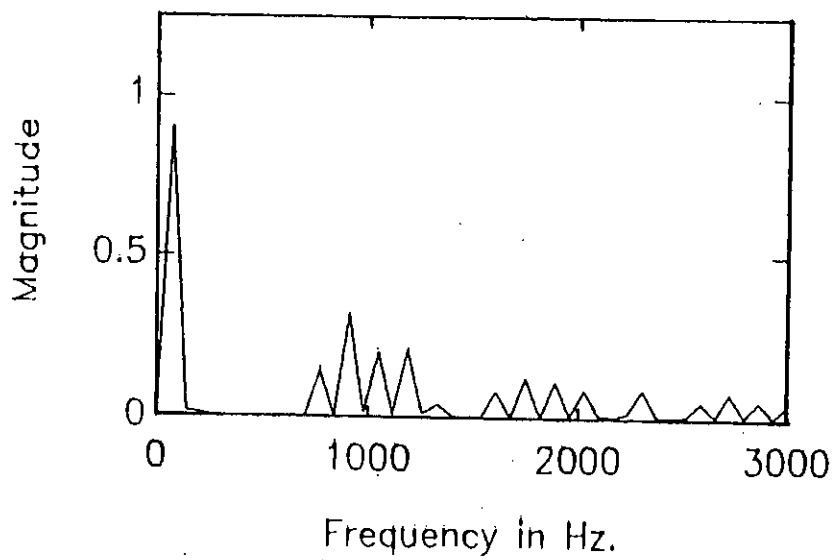


Figure 2.5.7(b) Spectrum of PWM waveform at  $f = 70\text{Hz}$ ,  $N = 7$  and  $m = .9$

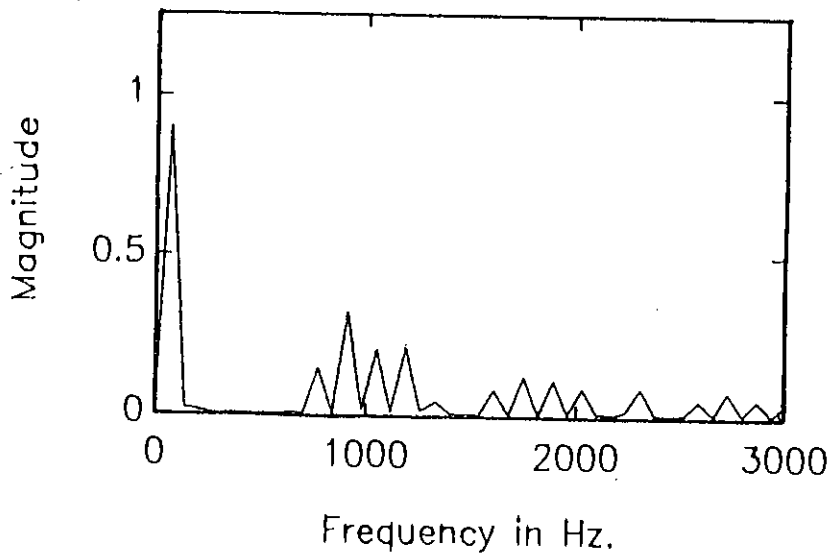
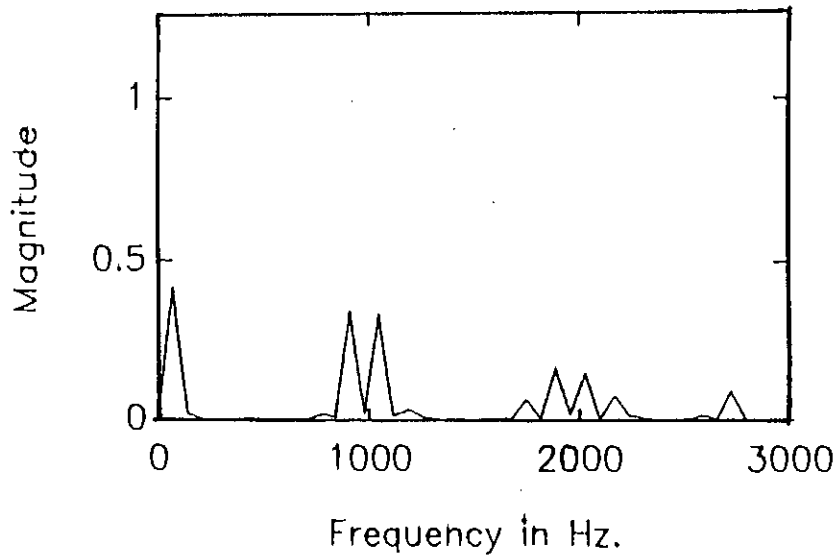


Figure 2.5.8(a) Spectrum of PWM waveform at  $f = 30\text{Hz}$ ,  $N = 15$  and  $m = .4$   
 (b) Spectrum of PWM waveform at  $f = 30\text{Hz}$ ,  $N = 15$  and  $m = .9$

No. of carrier pulse per half cycle.	Harmonics occurs at frequency in Hz	Magnitude in P.U			
10	1000	.05	.375	.35	.07
10	2000	.00	.10	.10	.07
10	3000	0.0	.07	.05	0.0
20	1000	0.0	0.0	0.0	0.0
20	2000	0.5	.375	.35	0.0
20	3000	0.0	.05	.05	0.0
30	1000	0.0	.05	.07	0.0
30	2000	0.0	.07	.06	0.0
30	3000	0.0	.35	.35	0.0

Table 2.2. Result of spectral analysis of SPWM wave for  $f = 50$ ,  $m = .6$  and  $N = 10, 20$  and  $30$  respectively.

No. of carrier pulse per half cycle.	Harmonics occurs at frequency in Hz	Magnitude in P.U			
10	1000	.105	.375	.35	.12
10	2000	.08	.10	.10	.08
10	3000	0.5	.07	.07	0.5
20	1000	0.0	0.0	0.0	0.0
20	2000	0.1	.375	.35	0.12
20	3000	0.0	.05	.05	0.0
30	1000	0.0	.05	.07	0.0
30	2000	0.0	.07	.06	0.0
30	3000	0.0	.375	.35	0.0

Table 2.3. Result of spectral analysis of SPWM wave for  $f = 50$ ,  $m = .8$  and  $N = 10, 20$  and  $30$  respectively.

Modulation Index	Harmonics occurs at frequency in Hz	Magnitude in P.U			
.4	1000	.105	.375	.35	.12
.4	2000	.08	.10	.10	.08
.4	3000	0.5	.07	.07	0.5
.6	1000	0.0	0.0	0.0	0.0
.6	2000	0.1	.375	.35	0.12
.6	3000	0.0	.05	.05	0.0
.8	1000	0.0	.05	.07	0.0
.8	2000	0.0	.07	.06	0.0
.8	3000	0.0	.375	.35	0.0

Table 2.4. Result of spectral analysis of SPWM wave for  $f = 50$  Hz,  $N = 15$  carrier wave per half cycle and  $m = .4, .6$  and  $.8$  respectively.

The following properties of the PWM waveforms are evident from spectral study.

- For PWM waveforms, fundamental voltage increases with increasing modulation index when other parameters are maintained constant. The linearity remains in the modulation mode of operation. Variation of modulation index thus provides a means for voltage variation. Beyond the base frequency the modulated output goes into the square wave mode of operations and remains in this mode for higher operating frequencies. The fundamental component of square wave is constant and is given by  $(4V_s/\pi)$  volt.
- Increase in carrier frequency causes the dominant harmonics to occur at higher frequency which is an integer multiple of the fundamental operating frequency. This trend is shown in figures 2.6.1 to 2.6.5. The increase in carrier frequency is limited by switching frequency of the static devices of the inverter. Number of switching of these device increase as carrier frequency increases. As a result carrier frequency increase is limited by the highest switching frequency of the static devices of the converter.

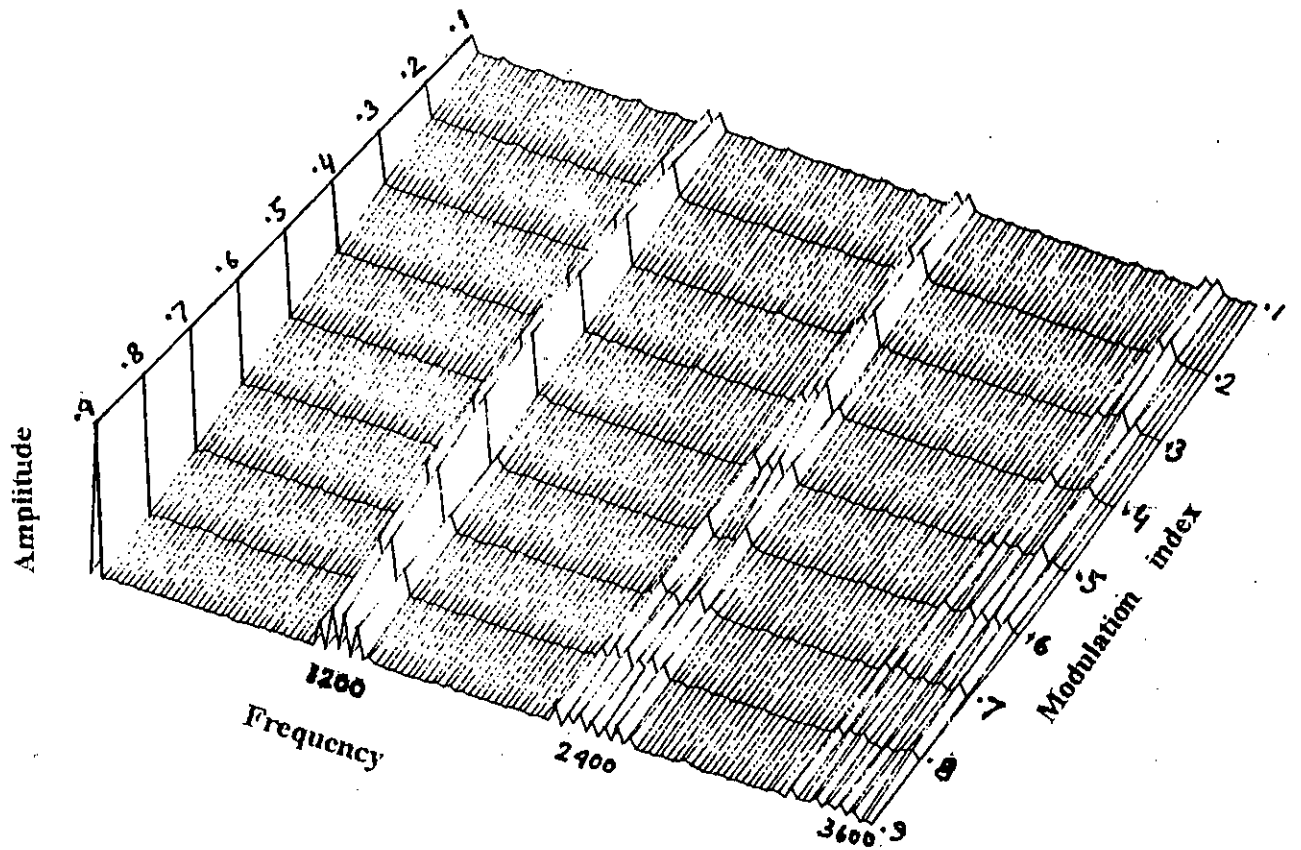


Figure 2.6.1(a) Three dimensional spectrum of spwm waveform at  $f = 30$  &  $N = 20$

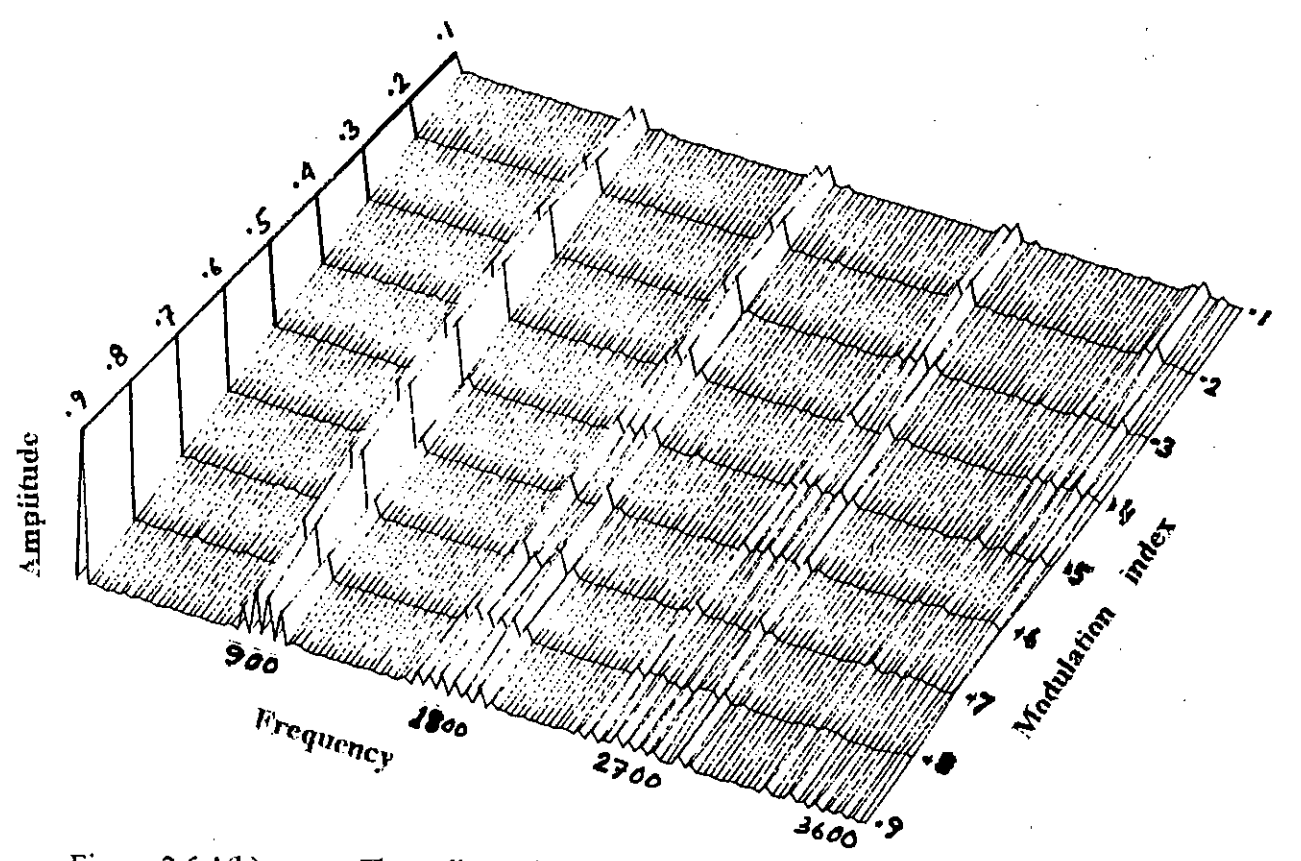


Figure 2.6.1(b) Three dimensional spectrum of pwm waveform at  $f = 30$  &  $N = 15$



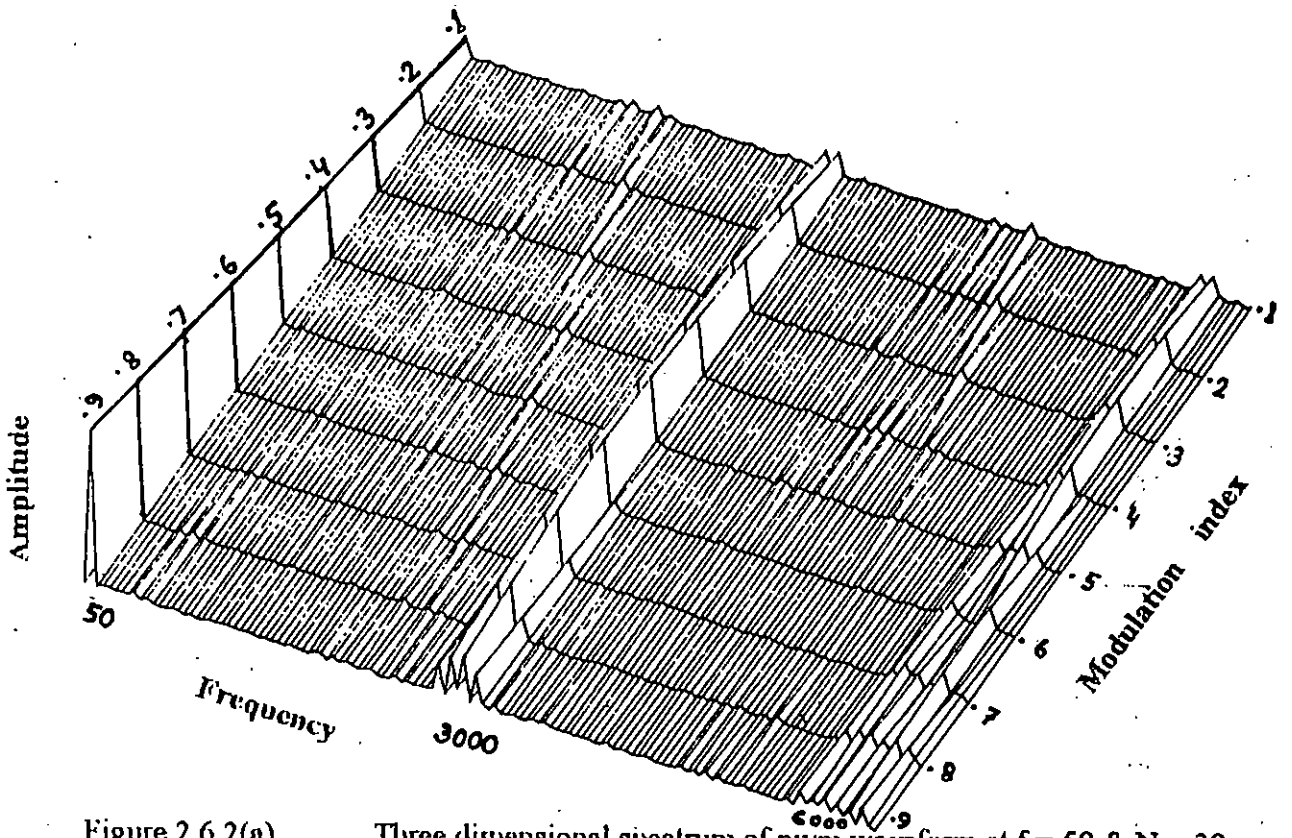


Figure 2.6.2(a) Three dimensional spectrum of pwm waveform at  $f = 50$  &  $N = 30$ .

