INTERACTIVE SOFTWARE FOR EVALUATING RELAY ALGORITHMS

A Thesis

Submitted to the College of Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Department of Electrical Engineering

University of Saskatchewan

by

MUKESH NAGPAL

Saskatoon, Saskatchewan

October 1986

dedicated to my beloved

parents

UNIVERSITY OF SASKATCHEWAN

DEPARTMENT OF ELECTRICAL ENGINEERING

CERTIFICATION OF THESIS WORK

We, the undersigned, certify that Mr. Mukesh Nagpal, candidate for the Degree of Master of Science in Electrical Engineering, has presented his thesis on the subject Interactive Software for Evaluating Relay Algorithms, that this thesis is acceptable in form and content, and that this student demonstrated a satisfactory knowledge of the field covered by his thesis in an oral examination held on November 3, 1986.

External Examiner: Dr. R.T. Burton

of Mechanical Engineering Dept

Internal Examiners:

M. S. Sachdev

A. M. El-Serafi

COPYRIGHT

ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to Dr. M. S. Sachdev, his supervisor. With a wise and nice mix of direction and latitude, he has guided the entire course of this research. Dr. M. S. Sachdev has always been a source of encouragement to the author.

The author takes this opportunity to acknowledge the encouragement and moral support provided by his parents and all other family members. Special thanks are extended to his friends, Anoop, Atul and Shankar for their valuable suggestions.

The author is grateful to the University of Saskatchewan, and the Natural Sciences and Engineering Research Council of Canada for providing financial assistance in the form of a scholarship.

UNIVERSITY OF SASKATCHEWAN

Electrical Engineering Abstract No. 86A274

INTERACTIVE SOFTWARE FOR EVALUATING RELAY ALGORITHMS

Student: Mukesh Nagpal **Supervisor: Dr M. S. Sachdev**

M. Sc. Thesis Presented to the

College of Graduate Studies and Research

October 1986

ABSTRACT

Recent advances in microelectronics have made available low cost micro-processors with enhanced capabilities. This has increased the interest for using micro-processors in power system protection. Several relay algorithms proposed in the past can be implemented on a micro-processor to produce numerical estimates of operating parameters of a power system. These estimates form the basis of relaying decisions.

The relay algorithms are designed assuming the nature of the waveforms of the inputs. The estimates provided by an algorithm are not correct when the waveform of an input is not similar to that used in the design of the algorithm. The accuracy of the estimates is further reduced by the words of the digital processors being of finite sizes. Therefore, a relay algorithm is chosen after studying its characteristics and investigating its performance.

This thesis is concerned with the development of a software that provides an interactive approach for the evaluation of relay algorithms. The software incorporates several algorithms that are designed by using the trigonometric, correlation and least error squares techniques. The software also includes facilities for generating data that can be used for testing the performance of the algorithms in situations that a relay might encounter in a power system.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background

An electric power system generates, transmits and distributes electric energy with minimum interruptions and at reasonable costs. During the design of a power system, normal operating conditions, such as load changes, and abnormal operating conditions, such as faults and load-generation imbalances, are considered in detail.

The security and reliability of the system can be enhanced by providing enough spares, adequate capacity margins, and by installing multiple circuits between generating stations and major load centers. The reliability can be improved further by dividing a system into protection zones, providing separate relays for detecting faults in each zone and initiating control actions for isolating the faulted zone. This practice ensures flexibility during normal operation and minimizes the part of the system that must be isolated during faults and abnormal operating conditions.

1.2 Power System Protection

Protection systems include equipment that is necessary for recognizing the existence of a fault, determining its location and initiating the opening of circuit breakers for isolating the faulted zone from the system. Relays are almost exclusively used for performing these functions. However, the use of relay and circuit breaker combinations may not be economically justified in some situations. In such cases, fuses are used to perform these functions.

There are many types of relays, used separately or collectively, depending on the type of protection and the type of equipment to be protected. The basic electrical quantities that change during faults and abnormal operating conditions are the voltages, the currents, the phase angles between the voltages and currents and the frequency. These changes can provide substantial information on the nature of a disturbance, the identity of the equipment that is experiencing a fault and the location of the fault.

In a protection scheme, each relay performs an assigned function and responds in a specific manner to the changes in the circuit quantities. For example, a relay could be designed to operate when the current in the relay exceeds a specified magnitude while another relay could be designed to operate when the ratio of the fundamental frequency voltage phasor to the fundamental frequency current phasor decreases below a set value. These relays are generally described by their connections, the actuating quantities and the disturbances to which they respond. On the other hand, the protection schemes derive their names from the equipment they protect.

1.3 Changing Trends in Power System Protection

The first automatic device used for isolating a fault in a power system was a fuse. One disadvantage of this device is that it has to be replaced every time it interrupts current. Also, it cannot discriminate a faulted zone from the unfaulted zones in a complex power system.

A significant improvement in power system protection was achieved with the introduction of automatic circuit breakers. This was soon followed by the development of electro-mechanical relays that could initiate the tripping of circuit breakers. Protective devices with greater sensitivity, better selectivity and faster speed were later developed to meet the requirements for applications in increasingly complex power systems.

The development of solid-state electronic relays started in the 1950's. In the early stages, these relays were not generally accepted by the users because of high failure rates of electronic components and inappropriate designs. Later developments used newer technology and improved designs. Today, several kinds of solid-state relays are being used in power systems.

With the advent of digital technology, development of the micro-processor based relays has received considerable attention. The use of real time computations for system protection was first proposed in 1966 by Last and Stalewsky [1]. Rockefeller [2] studied the feasibility of using a digital computer for the protection of a substation and the transmission lines emanating from it. A great deal of research is still being conducted in the area of computer relaying.

Several digital methods have been developed for estimating the electrical parameters of power systems. These methods use digital signal processing techniques and are commonly referred to as "relay algorithms". These algorithms can be implemented on micro-processors for producing numerical estimates of power system parameters from sequences of voltage and current samples. The estimates provided by the algorithms form the bases of relaying decisions.

Presently, several relaying algorithms are available and research is under way to develop newer algorithms. However, no single algorithm is optimum for all applications. Each algorithm has its own strengths and weaknesses. This is because the design of an algorithm is based on some assumptions concerning the waveforms of the inputs. An algorithm provides error free results when the waveforms of the inputs are those assumed during the design of the algorithm. The waveforms of power system voltages and currents during faults are distorted and contain components of several frequencies besides the fundamental frequency component. The errors depend on the magnitudes of the components not considered in the algorithm design and the response of the algorithms to those components. Two limiting factors in implementing the algorithms on a digital processor are the saturation of analog to digital converters, and the finite size of the words of the analog to digital converters and processors used. Therefore, to select an appropriate algorithm for a relay design, a tool to evaluate different relay algorithms would be useful. The tool should be suitable for testing an algorithm for its response to the inputs from power systems experiencing faults and abnormal operating conditions. The tool should be suitable for studying the influence of finite word size of digital devices and the saturation of analog to digital converters on the performance of an algorithm.

1.4 Objective of Thesis

The major objective of the work reported in this thesis, is the development of an interactive software package that would evaluate the performance of relay algorithms presently being used in industry and in research projects. The package, as the name suggests, should provide an interactive environment. It should also allow the simulation of the effects of limiting factors experienced in the implementation of a relaying algorithm on the digital processors.

1.5 Outline of Thesis

This thesis contains eight chapters and four appendices. The first chapter introduces the subject of the thesis and describes its organization. Chapter 2 outlines the advantages of using digital relays, describes the functional block diagram of a typical digital relay and, finally, identifies the sources of errors associated with a digital relay.

Chapter 3 describes the relaying algorithms that can be used to estimate the peak values and the phase angles of phasors representing power system voltages and currents. The recently developed techniques for estimating the frequency of a power system are then described. Major factors that generally influence the selection of relaying algorithms are discussed in Chapter 4. The interdependence of these factors and the trade-offs that must be made while designing a digital relay are also examined in this chapter.

The development of an interactive software that is suitable for evaluating the performance of digital algorithms is described in the fifth chapter. The specifications of the software and its structure are outlined. The overall organizational block-diagrams and brief descriptions of the major software programs are included.

Chapter 6 describes the testing of the software that includes the sample studies to evaluate the peak value and frequency estimation algorithms. Test results are included to demonstrate the capabilities of the developed software. The seventh chapter includes a summary of the thesis. The conclusions drawn from the work reported in the thesis are then presented. A list of references is given in Chapter 8.

The thesis contains four appendices. Appendix A describes a technique for obtaining the frequency response of a digital filter. A technique for modelling analog to digital converters is outlined in Appendix B. A procedure for performing multiplications in micro-processors using bit shifts, additions and subtractions is described in Appendix C. The last appendix, Appendix D, describes a piecewise linear approximation technique for estimating the amplitudes of phasors from their real and imaginary components.

Chapter 2

DIGITAL RELAYS IN POWER SYSTEM PROTECTION

2.1 Introduction

Power systems are, generally, protected by using combinations of relays and circuit breakers. Relays detect the onset of faults and, if necessary, initiate the opening of circuit breakers to isolate the faulty equipment. Modern power systems are complex networks. The complexity of these networks demands that the relays used for the protection be reliable, secure, accurate and take less time to make decisions. The electro-mechanical and static relays are, generally, used. However, several individuals and organizations have been conducting substantial research in the area of computer relaying for the last several years. The use of digital relays in power systems should be compared with the existing electro-mechanical and solid-state relays before adopting them for general use.

Early research in the field of computer relaying considered the use of a single computer (a mini-computer) for all the relaying functions in a substation. In case of computer break-downs, the use of a stand-alone computer would result in complete failures of substation protection. A standby computing system would be needed to avoid such failures. The use of two main-frame computers, a main and a standby, for a substation protection appeared too expensive to be commercially viable. However, recent advancements in microelectronics have resulted in the availability of low cost processors with enhanced capabilities. This has changed the present view to use individual micro-computers dedicated to specific relaying functions with facilities for data exchange among themselves. It is expected that this concept will result in realizing the advantages of computer relaying without the drawbacks of using a main-frame computer.

This chapter briefly describes three major aspects of digital relays. Firstly, it highlights the benefits that can accrue from the use of digital relays instead of the conventional relays. A typical block diagram of a digital relay is then presented. The results computed by digital relays would have inherent errors. The errors are a function of the word length of the digital processor and the number of bits in the analog to digital converter used. The errors in digital relaying that may be introduced by the finite word length of the analog to digital converters and digital processors are examined.

2.2 Perceived Benefits of Digital Relaying

Recent developments in the field of micro-processors have enabled digital relays to be a viable alternative to electro-mechanical and static relays. It seems that digital relays will eventually be more economical than the conventional relays. The cost of conventional relays has been increasing during the last twenty years whereas the cost of digital devices has been decreasing rapidly during the same time. Also, the software development cost per unit in a digital relaying system would be reduced if a large number of identical units are manufactured and sold.

In addition to the relaying functions, a digital relay has the potential to perform other tasks, such as self diagnosis, data analysis etc. This additional feature can offset, to some degree, the higher cost of presently available digital relays. A brief summary of the specific advantages of using digital relays is presented in this section.

2.2.1 Flexibility

A digital relay is a programmable device. Revisions and modifications in relay characteristics, necessitated by changes in operating conditions, can be made through pre-programmed modules. A single, general purpose hardware based relay can be designed to perform a variety of protection and control tasks. This would lead to a smaller inventory of spares for maintenance.

2.2.2 Reliability

The failure of a conventional relay becomes apparent only when it fails to operate upon encountering a fault or malfunctions under the normal operating conditions. However, most of the hardware failures in a digital relay can be detected as they occur. Additional diagnostic features, such as specific programs, can be executed to test the hardware, calibration of the analog to digital converter and the integrity of programs. Therefore, it is expected that most of the failures in a digital relaying system can be detected immediately, and this can be used to alert the operator for corrective measures.

2.2.3 Data-interface Access

A digital relay can be equipped with input/output ports for exchanging data and control commands. The pre-fault and post-fault signals can be stored in the relay memory and later transmitted to a central computer through a data link. This information can be used for further investigations that might lead to improved operating practices and relay designs.

2.2.4 Adaptive capabilities

A digital relay can be programmed to automatically change its characteristics depending upon the operating status of the power system. The change can be made either by considering the information locally provided to the relay or on receipt of a command from the central computer via a data link. The change may consist of selecting a new setting, or selecting a new protection routine.

2.2.5 Mathematical Capabilities

Designs of conventional relays are constrained by the characteristics and limitations of the electro-mechanical or solid-state components. But the digital relays can be programmed to provide almost any characteristics. Programming a complex characteristic is only nominally more difficult to implement than a simple characteristic.

2.3 Functional Details of a Digital Relay

Major functional blocks of a digital processor based relay are shown in Figure 2.1. The analog sub-system receives low level voltage and current signals from voltage transformers and current transformers, respectively. The sub-system isolates the relay from the power system and provides protection from transient over voltages. It also uses low pass filters to band-limit the signals. As analog to digital (A/D) converters accept only voltage signals as inputs, the sub-system converts all currents into equivalent voltages and reduces their levels to avoid saturation of A/D converters.

The outputs of the analog sub-system are applied to the analog interface sub-system. This sub-system includes sample and hold, A/D conversion and multiplexing hardware. The processed signals from the analog sub-system are sampled at a selected sampling rate. The sampling rate and the cut-off frequency of the analog filters (in analog sub-system) are inter-dependent. The sampling rate must be at least twice that of the frequency of the highest frequency component expected to be present in the analog inputs. The instantaneous values of the signals are held as voltages across capacitors. A multiplexer applies each voltage in turn to an A/D converter that converts the sampled values to equivalent digital representations. Alternatively, dedicated A/D converters can be used for each sampled signal. A multiplexer can then be used to read each information sequentially into the computer.

The digital input sub-system conveys the status of the power system circuit breakers and switches to the relay. Input wiring must be properly shielded to protect the relay from transient voltages that may be experienced on the wiring.

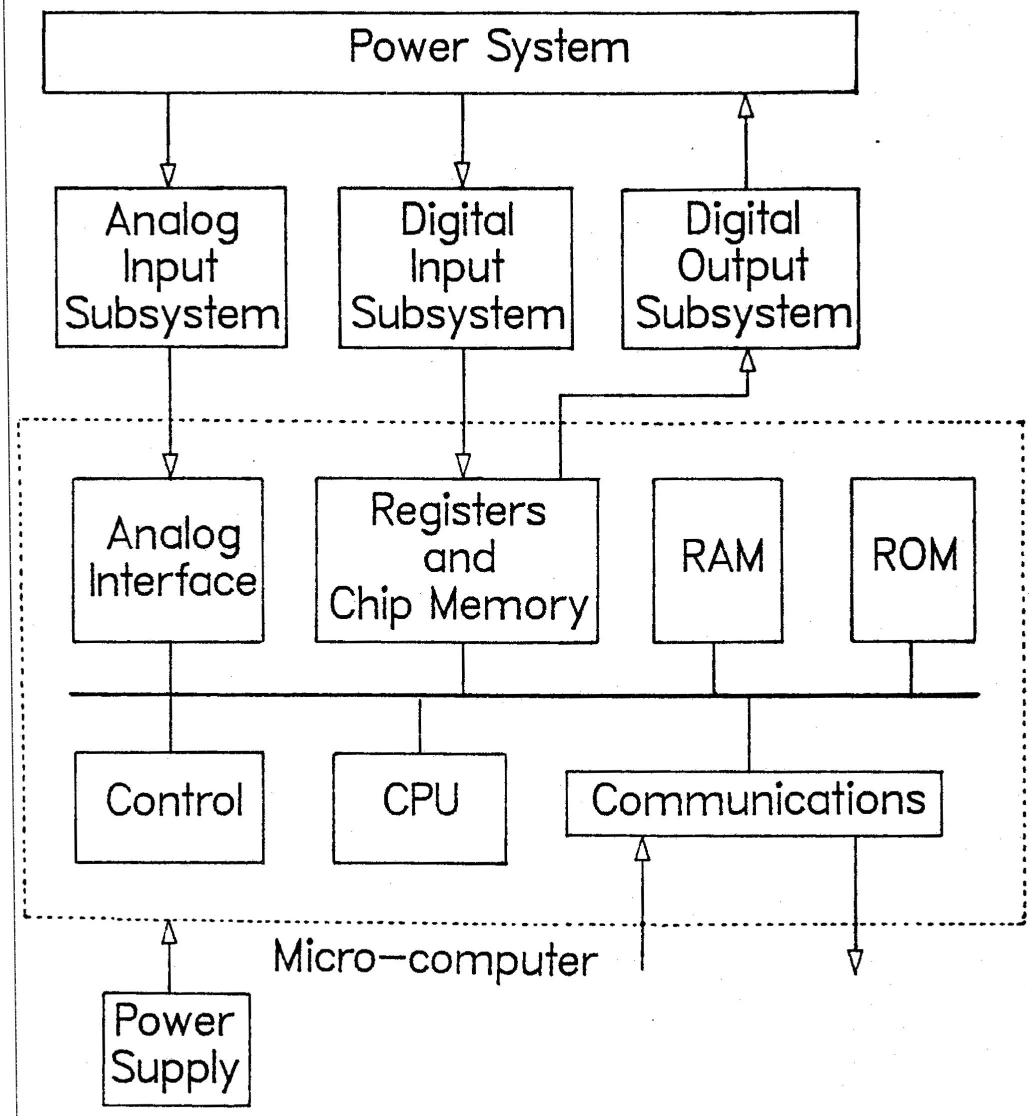


Figure 2.1 Functional block diagram of a digital relay.

The digitized data are then entered into the Random Access Memory (RAM). A record of significant events in the power system are saved in the Random Access Memory as historical files. The organizations and lengths of the data files depend on the needs of the users. The data stored in historical files should be moved to a secondary device (a local computer or a remote host) as soon as possible, thus freeing the RAM for storing information on the next occurrence of a transient. The relay programs reside in a non-volatile, Read Only Memory (ROM). The controllers, central processing unit (CPU) and the registers work as a group to execute the programs, one statement at a time.

The digital output sub-system conveys the decisions of the relay to the power system. The outputs from the relay, generally, provide signals for tripping circuit-breakers, annunciators etc.

A digital relay requires uninterrupted supply of power. A battery and an AC to DC converter are usually used to supply power to the digital relays. AC to DC converters used in these relays are designed for continuously supplying the power demand of the relay and sufficient power to keep the battery fully charged. Whenever the AC supply to the converter fails, the battery starts supplying uninterrupted power to the relay.

2.4 Errors in Digital Relays

The advantages of using digital relays have been discussed in Section 2.2. However, these relays have the inherent problem of limited accuracy. This is because all digital devices have finite number of bits in a word, for example, the number of bits in an A/D converter and word length of a digital processor. The coefficients of digital filters (relay algorithms) can not be represented precisely in processors. Therefore, the responses of filters implemented on micro-processors differ from those of the precise implementations. Also, the errors in computations are introduced when results of arithmetic operations are stored in words of finite sizes. An A/D converter is designed for a specified range of voltage inputs. When an input is out of this range, the A/D converter saturates resulting in an incorrect representation of the input. References [3,4] describe the effects of finite word size in digital filters. This section briefly describes these sources of errors associated with a digital relay.

2.4.1 Analog to Digital Converter Errors

Consider an A/D converter that has b+1 bits. Also consider that an analog voltage, v, is applied to this converter and the binary output is $Q(v)$. In most cases, the binary representation will not be equal to the true value of the analog input. Theoretically, an infinite number of bits may be required to correctly represent some analog inputs within the specified range. For practical implementation, these numbers are either truncated or rounded to fit into a selected word length.

2.4.1.1 Truncation Error

For a truncated binary representation, the difference between a quantized value and the true value, $Q(v)-v$, is defined as the truncation error. For positive numbers, the largest error occurs when all the truncated bits are unity. The truncation error is, therefore, negative and its magnitude is less than the value of least significant bit (LSB). The error, ε, can, therefore, be defined as

$$0 \geq \varepsilon \geq -2^{-b}. \tag{2.1}$$

For negative numbers, the truncation errors depend on the method used for their representation. Negative numbers are, commonly, represented by sign and magnitude, one's complement and two's complement representations [5]. Truncation errors for negative numbers are positive if sign and magnitude, and one's complement representations are used. The error, ε, can be mathematically expressed as follows:

$$0 \leq \varepsilon \leq 2^{-b} \tag{2.2}$$

If two's complement representations are used, truncation errors are negative and can be expressed as follows:

$$-2^{-b} \leq \varepsilon \leq 0 \tag{2.3}$$

where

ε is the truncation error.

2.4.1.2 Rounding Error

Instead of truncating, numbers can be rounded to fit into a finite-length word. This process consists of choosing the closest quantization level. Numbers exactly halfway between two quantization levels may either be rounded up or down. Assuming that a number falling exactly halfway between two steps is always rounded up, the rounding error, ε, is

$$-\frac{1}{2}2^{-b} < \varepsilon \leq \frac{1}{2}2^{-b}. \tag{2.4}$$

Both truncation and rounding are non-linear in nature and are graphically shown in Figure 2.2.

2.4.1.3 Saturation Errors

An A/D converter is designed to receive signals in a fixed range of voltage levels. If the inputs do not lie in this range, the A/D converter will saturate. In such cases, the digital representations of the inputs will not be correct. As a consequence, computations performed using these representations, will also not be correct. Figure 2.3 shows an input of 12V peak signal as interpreted by an A/D converter designed for an input range of +10V to -10V.

2.4.2 Digital Processor Errors

In addition to the truncations of the inputs by the A/D converter, inputs may also be subjected to further truncations if the word-length used in the processor is shorter than that of the A/D converter (but this is seldom the case). However, during the arithmetic operations, such as additions, subtractions, multiplications or divisions, the results can not always be stored in the finite word size of the processor. For example, when two numbers of b+1 bits are multiplied, the product is a number that can not be stored in a word of b+1 bits. At such instances, truncation or rounding of results will be needed. As the computations proceed, the errors because of truncation or rounding will accrue. However, these errors can be reduced significantly by making use of multiple word lengths (i.e. two or more words of b+1 bits).

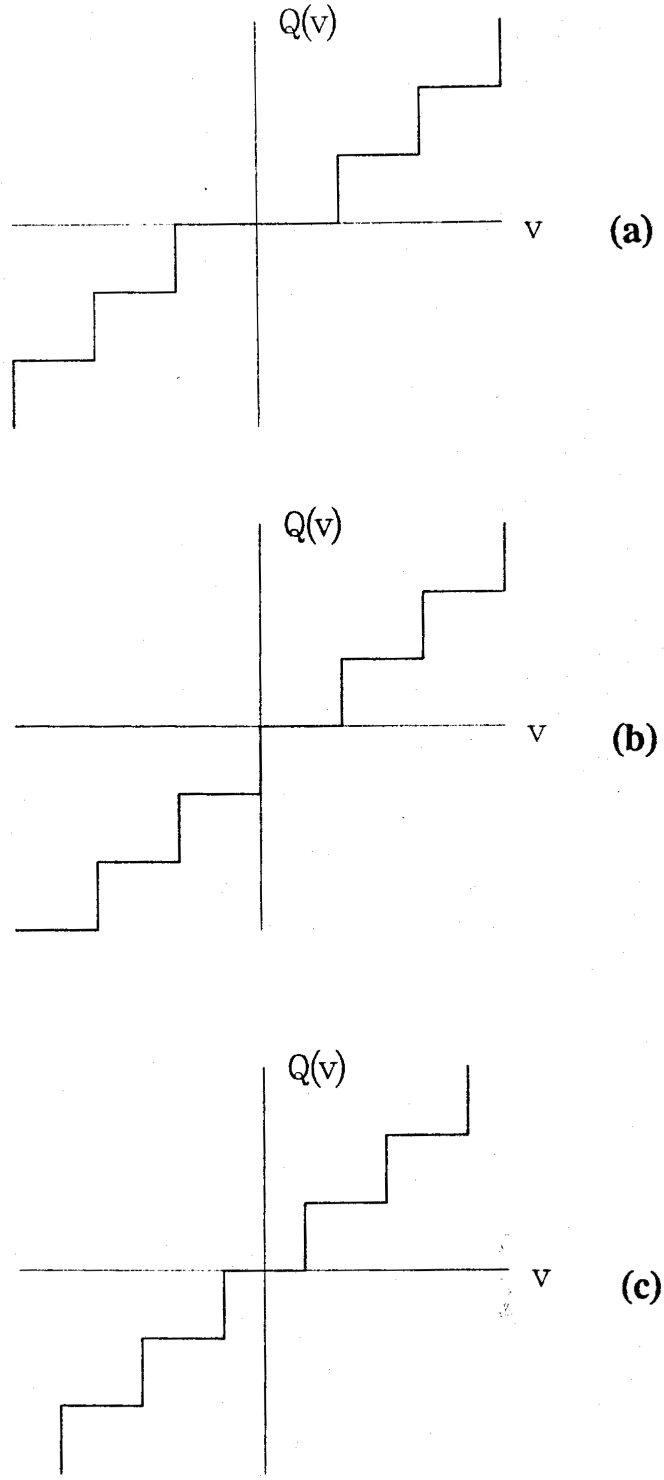


Figure 2.2 Quantization of analog information (a) truncation for sign and magnitude, and one's complement representation of numbers, (b) truncation for two's complement representation of numbers and (c) rounding.

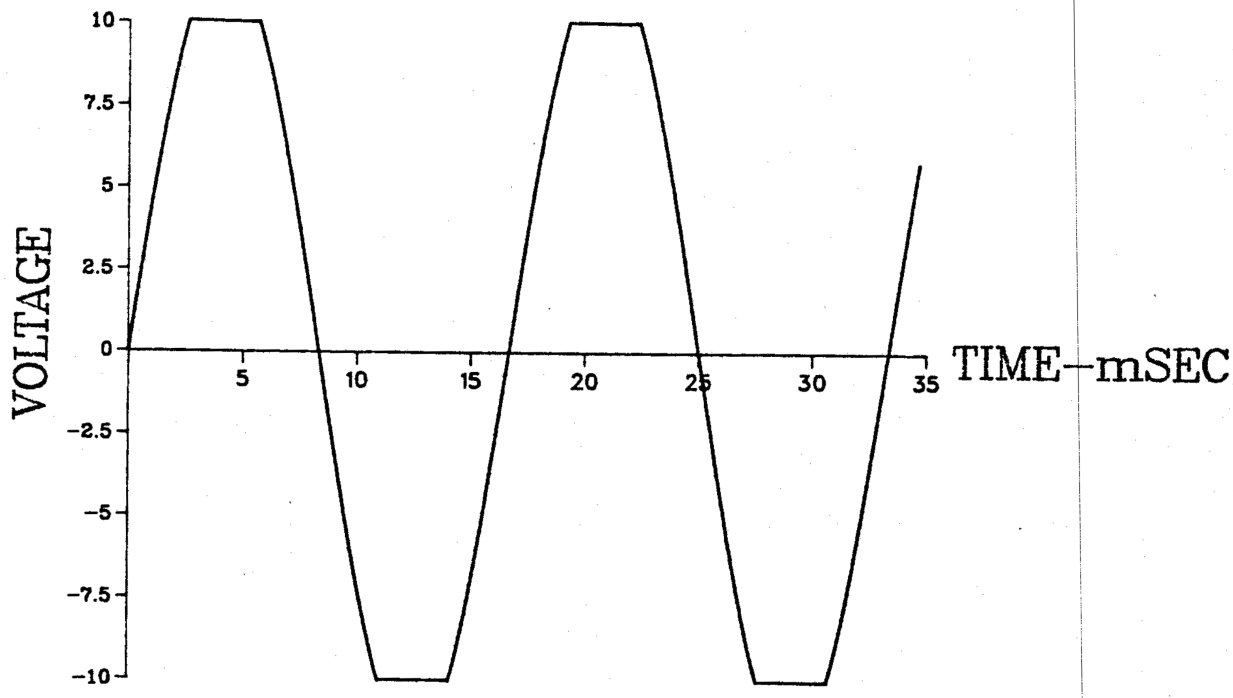


Figure 2.3 Representation of a signal as interpreted by a saturated A/D converter.

