NUMERICAL ALOGRITHMS FOR PWM MODULATORS

by

WALTER BATTMAN GREEN

Bachelor of Science in Engineering

Master of Science in Engineering

08

Submitted in part fulfilment of the Degree of Doctor of Philosophy in the Department of Electrical Engineering, University of Natal, Durban, South Africa.

March 1989

I hereby declare that all the material incorporated in this thesis is my own original unaided work except where specific acknowledgement is made by name or in the form of a numbered reference. The work contained herein has not been submitted for a degree at any other University.

Signed: 

ABSTRACT

The development of a simple efficient Pulse Width Modulation (PWM) modulator has been a goal for many research workers. In general three techniques have been used, namely; the analogue triangular wave technique; the use of look–up tables, and the use of Analogue to Digital converters together with analogue circuitry. The modulator described in this thesis is based on an iterative numerical algorithm, and is thus fundamentally different from all previous techniques. The algorithm is limited only by the speed and precision of the associated digital circuitry and can achieve higher modulating frequencies with greater accuracy than can be realised using any of the methods that have previously been investigated. The use of high switching frequencies simplifies the design of filters to reduce both unwanted harmonics and acoustic noise.

In this thesis, an equation of a multiphase digital oscillator is derived which is simple to implement and will operate over a wide range of frequencies. The conditions for stable oscillation are derived, and two classes of oscillator are developed. It is shown how the frequency and amplitude of oscillations can be independently and continuously varied. The errors in computing the amplitude and frequency are analysed, and are shown to be cyclic. Upper bounds for the amplitude errors are derived.

Single and three phase PWM modulators are described and the implementation procedures for their practical realisation are developed. Two specific implementations of the algorithm are investigated and experimental results confirm theoretical analyses.

The algorithm can be incorporated in the Space Vector Modulation (SVM) method of PWM, to improve the resolution at low speeds and to enable the SVM technique to be applied at high gear ratios.

A 3–phase 16–bit PWM modulator was built and operated satisfactorily with a pulse switching frequency of 20 kHz and an output frequency range of 1000:1.

Acknowledgements

The author would like to record his sincere gratitude to the folowing persons and organisations for their invaluable assistance during the project:

to Prof R G Harley, my supervisor, for his advice, help, guidance and support,

to the staff of the Electrical Engineering Workshop for their assistance in constructing the modulator and switching circuits,

to Dr Kreer, Dr M Shanblatt and Prof Wiley of Michigan State University, Lansing, Michigan for their advice and guidance, and for the provision of computing and library search facilities,

to the Foundation for Research and Development for financial support to complete project,

to Mr W Law of Powertronics for the use of GTO's and power diodes for the inverter circuit,

to Loughborough University for procuring the components for the MOSFET inverter,

to SAMES and the IC Design Centre for the use of their Integrated Circuit Design facilities,

to my wife, Vivian, who typed the manuscript, and who provided encouragement throughout the project,

and finally to all other persons who assisted in some measure, great or small, and without whose collective effort the project would not have been completed.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

T	Time interval between carrier pulses.
θ	Angle between adjacent phases.
ϕ	Arbitrary initial phase angle
U, U_0	Maximum amplitude of each phase.
U_n, U_{nj}	Maximum amplitude of each phase after n iterations.
δ	Step or incremental angle.
f	Frequency (Hz).
M, M'	Gear Ratio, or the number of steps required to complete one cycle.
N	Number of phases.
n	Number of steps or iterations.
$[U.Sin_j\phi]$	Row vector containing the projection of vectors $\bar{U}_j$ onto the imaginary axis.
$[U.Cos_j\phi]$	Row vector containing the projection of vectors $\bar{U}_j$ onto the real axis.
$\omega_c = 1/T$	Carrier frequency.
$\omega_m = 1/(M.T)$	Modulating frequency or inverter output frequency.
k	$\delta/\tan(\pi/3) = \delta/\sqrt{3}$.
ℓ	$\delta/\tan(\pi/5)$
ϵ	Truncation or round off error.
α_n	Angle of rotation of vector $\bar{U}$ after n steps.
λ_n	Difference in angle between α_n and the true angle of rotation after n steps.
$\xi_j(n)$	Pulse width error.
L	Wordlength of computer or logic circuit.
D	Integer value used to define δ.
δ_c	$2\pi/M'$ True angle of rotation for one step.
μ_n	n th Eigenvalue of a matrix.

[O] A stable 2–phase linear rotation matrix.

[U] An unstable 2–phase oscillator matrix.

[T] A stable 2–phase Simplex oscillator matrix derived from the [U] matrix.

[H] A stable 3–phase linear rotation matrix.

[I] A stable 3–phase Simplex oscillator matrix.

[G] A stable N–odd phase Simplex matrix in which the pulse width and frequency can be varied independently.

[F] A stable 5–phase Simplex oscillator matrix.

GLOSSARY

modulation — A process for superimposing one signal or (modulating) wave on another signal or (carrier) wave. The modulation process usually varies one or more of the following parameters of the carrier wave; amplitude, phase, frequency or pulse–width.

modulating signal or wave — The signal, wave or information which is superimposed onto the carrier wave. Synonyms for this term are: reference signal, desired signal, wanted frequency or inverter output frequency.

carrier — An abbreviation of the phrase "carrier signal" or "carrier wave"

PWM — Pulse Width Modulation. The amplitude of the modulating signal is directly proportional to the width of the pulses of the carrier wave.

carrier frequency ω_c — The average frequency of the carrier signal, or the average pulse repetition rate in a pulse modulated system.

modulating frequency ω_p — The frequency of the modulating signal. Synonyms for this term are: inverter frequency, inverter output frequency, desired frequency or reference frequency.

sampling period or sampling interval T — The average time interval between successive pulses. For PWM systems $T = 1/\omega_c$

gear ratio M — The ratio between the modulating frequency and the carrier frequency, or the number of carrier pulses that are used to approximate one cycle of the modulating wave.
For PWM systems $M = \omega_c/\omega_p$

CHAPTER ONE

INTRODUCTION

1.1 Principles of PWM Inverters

There are many processes in industry which require a variable speed motor. The DC motor has traditionally been used for this purpose, but it is a machine which requires a considerable degree of maintenance. An alternative approach is to use a variable frequency AC supply and the cheaper and more rugged induction motor. The development of the power diode and the thyristor has led to the introduction of the Pulse Width Modulated (PWM) variable frequency inverter, which provides a variable speed output from an induction motor.

To produce a variable frequency source of supply the AC mains is first rectified and then chopped into pulses of constant amplitude and varying widths as shown in the lower diagram of Fig 1.1. Narrow pulses correspond to low average voltages or currents and wide pulses correspond to high average voltages or currents. A typical cycle could start with a narrow pulse (low voltage); the widths of successive pulses are increased (in a sinusoidal fashion) until a wide pulse (or high voltage condition) is reached; thereafter the pulse widths are progressively decreased until a narrow pulse (or low voltage condition) is once again produced. A variable frequency supply can therefore be produced by varying the rate at which the pulse widths are lengthened or shortened.

An important parameter in all PWM inverters is the Gear Ratio (M), which is defined as the number of pulses occuring in one cycle of the inverter output wave. The gear ratio can also be defined as the ratio of the pulse switching (carrier) frequency and the required modulating frequency.

$$M = \text{No. of Pulses in one cycle} = \frac{\text{Carrier frequency}}{\text{Modulating frequency}}$$

A PWM inverter consists of a modulator and a power inverter system. The modulator produces the required PWM pulses for the power switching inverter, which controls the power fed to the induction motor.

A digital PWM modulator can be divided into two subsections; a numerical algorithm and a pulse generator. The numerical algorithm uses a series of arithmetic operations to

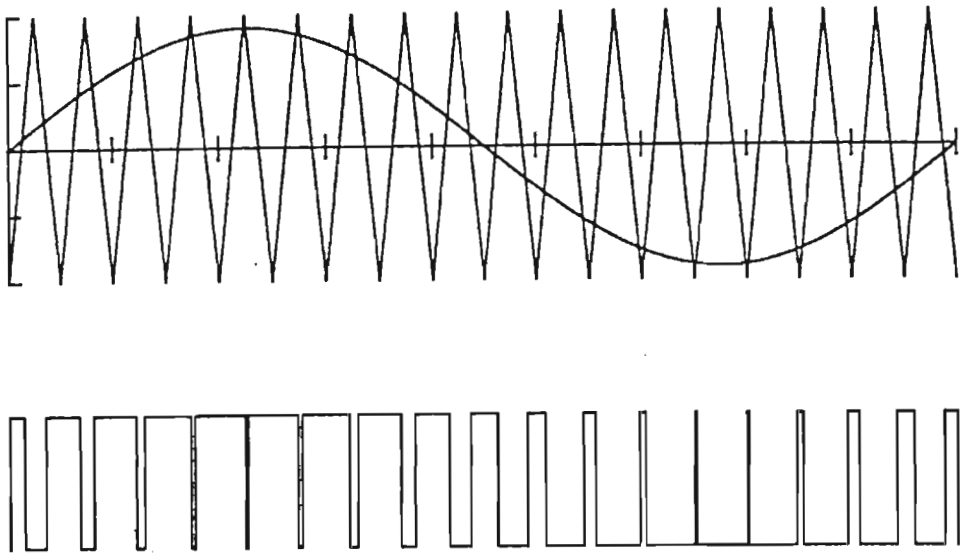


Fig. 1.1 Triangular Wave PWM

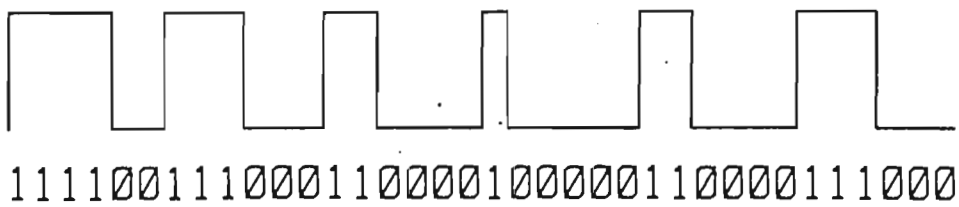


Fig. 1.2 Digital Serial PWM

calculate the binary values which represent the widths of the pulses. The algorithm may be implemented as a program for a microcomputer, or as a special purpose logic circuit. The pulse generator then converts the binary values into PWM pulses. In one sampling interval or iteration the values of all the widths of the N phases are calculated and then transmitted to the pulse generator.

This thesis describes the development and analysis of a new algorithm for a PWM modulator. The design principles for the power inverter switches and their associated drivers have been described elsewhere [9,10].

Existing PWM modulators are based on analogue or look–up table methods, and have a number of limitations. Analogue single phase PWM modulators are relatively simple and can be implemented using a comparator, a sine generator and a triangular wave generator as shown in Fig 1.1. The sine generator produces the modulating signal or the desired inverter output wave, which modulates the triangular (carrier) wave. Multi–phase modulators are much more complex because more than one pulse sequence has to be produced, whilst maintaining the correct phase difference between each of the modulated pulse sequences. At high motor speeds (high inverter output frequencies) it is necessary to phase lock the N modulating signals to the carrier in order to ensure a balance between the phases.

Green and Boys [11,12] have produced a successful analogue PWM modulator using this principle. A 3–phase signal generator produces sinusoids at the required frequency, and each phase is compared with a triangular wave which is 3*M times the frequency of the 3–phase oscillator. The fundamental frequency of the triangular wave generator and the desired 3–phase waves must be co–phasal, otherwise "—— machine losses will increase, and insulation life be degraded significantly if the line–to–line voltages differ by more than one percent" (Woll [35]). While Green and Boys have developed a satisfactory analogue implementation of the PWM modulator to generate 3–phase PWM signals, and have solved the phase control problems, high modulation frequencies are not possible. This is due to the ever increasing demands on the phase–lock–loop (PLL) circuitry at higher frequencies, the slew rate of the triangular wave generator and the speed of operation of the comparators. Another restriction of this method is the limited number of values that can be used as gear ratios. Green and Boys therefore used a different modulation technique (not requiring PLL) for operation at the higher frequencies. A secondary problem is the number of components required to implement the modulator. While the analogue triangular wave modulator was a major step forward it was clearly unsuitable for high modulation frequencies (350 Hz as used by Green and Boys).

One major disadvantage of using low modulation frequencies is the amount of acoustic noise produced by magneto–striction in the magnetic circuit of the motor. The use of switching frequencies above 15 kHz eliminates the acoustic noise problem.

With the advent of the microcomputer, a digital algorithm for a PWM modulator has been searched for by a number of investigators. Early developments were based on a look–up table [30], since the time taken to compute a sine value was prohibitively long. Dwyer and Ooi [7] describe a typical PWM modulator using a Z 80 microprocessor and a look–up table containing pre–computed sine values. The computer sequences through the look–up table and sends the sine value to a presettable down counter. The larger the value loaded into the counter, the longer the pulse. Inverter frequency (or motor speed) control is achieved by a rate multiplier, or by varying the sequence in which the data is retrieved from the look–up table.

Dwyer and Ooi also investigated the problem of unwanted harmonics generated by certain sequences of PWM pulses. They selected the widths of the pulses so that the amplitudes of the 5th, 7th and 11th harmonics were reduced.

De Buck et al [6] have developed an optimised waveform for a look–up based modulator with a gear ratio of 28. The waveform minimises the losses in an induction motor by reducing the harmonics, and is an improvement on the work of Dwyer and Ooi. The numerical algorithm developed in this thesis can suport a range of gear ratios from 20 to 20 000 which is far greater than the maximum value of 28 devloped by De Buck et al.

A novel variant of the look–up–table method is described by Edwards [8] who used a digital–serial PWM modulator. His look–up table contains strings of 1's and 0's instead of sine values. The longer the string of 1's the wider the pulse. Successive memory locations are read out, and placed in a parallel–to–serial converter. Frequency control is achieved by varying the rate at which the bits are read out of the serial converter. The major disadvantage of this method is the limited number of gear ratios that can be used, since each gear ratio has to be stored separately. Fig 1.2 shows how the pulse widths are varied using the Edwards method.

Grant et al [9] have produced a high speed 3–phase PWM modulator operating at pulse frequencies up to 18 kHz. Grant used a series of counters, a sine look–up table, a hard–wired multiplier, and an A/D converter to produce the PWM pulses. Various refinements have been made to the circuit which have enabled the pulse switching frequency to be increased to 20 kHz [10]. Whilst the circuit is fairly complicated it might

be possible to integrate all the functions into a single VLSI chip. Recent developments for enhancing the peformance of digital PWM modulators using the triangular–wave modulation technique are described by Agbinya [2], however significant stability and accuracy problems still exist.

1.2 Harmonic Control

The presence of unwanted harmonics or subharmonics in a PWM waveform can degrade the performance of an induction motor [35]. Bowes [3,4] has derived an expression for the amplitudes of harmonics present in PWM waveforms; he has also shown that subharmonics are produced if non–integer gear ratios are used. However for large gear ratios, subharmonics and harmonics are reduced to negligible proportions.

The speed of the inverter power switching device determines the upper limit for the carrier or pulse switching frequency. For slow or low speed switching devices it is important to provide some control of the gear ratio, in order to eliminate subharmonics and reduce the amplitudes of the harmonics. High speed switching devices support a wide range of large gear ratios, and produce a very small harmonic content which can be neglected in practical applications.

1.3 PWM Modulator Specifications.

The requirements for a digital PWM modulator can now be summarised.

a) To eliminate acoustic noise the carrier frequency must be as high as possible and at least 20 kHz.

b) The maximum value of the gear ratio should be as large as possible to reduce harmonics and enable low frequency inverter outputs to be produced.

e.g. for a 1 Hz inverter output signal, the gear ratio must be at least

$$M = \frac{20\ 000\ \text{Hz}}{1\ \ \text{Hz}} = 20\ 000$$

Note: a gear ratio of only 20 would therefore result in an inverter output frequency of 1 000 Hz.

c) The range of permissible gear ratios should be as large as possible to enable a wide range of frequencies (and motor speeds) to be produced.

d) The PWM numerical algorithm should be as simple as possible.

One method by which the above specifications could be met is by the use of a high speed microcomputer or VLSI circuit. Specification a) implies that the minimum number of arithmetic operations be used, especially multiplications or divisions. The need to limit the number of arithmetic operations is an important design parameter and has affected many aspects of the design of the PWM modulator described in the following chapters. Specifications b) and c) preclude the use of look–up tables or analogue techniques. They also prescribe the precision of the digital circuits that are used.

In any practical implementation of a PWM modulator it is also necessary to control the amplitude of the inverter output wave fed to an induction motor. For optimum performance the motor flux should be kept constant, except at the extreme ends of the frequency range. At high frequencies the voltage cannot increase indefinitely because the windings must not be overstressed; at low frequencies the resistances of the windings become significant and the V/f ratio has to increase to maintain flux. This implies that the voltage/frequency ratio should be capable of being varied.

1.4 Summary and New Findings

This thesis describes a new PWM numerical algorithm which meets the specifications given above and which can be implemented in a digital modulator using a minimal number of components.

The single and 2–phase oscillator equations have been known for some time. The development of the 3–phase oscillator matrix and related properties is claimed to be original, including the derivation and proofs in Appendices A to D. The application of the [I] oscillator matrix (derived in chapter 3) in a PWM modulator is also claimed to be original work. Some of the findings in this thesis have already been published and patented [14,15,16,17]. There were no prior citations recorded in the patent searches for all three patent registrations.

The design of the power inverter circuits and harmonic analysis have been taken from published work and are acknowledged where relevant.

The design and performance of the digital modulator is based on the assumption that the rectified AC produces a DC to the inverter which does not contain any significant amount of ripple. Additional equipment can be used to measure the amplitude of the DC link

voltage, and to modify the width of the pulses to substantially reduce the effects of the variations in the supply voltage.

Many types of PWM inverters have been produced by others, but all have been limited either by their low gear ratio or circuit complexity. All the digital modulators developed to date have been based on look–up tables, or combinations of A/D converters and counters. None has been based on a numerical, iterative procedure. The digital oscillator equation described in this thesis should enable a simple, high–gear–ratio PWM modulator to be constructed in a single VLSI circuit.

Chapters 2 to 4 discuss the theory and performance of multi–phase digital oscillator matrices and then evaluate a number of different matrices; the results clearly show that certain matrices are more useful than others. Chapters 5 to 7 show how the preferred oscillator matrices can be applied in practice.

CHAPTER TWO

TWO PHASE DIGITAL OSCILLATOR EQUATIONS

2.1 Introduction

Digital Oscillator equations are equations which which describe the numerical computation of the magnitude of the various phases of an N–phase oscillator. This chapter describes the derivation of a set of equations for a 2–phase oscillator.

The algorithms described in this thesis are based on a series of recursive linear difference equations. The new magnitude of the phase (and hence pulse width) $x(n+1)$ at step $(n+1)$ is calculated from an incremental value δ and the magnitudes of the phases $x(n)$ at the previous step (n) as follows

$$x(n+1) = f(x(n),\delta) \tag{2.1}$$

An example of a simple second order oscillator equation which is a solution to eq. (2.1) is

$$x(n+1) = (2.\mathrm{Cos}\delta).x(n) + x(n-1) \tag{2.2}$$

which is based on the fact that the values of $x(n)$ are equal to $\mathrm{Sin}(\phi+n\delta)$.

Given two suitable initial values for $x(0)$ and $x(1)$, $x(2)$ can be calculated. By repetitive application of eq. (2.2), $x(2)$, $x(3)$, ... $x(n)$ can be evaluated.

It should be noted that δ controls the rate of change of $x(n)$.

Eq. (2.2) is limited to producing a single sinusoidal function, and as Acha [1] has shown, truncation and round–off errors make the equation unstable. Let the cumulative errors be ϵ. Then instead of $x(n)$ being equal to $\mathrm{Sin}(\phi+n\delta)$,

$$x(n) \equiv \mathrm{Sin}(\phi+n\delta) + \epsilon$$

Owing to the unstable nature of eq. (2.2), the error term $\epsilon \to \infty$ as $n \to \infty$ and hence $x(n) \to \infty$ as $n \to \infty$. Thus for a limited range of values of n, eq. (2.2) produces a sinusoidal function, while for large n the arithmetic errors predominate. This undesirable characteristic is overcome by using a multi–phase oscillator equation.

Eq. (2.1) can be extended to include N phases by letting

$$x_1(n) = \mathrm{Sin}(\phi+n\delta) \tag{2.3.1}$$
$$x_2(n) = \mathrm{Sin}(\phi+\theta+n\delta) \tag{2.3.2}$$
$$x_3(n) = \mathrm{Sin}(\phi+2\theta+n\delta) \tag{2.3.3}$$
$$\vdots$$
$$x_N(n) = \mathrm{Sin}(\phi+(N-1)\theta+n\delta) \tag{2.3.N}$$

and $\overline{x}(n) = [\, x_1(n), x_2(n), x_3(n),\ldots x_N(n)\,]$

where the angle between adjacent phases is $\theta = 2\pi/N$.

Eq. (2.1) can be rewritten for the N phase system as

$$\overline{x}(n+1) = f(\overline{x}(n),\delta) \tag{2.4}$$

and this can be expanded into a series of equations of the form

$$x_i(n+1) = f_i(x_1(n),x_2(n),x_3(n),\ldots.. x_N(n),\delta) \tag{2.5.1}$$

To reduce the time taken to evaluate the N equations, only linear functions of x(n) are chosen, i.e.

$$x_i(n+1)=a_{i_1}(\delta).x_1(n)+a_{i_2}(\delta).x_2(n)+\ldots+a_{iN}(\delta).x_N(n) \tag{2.5.2}$$

Since there are N equations and each equation consists of N components it is possible to combine the N equations into a single matrix equation such as

$$\overline{x}(n+1) = \overline{x}(n) * \begin{bmatrix} a_{11}(\delta) & a_{12}(\delta) & \cdots & a_{1N}(\delta) \\ a_{21}(\delta) & a_{22}(\delta) & \cdots & a_{2N}(\delta) \\ \vdots & & \ddots & \vdots \\ a_{N1}(\delta) & a_{N2}(\delta) & \cdots & a_{NN}(\delta) \end{bmatrix} \tag{2.6}$$

where the N * N matrix is called an Oscillator matrix because the elements of $\overline{x}(n)$ vary in a sinusoidal manner.

The calculation of $\overline{x}(n+1)$ from $\overline{x}(n)$ and δ requires the evaluation of all the N eq. (2.5.2), and hence is referred to as one iteration of eq. (2.6). Furthermore it can be proved by induction that

$$\overline{x}(n+m) = \overline{x}(n) * \begin{bmatrix} a_{11}(\delta) & a_{12}(\delta) & \cdots & a_{1N}(\delta) \\ a_{21}(\delta) & a_{22}(\delta) & \cdots & a_{2N}(\delta) \\ \vdots & & \ddots & \vdots \\ a_{N1}(\delta) & a_{N2}(\delta) & \cdots & a_{NN}(\delta) \end{bmatrix}^m \qquad (2.7)$$

$$\text{for } m = \pm 1, \pm 2, \pm 3, \ldots .$$

One of the objectives of this thesis is to obtain a solution of eq. (2.4) so that stable sinusoidal functions are produced, and arithmetic errors are bounded. This chapter derives values for $a_{ij}(\delta)$ when $N = 2$ and chapter 3 develops the general solution for $a_{ij}(\delta)$ when $N = 3,5,7,9,\ldots$ (N odd).

It is simple to derive expressions for $a_{ij}(\delta)$ which include trigonometric functions of δ. By making a number of suitable approximations it is possible to obtain expressions for $a_{ij}(\delta)$ which are simple polynomials of δ only, and hence the number of arithmetic operations to calculate $a_{ij}(\delta)$ and $\overline{x}(n)$ are substantially reduced. This procedure is followed for the remainder of this chapter and in chapter 3.

2.2 Basic Solution for the Two Phase Oscillator

A solution to eq. (2.4) for the two–phase oscillator equation is derived from the trigonometric sine and cosine difference formulae.

$$U.\mathrm{Sin}(\phi+\delta) = U.\mathrm{Cos}\delta.\mathrm{Sin}\phi + U.\mathrm{Sin}\delta.\mathrm{Cos}\phi \qquad (2.8.1)$$

$$U.\mathrm{Cos}(\phi+\delta) = U.\mathrm{Cos}\delta.\mathrm{Cos}\phi - U.\mathrm{Sin}\delta.\mathrm{Sin}\phi \qquad (2.8.2)$$

Where U is the peak value of the phase magnitude or the maximum pulse width, and ϕ is any arbitrary angle. Eqs. (2.8) can be re–written in the form of eq. (2.6) by letting $x_1(n) = U.\mathrm{Sin}(\phi+n\delta)$ and $x_2(n) = U.\mathrm{Cos}(\phi+n\delta)$

hence $$\overline{x}(1) = \overline{x}(0) * \begin{bmatrix} \mathrm{Cos}\delta & -\mathrm{Sin}\delta \\ \mathrm{Sin}\delta & \mathrm{Cos}\delta \end{bmatrix} \qquad (2.9)$$

thus $$a_{11}(\delta) = a_{22}(\delta) = \mathrm{Cos}\delta$$
and $$a_{21}(\delta) = -a_{12}(\delta) = \mathrm{Sin}\delta$$

The 2 * 2 matrix defined in eq. (2.9) is one possible solution of eq. (2.6) and is defined as the [O] matrix in order not to confuse it with other solutions.

Eqs. (2.8) also represent the rotation of a vector of length $\overline{U}$ through an angle of δ radians (Kreysig [26]) and hence after $M = 2\pi/\delta$ steps the vector has rotated through one complete revolution (or one cycle of the modulating wave). Thus δ governs the rate at which the values of $\overline{x}(n)$ change. Furthermore the values of $\overline{x}(n)$ are the projections of the vector $\overline{U}$ on the real ($U.\mathrm{Cos}\phi$) and imaginary ($U.\mathrm{Sin}\phi$) axes and hence also represent the magnitudes of the widths of the PWM pulses needed to produce a two–phase ac supply. The implications of eqs. (2.8) are that the widths of the pulses for the next step or sampling interval for a two phase inverter can be calculated by multiplying the values of $\overline{x}(n)$ by the [O] matrix. It should be noted that only arithmetic operations (four multiplications, an addition and a subtraction) are required to calculate $\overline{x}(n+1)$ and there is no need to evaluate any trigonometric functions.

If δ is changed then $\mathrm{Sin}\delta$ and $\mathrm{Cos}\delta$ have to be evaluated before the new values of $\overline{x}(n+1)$ can be calculated. Thus any change in δ (or inverter output frequency) requires the evaluation of trignometric functions.

Fig 2.1 shows the function produced by repeatedly multiplying $\overline{x}(n)$ by [O]. The frequency is dependent on the value of δ only, and the correct phase relationship was maintained after δ was changed.

Eq. (2.9) is similar to the autonomous difference equations of the State Space model,

$$\overline{x}(n+1) = [G].\overline{x}(n) + [H].\overline{u}$$

$$\overline{y}(n) = [C].\overline{x}(n) + [D].\overline{u}$$

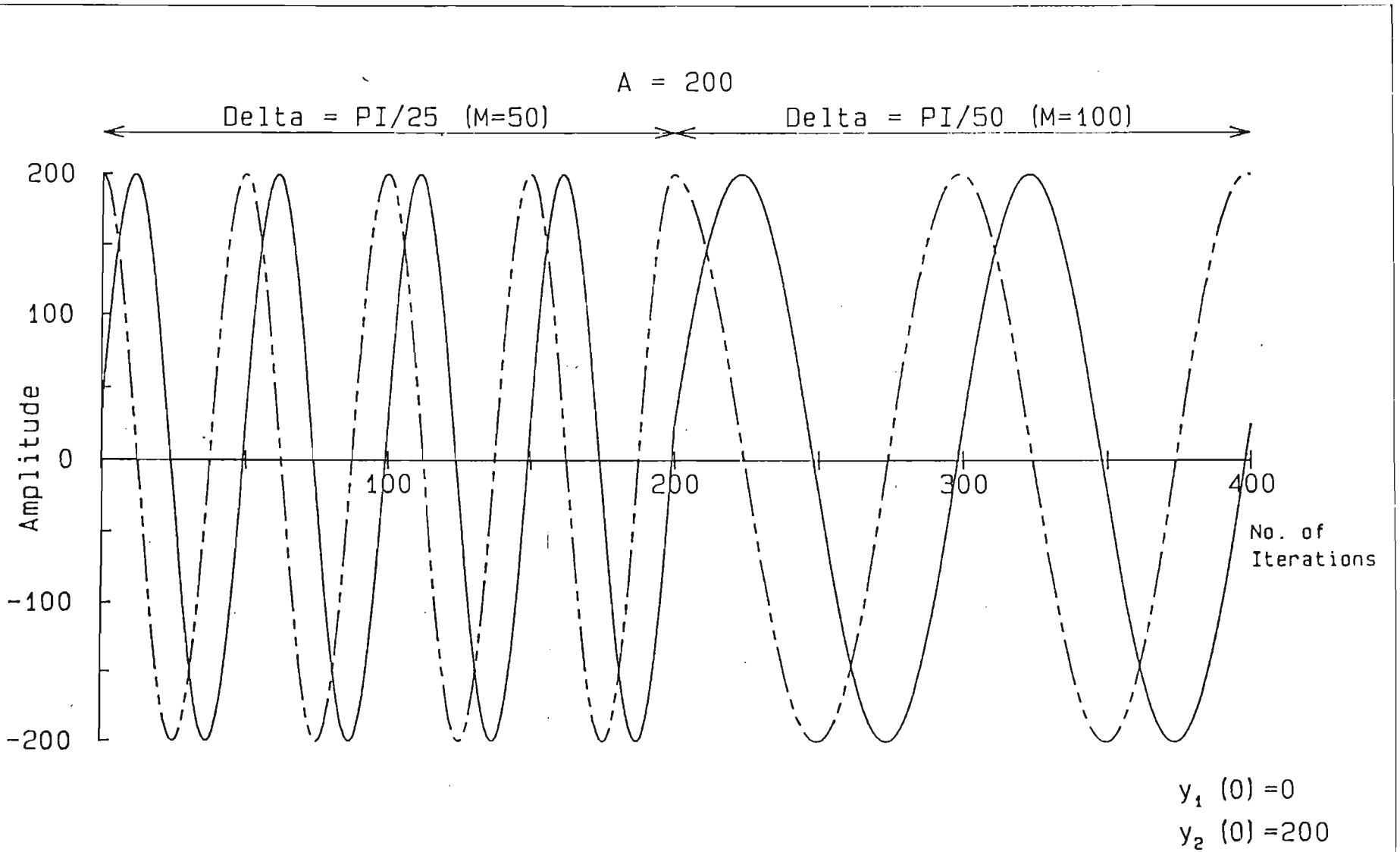


Fig. 2.1 Function Generated by the [O] Matrix

which is commonly used to describe control systems. $\overline{x}(n)$ are the state variables and the [O] matrix is equivalent to a 2 * 2 [G] matrix of the state space model.

The output values $\overline{y}(n)$ are chosen to be equal to $\overline{x}(n)$ to simplify the analysis. Hence the state space [C] matrix which relates the output values $y_i(n)$ to the state variables $x_i(n)$ is equal to the Identity matrix,

hence $x_i(n) \equiv y_i(n)$ for all i and n.

The [H] and [D] matrices are not required and are set to zero.

This equivalence enables the stability of all the oscillator equations to be analysed using the direct method of LIAPUNOV [18] for weakly or critically stable systems.

In most control systems it is desirable for the disturbances and variations about the reference or input signal to be damped away as quickly as possible. However for an oscillator the oscillations must not decay nor increase with time. This is equivalent to a control system which only has poles on the imaginary axis. Such a system is characterised by having eigenvalues of unity magitude and is defined by Liapunov[18] as being weakly stable. It is possible to show that the eigenvalues of the [O] matrix are $\mu_{1,2} = \mathrm{Cos}\delta \pm \mathrm{j.Sin}\delta$. Thus the magnitude of the eigenvalues $|\mu_{1,2}| = 1$ for all δ. Hence the function produced by the [O] matrix is an undamped series of sinewaves.

2.3 An Unstable Oscillator Matrix

Eq. (2.9) is a stable solution to eq. (2.6), however the need to calculate $\mathrm{Cos}\delta$ and $\mathrm{Sin}\delta$ every time δ is changed constitutes a major disadvantage. A more useful solution is where the value of δ can be changed without the need to calculate any of the coefficients $a_{ij}(\delta)$. The derivation of the so–called [U] matrix is the first step in formulating a solution to eq. (2.6) which meets the above requirement.

The [O] matrix can be simplified by noting that

$$\mathrm{Cos}\delta \approx 1 \qquad \text{and} \qquad \mathrm{Sin}\delta \approx \delta \qquad \text{for } \delta << 1 \tag{2.10.1}$$

The above approximations may produce an arbitrary function and hence the State Space state variables $x_j(n)$ are used, and letting

$$x_1(n) = U.\mathrm{Sin}(\phi+n\delta) \quad \text{and} \quad x_2(n) = U.\mathrm{Cos}(\phi+n\delta) \tag{2.10.2}$$

Substituting eqs. (2.10) into eqs. (2.8) yields

$$x_1(n) = x_1(n-1) + \delta.x_2(n-1) \tag{2.11.1}$$
$$x_2(n) = x_2(n-1) - \delta.x_1(n-1) \tag{2.11.2}$$

From eqs. (2.11) the [U] matrix is derived by equating terms and hence

$[U] = \begin{bmatrix} 1 & -\delta \\ \delta & 1 \end{bmatrix}$ or

$$a_{11}(\delta) = a_{22}(\delta) = 1$$

$$a_{21}(\delta) = -a_{12}(\delta) = \delta$$

The digital oscillator described by eqs. (2.11) has the undesirable property of increasing the values of $\overline{x}(n)$ as n increases as shown in Fig 2.2. The value of n does not affect the phase relationship between the components of $\overline{x}(n)$ and only the magnitudes are changed.