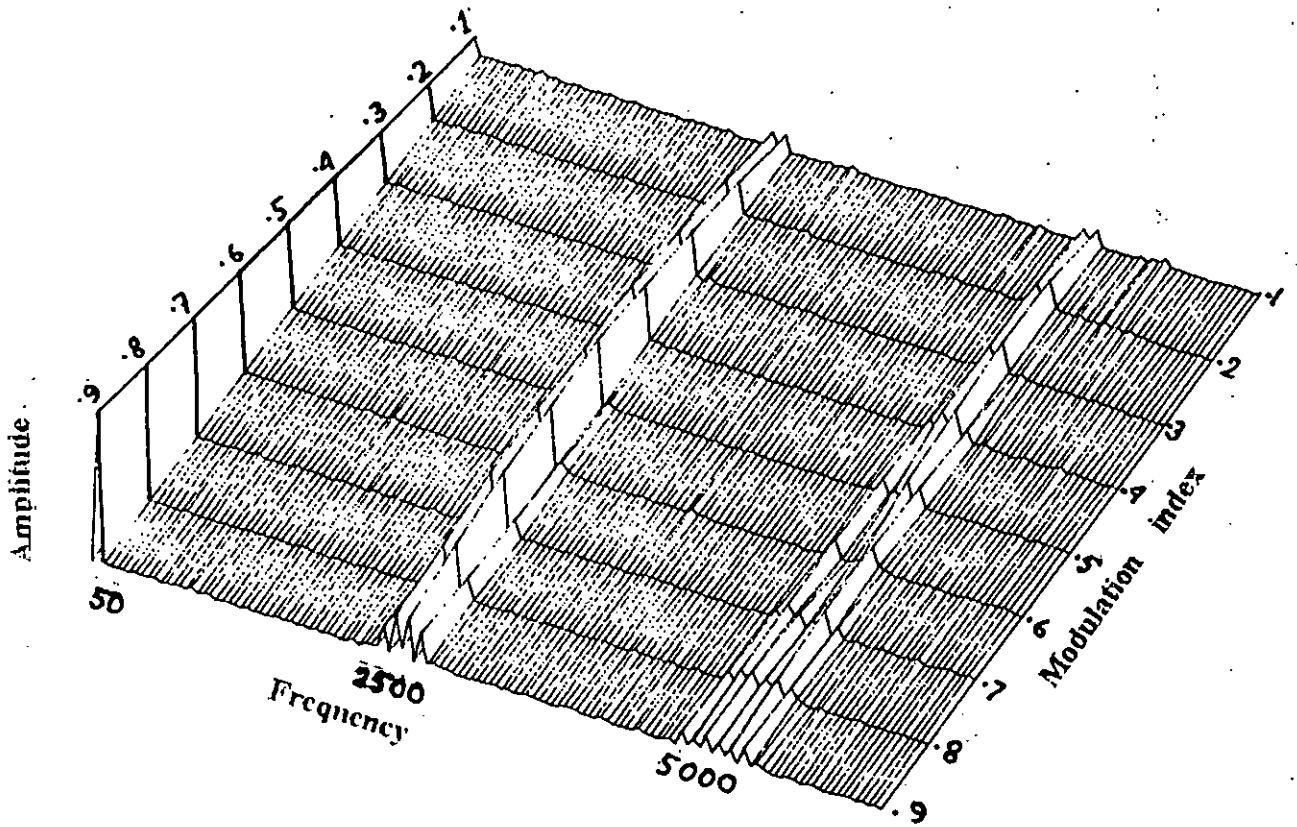


Figure 2.6.2(b) Three dimensional spectrum of pwm waveform at  $f = 50$  &  $N = 25$ .

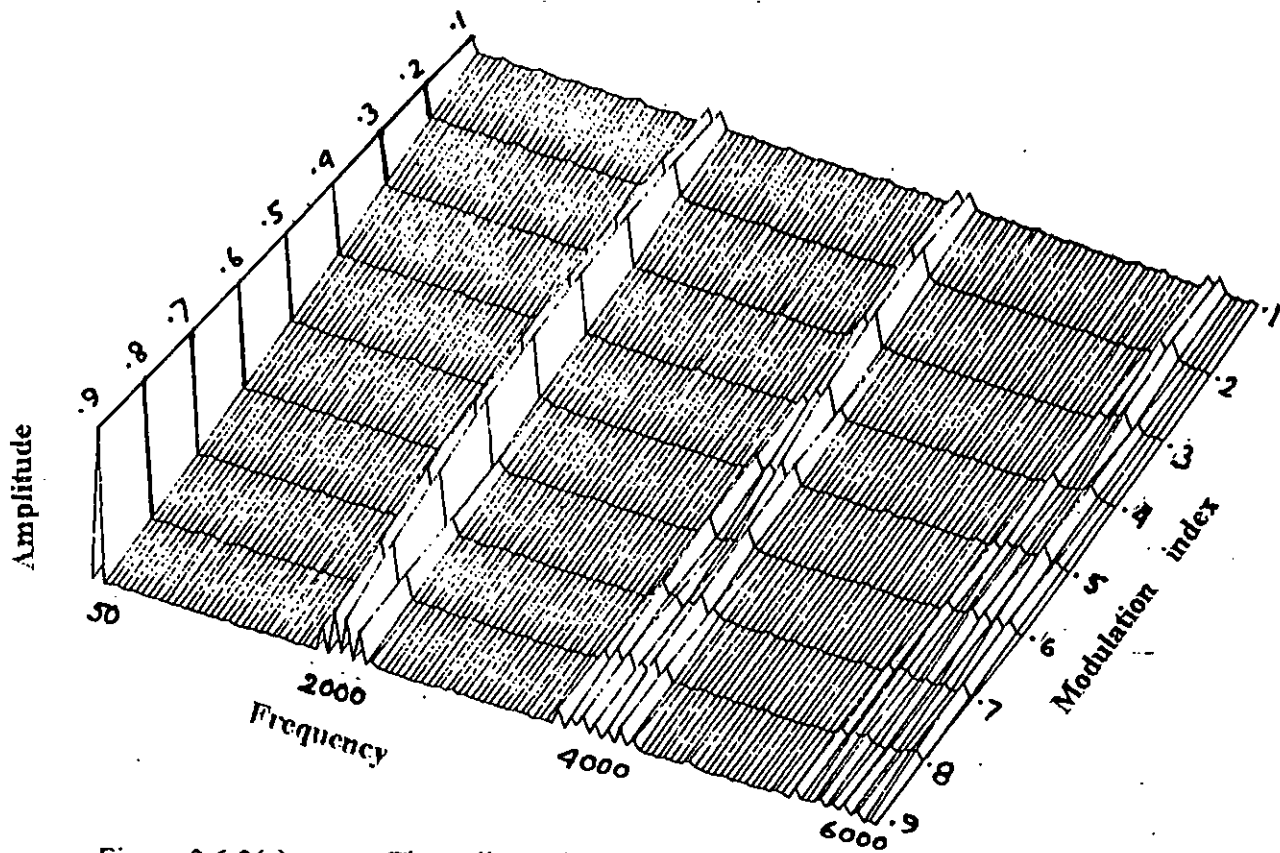


Figure 2.6.3(a) Three dimensional spectrum of pwm waveform at  $f = 50$  &  $N = 20$ .

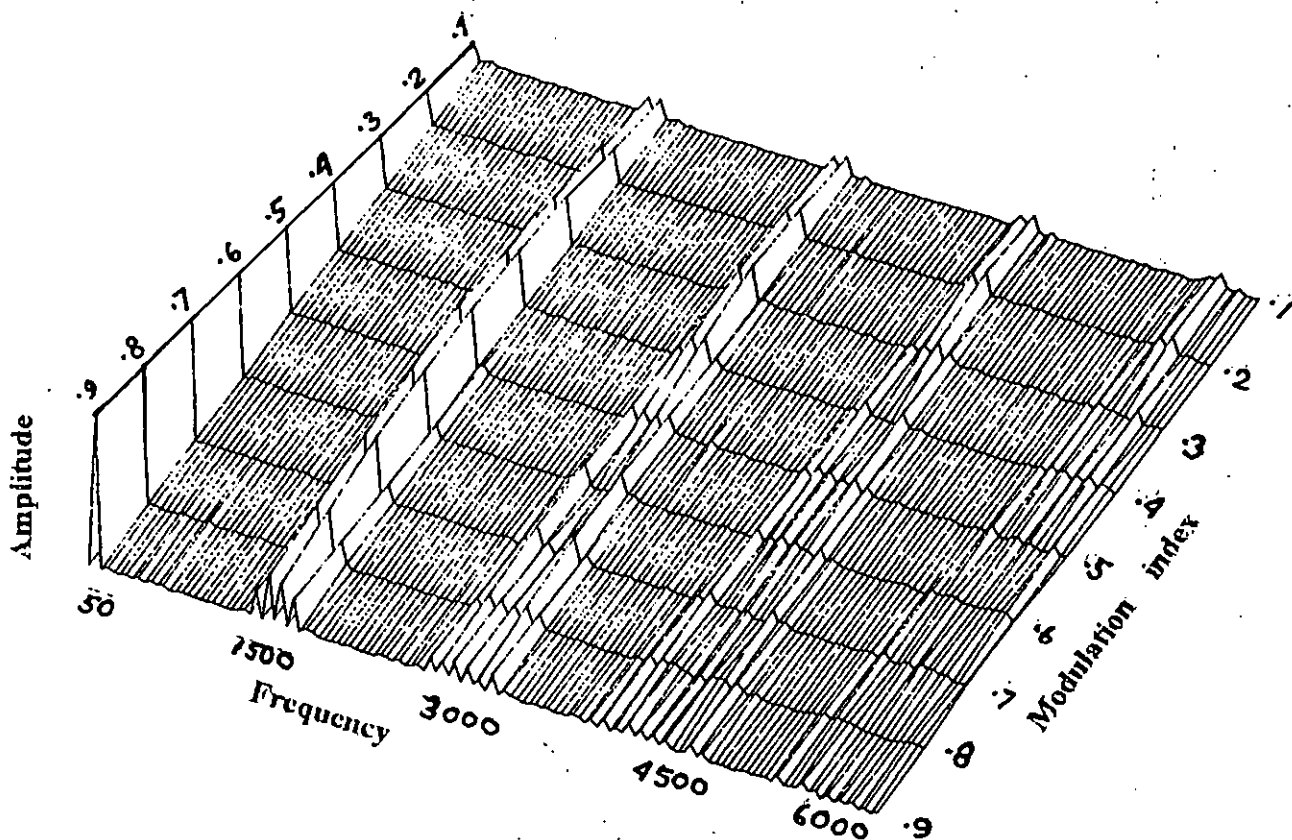


Figure 2.6.3(b) Three dimensional spectrum of pwm waveform at  $f = 50$  &  $N = 15$ .

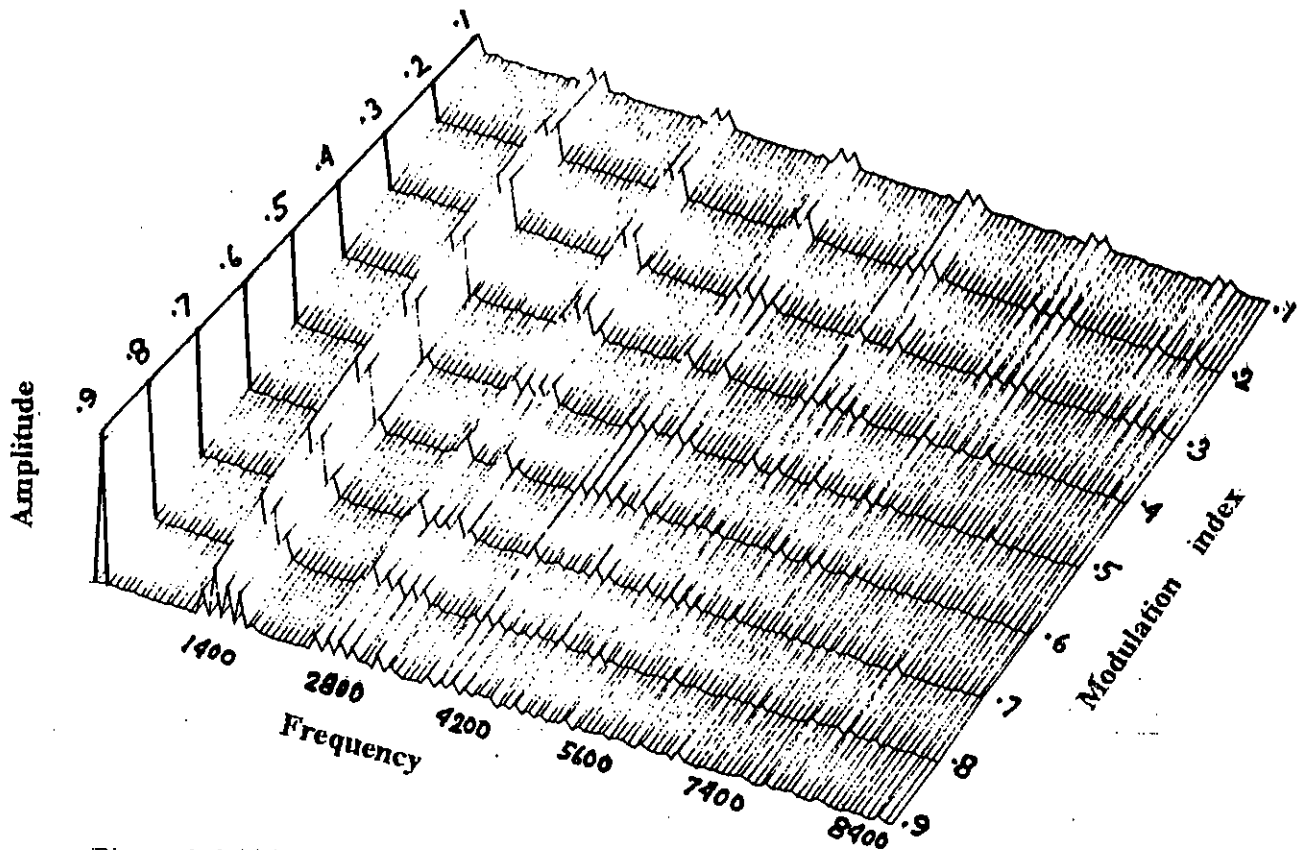


Figure 2.6.4(a) Three dimensional spectrum of pwm waveform at  $f = 70$  &  $N = 10$ .