In some situations, the result of an addition or subtraction can not be stored in a single word, overflow or underflow would arise. Results computed by digital relays would be inaccurate if some means is not provided for detecting overflows and underflows and taking some suitable corrective actions.

2.4.3 Software Errors

Relay software errors can be due to several reasons. Two major sources of errors are identified in this section. One source is the approximations made while representing the filter coefficients by their equivalent binary numbers. The second source is the approximations used for representing functions, such as using a bit shifting approach for multiplications and divisions.

2.4.3.1 Effect of Inaccuracies in Coefficient Representations

To implement a relaying algorithm on a micro-processor, the filter coefficients are replaced by their equivalent binary numbers. Because of the finite word length of the processor, each coefficient is represented by its binary equivalent of b+1 bits. The difference between binary representations and the true values will alter the filter characteristics. This may adversely affect the accuracy of results.

2.4.3.2 Errors Due to Bit Shift Approach of Multiplication

A bit shift approach for multiplication is commonly used in micro-computers that do not use hardware multipliers. In some situations, this approach is also used to reduce the software execution time. In digital relays, digitized values of the inputs that are integers, are required to be multiplied with filter coefficients that are fractional values. This can be achieved by representing each filter coefficient as a series, and implementing the multiplications by a set of bit shift, add and subtract instructions. This approach introduces some errors in computations because the series may not precisely represent the filter coefficients. Errors are also introduced when right shift operations are performed on binary numbers. These errors also depend on whether the numbers are truncated or rounded. For multiple shift operations, the errors can continue to accumulate.

2.5 Summary

In this chapter, advantages of digital relays over the electro-mechanical and static relays have been discussed. A brief description of the functional block diagram of a typical digital relay has been presented. Lastly, the sources of errors in digital relays have been identified and discussed.

Chapter 3

ALGORITHMS FOR PROTECTIVE RELAYING

3.1 Introduction

Digital relays are software controlled devices that use algorithms for estimating the operating parameters of a system and use selected characteristics to make appropriate decisions. A relaying algorithm usually consists of a few mathematical equations that are implemented to produce numerical estimates of frequency, voltage and current phasors and derived quantities. Data sampled at prespecified rates are used by the algorithms to obtain the estimates. The outputs of these algorithms are compared with the relay characteristics for making decisions.

Substantial research towards the development of relaying algorithms has been conducted during the last fifteen years. Several algorithms developed so far estimate peak values and phase angles of voltage and current phasors. More recently, algorithms for estimating power system frequency have also been proposed.

This chapter reviews the mathematical basis of the algorithms that can compute the magnitudes and phase angles of phasors representing voltage and current waveforms. Recently developed techniques for estimating the power system frequency are then discussed.

3.2 Algorithms for Estimating Phasors

Digital algorithms can be divided in two classes, nonrecursive and recursive algorithms. The nonrecursive algorithms use a finite number of data samples to obtain an estimate. The time period that contains these samples is called a data window. The outputs of the algorithms depend solely on the values of the samples in a data window. The outputs of the recursive algorithms are functions of the present inputs as well as all the previous inputs. Digital signal processing texts [5] refer to the nonrecursive algorithms as finite impulse response filters and recursive algorithms as infinite impulse response filters.

3.2.1 Nonrecursive Algorithms

The nonrecursive relaying algorithms proposed in the literature [6] can be classified in the following four categories.

(1) Trigonometric algorithms.

(2) Correlation algorithms.

(3) Least error squares algorithms.

(4) Others.

These classifications are based on the mathematical techniques used in developing the algorithms. The discussion in this section assumes that signals being processed are power system voltages. Similar procedures can be used for processing data representing power system currents.

3.2.1.1 Trigonometric Algorithms

The trigonometric algorithms assume that the input voltages and currents are sinusoidal. However, the voltage and current waveforms during faults contain noise and components of harmonic and other frequencies. If the analog filters, that form a part of the analog subsystem, are suitable for eliminating noise and all non-fundamental frequency components, the outputs would be sinusoidal waveforms of the fundamental frequency. A voltage waveform of this type can, therefore, be defined as

$$v = V_p \sin(\omega_0 t + \theta_v) \tag{3.1}$$

where

v is the instantaneous value of the voltage,
V_p is the peak value of the sinusoidal voltage,
ω_0 is the nominal frequency of the voltage,
t is the time in seconds and
θ_v is the phase angle at $t=0$.

The first and the second derivatives of the voltage can be expressed by Equations 3.2 and 3.3, respectively.

$$\dot{v} = \omega_0 V_p \cos(\omega_0 t + \theta_v) \tag{3.2}$$

and

$$\ddot{v} = -\omega_0^2 v_p \sin(\omega_0 t + \theta_v) \tag{3.3}$$

The trigonometric algorithms are based on these equations. Three trigonometric algorithms are described in this section. These are the Makino and Miki algorithm [7], the Mann and Morrison algorithm [8], and the Gilcrest, Rockefeller and Udren algorithm [9,10].

(a) Makino and Miki Algorithm

Makino and Miki [7] proposed an algorithm that requires two samples of the signal that is to be processed. The instantaneous values of a voltage waveform at times $(k-1)\Delta T$ and $k\Delta T$ can be represented as

$$V_p \sin(\omega_0 t + \theta_v - \omega_0 \Delta T) = v_{k-1} \tag{3.4}$$

$$V_p \sin(\omega_0 t + \theta_v) = v_k . \tag{3.5}$$

Using the well known rules of trigonometry, Equation 3.4 can be expanded to the following form.

$$V_p \sin(\omega_0 t + \theta_v) \cos(\omega_0 \Delta T) - V_p \cos(\omega_0 t + \theta_v) \sin(\omega_0 \Delta T) = v_{k-1} \tag{3.6}$$

Substituting in this equation the value of $V_p \sin(\omega_0 t + \theta_v)$ from Equation 3.5, and rearranging, Equation 3.7 can be obtained.

$$V_p \cos(\omega_0 t + \theta_v) = \frac{v_k \cos(\omega_0 \Delta T) - v_{k-1}}{\sin(\omega_0 \Delta T)} \tag{3.7}$$

The peak value and phase angle of the phasor representing the voltage waveform can now be estimated using Equations 3.5 and 3.7 as follows:

$$V_p = \left\{ v_k^2 + \left[\frac{v_k \cos(\omega_0 \Delta T) - v_{k-1}}{\sin(\omega_0 \Delta T)} \right]^2 \right\}^{\frac{1}{2}} \tag{3.8}$$

$$\omega_0 t + \theta_v = \arctan \frac{v_k \sin(\omega_0 \Delta T)}{v_k \cos(\omega_0 \Delta T) - v_{k-1}} \tag{3.9}$$

(b) Mann and Morrison Algorithm

Mann and Morrison [8] used the values of the samples and first derivatives to estimate the peak values and phase angles of the phasors representing the input sinusoidal voltage and current waveforms. They also suggested that the first derivative, $\dot{v}$, can be estimated from three consecutive samples of the voltage. If three samples of a voltage are taken at $(k-1)\Delta T$, $k\Delta T$ and $(k+1)\Delta T$ seconds, the first derivative at time $k\Delta T$ can be estimated as follows:

$$\dot{v} = \frac{(v_{k+1} - v_{k-1})}{2\Delta T} \tag{3.10}$$

4

Substituting for $\dot{v}$ from Equation 3.2 it follows that

$$V_p \cos(\omega_0 t + \theta_v) = \frac{v_{k+1} - v_{k-1}}{2\omega_0 \Delta T}. \tag{3.11}$$

5

The instantaneous value of the voltage waveform at $k\Delta T$ seconds was defined in Equation 3.5 and is reproduced in Equation 3.12.

$$V_p \sin(\omega_0 t + \theta_v) = v_k \tag{3.12}$$

The peak value and phase angle of the phasor representing the voltage waveform can now be estimated using Equations 3.11 and 3.12 as follows:

$$V_p = \left\{ v_k^2 + \left[\frac{v_{k+1} - v_{k-1}}{2\omega_0 \Delta T} \right]^2 \right\}^{\frac{1}{2}} \tag{3.13}$$

$$\omega_0 t + \theta_v = \arctan \frac{2 v_k \omega_0 \Delta T}{v_{k+1} - v_{k-1}} \tag{3.14}$$

(c) Gilcrest, Rockefeller and Udren Algorithm

Gilcrest et al [9,10] used the first and second derivatives of voltages and currents to estimate the peak values and phase angles of the phasors representing them. This approach was used to minimize the effect of DC offsets that are present in voltages and currents. The proposed technique uses three samples of a sinusoidal voltage taken at times $(k-1)\Delta T$, $k\Delta T$ and $(k+1)\Delta T$. The first derivatives of the voltage at times $(k-\frac{1}{2})\Delta T$ and $(k+\frac{1}{2})\Delta T$ can be approximated as follows:

$$\dot{v}_{k-\frac{1}{2}} = \frac{v_k - v_{k-1}}{\Delta T} \tag{3.15}$$

and

$$\dot{v}_{k+\frac{1}{2}} = \frac{v_{k+1} - v_k}{\Delta T} \tag{3.16}$$

The first derivatives obtained from Equations 3.15 and 3.16 can now be used to estimate the second derivative at time $k\Delta T$ using the following equations.

$$\ddot{v}_k = \frac{1}{\Delta T}\left\{ \dot{v}_{k+\frac{1}{2}} - \dot{v}_{k-\frac{1}{2}} \right\} \tag{3.17}$$

$$\ddot{v}_k = \left(\frac{1}{\Delta T}\right)^2 (v_{k+1} - 2v_k + v_{k-1}) \tag{3.18}$$

Substituting for $\ddot{v}$ from Equation 3.3 it follows that

$$V_p \, sin\,(\omega_0 t + \theta_v) = -\left(\frac{1}{\omega_0 \Delta T}\right)^2 (v_{k+1} - 2v_k + v_{k-1}). \tag{3.19}$$

In Equation 3.11, $V_p \, cos\,(\omega_0 t + \theta_v)$ was defined as follows:

$$V_p \, cos\,(\omega_0 t + \theta_v) = \frac{v_{k+1} - v_{k-1}}{2\omega_0 \Delta T} \tag{3.20}$$

The peak value and phase angle of the phasor representing the voltage waveform can now be estimated using Equations 3.19 and 3.20 as follows:

$$V_p = \left(\frac{1}{\omega_0 \Delta T}\right)\left\{ \left[\frac{v_{k+1} - v_{k-1}}{2}\right]^2 + \left[\frac{v_{k+1} - 2v_k + v_{k-1}}{\omega_0 \Delta T}\right]^2 \right\}^{\frac{1}{2}} \tag{3.21}$$

$$\omega_0 t + \theta_v = arctan \, \frac{-2(v_{k+1} - 2v_k + v_{k-1})}{\omega_0 \Delta T \, (v_{k+1} - v_{k-1})} \tag{3.22}$$

3.2.1.2 Correlation Algorithms

The process of correlation of an input waveform with a set of orthogonal functions can be used to extract components of selected frequencies from the input waveform. Several orthogonal functions have been proposed in the past. The functions used in digital relaying are usually chosen from the following:

(1) Sine and cosine functions and

(2) Even and odd rectangular waves.

A power system signal can also be correlated to itself or to another power system signal. Correlating a signal with itself is referred to as auto-correlation and correlating a signal with another signal is referred to as cross-correlation. The correlation technique does not assume that the input signal is sinusoidal and can be applied to signals that contain components of non-fundamental frequencies in addition to the fundamental frequency components.

(a) Correlation with a Set of Orthogonal Functions

Ramamoorty [11] suggested that the information concerning the fundamental frequency voltage and current phasors can be extracted from the fault transients by correlating one cycle of data samples with the samples of sine and cosine waves of that frequency. The correlation of a signal with unit amplitude sine and cosine waveforms can be used to extract the real and imaginary parts of a phasor. This process is illustrated in Figure 3.1 and can mathematically be expressed as follows:

$$v_r = \frac{1}{\pi} V_p \int_0^{2\pi} \sin(\omega_0 t + \theta_v) \sin(\omega_0 t) \, d\,\omega_0 t \tag{3.23}$$

$$v_i = \frac{1}{\pi} V_p \int_0^{2\pi} \sin(\omega_0 t + \theta_v) \cos(\omega_0 t) \, d\,\omega_0 t \tag{3.24}$$

where

v_r is the real part of the fundamental frequency voltage phasor and

v_i is the imaginary part of the fundamental frequency voltage phasor.

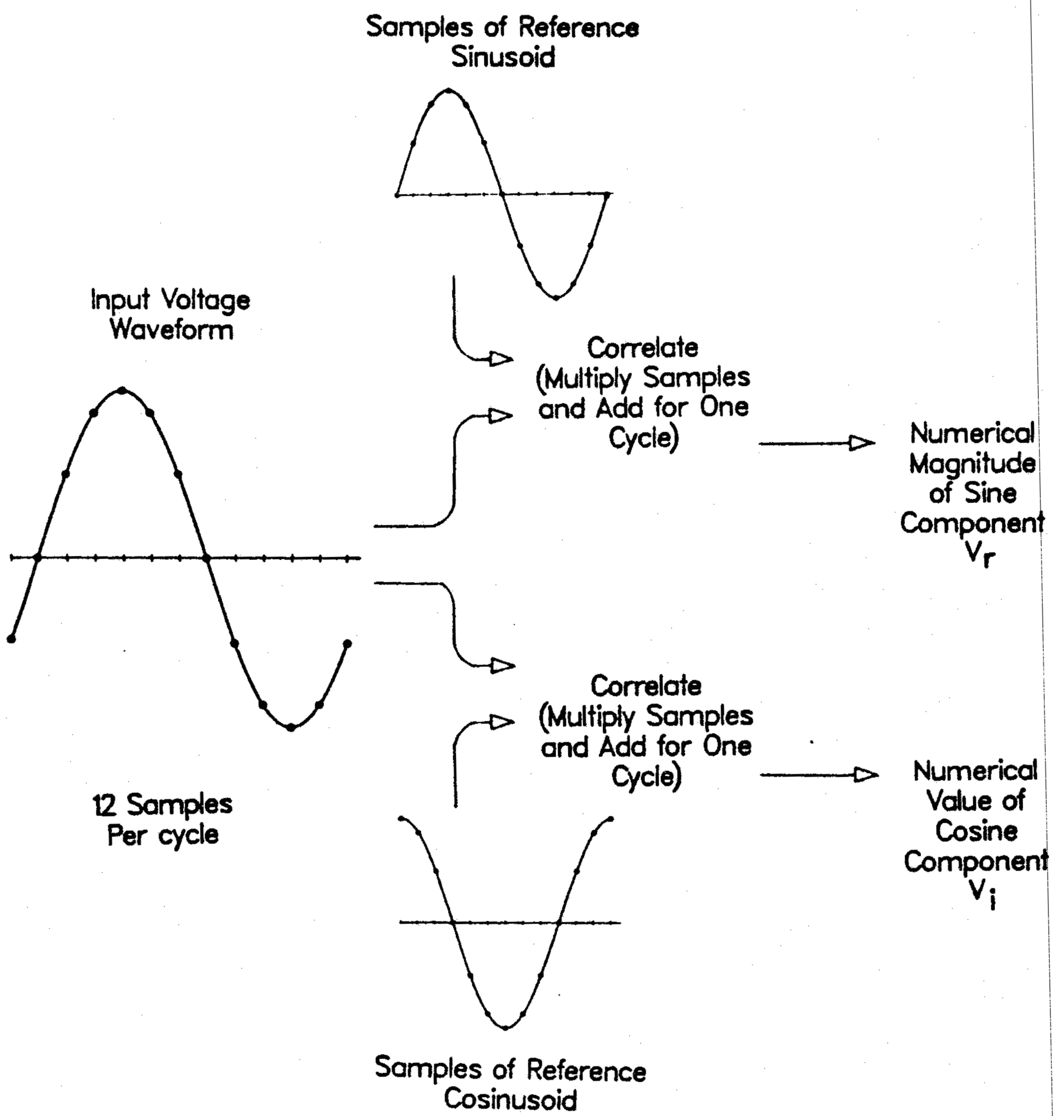


Figure 3.1 The correlation of an input waveform with sine and cosine waveforms.

In digital relaying applications, the input signals are available at discrete instants of time and, therefore, the numerical techniques are used to evaluate the integrations. Using the rectangular rule of integration, the real and imaginary parts of the voltage phasor can be estimated from the following equations.

$$v_r = (\frac{2}{m}) \sum_{n=0}^{m-1} v_{k+n-m+1} sin\,(\frac{2\pi n}{m}) \tag{3.25}$$

$$v_i = (\frac{2}{m}) \sum_{n=0}^{m-1} v_{k+n-m+1} cos\,(\frac{2\pi n}{m}) \tag{3.26}$$

In these equations, m represents the number of samples taken in one cycle of the fundamental frequency. The peak value and phase angle of the voltage phasor can now be determined as follows:

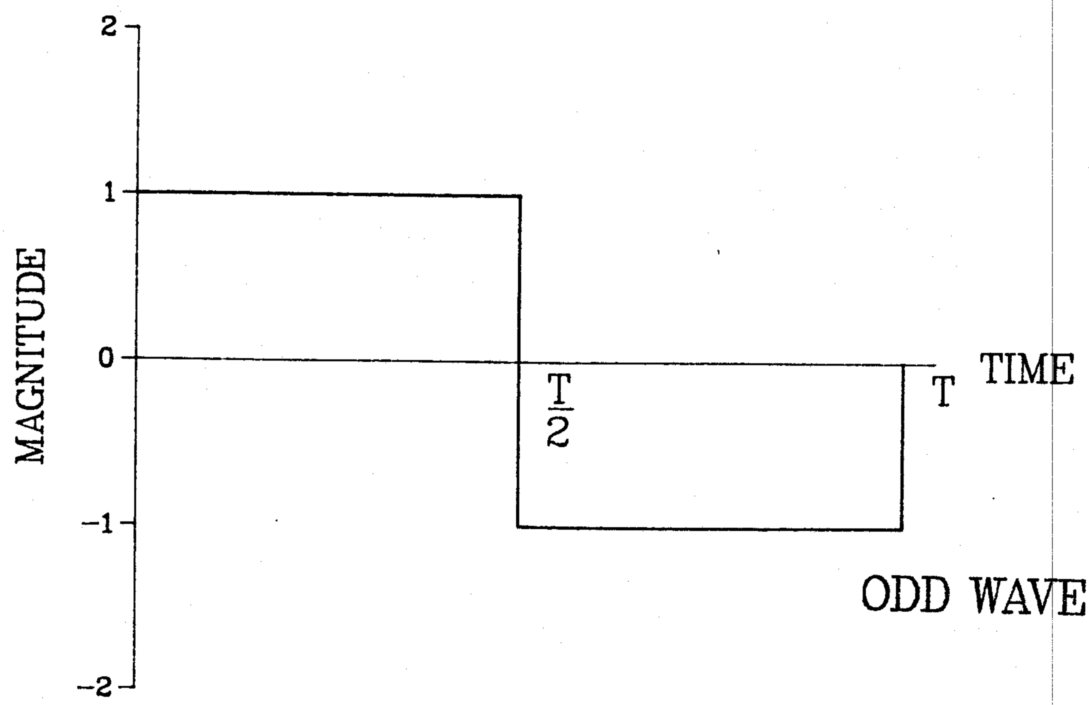
$$V_p = \sqrt{v_r^2 + v_i^2} \tag{3.27}$$

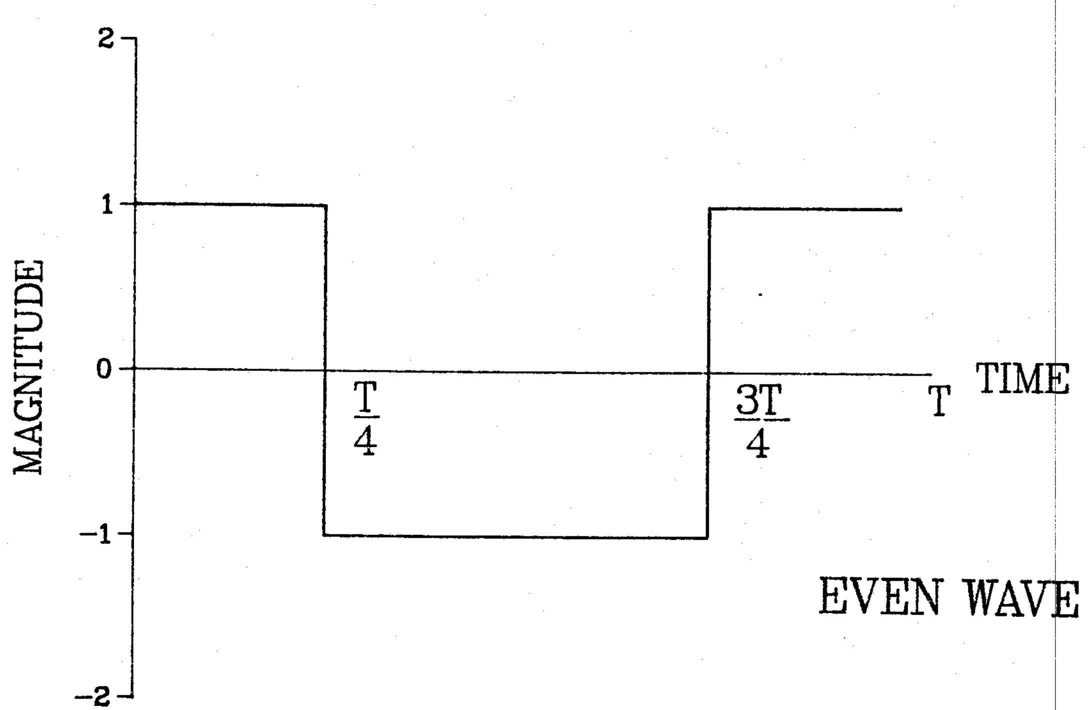
$$\theta_v = arctan\,(\frac{v_i}{v_r}) \tag{3.28}$$

This procedure is referred to as the Fourier algorithm. It can be extended to use multiple cycles of data to extract the phasor quantities. Phadke et al [12], applied the correlation approach using a data window of one half cycle plus one sample for faster response. The accuracy of results obtained by this approach is affected by the off-nominal frequency components and the DC offsets.

Another variation to this approach consists of correlating the data with even and odd rectangular waves [13]. These waves are shown in Figure 3.2 and can be mathematically defined as follows:

$$W_r(t) = signum \left\{ \sin(\omega_0 t) \right\} \tag{3.29}$$





Even and odd rectangular waves.

$$W_i(t) = signum\left\{ \cos(\omega_0 t) \right\} \tag{3.30}$$

where

$$\begin{aligned} signum(x) &= -1 \text{ for } x < 0 \\ &= 0 \text{ for } x = 0 \\ &= 1 \text{ for } x > 0, \end{aligned}$$

$W_r(t)$ is the odd rectangular wave and

$W_i(t)$ is the even rectangular wave.

The real and imaginary parts of the fundamental frequency voltage phasor can now be determined by using Equations 3.31 and 3.32.

$$v_r = (\frac{1}{A}) \sum_{n=0}^{m-1} v_{k+n-m+1}\, signum\left\{ sin(\frac{2\pi n}{m}) \right\} \tag{3.31}$$

$$v_i = (\frac{1}{A}) \sum_{n=0}^{m-1} v_{k+n-m+1}\, signum\left\{ cos(\frac{2\pi n}{m}) \right\} \tag{3.32}$$

where

A is the scaling factor.

The advantage of correlating with the even and odd rectangular waves is that the computations consist of additions and subtractions only.

(b) Auto-Correlation and Cross-Correlation

Hope et al [14], proposed the use of auto-correlation and cross-correlation of voltage and current signals for transmission line protection. The suggested procedure can be mathematically defined as follows:

$$\psi_1(t) = (\frac{1}{N}) \sum_{n=1}^{N} i_{k+n}\, i_{k+n+m} \tag{3.33}$$

$$\psi_2(t) = (\frac{1}{N}) \sum_{n=1}^{N} i_{k+n} v_{k+n+m} \tag{3.34}$$

where

ψ_1 is the auto-correlation function,

ψ_2 is the cross-correlation function and

m is the lag.

The ratios of the auto-correlation and cross-correlation functions can provide the information required to decide if a fault is in the zone of protection of a transmission line.

Gilbert and Shovlin [15] used auto and cross correlations for directly calculating the resistance, r_f, and the reactance, x_f, as seen by a relay installed at a transmission line terminal.

Consider that three samples of a current taken at times $(k-1)\Delta T$, $k\Delta T$ and $(k+1)\Delta T$ are defined as i_{k-1}, i_k and i_{k+1}, respectively. The instantaneous values of the current can be mathematically represented by the following expressions.

$$I_p \, sin\,(\omega_0 t + \theta_i - \omega_0 \Delta T) = i_{k-1} \tag{3.35}$$

$$I_p \, sin\,(\omega_0 t + \theta_i) = i_k \tag{3.36}$$

$$I_p \, sin\,(\omega_0 t + \theta_i + \omega_0 \Delta T) = i_{k+1} \tag{3.37}$$

Multiplying the left and right hand sides of Equation 3.35 with the corresponding sides of Equation 3.37 and performing trigonometric manipulations, Equation 3.38 can be obtained.

$$\frac{I_p^2}{2} \left\{ cos\,2(\omega_0 \Delta T) - \cos 2(\omega_0 t + \theta_i) \right\} = i_{k+1} i_{k-1} \tag{3.38}$$

Squaring both the sides of Equation 3.36 and performing trigonometric manipulations, Equation 3.39 can be obtained.

$$\frac{I_p^2}{2}\left\{ 1 - \cos 2(\omega_0 t + \theta_i) \right\} = i_k^2 \tag{3.39}$$

Subtracting Equation 3.38 from Equation 3.39 yields the following expression.

$$\frac{I_p^2}{2}\left\{ 1 - cos\, 2(\omega_0 \Delta T) \right\} = i_k^2 - i_{k+1}\, i_{k-1} \tag{3.40}$$

Since $1 - cos\, 2(\omega_0 \Delta T) = 2sin^2(\omega_0 \Delta T)$, Equation 3.40 can be rewritten as

$$I_p^2 \sin^2(\omega_0\, \Delta\, T) = i_k^2 - i_{k+1} i_{k-1}. \tag{3.41}$$

Similarly it can be shown that

$$2I_p V_p cos\,(\theta_v - \theta_i) \sin^2(\omega_0 \Delta T) = 2 v_k i_k - v_{k+1} i_{k-1} - v_{k-1} i_{k+1} \tag{3.42}$$

and

$$I_p V_p sin\,(\theta_v - \theta_i) \sin(\omega_0 \Delta T) = v_k i_{k+1} - v_{k+1} i_k . \tag{3.43}$$

Dividing Equation 3.42 by Equation 3.41 and Equation 3.43 by Equation 3.41 provide the following equations for the resistance, r_f, and the reactance, x_f.

$$r_f = \frac{V_p}{I_p}\cos(\theta_v - \theta_i) = \frac{2\, v_k i_k - v_{k+1} i_{k-1} - v_{k-1} i_{k+1}}{2\left(i_k^2 - i_{k+1} i_{k-1} \right)} \tag{3.44}$$

$$x_f = \frac{V_p}{I_p} sin\,(\theta_v - \theta_i) = \frac{v_k i_{k+1} - v_{k+1} i_k}{i_k^2 - i_{k+1} i_{k-1}} \sin(\omega_0 \Delta T) \tag{3.45}$$

3.2.1.3 Least Error Squares Approach

Luckett et al [16], proposed the use of the least error squares approach for computing the peak values and phase angles of voltage and current phasors. Brooks [17], also used the least error squares approach in which he assumed that the input signal is composed of a DC component and a fundamental frequency component. Sachdev and Baribeau [18], reported further developments in this approach and demonstrated that a major part of the computations can be performed off-line to develop the required algorithm. The procedure to implement the least error squares technique comprises of the following steps.