An explanation for the cause of the divergence of the function produced by the [U] matrix is based on one of the fundamental matrix theorems which states that " if all the elements of one row (or column) are multiplied by the same scalar v, then the determinant is v times the given determinant. " (Kreyszig [22] p247.)

$$\text{Now} \quad \mathrm{Det}[U] = 1 + \delta^2 \tag{2.12.1}$$
$$\text{hence} \quad \mathrm{Det}[U]^n = (1 + \delta^2)^n \tag{2.12.2}$$

This implies that the elements of [U] increase by $1+\delta^2$ at each step, so that

$$|\overline{x}(n+1)| = |\overline{x}(n)| * (1+\delta^2) \tag{2.13}$$

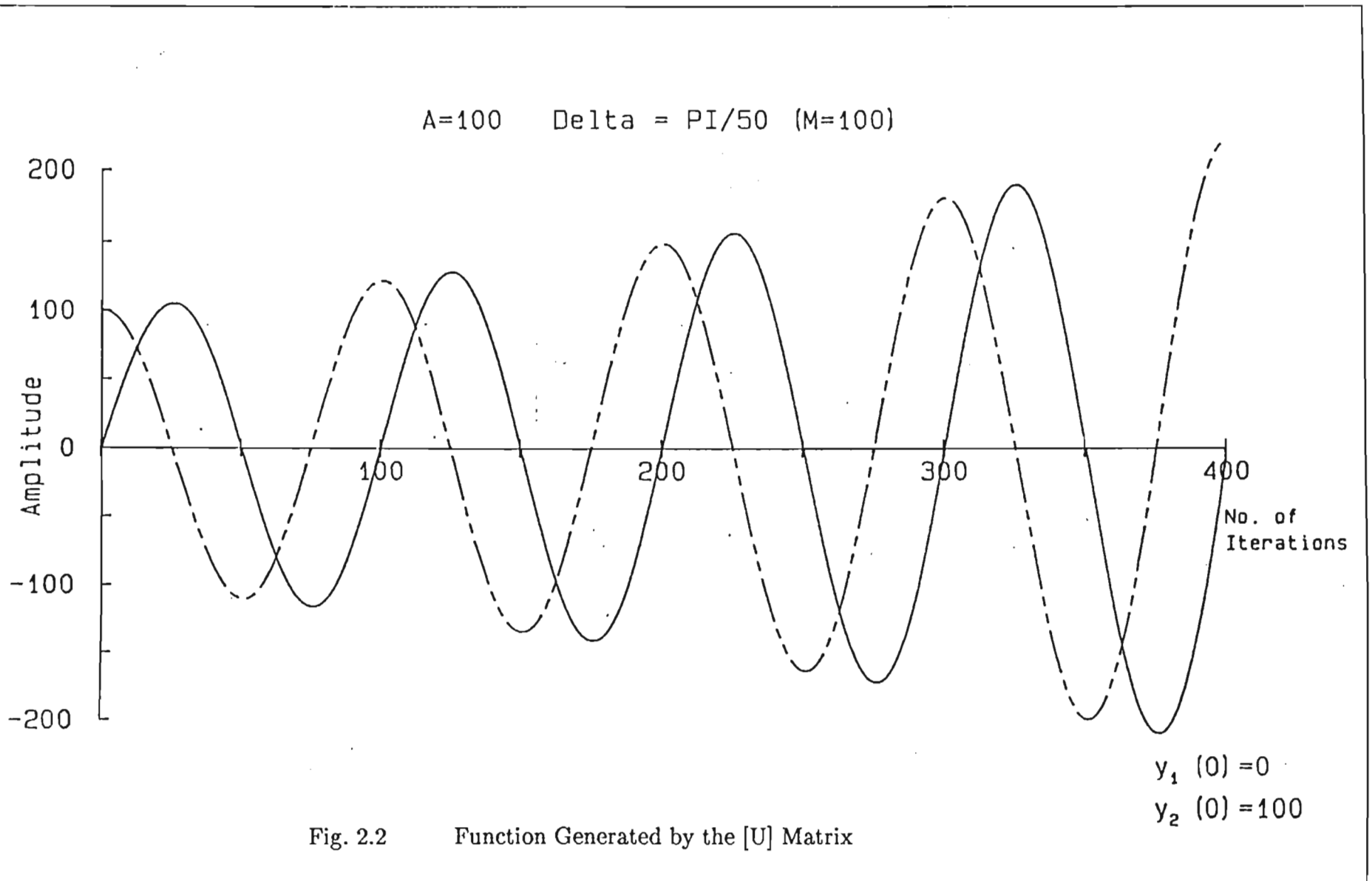


Fig. 2.2 Function Generated by the [U] Matrix

Thus a necessary condition for a stable oscillator equation is an oscillator matrix with a unity determinant.

The eigenvalues for the [U] matrix are $\mu_{1,2} = 1 \pm j\delta$ and hence $|\mu_{1,2}| = 1+\delta^2 > 1$. This is equivalent to a system which has poles in the right half of the s plane and is therefore unstable.

2.4 A Stable Two–Phase Oscillator Matrix

A two dimensional matrix with unity determinant can be derived from eqs. (2.11) by using the term $x_1(n)$ instead of $x_2(n-1)$ in eq. (2.11.2).

$$x_1(n) = x_1(n-1) + \delta.x_2(n-1) \qquad (2.14.1)$$
$$x_2(n) = x_2(n-1) - \delta.x_1(n) \qquad (2.14.2)$$

Substituting eq. (2.14.1) into eq. (2.14.2) and collecting like terms yields

$$x_2(n) = (1-\delta^2).x_2(n-1) - \delta.x_1(n) \qquad (2.14.3)$$

Combining eq. (2.14.1) and eq. (2.14.3) and then using matrix notation yields

$$\overline{x}(n+1) = \overline{x}(n) * \begin{bmatrix} 1 & -\delta \\ \delta & 1-\delta^2 \end{bmatrix} \qquad (2.15)$$

The 2 * 2 matrix in eq. (2.15) is defined as the [T] matrix. Det[T] = 1 for all δ even though the approximations given in eq. (2.10.1) are not valid. Furthermore it can be shown that the eigenvalues of [T] are given by

$$\mu_{1,2} = \frac{2-\delta^2}{2} \pm j.\frac{\delta}{2}.\sqrt{4 - \delta^2} \qquad (2.16.1)$$

Now for $|\delta| < 2$

$$|\mu_{1,2}| = \sqrt{(1-\delta^2/2)^2 + \delta^2.(1-\delta^2/4)} \qquad (2.16.2)$$

and $|\mu_{1,2}| = 1$ for all $|\delta| < 2$

Thus the [T] matrix is stable for values of $|\delta| < 2$ and the errors are bounded. Fig 2.3 shows the graph of the function generated by the [T] matrix where $M = 10$ or $\delta = \pi/5 = 0.628$. i.e. $\mathrm{Sin}\delta \neq \delta$. The amplitude of each cycle was the same even though there was considerable distortion. In Fig. 2.4 a value of $M = 100$ or $\pi/50 = 0.0628$ was used and a constant amplitude function has been produced.

For $|\delta| > 2$ the [T] matrix is unstable and the errors are unbounded since

$$|\mu_1| = |1-\delta^2/2 - \delta/2.\sqrt{\delta^2-4}| > 1 \qquad (2.16.3)$$

In Fig. 2.3 the number of steps required for one cycle is not $M = 10$, as expected, but slightly less than 10. Fig. 2.4 shows a smaller discrepancy between M and the true number of steps required to complete one cycle, The discrepancy is due to the approximations $\mathrm{Cos}\delta = 1$ and $\mathrm{Sin}\delta = \delta$. The magnitude and cause of the difference are analysed in detail in Chapter 4.

The pulse widths for a two–phase PWM inverter can be calculated from the pulse widths of the previous step and δ. The elements of the [T] matrix need not be calculated if eqs. (2.14) are used. Furthermore no trigonometric functions need be evaluated when δ is changed, and only two multiplications, an addition and a subtraction are required to compute the new pulse widths. A second advantage in using eqs. (2.14) is that there is no need to save the value of $x_1(n)$ for the second part (eq. 2.14.2) of the calculation.

The only restriction on the value of δ is that the magnitude must be less than 2, although small values of δ are required to limit the the errors introduced by the approximations $\mathrm{Cos}\delta = 1$ and $\mathrm{Sin}\delta = \delta$. The sign of δ affects the phase relationship between the two phases, and will control the direction of rotation of the vector $\bar{U}$ about the origin, and hence will also control the direction of rotation of a two–phase motor.

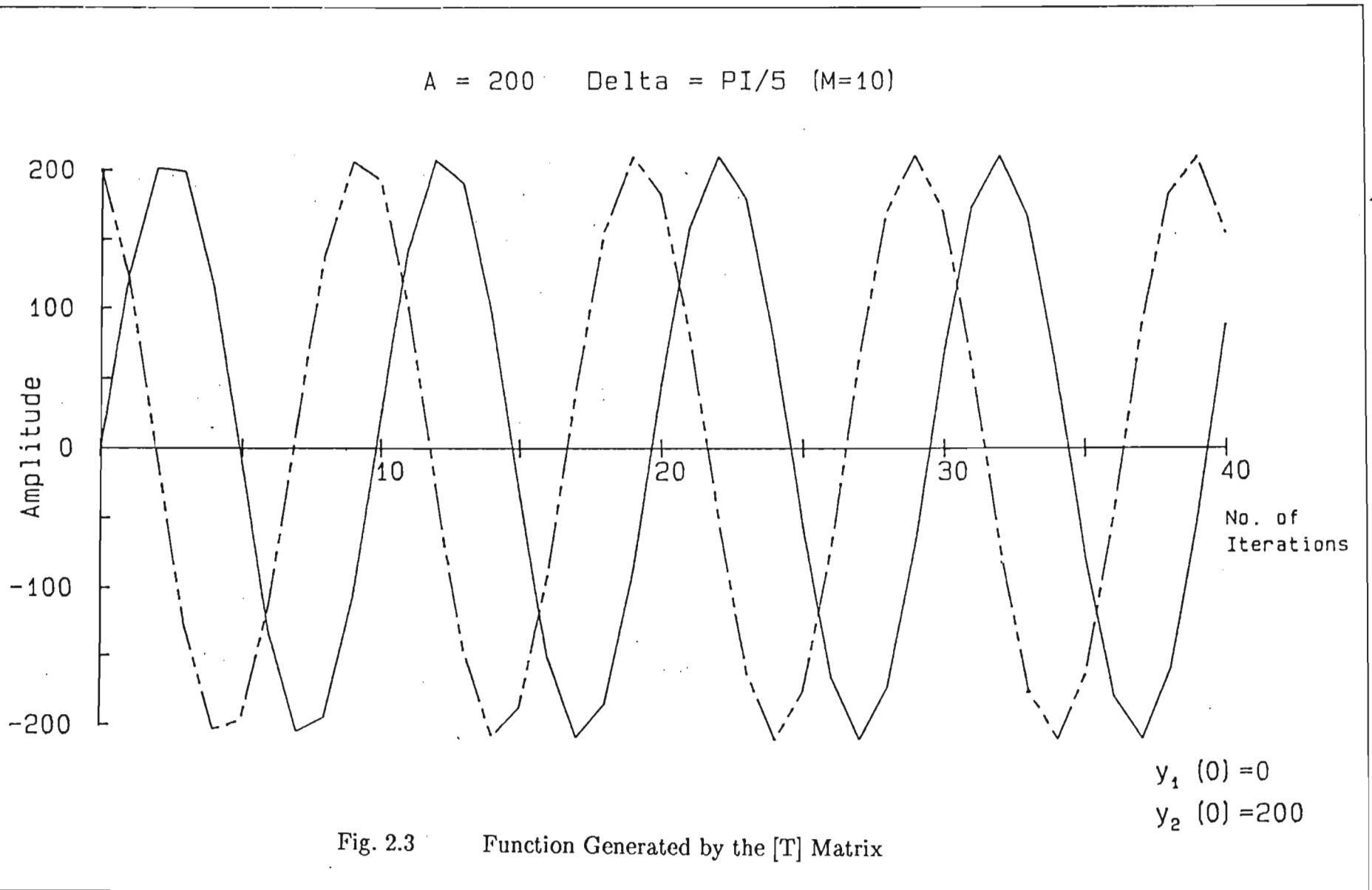


Fig. 2.3 Function Generated by the [T] Matrix

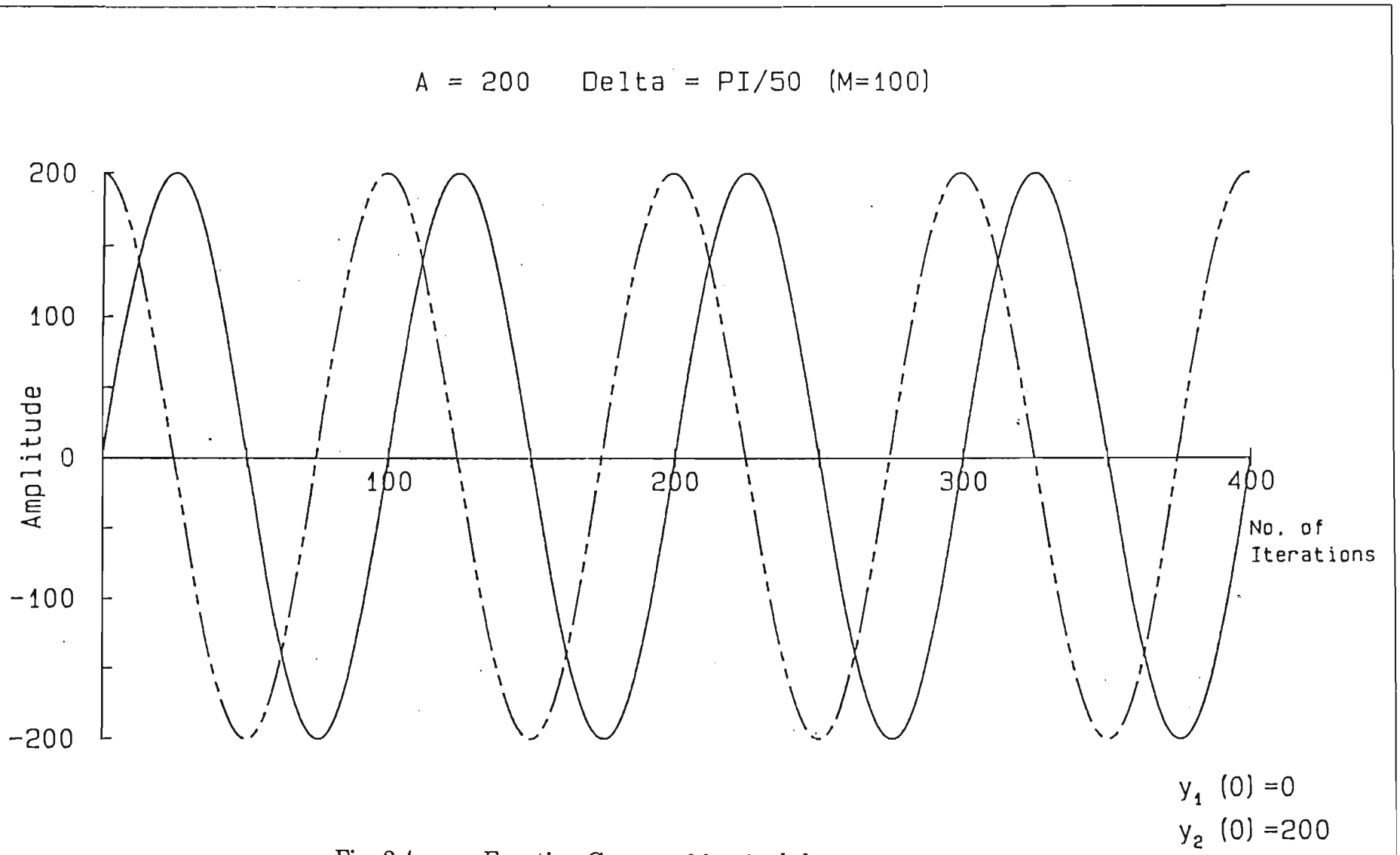


Fig. 2.4 Function Generated by the [T] Matrix

2.5 Conclusions

The concept of a digital oscillator matrix has been explained and conditions for stable oscillation have been derived.

The [T] matrix enables a simple digital oscillator to be implemented as a computer program or in a special purpose logic circuit. The approximations used in eq. (2.10.1) produce a slightly distorted sine function, and an inverter output frequency which is less than that expected from the value of $M = 2\pi/\delta$. A full discussion of the errors is given in Chapter 4.

Chapter 7 describes the implementation of a two–phase digital oscillator using the [T] matrix .

The two–phase [T] oscillator matrix has a limited number of applications, and can only usefully be applied to aplications which need a variable single frequency supply. eg. 17 Hz ringing generators in telephone exchanges.

The concepts developed in this chapter form the the basis of the theoretical development of the oscillator matrix for a multi–phase system, which is the subject of the next chapter.

CHAPTER THREE

THREE OR N–ODD PHASE OSCILLATOR MATRIX

3.1 Introduction

In this chapter, the fundamental equation for an N–odd multiphase digital oscillator matrix is developed, using a procedure similar to that described in Chapter Two. The conditions for stable oscillation are also derived; however it should be noted that for oscillator matrices with three or more phases, additional conditions need to be satisfied.

There are two known classes of matrices which are solutions to eq. (2.4) and are used as oscillator matrices. The first class is the well known linear rotation matrix, of which the [O] matrix, and the [H] matrix derived in this chapter, belong to this class. The properties of linear rotation matrices are described in many mathematical texts, (e.g. Kreyszig[26] and Marcus[29]). The elements of these matrices are trigonometric functions and describe the linear rotation of a vector. The second class of oscillator matrices, referred to as the Simplex matrices, and their associated stability conditions are derived in this chapter. Simplex matrices have elements that are polynomials of the step angle δ and describe a non–linear rotation of a vector. The [T] matrix developed in chapter 2 is a member of this class. The name Simplex has been chosen because of their simplicity of implementation in a logic circuit or computer program.

3.2 Multiphase Digital Oscillator Equations

A general N–odd phase solution to eq. (2.4) is derived from either the sine or cosine summation formulae

$$\mathrm{Sin}(\phi+\delta) = \mathrm{Cos}\delta.\mathrm{Sin}\phi + \mathrm{Sin}\delta.\mathrm{Cos}\phi \qquad (3.1.1)$$

$$\mathrm{Cos}(\phi+\delta) = \mathrm{Cos}\delta.\mathrm{Cos}\phi - \mathrm{Sin}\delta.\mathrm{Sin}\phi \qquad (3.1.2)$$

Eqs. (3.1) describe the rotation of a unit vector through δ radians and have to be extended to cater for N vectors. Assume that there are N vectors or phasors of length U and that each phasor is $\theta = 2\pi/\mathrm{N}$ radians apart. The magnitudes of all the pulse widths are the projections of the vectors onto the imaginary (or real) axis and at any instant are given by

$$[U.Sin_j\phi] = [U.Sin\phi,\ U.Sin(\phi+\theta),\ U.Sin(\phi+2\theta),U.Sin(\phi+3\theta),\ \ldots \ldots\ U.Sin(\phi+(N-1)\theta)\ ] \tag{3.2.1}$$

or for the real axis

$$[U.Cos_j\phi] = [U.Cos\phi,\ U.Cos(\phi+\theta),\ U.Cos(\phi+2\theta),\ U.Cos(\phi+3\theta),\ \ldots \ldots\ U.Cos(\phi+(N-1)\theta)\ ] \tag{3.2.2}$$

Eqs. (3.2) are used to calculate the initial values of the pulse widths for any arbitrary initial phase angle ϕ.

The main objective in the development of the multiphase oscillator matrix is to compute the magnitude of one phase or pulse width from the magnitudes of all the other phases while using the least possible number of arithmetic operations.

In eq. (3.1.1) the second term on the r.h.s. contains a $Cos\phi$ term and it is proved in Appendix A that for N–odd

$$\tan(\theta/2).Cos\phi = Sin(\phi+\theta)-Sin(\phi+2\theta)+ \ldots -Sin(\phi+(N-1)\theta) \tag{A.6}$$

or

$$Cos\phi = \frac{1}{\tan(\theta/2)}.\ [Sin(\phi+\theta) - Sin(\phi+2\theta) + \ldots - Sin(\phi+(N-1)\theta)] \tag{3.3}$$

Substituting eq. (3.3) into eq. (3.1.1)

$$Sin(\phi+\delta) = Cos\delta.Sin\phi + \frac{Sin\delta}{\tan(\theta/2)}\ [Sin(\phi+\theta)- \ldots -Sin(\phi+(N-1)\theta)] \tag{3.4}$$

This is the fundamental equation for a multiphase digital oscillator matrix, and shows how the new pulse width of one phase can be computed from the widths of all the other phases.

To reduce the number of multiplication operations required the approximations $Cos\delta \approx 1$ and $Sin\delta \approx \delta$ are used to give

$$Sin(\phi+\delta) \cong Sin\phi + \frac{\delta}{\tan(\theta/2)}\ [Sin(\phi+\theta) - Sin(\phi+2\theta) + \ldots \ldots - Sin(\phi+(N-1)\theta)] \tag{3.5}$$

A similar equation can be derived from the cosine formulae, using eq. (3.1.2)

$$\mathrm{Cos}(\phi+\delta) = \mathrm{Cos}\phi.\mathrm{Cos}\delta - \mathrm{Sin}\delta.\mathrm{Sin}\phi \tag{3.1.2}$$

and the identity proven in Appendix A,

$$-\tan(\theta/2).\mathrm{Sin}\phi = \mathrm{Cos}(\phi+\theta) - \mathrm{Cos}(\phi+2\theta) + \ldots - \mathrm{Cos}(\phi+(N-1)\theta) \tag{A.10}$$

Substituting eq. (A.10) into eq. (3.1.2) yields

$$\mathrm{Cos}(\phi+\delta) = \mathrm{Cos}\delta.\mathrm{Cos}\phi + \frac{\mathrm{Sin}\delta}{\tan(\theta/2)}\left[\mathrm{Cos}(\phi+\theta) - \mathrm{Cos}(\phi+2\theta) + \ldots \right.$$
$$\left. \ldots - \mathrm{Cos}(\phi+(N-1)\theta)\right] \tag{3.6}$$

This is similar in form to eq. (3.4), showing that the format of the fundamental equation applies in either Cosine or Sine form, or, the magnitude of each pulse width can be represented by the projection of the vectors onto either the real or imaginary axis.

3.3 Three Phase Oscillator Matrix

Consider the case for a 3–phase system for which $N = 3$ and let

$$\mathrm{Sin}_1(\phi) = \mathrm{Sin}(\phi) \tag{3.7.1}$$
$$\mathrm{Sin}_2(\phi) = \mathrm{Sin}(\phi+\theta) \tag{3.7.2}$$
$$\mathrm{Sin}_3(\phi) = \mathrm{Sin}(\phi+2\theta) \tag{3.7.3}$$

where $\theta = 2\pi/3$ radians

and $\tan(\theta/2) = \sqrt{3}$

Substituting eqs. (3.7) into eq. (3.4) yields

$$\mathrm{Sin}_1(\phi+\delta) = \mathrm{Cos}\delta.\mathrm{Sin}_1\phi + \frac{\mathrm{Sin}\delta}{\sqrt{3}}\left[\mathrm{Sin}_2\phi - \mathrm{Sin}_3\phi\right] \tag{3.8.1}$$

$$\mathrm{Sin}_2(\phi+\delta) = \mathrm{Cos}\delta.\mathrm{Sin}_2\phi + \frac{\mathrm{Sin}\delta}{\sqrt{3}}\left[\mathrm{Sin}_3\phi - \mathrm{Sin}_1\phi\right] \tag{3.8.2}$$

$$\mathrm{Sin}_3(\phi+\delta) = \mathrm{Cos}\delta.\mathrm{Sin}_3\phi + \frac{\mathrm{Sin}\delta}{\sqrt{3}}\left[\mathrm{Sin}_1\phi - \mathrm{Sin}_2\phi\right] \tag{3.8.3}$$

Substituting $s = \mathrm{Sin}\delta/\sqrt{3}$ and using the row vectors

$[\mathrm{Sin}_j\phi] = [\mathrm{Sin}_1\phi, \mathrm{Sin}_2\phi, \mathrm{Sin}_3\phi]$

and $[\mathrm{Sin}_j(\phi+\delta)] = [\mathrm{Sin}_1(\phi+\delta), \mathrm{Sin}_2(\phi+\delta), \mathrm{Sin}_3(\phi+\delta)]$

eqs. (3.8) can be rewritten using matrix notation yielding

$$[\mathrm{Sin}_j(\phi+\delta)] = [\mathrm{Sin}_j\phi] * \begin{bmatrix} \mathrm{Cos}\delta & -s & s \\ s & \mathrm{Cos}\delta & -s \\ -s & s & \mathrm{Cos}\delta \end{bmatrix} \tag{3.9}$$

where the 3 * 3 matrix is defined as the [H] oscillator matrix.

The [H] matrix is a linear rotation matrix which defines the rotation of three vectors (spaced 120^0 apart) by δ radians in the x,y plane. The important property of the [H] matrix is that it represents a linear transformation, where "the lengths of vectors are unchanged and the angles between vectors are not affected" (Kreyszig[26] p.281). The projections of the vectors onto the real or imaginary axes represent the widths of the pulses for each phase.

Other linear matrices were investigated, but none of them had the simplicity and flexibility of the Simplex class of matrices which are discussed in this chapter.

The function generated by the [H] matrix is shown in Fig. 3.1. It should be noted that the phase relationship is maintained, and that stable oscillation occurs even though the value of δ is increased. The number of steps to complete one cycle is $M = 2\pi/\delta$.

The elements of the [H] matrix are trigonometric functions of δ, and hence every time δ is changed, all the $a_{ij}(\delta)$ elements have to be evaluated. This problem can be solved by using eq. (3.5). Since $\delta/\sqrt{3}$ is a common term in all the equations relating to the following matrices the substitution

$$k = \delta/\sqrt{3} \tag{3.10}$$

is used to simplify the analysis.

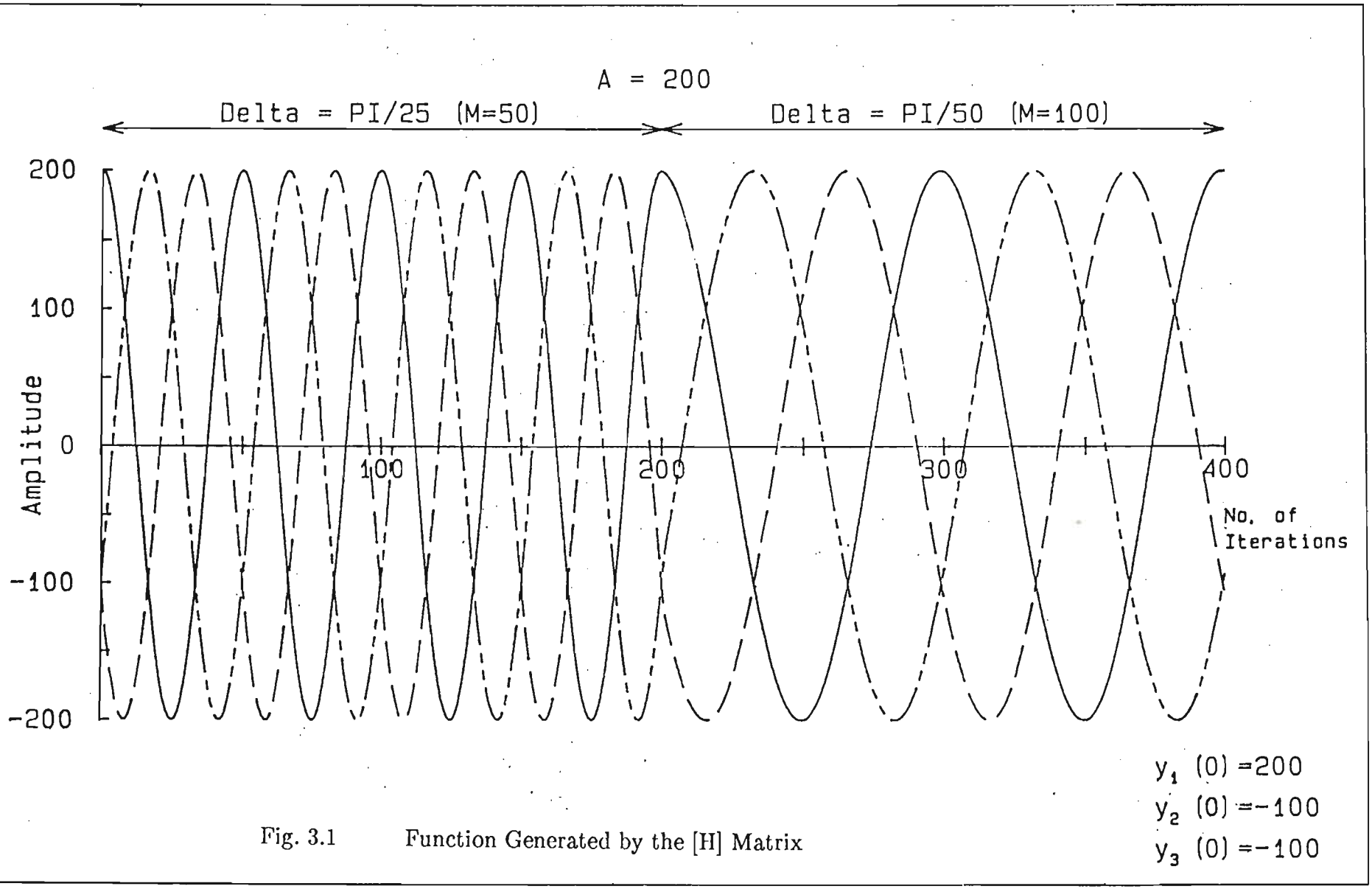


Fig. 3.1 Function Generated by the [H] Matrix

The approximations $\mathrm{Cos}\delta \approx 1$ and $\mathrm{Sin}\delta \approx \delta$ have been used in the derivation of eq. (3.5) and hence the trigonomtric identities used in eqs. (3.1) are no longer strictly valid. The use of the above approximations produces a simpler matrix, which has the following properties;

a) the function that is generated is not a true sinusoidal function (but is a very good approximation), and

b) the rotation of the vectors is not linear.

To clearly differentiate between the linear rotation of the [H] matrix and the non–linear rotation of the following Simplex matrices a change of variable is introduced. i.e.

$$x_j(n) \cong U.\mathrm{Sin}(\phi + n\delta) \tag{3.11}$$

and the row vector $[x_j(n)]$ is defined as

$$[x_j(n)] = [x_1(n), x_2(n), x_3(n), \ldots x_N(n)]$$

Substituting eq. (3.10) and eq. (3.11) into eq. (3.5) yields

$$x_1(n+1) = x_1(n) + k[x_2(n) - x_3(n)] \tag{3.12.1}$$

$$x_2(n+1) = x_2(n) + k[x_3(n) - x_1(n)] \tag{3.12.2}$$

$$x_3(n+1) = x_3(n) + k[x_1(n) - x_2(n)] \tag{3.12.3}$$

Using the row vector $[x_j(n)]$, eqs. (3.12) can be written in matrix notation yielding

$$[x_j(n+1)] = [x_j(n)] * \begin{bmatrix} 1 & -1 & k \\ k & 1 & -k \\ -k & k & 1 \end{bmatrix} \tag{3.13}$$

A graph of the function produced by the above matrix is shown in Fig. 3.2 and it can be seen that the magnitudes of all the phases uniformly increase in size. This is due to the determinant being greater than unity i.e.

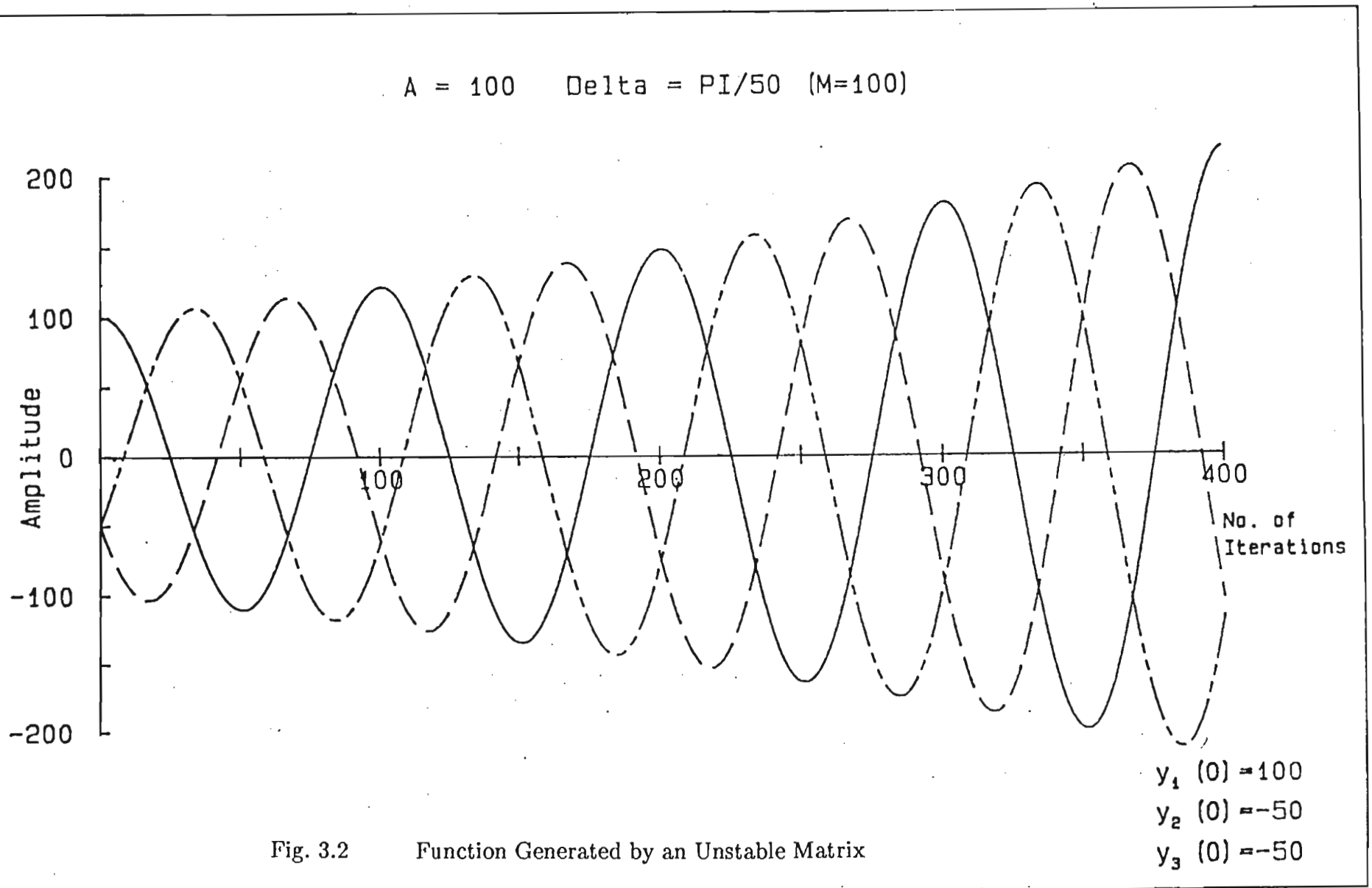


Fig. 3.2 Function Generated by an Unstable Matrix

$$\text{Det}\begin{bmatrix} 1 & -1 & k \\ k & 1 & -k \\ -k & k & 1 \end{bmatrix} = 1+3k^2 = 1+\delta^2.$$

It can also be shown that at least one of the eigenvalues of the above matrix is greater than one, and hence this matrix is unstable.

3.4 A Stable N–Odd Phase Oscillator Matrix

A stable oscillator matrix can be derived from eq. (3.5) by using a sequential calculation procedure whereby the new pulse widths for each of the phases are calculated sequentially.