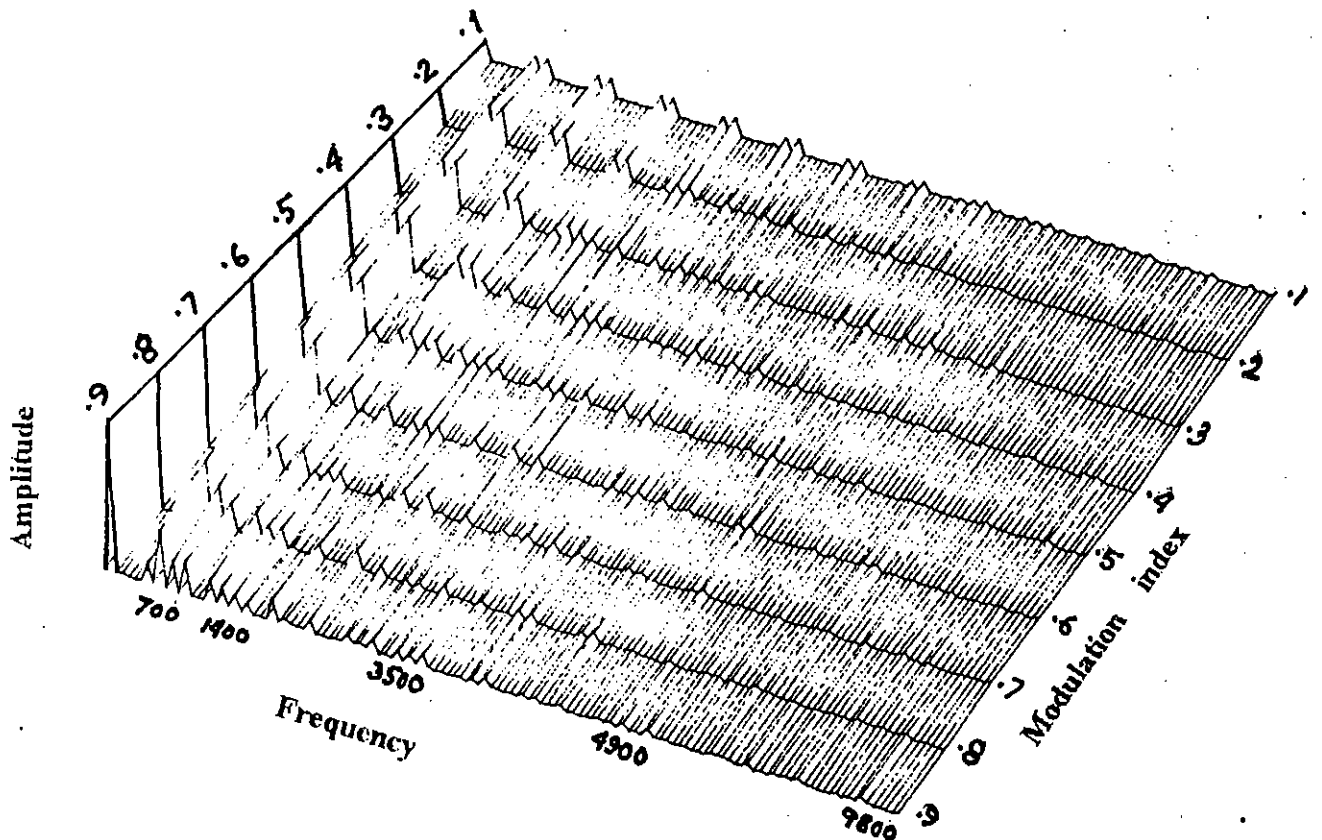


Figure 2.6.4(b) Three dimensional spectrum of pwm waveform at  $f = 70$  &  $N = 5$ .

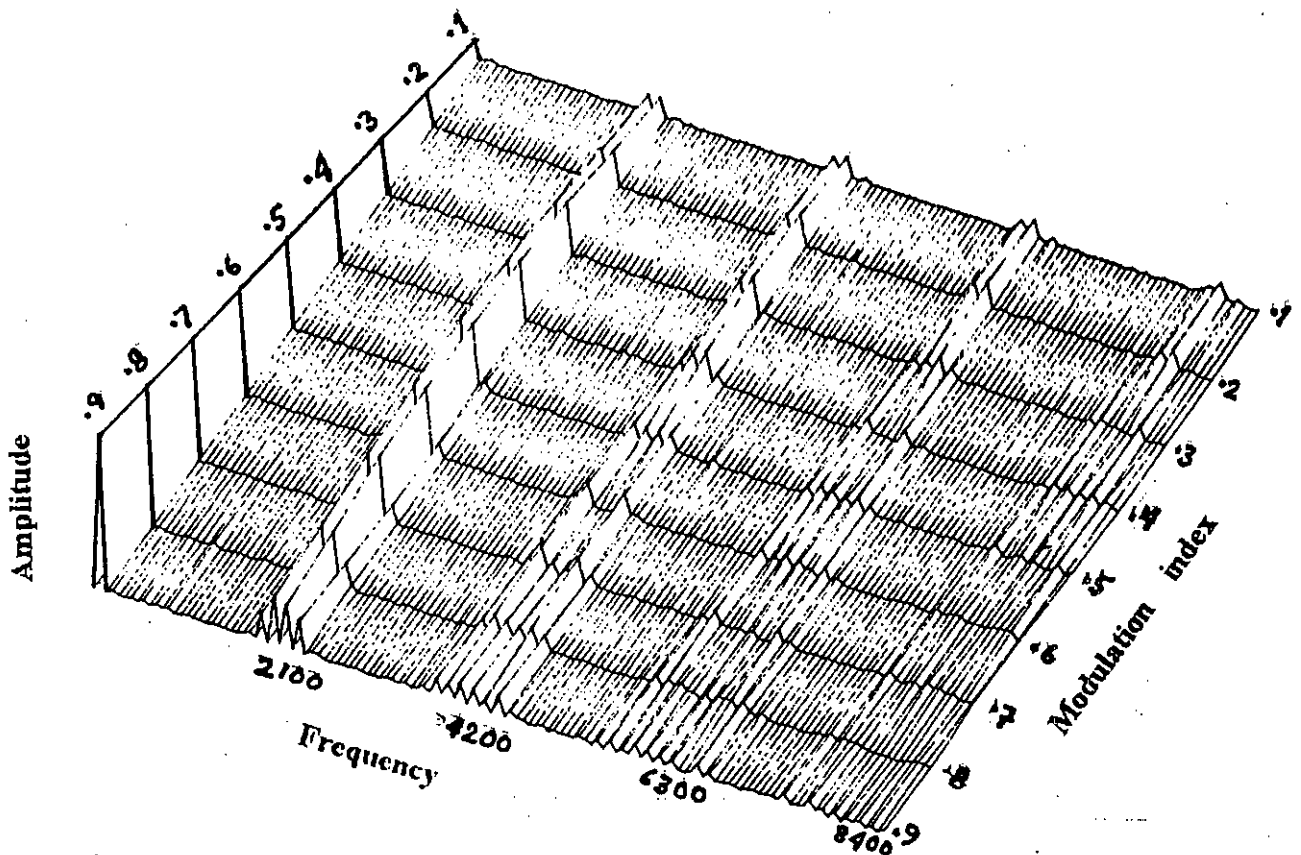


Figure 2.6.5(a) Three dimensional spectrum of pwm waveform at  $f = 70$  &  $N = 15$ .

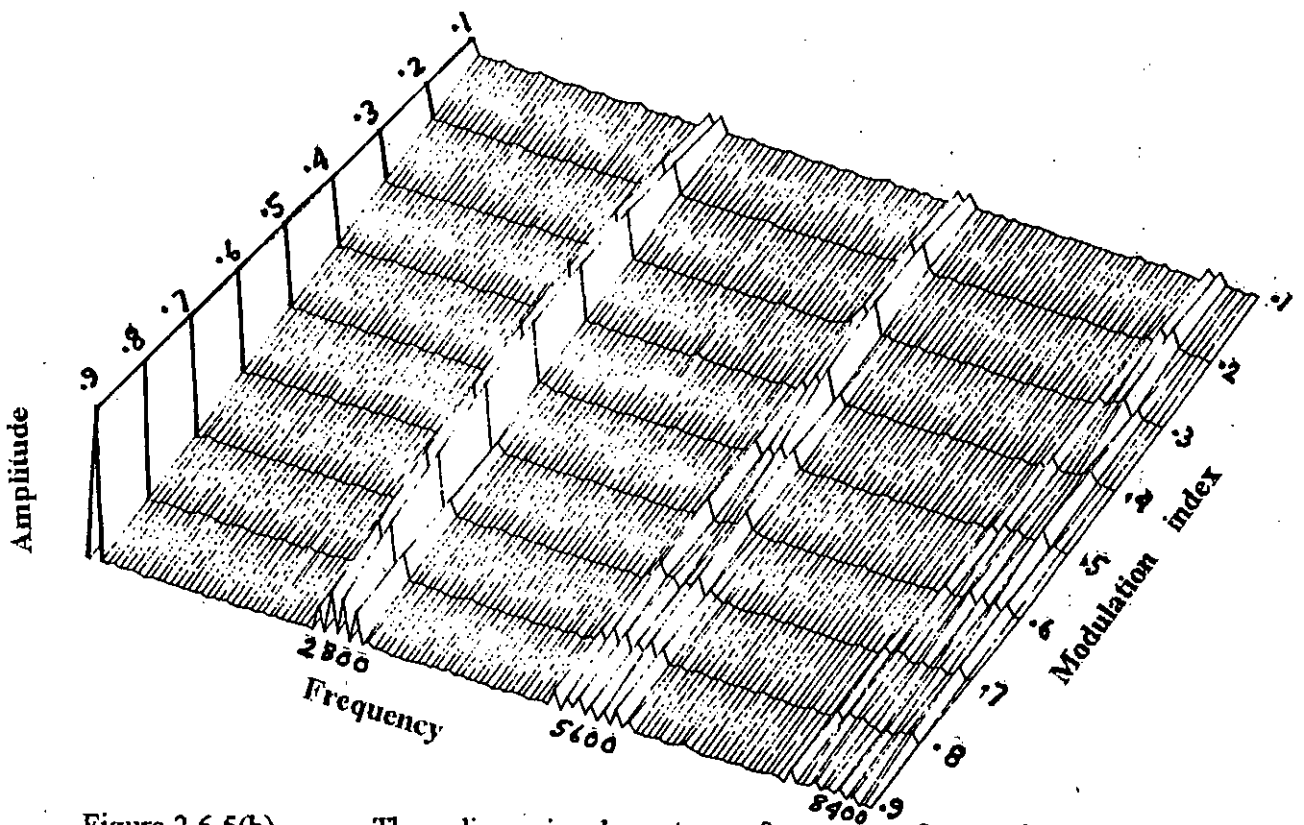


Figure 2.6.5(b) Three dimensional spectrum of pwm waveform at  $f = 70$  &  $N = 20$ .

- Variation of frequency requires simultaneous voltage control and carrier frequency control in drive application. Carrier frequency adjustment is necessary to retain switching frequency of the devices / second within device switching limit.
- A selection was done at different fundamental frequency, modulation index and at different number of carrier wave per half cycle to carried out the experiment. The result of selection for experimental implementation of the PWM are summarized in the table 2.5.

Sl. No.	Frequency	Modulation index	No. of carrier wave/ half cycle
1	30	.4	35
2	40	.6	30
3	50	.8	25
4	60	.9	20
5	70	.9	15

Table 2.5 Result of DFT analysis.

---

# CHAPTER THREE

---

## MICROPROCESSOR IMPLEMENTATION OF SPWM

# MICROPROCESSOR IMPLEMENTATION OF SPWM

## 3-1 INTRODUCTION

In the era of microcomputer control and automation it has been a constant effort of the researchers to implement various PWM switching strategies to control static power converters. For real time implementation of PWM waveforms using microprocessor, there are different techniques of realization depending on the type of modulation process. The implementation technique so far reported are based on Transcendental equations relating PWM switching angles and lacks generality when implemented by microcomputer [55]. The purpose of this thesis is to investigate an easy solution method for finding switching instants of SPWM waves and to generate gating pulse of a single phase inverter using switching points found by the new method. It is hoped that the proposed solution technique will enable On-line inverter control in future.

## 3-2 FORMULATION OF THE MICROPROCESSOR IMPLEMENTATION OF PWM

The major steps involved in the microprocessor implementation of PWM wave are as follows:

- Calculation of switching instants and evaluation of ON time and OFF time of the inverter switches.
- Choice of the hardware for the implementation of PWM.
- Calculation of switching instants and evaluation of ON time and OFF time.
- Development of software for the generation of PWM wave.

Figure 3.1.1 shows a typical SPWM wave having switching instants  $t_i$  ( $i = 1$  to  $N$ ) where  $N$  is the number of carrier pulse per half cycle. At each of these switching instants the PWM waveform switches from logic '1' to logic '0' level or logic '0' to logic '1' level. The switching instants for a modulation process can be found out by an equation involving the modulation parameters i.e modulation index, modulating frequency and the number of carrier wave per half cycle. For a given modulating frequency and parameter the switching angles are represented in time

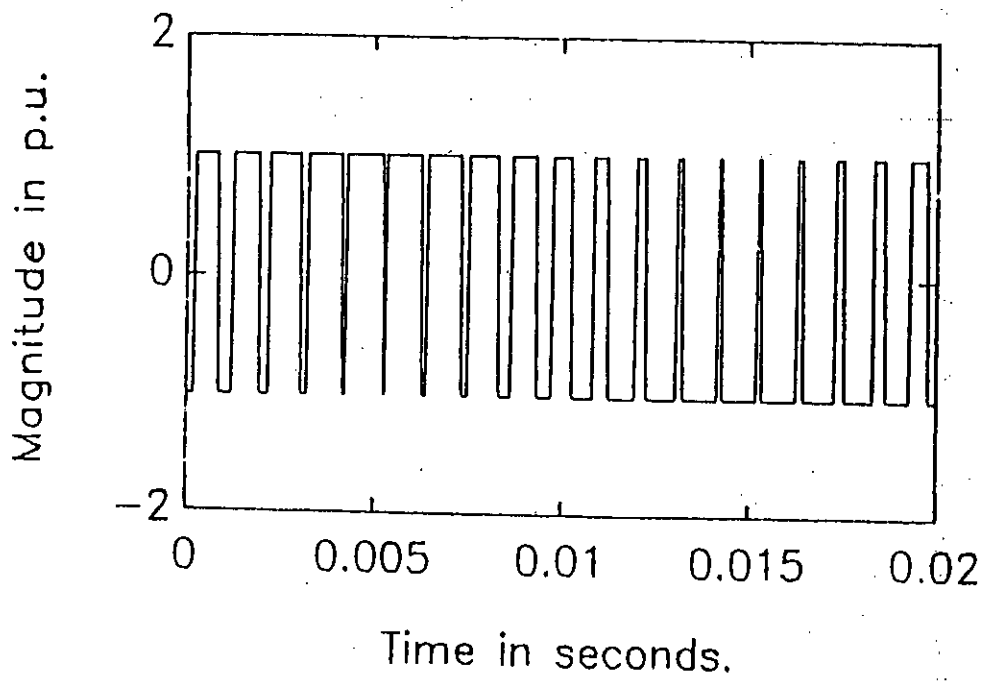
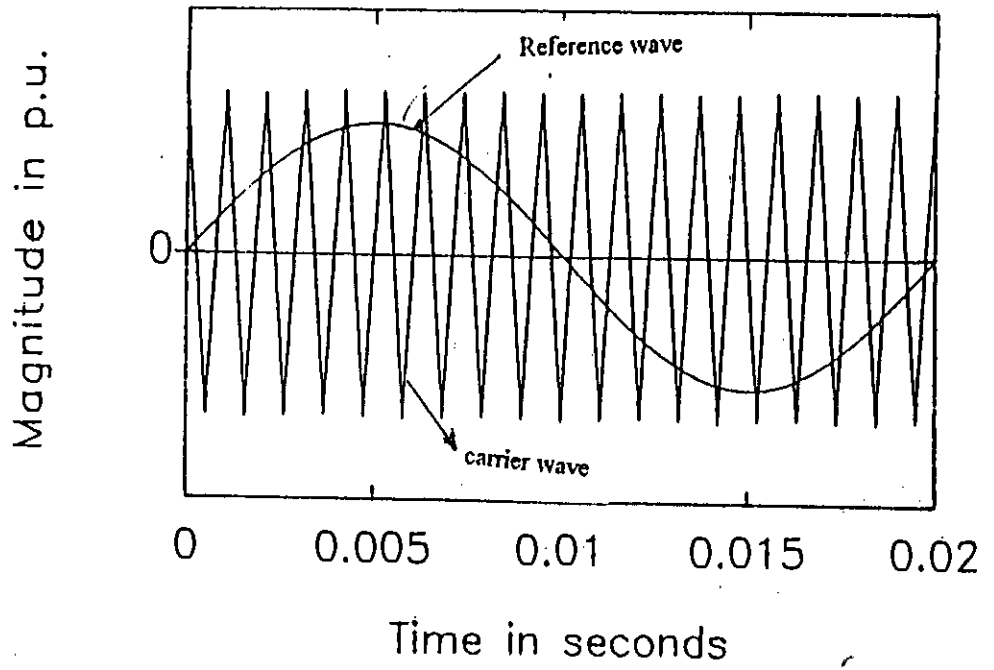


Figure 3.1.1

Typical sine pulse-width modulation (SPWM).



scales in units of microseconds. The modulation process involved in generation of SPWM is shown in figure 3.1.2 The expression for the ON time period is,

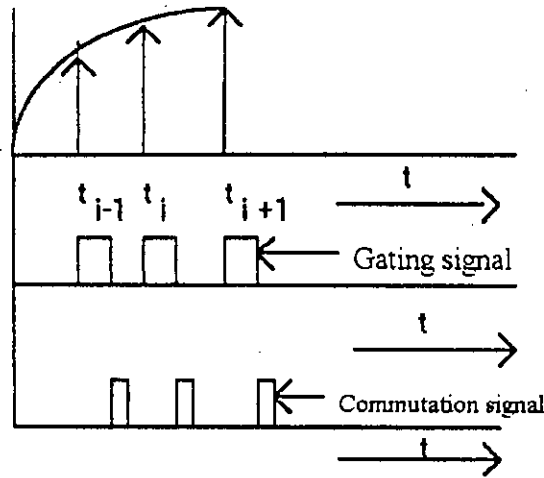


Figure 3.1.2 Typical switching instant for SPWM inverter

$$T_{1_{ON}} [i] = \frac{m}{2Nf} \sin \frac{\pi i}{N} \quad \text{for } 1 \leq i \leq (N - 1). \quad (3.1)$$

$$\text{wordcnt}_{ON} [i] = T_{1_{ON}} [i] * f_{clock} \quad (3.2)$$

We also know that,

$$\theta_i [i] = \frac{\pi i}{N}$$

and the OFF time period is,

$$T_{1_{OFF}} [i] = T_i [1] \quad \text{for } i=0 \text{ or } i=N \quad (3.3)$$

$$T_{1_{OFF}} [i] = T_i [i + 1] - T_1 [i] \quad \text{for } i=1 \text{ and } i= N-1$$

where,

$$T_i [i] = \theta_i [i] - \frac{T_{1_{ON}} [i]}{2}$$

$$T_i[i] = \theta_i [i] + \frac{T_{1ON}[i]}{2}$$

$$T_{1ON} [i] = \frac{m}{2Nf} \sin \frac{\pi i}{N}$$

$$\text{wordcnt}_{OFF} [i] = T_{1OFF} [i] * f_{\text{clock}} \quad (3.4)$$

Solving equation ( 3.1 ) and equation ( 3.3 ) the switching instants for a single phase inverter can be found out. The  $f_{\text{clock}}$  can be derived from program and from the hardware specification of the microprocessor and timer of the microprocessor kit. Solving equations (3.3) and (3.5) the hex value for the counter can be found out. Program developed for this purpose is written in 'C++' language and is included in Appendix-C. The ON and OFF-time equivalent hex value are then downloaded into hardware selected to conduct the research. This part of the research work was done on OFF-line basis. Two program was developed in assembly language to manipulate the ON and OFF time and to send required gating signal to the static switches. The flowchart describing the program logic is shown in fig. 3.2.1 and fig. 3.2.2. The main programs are included in Appendix - 4.1 and Appendix - 4.2. In the laboratory an SCR inverter is tested with the generated gating signals. Since the SCR inverter requires commutation signals, the software ( using software counter) also includes generation of commutation pulses as required .