(1) Select a suitable model for representing the signal.
(2) Linearize the model.
(3) Select a sampling rate.
(4) Select a data window size.
(5) Express the process in the matrix form.
(6) Determine the left pseudoinverse of the coefficients matrix.
(7) Estimate the values of the elements in the vector of unknowns.

A voltage signal can be described by a mathematical model of the form of Equation 3.46. The model represents a voltage that contains a decaying DC and a fundamental frequency component and its harmonics.

$$v(t) = K_0 e^{\frac{-t}{\tau}} + \sum_{m=1}^{N} K_m sin\,(m\,\omega_0 t + \theta_m) \tag{3.46}$$

where

K_0 is the magnitude of the decaying DC component at t=0,
τ is the time constant of the DC component,
K_m is the magnitude of the m^{th} harmonic component and

θ_m is the phase angle of the m^{th} harmonic component.

Assume that a voltage is composed of the fundamental frequency and second harmonic components, and a decaying DC component. This voltage can be defined by the following equation.

$$v(t) = K_0 e^{\frac{-t}{\tau}} + K_1 sin(\omega_0 t + \theta_1) + K_2 sin(2\omega_0 t + \theta_2) \tag{3.47}$$

The exponential term can be replaced by its Taylor series expansion. If the first two terms of the series are used to represent the exponential, $sin(\omega_0 t + \theta_1)$ is expanded to $\{\cos\theta_1\ sin(\omega_0 t) + \sin\theta_1\ cos(\omega_0 t)\}$ and $\sin(2\omega_0 t + \theta_2)$ is replaced by $\{\cos\theta_2\ sin(2\omega_0 t) + \sin\theta_2\ cos(2\omega_0 t)\}$, the following equation can be obtained.

$$v(t) = K_0 - \frac{K_0 t}{\tau} + K_1 \cos\theta_1\ sin(\omega_0 t) + K_1 \sin\theta_1\ cos(\omega_0 t)$$

$$+ K_2 \cos\theta_2\ sin(2\omega_0 t) + K_2 \sin\theta_2\ cos(2\omega_0 t) \tag{3.48}$$

At time $t = t_1$, the equation can be represented as follows:

$$v(t_1) = K_0 - \frac{K_0 t_1}{\tau} + K_1 \cos\theta_1\ sin(\omega_0 t_1) + {}_1 \sin\theta_1\ cos(\omega_0 t_1)$$

$$+ K_2 \cos\theta_2\ sin(2\omega_0 t_1) + K_2 \sin\theta_2\ cos(2\omega_0 t_1) \tag{3.49}$$

This equation can be written in the following linear form.

$$v(t_1) = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 + a_{16}x_6 \tag{3.50}$$

where

$$x_1 = K_0 \qquad a_{11} = 1$$

$$x_2 = \frac{-K_0}{\tau} \qquad a_{12} = t_1$$

$$x_3 = K_1 \cos\theta_1 \qquad a_{13} = sin\,(\omega_0 t_1)$$

$$x_4 = K_1 \sin\theta_1 \qquad a_{14} = cos\,(\omega_0 t_1)$$

$$x_5 = K_2 \cos\theta_2 \qquad a_{15} = sin\,(2\omega_0 t_1)$$

$$x_6 = K_2 \sin\theta_2 \qquad a_{16} = cos\,(2\omega_0 t_1)$$

Assume that the voltage is sampled at intervals of ΔT seconds, the next sample received at time $t_2 = (t_1 + \Delta T)$ can be described as follows:

$$v\,(t_1 + \Delta T) = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 + a_{26}x_6 \tag{3.51}$$

For a preselected time reference and a preselected sampling rate, the values of the a coefficients become specified. If p samples are taken, the equations can be written in the following matrix form.

$$\underset{p \times 6}{[A]}\ \underset{6 \times 1}{[X]} = \underset{p \times 1}{[V]} \tag{3.52}$$

For p greater than six, the vector of unknowns, $[X]$, can be determined by using the following equation.

$$\underset{6 \times 1}{[X]} = \underset{6 \times p}{[A]^{+}}\ \underset{p \times 1}{[V]} \tag{3.53}$$

In this equation, $[A]^{+}$ is the left pseudoinverse of $[A]$ and is defined as

$$\underset{6\times p}{[A]^{+}} = [\underset{6\times p}{[A]^{T}}\ \underset{p\times 6}{[A]}]^{-1}\ \underset{6\times p}{[A]^{T}}. \tag{3.54}$$

Since the elements of the matrix $[A]$ are known a priori, the elements of the left pseudoinverse matrix can be determined off-line. This substantially reduces the on-line calculations required to estimate the real and imaginary components of the fundamental and harmonic frequency phasors when the least error squares technique is used.

3.2.1.4 Others

McInnes and Morrison [19] used a differential equation approach for the distance protection of a transmission line. They used an R-L lumped-parameter model of the transmission line described by the following differential equation.

$$v = r_f i + l_f \frac{di}{dt} \tag{3.55}$$

where

r_f is the resistance of line seen from the relay location and
l_f is the inductance of line seen from the relay location.

Integrating the above equation over two successive time periods and solving the resulting simultaneous equations, the line parameters, r_f and l_f, can be obtained. Integration can be performed using a numerical technique, such as the trapezoidal rule. Ranjbar and Cory [20], suggested modifications by selecting overlapping limits of integration to suppress preselected low-order harmonics.

3.2.2 Recursive Algorithm

Algorithms described in Section 3.2.1, are of the nonrecursive type that use finite length windows and provide outputs that depend only on the data in the selected window. On the other hand, the output of a recursive algorithm depends on the present input and all previous inputs. The Kalman filter is one such approach. It filters as much of the noise as possible, while responding optimally to changes in the system states. The coefficients of the Kalman filter, called Kalman gains, are non-stationary. They optimize the square of the expected errors between the actual and the estimated values of the system states.

Girgis et al [21] and Dasgupta [22], applied the Kalman filtering technique in power system protection. Recently, Sachdev et al [23,24] explained the Kalman filtering technique in power system terminology and described a procedure for selecting suitable parameters for designing the Kalman filters. Based on the proposed procedure, they described designs of estimators for real and imaginary components of voltage and current phasors. They also extented the design procedure to include the presence of decaying DC and harmonic components in the inputs.

3.3 Algorithms for Frequency Estimation

Algorithms, studied in Section 3.2, estimate the peak values and phase angles of phasors representing power system voltages and currents of the nominal frequency. Recently, some algorithms have been developed for estimating the frequency at power system buses. Sachdev and Giray [25] used a digital computer to determine the frequency of the system voltage. The technique detects zero-crossings, measures the time between zero-crossings that are integral numbers of cycles apart and then estimates the frequency. Nguyen and Srinivasan [26] extended the zero-crossing approach by estimating the durations between the instantaneous voltages of equal magnitudes experienced one cycle apart. The voltage is sampled at a prespecified rate. A new estimate of the time period of a cycle is obtained at each sampling instant. Girgis and Ham [27,28], suggested a fast Fourier transform (FFT) based method for detecting changes in the power system frequency. Phadke et al [29] used the discrete Fourier transform for estimating the frequency at a power system bus. They estimate the phasors representing the positive sequence components of the bus voltages and then estimate the deviations of the frequency from its nominal value. Recently, Sachdev and Giray [30] used the least error squares technique to develop an algorithm that simultaneously estimates the amplitude and frequency of a voltage signal. These algorithms are briefly described in this section.

3.3.1 Frequency Estimation from Time Measurements

Sachdev and Giray [25] measured the time period of an input waveform for an integral number of cycles using a high frequency clock and then calculated the frequency of the signal by using the following equation.

$$f = \frac{pf_c}{N_c} \tag{3.56}$$

where

f_c is the frequency of clock and

N_c is the number of high frequency pulses counted in p cycles.

This method requires that a prespecified number (the product of p and f_c) be divided by a variable count, N_c. The frequency can also be evaluated from an equation derived by using the Taylor series expansion of $\frac{1}{f}$ as follows:

$$\frac{1}{f} = \frac{1}{f_0} - \frac{(f - f_0)}{f_0^2} + \frac{(f - f_0)^2}{f_0^3} - \dots \tag{3.57}$$

where

f_0 is the nominal frequency.

Substituting for f from Equation 3.56, considering the first two terms of the equation and rearranging the resulting equation provides

$$f = \frac{f_0^2}{pf_c}\left\{\frac{2pf_c}{f_0} - N_c\right\}. \tag{3.58}$$

The values of p and f_c in this equation can be selected such that $\{f_0^2/pf_c\}$ has a value of 10^{-k}, k being an integer. Also $\{2pf_c/f_0\}$ is a constant. The difference between the variable count N_c and the constant $\{2pf_c/f_0\}$ directly yields the frequency times 10^k. In this manner, the resolution of the frequency estimation can be controlled. Also, a division (in Equation 3.56) is replaced by a subtraction that is computationally a simpler operation than a division.

It can be shown that if the first three terms of the Taylor series expansion of $\frac{1}{f}$ are used, the frequency can be determined using the following equation.

$$f = \frac{3}{2} f_0 - \frac{f_0}{2} \sqrt{\frac{4f_0}{pf_c} N_c - 3} \tag{3.59}$$

The accuracy of the frequency estimates obtained by this equation is better than the one obtained by using Equation 3.58. However, the use of Equation 3.59 requires that a square root of a function of N_c be calculated; this is more complex and requires more computation time compared to the approach in Equation 3.58.

3.3.2 Tracking of Frequency Deviations Based on Level Crossings

Reference [26] presented a technique that extends the zero-crossing approach for computing the deviations of frequency from its nominal value. The computations of the time periods from the zero-crossings are supplemented by computations of the time periods from instantaneous voltages of equal magnitudes. A weighting function is used to obtain a "best estimate" of the time period of the signal.

Figure 3.3 illustrates a sinusoidal voltage of the nominal frequency. If the voltage is sampled N times in a cycle, and each sampling interval is ΔT seconds, $N \Delta T$ would be the time period, T, of the voltage. The instantaneous values of two voltage samples taken $N \Delta T$ seconds apart would be equal if the signal is not distorted and is free of noise. However, if the frequency deviates from its nominal value, the samples taken $N \Delta T$ seconds apart would no longer be equal. In such cases, the time period, T, can be estimated using linear interpolation and three samples, v_{k-N}, v_{k-1} and v_k as shown in Figure 3.4. The estimate can be mathematically expressed as

$$T = N \Delta T - \frac{v_k - v_{k-N}}{v_k - v_{k-1}} \Delta T. \tag{3.60}$$

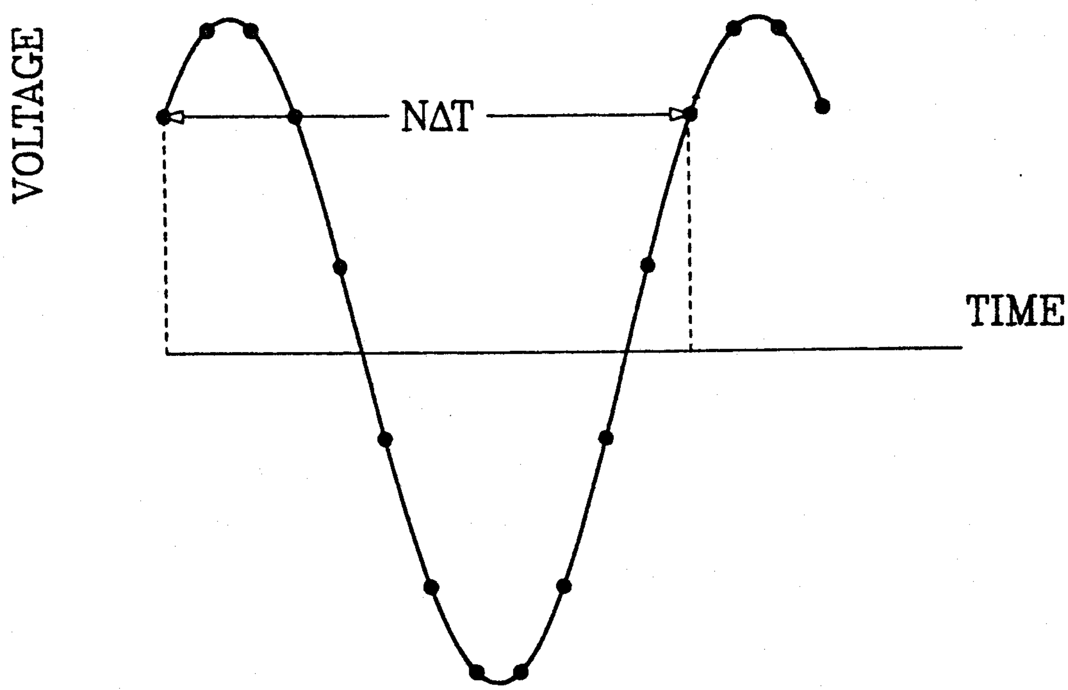


Figure 3.3 A sinusoidal voltage of nominal frequency sampled at intervals of ΔT seconds.

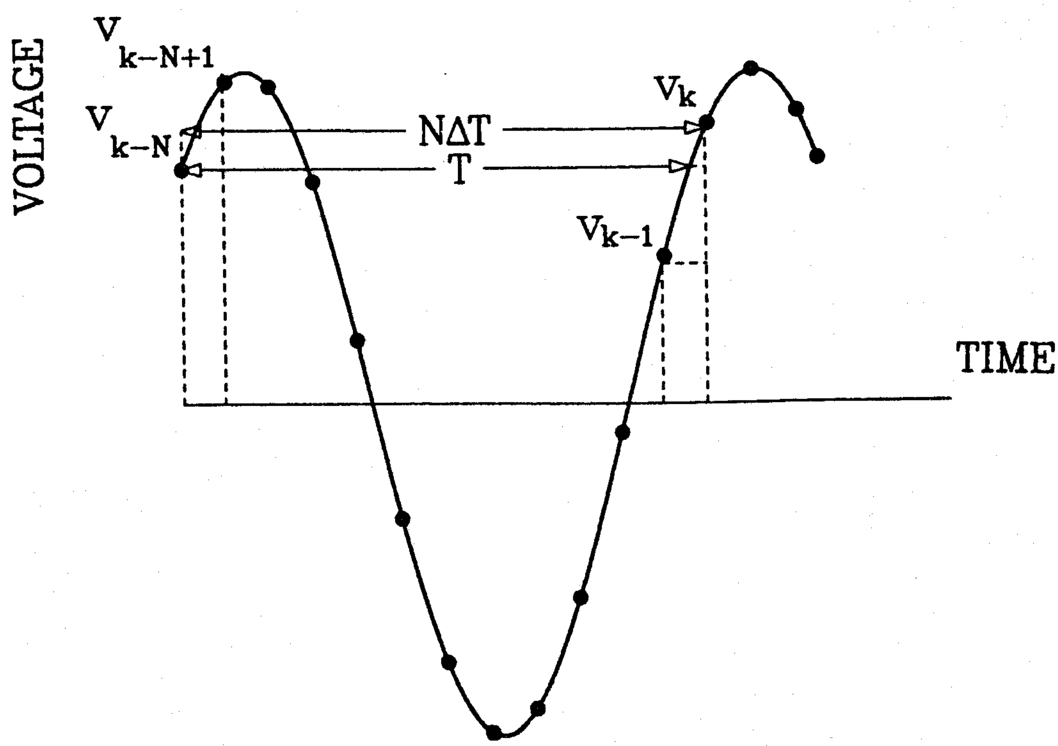


Figure 3.4 A sinusoidal voltage of off-nominal frequency sampled at intervals of ΔT seconds.

Another estimate of T can be obtained by considering the samples v_{k-N}, v_{k-N+1} and v_k. This estimate can be mathematically expressed as

$$T = N\Delta T - \frac{v_k - v_{k-N}}{v_{k-N+1} - v_{k-N}} \Delta T. \tag{3.61}$$

Averaging the two estimates provides Equation 3.62. The frequency of the voltage can now be estimated from this equation by computing $\{ \frac{1}{T(k)} \}$.

$$T(k) = N\Delta T - \frac{(v_k - v_{k-N})\,(v_k - v_{k-1} + v_{k-N+1} - v_{k-N})}{2(v_k - v_{k-1})(v_{k-N+1} - v_{k-N})} \Delta T \tag{3.62}$$

where

$T(k)$ is the average time period.

Using the described procedure, a new estimate of the period of a cycle can be obtained at each sampling instant. However, the accuracy of these estimates is not consistent. The estimates are more accurate near the zero crossings and less accurate near the peaks. To get a composite best estimate, a weighted mean of several individual estimates of the deviations of the time period $\{ N\Delta T - T(k) \}$ can be computed. The authors of Reference [26] used weights that are defined as

$$W(k) = (v_k - v_{k-1})(v_{k-N+1} - v_{k-N}). \tag{3.63}$$

The weights determined in this manner have minimum values near the peaks and maximum values near the zero-crossings. The weighted mean of m estimates of $T(k)$ can be expressed as follows:

$$T(m,k) = \frac{\sum_{k=1}^{m} T(k)W(k)}{\sum_{k=1}^{m} W(k)} \tag{3.64}$$

For three phase voltages, a composite estimate can be obtained as follows:

$$T^{a,b,c}(m,k) = \frac{\sum_{p=a,b,c} T_p(k) W_p(k)}{\sum_{p=a,b,c} W_p(k)}. \tag{3.65}$$

where

a, b and c refer to the three phases of the power system.

3.3.3 Fast Fourier Transform Based Algorithm

The fast Fourier transform (FFT) is an efficient method for computing a discrete Fourier transform (DFT). One of the major drawbacks of the FFT is the leakage effect [27] that is associated with its use. The leakage is observed when a time sequence is truncated such that a fraction of a cycle exists in the waveform that is subjected to the FFT. However, no leakage occurs if an integral number of cycles exist in the truncated time sequence. Girgis and Ham [28] used the phenomenon of leakage to detect changes in the frequency of a power system. The method considers that a sinusoidal voltage is sampled N times in a cycle of the nominal frequency and an FFT is performed using the sampled values. Estimates of the apparent amplitudes of the components of $\{ \frac{N}{2} - 1 \}$ frequencies are used to obtain the "leakage coefficient". The procedure can be mathematically expressed as

$$\eta = \frac{\left[\sum_{n=0}^{\frac{N}{2}-1} | V(n) | \right] - |V(1)|}{|V(1)|} \tag{3.66}$$

where

η is the leakage coefficient,
$V(n)$ denotes the DFT obtained from a set of N samples and
n is the order of frequency component.

The leakage coefficient is zero when the sampled signal is of the nominal frequency and consists of the fundamental frequency only. If the frequency deviates, the leakage coefficient will cease to be zero.

Figure 3.5 [28] shows a first order least error squares fit of the leakage coefficient versus deviation of the frequency from 60 Hz. In this case, the signal was sampled at 19.2 KHz. If the sampling starts at the positive going zero-crossing, the relationship between the leakage coefficient and the frequency deviation is found to be linear over the 0-5 Hz range. Using the slope of the line, the magnitude of the frequency deviation, Δf , can be estimated as follows:

$$|\Delta f| = \frac{\eta}{m}. \tag{3.67}$$

where

m is the slope of line for range of 5 Hz.

To determine whether the frequency of the signal is more than or is less than the nominal value, the real part of $V(1)$ is examined. If the real part of $V(1)$ is less than zero, it is concluded that the frequency has decreased. If the real part is more than zero, it is concluded that the frequency has increased.

The relationship between the leakage coefficient and the frequency deviation is non-linear if sampling of the input signal does not start at (or immediately after) the positive going zero-crossing. Therefore, a zero-crossing detector (ZCD) is used to start the sampling process at the positive zero-crossings. Since a ZCD is not a perfect device, some errors can occur in detecting the zero-crossings. This in turn affects the slope of the leakage coefficient versus frequency deviation characteristic. Figure 3.6 [28] shows that the slope of the line essentially remains constant when the zero-crossing detection errors are in the range of 0-5 degrees. Therefore, the frequency estimation is insensitive to small errors in zero-crossing detections.

3.3.4 Discrete Fourier Transform Algorithm

Reference [29] proposed a technique that performs a discrete Fourier transform (DFT) on a set of voltage samples for estimating the frequency of a three phase power system. The positive sequence voltage phasor is first estimated from the phasors of the three phase voltages. The frequency deviation from its nominal value is then estimated.

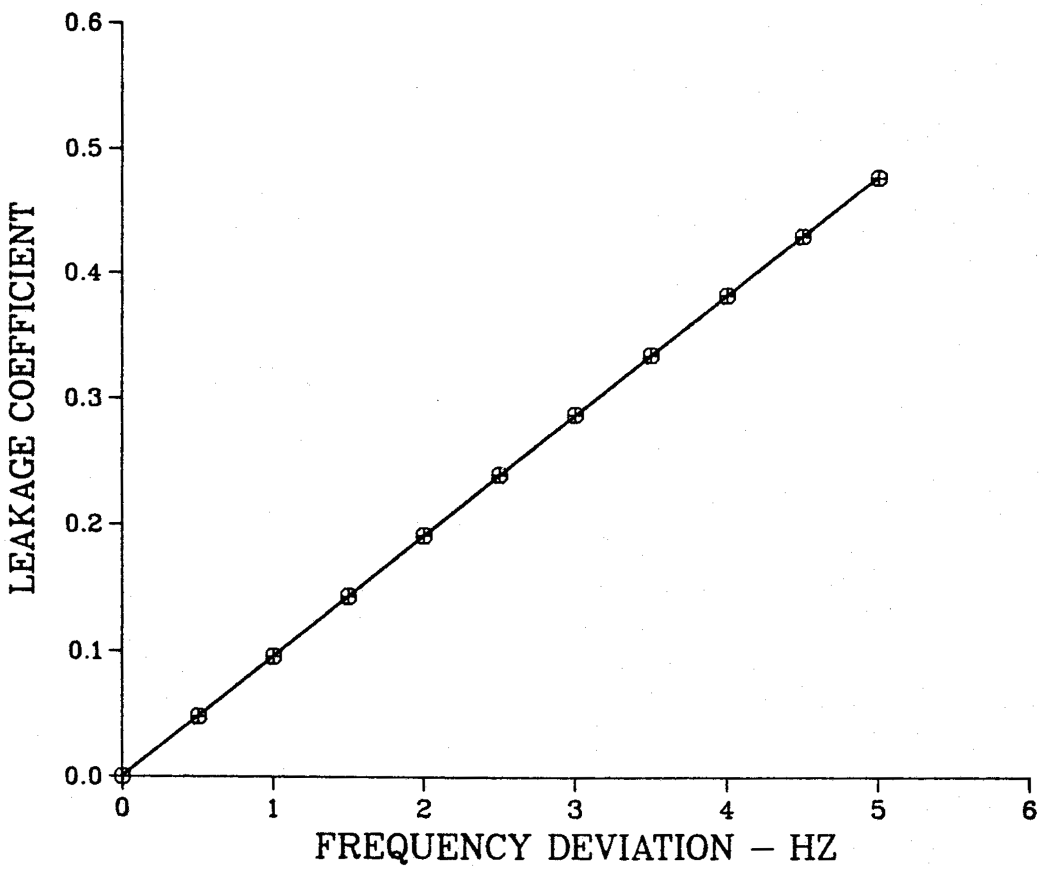


Figure 3. 5 The leakage coefficient versus Δf for sine wave for N=32.

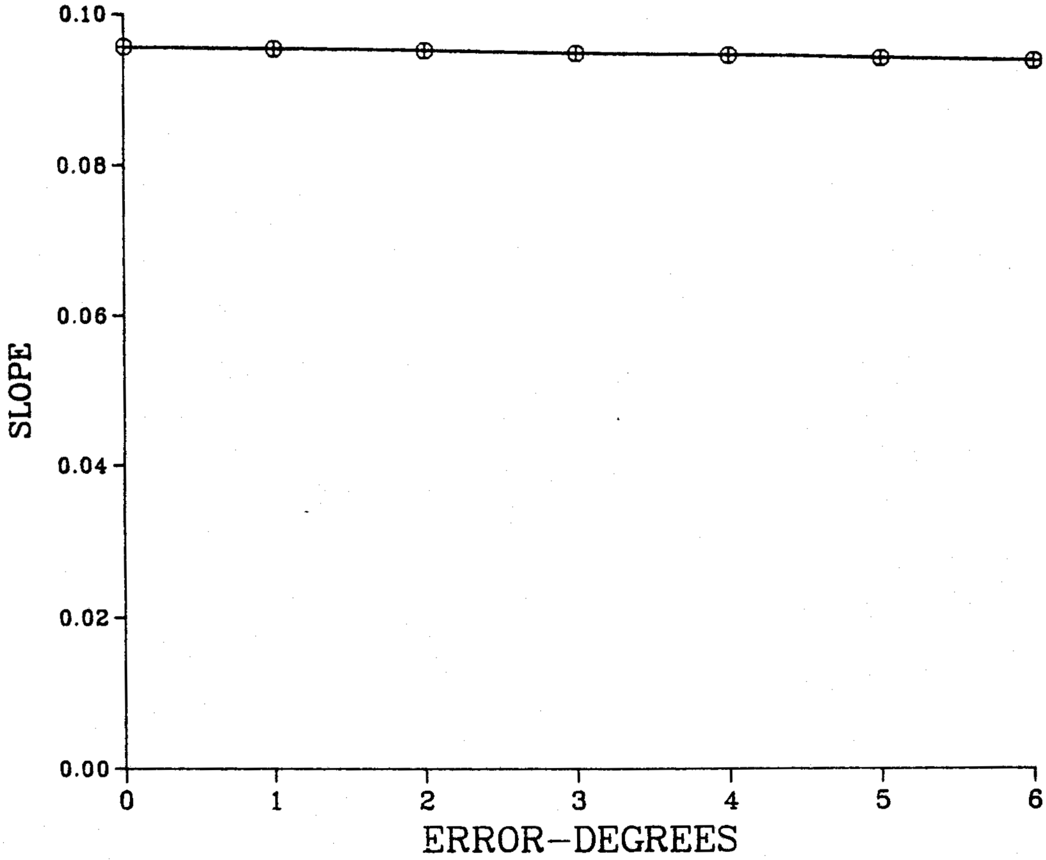


Figure 3.6 The slope of the leakage coefficient versus zero crossing detection errors.