Using eq. (3.5), together with k and $x_j(n)$ as defined above, yields

$$x_1(n+1) = x_1(n)+ k[x_2(n)- x_3(n)] \tag{3.14.1}$$

$$x_2(n+1) = x_2(n)+ k[x_3(n)- x_1(n+1)] \tag{3.14.2}$$

$$x_3(n+1) = x_3(n)+ k[x_1(n+1) - x_2(n+1)] \tag{3.14.3}$$

Writing eqs. (3.14) in matrix form requires the substitution of eq. (3.14.1) into eq. (3.14.2) and collecting like terms. A similar procedure is required to reduce eq. (3.14.3) to terms containing $x_j(n)$ only. The [I] oscillator matrix derived from eqs. (3.14) is

$$[I] = \begin{bmatrix} 1 & -k & k+k^2 \\ k & 1-k^2 & -k+k^2+k^3 \\ -k & k+k^2 & 1-2k^2-k^3 \end{bmatrix} \tag{3.15}$$

and hence $\alpha_{11}(\delta) = 1$ (3.16.1)

$\alpha_{12}(\delta) = \alpha_{31}(\delta) = -\delta\sqrt{3} = -k$ (3.16.2)

$\alpha_{21}(\delta) = \delta\sqrt{3} = k$ (3.16.3)

$\alpha_{13}(\delta) = k + k^2$ (3.16.4)

$\alpha_{22}(\delta) = 1 - k^2$ (3.16.5)

$$\alpha_{23}(\delta) = -k + k^2 + k^3 \qquad (3.16.6)$$

$$\alpha_{33}(\delta) = 1 - 2k^2 - k^3 \qquad (3.16.7)$$

The derivation of the [I] matrix and proof that this matrix satisfies all the conditions for stability given in sections 3.5 are shown in Appendix B. A graph of the stable sinusoidal function produced by the [I] matrix is shown in Fig 3.3.

3.5 Conditions for Stable Oscillation

In order to study the causes of instability of oscillator matrices it is necessary to use the real–imaginary or the x–y plane. The N–odd pulse widths are the projections of N vectors of length U which rotate about the origin. Ideally, all the vectors should rotate by δ radians at each step, and their lengths should not change. This is the case for the [H] matrix. For the [I] matrix the approximations $\mathrm{Sin}\delta \approx \delta$ and $\mathrm{Cos}\delta \approx 1$ cause the lengths of the vectors to change, however owing to the unity determinant the changes are cyclic.

The changes in lengths of the vectors are analysed by re–writing eqs. (3.14) in matrix form and letting the length of the j th vector after n rotations of approximately δ radians be represented by $\bar{U}_{nj}$. It is not possible to calculate $\bar{U}_{nj}$ directly, however it can be inferred from its components or projections on the real and imaginary axes.

Let U'_{nj} be the projection of the vector $\bar{U}_{nj}$ onto the imaginary axis and U''_{nj} be the projection of the same vector onto the real axis. $[I]^n$ is the $3 * 3$ matrix that describes the rotation of the vectors after n steps, and hence

$$U'_{nj}.\mathrm{Sin}_j(\phi+n\delta) = U_0.\mathrm{Sin}_j(\phi) \; * \; [I]^n \qquad (3.17.1)$$

$$U''_{nj}.\mathrm{Cos}_j(\phi+n\delta) = U_0.\mathrm{Cos}_j(\phi) \; * \; [I]^n \qquad (3.17.2)$$

Thus the lengths of the vectors after n steps are

$$U^2_{nj} = U'^2_{nj}.\mathrm{Sin}^2_j(\phi+n\delta) + U''^2_{nj}.\mathrm{Cos}^2_j(\phi+n\delta)$$

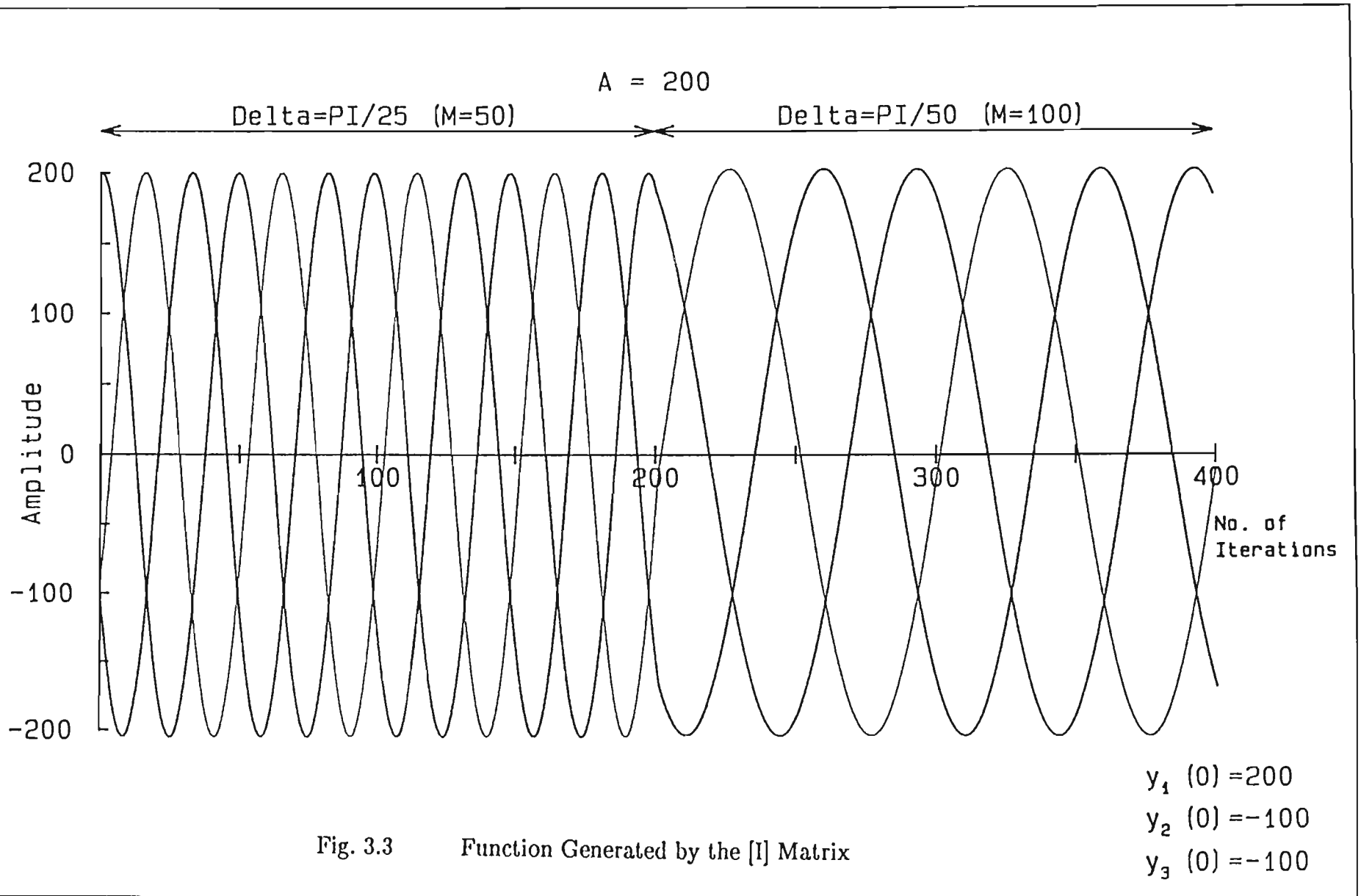


Fig. 3.3 Function Generated by the [I] Matrix

Hence the elements of $[I]^n$ state the relationship between U_0^2 and U_{nj}^2. It is important to note that $U_{nj}^{'}$ is not necessarily equal to $U_{nj}^{''}$ due to the approximations $\mathrm{Sin}\delta \approx \delta$ and $\mathrm{Cos}\delta \approx 1$. Several attempts were made to derive individual equations showing the relationship between $U_{n1}^2, U_{n2}^2, U_{n3}^2$ and U_0^2 but were unsuccessful due to the inter–relationship between U_{n1}^2, U_{n2}^2, and U_{n3}^2. A useful general equation can be derived by squaring and summing eqs. (3.17), which gives the sum of the squares of the lengths of the vectors after n steps.

Using the same reasoning and theorem as in section 2.4, it can be shown that since $\mathrm{Det}[I]^n = 1$ the sum of the squares of the lengths of the vectors will not increase or decrease monotonically with n.

It is possible for one vector to decrease in length while the other vectors increase in length. This is clearly shown by calculating the values for U_{1j} where

$$U_{11}^2 = U_0^2(1 + 3k^2) \tag{3.18.1}$$

$$U_{12}^2 = U_0^2(1 + 3k^3 + 3k^4) \tag{3.18.2}$$

$$U_{13}^2 = U_0^2(1 - 3k^2 - 3k^3 + 6k^4 + 9k^5 + 2k^6) \tag{3.18.3}$$

and $\sum_j U_{1j}^2 = U_0^2(3 + 9k^4 + 9k^5 + 2k^6) > 3U_0^2$

$\sum_j U_{1j}^2$ is greater than $3U_0^2$ because $U_{1j}^{'}$ and $U_{1j}^{''}$ are not the same.

$\sum_j U_{nj}^2$ does not increase monotonically, but varies in a sinusoidal manner as shown in Fig. 3.4 (due to the terms $\mathrm{Cos}(\phi+n\delta)$ and $\mathrm{Sin}(\phi+n\delta)$). Fig. 3.4 also shows the cyclic manner in which the lengths of the vectors vary with increasing n.

Thus a necessary, but not sufficient, condition for stable oscillation is that the determinant of the oscillator matrix must be unity.

A second condition for stability is required to ensure that one vector does not decrease monotonically while the others increase monotonically. This is achieved by introducing

the restriction that the sum of each column of the oscillator matrix be equal to unity.

Let each element of the [I] matrix be $\alpha_{ij}(\delta)$ and assume that for each column

$$\sum_i \alpha_{ik}(\delta) = Q = \sum_k \alpha_{kj}(\delta)$$

Let $[I'] = [I] * [I]$ and add the elements in each column.

Since $\alpha'_{ij}(\delta) = \sum_k \alpha_{ik}(\delta).\alpha_{kj}(\delta)$

$$\therefore \quad \sum_i \alpha'_{ij}(\delta) = \sum_i \sum_k (\alpha_{ik}(\delta).\alpha_{kj}(\delta))$$

Interchanging the order of summation and the terms in the brackets,

$$\sum_i \alpha'_{ij}(\delta) = \sum_k \sum_i (\alpha_{kj}(\delta).\alpha_{ik}(\delta)) = Q.\sum_k \alpha_{kj}(\delta) = Q^2$$

Thus if the sum of the elements in the columns is not unity, the values of $U.Sin_j\phi$ will change every time $U.Sin_j\phi$ is multiplied by [I]. If $Q = 1$ then for $[I]^n$, the sum of the elements of each column will always be unity for all n and the lengths of each of the vectors will not decrease or increase monotonically.

The above proof is a modification of a proof given in Hamming[20] p.83.

The above two conditions provide sufficient conditions for stable oscillation for the class of Simplex matrices generated from eq. (3.5).
These conditions may be summarised as follows

a) The Determinant of the oscillator matrix must be unity.
b) The sum of the elements of each column of the matrix must be unity.

The [I] matrix satisfies both conditions for all values of k and thus defines a stable oscillator matrix.

It is possible to derive a set of equations to construct matrices which satisfy both the criteria mentioned above, but such matrices become unstable and diverge because of the propagation of arithmetic or truncation errors. The [I] matrix does not suffer from this problem for a limited range of values of k. The stability criteria to control the propagation of errors is discussed in the next section.

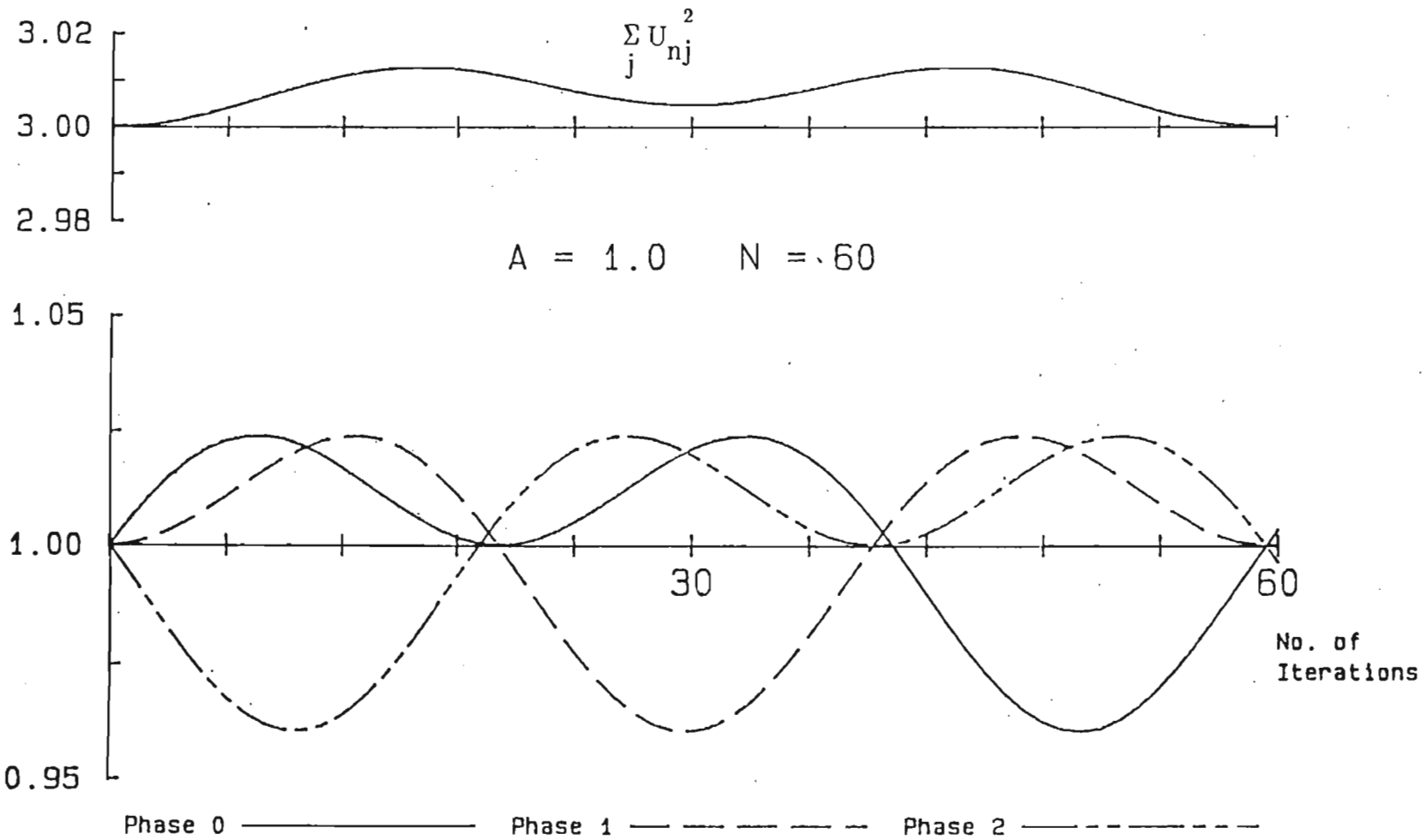


Fig. 3.4 Graph Showing the Changes in Lengths of the Vectors

3.6 Liapunov Stability Criteria

In section 2.2 it was shown that the oscillator matrices discussed in this thesis were equivalent to the [G] matrix in the standard State Space model of a control system. This equivalence enables the stability criteria developed by Liapunov [18] to be applied to oscillator matrices.

In most control systems it desirable for the unwanted variations or disturbances to decay as quickly as possible. This condition is characterised by a system which has eigenvalues which are less than unity. For oscillators, the output must enter a limit cycle and neither decay nor grow to infinity, and this condition is identified by the existence of one or more "critical eigenvalues" which have a magnitude of unity (or poles which lie on the unit circle). Systems with one or more critical eigenvalues are defined by Liapunov as being weakly or critically stable. Critically stable systems may become unstable because of external disturbances such as round–off or truncation errors. All the oscillator matrices discussed in this thesis have at least two critical eigenvalues.

Liapunov and Hahn [18] have analysed the stability of systems with one or two critical eigenvalues, and have shown that by introducing a suitable transformation which removes the critical eigenvalues, the stability of a critical system can be inferred from the resultant transformed or residual set of equations. The [O], [T] and [H] matrices satisfy the criteria (two complex critical eigenvalues) and for each matrix the residual equation is a constant which is less than or equal to unity. This implies a limit cycle and a bounded output for a bounded input. Thus arithmetic errors with zero mean value will not cause the above matrices to become unstable.

For $\delta < \sqrt{3}$ or $k < 1$ it can be proven that the [I] matrix has three critical eigenvalues namely

$$\mu_1 = 1 \tag{3.19.1}$$

$$\mu_{2,3} = 1-3k^2/2-k^3/2 \pm j.k/2.\sqrt{12+4k-9k^2-6k^3-k^4} \tag{3.19.2}$$

and furthermore that for $k < 1$

$$|\mu_{2,3}| = 1$$

Since the [I] matrix has three critical values (one real and two complex) the stability analysis applied to the [O], [T] and [H] matrices (which have only two critical eigenvalues) cannot be used. In order to assess the stability of the [I] matrix a set of transformation equations for a third order critical system have to be used. Hahn[18] has proved that such a set of transformation equations can be derived, however owing to the complexity of the task no such set of transformations has been derived to date. Therefore the proof of the stability of the [I] matrix is not possible, and the derivation of the of the transformation equations is beyond the scope of this thesis. However from observations of the similarity in behaviour of the [H] and [I] matrices it is anticipated that the residual equation for the [I] matrix is also a constant, and hence it is expected that arithmetic errors with zero mean value will not cause the [I] matrix to become unstable.

In section 3.8 it is proven that the addition of a constant value (D) to the pulse width does not affect the two stability criteria derived in the previous section. It has also been demonstrated in section 3.8 that for the [I] matrix, even though the value of D is changed at every step, stable oscillation is maintained. Further evidence of the stability of the [I] matrix is shown in section 4.3.2 where stable oscillation is maintained in the presence of truncation errors.

The above observations led the author to the conclusion that the [I] matrix is a stable oscillator matrix even though no formal proof is available.

3.7 Altering the Amplitudes of the Phases

It is necessary for a PWM modulator to be able to alter the lengths of all the vectors and hence the voltage amplitude of the inverter output, so that a variable voltage amplitude to frequency ratio can be realised. Thus when the frequency or k is changed, it must also be possible to change the length U_0 of each vector.

This is achieved by introducing a gain term "g" in eqs. (3.14), i.e.

$$g.x_1(n+1) = g.x_1(n) + k[x_2(n) - x_3(n)] \tag{3.20.1}$$

$$g.x_2(n+1) = g.x_2(n) + k[x_3(n) - g.x_1(n+1)] \tag{3.20.2}$$

$$g.x_3(n+1) = g.x_3(n) + k[g.x_1(n+1) - g.x_2(n+1)] \tag{3.20.3}$$

In Appendix B the [G] matrix is derived from eqs. (3.20) and it is also proven that

Det [G] $= g^3$ where

$$[G] = \begin{bmatrix} g & -gk & gk+gk^2 \\ k & g-k^2 & -gk+k^2+k^3 \\ -k & k+k^2 & g-2k^2-k^3 \end{bmatrix} \qquad (B.7)$$

The significance of this result is that the lengths of the vectors can be changed without affecting the frequency, since the determinant of the matrix is independent of k.

The main application of this proof is in the analysis of errors, where amplitude errors can be analysed independently from frequency errors. Figs. 3.5 and 3.6 are functions generated by the [G] matrix and show that the same phase relationship is maintained, when U_0 is changed smoothly or abruptly. The number of steps for one cycle is also less than M.

3.8 Five Phase Oscillator Matrix

The procedure for the development of a 5–phase oscillator matrix is similar to that of the 3–phase oscillator matrix except that $N = 5$ and $\theta = 2\pi/5$ radians.

i.e. $\tan(\theta/2) = \tan(\pi/5)$

Thus from eq. (3.5) the fundamental equation for a 5–phase oscillator matrix is

$$\mathrm{Sin}(\phi+\delta) \cong \mathrm{Sin}\phi+\frac{\delta}{\tan(\pi/5)}.[\mathrm{Sin}(\phi+\theta)-\mathrm{Sin}(\phi+2\theta)+\mathrm{Sin}(\phi+3\theta) - \mathrm{Sin}(\phi+4\theta)] \qquad (3.21)$$

where the approximations $\mathrm{Cos}\delta \approx 1$ and $\mathrm{Sin}\delta \approx \delta$ have been used.

Let $\ell = \delta/(\tan(\pi/5))$ and as for the [I] matrix substituting $[x_j(n)] = [U.\mathrm{Sin}_j(\phi+n\delta)]$ for the pulse widths, the five equations for the five phases are

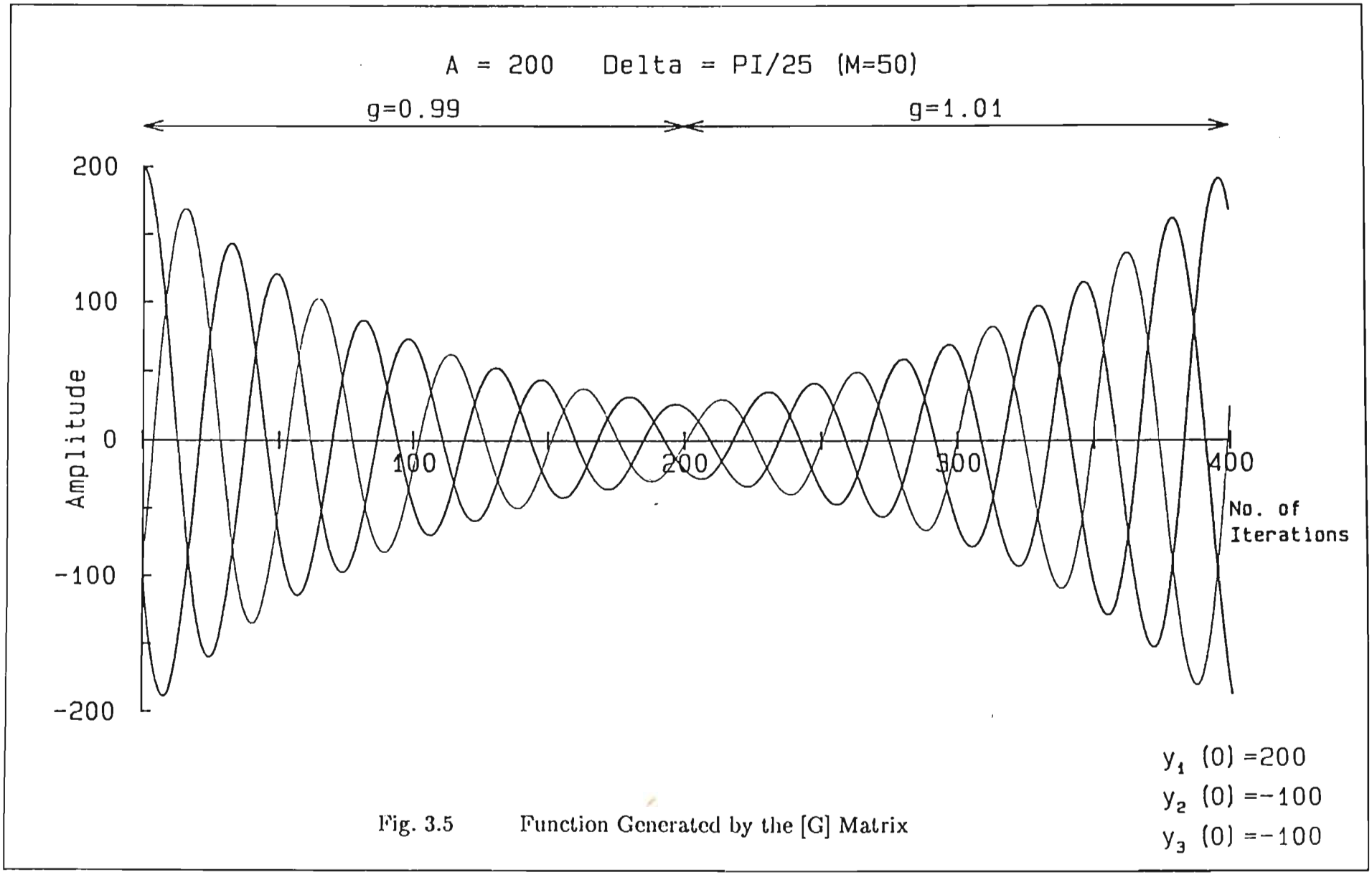


Fig. 3.5 Function Generated by the [G] Matrix

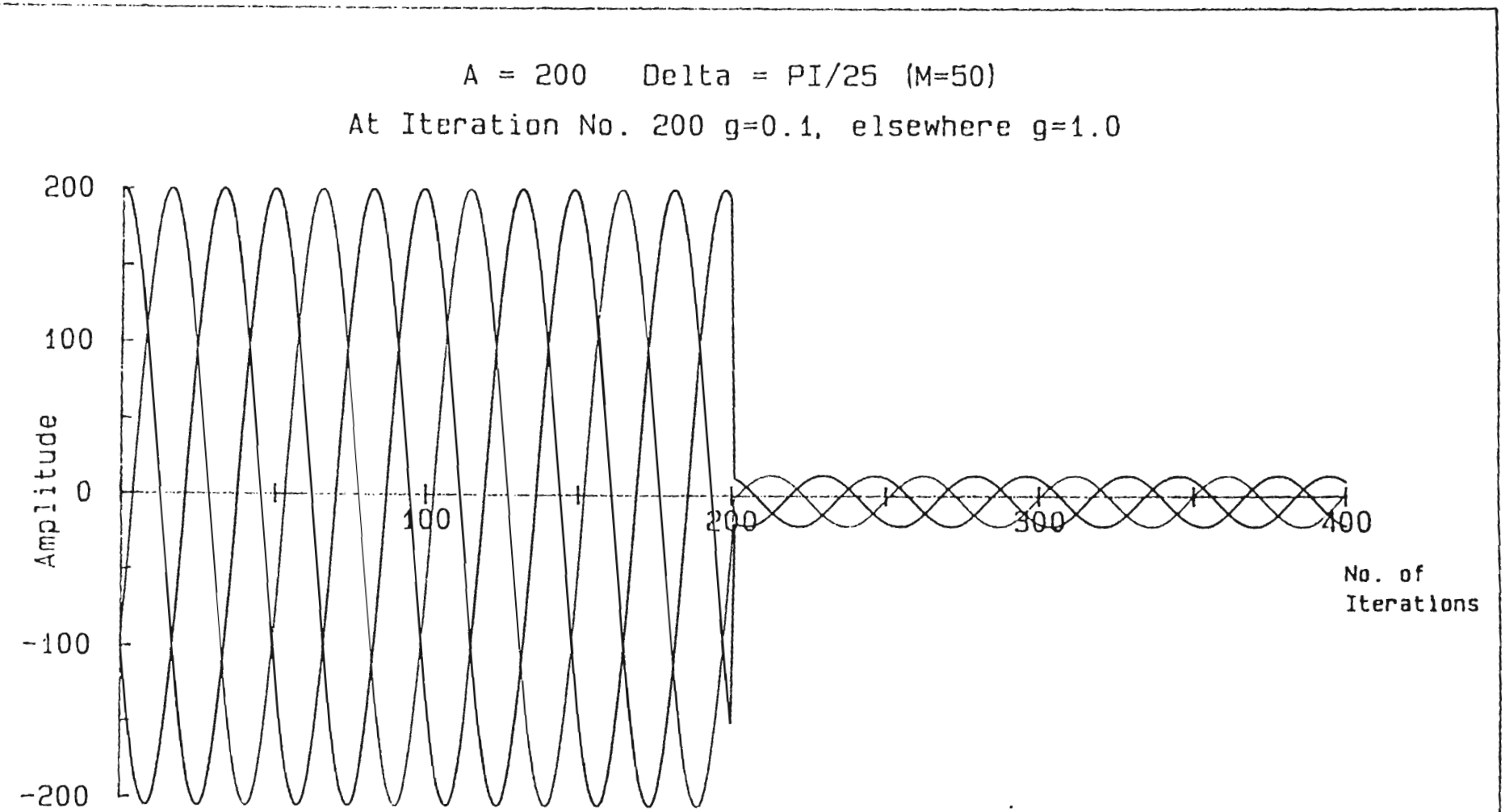


Fig. 3.6 Function Generated by the [G] Matrix

$$x_1(n+1) = x_1(n) + \ell.[x_2(n)-x_3(n)+x_4(n)-x_5(n)] \quad (3.22.1)$$

$$x_2(n+1) = x_2(n) + \ell.[x_3(n)-x_4(n)+x_5(n)-x_1(n+1)] \quad (3.22.2)$$

$$x_3(n+1) = x_3(n) + \ell.[x_4(n)-x_5(n)+x_1(n+1)-x_2(n+1)] \quad (3.22.3)$$

$$x_4(n+1) = x_4(n) + \ell.[x_5(n)-x_1(n+1)+x_2(n+1)-x_3(n+1)] \quad (3.22.4)$$

$$x_5(n+1) = x_5(n) + \ell.[x_1(n+1)-x_2(n+1)+x_3(n+1)-x\,(n+1)] \quad (3.22.5)$$

or $\quad [x_j(n+1)] = [x_j(n)] * [F]$

where the 5 * 5 oscillator matrix [F] is equal to

$$\begin{bmatrix} 1 & -\ell & \ell+\ell^2 & -\ell-2\ell^2-\ell^3 & \ell+3\ell^2+3\ell^3+\ell^4 \\ \ell & 1-\ell^2 & -\ell+\ell^2+\ell^3 & \ell-2\ell^3-\ell^4 & -\ell-\ell^2+2\ell^3+3\ell^4+\ell^5 \\ -\ell & \ell+\ell^2 & 1-2\ell^2-\ell^3 & -\ell+2\ell^2+3\ell^3+\ell^4 & \ell-\ell^2-5\ell^3-4\ell^4-\ell^5 \\ \ell & -\ell-\ell^2 & \ell+2\ell^2+\ell^3 & 1-3\ell^2-3\ell^3-\ell^4 & -\ell+3\ell^2+6\ell^3+6\ell^4+\ell^5 \\ -\ell & \ell+\ell^2 & -\ell-2\ell^2-\ell^3 & \ell+3\ell^2+3\ell^3+\ell^4 & 1-4\ell^2-6\ell^3-6\ell^4-\ell^5 \end{bmatrix}$$

The determinant of [F] can be evaluated by using the matrix reduction theorem; "where if [D] is the matrix obtained from [A] by adding ℓ times column s to column r, then Det[D] = Det[A] " (Marcus[29] p 126)

Now $F'_{i2} = F_{i2} + \ell.F_{i1}$

$F'_{i3} = F_{i3} + \ell.F_{i2} - \ell.F_{i1}$ etc.

Thus [F] is reduced to

$$[F] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \ell & 1 & 0 & 0 & 0 \\ -\ell & \ell & 1 & 0 & 0 \\ \ell & -\ell & \ell & 1 & 0 \\ -\ell & \ell & -\ell & \ell & 1 \end{bmatrix}$$

and hence Det[F] = 1 for all ℓ.

The sum of each column of [F] is also unity

i.e. $\sum_i F_{ij} = 1$ for $j = 1, 2, 3, 4, 5$

Thus both conditions for stable oscillation are satisfied.

The 5–phase digital oscillator equations were programmed into a HP 86 computer and stable oscillator conditions were observed over 10^6 steps. The magnitudes of the eigenvalues of the [F] matrix have been calculated using the commercially available MATLAB computer program, and for $\ell < 1$ all the eigenvalues have magnitudes that are equal to one. For $\ell > 1$ there is at least one eigenvalue which has a magnitude greater than unity, and hence the [F] matrix is unstable for these values.

3.9 Effect of a Constant Offset

It is possible to construct a digital oscillator when only an unsigned multiplication operation is available, by adding a constant value or offset (D) to each pulse width.

Let each of the vectors rotate about a point D, i.e. each pulse width is represented by $y_j(n) = D + x_j(n)$. Substituting this into eqs. (3.14) yields

$$D+x_1(n+1) = D+x_1(n) + k[(D+x_2(n)) - (D+x_3(n))] \quad (3.23.1)$$

$$D+x_2(n+1) = D+x_2(n) + k[(D+x_3(n)) - (D+x_1(n+1))] \quad (3.23.2)$$

$$D+x_3(n+1) = D+x_3(n) + k[(D+x_1(n+1)) - (D+x_2(n+1))] \quad (3.23.3)$$

hence

$$D + x_1(n+1) = D + x_1(n) + k[x_2(n) - x_3(n)] \quad (3.24.1)$$

$$D + x_2(n+1) = D + x_2(n) + k[x_3(n) - x_1(n+1)] \quad (3.24.2)$$

$$D + x_3(n+1) = D + x_3(n) + k[x_1(n+1) - x_2(n+1)] \quad (3.24.3)$$

Since the relationship in eqs. (3.23) still holds, and inside the square brackets the D terms cancel, a constant offset has no effect on the stability of an oscillator or the frequency. Since no restriction on the value of D was made in the above proof, (i.e. D is not dependent on any parameters of the previous step) the value of D can change at each step and stable oscillation is assured.

It should be noted that the use of the row vector $[D + x_j(n)]$ with the [G] matrix produces indeterminate results, since the offset D terms do affect the frequency, as shown in eq. (3.25.2).

i.e.

$$g(D+x_1(n+1)) = g(D+x_1(n)) + k[D+x_2(n) - (D+x_3(n))] \tag{3.25.1}$$

$$g(D+x_2(n+1)) \neq g(D+x_2(n)) + k[D+x_3(n) - g.(D+x_1(n+1))] \tag{3.25.2}$$

$$g(D+x_3(n+1)) = g(D+x_3(n)) + k[g(D+x_1(n+1)) - g(D+x_2(n+1))] \tag{3.25.3}$$

3.10 Computational Requirements

Eqs. (3.14) require three subtractions, three multiplications and three additions to compute all the new pulse widths of the phases for a 3–phase digital oscillator matrix. When δ is changed, there is no need to calculate the $\alpha_{ij}(\delta)$ polynomials given in eqs. (3.16) providing a sequential calculation procedure is used. Thus a simple, and very fast program can be written.

If the gain control function is required an additional three multiplications are needed at each step.

A linear gain control function can be obtained directly from eqs. (3.14) by noting that the second term of eq. (3.5) is equivalent to $\delta.\mathrm{Cos}(\phi+n\delta)$.

i.e. $k.(x_1(n) - x_2(n)) \propto \delta.\mathrm{Cos}(\phi+n\delta)$ (3.26)

Since δ and k are inversely proportional to frequency, and are proportional to amplitude, a linear gain (V/f) characteristic is available directly from the sequence of calculations for $x_j(n)$.

3.11 Conclusions

The basic equation for a N–odd digital oscillator was developed using both the sine and cosine difference equations. Using two simplifying approximations and a sequential calculation procedure, the stable 3–phase [I] oscillator matrix was derived. The three conditions for stable oscillation were also deduced and the [I] matrix was shown to satisfy two of these criteria. It is expected that the [I] matrix will also satisfy the Liapunov stability criteria since its observed behaviour is similar to that of the [O], [T] and [H] matrices.

The same procedure can be used to develope stable N–odd phase oscillator matrices, and a stable 5–phase oscillator matrix was developed.

The magnitudes of the pulse widths need not be restricted to values of $U.\mathrm{Cos}\phi$ or $U.\mathrm{Sin}\phi$, and values of $D + U.\mathrm{Sin}\phi$ or $D + U.\mathrm{Cos}\phi$ can also be used. This is useful where a particular microprocessor or logic circuit does not have a signed–multiply operation.

The lengths of the rotating vectors and hence the magnitudes of the pulse widths can be controlled by using a gain factor g, which does not affect the number of steps required to complete one cycle. The g term (and any errors) affects only the magnitudes of the pulse widths, while the k term (and any errors) affects only the rate of rotation of the vectors and hence the inverter frequency. The [G] matrix represents a digital oscillator matrix in which the amplitude and frequency can be independently controlled.