### 3-3 HARDWARE SCHEMATIC

It is evident at this stage that the switching instants can be calculated easily. A generalized hardware schematic is considered for realization of the PWM waveform. Such a scheme is presented in the following section. The hardware can be any standard microprocessor kit or any standard microcomputer having PTM card with random access memory sufficient to load the user defined software (greater than 4KB ), 2KB of read only memory (ROM), a programmable interval timer (8253), programmable interrupt controller (8259), programmable keyboard controller (8279) and general purpose programmable peripherals interface (8255). The basic hardware schematic diagram is shown in figure - 3.3. The experimental setup is shown in figure - 3.4. Port A of the 8255 PPI was configured as input port and port B of 8255 PPI was configured as output port. Port A (C8) was used to input the desired frequency in which the user wants to run the inverter. Port B ( C9 ) was used to send output gating signal to the static switches. Each bit of port B corresponds to a particular static switch. The outputs from port B are interfaced to

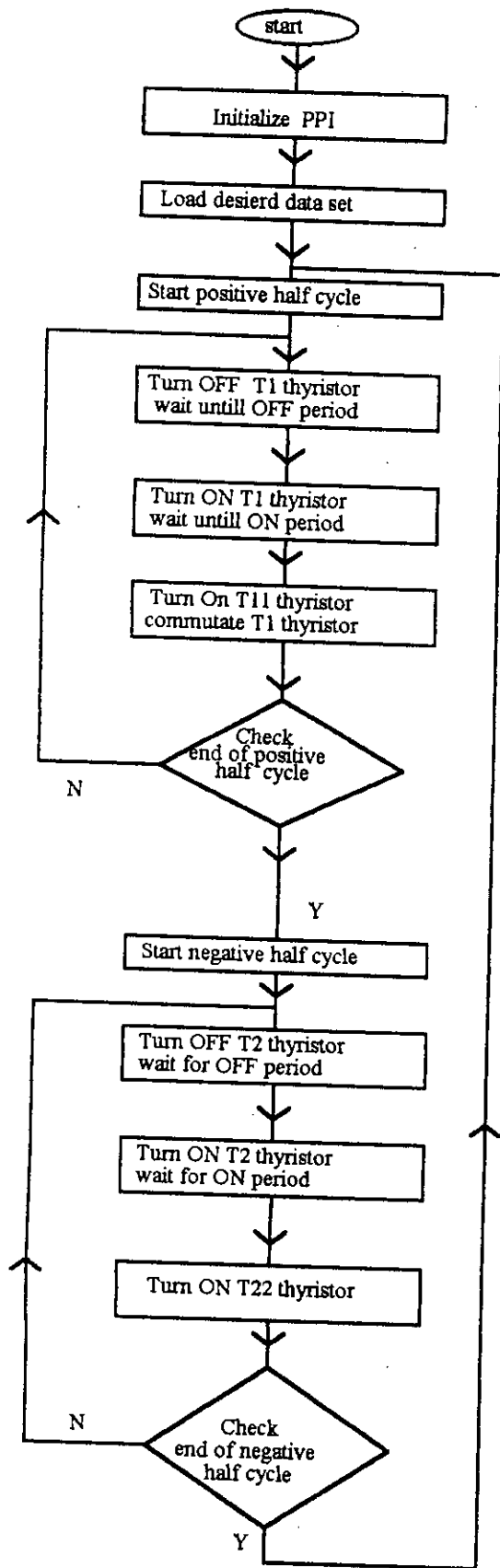


Figure 3.2.1 Flow chart of the main program  
(Using software counter)

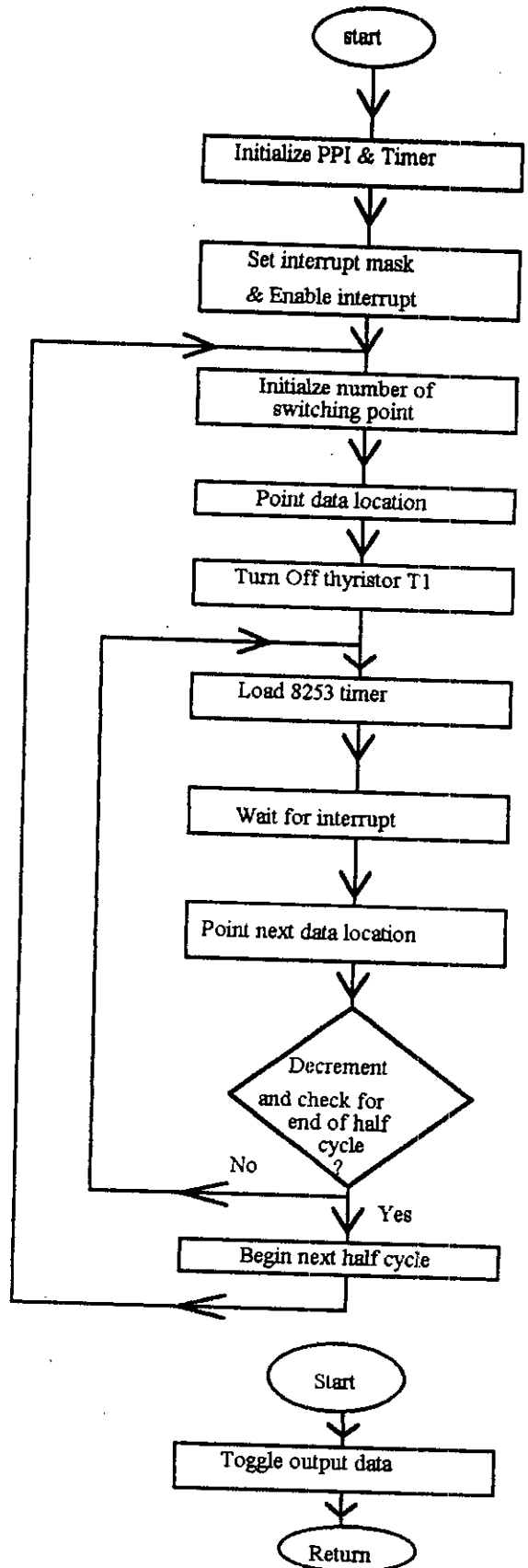


Figure 3.2.2 Flow chart of main program  
( Using 8253 programmable timer )

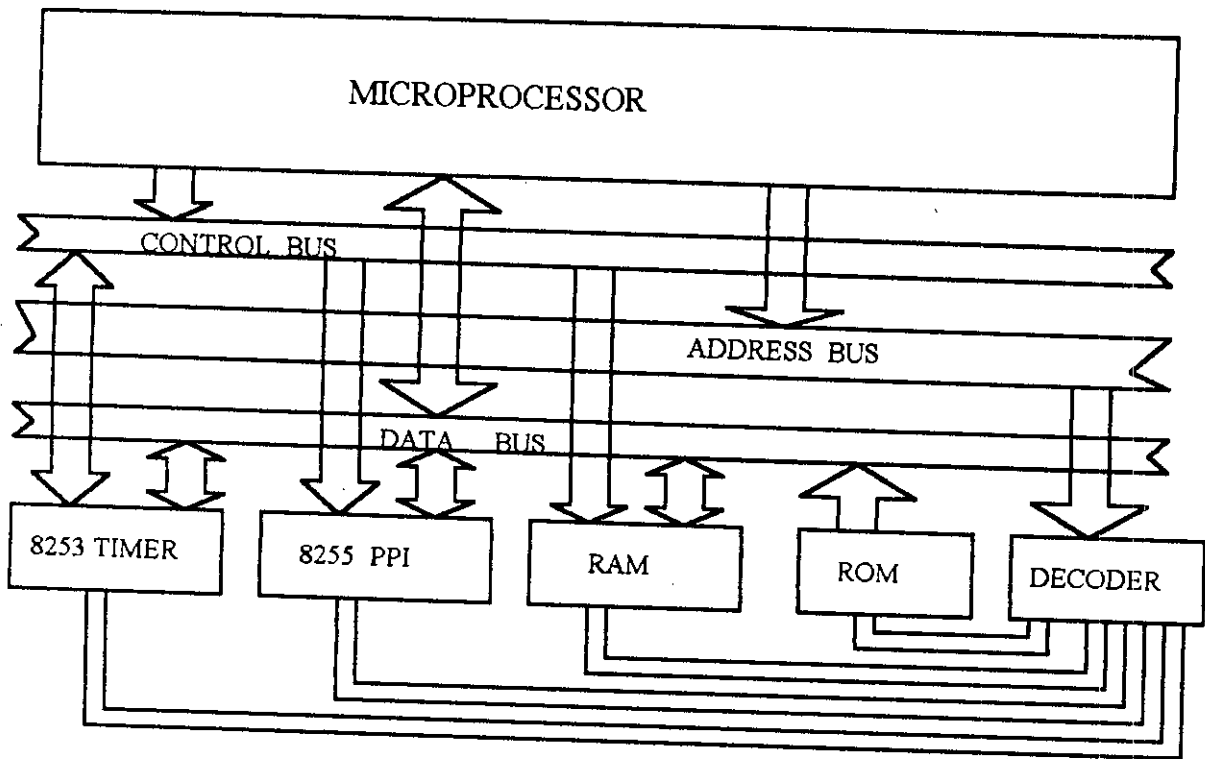


Figure- 3.3 Hardware schematic diagram

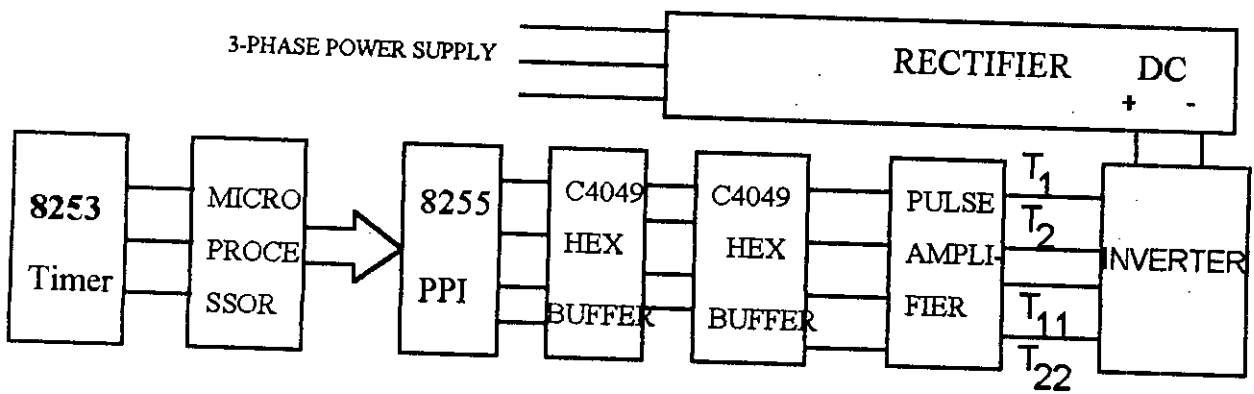


Figure 3.4 Block diagram of experimental setup

the gates of various static switches through buffer amplifiers and isolating circuits. The processor sends the OFF, ON and commutation signal to the inverter through 8255 PPI. It also keeps track of the positive and negative half cycle.

This method is not suitable for on line implementation as the processor is always busy. For On-line computation three timers necessary to control the OFF, ON and commutation signal and free the processor. The programmable timer 8253 can be used to control calculated switching instants. The timer can keep track of the starting and ending of each switching instants and commutation pulse. Timers of 8253 can be configured in mode 0 ( Interrupt on terminal count ) as a 16 bit counter to serve the purpose of delay on interrupt basis. First the timer was loaded with LSB, then with MSB. When control word in the status register and the count value in the counter is written, in the next cycle, timer will start to decrement the count value and as it reaches terminal count it will interrupt the processor to send new switching instants to the PPI to control the gate or base drive of the inverter. An interrupt signal will automatically be generated after the delay corresponding to the OFF , ON and commutation time (equivalent to some hex or decimal count value). After completion of the count, the processor is interrupted, the memory pointer is incremented. Then the processor will send the required control signal to each static switches through 8255 PPI. This way off or on time of gate pulses are controlled. The cycle will be repeated after completion of switching instants of one cycle. The processor will keep track of the starting and ending of the positive and negative half cycle and will be responsible to load the OFF , ON and commutation time equivalent hex count value. Block diagram of the experiment carried out in laboratory is presented in figures 3.4 and 3.5.

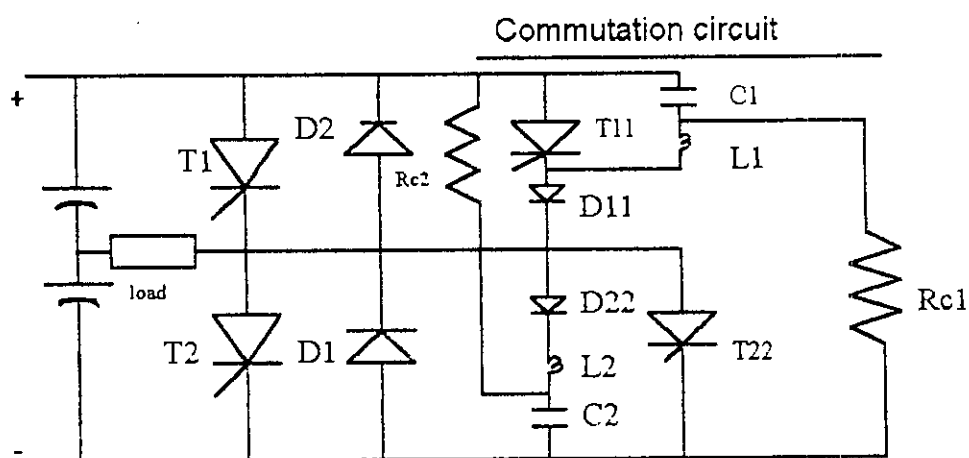


Figure 3.5 Half bridge Inverter with commutation circuit

### 3-4 PROGRAM DESCRIPTION

The execution of the main program begins by a start up procedure which initializes and configures the 8255 PPI. The program reads the ON-time and OFF-time periods stored in the lookup tables. The ON and OFF-time periods are stored in the lookup table as it is not possible to compute the ON and OFF period by 8 bit microprocessor kit ( As available in the laboratory ) . If a microprocessor kit of operating frequency above 15 MHz and instruction sets for direct division and multiplication are available then the ON and OFF time periods can be calculated on On-line basis. The 8255 PPI was configured in mode - 0. The processor first points the desired data set and starts cycle. At the positive half cycle the processor sends gating signal to turn OFF thyristor T1 and waits until OFF time period, then sends gating signal to turn ON T1 thyristor and waits until ON time completes. After ON time the processor sends gating signal to commutation thyristor T11 to commutate T1. This process will continue till the count value for positive half cycle finishes. It then starts negative half cycle and sends OFF, ON and commutation pulse accordingly. At the end of the negative half cycle the program repeats the entire cycle. A delay routine was used to serve purpose of required delay time. Several representative lookup tables are included in the tables 3.1 to 3.8 to supplement the experiment carried out in the laboratory and the result shown in the oscillogram. Some dummy instructions were used in the main program to maintain the accuracy of ON and OFF time and to provide the commutation signal to the commutation switch for commutating ON thyristor.