The DFT based algorithm considers that a sinusoidal voltage, v, is sampled $\{ f_o N \}$ times per second. If the frequency of voltage is f Hz, the k^{th} sample of the voltage can be described by the following equation.

$$v_k = \sqrt{2}\, V sin(\frac{2\pi k}{N} + \theta_v). \tag{3.68}$$

where

V is the rms value of voltage and

θ_v is the phase angle of the voltage at t=0.

The Discrete Fourier Transform of $\{ V_k \}$ contains a fundamental frequency component that is given by

$$V(1) = (\frac{2}{N}) \sum_{k=0}^{N-1} v_k \, e^{-j(\frac{2\pi}{N})k} . \tag{3.69}$$

Expressing the exponential term in the cosine and sine forms and substituting in Equation 3.69, the following equation can be obtained.

$$V(1) = (\frac{2}{N}) \sum_{k=0}^{N-1} v_k \cos(\frac{2\pi}{N} k) - (\frac{2}{N}) j \sum_{k=0}^{N-1} v_k \sin(\frac{2\pi}{N} k)$$

$$= V_c - jV_s . \tag{3.70}$$

where

$$V_c = (\frac{2}{N}) \sum_{k=0}^{N-1} v_k \cos(\frac{2\pi}{N} k) \quad \text{and}$$

$$V_s = (\frac{2}{N}) j \sum_{k=0}^{N-1} v_k \sin(\frac{2\pi}{N} k).$$

The rms phasor representation of a sinusoidal voltage is related to the fundamental frequency component of its DFT by

$$\bar{v} = \frac{1}{\sqrt{2}} jV(1) = \frac{1}{\sqrt{2}} (V_s + jV_c). \tag{3.71}$$

Equation 3.69 uses the sample set $\{v_k, k=0, \cdots, N-1\}$. The data window containing these samples can be classified as data window #1. A new sample is obtained after an elapsed time of ΔT seconds. This time corresponds to the angle $\frac{2\pi}{N}$ radians. Now data window #2 contains the sample set $\{v_k, k=1, \cdots, N\}$. In general, the data, $\{v_k, k=r-1, \cdots, N+r-2\}$, is in the r^{th} window. The discrete Fourier transform of this data provides

$$V_c^{(r)} = (\frac{2}{N}) \sum_{k=0}^{N-1} v_{k+r-1} \cos(\frac{2\pi}{N}k) \tag{3.72}$$

$$V_s^{(r)} = (\frac{2}{N}) \sum_{k=0}^{N-1} v_{k+r-1} sin(\frac{2\pi}{N}k). \tag{3.73}$$

The rms phasor representing the sinusoidal voltage can now be expressed as

$$\bar{v}^{(r)} = \frac{1}{\sqrt{2}} (V_s^{(r)} + jV_c^{(r)}). \tag{3.74}$$

It can also be shown that the phasor obtained from the r^{th} window is related to the phasor obtained from the $(r-1)^{th}$ window as follows:

$$\bar{v}^{(r)} = \bar{v}^{(r-1)} e^{j\frac{2\pi}{N}} \tag{3.75}$$

Equation 3.75, shows that the rms phasor obtained using the DFT rotates in the counter clockwise direction. Also this phasor advances by an angle of $\frac{2\pi}{N}$ when a new sample is incorporated in the DFT.

It is advantageous to calculate the phasor from the data window #1 using Equations 3.69 and 3.71. However, the data of window #2 can be processed using the Fourier coefficients advanced by a phase angle of $\frac{2\pi}{N}$ radians as follows:

$$V_c^{(2),\,\frac{2\pi}{N}} = (\frac{2}{N})\sum_{k=0}^{N-1} v_{k+1}\cos(\frac{2\pi}{N}k + \frac{2\pi}{N}) \tag{3.76}$$

$$V_s^{(2),\,\frac{2\pi}{N}} = (\frac{2}{N})\sum_{k=0}^{N-1} v_{k+1}\sin(\frac{2\pi}{N}k + \frac{2\pi}{N}). \tag{3.77}$$

Equations 3.76 and 3.77 can be rewritten as

$$V_c^{(2),\,\frac{2\pi}{N}} = \left\{(\frac{2}{N})\sum_{k=0}^{N-1} v_k \cos(\frac{2\pi}{N}k)\right\} + (\frac{2}{N})(v_N - v_0)\cos(2\pi) \tag{3.78}$$

$$V_s^{(2),\,\frac{2\pi}{N}} = \left\{(\frac{2}{N})\sum_{k=0}^{N-1} v_k \sin(\frac{2\pi}{N}k)\right\} + (\frac{2}{N})(v_N - v_0)\sin(2\pi). \tag{3.79}$$

The voltage phasor computed from data window #2 and the Fourier coefficients can now be expressed as

$$\overline{v}^{(2)} = \overline{v}^{(1)} + j\frac{1}{\sqrt{2}}(\frac{2}{N})(v_N - v_0)\, e^{-j2\pi}. \tag{3.80}$$

In general, the phasor using the r^{th} window can be computed from the phasor computed from the $(r-1)^{th}$ window as follows:

$$\overline{v}^{(r)} = \overline{v}^{(r-1)} + j\frac{1}{\sqrt{2}}(\frac{2}{N})(v_{N+r-2} - v_{r-2})\, e^{-j\frac{2\pi}{N}(r-2)} \qquad (3.81)$$

The phasor representing the positive sequence voltage, $\overline{v}_1$, can be obtained from the phasor voltages using the equation

$$\overline{v}_1 = \frac{1}{3}\,(\overline{v}_a + \alpha\,\overline{v}_b + \alpha^2\,\overline{v}_c) \qquad (3.82)$$

where

a, b and c refer to the three phases of the power system and
α is an operator given by $1\underline{|120^{\circ}}$.

Using Equations 3.81 and 3.82, the following recursive equation can be derived.

$$\overline{v}_1^{(r)} = \overline{v}^{(r-1)} + j\,(\frac{2}{N})\frac{1}{\sqrt{2}}\frac{1}{3}\Bigg[(v_{a,N+r-2} - v_{a,r-2})\, e^{-j\frac{2\pi}{N}(r-2)} + (v_{b,N+r-2} - v_{b,r-2})\, e^{-j\frac{2\pi}{N}(n+r-2)} + (v_{c,N+r-2} - v_{c,r-2})\, e^{-j\frac{2\pi}{N}(2n+r-2)}\Bigg] \qquad (3.83)$$

where

n is an integer and is equal to $\frac{N}{3}$.

When the recursive Equation 3.83 is used to calculate the phasor, the positive sequence voltage phasor remains stationary in the complex plane. When the frequency deviates from its nominal value, the phasor rotates with an angular velocity that is given by

$$\frac{d\phi}{dt} = \frac{\phi_r - \phi_{r-1}}{1/(f_0 N)} \quad radians/sec. \tag{3.84}$$

Substituting $2\pi\Delta f$ for the angular velocity $\frac{d\phi}{dt}$, Equation 3.84 can be rewritten in the following form.

$$\Delta f = \frac{\phi_r - \phi_{r-1}}{2\pi(1/f_0 N)} \quad Hz. \tag{3.85}$$

The frequency deviation is positive when the phasor rotates in the counter clockwise direction and is negative when the phasor rotates in the clockwise direction.

3.3.5 A Least Error Squares Technique

This section presents an algorithm that measures simultaneously the amplitude and the frequency of a voltage signal [30]. It assumes that the system frequency does not change during the time of the data window used for the measurements. The algorithm is developed using the least error squares approach.

The frequency at a power system bus is not required to be measured during a system fault. Therefore, it can be assumed that the input signal is a sinusoidal voltage that can be expressed as

$$v = V_m \sin(2\pi f t + \theta_v) \tag{3.86}$$

where

V_m is the peak value of the voltage,
f is the frequency of the voltage,
t is the time in seconds and
θ_v is the phase angle of the voltage at t=0.

Performing a trigonometric manipulation, Equation 3.86 can be expanded to obtained the following equation.

$$v = V_m \cos\theta_v \; \sin(2\pi f t) + V_m \sin\theta_v \; cos\,(2\pi f t). \tag{3.87}$$

The terms, $sin\,(2\pi f t)$ and $cos\,(2\pi f t)$, can be expanded in the neighborhood of the nominal frequency using the Taylor series. Substituting the first three terms of the expansions in Equation 3.87 and rearranging, the following equation can be obtained.

$$\begin{aligned} v = &\left\{ sin\,(2\pi f_0 t) \right\} V_m \cos\theta_v + \left\{ 2\pi t \;\; cos\,(2\pi f_0 t) \right\} (f - f_0) V_m \cos\theta_v \\ &+ \left\{ cos\,(2\pi f_0 t) \right\} V_m \sin\theta_v + \left\{ -2\pi t \;\; sin\,(2\pi f_0 t) \right\} (f - f_0) V_m \sin\theta_v \\ &+ \left\{ \frac{-(2\pi t)^2}{2} sin\,(2\pi t) \right\} (f - f_0)^2 V_m \cos\theta_v \\ &+ \left\{ \frac{-(2\pi t)^2}{2} cos\,(2\pi t) \right\} (f - f_0)^2 V_m \sin\theta_v \end{aligned} \tag{3.88}$$

Making the following substitutions in Equation 3.88, Equation 3.89 can be obtained.

$$a_{11} = sin\,(2\pi f_0 t) \qquad x_1 = V_m \cos\theta_v$$

$$a_{12} = 2\pi t \;\; cos\,(2\pi f_0 t) \qquad x_2 = \Delta f \;\; V_m \cos\theta_v$$

$$a_{13} = cos\,(2\pi f_0 t) \qquad x_3 = V_m \sin\theta_v$$

$$a_{14} = -2\pi t \;\; sin\,(2\pi f_0 t) \qquad x_4 = \Delta f \;\; V_m \sin\theta_v$$

$$a_{15} = -2(\pi t)^2 sin\,(2\pi f_0 t)' \qquad x_5 = \Delta f^2 V_m \cos\theta_v$$

$$a_{16} = -2(\pi t)^2 cos\,(2\pi f_0 t) \qquad x_6 = \Delta f^2 V_m \sin\theta_v$$

where

$$\Delta f = f - f_0$$

$$v = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 + a_{16}x_6 \tag{3.89}$$

The time t, can be arbitrarily selected and, therefore, the values of the a coefficients can be predetermined in an off-line mode. In this manner, the a's in Equation 3.89 are the known coefficients while all the x's are unknowns that are functions of V_m, θ_v and Δf. Since there are six unknowns in Equation 3.89, at least six equations are required to determine them. A voltage sample taken at time t_1 provides one equation. Subsequent five samples, taken at intervals of ΔT seconds, provide the other five equations. If p samples are taken, the process takes the form of p equations in six unknowns. These equations can be expressed in the matrix form as follows:

$$\underset{p\times 6}{[A]}\ \underset{6\times 1}{[X]} = \underset{p\times 1}{[V]} \tag{3.90}$$

For p greater than six, the vector of unknowns, $[X]$, can be determined by using the following equation.

$$\underset{6\times 1}{[X]} = \underset{6\times p}{[A]^+}\ \underset{p\times 1}{[V]} \tag{3.91}$$

where

$[A]^+$ is the left pseudoinverse of $[A]$ as defined in Equation 3.24.

Elements of the vector $[X]$, can be used to estimate the frequency deviation and the amplitude of the voltage. The frequency deviation can be estimated using any one of the following equations.

$$\Delta f = \frac{x_2}{x_1} = \frac{\Delta f\ V_m \cos\theta_v}{V_m \cos\theta_v} \tag{3.92}$$

$$\Delta f = \frac{x_4}{x_3} = \frac{\Delta f\ V_m \sin\theta_v}{V_m \sin\theta_v} \tag{3.93}$$

$$\Delta f = \frac{x_5}{x_2} = \frac{\Delta f^2 V_m \cos\theta_v}{\Delta f \ V_m \cos\theta_v} \tag{3.94}$$

$$\Delta f = \frac{x_6}{x_4} = \frac{\Delta f^2 V_m \sin\theta_v}{\Delta f \ V_m \sin\theta_v} \tag{3.95}$$

$$\Delta f = \frac{x_2 + x_4}{x_1 + x_2} = \frac{\Delta f \ V_m \cos\theta_v + \Delta f \ V_m \sin\theta_v}{V_m \cos\theta_v + V_m \sin\theta_v} \tag{3.96}$$

$$\Delta f = \frac{|\, x_2 \,| + |\, x_4 \,|}{|\, x_1 \,| + |\, x_3 \,|} = \frac{|\, \Delta f \ V_m \cos\theta_v \,| + |\, \Delta f \ V_m \sin\theta_v \,|}{|\, V_m \cos\theta_v \,| + |\, V_m \sin\theta_v \,|} \tag{3.97}$$

$$\Delta f = \left\{ \frac{x_2^2 + x_4^2}{x_1^2 + x_3^2} \right\}^{\frac{1}{2}} = \left\{ \frac{\left\{ \Delta f \ V_m \cos\theta_v \right\}^2 + \left\{ \Delta f \ V_m \sin\theta_v \right\}^2}{\left\{ V_m \cos\theta_v \right\}^2 + \left\{ V_m \sin\theta_v \right\}^2} \right\}^{\frac{1}{2}} \tag{3.98}$$

When the value of $V_m \cos\theta_v$ is very small, the frequency estimated by Equation 3.92 will not be accurate. Similarly, when the value of $V_m \sin\theta_v$ is small, frequency estimated by Equation 3.93 will not be accurate. However, when the value of $V_m \cos\theta_v$ is small, the value of $V_m \sin\theta_v$ is close to 1 p.u. and vice-versa. The estimate using one of these equations is, therefore, expected to have acceptable accuracy. In Equations 3.94 and 3.95, the value of the denominator would be very small when the system frequency is either equal to or close to the nominal value. Therefore, Equations 3.94 and 3.95 are not suitable for estimating the frequency deviations. The denominator of Equation 3.96 would be small at certain instants of time and at these times this equation will not provide accurate estimates of frequency deviations. Equation 3.97 is similar to Equation 3.96 except that the absolute values are used to alleviate the deficiency observed in Equation 3.96. However, the calculated values of Δf would always be positive. The sign of Δf can be, however, determined by comparing the numerator and denominator of Equations 3.92 and 3.93. Both, the numerator and the denominator would have the same signs, if Δf is positive and would be of the opposite signs if Δf is negative. The sign of Δf obtained also from Equation 3.98 is always positive. However, the procedure described for Equation 3.97 can be followed to determine the nature of frequency deviation. The major disadvantage of Equation 3.98 is that the computations required are more as compared to the other equations.

The amplitude of the voltage can, however, be estimated by using the Equation 3.99.

$$V_m = \sqrt{x_1^2 + x_3^2} \tag{3.99}$$

3.4 Summary

The algorithms that estimate the peak values and phase angles of voltage and current signals have been briefly described in this chapter. A few newly developed techniques for estimating the power system frequency have also been briefly presented.

Chapter 4

SELECTION CRITERION FOR RELAY ALGORITHMS

4.1 Introduction

Digital relay algorithms proposed in the past have been reviewed in Chapter 3. The algorithms can be implemented on micro-processors to estimate the electrical parameters of power systems and make decisions from those estimates. The selection of an algorithm for use in a digital relay is basically governed by the available hardware and the manner in which the relay is expected to respond.

If the relay has to be designed using the selected hardware, a suitable algorithm and a sampling rate must be selected. The computational burden associated with the selected algorithm must be estimated to check the ability of the available processor to perform the required computations in a sampling period. If the processor can not perform the computations in the available time, the suitability of reducing the sampling rate should be investigated. The response of the relay using the selected algorithm and sampling rate should be evaluated. The algorithm should be used if the response is acceptable; otherwise, the possibility of the use of a different algorithm should be investigated.

In case the relay response is specified, an algorithm and a sampling rate that will provide the desired response must be chosen. Suitable hardware should be selected to meet the computational burden that is associated with the selected algorithm and the sampling rate. If the computational burden proves to be too large for the available micro-processors, the use of a different algorithm should be investigated.

The procedure outlined in this section is illustrated in Figure 4.1 and the details of the concepts are discussed in this chapter.

4.2 Capabilities of the Available Processor

In some situations, it may be required that a relay be developed using a specified micro-processor. This objective would dictate that the selected algorithm be suitable for implementation on the available processor. Under these circumstances, selecting the algorithm and programming it could be a challenging problem.

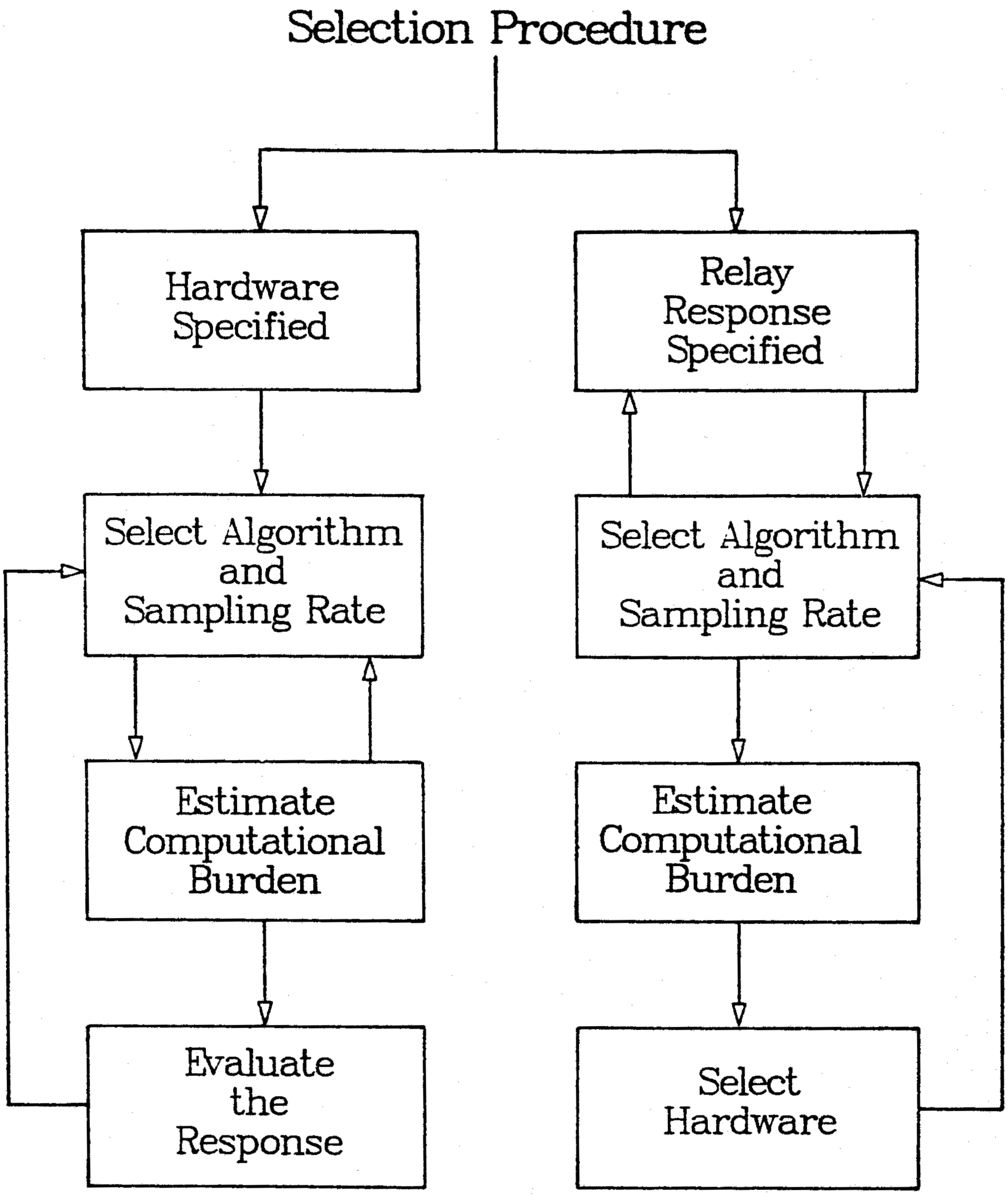


Figure 4.1 A procedure for selecting relay algorithms.

Algorithms are generally required to perform computations involving additions, subtractions, multiplications, divisions and logic functions. The necessary computations are repeated every time a new set of samples is received. In power system relaying applications, a set of computations must be completed within one inter-sampling interval. The processor speed and ability to perform arithmetic computations are, therefore, two critical factors in the selection of an algorithm.

4.3 Computational Burden

The computational time required by an algorithm depends on the type and the number of computations that must be performed. The number of computations, in turn, depends on the number of consecutive samples that have to be considered at a time. The algorithms based on the assumption of the inputs being sinusoids of a fundamental frequency, generally use short data windows of two to five consecutive samples. These algorithms, therefore, require fewer computations and involve less computational burdens on the micro-processors. The algorithms designed by assuming complex waveforms, use long data windows, eight or more consecutive samples, to extract the fundamental frequency component. It generally takes longer to perform the computations for such algorithms. An algorithm provides accurate results when the waveforms of the inputs are similar to those considered during the design of the algorithm. The power system voltages and currents experienced during faults are distorted and may contain components of several frequencies. The designs of short window algorithms generally do not consider the presence of non-fundamental frequency components in the input waveforms. Long window algorithms are, therefore, expected to provide more accurate results than the short window algorithms. It is important to appreciate that the computational time and accuracy of an algorithm are inter-dependent. In general, the algorithms that provide accurate results require more time to perform computations. Conversely, the results obtained by using algorithms that require fewer computations, may not have the desired accuracy. A computational burden and accuracy trade-off is, therefore, necessary and an algorithm should be selected after proper investigation.

4.4 Frequency Response

Under normal operating conditions, the system voltages and currents mainly consist of sinusoids of the fundamental frequency. However, the voltage and current waveforms experienced during faults are distorted and their nature can not be exactly predicted in advance. Fault currents may include exponentially decaying DC components. The magnitudes of these components depend on the pre-fault load currents, fault inception angles, and magnitudes of the post-fault currents.

Loads comprising of nonlinear-impedances produce harmonic components that are noticeable in voltage waveforms. Non-harmonic high-frequency components and noise are also present in the voltage and current waveforms. The distortion in the waveforms during faults results in the algorithms providing incorrect estimates. The errors in the estimates depend on the magnitudes of the undesired components and the response of the algorithm to these components. The frequency response of the algorithm, therefore, must be evaluated. A technique that can be used to obtain the frequency response of a digital filter, is discussed in Appendix A.

4.5 Dynamic Response

Relay algorithms operate on a data window that contains a set of samples. The data window is updated by taking new samples and discarding an equal number of old samples. The algorithms repeat the computations using the latest data window. In the pre-fault state, the algorithms yield identical and correct results. When a fault occurs, the data window starts incorporating the post-fault data. The computations that use both the pre-fault and post-fault samples can yield unpredictable results. The algorithms that operate on short data windows tend to yield poor results when data from pre-fault and post-fault states are used. However, long window algorithms are expected to provide a smooth transition from pre-fault to post-fault states. Therefore, the performance of algorithms must be investigated for their dynamic response.

4.6 Summary

A few of the criteria and trade-offs that are used for selecting relaying algorithms have been discussed in this chapter. An important factor that is associated with the selection of an algorithm is the selection of an optimum sampling rate. The use of a high sampling rate requires large computational burden because the computations must be performed in lesser time. The implementations of algorithms designed using high sampling rates may, therefore, require expensive hardware. On the other hand, simple and inexpensive hardware can be used for implementing algorithms with low sampling rates.

Chapter 5

INTERACTIVE SOFTWARE FOR EVALUATING DIGITAL RELAYING ALGORITHMS

5.1 Introduction

In the preceding chapters, digital relaying algorithms developed in the past and major factors that govern their selection have been discussed. The prime sources of errors, such as, A/D converters, micro-processors and relay software have also been identified.

An algorithm should be evaluated before using it in a digital relay. The performance of the algorithm must be evaluated for a variety of situations that might be encountered in a power system. The selection of inappropriate hardware, especially, the A/D converter and the micro-processor, will cause errors in the estimates of the electrical parameters. An A/D converter saturates when the input exceeds the specified range. This results in incorrect representations of the input that cause errors in the estimates. Errors are also introduced because word sizes of the A/D converters and micro-processors are finite. Therefore, the errors introduced during the computations by the selected A/D converter and micro-processor should be investigated.

The evaluation of an algorithm requires extensive computations. An interactive computer software, incorporating the existing algorithms, would provide a useful tool for selecting the most appropriate algorithm. The advantages of developing such a software are as follows:

(1) Man-computer interaction allows the software to be used effectually without the need of detailed program manuals.

(2) The results of intermediate computations can be displayed. If the results are not within acceptable limits, the evaluation process can be terminated.

This chapter describes an interactive software that evaluates the performance of relaying algorithms. The software has the facilities for simulating the algorithm and checking the errors due to the relay software and finite word sizes of A/D converters and micro-processors. In the first part of this chapter, specifications for the software are presented. The software structure is then outlined. Interactive features of the developed software are also discussed.

5.2 Background Work

A software package was previously developed by Cline and Sachdev [31]. That package comprised of four programs, INDATG.APL, PEAKVA.APL, ZVALUE.APL and PLOTER.FOR. INDATG.APL generates data that can be used to represent voltages and currents. The program can generate sampled data from the following components of power system voltages and currents.

(1) The fundamental frequency component.

(2) DC components.

(3) Harmonic components.

(4) Non-harmonic components.

(5) Random noise.

The sampled data representing the components that are expected in a power system voltage or current, can be computed and added to obtain the required signal.

PEAKVA.APL estimates the phasors representing components of a specified frequency from the sampled data. The program includes functions that simulate several existing algorithms.

The ZVALUE.APL computes the impedance of a transmission line as seen by a digital relay. An algorithm is first used to estimate the real and imaginary components of the phasors representing the voltages and currents at the relay location. Impedances are then estimated from the phasors. ZVALUE.APL also includes functions that simulate several algorithms for estimating the phasors.

PLOTER.FOR is a Fortran program that uses the plotting package (CALCOMP) and produces graphs from specially formatted files created by other programs.

The programs INDATG.APL, PEAKVA.APL and ZVALUE.APL are discussed in detail later in this chapter.

5.3 Drawbacks of the Previously Developed Software

The previously developed programs suffer from the following drawbacks.

(1) The three programs, INDATG.APL, PEAKVA.APL and ZVALUE.APL, were developed independently. The subroutines that could be shared by the programs, are not shared but are included in each program.

(2) The programs do not include facilities for simulating the performance of an algorithm when implemented on a digital device that uses words of a finite length.

(3) The programs do not check the validity of user's inputs. No provision is made for tracing back to a previous question or aborting a program before it is completely executed.

5.4 Reorganization and Extension of the Software

The intent of this work was to develop an interactive software package for evaluating the performance of algorithms that are suitable for estimating the phasors of power system voltages and currents and the algorithms that are suitable for estimating the frequency of power system voltages. To develop such a software, the following objectives were set for the present work.

(1) Revise the previously developed programs and rewrite them to alleviate the drawbacks and deficiencies mentioned in Section 5.3.

(2) Develop new programs to extend the software. The extension should include programs for

  (*i*) evaluating the performance of the algorithms that estimate the frequency of a power system voltage.

  (*ii*) calculating the frequency response of the algorithms.

The revision of the older programs and the development of the new programs should provide a software package whose specifications are outlined in the next section.

5.4.1 Software Specifications

The following specifications were selected for the software package.

(1) The software should be able to interact with the user. The software should be user friendly; it should check inputs for errors, allow the user to make corrections and guide the user through the programs with expressive and meaningful prompts.