All the equations developed in this chapter have been implemented on a HP 86 computer, and only the [H], [I], [G] and [F] matrices produced stable oscillations. The relevant program listings showing the ease of implementation of the [T] and [I] matrices are given in Appendix C.

The equations derived in this chapter have implicitly assumed infinite precision, and the fact that δ was small. The effects of these approximations and the influence of arithmetic errors are analysed in the next chapter.

CHAPTER FOUR

ERROR ANALYSIS

4.1 Introduction

In the preceding chapters the approximations $\mathrm{Cos}\delta = 1$ and $\mathrm{Sin}\delta = \delta$ were used in deriving the fundamental oscillator equations for the 2–phase and N–odd phase oscillator matrices. It was also shown that stable oscillation occurs for values of $\delta < \sqrt{3}$, regardless of the validity of the two trigonometric approximations. While stable oscillation is assured, there is an error introduced in computing the pulse widths, and in the number of steps required to complete one cycle.

In order to analyse the errors introduced by the approximations, it is necessary to consider the rotation of one or N vectors in the $x - y$ plane. The pulse widths are the projections of the vectors onto the x or y axes, as shown in Fig. 4.1 (a). The N vectors are spaced $2\pi/N$ radians apart and have an initial value U_0. At each step the vectors are rotated by approximately δ radians and change in length as shown in Fig. 4.1 (b). Fig 4.1(b) also shows how the pulse width errors are calculated from the vector errors. Fig 4.1 (c) shows the relative changes in lengths for the three vectors described by eqs. (3.14) with a large value of δ. It is important to note that each vector rotates through a different angle.

The projection of one vector onto the x and y axes gives the required magnitudes of the two pulse widths $U_n.\mathrm{Cos}(\phi+n\delta)$ and $U_n.\mathrm{Sin}(\phi+\delta)$ for the 2–phase digital oscillator matrix as shown in Fig. 4.1 (a).

4.2 Two Phase Oscillator Errors

4.2.1 Phase and frequency errors

Assume that the vector $\bar{U}$ has an initial (arbitrary) angle ϕ to the x axis. Also let the true angle of rotation after n steps be α_n radians and let λ_n be the difference between the expected angle of rotation $\phi+n\delta$ and α_n i.e.

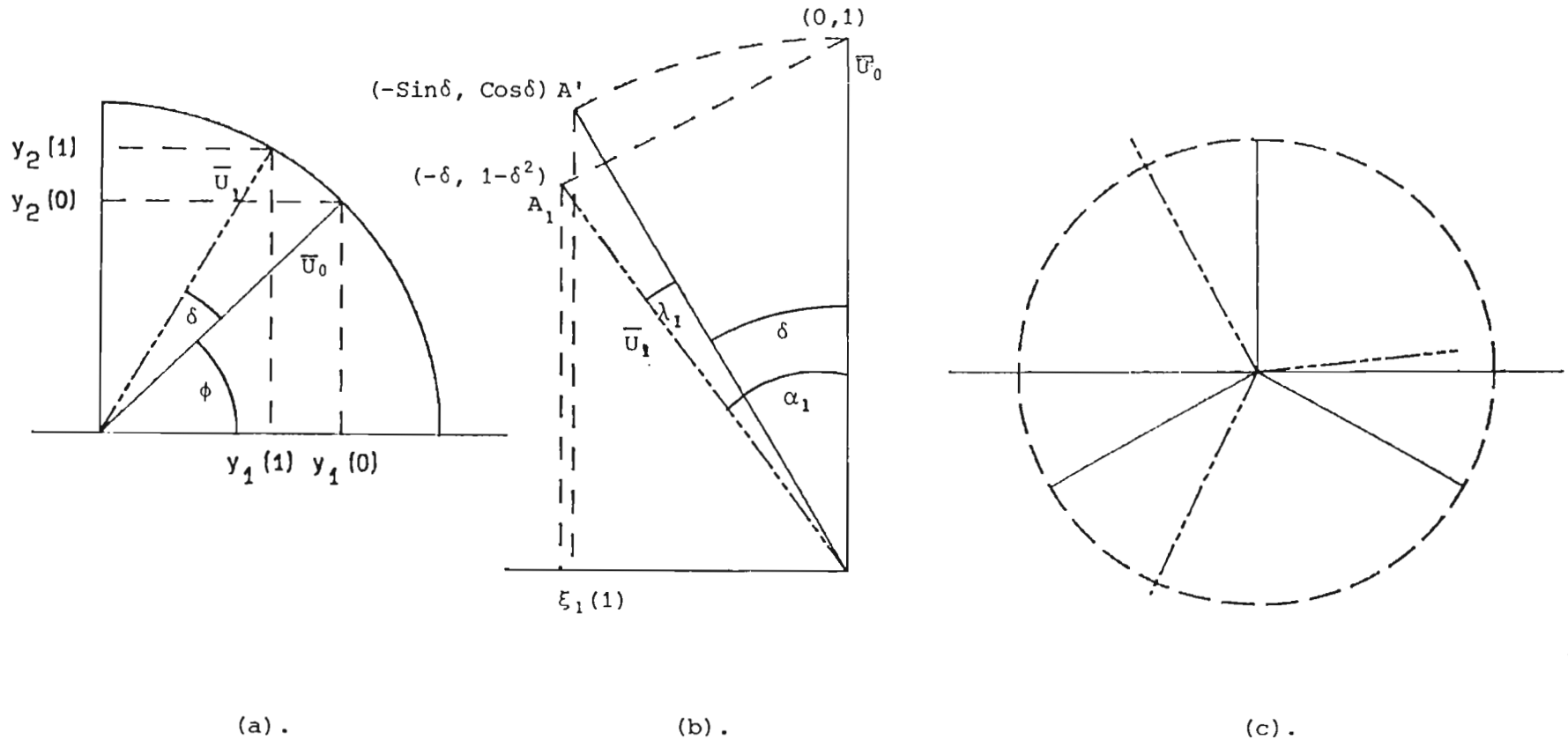


Fig. 4.1 Error Analysis of Simplex Vectors

$$\lambda_n = \alpha_n - (\phi + n\delta) \tag{4.1}$$

where α_1 and λ_1 are shown in Fig. (4.1(b).

Ideally vector $\bar{U}$ should rotate by δ radians to point A' and not change in length as shown by point A_1 in Fig 4.1 (b). The trigonometric approximations cause the vector to change length and to rotate by approximately δ radians. In Fig 4.1 (b) the co–ordinates of U_0 are (0,1) and after the first step the co–ordinates of U_1 are $(-\delta, 1-\delta^2)$ instead of $(-\text{Sin}\delta, \text{Cos}\delta)$.

From Fig. 4.1(b) and by inspection

$$\alpha_1 = \text{atan}(\delta/(1-\delta^2)) \tag{4.2.1}$$

$$\text{and} \qquad \lambda_1 = \text{atan}(\delta/(1-\delta^2)) - \delta \text{ radians} \tag{4.2.2}$$

A more general formula for α_1 and λ_1 was derived from eq. (2.14) and using the initial conditions $U_0.\text{Sin}\phi$ and $U_0.\text{Cos}\phi$ for $x_1(0)$ and $x_2(0)$ respectively.

$$x_1(1) = U_0.\text{Sin}\phi + \delta.U_0.\text{Cos}\phi \tag{4.3.1}$$

$$x_2(1) = U_0.\text{Cos}\phi(1-\delta^2) - \delta.U_0.\text{Sin}\phi \tag{4.3.2}$$

$$\text{Noting that} \qquad \alpha_1 = \text{atan}(x_1(1)/x_2(1)) \tag{4.4.1}$$

and substituting eqs. (4.3) into eq. (4.4.1) yields

$$\alpha_1 = \text{atan}\left[\frac{\tan\phi + \delta}{(1-\delta^2) - \delta.\tan\phi}\right] \tag{4.4.2}$$

For the special case where $\phi = \pi/2$ radians

$$\alpha_1 = \text{atan}(\delta)$$

$$\text{and} \qquad \lambda_1 = \text{atan}(\delta) - \delta \tag{4.4.3}$$

By using a similar procedure it can be shown that

$$\alpha_2 = \text{atan}\left[\frac{\tan\phi(1-\delta^2) + 2\delta - \delta^3}{\tan\phi(-2\delta + \delta^3) + 1 - 3\delta^2 + \delta^4}\right] \tag{4.5.1}$$

with the special case for $\phi = \pi/2$, where

$$\alpha_2 = \text{atan}\left[\frac{1 - \delta^2}{-2\delta + \delta^3}\right] \tag{4.5.2}$$

By comparing the elements of $[T]^n$ with eq. (4.4.2) and eq. (4.5.2), a general formula for δ can be deduced.

Let $[T]^n = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$

$$\text{then} \quad \alpha_n = \text{atan}\left[\frac{t_{11}.\tan\phi + t_{12}}{t_{21}.\tan\phi + t_{22}}\right] \tag{4.6.1}$$

with the special case for $\phi = \pi/2$

$$\alpha_n = \text{atan}\left[\frac{t_{11}}{t_{21}}\right] \tag{4.6.2}$$

Substituting eq. (4.6.1) into eq. (4.1) yields

$$\lambda_n = \text{atan}\left[\frac{t_{11}.\tan\phi + t_{12}}{t_{21}.\tan\phi + t_{22}}\right] - (\phi + n\delta)\text{modulo } 2\pi \tag{4.7}$$

The difference between the predicted value given by eq. (4.6.1) and the true value of α_n was less than 10^{-8} (or the limit of precision of the computer that was used to test the accuracy of eq. (4.7)).

A graph of eq. (4.7) is shown by the dashed line in Fig. 4.2, as well as a graph of eq. (4.11.2) which shows the amplitude or pulse width error for the [T] matrix. The maximum values of λ_n occur at values α_n near $(M+1/2)\pi$. There is no

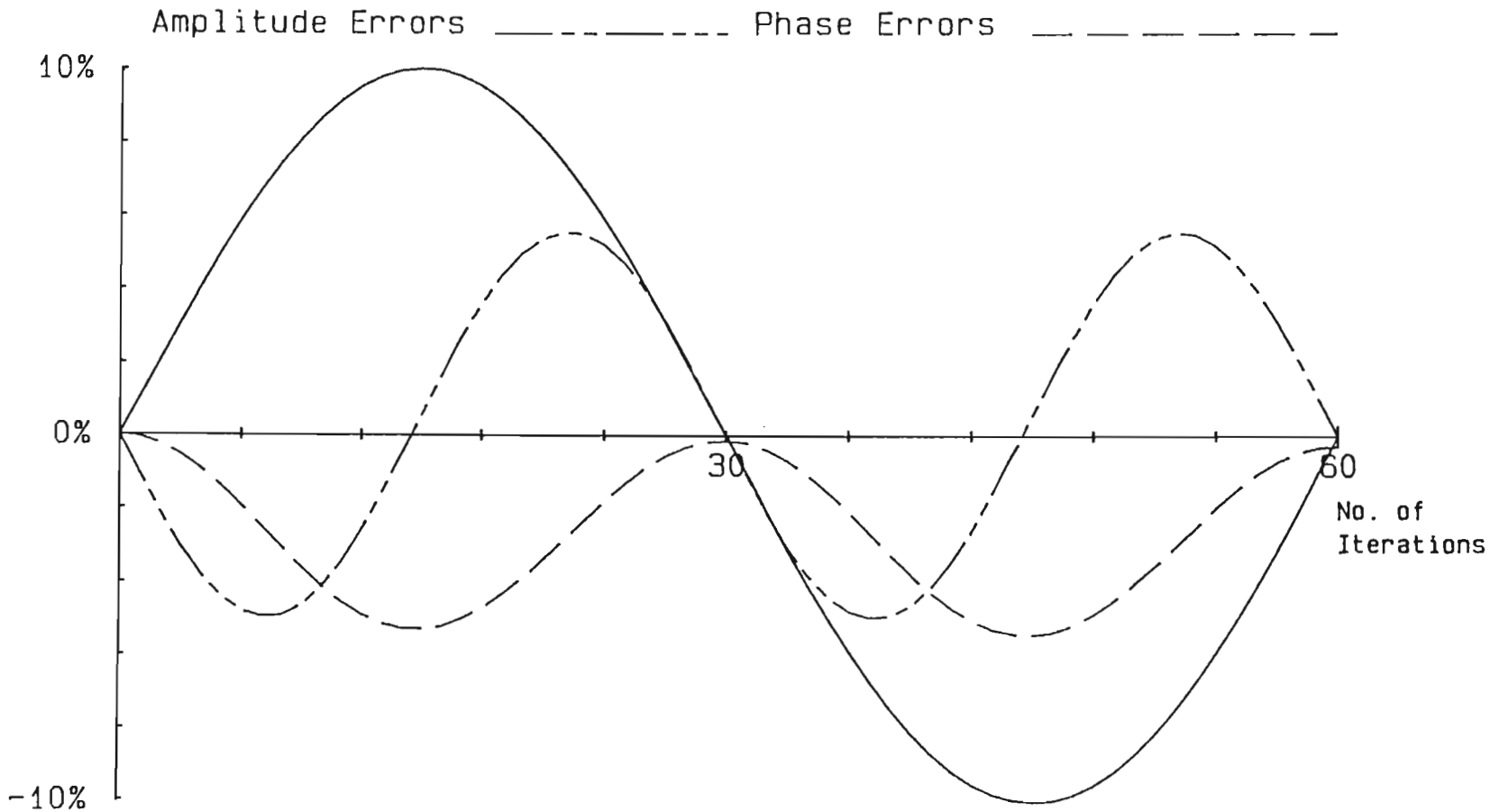


Fig. 4.2 Amplitude and Phase Errors for the [T] Matrix

direct method to determine the maximum value of λ_n except by evaluation of eq. (4.7) for a number of different values of δ. The maximum values of λ_n that have been observed for different gear ratios M are given in Table 4.1.

In Fig. 4.2 the λ_n curve does not quite return to the x axis after M/2 steps. This is due to the fact that less than $2\pi/\delta$ steps are required to complete one cycle. When the true number of steps required to complete one cycle (M') was used, the phase–error curve (λ_n) just touched the x axis at each half cycle.

In view of the complexity of eq. (4.7), a simpler but less accurate formula for the true number of steps for each cycle M' was investigated. The value of M' for a number of different values of δ' was calculated using two methods

a) The crossing point for each half cycle was estimated by linearly iterpolating between two steps on either side of the real axis. The difference between two successive crossing points was assumed to be M'/2.

b) By determining the number of steps required to complete a large number of cycles.

Both methods produced almost identical values which are summarised in Table 4.1. An approximate formula for M' was determined by a Chebyshev curve fitting program.

$$M' = \frac{2\pi}{\delta} + \frac{\delta}{4} \tag{4.8}$$

Table 4.1 shows an estimate of the accuracy of eq. (4.8) for a number of gear ratios M. In general eq. (4.8) overestimates the value of M'. For $M > 20$ the error is less than 0.025%.

4.2.2 Pulse width errors

Pulse width errors are calculated from the length of the vector U_n and the phase error α_n as shown in Fig 4.1(b). By definition

$$U_1^2 = U_0^2.\mathrm{Cos}^2(\phi+\delta) + U_0^2.\mathrm{Sin}^2(\phi+\delta) = x_1^2(n) + x_2^2(n) \tag{4.9.1}$$

TABLE 4.1

Summary of Error Analysis Data for the [T] Matrix

M	δ	Eq. (4.8)	Approx–imation error for Eq. (4.8)	True No. of steps for one cycle M'	Maximum observed phase error (radians)	Maximum observed pulse width error
10	0.6283	9.843	(0.127%)	9.8305	0.342	33.1%
20	0.3142	19.922	(0.025%)	19.917	0.162	15.9%
30	0.2094	29.948	(0.010%)	29.945	0.107	10.5%
40	0.1571	39.961	(0.005%)	39.959	0.080	7.9%
50	0.1257	49.969	(0.004%)	49.967	0.064	6.3%
60	0.1045	59.974	(0.002%)	59.973	0.053	5.25%
90	0.0698	89.983	(0.001%)	89.982	0.035	3.50%
120	0.05236	119.987	(0.0008%)	119.986	0.026	2.62%
240	0.02618	239.993	(0.0002%)	239.993	0.013	1.32%

also

$$x_1(n) = U_0.\mathrm{Sin}\phi + \delta.U_0.\mathrm{Cos}\phi \tag{4.9.2}$$

$$x_2(n) = -\delta.U_0.\mathrm{Sin}\phi + (1-\delta^2).U_0.\mathrm{Cos}\phi \tag{4.9.3}$$

Substituting eq. (4.9.2) and eq. (4.9.3) into eq. (4.9.1) and collecting like terms yields

$$U_1^2 = U_0^2.(1-\delta^2.\mathrm{Cos}2\phi + \delta^3.\mathrm{Sin}2\phi + \delta^4.\mathrm{Cos}\,\phi\,) \tag{4.10}$$

Eq. (4.10) gives the change in length of U_0 after one step for any arbitrary phase angle. There is no restriction on the value of U_0 and hence eq. (4.10) may be used to calculate U_n by recursive multiplication. i.e.

$$U_n^2 = U_{n-1}^2.(1-\delta^2.\mathrm{Cos}2\alpha_{n-1}+\delta^3.\mathrm{Sin}2\alpha_{n-1}+\delta^4.\mathrm{Cos}^2\alpha_{n-1}\,) \tag{4.11.1}$$

or alternatively

$$U_n^2 = U_0^2.\prod_{i=0}^{n}(1-\delta^2.\mathrm{Cos}2\alpha_i+\delta^3.\mathrm{Sin}2\alpha_i+\delta^4.\mathrm{Cos}^2\alpha_i\,) \tag{4.11.2}$$

where $\alpha_0 = \phi$

The validity of eqs. (4.11) and eq. (4.6.1) were checked by comparing the true value of U_n with the value predicted by eq. (4.11.2). For various values of δ the maximum difference was less than 10^{-8} over 4000 steps. (The precision of the computer was 10^{-10} and the difference of 10^{-8} was probably due to round–off error).

Eq. (4.11.2) can be simplified by substituting

$$1/2.\mathrm{Cos}2\alpha_n + 1/2 = \mathrm{Cos}^2\alpha_n$$

and collecting like terms yielding

$$U_n^2/U_{n-1}^2 = \left[(1+\delta^4/2)-(\delta^2-\delta^4/2).\mathrm{Cos}2\alpha_{n-1}+\delta^3.\mathrm{Sin}2\alpha_{n-1}\right] \tag{4.12}$$

Eq. (4.12) implies that U_n tends to infinity, since the average value of eq. (4.12) is greater than unity. However if the effects of the phase errors are taken into account then U_n does not tend to infinity. Due to the phase error λ_n, the average value of U_n^2/U_{n-1}^2 is unity, and U_n oscillates about a mean value of U_0. Further investigations were carried out but no meaningful conclusions or results were obtained by algebraic manipulation or computer analysis. In order to derive estimates of the upper bounds for the values of U_n a different analytical approach was used.

4.2.3 Estimate of the upper bound for vector length errors.

An upper bound for the maximum vector length error was derived from eqs. (2.14) as follows

$$x_1(n+1) = x_1(n) + \delta.x_2(n) \qquad (2.14.1)$$

$$x_2(n+1) = -\delta.x_1(n) + (1-\delta^2)x_2(n) \qquad (2.14.2)$$

Assuming no errors in $x_1(n)$ or $x_2(n)$ then the true values of $x'_1(n+1)$ and $x'_2(n+1)$ are

$$x'_1(n+1) = x_1(n).\mathrm{Cos}\delta + x_2(n).\mathrm{Sin}\delta \qquad (4.13.1)$$

$$x'_2(n+1) = -x_1(n).\mathrm{Sin}\delta + x_2(n).\mathrm{Cos}\delta \qquad (4.13.2)$$

Hence the estimated errors are given by

$$\begin{aligned}\epsilon_1(n) &= x_1(n+1) - x'_1(n+1) \\ &= x_1(n).(1-\mathrm{Cos}\delta) + x_2(n).(\delta-\mathrm{Sin}\delta)\end{aligned} \qquad (4.14.1)$$

$$\begin{aligned}\epsilon_2(n) &= x_2(n+1) - x'_2(n+1) \\ &= -x_1(n).(\delta-\mathrm{Sin}\delta) + x_2(n).(1-\delta^2-\mathrm{Cos}\delta)\end{aligned} \qquad (4.14.2)$$

Substituting the Maclaurin's expansion for $\mathrm{Sin}\delta$ and $\mathrm{Cos}\delta$ and collecting like terms yields

$$\epsilon_1(n) = x_1(n).\left[\frac{\delta^2}{2} - \frac{\delta^4}{4!}\right] - x_2(n).\left[\frac{\delta^3}{3!} - \frac{\delta^5}{5!}\right] \qquad (4.15.1)$$

$$\epsilon_2(n) = -x_1(n).\left[\frac{\delta^3}{3!} - \frac{\delta^5}{5!}\right] - x_2(n).\left[\frac{\delta^2}{2} - \frac{\delta^4}{4!}\right] \qquad (4.15.2)$$

An analysis of the properties of the [G] matrix shows that the elements on the main diagonal control the changes in lengths of the vectors. Thus the maximum change in length of U_n in one step is

$$\frac{\delta^2}{2}.U_n.\mathrm{Sin}(\phi+n\delta) \quad \text{since} \quad x_1(n) = U_n.\mathrm{Sin}(\phi+n\delta)$$

For $\phi = 0$ the maximum error ϵ_{max} is obtained by summing $\frac{\delta^2}{2}.U_n.\mathrm{Sin}(\phi+n\delta)$ over $M/2$ steps (which is equivalent integrating over the range 0 to π.)

$$\therefore \qquad \epsilon_{max} = \sum_{n=0}^{M/2} \frac{\delta^2}{2}.U_n.\mathrm{Sin}(\phi+n\delta) \qquad (4.16.1)$$

It is shown in Appendix D that eq. (4.16.1) is equal to

$$\epsilon_{max} = \sum_{n=0}^{M/2} \frac{\delta^2}{2}.U_n.\mathrm{Sin}(\phi+n\delta) = \frac{\delta^2.U_n}{2.\tan(\delta/2)} \qquad (4.16.2)$$

since $\delta/2 << 1$ $\tan(\delta/2) \cong \delta/2$ hence

$$\epsilon_{max} \cong \delta.U_n \qquad (4.17)$$

In practice the value of $\epsilon_{max} = \delta.U_n$ will never be reached because the maximum change of U_n at each step is less than $\delta^2/2$ owing to the influence of the negative phase error λ_n. (For larger negative λ_n the smaller the overall error). Furthermore all the worst case conditions are unlikely to simultaneously

The maximum error for a number of values of δ has been observed to be approximately $\delta/2$ as shown in Table 4.1. Fig. 4.3 shows a typical graph of the pulse width error produced by the [T] matrix. The irregular shape of the error curve was due to truncation errors. It should also be noted that the peak error (1.2%) was less than $\delta/2$ (12%). It has been observed for other values of δ that truncation errors reduce the magnitude of the maximum error, though the improvement was not always as large as that implied in Fig. 4.3.

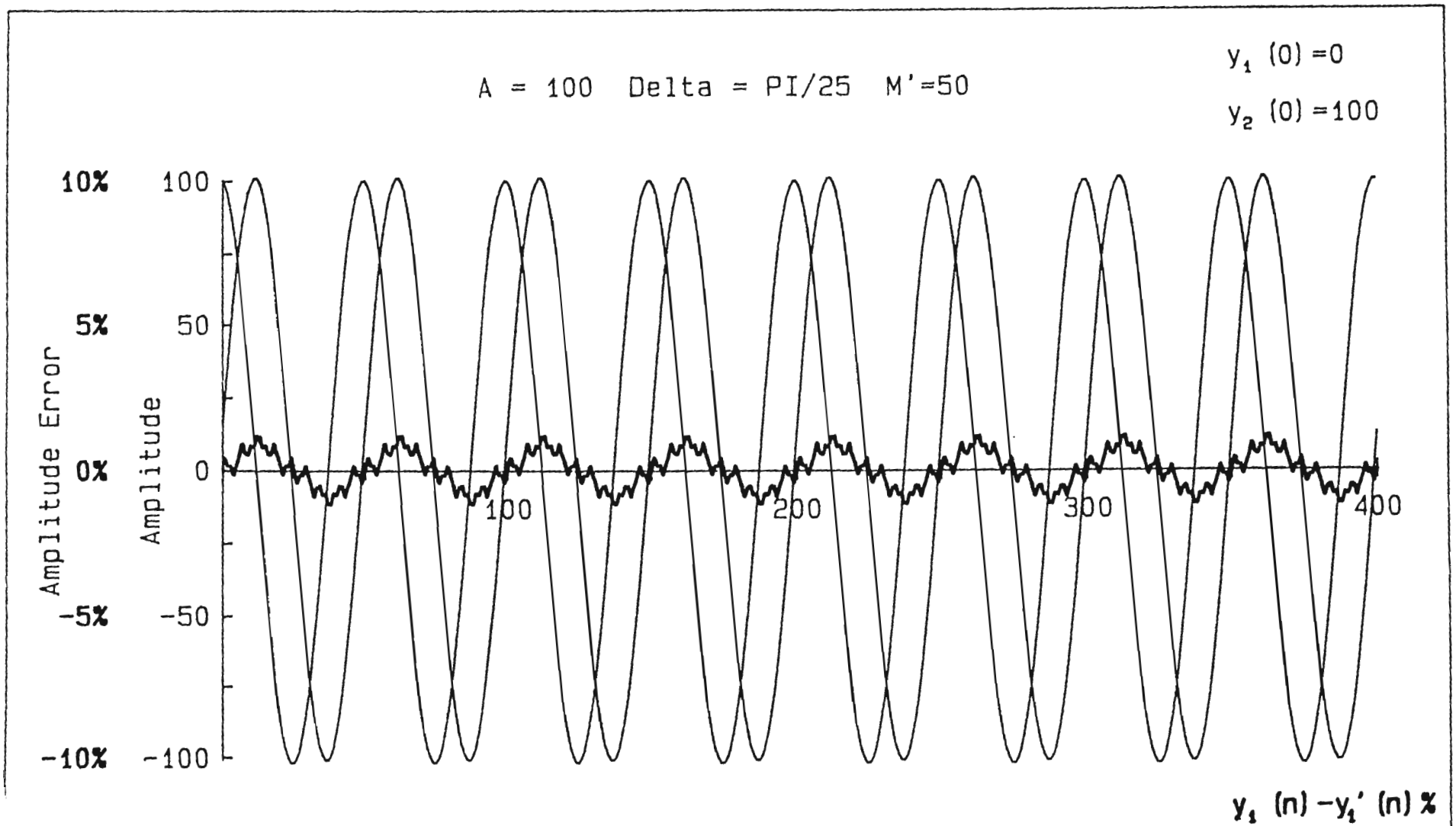


Fig. 4.3 Graph of Pulse Width Error of The [T] Matrix

4.3 Three Phase Digital Oscillator Errors

4.3.1 Introduction

There are no simple formulae for describing the changes in lengths, and the phase angle errors for the three vectors shown in Fig. 4.1 (c). The difficulty with all analyses is that the three vectors have different changes in length and rotate by differing amounts. For example it can be shown that for the first step the changes in length of the three vectors respectively are given by

$$U_{11}.\mathrm{Sin}(\phi+\delta) = U_0.\mathrm{Sin}\phi.\sqrt{1+3k^2}$$

$$U_{12}.\mathrm{Sin}(\phi+\theta+\delta) = U_0.\mathrm{Sin}(\phi+\theta).\sqrt{1+3k^3}$$

$$U_{13}.\mathrm{Sin}(\phi+2\theta+\delta) = U_0.\mathrm{Sin}\,(\phi+2\theta).\sqrt{1-3k^2-3k^3+6k^4+9k^5+2k^6}$$

More useful information on the errors and the behaviour of the 3–phase digital oscillator equation was obtained by experimenting with various values of δ and observing the results.

4.3.2 Cumulative Errors

The wide spread use and simplicity of two's–complement arithmetic makes it likely to be used in many practical applications of the [I] matrix, and therefore the effects of truncation errors have to be analysed. An undesirable property of two's complement arithmetic is that truncation errors always reduce the value of number, i.e. 2.5 is truncated to 2 and –2.5 is truncated to –3. Therefore the average truncation error is 1/2 LSB (Least Significant Bit).

Let the mean truncation error for phase 1 be

$$\overline{\epsilon_{1T}} = 1/2 \text{ LSB}$$

Similarly let the mean truncation error for phases 2 and 3 be

$$\overline{\epsilon_{2T}} = \overline{\epsilon_{3T}} = 1/2 \text{ LSB}$$

Then

$$x_1(n+1) = x_1(n) + k(x_2(n) - x_3(n)) - \epsilon_{1T}. \tag{4.18.1}$$

$$x_2(n+1) = x_2(n) + k.x_3(n) - k[x_1(n)+k(x_2(n)-x_3(n)-\epsilon_{1T})]-\epsilon_{2T}$$

Collecting like terms yields

$$x_2(n+1) = -k.x_1(n) + (1-k^2)x_2(n) + (k+k^2)x_3(n) + k.\epsilon_{1T}-\epsilon_{2T} \tag{4.18.2}$$

Similarly

$$x_3(n+1) = x_3(n) + k(x_1(n+1) - x_2(n+1)) - \epsilon_{3T}$$

Expanding and collecting like terms yields

$$x_3(n+1) = (k+k^2)x_1(n) + (-k+k^2+k^3)x_2(n) + (1-2k^2-k^3)x_3(n) - \epsilon_{3T} + k(\epsilon_{2T}-\epsilon_{1T}-k.\epsilon_{1T}) \tag{4.18.3}$$

In practice $k < 0.25$ and therefore all terms containing $k.\epsilon_{jT}$ may be neglected. Thus eqs. (4.18) may be rewritten as follows

$$[x_j(n+1)] = [x_j(n)] * [I] - [\overline{\epsilon_{jT}}] \tag{4.19}$$

At each step $x_j(n+1)$ is reduced by $\overline{\epsilon_{jT}}$, which is equivalent to adding a negative constant offset. It was proven in section 3.8 that the [I] matrix is stable even when a constant offset is added or subtracted from the magnitude of each pulse width. In eq. (4.19) the negative offset is increased at each step, but stable conditions persist.

When the output of the [I] digital oscillator is observed on an oscilloscope, the effects of the truncation errors cause the sine wave to drift slowly downwards. Owing to the overflow properties of two's complement arithmetic, the sine wave drifts downwards off the bottom of the oscilloscope screen, and reappears at the top as shown in Fig. 4.4.

Therefore in any practical application of the [I] matrix it is important to include a round–up procedure in the multiplier algorithm, or multiplier hardware.

4.3.3 Phase and frequency errors

The observed behaviour of the vectors for N–phase oscillator matrices was similar to the behaviour of the vector for the 2–phase digital oscillator matrix. Fig. 4.5 shows the phase error $\lambda_j(n)$ for each of the three vectors over two complete cycles. The shapes of the phase error curves are independent of the initial phase angle ϕ, and the number of steps required to complete one cycle. The magnitude of the error is proportional to δ or $1/M$. It should be noted that all three curves coincide at one point which was less than M steps and that all the curves drift downwards. This was due to the fact that fewer than M steps are required to complete one cycle. If M' was used to calculate the angle of rotation, then all three curves coincided on the x axis.