**Derivation of fclock ( software counter):-** The delay routine used in the main program is given below.

LABEL	MNEMONICS	No. of clock cycle
	CALL DELAY	18
DELAY :	MOV C , M	7
DELAY1 :	DCR C	4
DCX D		6
NOP		4
	JNZ DELAY1	7 / 10
RET		10

In the above program segment, if the JNZ condition is met ( i.e Z = 1 ) then ten cycles are required for each execution of JNZ instruction. However, if the JNZ condition is not satisfied then JNZ instruction requires seven clock cycles and the program branches back to the DCR C instruction. For each iteration in which the

JNZ is satisfied the number of clock cycle is equal to clock cycles for DCR C + cycles for DCX D + clock cycles for NOP + cycles for JNZ = 4 + 6 + 4 + 10 = 24 cycles. Twenty four ( 24 ) cycles will be performed ( X - 1 ) times, where X is the contents of the memory location pointed by HL register pair. For final iteration in which the JNZ condition doesn't comply, the number of clock cycles is equal to clock cycles for DCR C + cycles for DCX D + clock cycles for NOP + cycles for JNZ = 4 + 6 + 4 + 7 = 21 cycles. The time for loading count value from HL register pair to C register and CALL and RET must also be taken into account. Therefore the Delay loop time is = 18 + 7 + 24 ( X - 1 ) + 21 + 10 = 24X + 29 + 3 = 25 X ( if we choose 15 carrier wave per half cycle then we have 29 switching instants then 24\*X is a big number compared to 3. Hence the approximation has minor effect on the time calculation). Each clock cycle of the processor is 1/3.07 μs. From the above discussion we can devise a general expression for f<sub>clock</sub> in the following form.

$$f_{\text{clock}} = 3.07 * 1\text{E}+06 / 25 \text{ Hz} . \quad (3.5)$$

This equation was used for calculating On and Off time of the inverter .

The data generated for programmable timer implementation of this scheme needs analysis of the following program section.

LABEL	MNEMONICS	NO. OF CLOCK CYCLE
	HALT	
ISR:	IN D8	10
	CPI 01H	06
	CMA	04
	OUT D9	10
	EI	04
	RET	10
INT. XFER:	JMP ISR	10
POINT		

This section reveals that to change the output of PPI a constant time of 19 micro second ( 10 + 06 + 04 +10 + 04 +10 + 10 + 06 = 60 processor clock ) is needed by the program. The frequency of the timer 8253 used in the system is 1.56 MHz. Hence for calculating the switching instants using 8253 programmable timer needs a slight modification .

$$\text{TION} / \text{T1 OFF ( in decimal)} = ([\text{TION} / \text{T1 OFF} - 19 ] * 1.56). \quad (3.6)$$

The data generated taking into account the modification mentioned above are also presented in tables 3.1 to 3.8. The timer was configured in mode-0 ( Interrupt on terminal count), 16- bit and as a decimal down counter. The program developed for generating the gating pulse for the inverter are presented in Appendix - 4.2.

T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value T1ON time (software counter)	Hex-value T1Off time (software counter)	T1ON time 8253 timer	T1Off time 8253 timer
0	432	00	34	00	642
89	345	0A	2A	108	507
174	264	15	20	240	380
251	192	1E	17	360	268
318	132	26	10	464	174
371	89	2D	0A	547	107
407	60	31	07	603	62
426	50	33	06	633	46
426	60	33	07	633	62
407	88	31	0A	603	107
371	133	2D	10	547	176
318	192	26	17	464	268
251	264	1E	20	360	380
174	345	15	2A	240	507
89	432	0A	34	108	642

Table-3.1: The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 70$  Hz, modulation index  $m = .9$ , 15 carrier pulse / half Cycle.

T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value T1 ON (software counter)	Hex-value T1off (software counter)	T1On time 8253 counter	T1Off time 8253 counter
00	457	00	37	00	681
39	419	04	33	29	622
77	383	09	2E	88	566
111	350	0D	2A	141	514
141	324	11	27	188	474
164	305	14	25	224	444
181	292	16	23	251	424
189	288	17	23	263	418
189	292	17	23	263	424
181	304	16	25	251	443
164	325	14	27	224	475
141	351	11	2A	188	516
111	383	0D	2E	141	566
77	419	09	33	88	622
39	457	04	37	29	681

Table:- 3.2 The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 70$  Hz, modulation index  $m = .4$ ,  $N = 15$  carrier pulse / half Cycle.



T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value T1 ON (software counter)	Hex-value T1off time (software counter)	T1On time 8253 counter	T1Off time 8253 counter
0	821	0	64	00	1291
398	462	30	38	589	790
718	215	57	1A	1090	390
895	126	6D	0F	1365	168
895	215	6D	1A	1365	390
718	462	57	38	1090	790
398	821	30	64	589	1291

Table-3.3: The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 70$  Hz, modulation index  $m = .9$ ,  $N=7$  carrier pulse / half Cycle.

T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value T1 ON (software counter)	Hex-value T1Off (software counter)	T1On time 8253 counter	T1Off time 8253 counter
0	1065	00	81	000	1630
92	975	0B	76	112	1489
180	891	15	6C	249	1358
261	816	1F	63	375	1241
330	754	28	5B	483	1145
384	708	2E	56	567	1073
422	679	33	52	627	1028
442	669	35	51	658	1014
442	680	35	52	658	1029
422	708	33	56	627	1073
384	754	2E	5B	567	1145
330	816	28	63	483	1241
261	891	1F	6C	375	1358
180	975	15	76	249	1489
92	1065	0B	81	112	1630

Table-3.4: The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 30$  Hz, modulation index  $m = .4$ ,  $N=15$  carrier wave / half Cycle.

T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value T1 ON (software counter)	Hex-value T1 Off (software counter)	T1On time 8253 timer	T1Off time 8253 timer
0	932	00	71	00	1422
177	773	15	5E	244	1174
319	664	26	51	466	1004
397	624	30	4C	588	942
397	664	30	51	588	1004
319	773	26	5E	466	1174
177	932	15	71	244	1422

Table-3.5: The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 70$  Hz, modulation index  $m = .4$ ,  $N=7$  carrier pulse / half Cycle.

T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value T1 ON (software counter)	Hex-value T1Off (software counter)	T1On time 8253 timer	T1Off time 8253 timer
0	1008	0	7A	00	1541
207	805	19	62	291	1224
406	615	31	4B	602	928
587	447	47	36	884	666
743	307	5A	25	1127	447
866	203	69	18	1319	285
951	139	74	10	1452	185
994	117	79	0E	1519	151
994	140	79	11	1519	187
951	203	74	18	1452	285
866	307	69	25	1319	447
743	447	5A	36	1127	666
587	615	47	4B	884	928
406	805	31	62	602	1224
207	1008	19	7A	291	1541

Table-3.6: The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 30$  Hz,  $m = .9$  modulation index,  $N=15$  carrier wave/half Cycle.

T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value T1 ON (software counter)	Hex-value T1Off (software counter)	T1On time 8253 counter	T1Off time 8253 timer
0	468	0	39	00	698
17	451	2	37	01	672
34	434	4	34	21	645
50	418	6	32	46	620
66	402	8	31	71	595
82	388	A	2F	97	574
97	373	B	2D	120	550
111	359	D	2B	141	528
125	346	F	2A	164	508
137	334	10	28	182	489
148	324	12	27	199	474
158	314	13	26	215	458
167	306	14	25	229	446
175	299	15	24	241	435
181	294	16	23	256	427
185	291	16	23	257	422
188	287	16	23	262	416
190	286	17	22	265	414
190	287	17	23	265	416
188	290	16	23	262	421
185	295	16	23	257	429
181	299	16	24	251	435
175	306	15	25	241	446
167	314	14	26	229	458
158	323	13	27	215	472
148	334	12	28	199	489
137	347	10	2A	182	510
125	359	F	2B	163	528
111	373	D	2D	141	550
97	387	B	2F	120	572
82	402	A	31	96	595
66	419	8	33	71	622
50	434	6	34	46	645
34	451	4	37	21	672
17	468	2	39	01	698

Table-3.7: The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 30$  Hz,  $m = .4$  modulation index,  $N=35$  carrier wave/half Cycle.

T1 on time in $\mu$ -secs	T1 off time in $\mu$ -secs	Hex-value (T1 ON)	Hex-value T1Off (software counter)	T1On time 8253 timer	T1Off time 8253 timer
0	457	0	37	00	589
38	419	4	33	28	538
76	381	9	2E	87	487
114	344	D	29	146	437
150	309	12	25	202	390
185	276	16	21	257	345
219	242	1A	1D	310	299
251	211	1E	19	360	257
281	182	22	16	407	218
309	155	25	12	450	182
335	132	28	10	491	151
357	110	2B	0D	525	121
377	91	2D	0B	556	95
394	76	30	09	583	75
407	65	31	07	603	60
417	57	32	06	619	49
424	50	33	06	630	40
428	48	34	06	636	37
424	56	33	06	630	48
417	66	32	08	619	62
407	76	31	09	603	75
394	91	30	0B	583	95
377	110	2D	0D	556	121
357	131	2B	0F	525	149
335	155	28	12	491	182
309	183	25	16	450	218
281	211	22	19	407	257
251	242	1E	1D	360	299
219	275	1A	21	310	344
185	309	16	25	257	390
150	345	12	2A	202	438
114	381	D	2E	146	487
76	419	9	33	87	538
38	457	4	37	28	589

58228

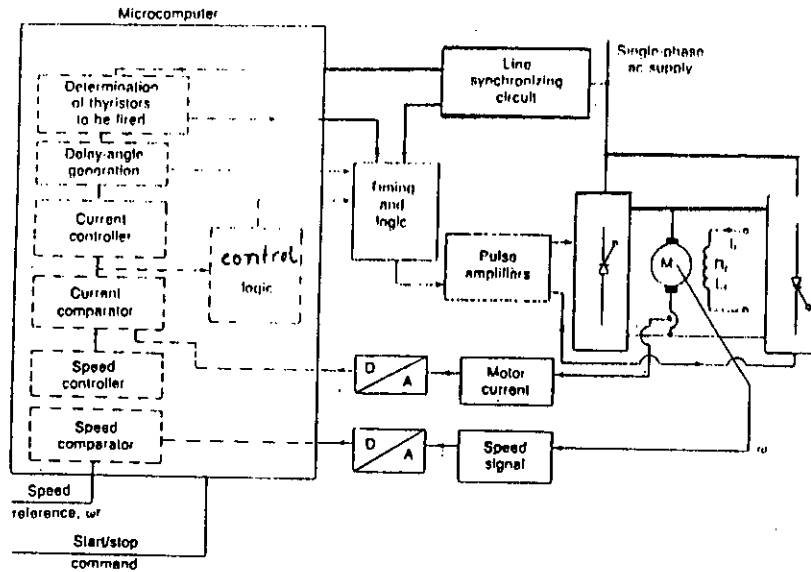
Table-3.8: The Switching Instants and their Hex Values (software counter) and their decimal values for 8253 programmable timer at  $f = 30$  Hz,  $m = .9$  modulation index,  $N=35$  carrier wave/half Cycle.

### 3.5 INVESTIGATION FOR ON-LINE IMPLEMENTATION

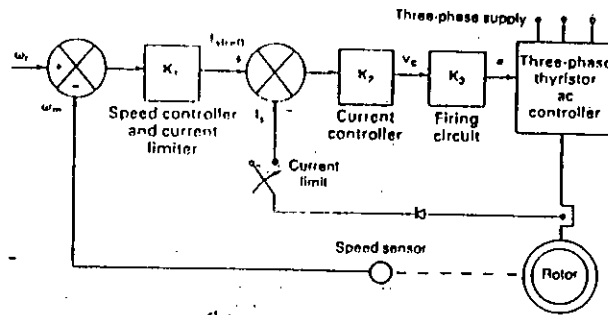
By ON-line computation and control by microprocessor, we describe a process control in which the computational procedure will not stop generating controlling signals even if certain computation require art of routine calculation. Such computation is done in very short time and the process may take its new operating sequence quickly. Such On-line computation are very common in close loop control of drives. A closed loop control is normally required in order to satisfy the steady state and transient performance specifications of any system. Digital computer plays a significant role in modern closed loop control systems. A control system is generally characterized by the hierarchy of the control loops, where the outer loop controls the inner loops. The inner loops are designed to execute progressively faster. The loops are normally designed to have limited command excursion. Some representative closed loop control of induction motors and D.C motors are presented in figure 3.6. For D.C motor control the speed signal is fed into the microcomputer using an A/D ( Analog to digital converter). To limit the armature current of the motor the inner current control loop is used as shown in figure 3.6 (a). The armature current signal can be fed into the microcomputer through an A/D converter by sampling the armature current. The line synchronization circuit is required to synchronize the generation of the firing pulses with the supply line frequency. The pulse amplifier provides the necessary isolation and produce gate pulse of the required magnitude and duration [6].

The dynamic model of induction motor differs significantly from that of DC motors. Arrangement shown in figures 3.6(b) to 3.6(d) are different strategies of closed-loop induction motor control. The speed controller, K1, process the speed error and generates the reference current  $I_s(\text{ref})$ . K2 is the current controller. K3 generates the gating pulse for thyristor converter. The speed controller K1 may be simple gain proportional type or proportional integral type or lag-lead compensator[6]. The gating pulse / firing pulse generator may be a scheme studied and analyzed in the research work.

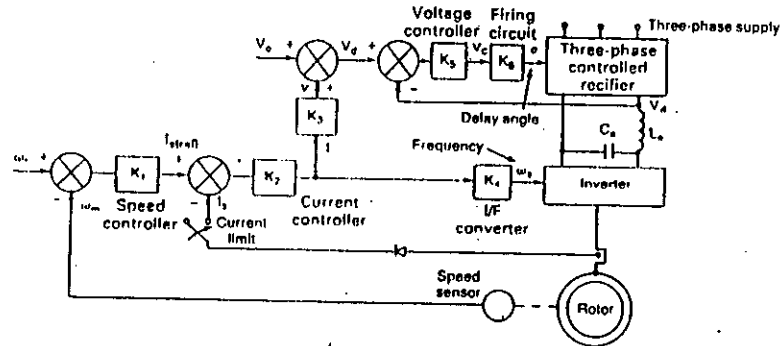
Study of these systems reveals that accurate and efficient inverter output control have significant role. For controlled inverter output numerous methods have so far been used and reported. One of the best technique is sine pulse width modulation (SPWM). Numerous techniques have so far been used to implement this technique using microprocessor, but all the implementation technique reported so far used ROM based lookup table or OFF line basis calculation of switching point. The main disadvantage of ROM base lookup table is it can't provide smooth control



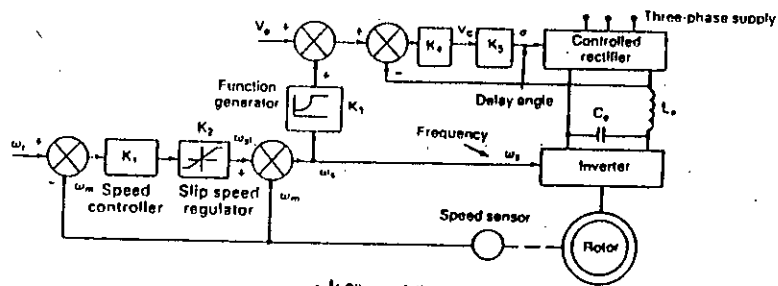
(a) Schematic diagram of computer-controlled four-quadrant dc drive.



(b) Stator voltage control



(c) Volts/hertz control

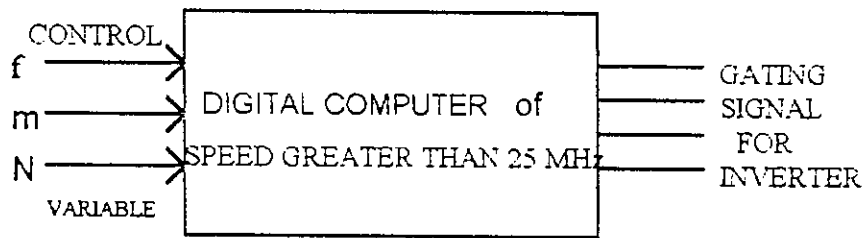


(d) Slip regulation

Figure 3.6 Closed-loop control D.C and induction motors

[6]

over wide range of variation of the control variable. The method studied in this research will enable to overcome this problem as microprocessors of higher operating frequency (greater than 25 MHz) and direct multiplication and division capability with co-processor having the capability of computing any trigonometric function are available. A block diagram of possible On-line switching point calculation is presented in figure 3.7.