(2) The software should include programs that can generate instantaneous values of sampled data to be used for testing or evaluating of algorithms.

(3) The software should incorporate the existing algorithms that estimate peak values and phase angles of the phasors representing power system voltages and currents.

(4) The software should allow the user to calculate the impedance of a transmission line as seen by a digital relay using algorithms selected by the user.

(5) Existing algorithms for estimating the frequency of power system voltages, should be incorporated in the software package.

(6) The software should allow the user to examine the frequency response of a selected algorithm.

(7) The software should facilitate the execution of the programs in two different modes. In one mode, all computations should be performed using the floating point representation of numbers. This mode will be useful for determining the extent of errors that arise from the design of an algorithm. In the other mode, the effects of the analog to digital converter, truncation or rounding and the finite word size of the micro-processor should be incorporated. All computations should be carried out as a digital processor would do. This mode of operation will be useful for simulating the performance of a selected algorithm when it is implemented on a micro-processor.

(8) Subroutines needed to perform identical functions in different programs must be written once and stored to form a separate library.

In general, the software should be written and documented in such a way that future programmers can interpret, alter, and add to the existing programs.

5.4.2 Programming Language

The previously developed programs were written in APL (A Programming Language). The same language was used for developing the new programs. This section highlights some of the reasons for using the APL as a programming language.

(a) Dynamic Memory

The computer storage for the data arrays is handled dynamically by the APL system. The size of an array can be changed as the computations progress. This frees the user from prior memory allocation for data arrays.

(b) Workspace Concept

A terminal connected to the APL system is said to be active. Associated with each active terminal is a fixed-size block of storage in the central computer, called an active workspace. The APL programs are not saved as individual files. The active workspace containing the programs is saved. The user can interrupt a session, store the workspace, and deactivate the terminal. Later, the user can connect to the system again, load the stored workspace and continue the session where it was left. The programs and subroutines in the APL are normally referred to as "functions".

(c) Matrix Operation

APL is a matrix oriented language. It provides many readily available functions for mathematical manipulation of matrices. The available functions are not scientific subroutines but are components of the APL command set. This feature eliminates the need of calling subroutines whenever matrix operations are required. APL is especially suitable for this software as most of the algorithms require matrix oriented computations.

(d) Ability to Handle Character Information

The user interactive software requires the handling of large amounts of character information. APL has ability to handle the character information with ease. APL variables can be used to represent the alphanumeric data with no special declarations. Arrays of character data can easily be manipulated and processed.

5.5 Software Structure

The software developed for this project is in ten APL workspaces. Initially, the software was developed on the DEC-2060 computing system, that was available at the University of Saskatchewan. Recently, a VAX-8600 computing system (VMS operating system, version 4.3), has replaced the DEC-2060. The software has been modified for use on the new computing system.

For a better understanding of the software, APL terminologies have been used in this section. For example, the word "function" refers to a program or a subroutine and the term "workspace" refers to a collection of programs and subroutines. The overall software structure comprising of nine APL workspaces, is shown in the Figure 5.1. The workspaces are:

(1) FVALUE.APL,

(2) ALLFVA.APL,

(3) PEAKVA.APL,

(4) ALLPKV.APL,

(5) ZVALUE.APL,

(6) FRQRES.APL,

(7) INDATG.APL,

(8) ALLCMP.APL,

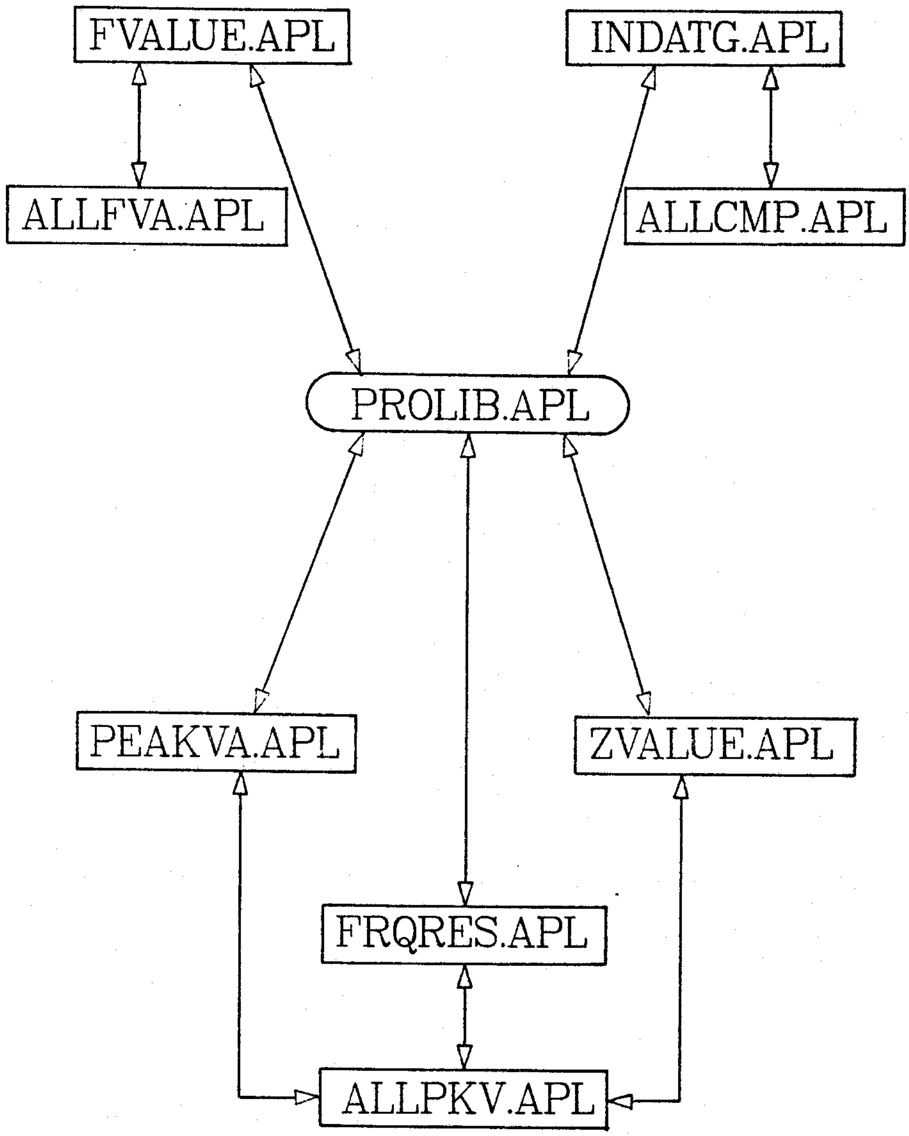


Figure 5.1 An overall structure of the software.

(9) PROLIB.APL.

The tenth workspace is INTPAK.APL. The user is required to load this workspace into the active workspace and execute its function for using this software package. It is a master workspace and contains the function INTPAK that is an interactive program. This function displays on the terminal the menu of the five major workspaces, FVALUE.APL, PEAKVA.APL, ZVALUE.APL, FRQRES.APL and INDATG.APL that are arranged as shown in Figure 5.2. Their functional characteristics are also displayed. The user is prompted to select one of the five workspaces. On being selected, the function in that workspace is copied into the active workspace and is executed. After the execution is completed, the function is erased from the active workspace. The user is then allowed to select a different workspace or to terminate the session. This feature facilitates the software to be used effectively without the need of detailed knowledge of the available workspaces and their functions. The functional details of the functions in the other nine workspaces are presented below.

5.5.1 FVALUE.APL

FVALUE.APL is a workspace that contains the function FREQCOM. This function is an interactive program that assists the user in estimating the frequency of power system voltages as seen by a digital relay. The functional block diagram for the function is shown in Figure 5.3. Also indicated in this figure are the workspaces from which functions are called by FREQCOM. This function can be used by loading FVALUE.APL to the active workspace and then executing it (FREQCOM).

The function FREQCOM interacts with the user and controls the inputs and the outputs. The program structure of the function FREQCOM can be divided into four blocks, viz., input, calculation, display and output blocks. The input block first prompts the user to enter data that includes system frequency, sampling frequency, number of samples to be processed etc. FREQCOM then asks the user to provide the values representing the samples. The sampled values can be entered either from the terminal or from a specially formatted data file.

The calculation block requires the user to chose if the effect of an analog to digital converter is to be simulated or not. If an A/D converter is to be simulated the program assumes that the finite word-length of the micro-processor is also required to be simulated and prompts the user to enter the parameters of the analog to digital converter and the micro-processor. The user is then asked to select the relay algorithm he wishes to use. The function FREQCOM copies the function of the

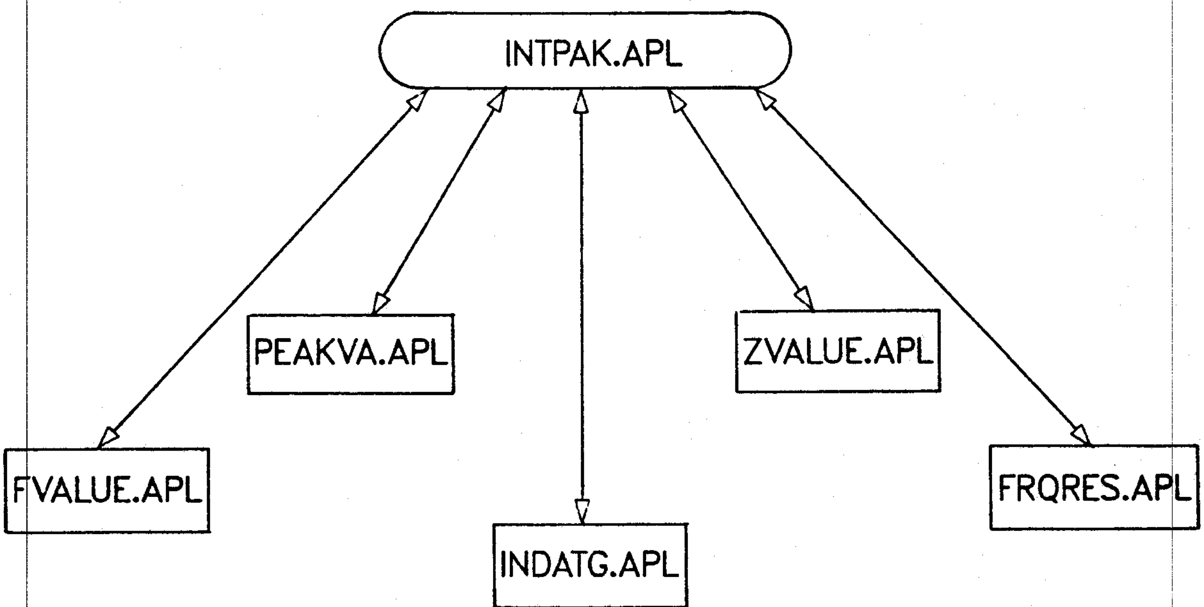


Figure 5.2 The master workspace, INTPAK.APL, and other workspaces.

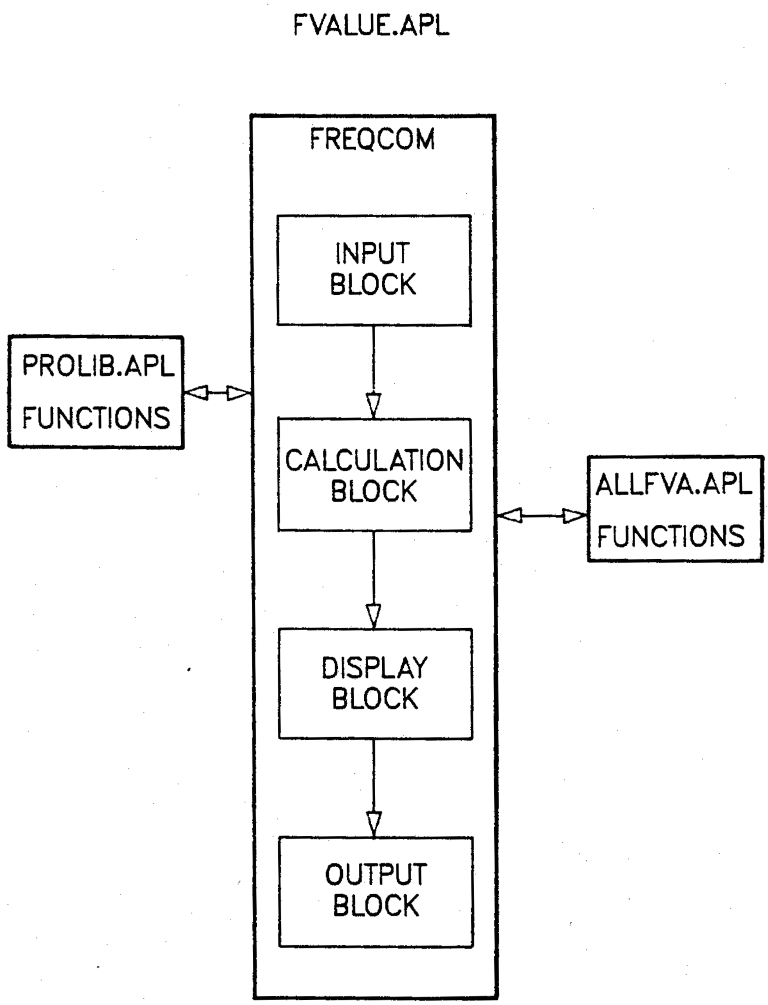


Figure 5.3 A functional block-diagram of the function in the workspace FVALUE.APL.

selected relay algorithm from the workspace ALLFVA.APL and computes the estimates using the provided data set. In general, the relay algorithms compute the frequency only. However, if the least error squares and discrete Fourier transform algorithms are used, the amplitude of the signal can also be calculated. When one of these methods is chosen, the user is prompted to indicate if the amplitude of the signal should be computed or not.

After the computations are over, the computed results are displayed in a tabular form on the terminal. The user can direct the output into files. The results can also be displayed in a graphical form on the terminal. The graphs are plotted using the Fortran program PLOTER.FOR.

The functions that estimate the frequency of power system voltages are located in the workspace ALLFVA.APL. The function for a selected algorithm is copied into the active workspace. The functions for reading the input data files, writing the outputs on files, modelling analog to digital converters etc., reside in the workspace PROLIB.APL. The function FREQCOM copies the functions from that workspace as and when they are to be executed. Functions copied from the workspaces, ALLFVA.APL and PROLIB.APL, are erased from the active workspace after they perform their intended functions. This procedure prevents the active workspace getting filled with the functions that are not needed any more.

5.5.2 ALLFVA.APL

The workspace ALLFVA.APL contains the functions that estimate the frequency of a power system voltage using a set of data samples. The functions are used by the program FREQCOM in the active workspace for computing the estimates. The workspace ALLFVA.APL includes the functions that simulate the following algorithms.

(1) Linear Interpolation Algorithm [26] for
- (*i*) single phase data.
- (*ii*) three phase data.

(2) Fast Fourier Transform Algorithm for single phase data.

(3) Discrete Fourier Transform Algorithm for
- (*i*) single phase data.
- (*ii*) three phase data.

(4) Least Error Squares Algorithm for
  - (i) single phase data.
  - (ii) three phase data.

5.5.3 PEAKVA.APL

PEAKVA.APL is an APL workspace that contains a function PEAKVALUE. The program structure of PEAKVALUE is similar to the function FREQCOM. The function PEAKVALUE interacts with the user for estimating the peak values and phase angles of the phasors representing the power system voltages and currents. The function can be used by loading the workspace PEAKVA.APL into the active workspace and executing the function PEAKVALUE. Like the function, FREQCOM, PEAKVALUE interacts with the user, requests for the inputs, checks their validity and allows the user to make corrections. It also allows the user to simulate the effects of the finite word sizes of the A/D converters and micro-processors. The function PEAKVALUE asks the user to select the relay algorithm. It copies the function of the selected algorithm from the workspace ALLPKV.APL and estimates peak values and phase angles of the phasors using the sampled data representing the input signals. The computed estimates are displayed in a tabular form on the terminal. The estimates can be stored in data files. The output files created can also be used for plotting of results using the program PLOTER.FOR

The workspace ALLPKV.APL contains the functions for the relay algorithms that estimate the peak values and the phase angles of power system voltages and currents. The function of the selected algorithm is copied into the active workspace. PEAKVALUE also copies functions from the workspace PROLIB.APL for reading input data files, modelling the A/D converters, creating the output files etc. Functions copied from the workspaces, ALLPKV.APL and PROLIB.APL, are erased from the active workspace after they have performed their intended functions.

5.5.4 ALLPKV.APL

The workspace ALLPKV.APL contains the functions that are used for estimating peak values and phasor angles of phasors representing power system voltages and currents. The functions use sample data for computing the estimates. The programs in the workspaces PEAKVA.APL and ZVALUE.APL call functions from ALLPKV.APL and execute them to obtain the estimates. The phasor estimates are returned in the polar form for PEAKVA.APL and in the cartesian form for ZVALUE.APL. The program in the workspace FRQRES.APL also calls these functions for calculating the frequency responses of the relay algorithms. The workspace ALLPKV.APL contains the functions for the following algorithms.

(1) Trigonometric Algorithms:

- (*i*) Makino and Miki algorithm.
- (*ii*) Mann and Morrison algorithm.
- (*iii*) Gilcrest, Rockefeller and Udren algorithm.

(2) Correlation Algorithms:

- (*i*) Full cycle Fourier analysis algorithm.
- (*ii*) Half cycle Fourier analysis algorithm.
- (*iii*) Full cycle even and odd rectangular waves algorithm.
- (*iv*) Half cycle even and odd rectangular waves algorithm.
- (*v*) Gilbert and Shovlin algorithm.

(3) Least Error Squares (*LES*) Algorithms:

- (*i*) Three samples *LES* algorithm assuming that the input signal consists of a fundamental frequency component only.
- (*ii*) Five samples *LES* algorithm assuming that the input signal consists of a fundamental frequency component only.
- (*iii*) The *LES* algorithm assuming that the input signal consists of a fundamental frequency component only. The data window size is selected by the user.
- (*iv*) Three samples *LES* algorithm assuming that the input signal consists of a fundamental frequency component added to a constant DC value.
- (*v*) Five samples *LES* algorithm assuming that the input signal consists of a fundamental frequency component added to a constant DC value.
- (*vi*) The *LES* algorithm assuming that the input signal consists of a fundamental frequency component added to a constant DC value. The data window size is selected by the user.
- (*vii*) The *LES* algorithm assuming that the input signal consists of components of fundamental frequency and second harmonic added to a constant DC value. The data window size is selected by the user.
- (*viii*) The *LES* algorithm assuming that the input signal consists of components of fundamental frequency and second harmonic added to a decaying DC value. The data window size is selected by the user.
- (*ix*) The *LES* algorithm assuming that the input signal comprises of the components of the fundamental frequency, second harmonic, third harmonic, fourth harmonic and fifth harmonic added to a decaying DC value. The data window size is selected by the user.

5.5.5 ZVALUE.APL

The workspace ZVALUE.APL contains two functions ZVALUE and IMPVAL. The function ZVALUE guides the user through the programs for computing the impedance of a transmission line as seen by a digital relay. The functional block-diagram of the function ZVALUE is shown in Figure 5.4.

The function ZVALUE controls the inputs and outputs and interacts with the user. The input block of the function prompts the user to enter the inputs that includes the system data, the sampled data representing voltages and currents, the relay connections and characteristics etc. In the calculation block, the function allows the user to incorporate the effects of the finite word sizes of the A/D converters and the micro-processors. ZVALUE then asks the user to select the relay algorithm and executes the function IMPVAL that pre-processes the data before use by the relay algorithm. IMPVAL also copies the function of the selected relay algorithm from the workspace ALLPKV.APL into the active workspace and executes that function to obtain the estimates of the voltage and current phasors. The estimates are computed in the cartesian form. After the estimates of the voltage and current phasors become available, IMPVAL erases the algorithm function and then computes the impedance of a transmission line seen at a relay location using Equations 5.1 and 5.2.

$$Z_r = \frac{V_r\, I_r + V_i\, I_i}{I_i\, I_i + I_r\, I_r} \tag{5.1}$$

$$Z_i = \frac{V_i\, I_r - V_r\, I_i}{I_i\, I_i + I_r\, I_r} \tag{5.2}$$

where

V_r is the real part of the voltage phasor,
I_r is the real part of the current phasor,
V_i is the imaginary part of the voltage phasor,
I_i is the imaginary part of the current phasor.

The function IMPVAL assigns the impedance values to variables that are used by the function ZVALUE. ZVALUE displays the values of the computed impedances either in the cartesian form or in the polar form as desired by the user. The calculated impedances can also be saved in output files

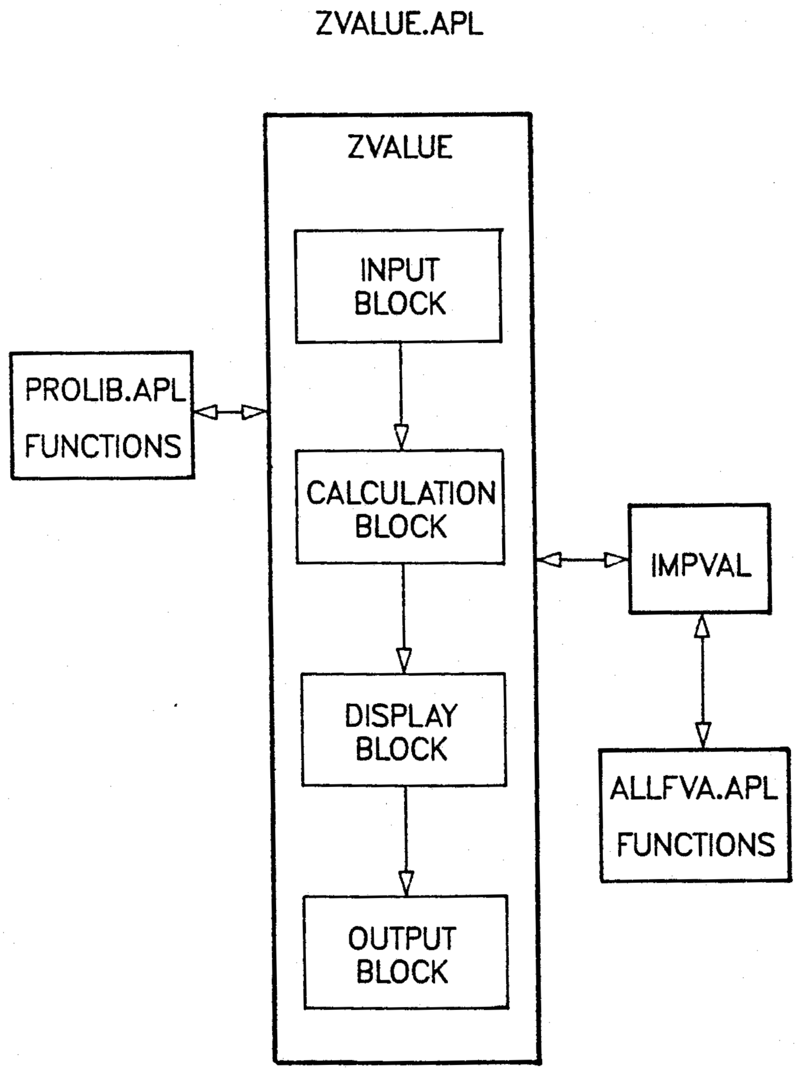


Figure 5.4 A functional block-diagram of the function in the workspace ZVALUE.APL.

and subsequently plotted using the program PLOTER.FOR.

Functions for reading input files, writing output files etc., are called from the workspace PROLIB.APL. All the functions called from that workspace are erased from the active workspace after they have performed their intended functions.

5.5.6 FRQRES.APL

Most of the digital relaying algorithms can be viewed as a pair of orthogonal (sine and cosine) digital filters. FRQRES.APL workspace contains a function FRQRES that is developed to obtain the frequency response of the relaying algorithms. For each pair of orthogonal filters, the response of individual filters and their composite response can be obtained.

The function FRQRES is an interactive program. It uses the coefficients of a relay algorithm (filter) for calculating the frequency response. Appendix A discusses a method used for obtaining the frequency response of a digital filter. The function FRQRES first prompts the user for inputs, such as, the nominal frequency, the sampling frequency etc. It then displays, on the terminal, a list of the algorithms for which the frequency response can be calculated. The user is prompted to select an algorithm and then the function of the selected algorithm is copied from the workspace ALLPKV.APL into the active workspace. The algorithm function is then executed to obtain the filter coefficients. Alternatively, the user is provided an option of entering the coefficients of the filter either directly from the terminal or from a data file. This option is provided, in case the user does not wish to calculate the response of algorithms in the workspace ALLPKV.APL. After the filter coefficients are obtained, FRQRES computes individual and composite frequency responses and displays them on the terminal. The calculated responses can also be stored into data files and subsequently plotted. The workspace PROLIB.APL provides the functions for reading the files that contain the filter coefficients and writing the output files. The functions copied from that workspace are erased from the active workspace after they perform their intended functions.

5.5.7 INDATG.APL

The workspace INDATG.APL contains a function INDATGEN. The functional block diagram of INDATGEN is shown in Figure 5.5. The function INDATGEN is an interactive program that can be used to generate the instantaneous values of sampled data for representing power system voltages and currents. The function INDATGEN interacts with the user and controls the inputs and outputs. It requests the user to input the data that includes system frequency, sampling frequency, number of

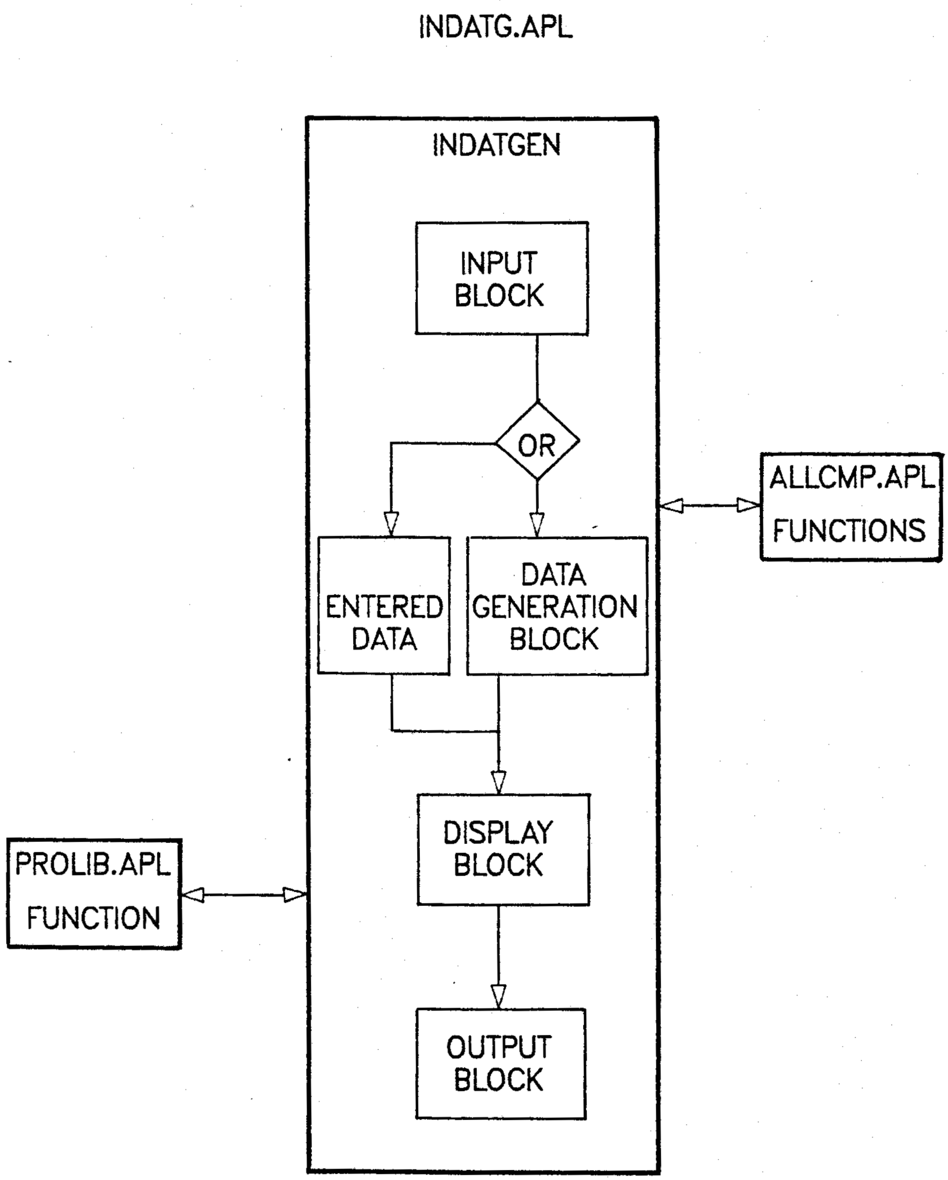


Figure 5.5 A functional block-diagram of the function in the workspace INDATG.APL.

samples required etc.