The regular shape of the curves suggested that a simple Fourier expression could be derived. Each curve was analysed using a Fast Fourier Transform (FFT) program for $M = 2^q$ points (q integer). Only the fundamental and second harmonics were significant. It was also noted that for a given value of δ the Fourier coefficients were identical for different initial phase angles $\phi < 120^0$. It was also noted that the Fourier coefficients for the sets of curves for $M = 32, 64, 128$ were similar. Using the average value of the respective Fourier coefficients for the above curves, the following equations were derived

$$\lambda_1(n) = \frac{1}{M'} [-2.2 + \text{Cos}(n\delta) + 1.2*\text{Cos}(2n\delta) + 0.73*\text{Sin}(n\delta) - \\ - 0.54*\text{Sin}(2n\delta)] \qquad (4.20.1)$$

$$\lambda_2(n) = \frac{1}{M'} [-0.05 + 0.05*\text{Cos}(n\delta) - \text{Sin}(n\delta) + 1.27*\text{Sin}(2n\delta)] \qquad (4.20.2)$$

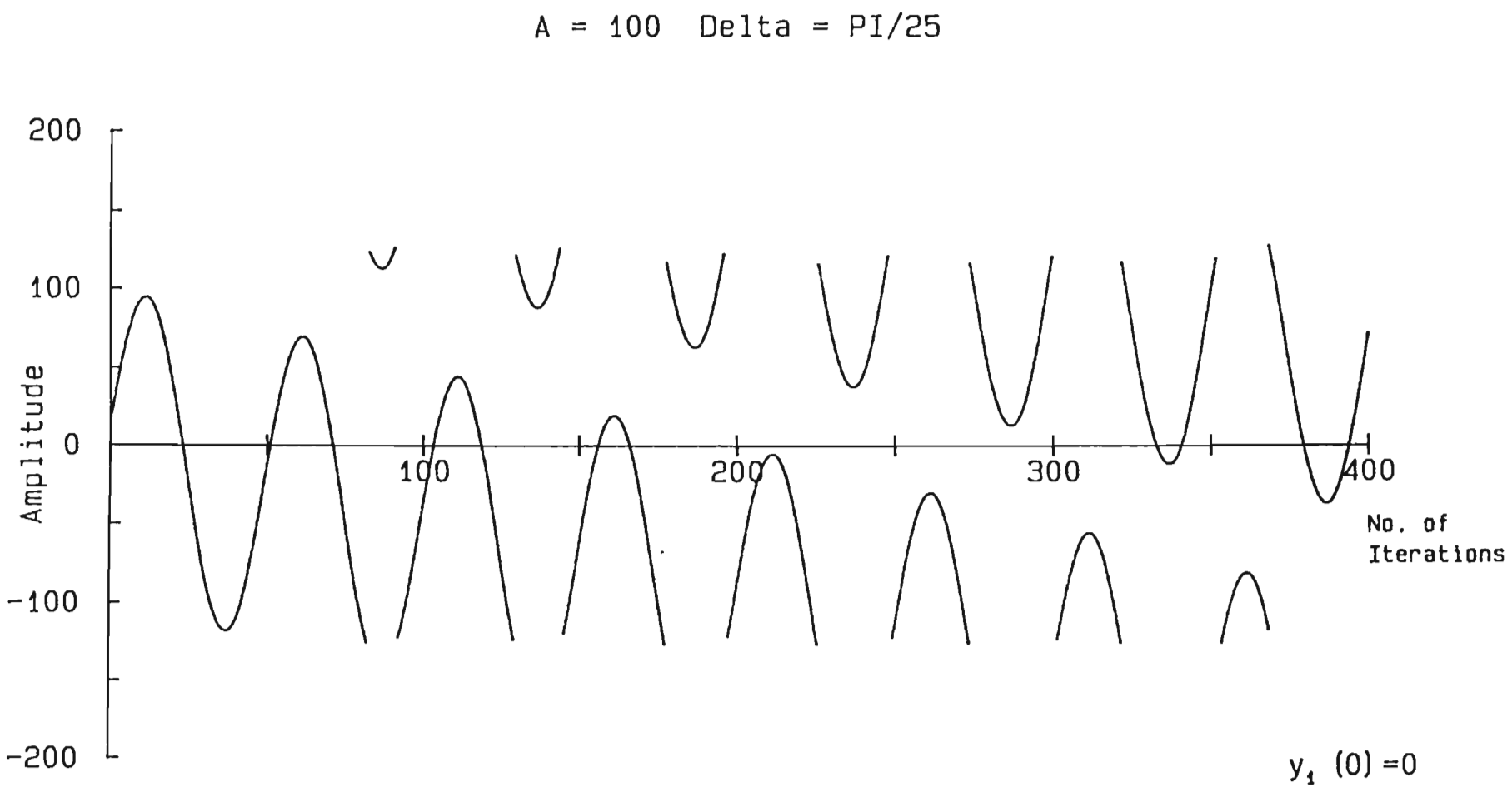


Fig. 4.4 Effect of Truncation Error on the [I] Matrix

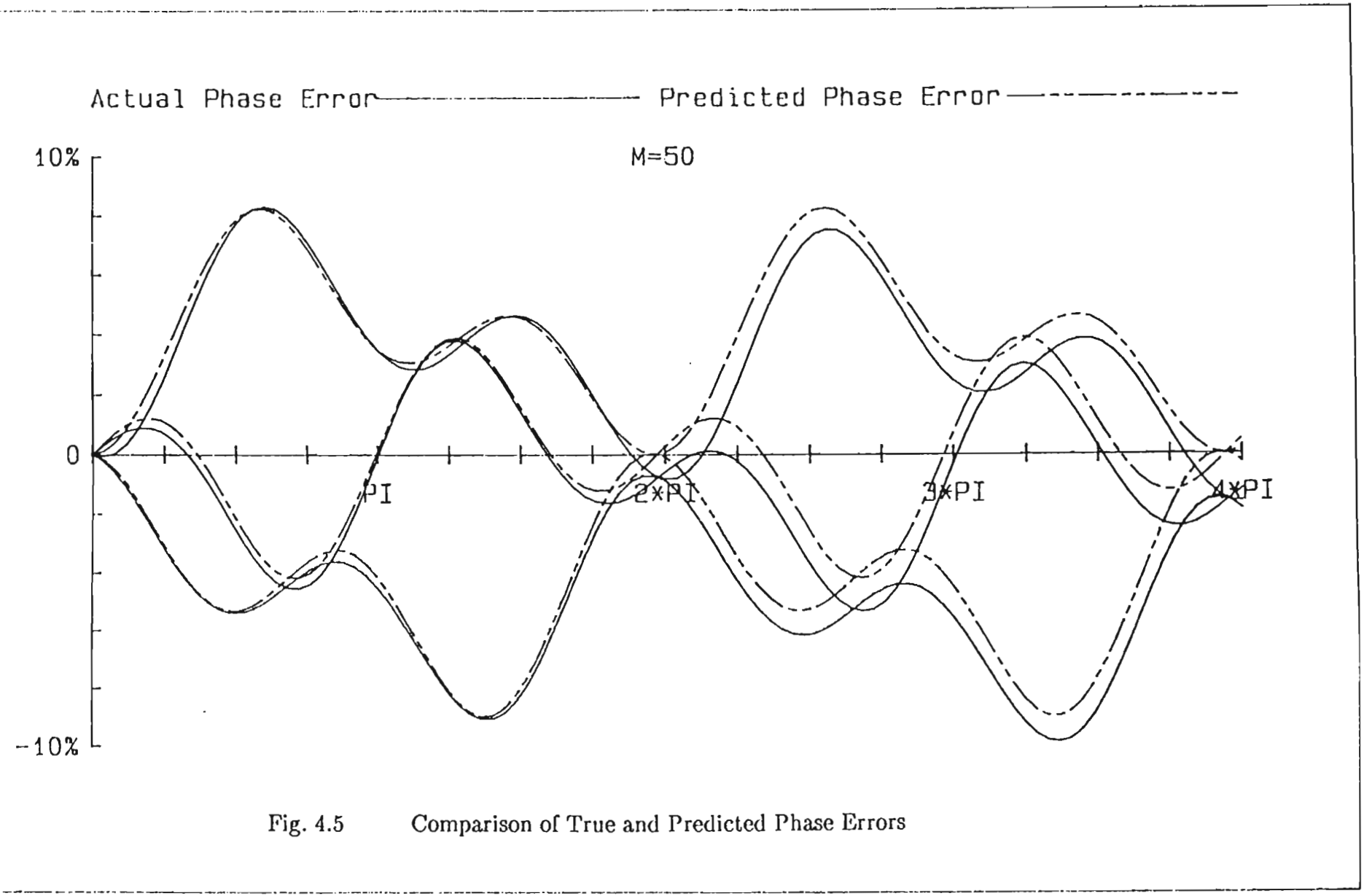


Fig. 4.5 Comparison of True and Predicted Phase Errors

$$\lambda_3(n) = \frac{1}{M'}\,[2.0 - 0.9*\mathrm{Cos}(n\delta) - 1.1*\mathrm{Cos}(2n\delta) + 0.8*\mathrm{Sin}(n\delta) - 0.32*\mathrm{Sin}(2n\delta)] \qquad (4.20.3)$$

The above expressions hold for values of M other than 32, 64, 128, and in Fig. 4.5 a value of $M = 50$ has been used. Prime and non–integer values of M have also been used to confirm that the expressions hold for a wide range of values of M' and δ'.

The above equations estimate the phase error and cannot be used to calculate M'. A simple formula for estimating M' was derived by experimentally determining a number of values for M' and then applying curve fitting algorithms. Values of M' were derived using techniques similar to those described in section 4.2.1 for the 2–phase oscillator matrix. The results for the [I] matrix are shown in Table 4.2. Linear regression techniques and a Chebyshev curve fitting algorithm both led to the following formula

$$\delta' = \delta + \frac{\delta^2}{9} = \delta.(1 + \delta/9) \qquad (4.21.1)$$

and $$M' = \frac{2\pi}{\delta.(1 + \delta/9)} \qquad (4.21.2)$$

Equations with higher order coefficients did not provide any improvement in accuracy, and the above equation was selected as the best fit.

The accuracy of eq. (4.21.2) is shown in Table 4.3 where error of approximation was always less than 0.15% except for values of $M < 20$. Attempts to derive an equation which was more accurate for $M < 20$ resulted in a poorer overall approximation. It should also be noted that

$$\frac{\delta}{\delta'} = \frac{\delta}{\delta(1+\delta/9)} = \frac{1}{(1+\delta/9)} \qquad (4.21.3)$$

Hence the $\lim_{\delta \to 0}(\delta') \to \delta$

and eq. (4.21.1) is consistent with observations of small values of δ. The above equation is not valid when truncation errors are significant. The effects of truncation errors on M' for small δ are discussed in Section 6.4. For small δ the frequency is also dependant on U_0, the initial lenghth of the vector as shown in Tables 6.6, 6.7 and 6.8.

Table 4.2

Comparison of Frequency and Vector Length Errors

M	δ	M'	(M'–M)	Observed Maximum $\xi'_j(n)$	Calculated Maximum $\xi_j(n)$
10	0.62832	9.26587	7.34%	42.0%	35.0%
20	0.31416	19.33288	3.34%	21.0%	20.0%
30	0.20944	29.36068	2.13%	14.0%	13.5%
40	0.15708	39.36833	1.58%	11.0%	10.3%
50	0.12566	49.35731	1.29%	8.6%	8.3%
60	0.10472	59.38738	1.02%	7.0%	7.0%
120	0.05236	119.38410	0.51%	3.5%	3.5%

Table 4.3

Comparison of True and Estimated Values of δ'

$$\delta_c = 2\pi/M'$$

M	δ	δ_c	δ' Eq. (4.21)	$(\delta_c - \delta')$
10	0.62832	0.67810	0.67218	0.873%
20	0.31416	0.32500	0.32513	–0.039%
30	0.20944	0.21400	0.21431	–0.146%
40	0.15708	0.15960	0.15982	–0.139%
50	0.12566	0.12730	0.12742	–0.093%
60	0.10472	0.10580	0.10594	–0.131%
120	0.05236	0.05263	0.05266	–0.066%

4.3.4 Vector length errors

The variation in the lengths of the three vectors rotated by the [I] matrix are shown in Fig. 4.6. The shapes of the three curves are identical, but are phase shifted by 120^{o} or $2\pi/3$ radians with respect to each other. The regular shape of the curves suggested that they could be described by a simple Fourier series. Using a FFT program for $M = 2^{q}$ points the following equations were derived.

$$\epsilon_1(n) = 1/M\,[0.13 - 0.68*\mathrm{Cos}(2n\delta') + \\ + 0.95*\mathrm{Sin}(n\delta') + 1.14*\mathrm{Sin}(2n\delta')] \qquad (4.22.1)$$

$$\epsilon_2(n) = 1/M\,[0.1 + 1.14*\mathrm{Cos}(n\delta') - 1.24*\mathrm{Cos}(2n\delta')] \qquad (4.22.2)$$

$$\epsilon_3(n) = 1/M\,[0.1 - 0.57*\mathrm{Cos}(n\delta') + 0.62*\mathrm{Cos}(2n\delta') - \\ - 0.987*\mathrm{Sin}(2n\delta') - 1.074*\mathrm{Sin}(2n\delta')] \qquad (4.22.3)$$

The dashed lines in Fig. 4.6 are the graphs produced by eqs. (4.22). The above equations were derived using values of M equal to 32, 64 and 128. The formulae were evaluated using prime and non–integer values of M and a good approximation to the true values was realised in each case. It should be noted that values of δ' were used to derive eqs. (4.22), and therefore δ' should be used instead of δ.

4.3.5 Pulse width errors

The pulse width errors for the [I] matrix are calculated using the following procedure

a) Given M, calculate δ' and M' from equations (4.21);

b) Calculate $n\delta'$ and hence $\lambda_j(n)$ and $\epsilon_j(n)$ using eqs. (4.20) and eqs. (4.22);

c) Calculate the pulse width error $\xi_j(n)$ from

$$\xi_j(n) = U_o.[1+\epsilon_j(n)].\mathrm{Cos}[n\delta'+\lambda_j(n)+(j-1).\theta+\phi] - U_o \qquad (4.23)$$

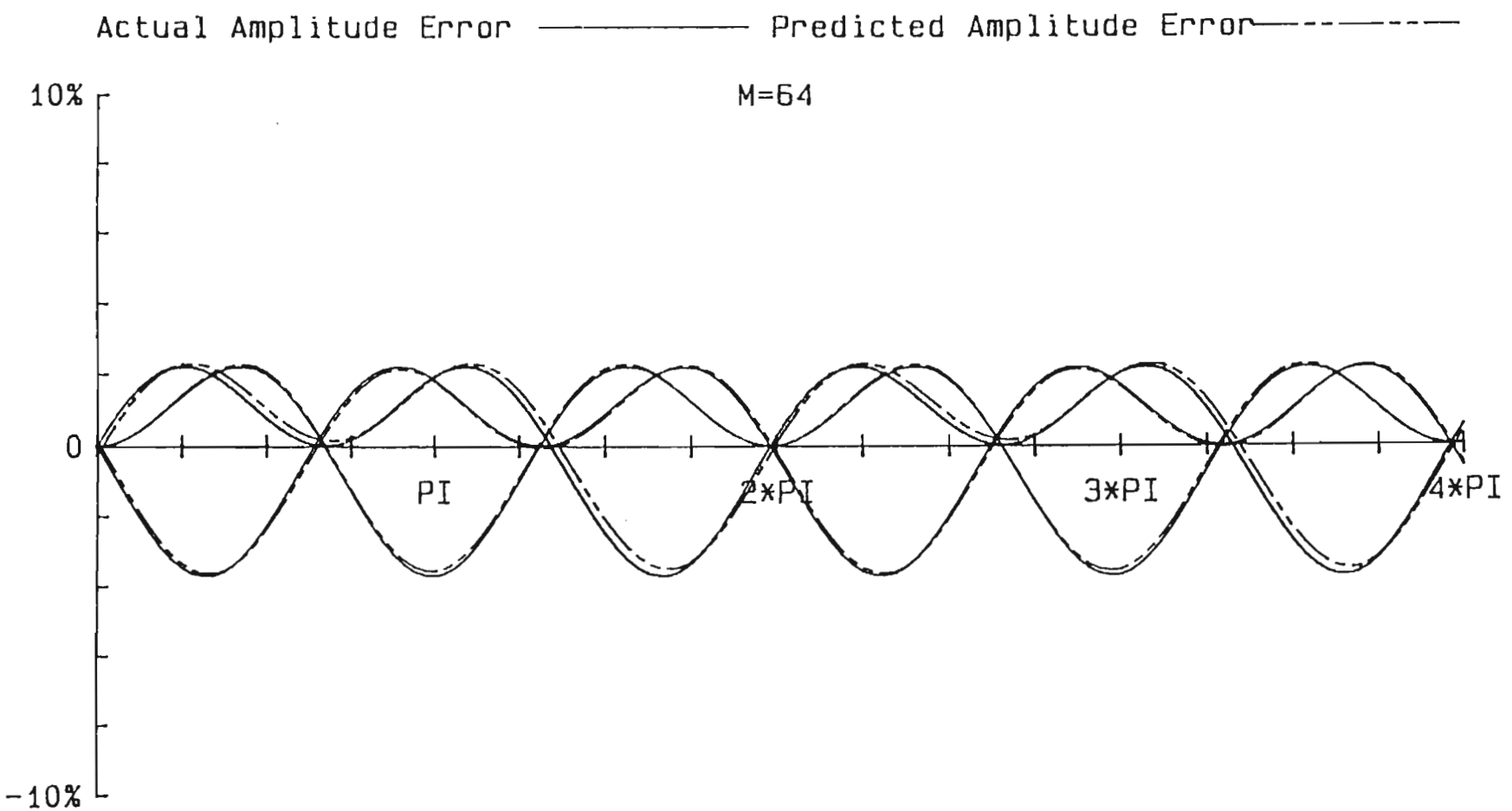


Fig. 4.6 Comparison of True and Predicted Vector Length Errors

The $[1+\epsilon_j(n)]$ term represents the change in length of the vectors and the $[n\delta'+\lambda_j(n)+(j-1).\theta+\phi]$ term the angle of rotation of the vector. Eq. (4.23) slightly underestimates the true value of $\xi'_j(n)$ as shown in Table 4.2.

The true value of the pulse width error was calculated from

$$\xi'_j(n) = x_j(n) - U_0.\mathrm{Cos}(n\delta'+(j-1).\theta+\phi) \tag{4.24}$$

Fig. 4.7 compares the true error of the magnitudes of the pulse widths and the error estimated by eq. (4.23). Both sets of curves follow the same trends, with the exception of Phase 2 for $0 < n\delta < \pi/3$. This was due to the poor approximation of $\lambda_2(n)$ in the range $0 < n < M/3$. Several attempts were made to obtain an explanation for the discrepancy and to derive a better curve but were unsuccessful. When the coefficients were altered to produce a more accurate representation in the range $0 < n < M/3$ a much poorer approximation to $\lambda_2(n)$ was realised in the range $M/3 < n < M$. Furthermore, the inclusion of higher order terms did not produce any significant improvement. It was considered that the equation for $\lambda_2(n)$ was adequate for most purposes, provided that the above limitation was taken into account.

Attempts were made to find a simple formula to calculate $\xi_j(n)$ directly by curve fitting and by Fourier transforms. However, no simple formula could be found which was as accurate as eq. (4.23) for as wide a range of values of M.

The term $\mathrm{Cos}(n\delta+\lambda_j(n)+j\theta)$ has some similarity with the fundamental equation for Phase Modulation, i.e.

$\mathrm{Cos}(\omega t+f(t)+\theta)$

It is possible that $\xi_j(n)$ could be calculated directly from Bessel functions, however this particular approach was not pursued owing to the relative difficulty of computing Bessel functions.

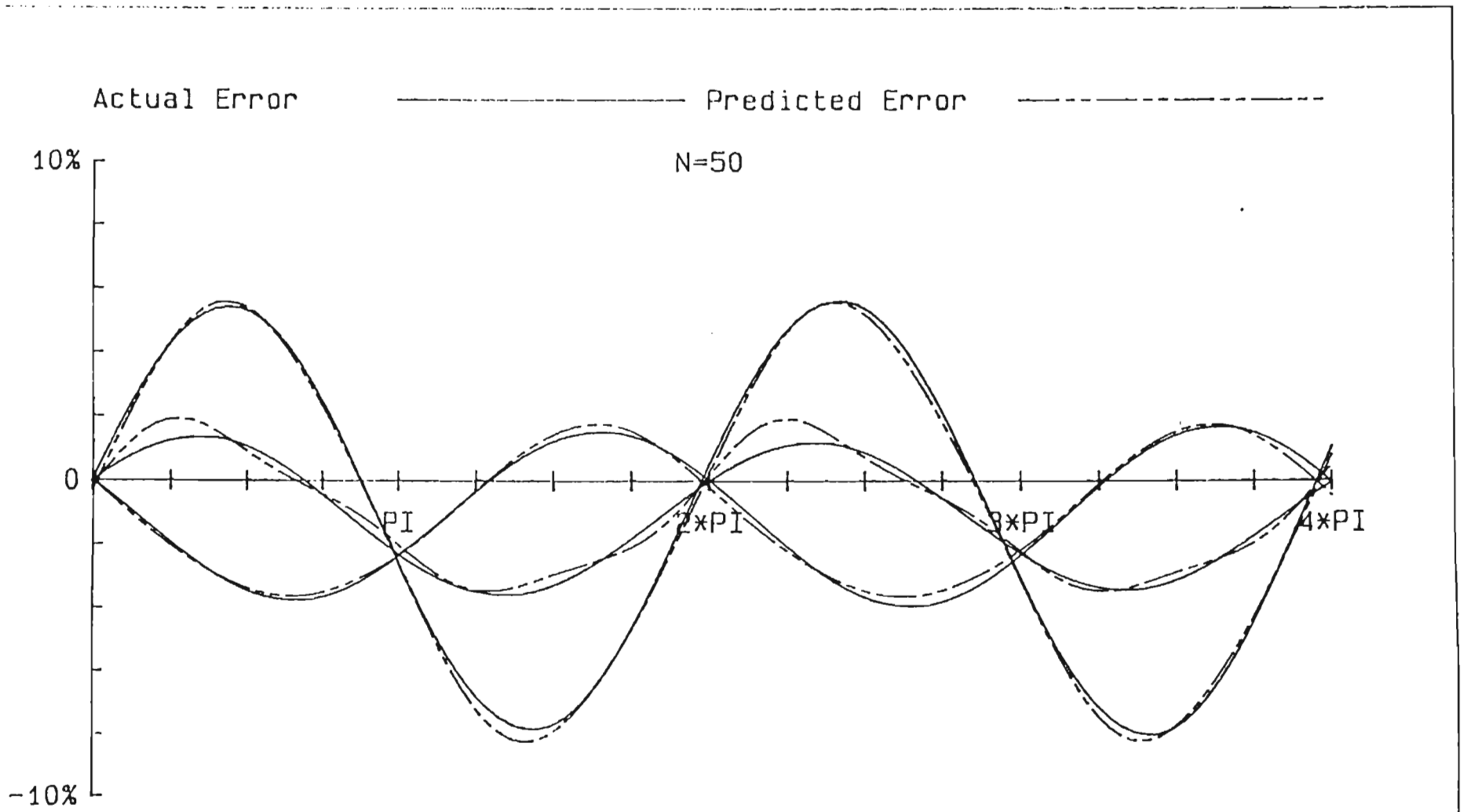


Fig. 4.7 Comparison of True and Predicted Pulse Width Errors

4.4 Conclusions

The magnitudes of the frequency and amplitude errors have been derived and the importance of rounding up for the N–odd phase oscillators has been demonstrated.

The frequency errors for PWM modulators appears as a slight increase in output frequency and, in general, is less than the slip frequency of an induction motor and may be neglected for most applications. If it is necessary to know the frequency accurately then either eq. (4.8) for the [T] matrix, or eq. (4.21) for the [I] matrix can be used to obtain a better estimate of the true frequency.

The upper–bound for the pulse width error is proportional to δ for both the two and N–odd phase oscillators. The largest error noted during numerous experiments was approximately $\delta/2$ and $2\delta/3$ for $N = 2$ and $N = 3$ respectively. Thus if δ is halved and two iterations are executed, the overall error is also halved. Hence it is advantageous to use a small value of δ and to execute a larger number of steps for each cycle.

Observations from many values of δ show that the minimum gear ratio M should be greater than 20 since lower values of M produce large errors. In practical applications the gear ratio should be larger than 40 and then the errors are less than 10% and should not significantly affect the performance of the digital PWM modulator.

CHAPTER FIVE

SPACE VECTOR MODULATION

5.1 Introduction

Space Vector Modulation (SVM) is an alternative PWM strategy to that described in Chapters 2 and 3, where both switching elements in one inverter leg are allowed to be switched off simultaneously. SVM requires at least one element to be turned on at any instant. SVM is subject to the same limitations as ordinary PWM techniques because high gear ratios require large look–up tables to store enough values to obtain sufficient accuracy. The two–phase [T] oscillator matrix can be applied to SVM to eliminate the need for a look–up table containing values of $\mathrm{Sin}\gamma$. In existing applications of SVM, the system controller produces two parameters, U_n and γ_n where γ_n is the phase angle of a vector of length U_n to the x axis. The SVM modulation process converts these two parameters into three pulse width values.

5.2 Description of SVM

The following brief explanation of SVM is taken from Kunz[27]. The inverter strategy used in SVM is based on the principle that at least one switching element in each inverter leg is turned on at any given time (excluding dead–band delays to allow for snubbing or turn–off delays). The time that each element is turned on is calculated from

$$\bar{U}(n) = U_n.e^{j\gamma_n} \tag{5.1}$$

where U_n and γ_n are calculated by the system controller. Since there are three inverter legs there are $2^3 = 8$ possible switching states defined as S_p, as shown in Fig. 5.1 i.e.

$$\bar{S}_p = 2/3.U.e^{j\gamma_p} \tag{5.2}$$

where U is the link voltage.
The sequence of switching states are arranged so that only one inverter leg changes state at any one instant. In one sampling interval T a symmetrical switching sequence is

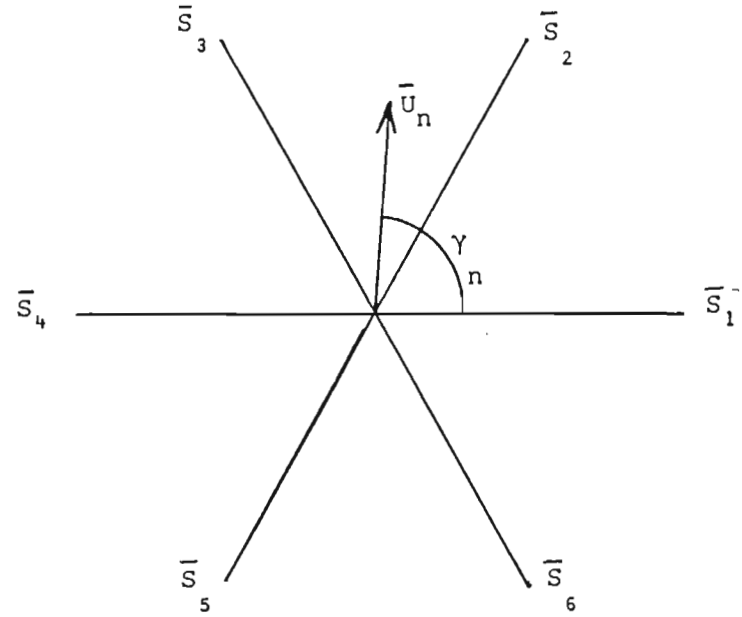


Fig. 5.1 Space Vectors in the Complex Plane

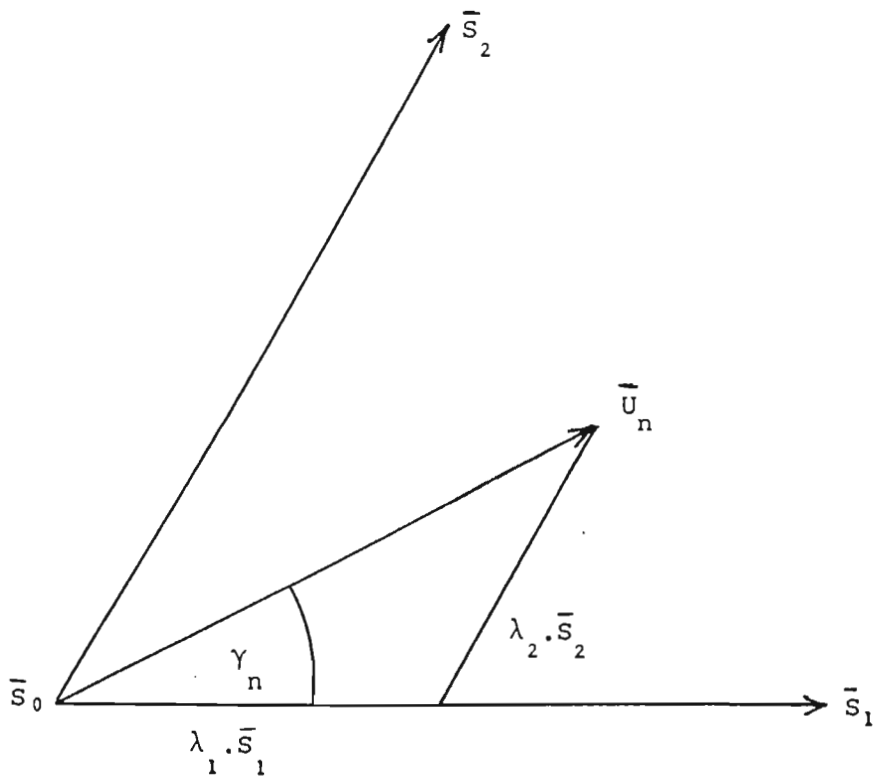


Fig. 5.2 Composition of Vector $\bar{U}_n$

selected as follows

$$S_0,\ S_p,\ S_{p+1},\ S_7,\ S_7,\ S_{p+1},\ S_p,\ S_0.$$

where S_0 and S_7 correspond to the states where all three phases are simultaneously connected to the negative or positive DC busbars. Let the time spent in each state be T_0, T_p, T_{p+1}, and T_7, and let $T_7 = T_0$ since both the states S_0 and S_7 represent conditions of zero voltage across the phases. All these states occur in one sampling interval T and hence

$$T = 2.T_0 + 2.T_p + 2.T_{p+1} + 2.T_7$$

or $$T = 4.T_0 + 2.T_p + 2.T_{p+1} \qquad (5.3.1)$$

Normalise eq. (5.3.1) by letting

$$\lambda_0 = T_0/T\ ,\quad \lambda_p = T_p/T \quad \text{and} \quad \lambda_{p+1} = T_{p+1}/T$$

hence

$$1 = 4.\lambda_0 + 2.\lambda_p + 2.\lambda_{p+1} \qquad (5.3.2)$$

The vector $\overline{U}_n$ is therefore the sum of three vectors whose lengths are proportional to the times (λ_p) spent in each state and whose directions are given by γ_p as shown in Fig. 5.2. Therefore,

$$\overline{U}_n = 4.\lambda_0.\overline{U}_0 + 2.\lambda_p.\overline{U}_p + 2.\lambda_{p+1}.\overline{U}_{p+1} \qquad (5.4)$$

Now $\overline{U}_0 = 0$ (N.B. A large λ_0 produces a small U_n)
and from eq. (5.2)

$$\overline{U}_p = 2/3.U.e^{j\gamma_p}$$

$$\overline{U}_{p+1} = 2/3.U.e^{j\gamma_{p+1}}$$

i.e. from eq. (5.4)

$$\overline{U}_n = U_n.e^{j\gamma_n} = 2.\lambda_p.\frac{2}{3}.U.e^{j\gamma_p} + 2.\lambda_{p+1}.\frac{2}{3}.U.e^{j\gamma_{p+1}} \qquad (5.5)$$

Using the Euler identity and seperating real and imaginary parts yields

$$U_n.Cos\gamma_n = 2.\lambda_p.\frac{2}{3}.U.Cos\gamma_p + 2.\lambda_{p+1}.\frac{2}{3}.U.Cos\gamma_{p+1} \tag{5.6.1}$$

$$U_n.Sin\gamma_n = 2.\lambda_p.\frac{2}{3}.U.Sin\gamma_p + 2.\lambda_{p+1}.\frac{2}{3}.U.Sin\gamma_{p+1} \tag{5.6.2}$$

Solving for λ_p and λ_{p+1} and noting that

$$Sin(\gamma_{p+1} - \gamma_p) = \sqrt{3}/2$$

yields

$$\lambda_p = \frac{\sqrt{3}.U_n.Sin(\gamma_{p+1} - \gamma_n)}{2.U} \tag{5.7.1}$$

$$\lambda_{p+1} = \frac{\sqrt{3}.U_n.Sin(\gamma_n - \gamma_p)}{2.U} \tag{5.7.2}$$

and from eq. (5.3.2)

$$2.\lambda_0 = 1/2 - \lambda_p - \lambda_{p+1} \tag{5.7.3}$$

5.3 Extension of SVM Formulae

At each step the eqs. (5.7) have to be evaluated, and this requires the calculation of two sine functions. The use of the [T] matrix eliminates the requirement for calculating trigonometric functions. However a modification to eqs. (5.7) is required so that they only contain $Sin\gamma$ and $Cos\gamma$ terms

The angle γ_n selects the two appropriate switching states as shown in Fig. 5.1 and therefore within a pair of switching states the angle γ_n is reduced to

$$\gamma'_n = \gamma_n - \gamma_p \tag{5.8.1}$$

and also

$$\gamma'_{p+1} = \gamma_{p+1} - \gamma_n \tag{5.8.2}$$

both of which are less than 60^0. It can also be proven that

$$\gamma'_{p+1} = 60 - \gamma'_n \tag{5.8.3}$$

Substituting eq. (5.8.1) and eq. (5.8.3) into eqs. (5.7) yields

$$\lambda_p = \frac{\sqrt{3}.U_n.\mathrm{Sin}(60 - \gamma'_n)}{2.U} \tag{5.9.1}$$

$$\lambda_{p+1} = \frac{\sqrt{3}.U_n.\mathrm{Sin}(\gamma'_n)}{2.U} \tag{5.9.2}$$

Expanding eq. (5.9.1)

$$\lambda_p = \frac{\sqrt{3}.U_n.}{2.U}\left[\mathrm{Sin}60.\mathrm{Cos}\gamma'_n - \mathrm{Cos}60.\mathrm{Sin}\gamma'_n\right] \tag{5.9.3}$$

$$\lambda_p = \frac{\sqrt{3}.U_n.}{2.U}\left[\frac{\sqrt{3}}{2}.\mathrm{Cos}\gamma'_n - \frac{1}{2}.\mathrm{Sin}\gamma'_n\right] \tag{5.9.4}$$

$$\lambda_p = \frac{\sqrt{3}.U_n.}{4.U}\left[\sqrt{3}.\mathrm{Cos}\gamma'_n - \mathrm{Sin}\gamma'_n\right] \tag{5.9.5}$$

The [T] matrix calculates the values of $\mathrm{Cos}\gamma'_n$ and $\mathrm{Sin}\gamma'_n$ from the values of $\mathrm{Cos}\gamma'_{n-1}$ and $\mathrm{Sin}\gamma'_{n-1}$ and δ where

$$\delta = \gamma'_n - \gamma'_{n-1} \tag{5.10}$$

The initial values for $\mathrm{Cos}\gamma'_n$ and $\mathrm{Sin}\gamma'_n$ are given by

$$[x_j(0)] = [\,\mathrm{Sin}\phi, \mathrm{Cos}\phi\,]$$

where $\phi = 0$ for $\gamma'_n > 60$ and

$\phi = 60$ for $\gamma'_n < 0$.

$\gamma'_n > 60$ and $\gamma'_n < 0$ occur when the vector U_n rotates to an adjacent pair of switching states and hence γ_p and γ_{p+1} assume new values.

Now λ_p and λ_{p+1} are calculated using eqs. (2.14), eq. (5.9.5) and eq. (5.9.2) as follows

$$\text{Sin}\gamma'_n \cong x_1(n) = x_1(n-1) + \delta * x_2(n-1) \quad (5.11.1)$$

$$\text{Cos}\gamma'_n \cong x_2(n) = x_2(n-1) - \delta * x_1(n) \quad (5.11.2)$$

Hence eq. (5.10.3) becomes

$$\lambda_p = \frac{\sqrt{3}.U_n.}{4.U}\left[\sqrt{3} * x_2(n) - x_1(n)\right] \quad (5.11.3)$$

and eq. (5.9.2) becomes

$$\lambda_{p+1} = \frac{\sqrt{3}.U_n}{2.U} * x_1(n) \quad (5.11.4)$$

$$2.\lambda_0 = 1/2 - \lambda_p - \lambda_{p+1} \quad (5.11.5)$$

It is possible for the system controller to be programmed to calculate $\frac{\sqrt{3}.U_n}{2.U}$ and δ directly and hence only five multiplications, one addition, one shift (divide by two) and five subtractions are required to calculate λ_0, λ_p, λ_{p+1} according to eqs. (5.11).

If a double sided pulse generator similar to that described in Appendix E.2 is used then the pulse widths P_a, P_b and P_c are calculated according to the previously chosen switching sequence. For example the symmetrical pulse width for P_b is given by the time λ_{p+1} spent in state S_{p+1}, λ_0 in state S_7 and so forth

$$S_0,\ S_p,\ S_{p+1},\ S_7,\ S_7,\ S_{p+1},\ S_p,\ S_0.$$

$$P_a = \lambda_0 + \lambda_0 = 2.\lambda_0 \quad (5.12.1)$$

$$P_b = \lambda_{p+1} + \lambda_0 + \lambda_0 + \lambda_{p+1} = 2.\lambda_0 + 2.\lambda_{p+1} \quad (5.12.2)$$

$$P_c = \lambda_p + \lambda_{p+1} + \lambda_0 + \lambda_0 + \lambda_{p+1} + \lambda_p$$

$$= 2.\lambda_0 + 2.\lambda_{p+1} + 2.\lambda_p \quad (5.12.3)$$

It is advantageous to modify eqs. (5.11) so that $2.\lambda_0$, $2.\lambda_p$ and $2.\lambda_{p+1}$ are calculated directly, and hence a higher precision is attained and fewer arithmetic operations are required to compute P_a, P_b and P_c

5.4 Compensation for Phase Errors

The system controller may have to compensate for the phase error introduced by the use of the [T] matrix. Eq. (4.8) provides a simple but adequate estimate of the true rotation of the vector $\overline{U}_n$. Owing to the need to re–initialise the values of $x_j(n)$ every time a new pair of switching states are used, it is possible to ignore the phase errors for high gear ratios ($M > 100$).

5.5 Conclusions

This chapter has shown how the [T] matrix provides a simple method to reduce the computational requirements for Space Vector Modulation. However the limitations of the size of δ should be considered in any application. If large values of δ are expected to occur in an application then it would be advantageous to execute two calculation steps of the [T] matrix to reduce errors. It is possible to overlook the phase error for systems with high gear ratios, however eq. (4.8) could be used to calculate the true angle of rotation if required.