Figure, 3.7 Block diagram of On-line gating pulse generation for inverter

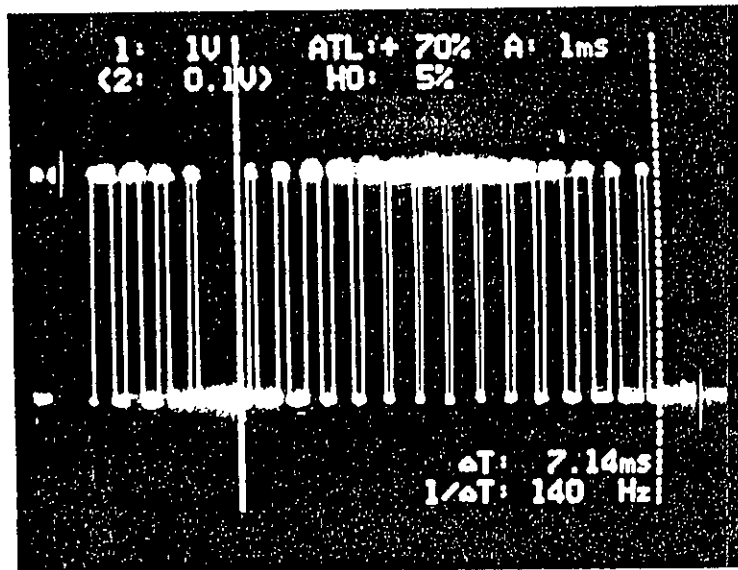
The method involved is simple and easy. For On-line gating pulse generation we have to calculate the On time and Off time of each inverter. To implement this we have to calculate the hexadecimal equivalent of On and Off time to load programmable timer to keep track of On and Off time of each inverter and to free the processor for computation of new switching instant according to system requirement or user request. For the purpose of solution of equations (3.1) - (3.4), we need five multiplication four division two subtraction, one addition and one sine function calculation instruction. Typically (for 80386 microprocessor) these instructions need 13-22, 14-25, 2-7, 2-7 and 0 (if co-processor available otherwise a sine function calculation routine have to be incorporated) clock cycles respectively. Hence for single calculation we need  $5 \times 22 + 4 \times 25 + 2 \times 7 + 1 \times 7 + 0 = 231$  processor clock cycles (worst case). If the processor clock is 35 MHz then it needs approximately 13.2 micro second and if we think about fast RISC (Reduced instruction set computer) architecture based processor of operating frequency 20 or 25 MHz, like Motorola 88100 or Intel - 486 having operating speed about 100 MHz then it may need only a few micro second to compute the switching instants. This part of the research was not carried out during the tenure of the research.

### 3.6 EXPERIMENTAL RESULTS

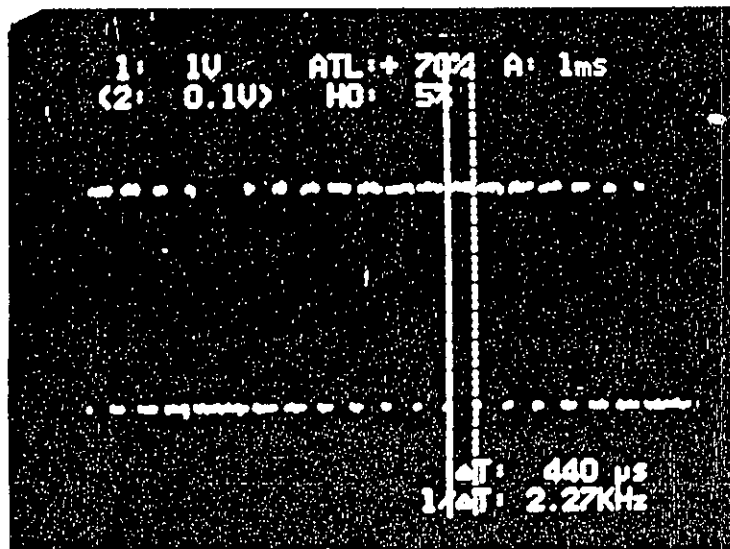
To compare the performance of the microprocessor controlled gating pulse with the gating pulse obtained from analysis carried out using MATLAB software, the oscillogram of the experimental results are shown in figures 3.8.1 to 3.8.4. The comparison of these oscillograms show a close agreement between the shapes, size and location of the gating pulses for a desired value of frequency. A single phase inverter was driven at various operating frequencies with the microprocessor generated gating signals. The waveforms for 30Hz, modulation index .9 and .4 and 70 Hz, modulation index .4 having number of pulses 7 and 15 are presented in figures 3.9.1 to 3.9.2. Experimental spectral analysis was not carried out because of unavailability of spectral analyzer. To maintain the cycle and to send the gating signal at output it is essential to execute some instruction. hence to adjust the cycle time and to maintain the accuracy of the desired output some dummy instructions were included in the program . This work was done on Off-line basis as it is not possible to compute the switching instant and simultaneously sending the gating signal to the inverter using microprocessor of low operating frequency. It needs microprocessors of higher operating frequency like ( Intel 486 , pentium etc.). The processor of higher operating frequency provides faster computing facilities as they have highly parallel processing and pipeline architecture. To carryout this work on On-line basis one needs the processor to set free for computing new switching instants. The idea for this purpose is simple and was tested using 8253 programmable timer. Oscillogram of the gating pulse at 70Hz, .9 modulation index and 7 carrier wave per half cycle is presented in figure 3.9.3.

The experimental output of the inverters shows quite compliance with the simulated PWM wave form. The errors found are within range 0 - 6% these errors are essentially due to observation. Some spikes are noticed at the photograph of inverter output. These are inherent characteristics of the static switches (turning On and Off are the fact behind this phenomenon). High frequency switching causes switching loss of the static devices to increase and may damage the devices if device switching limit exceeds. It is essential to operate the devices within device switching limit.





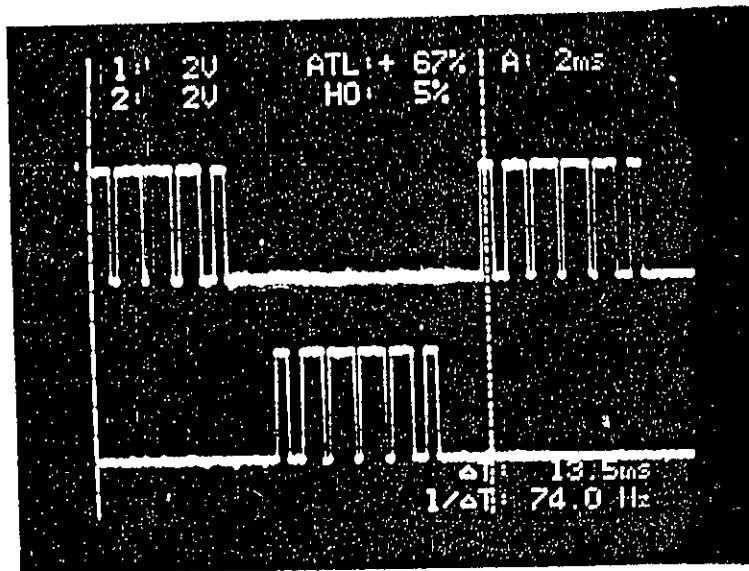
(a)



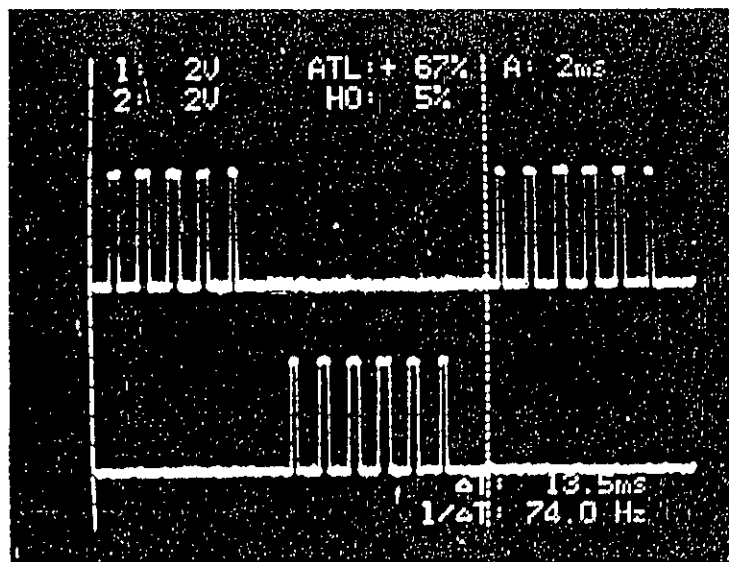
(b)

Figure 3.8.1(a) Typical experimental oscillogram of PWM wave

(b) Typical experimental PWM wave



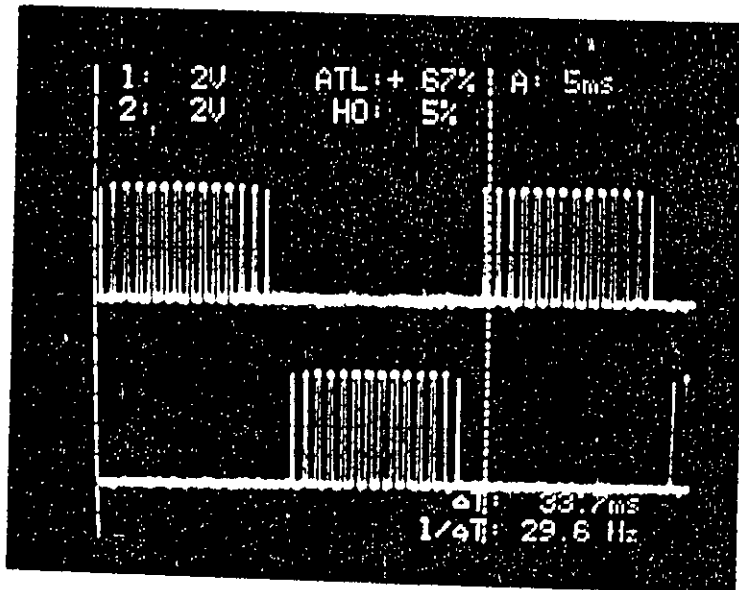
(a)



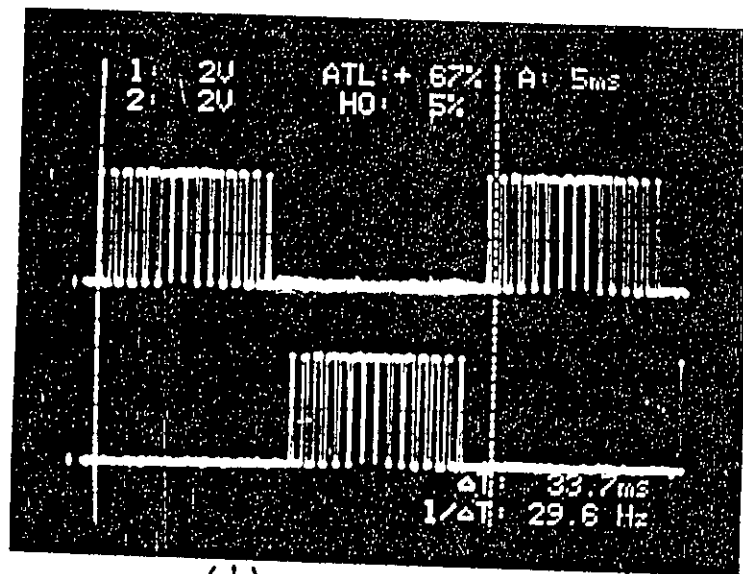
(b)

Figure 3.8.2(a) Oscilloscope of PWM gating wave at  $f = 70 Hz$ ,  $m = .9$  and  $N = 7$

(b) Oscilloscope of PWM gating wave at  $f = 70 Hz$ ,  $m = .4$  and  $N = 7$



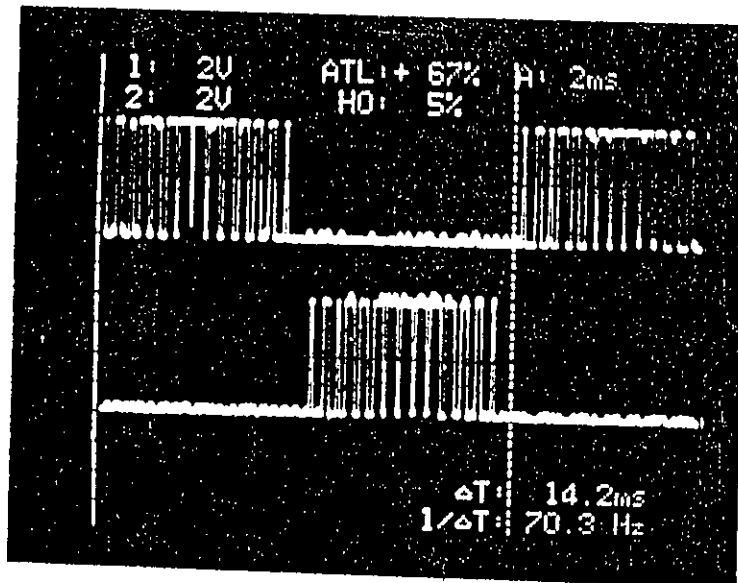
(a)



(b)

Figure 3.8.3(a) Oscilloscope of PWM gating wave at  $f = 30$  Hz,  $m = .4$  and  $N = 15$

(b) Oscilloscope of PWM gating wave at  $f = 30$  Hz,  $m = .9$  and  $N = 15$



(a)

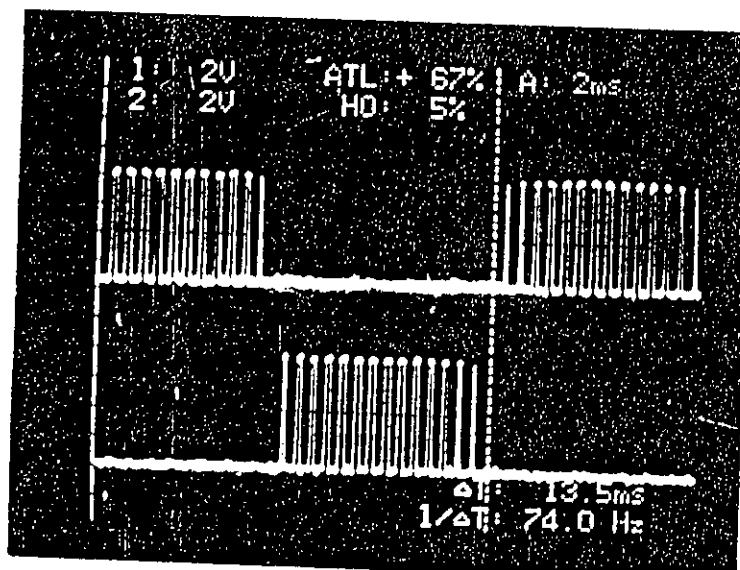
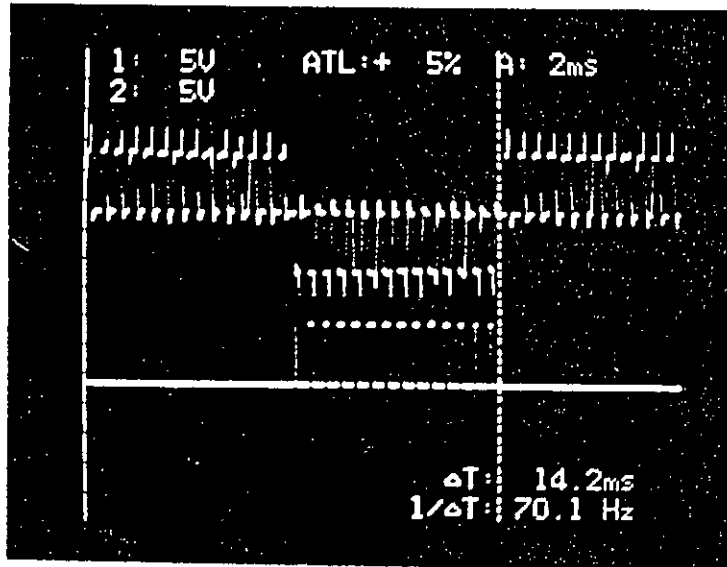


Figure 3.8.4(a) (b) Oscilloscope of PWM gating wave at  $f = 70 Hz$ ,  $m = .9$  and  $N = 15$

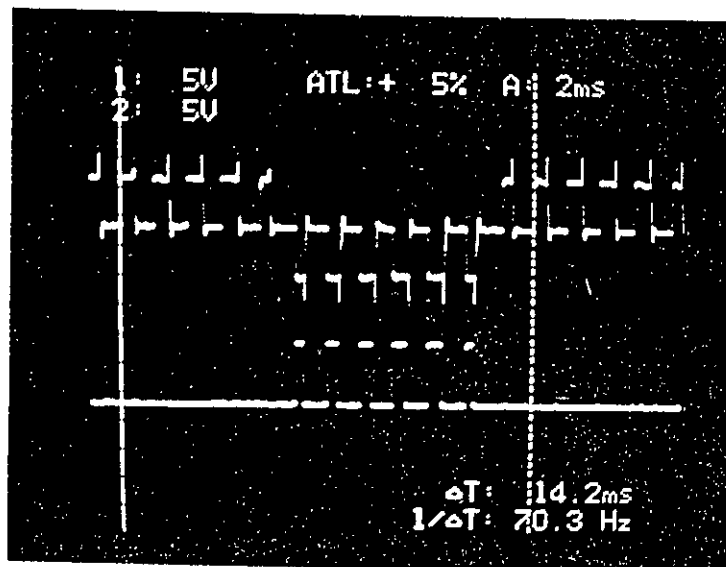
(b) Oscilloscope of PWM gating wave at  $f = 70 Hz$ ,  $m = .4$  and  $N = 15$



Inverter output

output of 8255 PPI  
(Gating pulse of  
one inverter.)