The sampled data are generated by executing the functions invoked from the workspace ALLCMP.APL. The workspace contains functions that can generate the following components of a power system voltage or current.

(1) The Fundamental Frequency Component of
- (*i*) constant amplitude.
- (*ii*) decaying amplitude.
- (*iii*) rising amplitude.

(2) DC Components of
- (*i*) constant amplitude.
- (*ii*) decaying amplitude.
- (*iii*) rising amplitude.

(3) Harmonic Components of
- (*i*) constant amplitude.
- (*ii*) decaying amplitude.
- (*iii*) rising amplitude.

(4) Components of Non-harmonic Frequencies with
- (*i*) constant amplitude.
- (*ii*) decaying amplitude.
- (*iii*) rising amplitude.

(5) Random Noise.

The function INDATGEN allows the user to optionally select the above components, in succession, that are to be included in the required signal. The function for a selected component is copied into the active workspace from ALLCMP.APL. The copied function is then executed to compute the sampled data that represent the selected component. The function called from the workspace ALLCMP.APL is erased from the active workspace after it has been executed. This procedure is repeated until the user enters a stop instruction. The sampled data representing the selected components are added to obtain the required signal.

The user is also provided with an alternative to enter the data directly from the terminal instead of generating them by using the program. The generated data (or terminal entered data) representing the required signal can be displayed on the terminal. The user can direct the data into a file and can view the signal in a graphical form. The output file is created using a function copied from a workspace PROLIB.APL. This function is erased from the active workspace after the output file is created. The output file can be used by the other programs of this package.

5.5.8 PROLIB.APL

The workspace PROLIB.APL is a collection of functions (subroutines) that form a function library for use by the programs in this software package. The executing function in the active workspace copies the functions from PROLIB.APL into the active workspace and uses them. The functions stored in the PROLIB.APL workspace are described below.

(a) INPTFILE

The purpose of this function is to take data from a data file and convert it into a form that can be read by the functions of active workspace. This function is designed to be used in conjunction with two other functions, RDFILE and PRFILE. RDFILE reads a specified number of data samples from a file. PRFILE prints the data file information on the terminal after the file has been read by RDFILE. INPTFILE compares the key parameters entered by the user, such as, system frequency and sampling frequency, with those read from the the header of the file. This is a check against the use of inappropriate data files.

(b) WRFILE

This function creates an output file of the results computed in the active workspace. The data files created by this function are in a format that can be read by INPTFILE. The data files are also compatible with the Fortran program PLOTER.FOR for plotting graphs.

(c) ADCON

ADCON is a function designed to simulate the effect of analog to digital converters in digital relays. The procedure for modelling of analog to digital converters is described in Appendix B.

(d) MUL

MUL is a function that is used for obtaining the product of an integer number with a decimal number using the bit shift approach. The use of this function allows the user to demonstrate the accuracy of an algorithm when used in a micro-processor. The function is used in conjunction with either of the two functions MULSGP and MULDBP. MULSGP is used to implement the ordinary bit shift approach and the function MULDBP is used for implementing the extended bit shift approach. The bit shift approaches used in these functions are discussed in Appendix C.

(e) CMUL

CMUL is a function that is used for obtaining the product of two complex numbers. The function is used in conjunction with MUL.

(f) SQRT

Most of the digital relaying applications involve the computation of the amplitude of a phasor from its real and imaginary components. The implementation of this requires two squaring operations and one square root operation. These operations are computationally inefficient. The function SQRT provides a piecewise linear approximation to relace these calculations. The method of obtaining the square roots, is simulated because it is suitable for use on micro-processors. Details of the procedure are given in Appendix D.

(g) ACCURACY

The function ACCURACY permits the user to select:

(1) the type of quantization-truncation or rounding, in analog to digital converters and micro-processors.

(2) the number of bits of accuracy for A/D converters and micro-processors.

(3) the bit shift multiplication-ordinary approach or extended approach.

(4) the number of regions of approximations to be implemented in estimating the square roots.

(h) FFT

The function FFT performs the fast Fourier transform on a set of data samples and computes the leakage coefficient. The function FFT has been developed for use by fast Fourier transform algorithm.

(i) ARCTAN

The function ARCTAN has been developed for use by the discrete Fourier transform algorithm. This function simulates a lookup table suitable for estimating an angle from the values of its tangent. It is suitable for providing the angle when its tangent is in the range of 0–2.0. This range is suitable for obtaining correct estimates of the frequency deviation between 0 to 5 Hz.

5.6 Testing the Software

To ensure high reliability of the software, tests were conducted to search for bugs and errors. The testing involved executing the software using data for which results are known and examining the results for any errors. The software was checked and modified until no errors were encountered. Some of the sample test studies are included in the next chapter.

5.7 Future Developments

The software has been developed for estimating the electrical parameters of power systems. The logic and characteristics of relays are not modelled in the software. Programs for simulating those functions can be incorporated in the software. The new functions would compare the estimated parameters with the pre-programmed relay settings and characteristics and simulate the decision making process of relays. Also, the developed software includes only the non-recursive algorithms. It can be extended to include the recursive algorithms, such as, Kalman filters etc.

5.8 Summary

The software developed for this project has been described in this chapter. The specifications of the software have been outlined. The APL programs developed for this software reside in ten workspaces that are available on the VAX-8600 at the University of Saskatchewan. The software package consists of "question and answer" type interactive programs and the dialogue between the user and the computer is governed by the user's answers to the questions put forth by the computer. This feature provides effective user control of the programs. The present version of the software includes programs for simulating the performance of algorithms that are suitable for estimating the phasors representing power system signals and the frequency of power system voltages. The software can be used for selecting appropriate relaying algorithms for use in an application. It can also be used as an educational aid for studying and comparing the performance of relay algorithms.

Chapter 6

TESTING OF THE SOFTWARE

6.1 Introduction

A software package for the evaluation of relay algorithms has been described in the last chapter. The specifications and the structure of the software have been outlined. The software was tested for investigating its capabilities. Some results and sample studies are presented in this chapter to demonstrate the reliability and capabilities of the developed software. The effect of finite word size on the accuracy of the computed results are also reported.

6.2 Modules for Simulating Multiplications, A/D converters and the Amplitude Estimator

This section presents a few studies that were conducted to illustrate that the multiplication, A/D converter and Amplitude estimator modules perform their intended functions properly. The multiplication module provides an output that is equal to the product of an integer number and a decimal value. The A/D converter module uses a model that simulates the effect of an A/D converter. The amplitude estimator module uses piecewise linear approximation for estimating the amplitude of a phasor from its real and imaginary components.

The two forms of multiplications, ordinary and extended, are incorporated in the multiplication module. The approaches are similar except that, in the extended form, the integer number is first multiplied with the largest number that can be stored in the word of the processor. The result is multiplied with the decimal value using the bit shift approach. The product is then divided by the number with which the integer was multiplied. This results in a reduction in the truncation or rounding errors. The details of the multiplication approaches are discussed in Appendix C. The module was tested using the integer 1000 and the decimal value of 0.707106. The product of these numbers is 707.106. Figure 6.1 shows the products obtained when the numbers were multiplied using the two approaches. The figure shows that the errors in the results reduce as the number of bits used for simulating the multiplications increases. The errors observed in the extended bit shift approach are noticeably less than those obtained by the ordinary approach.

The A/D converter module models the conversion of analog signals to the digitized form. The input to A/D converter module consists of values in the floating point representations. The output of the module provides the equivalent integer values. The procedure of modelling of the A/D converter is discussed in Appendix B. The performance of the A/D converter module was tested by simulating

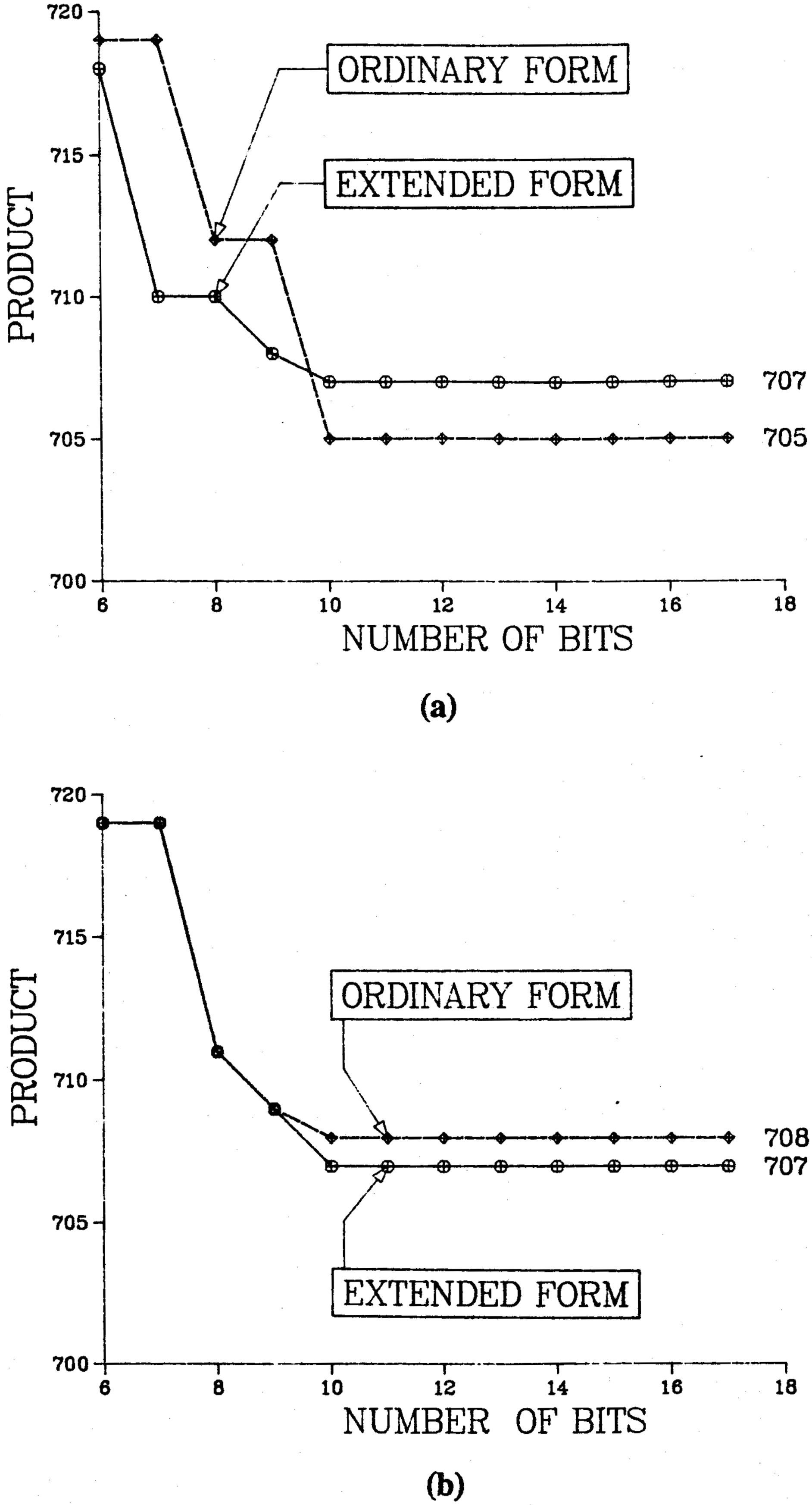


Figure 6.1 The product of 1000 and 0.707106 obtained by bit shift multiplications (a) with truncation and (b) with rounding.

a sinusoidal signal of 10 volts peak and converting it to the integer form using Equation b.2. The specifications of the A/D converter model used in this study are as follows.

(1) Number of bits=8.

(2) Positive saturation voltage level=15 volts.

(3) Negative saturation voltage level=-15 volts.

(4) Truncation during the digitization.

The input signal and the output of the module are shown in Figure 6.2. This figure shows that when the signal is 8.66 volts, the output of the module is 73. In fact, this number corresponds to an input level of 8.554 volts. The difference in the two voltage levels is due to truncation during the conversion process. The figure also reveals that when the input is -8.66 volts, the output of the module is -74 instead of -73. This is because the errors due to truncation are negative for the positive and negative inputs.

The amplitude estimator uses a piecewise linear approximation for estimating the amplitude of a phasor from its real and imaginary components. The technique first redefines the phasor whose amplitude is to be estimated. The real component of the redefined phasor is the larger of the absolute values of the real and the imaginary components of the original phasor. The imaginary component is the smaller of the absolute values. The amplitude of the phasor redefined in this manner is the same as the amplitude of the original phasor but the redefined phasor lies in the first octant. The amplitude of the redefined phasor is determined by multiplying its real and imaginary components by suitable coefficients and then adding the products. The accuracy of the technique can be improved by sub-dividing the octant in a number of regions. For each region, a pair of coefficients is defined that is valid for use when the redefined phasor is in that region. The method is discussed in detail in Appendix D. The estimator module was tested by estimating the amplitude of a phasor from its real and imaginary components as the phasor rotated in the first octant in steps of 2^o. The estimates were computed for three cases. In the first case, the octant was not sub-divided and was used as one region. In the other two cases, the octant was sub-divided in four regions and sixteen regions. Figure 6.3 shows the percentage errors observed in the amplitude estimation for the three cases. A study of the figure shows that the errors decrease as the number of regions in the octant increase. The maximum error observed in the one region case is 6.18%. It reduces to 0.49% and 0.03% for the four region and sixteen region cases.

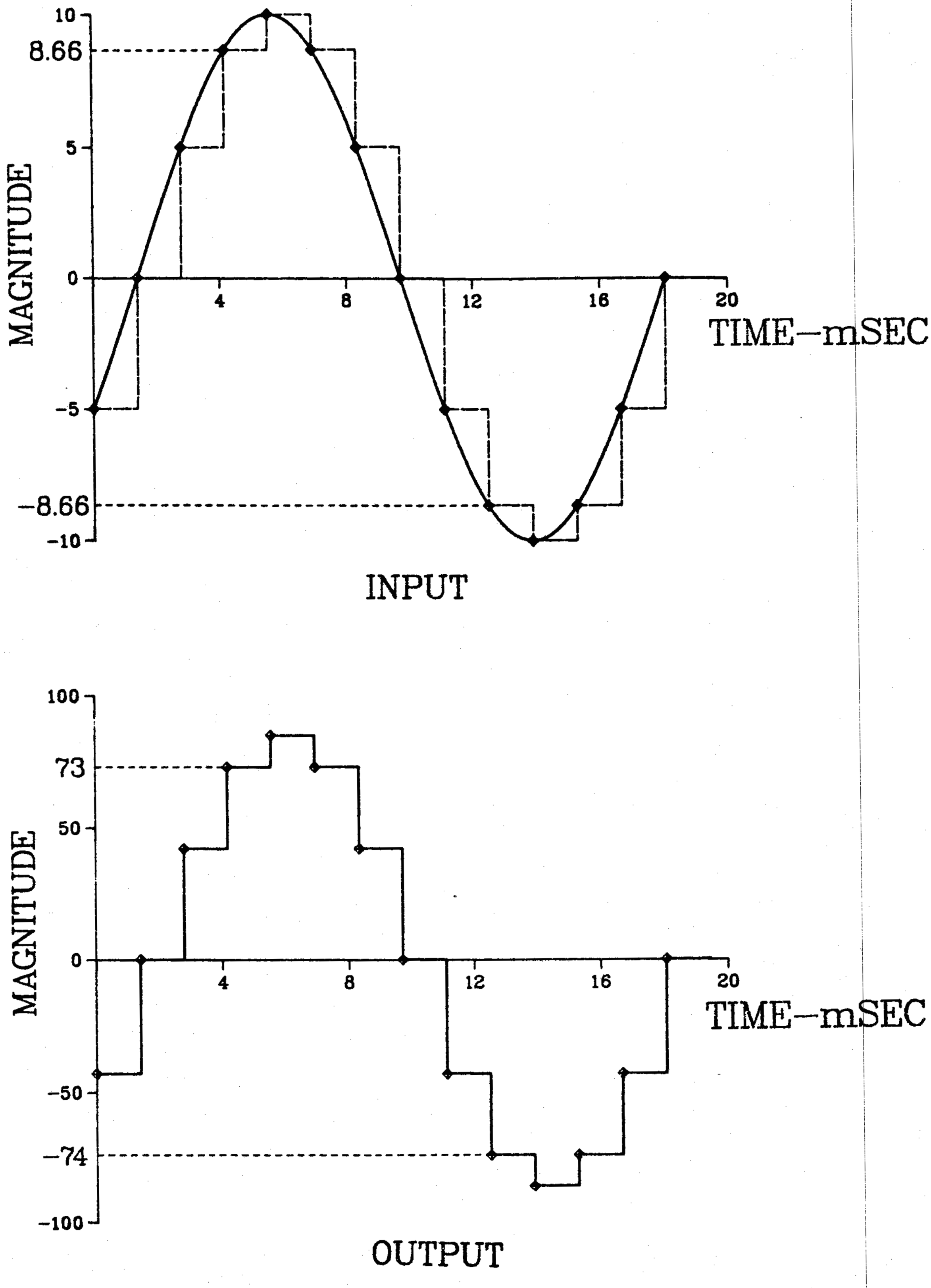


Figure 6.2 A sample input and output of the A/D converter module.

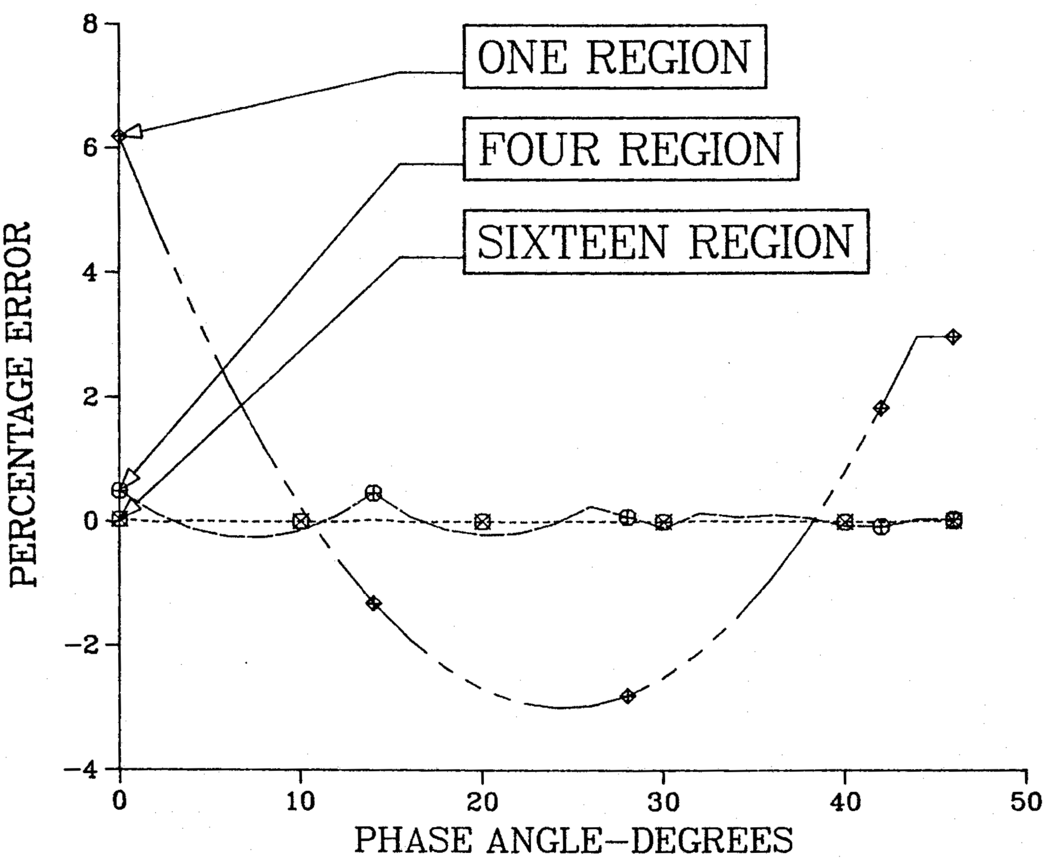


Figure 6.3 The percentage errors in the amplitudes estimated by the amplitude estimator module versus the phasor angles of the phasor.

6.3 Studies on Estimating Phasors

Algorithms for computing the real and imaginary components of voltage and current phasors, have already been discussed in Chapter 3. The programs developed for simulating the algorithms were tested. The estimates were computed using sinusoidal input data and floating point representation of numbers. The estimates obtained were compared with the theoretical values. Whenever, the estimates computed did not conform with the expected values, the programs were checked, the errors were identified and corrected. The programs were then tested to study if finite word sizes of A/D converters and micro-processors could be simulated effectively by the developed software. Several tests were conducted and one such study is reported in this section.

6.3.1 Effect of Finite Word Size of Processors and A/D converters

The factors affecting the performance of digital relays have been discussed in Chapter 2. The accuracy of the estimates provided by the digital relays is limited by the finite number of bits in the words of A/D converters and micro-processors. Also, an A/D converter is designed for a specified range of input voltages. The converter saturates, when the input exceeds the specified range resulting in incorrect representation of the inputs and consequently introducing substantial errors in the results.

The effects of using words of finite sizes and A/D converter saturation were examined using the least error squares algorithm. This algorithm assumes that the input is composed of an exponentially decaying DC component and a fundamental frequency component. Test data for a sinusoidal waveform were computed using the program INDATGEN that resides in the workspace INDATG.APL. The instantaneous values of the waveform were calculated at intervals of $\frac{1}{720}$ seconds. The amplitude and frequency of the waveform were selected to be 1000 volts and 60 Hz, respectively. The least error squares algorithm was used for computing the phasor representations from the data. Floating point representation of numbers was used. The amplitude and the phase angles obtained are plotted in Figure 6.4. As expected, the computed peak values are close to 1000 volts.

The phasors were estimated using the above data, the least error squares algorithm, and the A/D converter, multiplication and amplitude estimator modules. The instantaneous values of the sampled data were used as input to the A/D converter module. The module first scaled down the data using a reduction ratio of 100/1 and then converted the low level data to equivalent integer values using an eight bit A/D converter model of ± 10 volts dynamic range. The output of the A/D converter module was used by the algorithm. The ordinary bit shift approach of multiplication was simulated using twelve bit words. The four region approximation was used for estimating the peak value of the phasors from their real and imaginary components. The truncation was implemented during digitization and calculations. Figure 6.5 shows the computed results. An examination of the figure clearly depicts that, as expected, the errors are encountered due to the finite size of the words. The estimates of the peak values computed by the algorithm are in the 875-992 volts range. The study was repeated using different word lengths. Figure 6.6 shows the estimates obtained, when a 12-bit A/D converter model and 16-bit words for the bit multiplication were simulated. In this case, the peak value estimates are between 990 and 998 volts. As expected, the results show an improvement in the accuracy of computations when the number of bits per word is increased.

The study was then repeated to test the ability of the A/D converter module to simulate the saturation of the converter. The saturation can be simulated by using the inputs that exceed the dynamic range of conversion of the A/D converter. The input data representing a sinusoidal signal of 1300 volts peak was generated and applied as input to the A/D converter module with 12-bit A/D conversion. The output of the module was used by the algorithm for computing peak values. The bit shift multiplication was simulated for the word length of sixteen bits. Figure 6.7 illustrates the estimates obtained in this manner. An examination of the figure shows that the estimates of the peak value are between 1087 and 1133 volts. These estimates are substantially different from the expected value of 1300 volts. The differences are mainly due to the saturation of the A/D converter.

6.4 Sample Studies on Frequency Estimation Algorithms

The recently developed algorithms that estimate the frequency of power system voltages have been discussed in Chapter 3. The frequency of the power system is, generally, required to be measured during the steady-state operation only. The developers of algorithms, therefore, assumed that the inputs are waveforms of a single frequency. When the frequency of an input deviates from its nominal value, the algorithms compute the frequency deviation. The frequency of the input voltage waveform is then estimated.

The programs that simulate the frequency algorithms were tested to ensure their reliability. Input data representing signals of known frequencies were used in the tests. The estimates of the frequency provided by the programs were compared with the frequency of the input signals. The programs were debugged, if the computed estimates did not match with the expected values. Results from two of the several tests that were conducted, are reported in this section. For these tests, data

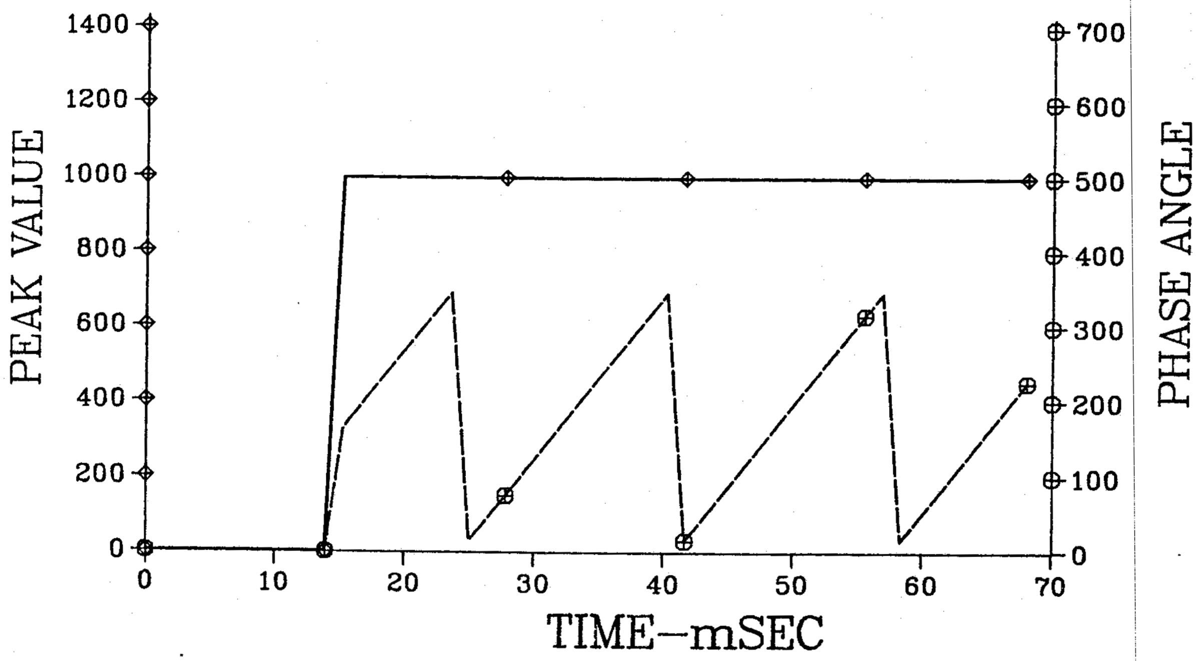


Figure 6.4 The peak value and phase angle estimates obtained by the *LES* algorithm using floating point representation of numbers.