CHAPTER SIX

APPLICATION DESIGN CRITERIA

6.1 Introduction

In the preceding chapters the characteristics of multi–phase oscillator matrices have been derived assuming infinite precision. Finite precision and other factors must be considered before the equations are implemented in a practical circuit. The requirement for rounding–up rather than truncation has already been demonstrated in section 4.3.2. The precision of the arithmetic unit affects not only the maximum range of U_o and δ, but will also affect the quality or purity of the sinewave that is generated. Unwanted harmonics are produced by PWM pulses and the errors introduced by the precision of the PWM modulator. The presence of undesirable harmonics was a major parameter in determining the acceptable range of values of δ or k.

The word length L of the PWM modulator defines the maximum value of U_o, the magnitudes of harmonics, the frequency range and the number of steps which can be performed in a given time.

6.2 Maximum Value of U_o

Assuming two's complement arithmetic, the range of numbers for an L–bit processor is $\pm 2^{L-1}$. However, owing to the possibility of errors as large as $\delta_{max}/2$, allowance must be made for a maximum value of $U_o(1+\delta_{max}/2)$. For the 2–phase oscillator, this is an adequate safety margin, and hence

$$U_o = \frac{2^{L-1}}{(1+\delta_{max}/2)} \qquad (6.1)$$

An additional factor must be taken into account for 3–phase calculations. Allowance must be made for the maximum value of

$$[U.\mathrm{Sin}(\phi+\theta) - U.\mathrm{Sin}(\phi+2\theta)]$$

In Appendix A it is proven that

$$U.Cos\phi.\tan(\theta/2) = U.Sin(\phi+\theta) - U.Sin(\phi+2\theta) \quad (A.6)$$

Hence the maximum value which could occur when evaluating the above equation is $U_o(1+\delta_{max}/2).(\tan(\theta/2))$. Thus the maximum value of U_o for a 3–phase system using a L–bit processor is

$$U_o = \frac{2^{L-1}}{(\tan(\theta/2)).(1+\delta_{max}/2)} \quad (6.2)$$

or

$$U_o = \frac{2^{L-1}}{(\tan(\pi/N)).(1+\delta_{max}/2)} \quad (6.3)$$

Substituting for $N = 3$ yields

$$U_o = \frac{2^{L-1}}{\sqrt{3}.(1+\delta_{max}/2)} \quad (6.4)$$

Since $\tan(\pi/N) < 1$ for $N \geq 5$, for all other matrices, the maximum value of U_o that could occur is given by

$$U_o = \frac{2^{L-1}}{(1+\delta_{max}/2)} \quad (6.5)$$

6.3 Harmonics

6.3.1 Basic concepts

The performance of an induction motor is adversely affected by the presence of harmonics and subharmonics and it is important to determine what harmonics are produced by PWM pulses. Bowes[3,4] has carried out a rigorous analysis of PWM waveforms and associated harmonics. The important conclusions that can be derived from both these papers are

a) Double edge PWM sampling produces fewer harmonics than single edge PWM sampling (Bowes[3] p.608).

b) Subharmonics of large amplitude can exist if the ratio of the modulating frequency to the carrier frequency is non–integer (Bowes[3] p.610).

c) For 3–phase PWM systems, if even values of M' are used, then only odd harmonics of the modulating frequency exist. However, if there exists a significant phase displacement between the 3–phase modulating signals and the carrier, then amplitude and phase unbalance can occur at certain frequency ratios. (Bowes[4] p.1283).

d) The highest possible value of M' should be used. (Bowes[4] p.1283).

6.3.2 Amplitudes of harmonics due to non–intger M.

In any practical implementation the value of k is a ratio of two integer numbers The denominator is 2^L if the sign of δ is restricted to positive values, or 2^{L-1} if both positive and negative values are allowed.

hence $\delta = D/2^L$ or $\delta = D/2^{L-1}$ respectively

and $M = 2\pi/\delta = \frac{2\pi.\ 2^L}{D}$ for integer $D < 2^L$

Thus M and M' are both non–integer since π is an irrational number. Since M is by definition non–integer, subharmonics will occur in any practical implementation. However Bowes[4] has shown that the amplitudes of the harmonics decrease for increasing M, and in practice low amplitude subharmonics are produced. An experiment was carried out to investigate the effect of non–integer values of M', and to establish the amplitudes of both harmonics and subharmonics under normal operating conditions with a trailing–edge modulator. Using the 8–bit 3–phase modulator described in chapter 7, the output of Phase 1 of was connected to a real time FFT Analyser. The average of a number of measurements was calculated and the log(magnitude) of every harmonic was recorded (the range of frequencies was from dc to twice the carrier frequency). The power of all the significant harmonics were then added together (a significant harmonic was 10 dB above the general background noise level of –45 dB). The results of the experiment are given in Tables 6.1. and 6.2.

Table 6.1

Three Phase Harmonic Analysis for M < 30

$U_o = 64$ Depth of Modulation = 80% $\delta = D/128$

M	D	δ	f' (Hz)	f (Hz)	Harmonic Power (dB)	Sub–Harmonic Power (dB)
15	62	0.419	161	163	–20	–22
16	58	0.393	151	146	–17	–21
17	55	0.372	142	140	–20	–32
18	52	0.352	134	131	–23	–23
19	49	0.331	126	123	–23	–21
20	46	0.312	118	116	–23	–23
22	43	0.284	110	107	–25	–25
25	37	0.251	95	97	–27	–35
27	34	0.230	87	85	–25	–25
30	31	0.209	79	79	–25	–24

Table 6.2

Three Phase Harmonic Analysis for M > 100

$U_o = 64$ Depth of Modulation = 80% $\delta = D/128$

M	D	δ	f' (Hz)	f (Hz)	Harmonic Power (dB)	Sub–Harmonic Power (dB)
310	3	0.020	7.5	6.6	–25	–36
232	4	0.027	10.0	10.8	–29	–42
186	5	0.034	12.5	12.4	–24	–34
155	6	0.041	15.0	14.6	–24	–38
133	7	0.047	17.5	17.6	–21	–26
116	8	0.054	20.0	21.0	–20	–22
103	9	0.061	22.5	22.4	–21	–23

All dB measurements are relative to the power of the modulating wave

For a wide range of values of δ, there were always two sidebands close to the modulating frequency. The remaining significant harmonics were the second, third and carrier frequency harmonics. In Tables 6.1 and 6.2 the powers of all the significant harmonics and significant subharmonics have been calculated seperately. The harmonics related to the carrier frequencies have been excluded because they are easily removed by filtering.

For values of M less than 20, the number of significant subharmonics increased and the amplitudes of higher order–harmonics increased exponentially. The number and amplitude of significant harmonics and subharmonics decreased as M increased in value. The amplitude of the largest harmonic was always less than 3% of the desired output wave for $M \geq 20$. The main conclusion of the above tests was that values of M less than 20 are undesirable.

For smaller values of δ or large values of M, the amplitudes of the subharmonics decreased as δ decreased, even when the fundamental wave was distorted. The total power of the harmonics was reasonably constant for values of $(\delta > 5/(2^{L-1})$; however for values of $\delta < 5/(2^{L-1})$ the number and amplitude of significant harmonics increased considerably. Thus the minimum acceptable value of δ is

$$\delta_{min} = 5/(2^{L-1}).$$

Tables 6.1 and 6.2 also compare the expected frequency f with the true frequency f' realised in the 8–bit system described in Chapter 7. It is important to note that for $M < 20$ and for very small δ there were some significant differences between eq. (4.21) and the true values realised in practice. These discrepancies are discussed at the end of section 6.4.

6.4 Maximum Range of δ or k

The upper limit for δ or k is controlled by the maximum permissible error or the amount of power in the harmonics (see section 6.3) that can be tolerated. Experimental evaluation of the performance of the [I] and [T] oscillator matrices indicated that

$$\delta_{max} < 2\pi/20 \quad \text{or} \quad M > 20$$

Higher values of δ_{max} produced unacceptably large amplitude errors and hence

$\delta_{max} \leq 0.314$

thus for $N = 3 \quad k_{max} = 0.181$
$N = 5 \quad \ell_{max} = 0.432$

In section 6.3 it was shown that the minimum acceptable value of δ was $5/(2^{L-1})$. This is verified by examining eqs. (2.14) as follows

$$x_1(n+1) = x_1(n) + \delta.x_2(n) \tag{2.14.1}$$
$$x_2(n+1) = x_2(n) - \delta.x_1(n+1) \tag{2.14.2}$$

For integer arithmetic and $\delta = D/2^L$ then

$$\delta.x_1(n) = \frac{D}{2^L}.x_1(n) \tag{6.6}$$

Hence the integer representation of $\delta.x_1(n)$ is $D.x_1(n)$ and the division by 2^L is due to the wordlength of the L–bit processor.

If $D.x_1(n) < 1$ then $\delta.x_1(n)$ is truncated to 0. A similar analysis applies to $\delta.x_2(n)$. The above reasoning shows that if $D.x_2(n)$ or $D.x_1(n+1)$ are less than unity then the values of $x_1(n+1)$ and $x_2(n+1)$ are unchanged. If the maximum values of

$$0.5 \leq D.x_1(n+1) < 1.5, \quad \text{and}$$
$$0.5 \leq D.x_2(n) < 1.5$$

then a truncated triangular wave is produced, and the error in approximating a sinewave is large. In general, if the maximum integer value of $D.x_1(n)$ (or $D.x_2(n)$) is J, then each half of the sinewave is approximated by $2J + 1$ straight lines. Thus if J is too small a non–sinusoidal waveform is produced, the pulse width error increases and a large number of harmonics are generated. Tables 6.3, 6.4 and 6.5 show that the maximum pulse width error increases for very small δ. In all cases the error was less than $\delta_{max}/2$ except when the maximum value of

Table 6.3

Summary of Error Analysis for Small δ

Eight Bit Systems

$L = 8$ $U = 107$

$\delta = D/256$

D	M	M'	Max. Amplitude Error
3	536	598	9.8%
4	402	384	8.9%
5	322	316	3.8%
6	268	266.75	8.0%
7	230	223	5.8%
8	201	210	4.5%
9	179	174	2.4%
10	161	162	1.1%
11	146	146	1.2%
12	134	133	2.5%
13	124	126	1.5%
14	115	114	1.2%
15	107	108	2.0%
16	101	99.7	3.1%
17	95	94	1.2%

Table 6.4

Summary of Error Analysis for Small δ

12 Bit Systems

L = 12 U = 1707

δ = D/4096

D	M	M'	Max. Amplitude Error
7	3677	3557.67	2.66%
8	33217	3330.67	1.81%
9	2860	2780.50	2.43%
10	2574	2579.25	1.06%
11	2340	2338.00	1.77%
12	2145	2125.00	1.17%
13	1980	2025.8	1.39%

Table 6.5

Summary of Error Analysis for Small δ

16 Bit Systems

L = 16 U = 27307

δ = D/65536

D	M	M'	Max. Amplitude Error
12	34315	33997	1.13%
13	31675	32439	1.21%
14	29412	29101	1.23%
15	27452	27634	0.59%

$$D.x_1(n) \leq 3 \tag{6.7}$$

This is similar to the conclusion deduced in section 6.3 and Table 6.2 where the minimum acceptable product of $D_{min}.U_0$ was shown to be greater than 3.

i.e. for $D_{min} = 5$ or $\delta_{min} = 5/128$ and $U_0 = 64$

then $D_{min}.U_0 = 2.5$ (which is rounded up to 3)

Thus the minumum value of δ_{min} is estimated from eq. (6.7)

$$\text{hence} \qquad \delta_{min}.\frac{2^{L-1}}{(1+\delta_{max}/2)} = 3 \quad N \neq 3 \tag{6.8}$$

$$\text{thus} \qquad \delta_{min} = \frac{3.(1+\delta_{max}/2)}{2^{L-1}} \tag{6.9}$$

$$\text{for} \quad N = 3 \qquad \delta_{min} = \frac{\sqrt{3}.3.(1+\delta_{max}/2)}{2^{L-1}} \tag{6.10}$$

Since δ_{max} is limited to 0.314, δ_{min} and the maximum acceptable output frequency range can be calculated for any given L–bit computer or logic system.

Table 6.6 shows the maximum frequency range for various wordlengths, where the limitations of an 8–bit system are shown.

For low values of δ another undesirable property is the variation of M' with respect to U_0. This is due to the fact that the values of $|D_{min}.U_0.Sin\phi|$ are limited to the integer numbers 0 to $D_{min}.U_0$. If $D_{min}.U_0 = 3$ then $|D_{min}.U_0.Sin\phi|$ are limited to the integer values 0, 1, 2, 3. If U_0 is reduced slightly then $|D_{min}.U_0.Sin\phi|$ can only assume the integer values 0, 1, 2. Hence more steps are required to complete one cycle. This is shown in Tables 6.7 and 6.8 which show the variation of M' with U_0 for low values of δ.

There were a number of combinations of D and U_0 which gave rise to a varying M' for each cycle, and in every case it was due to the change in U_0 from arithmetic errors. This effect was only observed with small values of δ, in the ranges given in Tables 6.7

Table 6.6

Comparison of System Performance for

Different Wordlengths

		$\delta_{max}/(\tan(\pi/N))$	U_{max}	δ_{min}	$\frac{\delta_{max}}{\delta_{min}}$
	N = 2	0.314	110	0.027	12
L = 8	N = 3	0.181	64	0.047	4
	N = 5	0.431	110	0.027	16
	N = 2	0.314	1765	0.0017	184
L = 12	N = 3	0.181	1020	0.0029	62
	N = 5	0.431	1765	0.0017	253
	N = 2	0.314	28248	0.000106	2956
L = 16	N = 3	0.181	16310	0.000182	1000
	N = 5	0.431	28248	0.000106	4058

Table 6.7

Variation of M' with Vector Length U_o

L = 8 bit

D \ U_o	95	100	105	110
5	306	306	312	322
6	282 ±6	288 ±6	268 ±1	266
7	227.3	222	222	226
8	205 ±7	204 ±5	204 ±7	198 ±7
9	186	185.67	175.25	175
10	156 ±1	158	161 ±1	164 ±1
11	146	148	146	143.6
12	135 ±3	132 ±1	134 ±2	134

Table 6.8

Variation of M' with Vector Length U_o

L = 12 bit

D \ U_o	95	100	105	110
7	3618	3567.3	3518	3554.3
8	3176 ±4	3229 ±6	3276 ±4	3229 ±4
9	2990	2875	2830 ±1	2786
10	2511 ±1	2532 ±1	2555 ±1	2576
11	2344	2374	2418	2344
12	2154 ±2	2109 ±2	2110	2124 ±2

±6 indicates the range of variation of M' with each cycle

and 6.8.

6.5 System Design Criteria

In order to maintain constant flux in an induction motor, it is necessary to reduce the inverter output voltage for low frequencies. At high frequencies the highest possible voltage (without exceeding the insulation rating) should be used. At lower frequencies the applied voltage should be reduced to limit the current to the maximum permissible rating of the windings. Ideally a constant V/f ratio should be maintained. At very low frequencies the resistances of the windings become significant and a constant (or even slightly increasing) V/f characteristic may be required.

The speed range of most induction motors is 25:1, and hence 8–bit logic systems are inadequate, as shown in Table 6.6. The minimum word length of any bit–slice or VLSI controller should be 12–bits and the minimum word length of any microprocessor based controller should be 16–bits. Eight bit microprocessors are generally too slow, produce large truncation errors from the short word length, and undesirable harmonics are generated at maximum inverter frequency.

It has been proven in section 3.6 that the [G] matrix can be used to control the lengths of the vectors and thus the output voltage independently of the frequency. However, for small values of U_n a coarse, stepped sinewave is produced. Therefore, to minimise the distortion and hence the amplitudes of the harmonics, the value of U_n should be as high as possible, so that as close an approximation as possible to a sinewave is realised. However in practical applications, at low speed the output voltage is small and hence the value of U_n is small when k is small, and therefore at low speeds the [G] matrix will produce undersirable harmonics.

A simple solution is to have a constant U_n which is as large as possible, and to introduce a voltage control variable. The values produced by the [I] matrix are multiplied by the voltage control variable to produce the required output voltage. The V/f characteristic is then defined by the relationship between k and the voltage control variable. Thus any arbitrary V/f characteristic may be realised. This method uses the same number of arithmetic operations as the [G] matrix. In certain applications it is simpler to apply because the value of "g" does not have to be calculated. In section 3.9 it was shown that a constant V/f characteristic was available directly from the calculation sequence, $\delta.x_j(n)$.

6.6 N–Even Multiphase Controllers

Eq. (3.5) is only valid for systems with an odd number of phases. It is possible however to generate pulse widths for systems with an even number of phases. A 4–phase modulator is derived from the 2–phase oscillator matrix by simply changing the signs of the two computed phases. Similarly a 6–phase controller is derived by using a 3–phase oscillator matrix, and using the computed values with the two's complement (negative) value of each phase. Hence 4,6,10,14,18, ... phase modulators are implemented using 2,3,5,7,9, ... phase oscillator matrices. Eight, 12, 16, 24, ... phase modulators are constructed by using two or more independent 2 or 3–phase oscillator matrices, with the correct relative initial phase angles. An 8–phase modulator is constructed from two 2–phase matrices, with an initial phase difference of 45^{O} between them. Similarly a 12–phase system is derived from two 3–phase matrices with an initial phase difference of 30^{O} between them.

It is possible to derive N–even multiphase oscillator systems by using the 2–phase oscillator matrix and then computing the required pulse width with standard trigonometric formulae. However with the relatively large number of arithmetic operations required to calculate each phase amplitude, the methods described above are more efficient and require fewer arithmetic operations.

6.7 Speed Control

There are a number of applications demanding a constant speed. Although the synchronous motor satisfies this requirement, an induction motor can be used provided some form of speed sensing equipment is available to control the value of k. If the speed is too low then increasing the value of k causes the speed of the motor to increase, or if the speed is too high then k can be correspondingly decreased.

The value of k can be changed by adding or subtracting a constant value, which depends on the system time constants. Since the value of k directly controls the speed, no other computations are required. If the [O] or [H] matrices are used, every speed change would require new values of $\mathrm{Cos}\delta$ and $\mathrm{Sin}\delta$. Thus the [I] matrix enables a very simple speed control system to be constructed.

6.8 DC Ripple Reduction

The classical theoretical analysis of PWM waveforms assumes that a constant voltage dc supply is available. No allowances for the effects of dc voltage variations or ripple voltages are made. However in practice it is unlikely that the inverter bridge is supplied with a constant and ripple free dc voltage. Most industrial dc power supplies produce a significant amount of ripple, which introduces unwanted harmonics and distorts the waveform as shown in Fig. 6.1.

The undesirable disturbance of the ripple voltage and variations in supply voltage can be reduced by adjusting the widths of the PWM pulses to compensate for these fluctuations. Let V_r be a reference voltage (or the equivalent to an ideal dc supply voltage) and let V_s be the instantaneous dc link voltage as measured by an L–bit A/D converter. A corrected pulse width value $x_c(n)$ is computed from the calculated pulse width $x_j(n)$ (e.g. from eqs. (3.14)) using the equation

$$x_c(n) = x_j(n) * C_f \qquad (6.11)$$

where the correction factor $C_f = \dfrac{V_r}{V_s}$

Eq. (6.11) is derived from the requirement that the average voltage produced under ideal conditions should be equal to the average voltage produced by the corrected pulse width as shown in Fig. 6.2. i.e.

$$x_j(n) * V_r = x_c(n) * V_s$$

The calculation of $x_c(n)$ requires one division and one multiplication. The division operation is the slowest arithmetic operation in any logic circuit, and hence it is undesirable to use eq. (6.11). By introducing an appropriate change of variable and restricting the choice of V_r to a few selected values, the computional effort to calculate $x_c(n)$ can be reduced substantially as shown below.

Let $x_c(n)$, $x_j(n)$, V_r and V_s be L bit integers i.e.

$$0 \le x_c(n) \le 2^L - 1$$

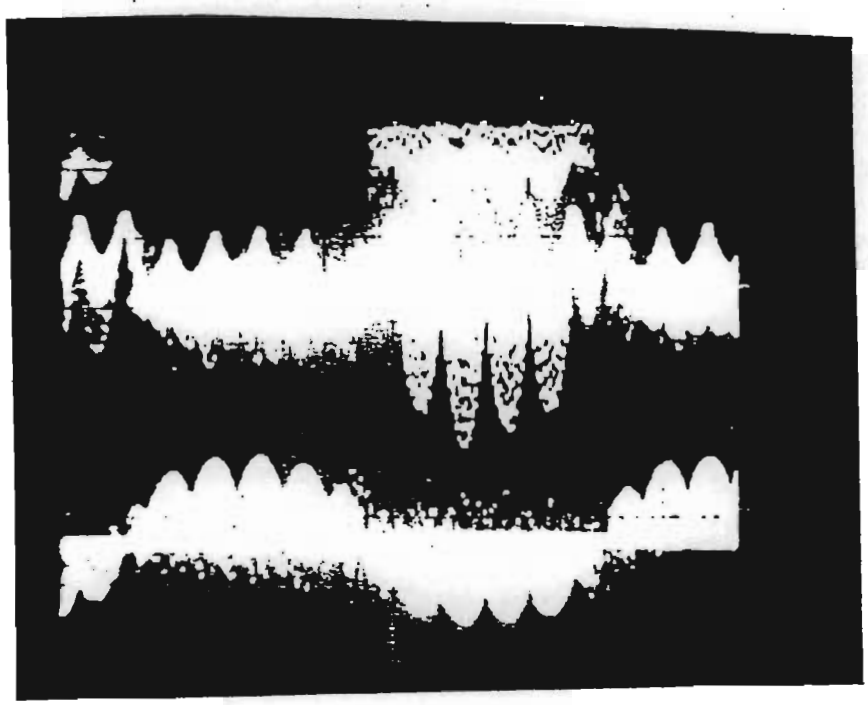


Fig. 6.1 Uncompensated Inverter Output

(a) Trigger Waveform Scale 20V/div

(b) Inverter Output Scale 50V/div

Horizontal Scale 5ms/div

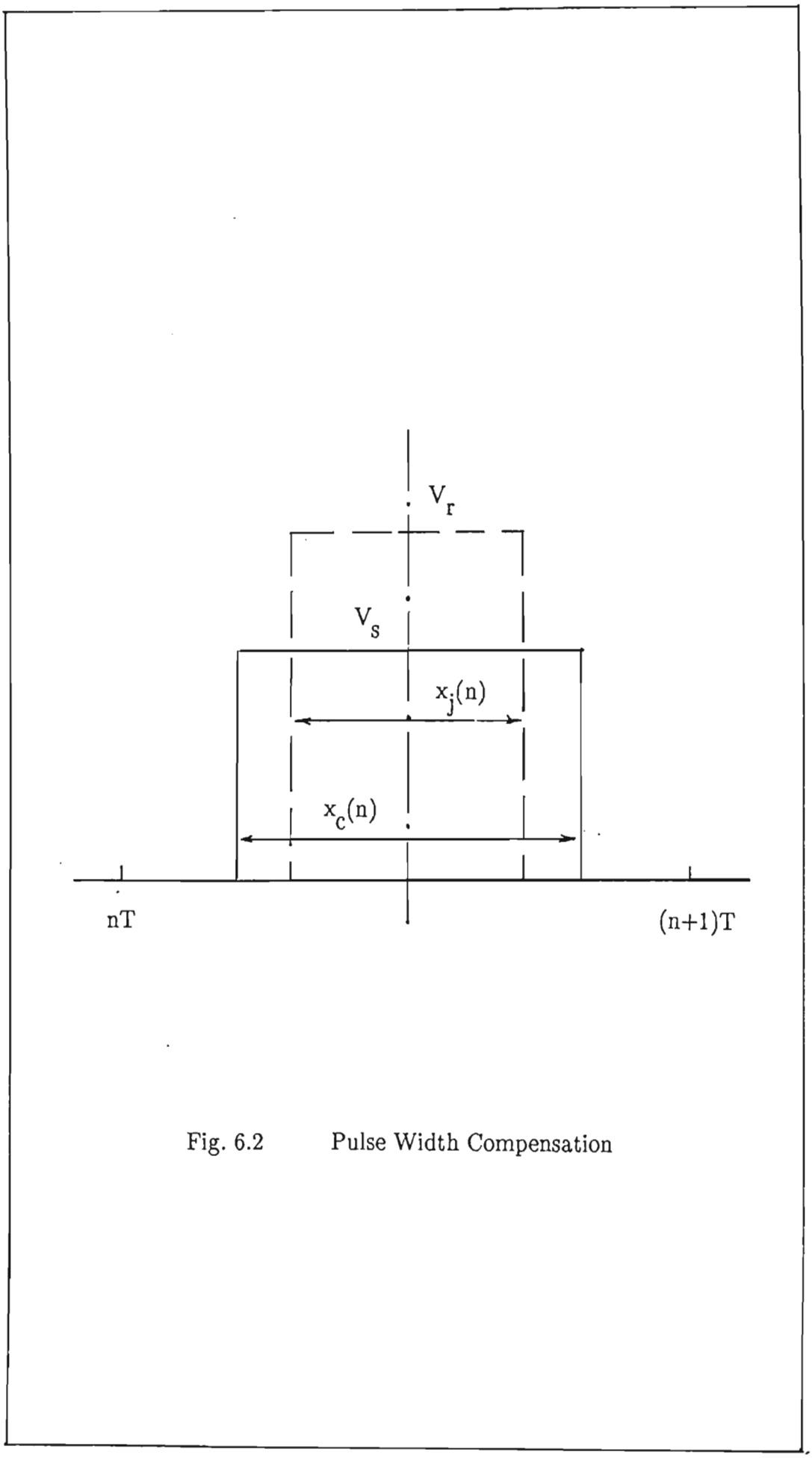


Fig. 6.2 Pulse Width Compensation

Introduce a change of variable σ where

$$V_s = V_r - \sigma \tag{6.12.1}$$

hence

$$C_f = \frac{V_r}{V_s} = \frac{V_r}{V_r - \sigma} = \frac{1}{1 - \sigma/V_r} \tag{6.12.2}$$

since $\sigma/V_r << 1$ the last part of eq. (6.12.2) can be approximated by

$$C_f = \frac{1}{1 - \sigma/V_r} \simeq 1 + \sigma/V_r \tag{6.12.3}$$

expanding and substituting eq. (6.12.1) to eliminate σ, yields

$$C_f \simeq \frac{1}{V_r} . \left[2.V_r - V_s\right] \tag{6.13}$$

Eq. (6.13) can be simplified by using selected values of V_r where

$$V_r = 2^L(1 - 2^{-m}) \tag{6.14}$$

and m is an integer $m = 0, 1, 2, 3, \ldots$

Now $2^{-m} < 1$ and hence

$$1/\tilde{V_r} = 1/\left[2^L.(1 - \frac{1}{2^m})\right] \simeq \frac{1}{2^L}.(1 + \frac{1}{2^m} + \frac{1}{2^{2m}}) \tag{6.15}$$

Substituting eq. (6.14) and eq. (6.15) into eq. (6.13) yields

$$C_f \simeq \frac{1}{2^L}\left[1 + \frac{1}{2^m} + \frac{1}{2^{2m}} \right].\left[2.2^L.(1 - \frac{1}{2^m}) - V_s \right] \tag{6.16.1}$$

$$C_f \simeq 2.\left[1 + \frac{1}{2^m} + \frac{1}{2^{2m}} \right].\left[1 - \frac{1}{2^m} - \frac{V_s}{2^{L+1}} \right] \tag{6.16.2}$$

When eq. (6.16.2) is expanded all the intermediate terms cancel out (c.f. $(1-a)*(1+a) = 1-a^2$) and hence

$$C_f \simeq \frac{1}{2^L}\left[2^{L+1}\left(1 - \frac{1}{2^{3m}}\right) - V_s - \frac{V_s}{2^{3m}} \right] \qquad (6.17)$$

The first term of eq. (6.17) is fixed for any given application since it is dependent on L and m. $V_s/2^m$ is computed by performing the simple logical m–bit right shift operation on V_s. Hence an approximate value for C_f can quickly be evaluated by using only two subtract operations and an m–bit shift operation

Practical experience in the application of eq. (6.17) has shown that $m = 3$ is an optimum value when $L = 8$. Other values of m give rise to larger errors of C_f over a 25% variation in supply voltage.

Table 6.9 shows the error of approximation of eq. (6.17) for $m = 3$ and $L = 8$. The values of V_s shown represent the unscaled binary numbers produced by an 8–bit A/D converter. For a 25% variation in V_s the maximum error in computing C_f is less than 1.2%. The speed advantage gained in using eq. (6.17) is significantly more than the negative effects of any errors incurred in computing C_f.

A block diagram showing a PWM modulator using eq. (6.17) is shown Fig. 6.3.

It should be noted that an [I] matrix modulator as shown in Fig. 6.3 can perform the same function as the SVM modulator described in chapter 5 (and hence allowing a choice of modulation strategies for all applications). "k" or δ are computed according to eq. (5.10) and V_f is proportional to the magnitude of the vector given in eq. (5.1).

Table 6.9

Error in Calculating C_f Using Eq. (6.17)

V_s	$C_f = V_r/V_s$	Error of Eq. (6.17)
192	1.167	1.22%
196	1.143	0.71%
200	1.120	0.25%
204	1.098	–0.14%
208	1.077	–0.47%
212	1.057	–0.74%
216	1.037	–0.95%
220	1.018	–1.09%
V_r = 224	1.000	–1.17%
228	0.983	–1.19%
232	0.966	–1.14%
236	0.949	–1.04%
240	0.933	–0.87%
244	0.918	–0.63%
248	0.903	–0.33%
252	0.889	0.02%
256	0.882	0.23%

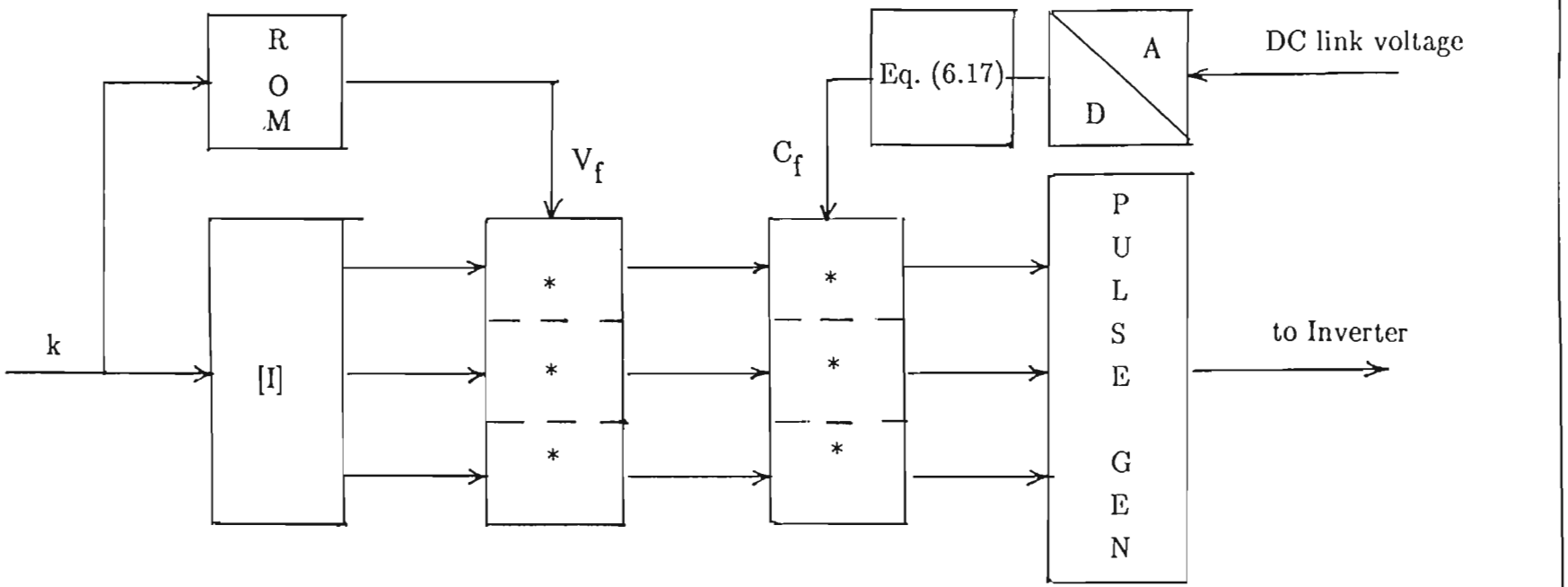


Fig. 6.3 Diagram of Compensated Pulse Width Modulator

6.9 Conclusions

It was shown in Table 6.4 that a minimum word length of 12–bits is required for a successful PWM modulator to minimize errors and to reduce harmonics. The minumum size of microprocessor that could be used is 16 bits, because an 8–bit microprocessor has a very limited speed range. Harmonics are further reduced by using a double sided pulse width modulation technique.

Maximum and minimum values of δ have been derived, and it has been shown that at the extreme values, high amplitude errors and frequency errors occur. The errors are minimised by using the maximum value of U_0 possible without causing arithmetic overflow.

Constant V/f (flux control) can be achieved by using additional multiply operations at the end of each step to control the depth of modulation and hence the output voltage to a motor. Supply voltage variations can also be reduced by multiplying the computed pulse widths by a correction factor C_f which is calculated according to eq. (6.17).

The algorithm is limited to N–odd phase oscillator matrices, although controllers producing an even number of phases can be implemented. By computing the two's complement of each phase, or for bridge inverters, by simply altering the wiring to the bridge driver circuits, a system with an even number of phases can be realised.

It is possible to include the controller in a closed loop control system. By adjusting the value of k to increase or decrease the frequency, the speed of a motor can be held at a constant speed despite variations in the slip frequency.

CHAPTER SEVEN

EIGHT AND SIXTEEN BIT PWM MODULATORS

7.1 Eight Bit 8085 PWM Modulator

During the early stages of the investigation reported in this thesis , it became necessary to varify parts of the theoretical analysis. The only available equipment was an INTEL SDK microcomputer development board, with an 8–bit 8085 microprocessor and various peripheral components.

Early attempts to prove the validity of the theorems involved the emulation of 8, 12 and 16 bit arithmetic systems. This was extremely time–consuming and it proved difficult to observe the performance of the matrices for extended periods of time.

The system shown in Fig 7.1 was developed, with eqs. (2.14) and eqs. (3.14) (the [T] and [I] matrices) programmed into two different sections of the ROM memory. Trailing edge regular PWM sampled pulses were produced by the 8253 Timer. A D/A converter was added to the system to provide a simple display of the values of the pulse widths, and the function produced by the algorithm under test.

The main purpose of the experiments was to establish the stability, and quality of the waveforms produced by the algorithms described in Chapters 2 and 3. The output of one of the phases was connected to a real–time FFT analyser in order to measure the harmonics of the PWM pulses.

7.2 Results

Stable oscillation occurred for values of δ less than 0.75. For values of $\delta > 0.75$ the distortion was so large that it was impossible to recognise any reptitive waveform or assess the stability of oscillation. After the harmonic analysis had been carried out it was clear that $\delta_{max} = 0.314$ was the largest acceptable value. The program was improved by including code to detect pulse width overflow and using as large a value of the vector length U_o as possible.