(a)



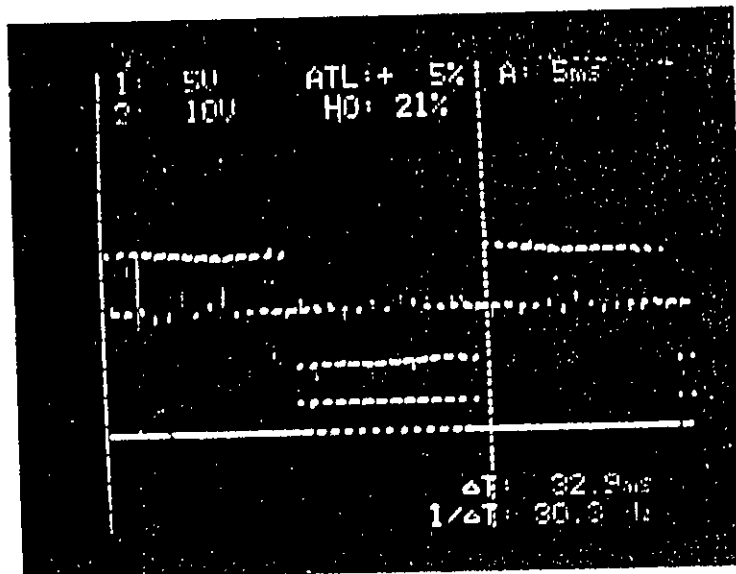
Inverter output

output of 8255 PPI  
(Gating pulse of  
one inverter.)

(b)

Figure 3.9.1(a) Experimental output of the inverter at  $f = 70$ ,  $m = 4$  and  $N = 15$

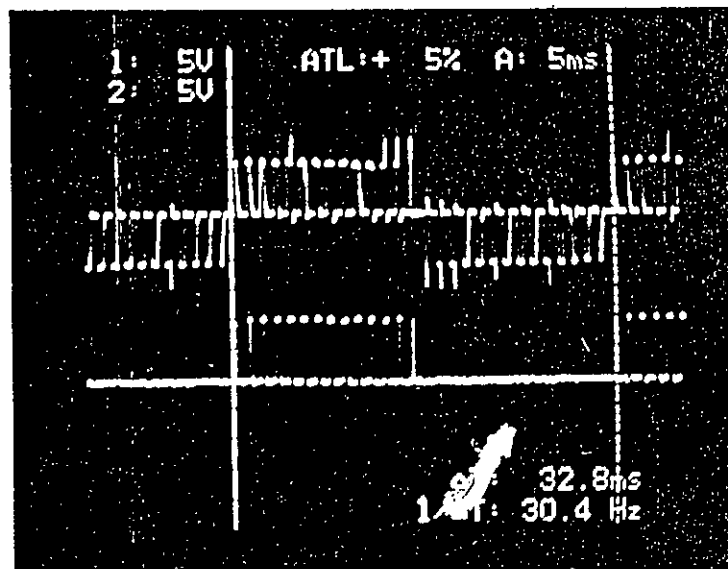
(b) Experimental output of the inverter at  $f = 70$ ,  $m = 4$  and  $N = 7$



Inverter output

output of 8255 PPI  
(Gating pulse of  
one inverter.)

(a)



Inverter output

output of 8255 PPI  
(Gating pulse of  
one inverter.)

(b)

Figure 3.9.2(a) Experimental output of the inverter at  $f=30$ ,  $m=.9$  and  $N=15$

(b) Experimental output of the inverter at  $f=30$ ,  $m=.4$  and  $N=15$

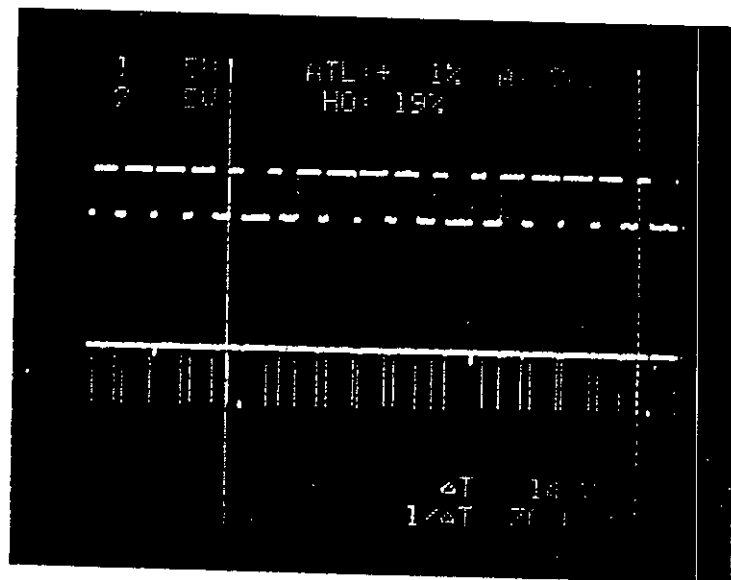


Figure 3.9.3 Oscillogram of PWM wave at 70 Hz,  $m = .9$  modulation index  
 (using software timer)  
 Upper pulse is output of 8255 PPI (gating pulse of inverter)  
 lower pulse is output of 8253 (interrupt pulse for RST 7.5)

---

**CHAPTER**

**FOUR**

---

**SUMMARY AND CONCLUSION**



## SUMMARY AND CONCLUSION

### 4-1 CONCLUSION

The objective of this research has been to investigate the microprocessor implementation of probable On-line gating pulse generation for a sine pulse width modulated inverter. Mathematical analysis and Fourier Transform analysis have been carried out for single phase inverter. Harmonic analysis was carried out on the simulated PWM waveforms and the observation of the analysis was summarized. A judicious selection was done at selected different frequencies ( 30Hz, 40 Hz, 50 Hz, 60 Hz, 70Hz) to choose number of carrier wave per half cycle and modulation index at which converter was supposed to offer enhanced performance. The results are practically implemented in the laboratory by running an SPWM inverter. The implementation has been realized in OFF-line basis due to the unavailability of faster microprocessor kit having direct multiplication and division capability. The possibility of the On-line implementation of this work has been analyzed and found to be feasible due to small computational time required for calculating the switching instants of the PWM wave. Microprocessor implementation of the technique has been provided with upgrading and future provision of change of control strategy by changing software only. This needs no extra hardware adjustment as required in dedicated hardware PWM scheme.

The features of microprocessor implementation of SPWM in the operation of a single phase inverter is easy to implement, very small low order harmonics at the output and volt / Hz controllable. Harmonic minimization can also be achieved in microprocessor based PWM control. High frequency carrier wave reduces output filter size of inverters. The analysis of inverter waveform has been carried out by FFT ( MATLAB ) and the following observations are made.

- In PWM waveforms fundamental voltage increases with increase in modulation index with other parameters remaining constant.
- The voltage variation is linear with modulation index, however, with increase in carrier frequency dominant harmonics occur at higher frequency.
- Variation of frequency to control speed requires simultaneous voltage control and carrier frequency control in drives. Carrier frequency adjustment is

necessary to retain switching frequency of devices / second constant. Voltage control is necessary to maintain constant (V/f) operation.

A novel technique has been adopted for microprocessor implementation of SPWM. A formula was derived for calculating the switching points in a minimum time. Solving this equation switching points are calculated and PWM wave was simulated. Harmonic analysis has been carried out on this simulated waveform. An selection was done visually from three dimensional spectrum analysis to select number of carrier wave per half cycle and modulation index at which the inverter is supposed to offer enhanced performance. The optimized result are implemented in OFF-line basis. A single phase inverter was driven at various operating frequencies with the microprocessor generated gating signals. Experimental spectral analysis was not carried out because of the absence of spectral analyzer.

#### 4-2. RECOMMENDATION FOR FUTURE WORK:

Microcomputer implementation of sinusoidal pulse width modulation for gating pulse generation for inverters has been reported in many literature's. The analysis and realization so far done considered the OFF-line calculation due to the inherent computational complexity. In this thesis possibility of ON-line calculation has been described. On-line implementation can be done in future using fast microprocessor kit having powerful instruction sets. This research effort is a step towards this goal. Future work may include the on-line implementation for polyphase inverter control of this technique. In the simulation area polyphase simulation of this technique can be examined extensively. On-line spectral estimation can be considered for further study because it will offer judicious selection for closed loop control scheme by analyzing harmonics of the present waveform of the running system. Microcomputer implementation of single phase / poly phase inverter, converter, cycloconverter using SPWM waveform can be implemented in On-line basis as a general case for the generation of gating pulse for inverter, converter, cycloconverter respectively.

## REFERENCES

- [1] Pentium processors user's manual, Vol:3, 1993, Intel corporation, USA.
- [2] S. Vadivel, G. Bhuvaneshwari, and G. siridhara Rao, "A unified approach to the real-time implementation of microprocessor-based PWM waveforms". IEEE Transaction on power electronics, Vol.6, No.4, October 1991, pp. 565-574.
- [3] S.R. Bowes and M.J. Mount, " Microprocessor control of PWM inverters". IEEE proc., Vol. 128, Pt.B, No. 6, Nov.1981, pp. 123-127.
- [4] F.C Zach, R.Martinez, S. Keplinger and A. Seiser. " Dynamically optimal switching patterns for PWM inverter drives". IEEE transaction, ind. appl. Vol.2 1, No.4, Jul/Aug. 1985, pp. 320-326.
- [5] M.Varnovitsky, " A microcomputer-based control signal generator for three phase switching PWM inverter", IEEE trans. ind. appl. vol. 1A-19, No.2, March / April 1983, pp. 187-194.
- [6] M.H. Rashid, " Power electronics circuits device and applications", Prentice Hall Inc., 1993, pp. 356 - 378, 484-487, 497-504.
- [7] R. G. Hoft, T.Khuwatsamrit, and R.McLaren, " Microprocessor applications for power electronics in the North America, " Microelectron. power electron. electr. drives conf. rec., Darmstadt, W.Germany, 1982 ,pp. 29-42.
- [8] Intel Microprocessor and Peripheral Handbook,1983.
- [9] F.Aldana, J. Piere ,and C.M.Penalver," Microprocessor control for power electronic systems,". Microelectron. power electron. electr. drives conf. rec., Darmstadt, W.Germany, 1982 , pp.111-115.
- [10] R. G. Hoft, R.W.McLaren, R.L.Pimmel,and K.P.Gokhale, " The impact of microelectronics and microprocessors on power electronics and variavle speed drives,". Int. power elec. and variable speed drives conf. rec., London, 1984, pp.191-198.
- [11] S. B. Dewan and A.Mirbod,"Microprocessor-based optimum control for four-quadrant chopper,". IEEE trans. ind. appl., IA-17, Jan./Feb.1981, pp.34-40.

- [12] D.A.Grant, M.Stevens, and J.A Houldsworth, " The effect of word length on the harmonic content of microprocessor-based PWM waveform generators,". IEEE trans. ind. apple., vol. IA-21, Jan. / Feb.1985, pp.218-225.
- [13] A.S.Fisher and R.W.McLaren, " Microprocessor control of a pulse-width modulated inverter," . IEEE / IAS annual meeting conf. rec., 1981, pp.660-665.
- [14] S.Sone and Y.Hori, " Microprocessor-based universal thyristor switch and its application to a PWM inverter for traction,". IEEE trans. ind. electron. conf. rec., vol. IECI-28 May 1981, pp.162-167,.
- [15] S.Bolognani, G.S.Buja, and D.Longo , "Hardware and performance effective microcomputer control of a three-phase PWM inverter,". Int. power electronic conf. rec., Tokyo, 1983, pp.360.371.
- [16] E.Dallago, D.Dontti, and P.Ferrari, " Application of power MOSFETs in a three phase inverter controlled by microprocessor,". Int. power electronic conf. rec., Tokyo, 1983, pp.1142-1149.
- [17] A.Bellini, C.D.Masro, G.Figalli, and G.Ulivi, "An approach for the implementation on a microcompute of the control circuit of variable frequency three-phase inverters," IEEE / IAS annual meeting conf. rec., 1981, pp.650-655.
- [18] V.V.Athani and S.M.Deshpande, " Microprocessor control of a three-phase inverte in induction motor speed control system," IEEE trans. ind. electron. conf. rec., vol. IECI-27, Nov./1980, pp. 291-298.
- [19] G.S.Buja and P.Fioni, "A microcomputer based, quasi-continuous output controller for PWM inverter,". IEEE/IECI conf. rec., 1980 , pp.107-111.
- [20] S.K.Tso and P.T.Ho , "Dedicated microprocessor scheme for thyristor phase control of multiphase converters,". Proc.IEE , vol. 128 , Mar. 1981, pp. 231-239.
- [21] P.L.G.Malapelle and L.A.M.Mortarino, "Microprocessor based controller for Ac/DC converters,". Microelectron. power electron. electr. drives conf. rec., Darmstadt, Germany, 1982, pp.163-170.

- [22] M.J.Case and P.Kulentic, "A microprocessor controller for the cycloconverter,". *microelectron. power electron. electr. drives conf. rec.*, Darmstadt, W.German, 1982, pp.171-181.
- [23] A.Bohuss, P.Buzas , and K.Ganszky, " Microcomputer controlled inverter for uninterruptible power supply ," *Microelectron power electron. electr. drives conf. rec.*, Darmstadt, W.Germany, 1982, pp. 203-206.
- [24] E.A. Rothwell, " The use of microprocessors and power transistors in modern uninterruptible power supplies," *Int. power elec. and variable speed drives conf. rec.*,1984, pp. 413-418.
- [25] T. Kutman, "Implementation of a microprocessor based technique for output filter optimization in UPS system,". *Int. power elec. and variable speed drives conf. rec.*, London, 1984, pp. 75-78
- [26] C.D. M. Oates, "Optimal PWM on a microcomputer,". *Int. power elec. and variable speed drives conf. rec.* London, 1984, pp.341-344.
- [27] G.N. Acharya, U.M.Rao, W. Shephard, S.S. Shekhawat, and Y.M.Nag, "Microprocessor based PWM inverter using modified regular sampling,". *IEEE / IAS annual meeting conf. rec.*, 1984, pp. 541-552.
- [28] G.S. Buja and P.Florini,"Microprocessor control of PWM inverters,". *IEEE trans ind. electron.*, vol. IF-29, Aug. 1982, pp.212-218.
- [29] S.Morinaga, Y.Sugiura, N.Muto, H.Okuda, K.Nandoh, H.Fujii, and K.Yajjima, "Microprocessor control system with I/O processing unit LSI for motor drive PWM inverter,". *IEEE trans ind. appl.*, vol. IA-20, Nov./Dec. 1984,pp..154-1553.
- [30]. S.J Lukas, "Microprocessor control of dc drive,". *IEEE / IAS annual meeting conf. rec.*, 1979, pp.881 - 885.
- [31] K. Kakiyama, T. Ohmae, N Azusawa and T. Koike, "Microprocessor based current and current rate controllers for speed control in industrial drives", *U.S. Japan seminar on microprocessor application in dc motor drive conf. rec.*,1982, pp.249 - 258.

- [32] F. Harashima and S. Kondo, "Microprocessor- based optimal speed control system of motor drives," IEEE / IECI confc. rec., 1981, pp. 252 - 257.
- [33] U Waschatz, "Adaptive control of electrical drives employing microprocessor", Microelectron, power electron, electr. drive conf. rec., Darmstadt, W. Germany, 1982, pp. 135 - 140.
- [34] W. J. O'Brien, "SILTRON microprocessor based adjustable speed drives", Industrial power system, june 1981, pp. 213-223.
- [35] Y.T. chan , a.J .Chmiel, and J. B Plant, " A micropropcessor based current controller for SCR-DC motor drives", IEEE trans. ind. electr. confr. instr., Vol IECI 27, Aug. 1980,pp. 543-567.
- [36] S. Franze, " Programmable counter simplifies measurement of feedback signal and generation of firing pulses for dc drives:, Microelectron. powerelectron., electron. drives conf. rec., darmstadt, W.Germany ,1982, pp. 245 - 248.
- [37] J.P Favre, " Microprocessor based speed control of an SCR DC motor drives," Microelectron. powerelectron., electron. drives conf. rec., Darmstadt, W.Germany , 1982, pp.265 - 271.
- [38] T. Izumi, M. Yamozoe, and T. nakano, "Microprocessor-based control system of thyristor converter fed dc motor drive", IEEE/IECI conf. rec., 1979, pp.284 - 288.
- [39] N.K De, S. Sinha, and A. K. Chattopadhyay, "Microcomputer as a programable controller for state feedback control of a DC motor employing thyristor amplifier", IEEE / IAS annual meeting conf. rec. 1984, pp. 586 - 592.
- [40] J.Caro, J. Dufour, and A. Jakubowicz, "A microprocessor based position control of DC drive taking into account the loads variation," IEEE / IAS annual meeting conf. rec. 1980, pp. 680 - 685.
- [41] W. S. Moak, and P. C Sen, " Induction motor drives with a microcomputer control system," IEEE / IAS annual meeting Conf. Rec. 1983, pp. 653 - 662.