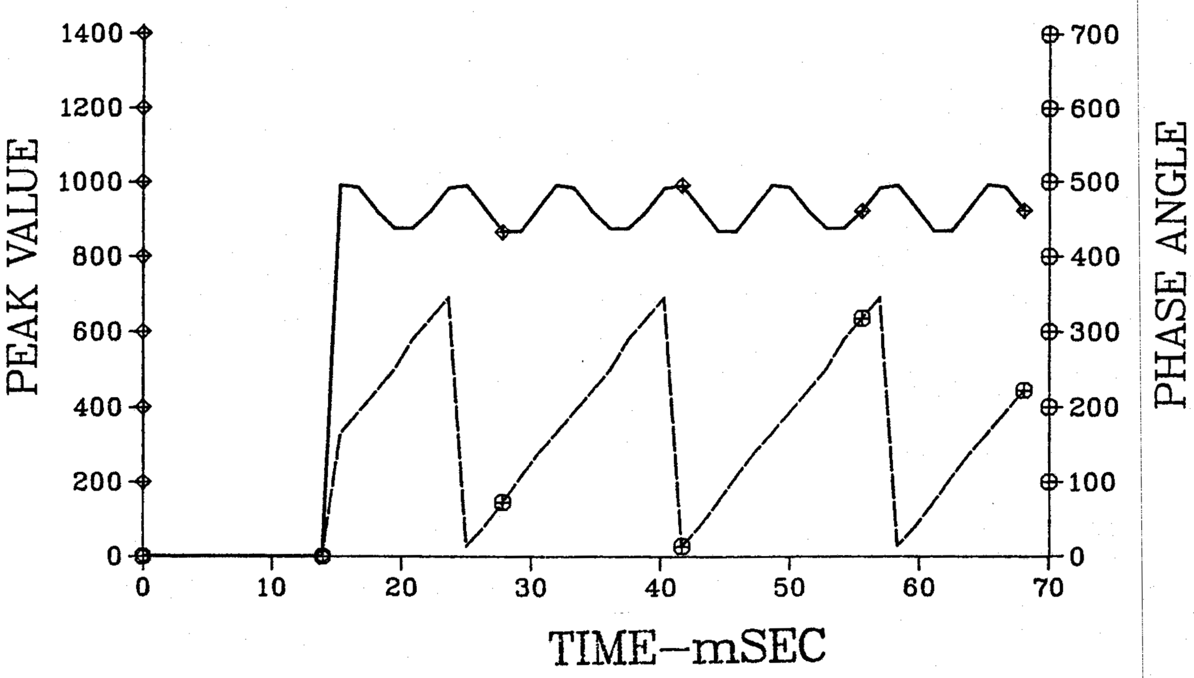


Figure 6.5 The peak value and phase angle estimates obtained by the *LES* algorithm using an 8-bit A/D converter model and 12-bit words for multiplications by the bit shift procedure.

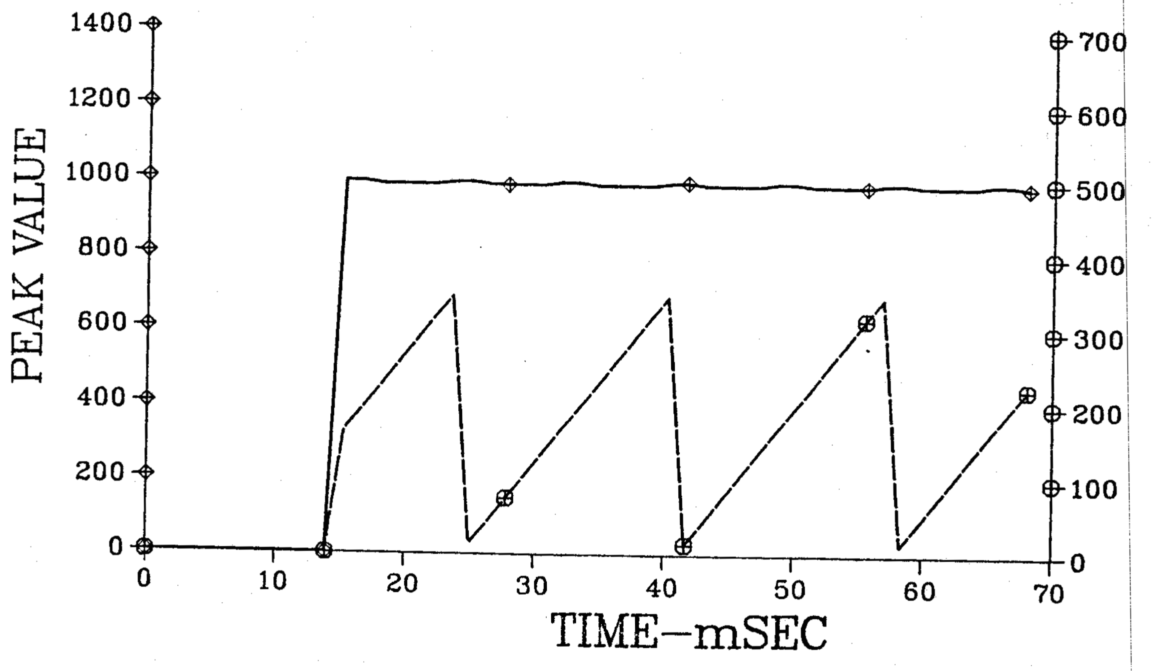


Figure 6.6 The peak value and phase angle estimates obtained by the *LES* algorithm using a 12-bit A/D converter model and 16-bit words for multiplications by the bit shift procedure.

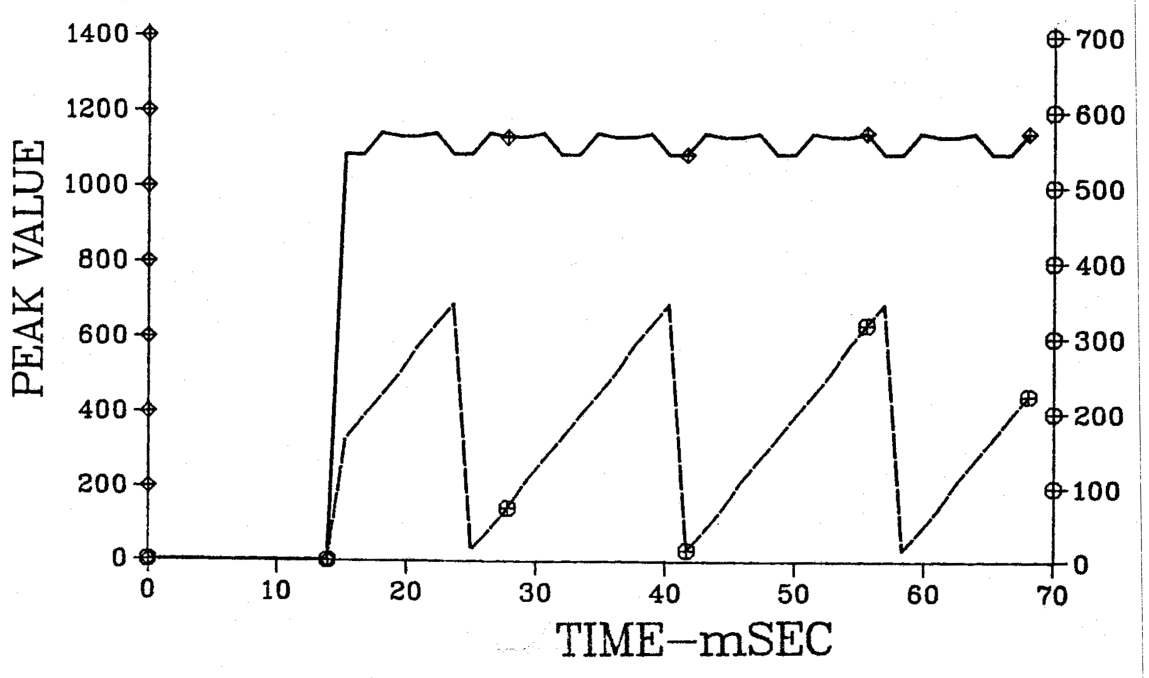


Figure 6.7 The peak value and phase angle estimates obtained by the *LES* algorithm using a 12-bit A/D converter model and 16-bit words for multiplications by the bit shift procedure. The input signal exceeded the range of the A/D converter.

representing three phase sinusoidal voltages were computed using the program INDATGEN. The amplitudes of the waveforms were selected to be 1000 volts. The frequency of the waveforms used in these tests were chosen to be 60 and 59 Hz.

(1) Sinusoidal Input - Nominal Frequency

The frequency of 60 Hz waveforms was estimated using the developed programs. These programs use the following algorithms.

(1) Single phase discrete Fourier transform (1ϕ *DFT*) algorithm.
(2) Three phase discrete Fourier transform (3ϕ *DFT*) algorithm.
(3) Single phase least error squares (1ϕ *LES*) algorithm.
(4) Three phase least error squares (3ϕ *LES*) algorithm.
(5) Single phase linear interpolation (1ϕ *LI*) algorithm.
(6) Three phase linear interpolation (3ϕ *LI*) algorithm.
(7) Single phase fast Fourier transform (1ϕ *FFT*) algorithm.

The estimates of the frequency were computed using the floating point representation of numbers. Except for the fast Fourier transform (*FFT*) algorithm, the sampling rate was used as 720 Hz and the data window size for the frequency measurement was selected as two cycles. The sampling rate and the data window size were selected to be 1920 Hz and one cycle for the *FFT* algorithm. The frequency estimates computed by the first six techniques are shown in Figure 6.8. The estimates obtained by the *FFT* algorithm are shown in Figure 6.9. An examination of the figures shows that the estimates provided by all the algorithms are accurate.

The discrete Fourier transform algorithms (*DFT*) and the least error squares algorithms (*LES*) also provide the estimates of the amplitude of the input signals in addition to the frequency estimates. Therefore, the amplitudes of the inputs were also computed by the programs that use the *DFT* and the *LES* algorithms. The percentage errors for the amplitude estimates were calculated and are plotted in Figure 6.10. This figure shows that the amplitude estimates are free of errors.

(2) Sinusoidal Input - Off-nominal Frequency

The frequency of the 59 Hz waveforms was then calculated using the programs. The data window sizes, the sampling rates and the representation of numbers used in Test 1 were also used in these cases. The estimates computed by the programs that incorporate the first six algorithms from the above list are shown in Figure 6.11. The estimates computed by the program that incorporates the *FFT* algorithm, are shown Figure 6.12. The estimates obtained from the programs that use the three phase *DFT* algorithm, the single phase and the three phase *LES* algorithms and the *FFT*

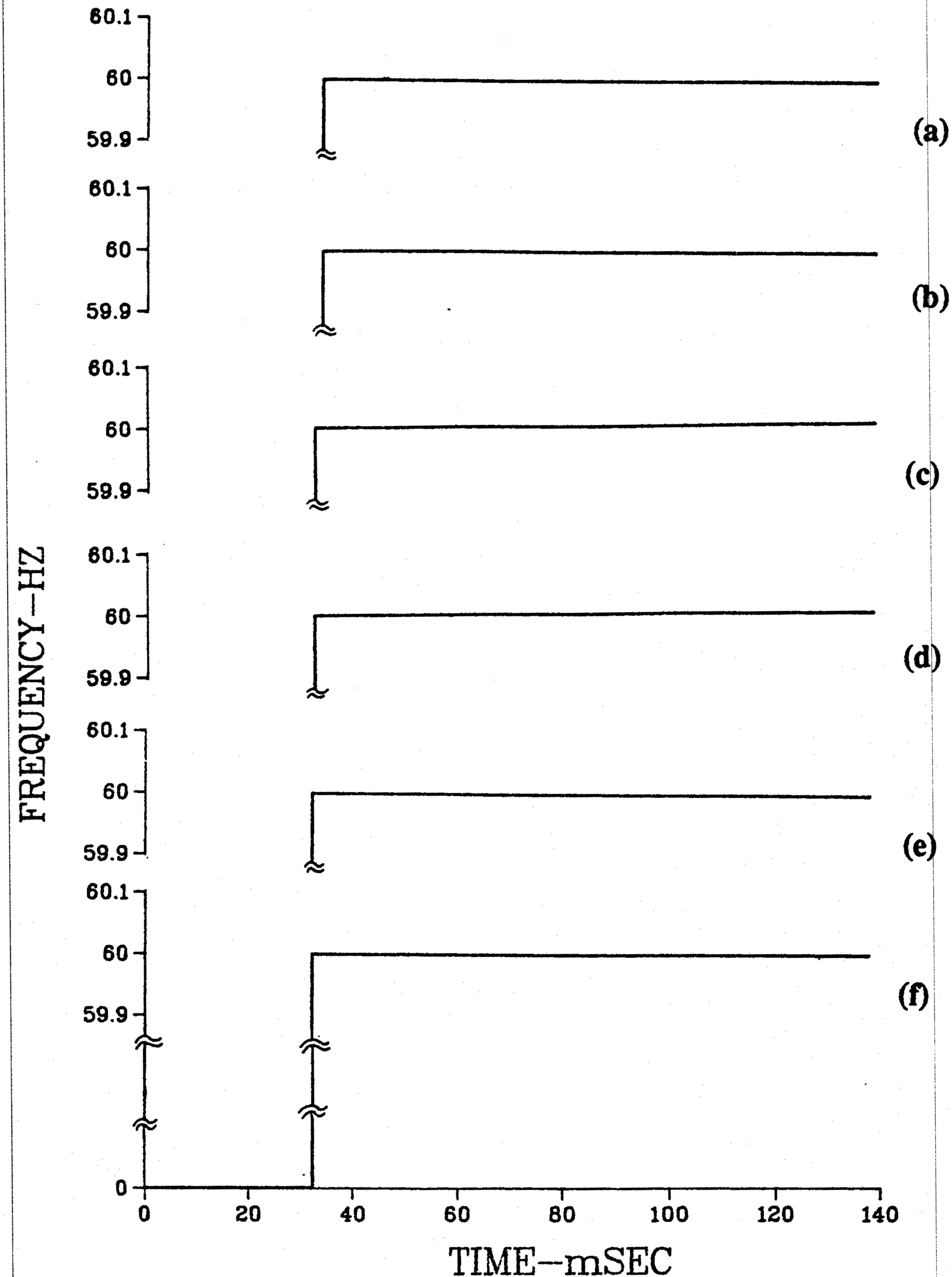


Figure 6.8 The frequency estimates for a 60 Hz input obtained by (a) the 1ϕ *DFT* algorithm, (b) the 3ϕ *DFT* algorithm, (c) the 1ϕ *LES* algorithm, (d) the 3ϕ *LES* algorithm, (e) the 1ϕ *LI* algorithm and (f) the 3ϕ *LI* algorithm.

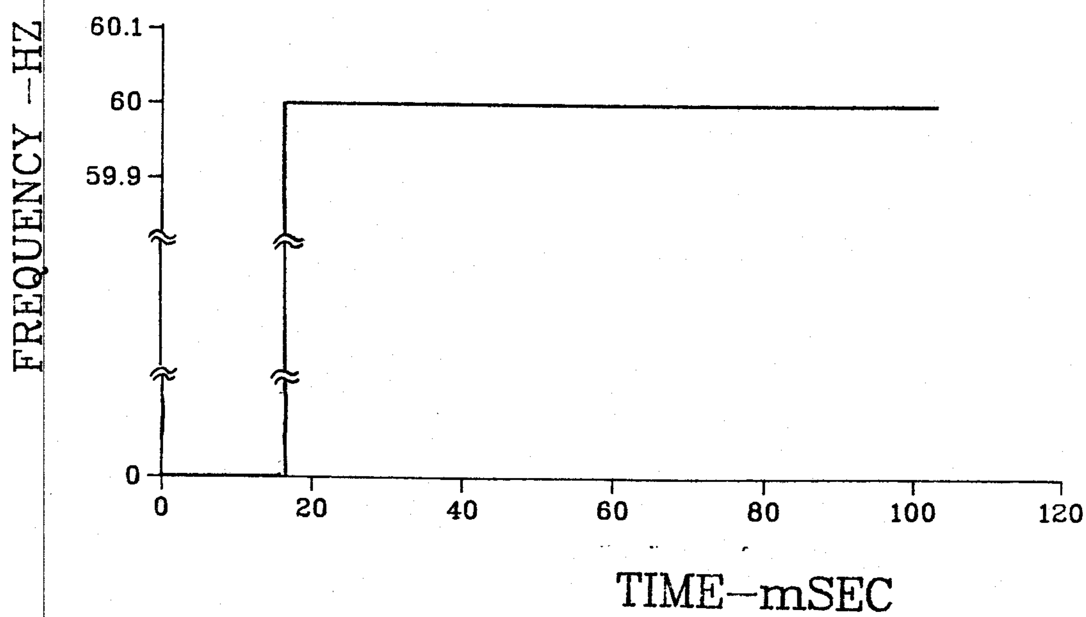


Figure 6.9 The frequency estimates for a 60 Hz input obtained by the *FFT* algorithm.

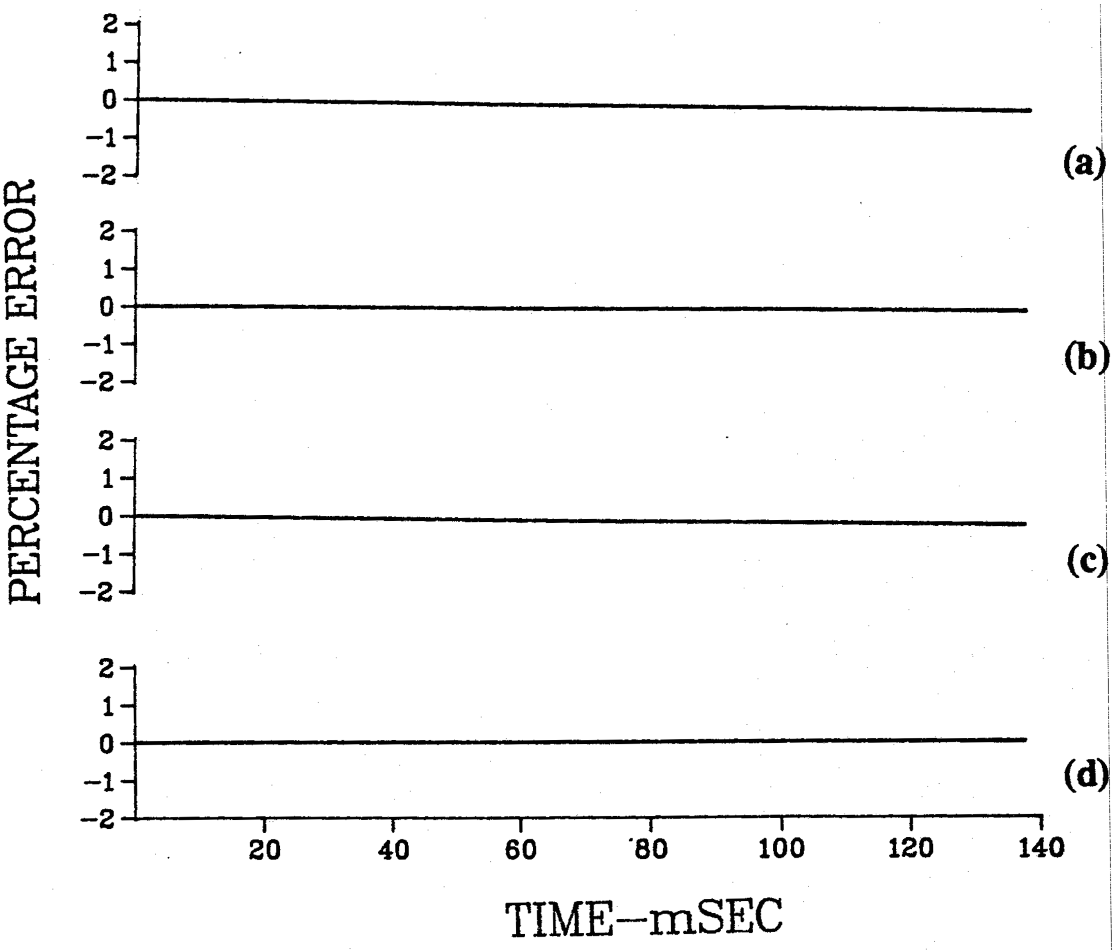


Figure 6.10 The percentage errors in the amplitude estimates for a 60 Hz input computed by (a) the 1ϕ *DFT* algorithm, (b) the 3ϕ *DFT* algorithm, (c) the 1ϕ *LES* algorithm and (d) the 3ϕ *LES* algorithm.

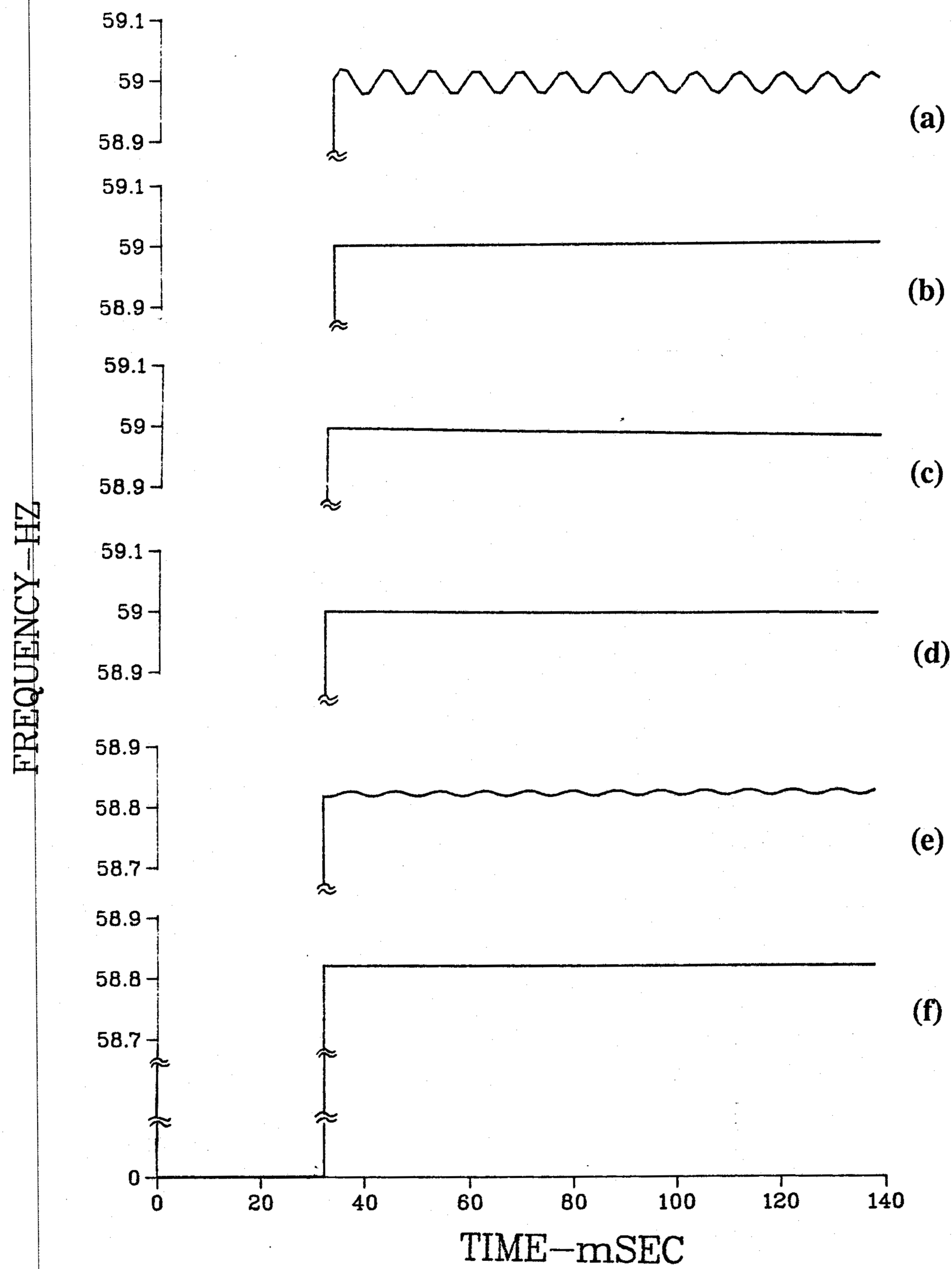


Figure 6.11 The frequency estimates for a 59 Hz input obtained by (a) the 1ϕ *DFT* algorithm, (b) the 3ϕ *DFT* algorithm, (c) the 1ϕ *LES* algorithm, (d) the 3ϕ *LES* algorithm, (e) the 1ϕ *LI* algorithm and (f) the 3ϕ *LI* algorithm.

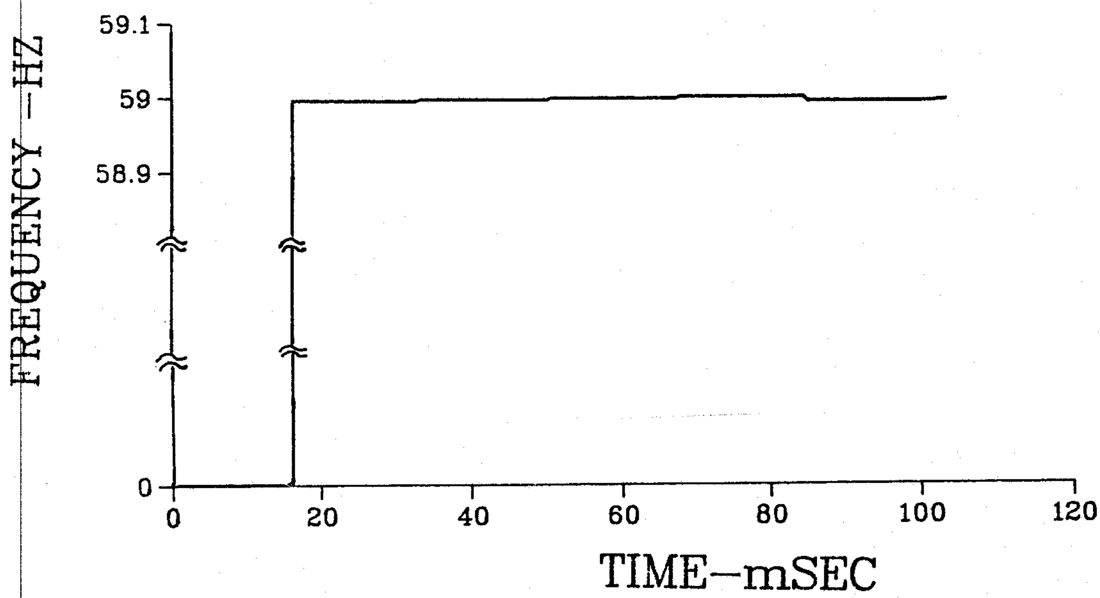


Figure 6.12 The frequency estimates for a 59 Hz input obtained by the *FFT* algorithm.

algorithm, are very close to the expected value of 59 Hz. However, the frequency estimates obtained by using the single phase *DFT* algorithm fluctuate about the expected value. The maximum error observed is 0.02 Hz. Also, the estimates fluctuate about a mean error 0.1788 Hz, when single phase linear interpolation (*LI*) algorithm was used. The estimates, in this case, are in the 58.8165 and 58.8258 Hz range. The estimates obtained from the three phase *LI* algorithm have a mean error of 0.1788 Hz. However, the estimates do not fluctuate about this value. The other tests showed that the fluctuations in the single phase *DFT* algorithm and the mean error in the *LI* algorithms increased as the frequency of the input signals deviated farther from the nominal value. The errors in the estimates are not due to errors in the programs, but, are due to the characteristics of the algorithms. A study reported by Sachdev and Giray [32] also indicated that the estimates obtained using the *DFT* technique and the single phase data are affected adversely at off-nominal frequency.