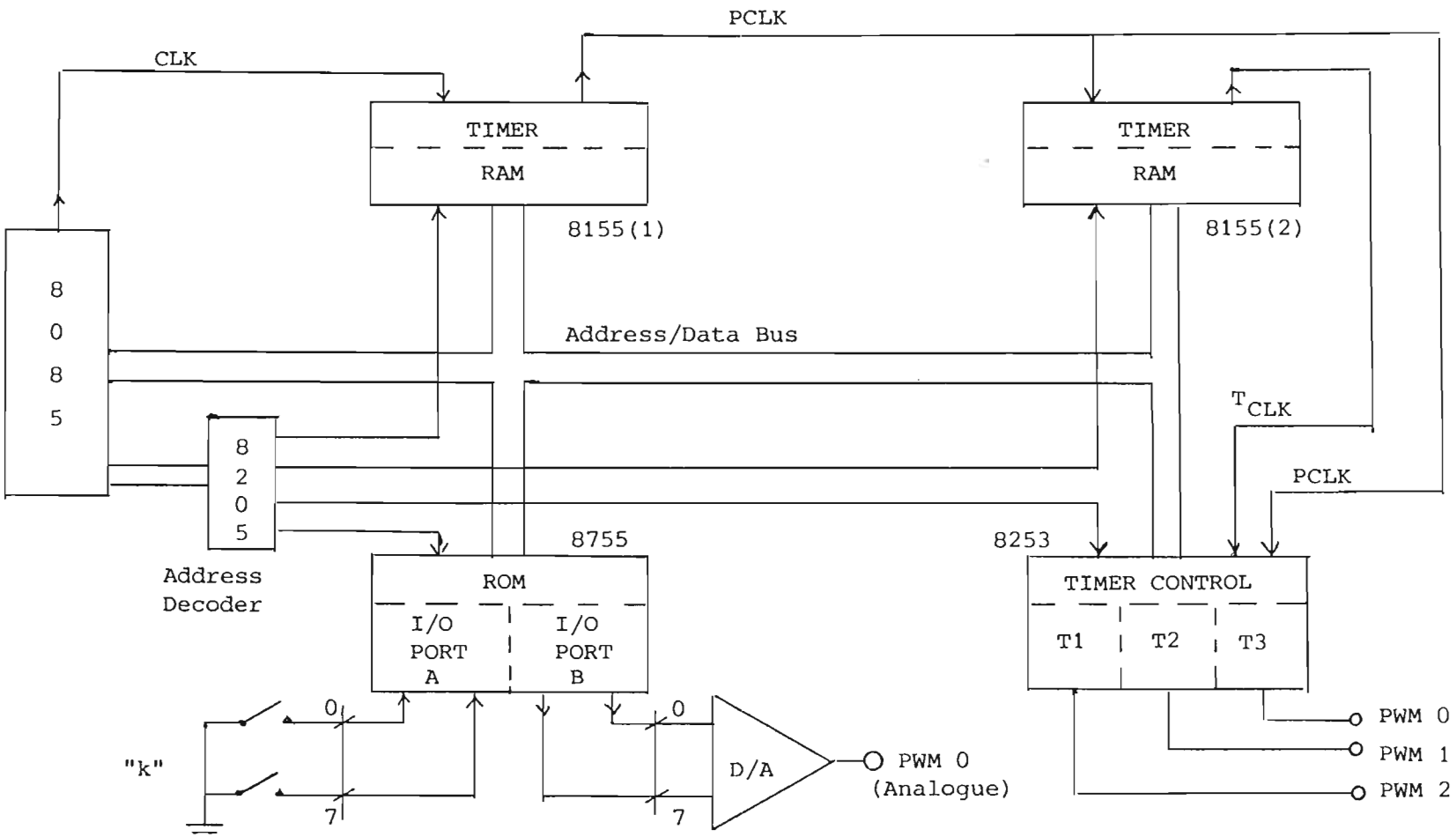


Fig. 7.1 Eight Bit PWM Modulator

The preceding set of tests were repeated using the revised program and the harmonic analysis results are given in Tables 6.1 and 6.2 The [T] and [I] oscillators were stable and functioned correctly for long periods of time with different values of δ. The frequency was changed at random and stable oscillation was maintained.

The time taken to evaluate eqs. (3.14) or one complete 3–phase calculation was approximately 430 μs, i.e. a carrier frequency of 2310 Hz. The range of output frequencies was from 12.6 Hz to 116 Hz in 41 steps, the lower limit being set by the precision attainable in an 8–bit configuration. No attempt was made to implement a variable amplitude PWM modulator because of the long multiplication times.

Fig 7.2 is a photograph of the three PWM (trailing–edge modulation) phases, and the analogue output of Phase 0 showing the variation in pulse widths with the correct phase relationship.

7.3 Conclusions of the 8–bit Modulator

The tests conducted on the 8085 PWM modulator showed that stable oscillation at different frequencies could be achieved over extended periods of time. The spectral analysis showed that harmonics were present but at a low level.

The limited speed range and long sampling interval proved that it was essential to use a 16–bit microcomputer to realise the full advantages of the algorithm.

7.4 Sixteen–Bit 8088 PWM Modulator

The knowledge gained in the development of the 8085 modulator led to the development of an 8088 16–bit PWM modulator, which incorporated a number of enhancements. In addition to improvements in the program, a double sided PWM pulse generator was developed to reduce harmonics.

A prototype 8088 single–board computer with a few additional components was used to implement the 16–bit modulator. The 8088 microprocessor has eight general purpose registers, and by careful optimisation of the assembler code, a fast and efficient implementation of the [I] matrix using eqs. (3.14) was realised. Where possible, the fast register instructions were used, and only one relatively slow data access instruction was needed. The major disadvantage of the 8088 was the slow muliply opcode, and the only

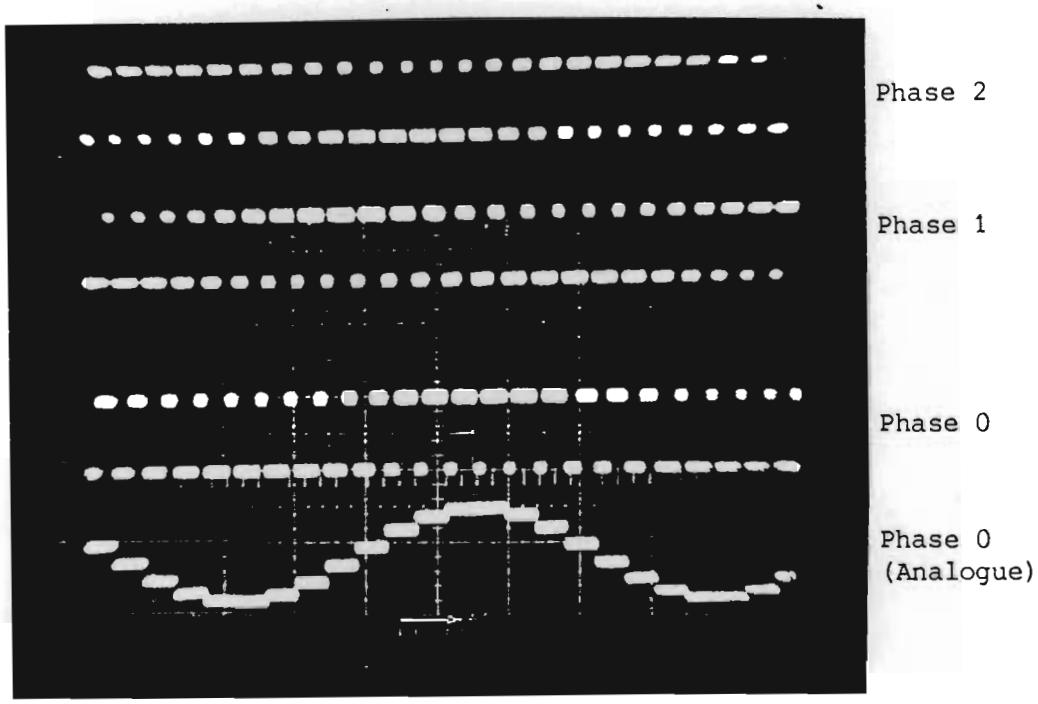


Fig. 7.2(a) Eight Bit PWM Modulator Output (k=0.21)

Horizontal Scale 1ms/div Vertical Scale 5V/div

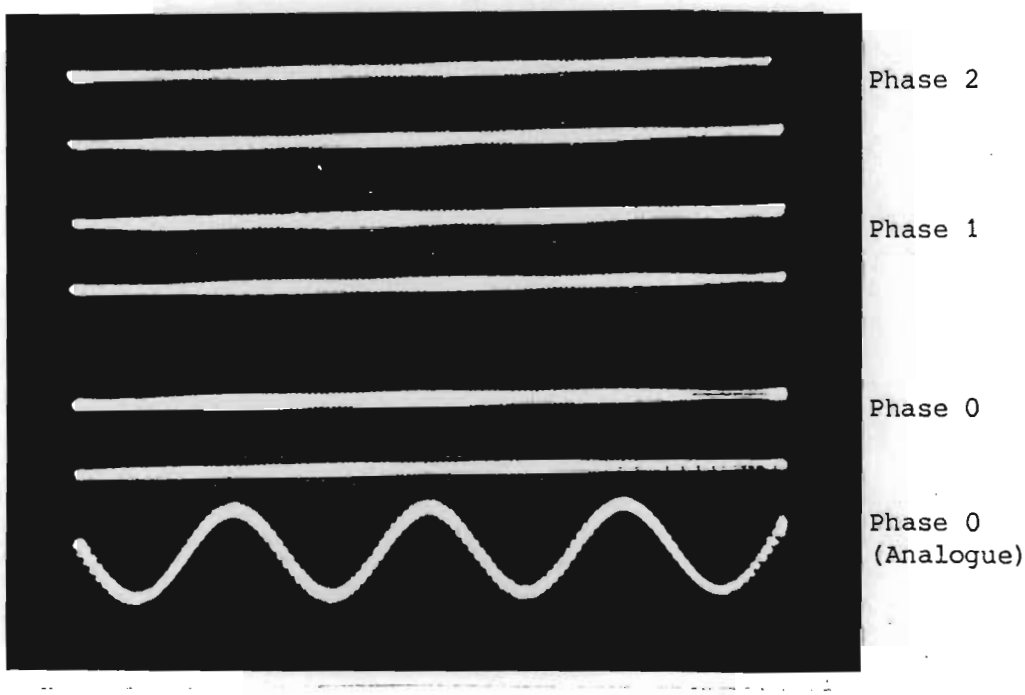


Fig. 7.2(b) Eight Bit PWM Modulator Output (k=0.11) ,

Horizontal Scale 5ms/div Vertical Scale 5V/div

way that the calculation time for each sampling interval could be reduced to less than 50 μs was to use an 80188 which had both a higher clock–speed and a faster multiply algorithm.

All the hardware for a complete 3–phase PWM inverter was built and tested. Fig. 7.3 shows the block diagram of the 8088 16–bit PWM modulator. A description of the double sided pulse generator is given in Appendix E.

7.5 Amplitude Control Table

The Voltage/frequency (V/f) characteristic of an induction motor is a non–linear function, and the simplest method of implementing such a characteristic is to use a V/f look–up table. The magnitude of the amplitude control variable was obtained from the V/f look–up table using the 8–bit value of k as an index variable.

The shape and the parameters used for constructing the V/f table are given in Appendix F. The graphs of the calculated V/f characteristic and the measured V/f characteristic are similar as shown in Fig 7.4.

7.6 80188 PWM Modulator

One of the objectives of the development of the digital modulator was to produce a system with a 20 kHz carrier frequency. Practical measurements on the 8088 microcomputer showed that the minimum calculation time for eqs. (3.14) was 426 μs. The time for an "end of loop test" was approximately 12 μs. To allow some margin of error it was necessary to allow for two "end of sequence loops" which gave a total calculation time of 460 μs or a carrier frequency of 2.18 kHz.

A detailed investigation of the time taken to complete one step showed that 48% of the time was spent in performing the six multiplications; three 16–bit multiplications to evaluate eqs. (3.14) and three 8–bit multiplications for the voltage control variaable. (Approximately 45 μs was required for each for 16–bit multiply). The enhanced version of the 8088, the 80188, offered a significantly improved multiply procedure (which only needed a fifth of the time) and a much higher clock frequency (24 MHz versus 8 MHz). It was not possible to purchase an 80188 but a realtime emulation facility was available at the agents for Intel products. Version 1 of the 8088 program was run on the 80188

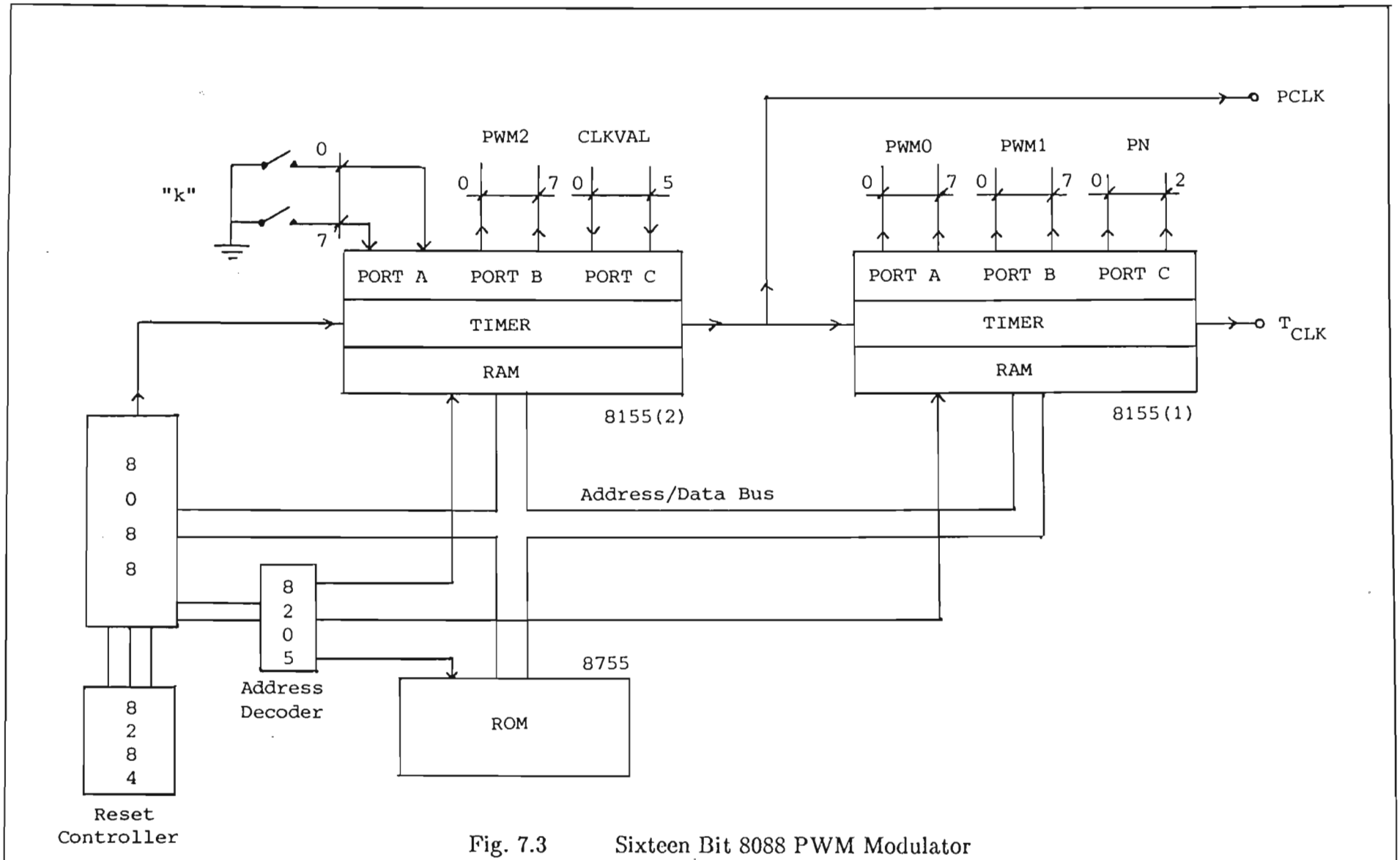


Fig. 7.3 Sixteen Bit 8088 PWM Modulator

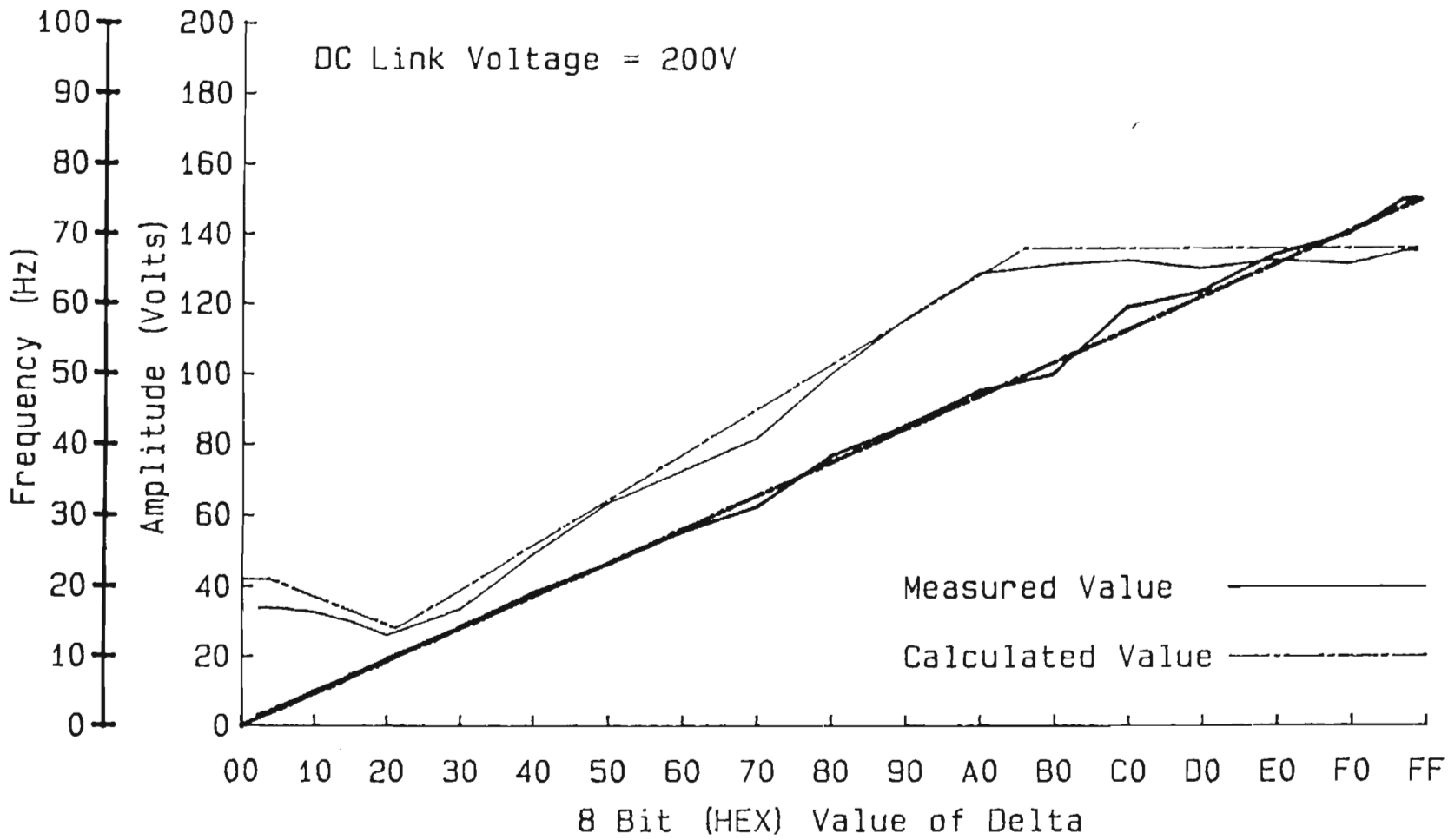


Fig. 7.4 Comparison of Frequency and Output Voltage Characteristics

emulator and the state count and timing calculations were verified and are listed in column three of Table 7.1. There was no "end of loop" sequence check and a carrier frequency of 21.7 kHz was produced.

Later enchanced versions of the 8088 program reduced the number of MOV instructions and eliminated a number of other instructions. Timing measurements using an HP 64000 emulation analyser confirmed the state count of 1229 states and a calculated time of 459 μs (measured time 460 μs) as shown in column one of Table 7.1. The expected time for the evalution of eqs. (3.14) for the improved program on the 80188 was calculated and is given in column two of Table 7.1. The "end of sequence loop" was also included in this calculation. The results show that an 80188 could produce 3–phase PWM pulses with a carrier frequency of 20 kHz.

Recent developments with high speed digital signal processing microcomputers (e.g. TMS 32010) indicate that calculation times could be reduced to less than 30 μs, i.e. carrier frequencies of more than 30 kHz. In this respect eqs. (3.14) were implented on a T414 Transputer and a calculation time of 40 μs was achieved. Thus it should be possible to construct high speed PWM modulators with a range of suitable microprocessors.

Investigations with an IC manufacturer using an uncommitted logic array of 2500 gates revealed that it would be possible to produce a modulator on a single chip using approximately 1800 gates. The design included the three pulse generators and V/f look–up table. Emulation studies of the chip showed that calculation times of 10 μs could be realised, which would enable a potential carrier frequency of 100 kHz to be achieved.

Table 7.1

Calculation of Times to Evaluate Eq. (3.14)

	8088	80188	80188 (Version) (one)
Number of states for eq. (3.14)	368 * 3 (1104)	105 * 3 (315)	115 * 3 (345)
Number of states for I/O	65	34	23
Number of states for test at end	30 (60)	27 (54)	0
Total No. of states	1229	403	368
Clock frequency (*3)	8 MHz	24 MHz	24 MHz
Time	459 μs	50 μs	46 μs
Carrier Frequency	2180 Hz	20 kHz	21.7 kHz

7.7 Conclusions of the 16–bit Modulator

The tests conducted on the 16–bit 8088 PWM modulator showed that it was possible to produce acceptable 3–phase PWM pulses using digital techniques. The predicted frequency/amplitude curve was closely realised in practice. Harmonics and sub–harmonics were present in the output waveform, however they were at an acceptably low level.

Investigations with an 80188 microcomputer proved that a 20 kHz digital PWM modulator could be constructed. With modern single–chip microcomputers and uncommitted logic arrays it should be possible to build PWM modulators that operate at higher carrier frequencies.

The results of the experiments and investigations of this chapter proved that it is feasible to design a digital PWM modulator which exceeds the performance of analogue or look–up table based modulators.

CHAPTER EIGHT

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK

8.1 Conclusions

This thesis has described the development of a digital multiphase PWM modulator. The effects of errors have been analysed, enabling the frequency range of a given system to be predicted. Formulae for estimating the maximum amplitude error that could occur have been developed, and practical experiments have shown that under normal conditions the maximum error is unlikely to be exceeded.

From the theoretical analysis, the design criteria for practical PWM modulator were developed. Early implementations using 8085 microcomputers confirmed the theoretical predictions. Developments with 16–bit 8088 and 80188 microcomputers showed that it was possible to design and build a 3–phase PWM modulator with an output frequency range of 1000:1 and a carrier frequency of 20 kHz. The simple structure of the digital modulator, and the minimal number of arithmetic operations, ensured that the entire modulator could be realised in a single integrated circuit. Other forms of digital oscillator matrices were investigated but were rejected in favour of the simple, easily implemented Simplex [I] digital oscillator equation.

8.2 Recommendations for Further Work

8.2.1 Digital oscillator matrix for an even number of phases

The implementation of the 2–phase oscillator matrix has been described, but no single oscillator matrix for 2^p phases has been discovered. It should be noted that digital modulators for an even number of phases, up to and including 14 may be produced using indirect methods as described in section 6.6.

Investigations into the development of a matrix for an even number of phases have been carried out and an equation using N/2 of the phases has been derived. Stable oscillation was not achieved because only one of the criteria for stability was satisfied. Further investigation into different types of equations is required before it will be possible to produce a general digital oscillator matrix for an even number of phases.

8.2.2 Variable pulse spacing

The analysis of the PWM signals in this thesis has assumed a constant time interval between the centres of the double–edged PWM pulses. It is possible to create a constant V/f characteristic by generating the same sequence of PWM pulses and varying the time interval between each pulse. i.e. a varible–width variable–delay pulse sequence.

An additional timer/counter could be added to the PWM pulse generator and the value loaded into this counter used to control the time interval between each pulse. The double–edged pulse modulator would have to be modified to generate a PWM Pulse only when activated by the pulse interval timer. Other minor modifications to the program would be required to control the pulse interval timer. It may be possible to use the above concept to control the amplitudes of the phase voltages, instead of performing additional multiply operations. This technique could simplify the design of an IC. It could also offer a faster execution time for a microprocessor program. Further research on the harmonic content would be required to confirm that no undesirable harmonics would be produced.

8.2.3 Integrated circuit PWM modulator

The IC design that was developed to verify the performance of a one chip PWM modulator was based on standard, predefined logic arrays, with simple state controllers. The design that was tested was not very efficient in a few critical areas, notably the multiplier. Further consideration could be given to designing a more efficient multiple–state controller to reduce the calculation time by employing parallel processing techniques.

Additional facilities, such as current limiting need to be investigated and if necessary included in the design or specification of PWM modulators.

8.2.4 Inverter circuit design

Experience with the design and construction of the inverter circuits shows that there is scope for research into the performance and design of high speed switching circuits. The characteristics of high–speed switching elements and their associated driver circuits need to be enhanced. The performance and design of snubber and filter circuits using non–linear devices is another area for further investigation.

8.2.5 Stability criteria and Liapunov's direct method

It was not possible to rigorously determine the stability of the [I] matrix from the eigenvalues due to the lack of a set of transformation equations for a third order critical system. The derivation of the transformation equations would enable a more thorough analysis to be carried out.

8.3 Summary

The objectives defined in Section 1.3 have been achieved, and the characteristics of a digital PWM modulator have been established. The permissable output frequency range and maximum carrier frequencies are far in excess of the capabilities of the majority of switching devices available at present, furthermore the output frequency range is much larger than the speed range of any induction motor.

The introduction of sub–harmonics and harmonics due to non–integer ratios of modulating and carrier frequencies have been shown to be within acceptable limits, and by using sufficiently high carrier frequencies undesirable harmonics are reduced to sufficiently low amplitudes.

A practical application of the 3–phase PWM inverter showed that it could be used to control a fractional–horsepower motor. The development of larger and more powerful inverters would enable larger motors to be controlled.

APPENDIX A

PROOF OF PHASE SUMMATION FORMULAE

Eq. (A.1.1) gives the relationship between $\mathrm{Cos}\phi$ and the sum of the magnitudes of the remaining pulse widths.

It is required to prove that

$$A.\mathrm{Cos}\phi = \sum_{n=1}^{N-1} (-1)^{n+1}.\mathrm{Sin}(\phi+n\theta) \tag{A.1.1}$$

Expanding the right hand side yields

$$A.\mathrm{Cos}\phi = \mathrm{Sin}(\phi+\theta)-\mathrm{Sin}(\phi+2\theta)+ \ldots . -\mathrm{Sin}(\phi+(N-1)\theta) \tag{A.1.2}$$

$$= \mathrm{Sin}\phi.\mathrm{Cos}\theta+\mathrm{Cos}\phi.\mathrm{Sin}\theta - \mathrm{Sin}\phi.\mathrm{Cos}2\theta-\mathrm{Cos}\phi.\mathrm{Sin}2\theta + - \mathrm{Sin}\phi.\mathrm{Cos}(N-1)\theta)-\mathrm{Sin}(N-1)\theta).\mathrm{Cos}\phi \tag{A.1.3}$$

collecting like terms

$$A.\mathrm{Cos}\phi = \mathrm{Sin}\phi.[\mathrm{Cos}\theta - \mathrm{Cos}2\theta + \mathrm{Cos}3\theta \ldots \mathrm{Cos}(N-1)\theta] + \mathrm{Cos}\phi.[\mathrm{Sin}\theta - \mathrm{Sin}2\theta + \mathrm{Sin}3\theta \ldots \mathrm{Sin}(N-1)\theta]$$

$$A.\mathrm{Cos}\phi = \mathrm{Sin}\phi.\Sigma(-1)^{n+1}.\mathrm{Cos}(n\theta) + \mathrm{Cos}\phi.\Sigma(-1)^{n+1}.\mathrm{Sin}(n\theta) \tag{A.2}$$

From eq. (A.2) it is required to prove that

$$\sum_{n=1}^{N-1} (-1)^{n+1}.\mathrm{Cos}(n\theta) = 0 \tag{A.3.1}$$

$$\sum_{n=1}^{N-1} (-1)^{n+1}.\mathrm{Sin}(n\theta) = A \tag{A.3.2}$$

Using the Euler Identity and combining eq. (A.3.1) and eq. (A.3.2) yields

$$\sum_{n=1}^{N-1} (-1)^{n+1}.[\mathrm{Cos}(n\theta) + j.\mathrm{Sin}(n\theta)] = \sum_{n=1}^{N-1} (-1)^{n+1}.e^{jn\theta} \qquad (A.4)$$

The r.h.s is reduced by applying the geometric summation identity

$$S_n = \frac{a.(1-r^n)}{1-r} \quad \text{with} \quad a = e^{j\theta} \quad \text{and} \quad r = -e^{j\theta}$$

hence

$$\sum_{n=1}^{N-1} (-1)^{n+1}.e^{jn\theta} = \frac{e^{jn\theta}\left[1 - (-1)^{N-1}.e^{j(N-2)\theta}\right]}{1 + e^{j\theta}}$$

dividing through by $e^{j\theta/2}$ and multiplying out yields

$$\sum_{n=1}^{N-1} (-1)^{n+1}.e^{jn\theta} = \frac{e^{j\theta/2} - (-1)^{N-1}.e^{j(2N-1)\theta/2}}{e^{-j\theta/2} + e^{j\theta/2}}$$

now $e^{jN\theta} = 1$ and hence $e^{j(2N-1)\theta/2} = e^{-j\theta/2}$

thus

$$\sum_{n=1}^{N-1} (-1)^{n+1}.e^{jn\theta} = \frac{e^{j\theta/2} - (-1)^{N-1}.e^{j\theta/2}}{e^{-j\theta/2} + e^{j\theta/2}}$$

for N odd

$$\sum_{n=1}^{N-1} (-1)^{n+1}.e^{jn\theta} = j.\frac{\mathrm{Sin}(\theta/2)}{\mathrm{Cos}(\theta/2)}$$

hence $$\sum_{n=1}^{N-1} (-1)^{n+1}.\mathrm{Sin}(n\theta) = \tan(\theta/2) = A \qquad (A.5.1)$$

and $$\sum_{n=1}^{N-1} (-1)^{n+1}.\mathrm{Cos}(n\theta) = 0 \qquad (A.5.2)$$

Substituting eq. (A.5.1) and eq. (A.5.2) into eq. (A.1.2)

$$\mathrm{Cos}\phi.\tan(\theta/2) = \mathrm{Sin}(\phi+\theta) - \mathrm{Sin}(\phi+2\theta) + \ldots - \mathrm{Sin}(\phi+(N-1)\theta) \qquad (A.6)$$

For N even

$$\sum_{n=1}^{N-1} (-1)^{n+1}.\mathrm{Sin}(n\theta) = 0$$

and $$\sum_{n=1}^{N-1} (-1)^{n+1}.\mathrm{Cos}(n\theta) = 1$$

and therfore cannot be used in eq. (A.1.2)

A similar result is obtained using the cosine summation formula

It is required to prove that

$$B.\mathrm{Sin}\phi = \mathrm{Cos}(\phi+\theta) - \mathrm{Cos}(\phi+2\theta) + \ldots - \mathrm{Cos}(\phi+(N-1)\theta) \qquad (A.7)$$

expanding and collecting like terms yields

$$B.\mathrm{Sin}\phi = \mathrm{Cos}\phi.\Sigma(-1)^{n+1}.\mathrm{Cos}(n\theta) + \mathrm{Sin}\phi.\Sigma(-1)^{n+1}.\mathrm{Sin}(n\theta) \qquad (A.8)$$

Substituting eqs. (A.5) into eq. (A.8) yields

$$B.\mathrm{Sin}\phi = \mathrm{Cos}\phi.0 - \mathrm{Sin}\phi.\tan(\theta/2) \qquad (A.9)$$

and hence $B = -\tan(\theta/2)$ for N–odd only

substitutuing $B = -\tan\theta$ into eq. (A.7) yields

$$-\tan(\theta/2).\mathrm{Sin}\phi = \mathrm{Cos}(\phi+\theta) - \mathrm{Cos}(\phi+2\theta) + \mathrm{Cos}(\phi+3\theta) - \ldots - \mathrm{Cos}(\phi+(N-1)\theta) \qquad (A.10)$$

Eq. (A.6) and eq. (A.10) prove that it possible to compute the magnitude of one pulse width from the widths of all the other pulses in an odd numbered multiphase PWM modulator.

APPENDIX B.

DERIVATION OF [I] AND [G] MATRICES

B.1 Introduction

In Chapter 3 the equations for a stable 3–phase oscillator matrix were derived. This appendix shows how the [I] and [G] matrices are derived from eqs. (3.14) and eqs. (3.20) respectively.

It is also proven that the [I] matrix satisfies the two conditions for stable oscillation.

B.2 Derivation of the [I] Matrix

The equations derived in chapter 3 for the [I] matrix are

$$x_1(n+1) = x_1(n) + k[x_2(n) - x_3(n)] \tag{3.14.1}$$

$$x_2(n+1) = x_2(n) + k[x_3(n) - x_1(n+1)] \tag{3.14.2}$$

$$x_3(n+1) = x_3(n) + k[x_1(n+1) - x_2(n+1)] \tag{3.14.3}$$

Substituting eq. (3.14.1) into eq. (3.14.2)

$$\begin{aligned} \mathrm{Sin}_2(n+1) &= \mathrm{Sin}_2\phi + k.\mathrm{Sin}_3\phi - k[\mathrm{Sin}_1\phi + k(\mathrm{Sin}_2\phi - \mathrm{Sin}_3\phi)] \\ &= (1-k^2)\mathrm{Sin}_2\phi + (k+k^2)\mathrm{Sin}_3\phi - k.\mathrm{Sin}_1\phi \end{aligned} \tag{B.1}$$

Substituting eq. (B.1) and eq. (3.14.1) into eq. (3.14.3) yields

$$\begin{aligned} \mathrm{Sin}_3(n+1) = \mathrm{Sin}_3\phi &+ k[\mathrm{Sin}_1\phi + k[\mathrm{Sin}_2\phi - \mathrm{Sin}_3\phi)] - \\ &- k[-k.\mathrm{Sin}_1\phi + (1-k^2)\mathrm{Sin}_2\phi + (k-k^2)\mathrm{Sin}_3\phi] \end{aligned} \tag{B.2}$$

multiplying out and collecting like terms

$$\mathrm{Sin}_3(n+1) = (k+k^2)\mathrm{Sin}_1\phi + (-k+k^2+k^3)\mathrm{Sin}_2\phi + (1-2k^2-k^3)\mathrm{Sin}_3\phi \tag{B.3}$$

hence

$$[I] = \begin{bmatrix} 1 & -k & k+k^2 \\ k & 1-k^2 & -k+k^2+k^3 \\ -k & k+k^2 & 1-2k^2-k^3 \end{bmatrix} \tag{B.4}$$

It should also be noted that [I] can be derived directly by noting that

$$I_{1j} = (1, k, -k)$$

$$I_{2j} = [0, 1, k] - k.I_{1j}$$

$$I_{3j} = [0, 0, 1] + k.I_{1j} - k.I_{2j}$$

since I_{1j} defines $Sin_1(n+1)$ and I_{2j} defines $Sin_2(n+1)$.