- [42] K. Vallindras and A. Safacas, "Microprocessor control of an asynchronous machine fed by a PWM inverter", Int. pelec. conf. rec., Tokyo, 1983, pp. 1813-1821.
- [43] J. M. D. Murphy, L. S. Howard, and R.G. Hoft, "Microprocessor control of a PWM inverter induction motor drive", IEEE power elec. spec. conf. rec., 1979, pp. 344- 348.
- [44] Y. Hori and S. Sone, " An advanced harmonic control method of microprocessor based PWM inverter-fed traction system", Int. power elec. conf. rec., Tokyo, 1983, pp. 1544-1555.
- [45] W. S. Moak, and P. C. Sen, "Modelling and stability analysis of microcomputer control of induction motor drives", IEEE/IAS Int. semi. power conv. conf. rec., 1982, pp. 239- 251.
- [46] P. Misic and V. Vukovic, " Multiprocessor control unit for variable speed induction motor drives", Power elec. electr. drive conf. rec., Darmstadt, W. Germany, 1982, pp. 321 -328.
- [47] A. V. Bossche, and E. Speltincx, " Microprocessor based voltage control of squirrel cage induction motor drive", Microelectron. power elec. electr. drive conf. rec., Darmstadt, W. Germany, 1982, pp. 341 - 348.
- [48] A. Moschetti, "Microprocessor-based PWM system for inverter-fed squirrel cage motor traction drives", Microelectron. power elec. electr. drive conf. rec., Darmstadt, W. Germany, 1982, pp. 425 - 431.
- [49] A. Bellini, G. Figalli, and G. Ulivi, " A microcomputer- based optimal control system to reduce the effects of the parametric variations and speed measurement errors in induction motor drives", IEEE / IAS Annual meeting conf. rec., 1984, pp. 612- 617.
- [50] F. Harashima, S. Kondo, K. Ohnishi, M. Kajita and M. Susono, " Multi-microprocessor based control system for quick response induction motor drive," IEEE / IAS Annual meeting conf. rec., 1984, pp. 605 - 611.

- [51] P. Crnosija, Z. Kovacic, S. Mihalic, F Scof, Z. Peles, and NB. perlic, " Time optimal microprocessor armature current control of a thyristor based DC motor drive." 14th annual sym. on incremental motion control system and device conference record, University of Illinois-Urbana, 1985, pp.145 - 152.
- [52] P. L. G. Malapelle and L. A. M. Mortarino, " Microprocessor based controller for AC / DC converters," Microelectron. power electr. electr. drives conf. rec., Darmstadt, Germany, 1982, pp.163-170.
- [53] Rahman, M.A, Quacoe, J.E, Esmail, A.D and Choudhury, M.A., 'Spectral analysis of delta modulated inverter waveforms using discrete Fourier Transform', IEEE IECON'88 conference record, Vol II, Singapore,1988, pp. 338-343.
- [54] Harris, F. J. , ' On the use of windows for harmonic analysis with discrete Fourier Transform'. Proc. IEEE, 66, no.1 / 1978, pp. 51 -83.
- [55] Bowes, S.R., Clark, P.R., 'Transputer based Harmonic elimination PWM control of inverter drives', IEEE transaction on Industry applications, Vol.28, No. 1, Jan/ Feb 1992, pp. 73 - 79.
- [56] M. A . Choudhury, " Analysis of delta PWM inverter", M.Sc Engg. thesis, Memorial University of Newfoundland, August / 1984, Canada, pp. 5-5 - 5-11.
- [57] M. A. Choudhury, "Analysis of Delta Modulated PWM inverter with application to submersible motors", Ph.D Thesis, Memorial University of Newfoundland, Dec./ 1988. Canada, pp. 94 - 96 .
- [58] B.K. Bose, H.A Sutherland, " A high performance pulse width modulator for an inverter-fed drive system using a microcomputer,". IEEE trans. ind. appl. Vol.IA-19, No.2, March / April 1983, pp. 235-243.



## APPENDIX - 1

Program for simulating PWM waveform.

The program was developed using MATLAB software and the software was developed based Fast Fourier Transform ( FFT) algorithm.

T = time period of a cycle  
F = frequency of modulated wave  
N = number of carrier pulse per half cycle  
am = modulation index  
thetai = mid position of the ith pulse  
ti = starting of the switching instatnt of the ith pulse  
ga( t,ti,ti+1) = gate function for defining the PWM wave

### Main program

```
T = 1./f;
NP = 512;
X = ( pi / N );
Y = 2. * pi * f;
for i = 1:N;
ti = X * am * sin( X*i ) *(1./ Y );
thetai = (X*i)*(1./());
a(i) = thetai - ti/2.; Y;
b(i) = thetai + ti/2.;
end
t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
w1 = ga(t,a(i),b(i));
w2 = ga(t,a(i)+T/2,b(i)+T/2);
w = w1+w2;
wsum = wsum+w;
end
axis ([0 T 2 2]);
```

```
subplot(221), plot(t, wsum);
xlabel('Time in sec. '), ylabel('Magnitude');
fs = (NP1)/T;
f = (fs/NP)*(0:(NP/2)-1);
y = fft(wsum, NP);
pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
axis([0 3000 0 1.25]);
subplot(223), plot(f, spec(1:NP/2));
xlabel('Frequency in Hz. '), ylabel('Magnitude')
```

## APPENDIX - 2

Program for three dimensional harmonic analysis to select the number of carrier pulse and the modulation index at different frequencies. The program was developed using MATLAB software and the software was developed based on Fast Fourier Transform algorithm.

T = time period of a cycle  
F = frequency of modulated wave  
N = number of carrier pulse per half cycle  
am = modulation index  
thetai = mid position of the ith pulse  
ti = starting of the switching instatnt of the ith pulse  
ga( t,ti,ti+1) = gate function for defining the PWM wave

Main program

```
am=.9;
T=1./f;
NP=512;
X = (pi / N);
Y = 2. * pi * f;
for i=1:N;
ti = X * am * sin( X*i) * (1./ Y);
thetai = ( X*i) * (1./ Y);
a(i) = thetai ti/2.;
b(i) = thetai + ti/2.;
end
t = 0:T/511:T;
wsum = 0.0;
for i =1:N;
w1 = ga(t,a(i),b(i));
w2 = ga(t,a(i)+T/2,b(i)+T/2);
w = w1+w2;
wsum = wsum+w;
end
fs = NP1)/T;
fr = (fs/NP)*(0:(NP/2)1);
y = fft(wsum,NP);
```

```

pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
d1= spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am =.8;
T =1./f;
NP =512;
for i = 1:N;
    ti = X*am*sin(X*i)*(1./Y);
    thetai = (X*i)*(1./ Y);
a(i) = thetai/2.;
b(i) = thetai+ti/2.;
end
.t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
w1 = ga(t,a(i),b(i));
w2 = ga(t,a(i)+T/2,b(i)+T/2);
w = w1+w2;
wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)1);
y = fft(wsum,NP);
pyy = ( y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
d2 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am=.7;
T = 1./f;
NP = 512;
for i=1:N;
    ti = X*am*sin(X*i)/N)*(1./Y);
    thetai = ((X*i)/N)*(1./(Y));
a(i) = thetai/2.;
b(i) = thetai+ti/2.;

```

```

end
t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
w1= ga(t,a(i),b(i));
w2 = ga(t,a(i)+T/2,b(i)+T/2);
w = w1+w2;
wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)-1);
y = fft(wsum,NP);
pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
d3 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am=.6;
T = 1./f;
NP = 512;
for i=1:N;
ti = X*am*sin(X*i)/N*(1./Y);
thetai = ((X*i)/N)*(1./Y);
a(i) = thetai/2.;
b(i) = thetai+ti/2.;
end
t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
w1= ga(t,a(i),b(i));
w2 = ga(t,a(i)+T/2,b(i)+T/2);
w = w1+w2;
wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)-1);
y = fft(wsum,NP);
pyy = (y.*conj(y))/(NP/2);

```

```

spec = sqrt(pyy/(NP/2));
d3 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am=.5;
T = 1./f;
NP = 512;
for i=1:N;
    ti = X*am*sin(X*i)/N)*(1./Y);
    thetai = ((X*i)/N)*(1./(Y));
    a(i) = thetai/2.;
    b(i) = thetai+ti/2.;
end
t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
    w1= ga(t,a(i),b(i));
    w2 = ga(t,a(i)+T/2,b(i)+T/2);
    w = w1+w2;
    wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)-1);
y = fft(wsum,NP);
pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
d3 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am=.4;
T = 1./f;
NP = 512;
for i=1:N;
    ti = X*am*sin(X*i)/N)*(1./Y);
    thetai = ((X*i)/N)*(1./(Y));
    a(i) = thetai/2.;
    b(i) = thetai+ti/2.;
end

```

```

    t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
    w1= ga(t,a(i),b(i));
    w2 = ga(t,a(i)+T/2,b(i)+T/2);
    w = w1+w2;
    wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)-1);
y = fft(wsum,NP);
pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
d3 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am=.3;
T = 1./f;
NP = 512;
for i=1:N;
    ti = X*am*sin(X*i)/N*(1./Y);
    thetai = ((X*i)/N)*(1./Y);
    a(i) = thetai/2.;
    b(i) = thetai+ti/2.;
end
    t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
    w1= ga(t,a(i),b(i));
    w2 = ga(t,a(i)+T/2,b(i)+T/2);
    w = w1+w2;
    wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)-1);
y = fft(wsum,NP);
pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));

```

```

d3 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am=.2;
T = 1./f;
NP = 512;
for i=1:N;
    ti = X*am*sin(X*i)/N*(1./Y);
    thetai = ((X*i)/N)*(1./(Y));
    a(i) = thetai/2.;
    b(i) = thetai + ti/2.;
end
t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;
    w1 = ga(t,a(i),b(i));
    w2 = ga(t,a(i)+T/2,b(i)+T/2);
    w = w1+w2;
    wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)1);
y = fft(wsum,NP);
pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
d3 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
am=.1; T = 1./f; NP = 512;
for i=1:N;
    ti = X*am*sin(X*i) /N*(1./Y);
    thetai = ((X*i) /N)*(1./(Y));
    a(i) = thetai ti/2.;
    b(i) = thetai + ti/2.;
end
t = 0:T/511:T;
wsum = 0.0;
for i = 1:N;

```



```

w1= ga(t,a(i),b(i));
w2 = ga(t,a(i)+T/2,b(i)+T/2);
w = w1+w2;
wsum = wsum+w;
end
fs = (NP1)/T;
fr = (fs/NP)*(0:(NP/2)-1);
y = fft(wsum,NP);
pyy = (y.*conj(y))/(NP/2);
spec = sqrt(pyy/(NP/2));
d3 = spec(1:NP/4);
clear spec; clear am; clear T; clear NP; clear ti; clear thetai; clear t; clear wsum;
clear w1; clear w2; clear w; clear fs; clear y; clear pyy; clear fr;
sp=[d1' d2' d3' d4' d5' d6' d7' d8' d9'];
mesh(sp,[65,75])

```

### APPENDIX - 3

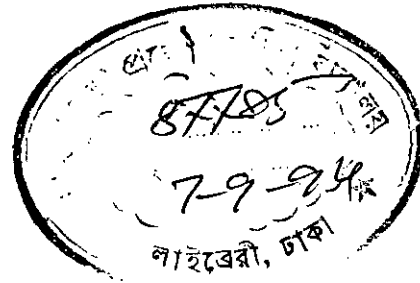
#### PROGRAMME FOR CALCULATING SWITCHING INSTANTS

```
#include <stdio.h>
#include <math.h>
main()
{
int N,i,f,T1on[50],Wordcnton[50],Wordcntoff[50],T1off[50];
float m,fclock;
int Thetaif[50],Thetais[50],Ti[50],T1[50],ti[50],t1[50];
FILE *fp;
printf ( " Enter the value of N: ");
scanf ("%d",&N);
printf ( " Enter the value of f: ");
scanf ( " %d", &f);
printf ( " Enter the value of m: ");
scanf ( " %f", &m );
fclock =(3.05 /25);
for ( i=0;i<= N1;i++)
{
T1on[i] = ((m/(2.0*(float)N*(float)f))*sin((M_PI*i) / (float)N)*1E+06);
Thetaif[i] =(((float)i / (2.0*(float)N*(float)f))*1E+06);
Ti[i] = Thetaif[i] T1on[i]/2;
T1[i] = Thetaif[i] + T1on[i]/2;
Wordcnton[i]=(T1on[i]*fclock);
}
for(i=0;i<=N1;i++)
{
if(i==0 || i==N1)
T1off[i]=Ti[1];
else
T1off[i]=(Ti[i+1]T1[i]);
Wordcntoff[i]=(T1off[i]*fclock);
}
if((fp = fopen("fdata5","w+")) != NULL)
{
fprintf(fp,"%s%d%s%f%s%d \n", " The data set for f=",f, "Hz, m =", m ," and N
=",N );
```

```

fprintf(fp,"%s\n"," T1on time T1off time Hexvalue(T1on) Hexvalue(T1off)");
fprintf(fp,"%s\n"," ");
for(i=0;i<=N1;i++){
printf(" %5d %5d %X %X \n",
T1on[i],T1off[i],Wordcnton[i],Wordcntoff[i]);
fprintf(fp,"%5d %5d %X %X \n",
T1on[i],T1off[i],Wordcnton[i],Wordcntoff[i]);
}
}
fclose(fp);
}

```



## APPENDIX- 4.1

### ASSEMBLY LANGUAGE PROGRAM FOR SINGLE PHASE INVERTER GATING PULSE GENERATION USING SOFTWARE COUNTER)

ADDRESS	OPCODE	LEBEL	MNEMONICS	COMMENT
C000	31 00 CF		LXI SP CF00	INITIALIZE STACK POINTER
	3E 90 D3 DB		MVI A,90 OUT DB	CONFIGURE 8255 PPI PORT A AS INPUT PORT ANDPORT B AS OUTPUT PORT
C007	21 00 CA	LOOP:	LXI H, CA00	POINT DATA LOCATION
	46		MOV B,M	INITIALIZE COUNTER TO KEEP TRACK CYCLE
C00B	3E 00 D3 D9	PHC:	MVI A,00H OUT D9	TURN OFF T1
	23		INX H	
	05		DCR B	
	CD 51 C0		CALL DELAY	WAIT FOR OFF TIME
	3E 01 D3 D9		MVI A 01H OUT D9	TURN ON T1
	23		INX H	
	05		DCR B	
	CD 51 C0		CALL DELAY	WAIT FOR ON TIME
	3E 04 D3 D9		MVI A, 04H OUT D9	TURN ON T11
	23		INX H	TO COMMUTATE T1
	CD 51 C0		CALL DELAY	
	05		DCR B	CHECK FOR END OF
	CA 2C C0		JZ NHC	POSITIVE HALF CYCLE
	C3 0B C0		JMP PHC	REPEAT POSITIVE HALF CYCLE
C02C	21 00 CA	NHC:	LXI H CA00	START NEGATIVE HALF CYCLE
	46		MOV B, M	
C030	3E 00 D3 D9	NHC1:	MVI A 00H OUT D9	TURN OFF T2
	23		INX H	
	05		DCR B	
	CD 51 C0		CALL DELAY	WAIT FOR OFF TIME

	3E 02 D3 D9		MVI A, 02H OUT D9	TURN ON T2
	23		INX H	
	05		DCR B	
	CD 51 C0		CALL DELAY	WAIT FOR ON TIME
	3E 08 D3 D9		MVI A, 08H OUT D9	TURN ON T22 TO COMMUTATE T2
	23		INX H	
	CD 51 C0		CALL DELAY	
	05		DCR B	CHECK FOR END OF
	CA 07 C0		JZ LOOP	NEGATIVE HALF CYCLE
	C3 30 C0		JMP NHC1	REPEAT NEGATIVE HALF CYCLE
C051	4E	DELAY:	MOV C, M	DELAY ROUTINE
C052	0D	DELAY1:	DCR C	
	1B		DCX D	DUMMY INSTRUCTION
	00		NOP	DUMMY INSTRUCTION
	C2 52 C0		JNZ DELAY1	
	C9		RET	

APPENDIX- 4.2

**ASSEMBLY LANGUAGE PROGRAM FOR THE GENERATION OF GATING PULSE FOR SINGLE PHASE INVERTER USINGG 8253 PROGRAMABLE TIMER**

ADDRESS	OP-CODE	LABEL	MNEMONICS	COMMENT
C000	31 00 CB		LXI SP, CB00H	INITIALIZE STACK POINTER
	3E 90 D3 DB		MVI A, 90H OUT DB	CON. 8255 PPI P <sub>A</sub> INPUT PORT P <sub>B</sub> OUT PORT
	3E 31 D3 CB		MVI A, 31H OUT CB	CONFIG. 8253 TIME- 0,16 BIT MODE ZERO.
C00B	06 0C	LOOP:	MVI B, 0CH	SET SWITCHING POINT COUNTER
	21 00 CA		LXI H, CA00H	POINT TO DATA LOCATION
	3E 1B 30 FB		MVI A, 1BH SIM EI	SET INTERRUPT MASK AND ENABLE INTERRUPT
	3E 00 D3 D9		MVI A, 00H OUT D9	TURN OFF T1 THYRISTOR
C018	7E D3 C8 23 7E D3 C8	LOOP1:	MOV A, M OUT C8 INX H MOV A, M OUT C8	LOAD 16 BIT COUNT VALUE TO TIMER ZERO
	76		HALT	WAIT FOR INT.
	23 05 CA 0B C0 C3 18 C0		INX H DCR B JZ LOOP JMP LOOP1	POINT TO NEXT DATA LOCATION, CHECK FOR END OF HALF CYCLE, CONTINUE CURRENT HALF CYCLE
C0 40	DB D8 E6 01H 2F D3 D9 FB C9	ISR:	IN D8 CPI 01H CMA OUT D9 EI RET	CHECK OUTPUT DATA, AND TOGGLE OUTPUT DATA. ENABLE INTERRUPT AND RETURN FROM ISR
The intrrupt transfer point for SDA-85 kit used for experimentation is FFB1				
	C3 40 C0	INTERRUPT TRANSFER POINT	JMP ISR	JUMP TO INTERRUPT SERVICE ROUTINE