B.3 Stability Conditions

The proof that $Det[I] = 1$ is shown by reducing the [I] matrix, or by applying the definition of the determinant directly.

The latter method is used, since the former method is explained in Section 3.7.

Taking the top row of [I] and expanding the 2 * 2 subdeterminants

$$\begin{aligned} Det[I] = 1.[(1-k^2)\,(1-\,2k^2-k^3) - (k+k^2)\,(-k+k_2+k^3)] - \\ - (-k).[k(1-2k^2-k^3) - (-k)\,(-k+k^2+k^3)] + \\ + (k+k^2).[k(k+k^2) - (-k)\,(1-k^2)] \end{aligned}$$

$$\begin{aligned} = 1 - 2k^2 - k^3 + \\ + k^2 - k^3 - k^4 + \\ + k^2 + 2k^3 + k^4 \end{aligned}$$

$$Det[I] = 1$$

For the second condition for stable oscillation, it can be seen from eq. (B.4) that the sum of each column is unity.

Thus eqs. (3.14) and the [I] matrix form a stable 3–phase digital oscillator matrix for values of $|k| \leq 1$ (where the limit $|k| \leq 1$ is due to error propagation.)

B.4 Derivation of the [G] Matrix and Determinant

In section 3.6 the equations for the Gain or [G] oscillator matrix are developed,

$$g.Sin_1(n+1) = g.Sin_1\phi + k[Sin_2\phi - Sin_3\phi] \tag{3.20.1}$$

$$g.Sin_2(\phi+\delta) = g.Sin_2\phi + k[Sin_3\phi - g.Sin_1(\phi+\delta)] \tag{3.20.2}$$

$$g.Sin_3(\phi+\delta) = g.Sin_3\phi + k[g.Sin_1(\phi+\delta) - g.Sin_2(\phi+\delta)] \tag{3.20.3}$$

Substituting eq. (3.20.1) into eq. (3.20.2) and collecting like terms

$$\begin{aligned} g.Sin_2(\phi+\delta) &= g.Sin_2\phi + k[Sin_3\phi - k.g.Sin_1\phi - k^2.Sin_2\phi + \\ &\qquad\qquad k^2.Sin_3\phi \\ &= g.k.Sin_1\phi + (g-k^2)Sin_2\phi + (k+k^2)Sin_3\phi \end{aligned} \tag{B.5}$$

Substituting eq. (B.5) and eq. (3.20.1) into eq. (3.20.3) and collecting like terms

$$\begin{aligned} g.Sin_3(\phi+\delta) &= g.Sin_3\phi + gk.Sin_1\phi + k^2.Sin_2\phi - k^2.Sin_3\phi + \\ &\qquad gk^2.Sin_1\phi - (-gk+k^2)Sin_2\phi - (k^2+k^3)Sin_3\phi \\ &= (gk+gk^2)Sin_1\phi + (-gk^2+k^3)Sin_2\phi + \\ &\qquad +(g-2k^2-k^3))Sin_3\phi \end{aligned} \tag{B.6}$$

hence the [G] matrix is

$$[G] = \begin{bmatrix} g & -gk & gk+gk^2 \\ k & gk^2 & -gk+k^2+k^3 \\ -k & k+k^2 & g-2k^2-k^3 \end{bmatrix} \tag{B.7}$$

To evaluate Det[G]

let $G'_{1j} = G_{1j}$

and $G'_{2j} = G_{2j} + k.G_{1j}$

and $G'_{3j} = G_{3j} - k.G_{1j} + k.G_{2j}$

Hence $$[G'] = \begin{bmatrix} g & 0 & 0 \\ k & g & 0 \\ -k & k & g \end{bmatrix}$$

thus $$\text{Det}[G] = \text{Det}[G'] = g^3$$

The implication of the above proof is that changes in vector length (g) are independent of frequency (k). The significance of this result is that pulse width and frequency errors may be analysed independently.

APPENDIX C

OSCILLATOR MATRIX PROGRAM LISTINGS

A general purpose plotter program was written for a HP 86 computer, and a unique subroutine was written for each matrix. The listings given below are extracts from each subroutine and show the ease of implementation, and subtle differences in program structure for the different matrices. The variable D represents δ.

C.1 Program for Fig. 2.1 [O] Matrix

```
200   C = COS(D) @ S = SIN(D)
210   FOR I = 1 TO 400
220   T1 = Y1 * C + Y2 * S
230   Y2 = Y2 * C - Y1 * S
240   Y1 = T1
```

C.2 Program for Fig. 2.2 [U] Matrix

```
200   FOR I = 1 TO 400
210   T1 = Y1 + D * Y2
220   Y2 = Y2 - D * Y1
230   Y1 = T1
```

C.3 Program for Fig. 2.3 [T] Matrix

```
200   FOR I = 1 TO 400
210   Y1 = Y1 + D * Y2
220   Y2 = Y2 - D * Y1
```

C.4 Program for Fig. 3.1 [H] Matrix

```
200  C = COS(D) @ S = SIN(D)/SQR(3)
210  FOR I = 1 TO 400
220  T1 = Y1 * C + S * (Y2 - Y3)
230  T2 = Y2 * C + S * (Y3 - Y1)
240  Y3 = Y3 * C + S * (Y1 - Y2)
250  Y1 = T1 @ Y2 = T2
```

C.5 Program for Fig. 3.2 An Unstable Matrix

```
200  K = D/SQR(3)
210  FOR I = 1 TO 400
220  T1 = Y1 + K * (Y2 - Y3)
230  T2 = Y2 + K * (Y3 - Y1)
240  Y3 = Y3 + K * (Y1 - Y2)
250  Y1 = T1 @ Y2 = T2
```

C.6 Program for Fig. 3.3 [I] Matrix

```
200  K = D/SQR(3)
210  FOR I = 1 TO 400
220  Y1 = Y1 + K * (Y2 - Y3)
230  Y2 = Y2 + K * (Y3 - Y1)
240  Y3 = Y3 + K * (Y1 - Y2)
```

C.7 Program for Fig. 3.5 [G] Matrix

```
200  K = D/SQR(3) @ G = 0.99
210  FOR I = 1 TO 400
220  Y1 = Y1 * G + K * (Y2 - Y3)
230  Y2 = Y2 * G + K * (Y3 - Y1)
240  Y3 = Y3 * G + K * (Y1 - Y2)
250  IF I=200 THEN G = 1/G
```

C.8 Program for Fig. 3.6 [G] Matrix: Step Function

```
200  K = D/SQR(3) @ G = 1
210  FOR I = 1 TO 400
220  IF I=200 THEN G = 0.1
230  Y1 = Y1 * G + K * (Y2 - Y3)
240  Y2 = Y2 * G + K * (Y3 - Y1)
250  Y3 = Y3 * G + K * (Y1 - Y2)
260  G = 1
```

C.9 Program for the [F] Matrix

```
200  L = D/TAN(PI/5)
210  FOR I = 1 TO 400
220  Y1 = Y1 + L * (Y2 - Y3 + Y4 - Y5)
230  Y2 = Y2 + L * (Y3 - Y4 + Y5 - Y1)
240  Y3 = Y3 + L * (Y4 - Y5 + Y1 - Y2)
250  Y4 = Y4 + L * (Y5 - Y1 + Y2 - Y3)
260  Y5 = Y5 + L * (Y1 - Y2 + Y3 - Y4)
```

APPENDIX D

PROOF OF ERROR SUMMATION FORMULAE

D.1 Introduction

The error introduced at each step is approximately equal to $\frac{\delta^2}{2}.x_1(n)$ and $\frac{\delta^2}{2}.x_2(n)$ and the sum of these terms over half a cycle or M/2 steps give an estimate of the upper bound for the maximum change in the length of the vector(s). i.e.

$$\epsilon_{max} = \sum_{n=0}^{M/2} \frac{\delta^2}{2}.\mathrm{Cos}(-\pi/2 + n\delta) \qquad M = \frac{2\pi}{\delta} \tag{D.1}$$

$$\text{or} \qquad \epsilon_{max} = \sum_{n=0}^{M/2} \frac{\delta^2}{2}.\mathrm{Sin}(n\delta) \tag{D.2}$$

D.2 Proof

Assume an arbitrary angle ϕ and substitute into eq. (D.1)

$$\epsilon_{max} = \sum_{n=0}^{M/2} \frac{\delta^2}{2}.\mathrm{Cos}(\phi + n\delta)$$

now

$$\Sigma\,\mathrm{Cos}(\phi+n\delta) = \Sigma[\mathrm{Cos}\phi.\mathrm{Cos}(n\delta) - \mathrm{Sin}\phi.\mathrm{Sin}(n\delta)] \tag{D.3}$$

and therfore it is required to find

$$\sum_{n=0}^{M/2} \mathrm{Cos}(n\delta) \quad \text{and} \quad \sum_{n=0}^{M/2} \mathrm{Sin}(n\delta) \tag{D.4}$$

Using the Euler Identity and combining eq. (D.4) yields

$$\sum_{n=0}^{M/2} [\mathrm{Cos}(n\delta) + j.\mathrm{Sin}(n\delta)] = \sum_{n=0}^{M/2} e^{jn\delta} \tag{D.5}$$

The r.h.s is reduced by applying the geometric summation identity

$$S_n = \frac{a.(1-r^n)}{1-r} \quad \text{with} \quad a = 1 \quad \text{and} \quad r = e^{j\delta}$$

hence

$$\sum_{n=0}^{M/2} e^{jn\delta} = \frac{1.\left[1 - e^{j(M/2+1)\delta}\right]}{1 - e^{j\delta}}$$

multiplying through by $e^{-j\delta/2}$ yields

$$\sum_{n=0}^{M/2} e^{jn\delta} = \frac{e^{-j\delta/2} - e^{j(M+1)\delta/2}}{e^{-j\delta/2} - e^{j\delta/2}}$$

now $e^{j(M+1)\delta/2} = e^{jM\delta/2}.e^{j\delta/2}$

and $e^{jM\delta/2} = e^{j\pi} = -1$

thus

$$\sum_{n=0}^{M/2} e^{jn\delta} = \frac{e^{-j\delta/2} - (-1).e^{j\delta/2}}{e^{-j\delta/2} - e^{j\delta/2}}$$

$$\sum_{n=0}^{M/2} e^{jn\delta} = \frac{2.\mathrm{Cos}(\delta/2)}{-2j.\mathrm{Sin}(\delta/2)} \qquad \text{(D.6)}$$

hence $$\sum_{n=0}^{M/2} \mathrm{Sin}(n\delta) = \frac{1}{\tan(\delta/2)} \qquad \text{(D.7)}$$

and $$\sum_{n=0}^{M/2} \mathrm{Cos}(n\delta) = 0 \qquad \text{(D.8)}$$

Substituting eq.(D.7) into eq. (D.2) yields

$$\epsilon_{max} = \sum_{n=0}^{M/2} \frac{\delta^2}{2}.\mathrm{Sin}(n\delta) = \frac{\delta^2}{2.\tan(\delta/2)} \tag{D.9}$$

Substituting eq. (D.9) and eq. (D.8) into eq. (D.3) yields

$$\Sigma\,\mathrm{Cos}(\phi+n\delta) = \mathrm{Cos}\phi.0 - \mathrm{Sin}\phi.\frac{1}{\tan(\delta/2)} \tag{D.10}$$

and letting $\phi = -\pi/2$ yields

$$\epsilon_{max} = \sum_{n=0}^{M/2} \frac{\delta^2}{2}.\mathrm{Cos}(-\pi/2 + n\delta) = \frac{\delta^2}{2.\tan(\delta/2)} \tag{D.11}$$

The upper bound for the change in the lengths of the N vectors for all conditions is

$$\epsilon_{max} = \frac{\delta^2}{2.\tan(\delta/2)}$$

APPENDIX E

DOUBLE–SIDED PULSE WIDTH GENERATOR

E.1 Double Sided Pulse Width Generation Techniques

In section 6.3.1 it was recorded that the number of harmonics present would be larger for systems using single–edge modulation waveforms than for double–edge modulation waveforms.

For trailing single–edge modulation the pulses start at fixed intervals of time and the position of the trailing–edge is varied as shown in Fig. E.1 (a). For double–edge modulation the centre of each pulse occurs at fixed time intervals and the positions of both edges are varied as shown in Fig E.1 (b).

Trailing–edge modulated pulses can be produced by loading the digital value into a count–down timer (eg. Intel type 8155, 8253 or 8254 integrated circuits). The length of the pulse depends on the value placed into the counter. A more complex system is required to produce a double–edged PWM waveform.

E.2 Principle of Operation

The double–edged modulator developed for this project used up–down counters and two clock pulse frequencies T_{CLK} and PCLK. The basic system timing information was derived from a train of square wave pulses T_{CLK} where the rising edge indicated the start of the pulse interval and the falling edge the centre of the pulse interval as shown in Fig E.1 (c). The block diagram for the pulse–generator is given in Fig. E.2.

A rising edge–trigger circuit ED detected the rising edge of the timer pulse T_{CLK}, and caused the digital value of the pulse width to be loaded into the counter. The output flip–flop circuit TO was forced into a high state. (This is required only for the first pulse and has no effect on later pulses). The counter was placed in a count–down mode and counted down to zero with the PCLK pulses. When the value in the counter was zero, the zero detect circuit ZD caused the trigger flip–flop TO to change state. The counter continued to count down until the falling edge of T_{CLK} occurred. The counter was then placed into a count–up mode and counted back up to zero. The zero–detect circuit then sensed the second zero condition and changed the state of the TO flip–flop back to its initial state. The counter continued to count up until the next rising edge of

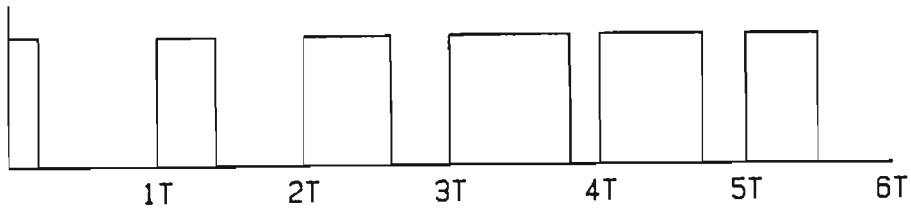


Fig E.1(a) Trailing Edge Modulation

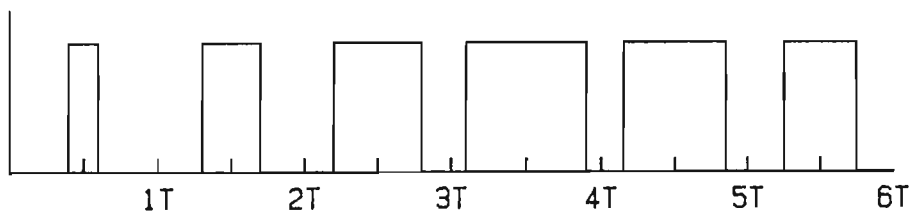


Fig E.1(b) Double Edged Modulation

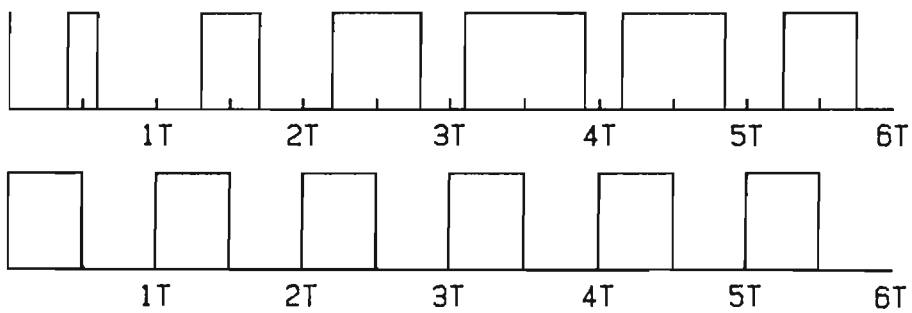
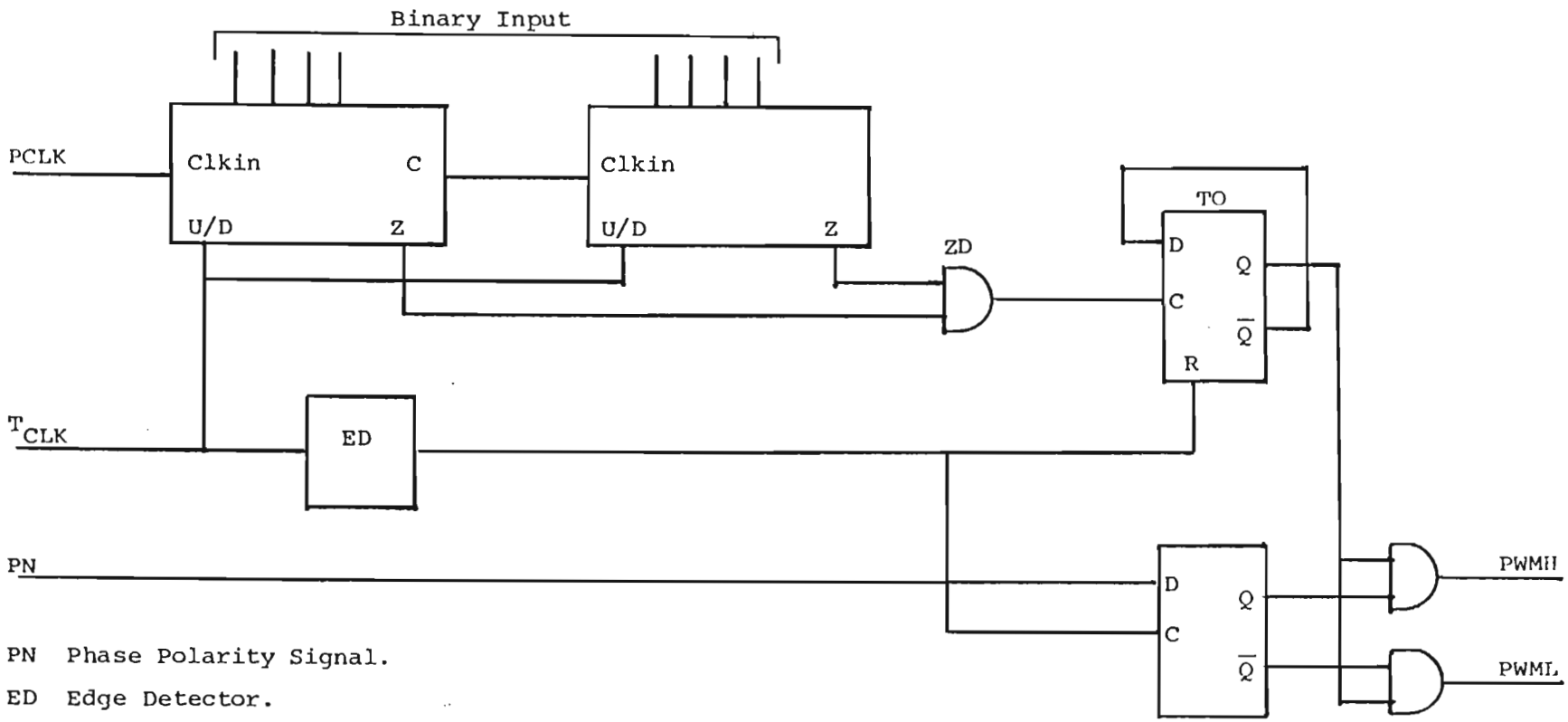


Fig E.1(c) Production of Double Edge Modulated Pulses



PN Phase Polarity Signal.

ED Edge Detector.

ZD Zero Detect Circuit.

U/D Up/Down.

Fig. E.2 Block Diagram of Double Sided Pulse Width Generator

T_{CLK} was reached, when the whole sequence was repeated. Hence each pulse was symmetrical about the falling edge of T_{CLK} as shown in Fig. E.1 (c).

The pulse–generator circuit was constructed with LS–TTL integrated circuits. The circuit operated successfully up to a maximum PCLK counting rate of 15 MHz. Higher speeds could have been achieved by using a rising–edge detector ED which produced a narrower trigger pulse.

The highest clock frequency available from the 8088 micro–computer was 15 MHz and there was no requirement to redesign the circuit to achieve a higher speed.

The pulse generator also contained an additional circuit to steer the PWM pulses to either the upper or lower driver circuit of the inverter leg. An additional control line PN from the computer was used to indicate whether the positive or negative half–cycle for each phase was being switched. In order to prevent the production of narrow pulses which were less than the minimum switching time of the inverter switching devices, narrow pulse inhibit circuits maintained the on or off condition of the inverter switch.

E.3 SVM Pulse Generator

The double edge pulse generator described in the previous section cannot be used for the Space Vector Modulation (SVM) technique since no phase control signal is produced by the SVM process. Two modifications are required; one to modify the selection of the switching device in the inverter and a second to provide a "dead band delay" circuit to prevent both switching devices being turned on simultaneously and causing a "shoot through fault".

The principle of operation of SVM pulse generator is the same, however the TO flip flop selects the required switching element as shown in Fig. E.3. Every time the Zero Detect circuit senses a zero condition in the U/D counters, a dead band delay (DBD) circuit is activated. This circuit consists of a down counter which is automatically loaded with a predertermined value. While the dead band circuit is in a non–zero state both switching devices are turned off. When the dead band counter reaches a zero state, the counter is disabled and the appropriate switching element turned on. The zero detect signal is also used to inhibit the drive circuits to prevent the generation of "glitches" or narrow destructive turn–on spikes.

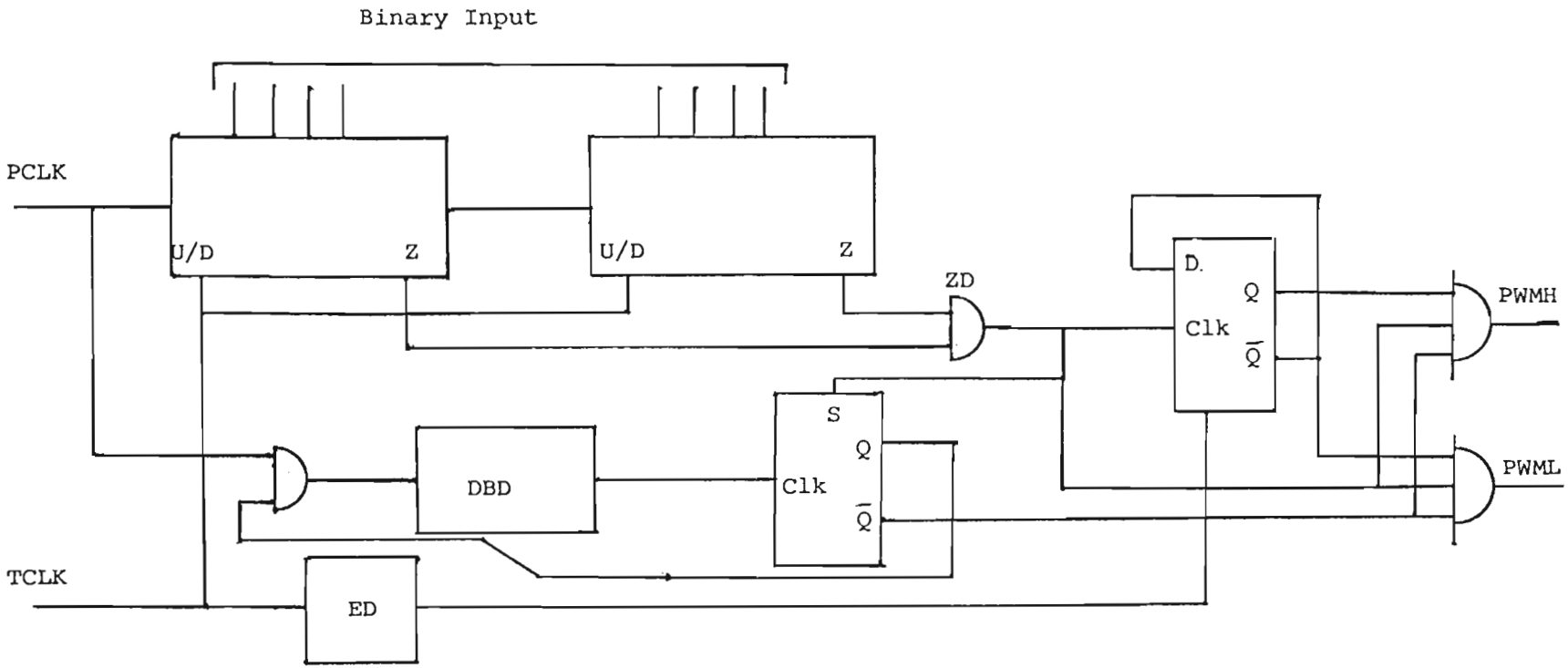


Fig. E.3 Block Diagram of SVM Pulse Generator

APPENDIX F

DERIVATION OF VOLTAGE / FREQUENCY TABLE

F.1 Introduction

In order to obtain constant torque, a constant flux–density is required. This is achieved by varying the amplitude of the supply voltage in proportion to the frequency. The exact form of the relationship is not linear and is dependant on a number of factors. The Voltage/frequency characteristic used in this project was based on current experience and is an example of what might be required in practice. Fig. 7.4 shows the shape of the characteristic.

At frequencies above 50 Hz the motor is operated in a constant power mode to prevent the insulation of the windings from being over–stressed. From 10 Hz to 50 Hz, the voltage is increased linearly with frequency. Between 2 Hz and 10 Hz the voltage is decreased by 10%, and below 2 Hz the voltage is kept constant.

F.2 System Constants

Using the cirteria given in Sections 6.1 and 6.2, the three initial pulse widths were chosen as 04000H, –02000H and –02000H or 16328, – 8192 and –8192 respectively. Allowing a 5% margin for amplitude errors, the maximum allowed value of a pulse–width is

16238 * 1.05 = 17204 or 04334H.

Hence the largest value for the 8–bit amplitude control variable was limited to

256/1.05 = 244 or 0F4H.

Attempts to use this value lead to a cumbersome and inaccurate table. A value of 0F2H was found to be more suitable and enabled a more precise table to be constructed.

The timing analysis of the program showed that with an 8 MHz crystal the calculation time for each step took 426 μs, with 12 μs for the "end–of–loop test". Owing to the variation in time for each multiplication operation it was necessary to allow for two end of sequence loops, or a total calculation interval of 460 μs.

The pulse generator required 512 pulses for each step or a T_{CLK} frequency of

$$T_{CLK} = \frac{460}{512}\,\mu s = 0.89\,\mu s$$

or approximately 1.09 MHz.

The above frequency was used in all the experiments for the 3–phase 16–bit PWM Controller.

F.3 Relationship between k and Output Frequency

The 16–bit value of k was generated by reading an 8 bit value into the most significant byte, and shifting it down by 3 bits. The five least significant bits were all set to 1 so that k could never be inadvertently set to zero.
i.e. the format of k was

000X XXXX XXX1 1111

and hence the maximum and minimum values of k were 01FFFH and 001FH respectively.

The maximum value of δ was therefore

$$\sqrt{3} * 01FFFH/0FFFFH = 0.125 * \sqrt{3} = 0.216$$

and hence the maximum expected gear ratio M was

$$M = 2\pi/\delta = 2\pi/0.216 = 29 \text{ (pulses per cycle).}$$

The time taken for one step was 460 μs and hence the highest frequency that could be generated was

$$T = 29 * 460\,\mu s = 13.3 \text{ ms}$$

and hence the maximum inverter frequency was

$F_{max} = 1000/13.3 = 74.9$ Hz

i.e. a digital value of k = OFFH corresponded to an output frequency of 75 Hz.

The general form of the equation relating the digital input value X to the frequency F is

$$k = ((X * 32) + 31)/2^{16} \quad (F.1)$$

$$\delta = \sqrt{3} * k = \sqrt{3} * ((X * 32) + 31)/65536 \quad (F.2)$$

now $M = \frac{2\pi}{\delta}$ and

$$T = M * 460 * 10^{-6} \quad (F.3.1)$$

$$F = 1/T = \frac{((X * 32) + 31) * \sqrt{3}}{2\pi * 65536 * 460} * 10^{6} \quad (F.3.2)$$

or $$F = ((X * 32) + 31) * 9.144 * 10^{-3} \quad (F.4)$$

by simple algebra the inverse relationship can be deduced

$$X = \frac{1}{32} \cdot \left[\frac{F}{9.144 * 10^{-3}} \right] - 31 \quad (F.5)$$

The principal values for the V/f table were then calculated and are shown in Table F.1.

F.4 V/f Control Table

Using the above values the V/f control table was constructed. The maximum value of the table was 0F2H (242) hence at 10 Hz the amplitude was one fifth or 242/5 = 48 and the value at 2 Hz was chosen to be 50% higher or 72. Thus the V/f table was defined as follows

a) From 50 Hz to 75 Hz the voltage control value is 242 (OF2H).

b) From 10 Hz to 50 Hz the voltage control value linearly increases from 48 (030H) to 242 (OF2H).

c) From 2 Hz to 10 Hz the value decreases from 72 (048H) to 48 (030H).

d) Below 2 Hz the value is constant at 72 (048H).

The above is shown graphically in Fig. 7.4

Table F.1

Relationship between Frequency and Digital Input Value

Frequency (Hz)	X (Decimal)	X (Hex)
75	255	0FFH
50	170	0AAH
10	34	022H
2	6	06H
0.28	0	0

REFERENCES

[1] Acha J I, Payan–Somet J and Civit A: "Design of Very Low Frequency Digital Oscillators", Proc. IEE, Vol 131, Part G. No 3, June 1984, pp 93–102.

[2] Agbinya J I: "Microprocessor Determination of PWM Signals using Second Order Difference Equations", IEEE Trans on Industrial Electronics, Vol IE–34, No. 4, Nov 1987, pp 494–496.

[3] Bowes S R and Bird B M: "Novel Approach to the Analysis Synthesis of Modulation Processes in Power Converters", Proc. IEE, Vol 122, May 1975, No. 5, pp 507–513.

[4] Bowes S R: "New Sinusoidal Pulsewidth Modulated Inverter" Proc. IEE, Vol 122, Nov 1975, No. 11, pp 1279–1285.

[5] Bowes S R and Mount M J: "Microprocessor Control of PWM Motors", Proc. IEE, Vol 128, Part B, No. 6, Nov 1981, pp 293–305.

[6] de Buck F, Gistelinck P and de Backer D: "Loss Optimal Waveforms for Variable Speed Induction Motor Drives", Proc. IEE, Vol 130, Part B, No. 5, Sept 1983, pp 310–320.

[7] Dwyer E and Ooi B T: "A Look–up Table Based Microprocessor Controller for a Three Phase PWM Inverter", IECI 79 Proceedings Industrial and Control Applications of Microprocessors, March 19–21 1979.

[8] Edwards J: "A Three Phase Digital PWM Controller", Proc. IEE, Vol 122, March 1975, pp 302–305.

[9] Grant D A, Houldsworth J A and Lower K N: "A New High Quality PWM AC Drive", IEEE Trans Industrial Applications, Vol IA–19, No. 2, 1983, pp 211–216

[10] Grant D A and Houldsworth J A: "PWM AC Motor Drive Employing Ultrasonic Carrier" Proc. IEE, To be published.

[11] Green R M and Boys J T: "Implementation of Pulse width Modulated Inverter", IEEE Trans Industrial Applications, Vol IA–18, No. 2, March/April 1982, pp 138–145.

[12] Green R M and Boys J T: "PWM Sequence Selection and Optimisation: A Novel Approach.", IEEE Trans Industrial Applications, Vol IA–18, No. 2, March/April 1982, pp 146–151.

[13] Green R M and Boys J T: "A Reconsideration of the Design of the Modified McMurray Inverter Circuit", IEEE Trans Industrial Applications, Vol IA–18, No. 2, March/April 1982, pp 152–162.

[14] Green W B: "Digital PWM Controller", United Kingdom Patent 8514913 B, 13th June 1985.

[15] Green W B: "Digital PWM Controller", South African Patent 85/4455, 13th June 1985.

[16] Green W B: "Digital PWM Controller", Zimbabwe Patent 85/84, 13th June 1985.

[17] Green W B and Harley R G: "Digital PWM Modulation Using Oscillator Matrices", Trans of the SAIEE. To be published.

[18] Hahn W: "Theory and Application of Liapunov's Direct Method", Prentice Hall, 1963, LCCN 63–9837.

[19] Hamming R W: "Digital Filters", Prentice Hall, 1977, LCCN 76–30514.

[20] Hamming R W: "Coding and Information Theory", Prentice Hall, 1980, LCCN 79–15159.

[21] Hayes J P: "Computer Architecture and Organisation", McGraw–Hill, 1978, LCCN 77–18898.

[22] Hess S F and Parker S R: "Heuristic Bounds for the Frequency of Digital Oscillators due to Quantisation Noise", Electronics Letters, February 1977, Vol 8, No. 4, pp 86–87.

[23] Houldsworth J A: "The Gate Turn Off Switch in PWM ac Motor Control", Pulse (South Africa), Feb 1983, pp 71–75.

[24] iAPX 86/88 Users Manual: Intel Corporation. August 1981.

[25] Korn G A: "Basic Tables in Electrical Engineering", McGraw–Hill, 1965, LCCN 64–66023.

[26] Kreyszig E: "Advanced Engineering Mathematics", Prentice Hall, 1972, LCCN 71–172951.

[27] Kunz U: "A Microprocessor Controlled PWM Inverter with GTO Thyristors", Proceedings Motor–Con. June 1988, pp 34–43.

[28] LeHuy H: "A Microprocessor Controlled PWM Inverter", IECI Proceedings Industrial Applications and Control, March 1978, pp 223–227.

[29] Marcus M and Minc H: "Elementary Linear Algebra", MacMillan, 1968, LCCN 68–11004.

[30] Mazur T: "A Digital Logic PWM Speed Control for Single and Polyphase AC Motors", IEEE Conference Record Annual Meeting of the Industrial Applications Society, Oct 1973, pp 1–9.

[31] McMurray W: "Optimum Snubbers for Power Semiconductors", IEEE Trans. Industrial Applications, Vol IA–8, No. 5, Sept/Oct 1972, pp 593–600.

[32] Sekiya T, Kobayashi T, Furuhata S, Shigehane H and Ichijo M: "Power Transistor Modules for Power converter", Fuji Electric Review, Vol 29, 1983, pp 45–50.

[33] Sifferlen T P and Vartanian V: "Digital Electronics with Engineering Applications", Prentice–Hall, 1970, LCCN 69–108608.

[34] Tsukahara T, Yanase T, Kawabata S, Banba C, Meada E and Kurosaki T: "FRENIC 5000 Series Power Transistor PWM Type VVVF Inverter", Fuji Electric Review, Vol 29, 1983, pp 51–58.

[35] Woll R F: "Effect of Unbalanced Voltage on the Operation of Polyphase Induction Motors", IEEE Trans Industrial Applications, Vol IA–11, No. 1, Jan/Feb 1975.