The amplitude estimates were also computed for the 59 Hz waveforms using the *DFT* and *LES* algorithms. Figure 6.13 shows the percentage errors observed in these cases. An examination of the figure shows that the estimates obtained by the single phase *DFT* algorithm fluctuate; the maximum error is 0.92%. The estimates from the three phase *DFT* algorithm are constant but have a mean error of 0.0454%. The estimates obtained using the *LES* algorithms are very accurate and are close to 1000 volts. The errors in the amplitude estimates observed, in the case of the *DFT* algorithms, are due to the characteristics of algorithms as reported in Reference [32].

The *LES* algorithms were also used to compute the frequency of the 59 Hz signal using data converted to the integer format by the A/D converter module. The multiplication module was used for obtaining the products of the integer numbers with the decimal numbers. The following specifications were selected for the A/D converter and multiplication modules.

(1) Reduction ratio 150/1.

(2) 8-bit A/D converter, dynamic range ± 10 volts and rounding during the conversion.

(3) Extended bit shift multiplication for the word of 12-bit.

The results of the calculations were rounded. The programs, in this mode of operation, perform the computations using the integer format of the numbers. For achieving high accuracy of the frequency estimates, the programs compute 1000 times the actual frequency deviation, Δf. For example, a frequency deviation, Δf, of 0.999 Hz is computed as 999. The frequency is then estimated by the programs as 59001. This is to be interpreted as 59.001 Hz. The frequency estimates obtained using the *LES* algorithm, in this manner, are shown in Figure 6.14. An examination of the figure, clearly indicates that the use of words of finite sizes has introduced errors in the estimates. The tests were

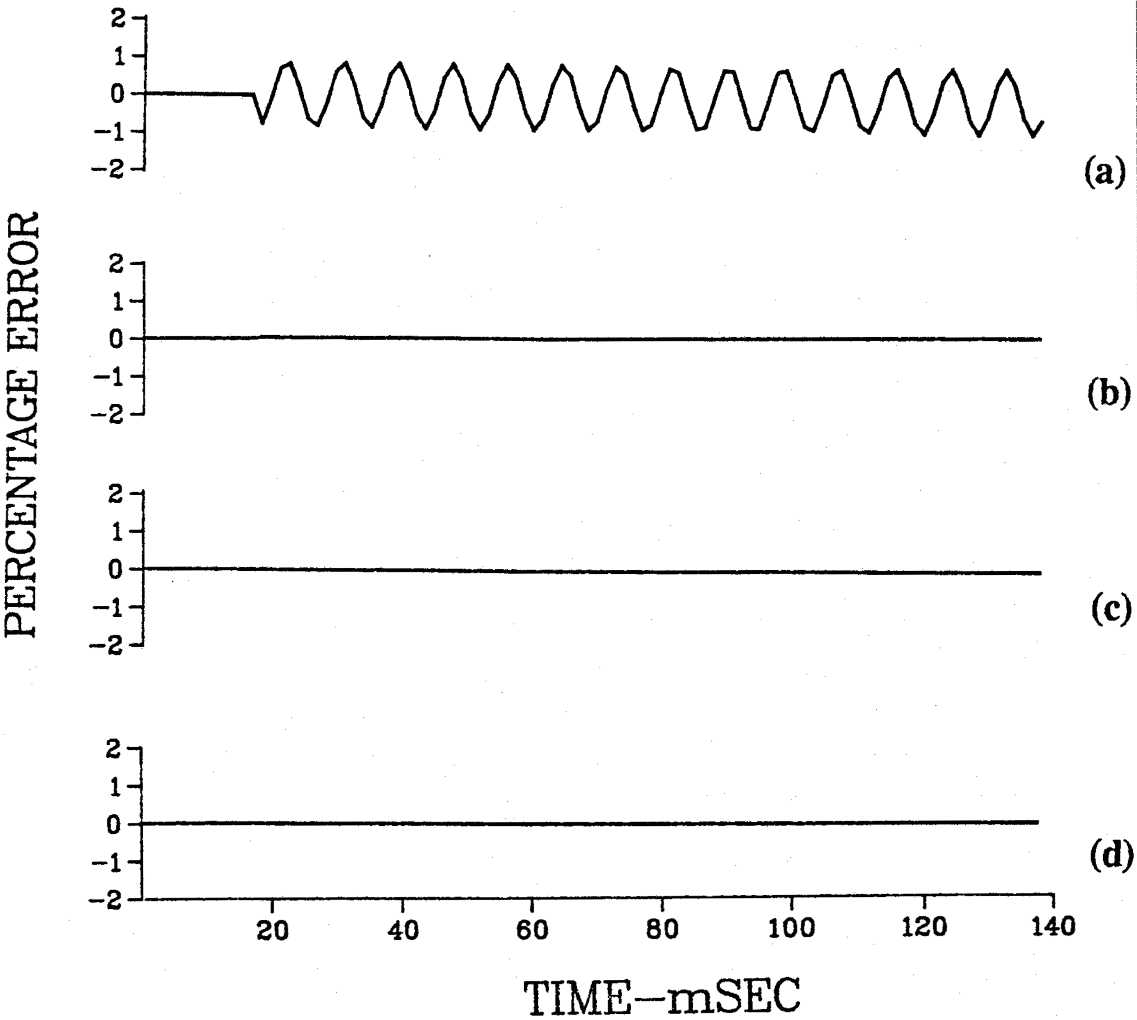


Figure 6.13 The percentage errors in the amplitude estimates for a 59 Hz input computed by (a) the 1ϕ *DFT* algorithm, (b) the 3ϕ *DFT* algorithm, (c) the 1ϕ *LES* algorithm and (d) the 3ϕ *LES* algorithm.

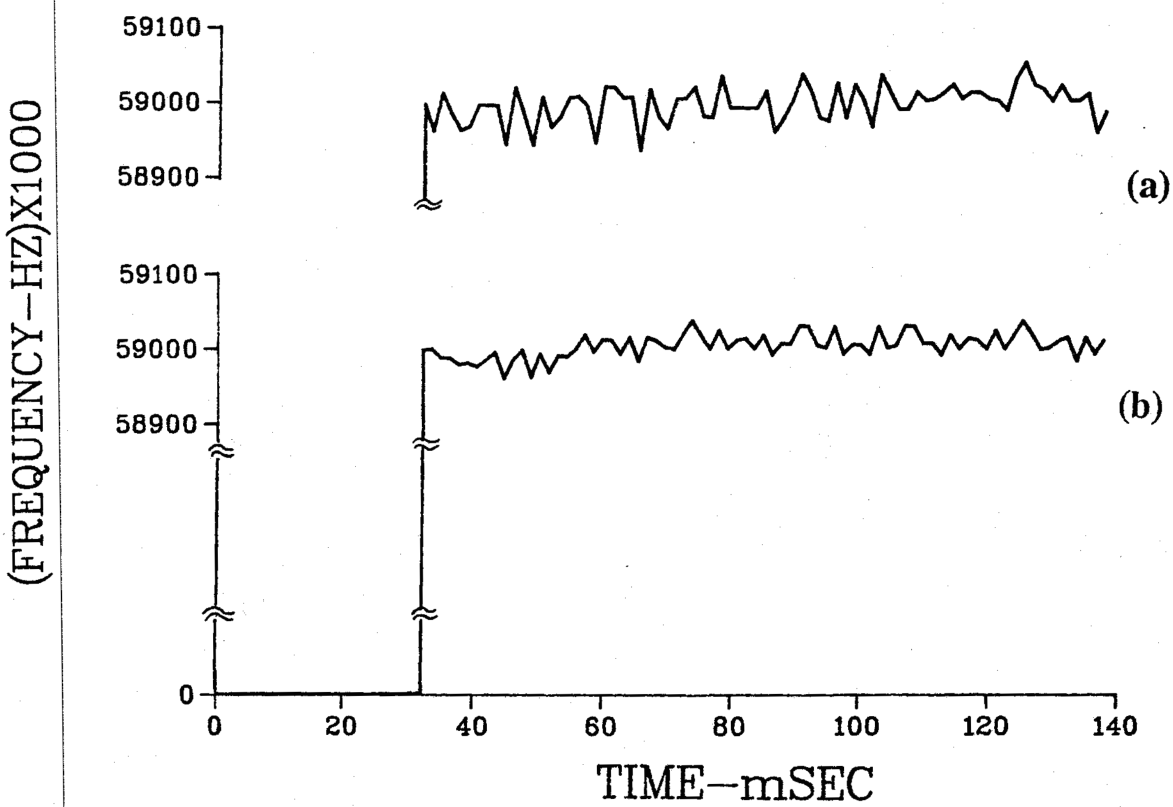


Figure 6.14 The frequency estimates for a 59 Hz input obtained by (a) the 1ϕ *LES* algorithm and (b) the 3ϕ *LES* algorithm using an 8-bit A/D converter model and 12-bit words for multiplications by the bit shift procedure.

repeated using words of different sizes. The frequency estimates obtained using a 12-A/D converter and the 16-bit words for bit multiplications are plotted in Figure 6.15. The errors observed, in this case, are very small. This indicates that the computational errors decrease as the number of bits in the words increase.

6.5 Frequency Response

The program FRQRES was designed to provide frequency response of algorithms. In the software package, the program resides in the workspace FRQRES.APL. This program was tested and debugged to make sure that it is free of errors. The testing procedure followed was similar to the one used for frequency and phasor estimation programs. The frequency responses of the relay algorithms were computed for the selected data windows and sampling rates. The computed responses were compared with the expected responses to check for the software errors.

The frequency response of the one cycle Fourier [11] algorithm was computed using the developed program. The algorithm used for this purpose extracts the 60 Hz components from the input. The sampling rate selected for the algorithm was 720 Hz. The magnitude of the frequency responses of the cosine and the sine filters, computed by the developed program, are plotted in Figure 6.16. As expected, the filter responses are 1.0 p.u. at 60 Hz and 0 p.u. at zero frequency and at frequencies that are multiples of 60 Hz. The computed frequency response of the cosine filter is found to be similar to the response of one cycle Fourier algorithm given in Reference [6].

6.6 Summary

This chapter has reported some of the tests that were carried out to demonstrate the reliability of the developed software package. It has been shown that the modules for A/D converter, multiplication and amplitude estimator perform satisfactorily. Sample tests that illustrate the effectiveness of using these modules with phasor estimation programs have also been reported. Tests carried out to check the programs that provide frequency estimates are then reported. Finally, the use of the program that computes the frequency response of algorithms is demonstrated.

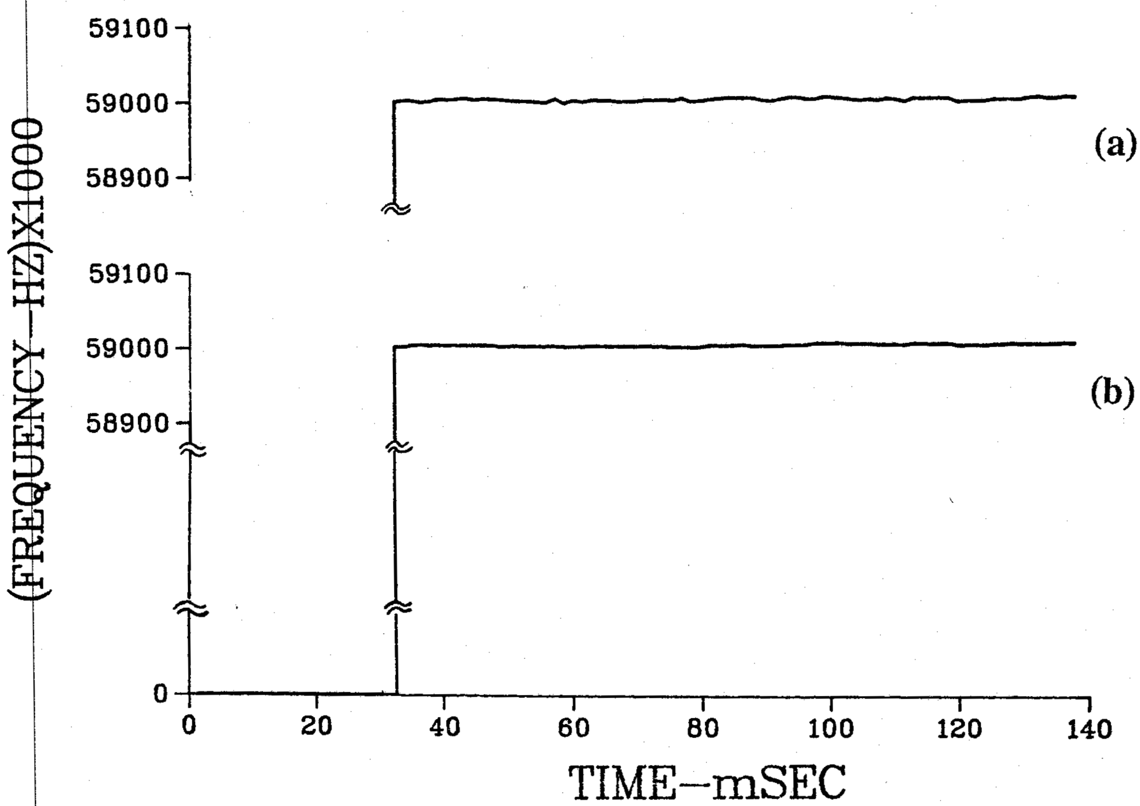
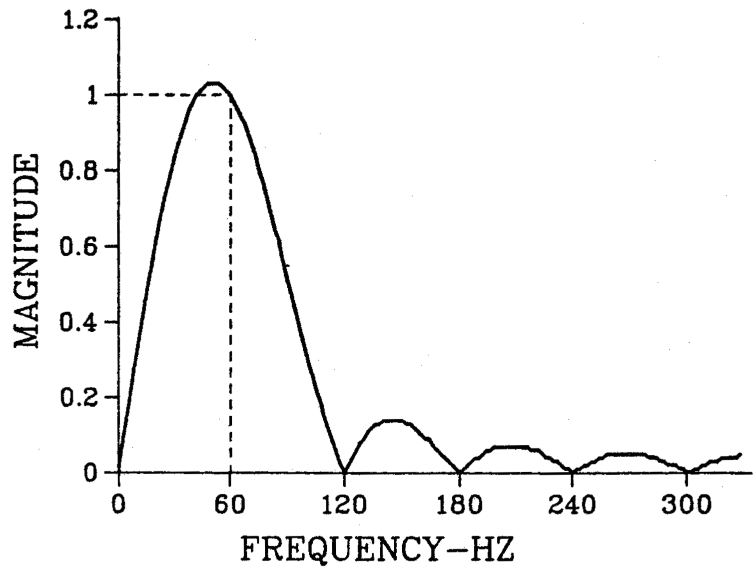
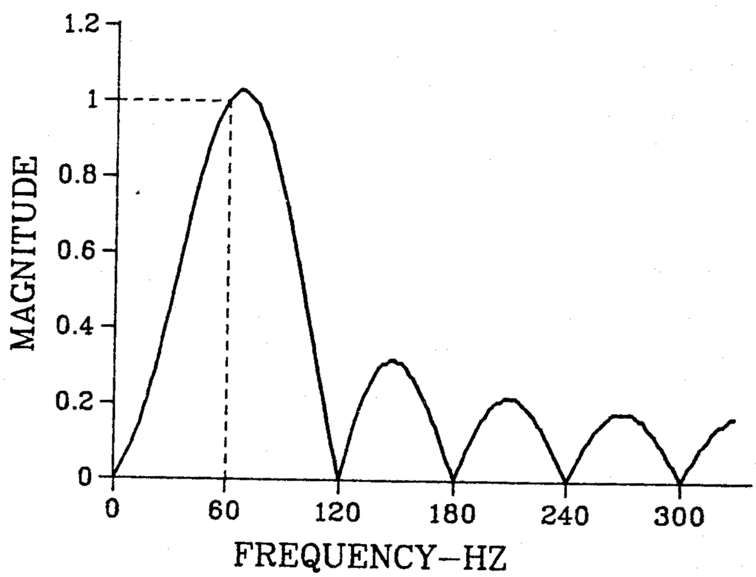


Figure 6.15 The frequency estimates for a 59 Hz input obtained by (a) the 1ϕ *LES* algorithm and (b) the 3ϕ *LES* algorithm using a 12-bit A/D converter model and 16-bit words for multiplications by bit shift procedure.



(a)



(b)

Figure 6.16 Frequency response of the one cycle Fourier algorithm (a) cosine filter and (b) sine filter.

Chapter 7

SUMMARY AND CONCLUSION

The object of this thesis was to develop an interactive software package for the evaluation of digital relaying algorithms including the accuracy of the results when the algorithms are implemented on digital processors.

The major factors that affect the accuracy of the results computed by digital relays are the word sizes of the micro-processors and the A/D converters, the saturation of A/D converters and the relay software. The details of these factors have been outlined in Chapter 2. An essential and critical part of the software of a digital relay is its algorithm. Several algorithms suggested in the past have been described in Chapter 3. Since the design of a digital relay includes the selection of an appropriate relaying algorithm, the factors that should be considered during the selection of an algorithm and the trade-offs that might be necessary have been examined in Chapter 4.

The software package, presented in this thesis, provides an interactive approach for studies of the characteristics of algorithms and their responses to a variety of data that might be encountered in a power system. The specifications of the software, its structure and its special features have been described in Chapter 5. The software developed for this project is of a "question and answer type". The programming language used is APL. The advantages of using APL have also been discussed in Chapter 5. The software includes the programs for

(1) evaluating the algorithms for estimating the phasors.
(2) evaluating the algorithms for estimating the frequency.
(3) computing the impedance of a transmission line as seen from a relay location.
(4) computing the frequency response of algorithms.
(5) generating the instantaneous values of voltage and current samples.

The software also includes the modules that simulate the effects of finite word sizes of micro-processors and A/D converters, and the saturation of A/D converters.

The software facilitates the execution of programs in two modes of operation. In the first mode, the computations are performed using the floating point format of numbers. In the second mode, the effects of A/D converters, truncations or roundings, and bit shift multiplications are incorporated. The computations are performed using the integer format of the numbers. The former mode provides information concerning the errors due to the inadequacies in the designs of algorithms. The latter mode simulates the performance of algorithms if implemented on a digital processor.

The software developed has been tested to check for any errors that may be present. Results from some of the tests are reported in Chapter 6. The tests include the computing of peak values, phase angles and the frequency. The frequency responses of several algorithms have been evaluated. The effects of finite word sizes on the accuracy of the estimates are also illustrated. The results indicate that the errors can be unacceptable if the word size of the A/D converter or the processor is inadequate. A study illustrating the saturation of the A/D converter is also reported.

The new algorithms can be added to the software. The software can also be extended to include programs that would model relay logic and characteristics. These programs could compare the estimates of the power system parameters with pre-programmed relay characteristics for simulating the decision making logic.

This thesis has demonstrated that it is possible to develop a software package for evaluating relay algorithms. The interactive features can be incorporated in the software to enable its effective utilization without the aid of detailed programming manuals. The programming experience gained during the development of this software indicated that APL is an appropriate programming language for this application. This is because APL makes the tasks of providing interactive features and manipulating matrices quite simple. The concept of workspace used in APL provides another useful feature; the user can stop the execution of the program at any stage and deactivate the terminal. Later, the system automatically loads the aborted program into the active workspace and starts executing it from the point where it was left, when the user reconnects the system.

The developed software is of a unique kind because it makes available for performance studies most of the relay algorithms developed in the past. This package, with the extensions suggested above, will prove to be a useful tool for designing digital relays. The present version of the software can be used for selecting an appropriate relay algorithm for use in an application. It can also serve as an educational aid for learning and studying the characteristics of relaying algorithms.

REFERENCES

[1] Last, F. H. and Stalewski, A., "Protective Gear as a Part of Automatic Power System Control", Symposium on Automatic Control in Electricity Supply, IEE Conference Publication No. 16, Part I, Manchester, March 1966.

[2] Rockefeller, G. D.,"Fault Protection with a Digital Computer", *IEEE Transactions on Power Apparatus and Systems*, Vol. 88, April 1969, pp. 438-464.

[3] Liu, B. , "Effect of Finite Word Length on the Accuracy of Digital Filters - A Review", *IEEE Transactions on Circuit Theory*, Vol. CT-18, Nov. 1971, pp. 670-677.

[4] Sachdev, M. S. and Hunchuk, D. G., "Performance of Analog to Digital Converters Used in Computer Relaying", Paper No. A 76 154-5, IEEE PES Winter Meeting and Tesla Symposium New York, N.Y., Jan. 25-30.

[5] Oppenheim, A. V. and Schafer R. W., *Digital Signal Processing*, Prentice Hall Inc., Englewood Cliffs, N.J., 1978.

[6] Sachdev, M. S., Coordinator, *Computer Relaying, Tutorial Text,* Publication No. 79 EH 0148-7-PWR, IEEE, New York, 1979.

[7] Makino, J. and Miki, Y., "Study of Operating Principles and Digital Filters for Protective Relays with Digital Computer", IEEE PES Winter Power Meeting, New York, 1975, Paper No. C75 197-9.

[8] Mann, B. J. and Morrison, I. F., "Digital Calculation of Impedances for Transmission Line Protection", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-90, No. 1, Jan./Feb. 1971, pp. 270-279.

[9] Gilcrest, G. B., Rockefeller, G. D. and Udren, E. A., "High Speed Distance Relaying Using a Digital Computer Part-I-System Description", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-91, No. 3, May/June 1972, pp.1235-1243.

[10] Rockefeller, G. D. and Udren, E. A., "High Speed Distance Relaying Using a Digital Computer Part-II", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-91, No. 3, May/June 1972, pp.1244-1256.

[11] Ramamoorty, M., "A Note on Impedance Measurement Using Digital Computer", *IEE-IRE Proceeding (India)* Vol. 9, No. 6, Nov./Dec. 1974, pp. 243-247.

[12] Phadke, A. G., Hlibka, T. and Ibrahim, M., "A Digital Computer System for EHV Substations: Analysis and Field Tests", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-95, No. 1, Jan./Feb. 1976, pp. 291-301.

[13] Hope, G. S. and Umamaheswaran V. S., "Sampling for Computer Protection of Transmission Lines", *IEEE Transactions on Power and Apparatus and Systems*, Vol. PAS-93, No. 5, Sept./Oct. 1974, pp. 1522-1533.

[14] Hope, G. S., Malik, O. P. and Dash, P. K., "A New Algorithm for Impedance Protection of Transmission Lines," Presented at Summer Meeting of IEEE Power Engineering Society, Vancouver, B. C., July, 1979.

[15] Gilbert, J. G. and Shovlin, R. J., "High-Speed Transmission-Line Fault Impedance Calculation Using a Dedicated Minicomputer", *IEEE Transactions on Power Apparatus and Systems,* Vol. 94, No. 3, May/June 1975, pp. 872-883.

[16] Luckett, R. G., Munday, P. J. and Murray, B. E., "A Substation Based Computer for Control and Protection", IEE Conference Publication No. 125, London, March, 1975, pp.252-260.

[17] Brooks, A. E., "Distance Relaying Using Least Squares Estimates of Voltage, Current and Impedance", Proceeding of 10th Power Industry Computer Application Conference, May, 1977, pp. 394-402.

[18] Sachdev, M. S., and Baribeau, M. A., "A New Algorithm for Digital Impedance Relays", *IEEE Transactions on Power and Apparatus and Systems,* Vol. PAS-98, No. 6, Nov./Dec. 1979, pp. 2232-2240.

[19] McInnes, A. D. and Morrison, I. F., "Real Time Calculation of Resistance and Reactance for Transmission Line Protection by Digital Computer", *EE Transactions of Institution of Engineers, Australia,* Vol. EE 7, No. 1, 1970, pp. 16-23.

[20] Ranjbar, A. M. and Cory, B. J., "An Important Method for the Digital Protection of High Voltage Transmission Lines", *IEEE Transactions on Power Apparatus and Systems,* Vol. 94, No. 2, March/April 1975, pp. 544-550.

[21] Girgis, A. A., Brown, R. G., "Application of Kalman Filtering in Computer Relaying", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-100, No. 7, July 1981, pp. 3387-3397.

[22] Dasgupta, K. S., Malik, O. P. and Hope, G. S., "Kalman Filtering Approach for Impedance Protection", *Transactions of the Engineering & Operating Division Canadian Electrical Association,* Vol. 22, Part 4, No. 83-SP-171.

[23] Sachdev, M. S., Wood, H. C. and Johnson, N. G., "Kalman Filtering Applied to Power System Measurements for Relaying", Paper No. 85 WM 130-0 Presented at IEEE PES Winter Meeting, New York, N.Y., 1985.

[24] Johnson, N. G., Sachdev M. S. and Wood, H. C., "Use of Kalman Filtering Technique for Power System Protection-An Insight", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-104, No. 12, Dec. 1985, pp. 3065-3073.

[25] Sachdev, M. S. and Giray M. M., "A Digital Frequency and Rate of Change of Frequency Relay", *Transactions of the Engineering & Operating Division Canadian Electrical Association,* Vol. 17, Part 3, 1978, No. 78-SP-145.

[26] Nguyen C. T. and Srinivasan, K., "A New Technique for Rapid Tracking of Frequency Deviations Based on Level Crossings", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-103, No. 8, August 1984, pp. 2230-2237.

[27] Girgis, A. A. and Ham, F. M., "A Quantitative Study of Pitfalls in the FFT", *IEEE Transactions on Aerospace and Electronic Systems,* Vol. AES-16, No. 4, July 1980, pp.434-439.

[28] Girgis, A. A. and Ham, F. M., "A New FFT-Based Relay for Load Shedding", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-101, No. 2, Feb. 1982, pp. 433-439.

[29] Phadke A. G., Throp, J. S. and Adamiak, M. G., "A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency, and Rate of Change of Frequency", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-102, No. 5, May 1983, pp. 1025-1038.

[30] Sachdev, M. S. and Giray, M. M., "A Least Errors Squares Technique for Determinating Power System Frequency", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-104, No. 2, Feb. 1985, pp. 437-444.

[31] Cline C. and Sachdev M. S., "Documentation on files PEAKVA.APL, ZVALUE.APL INDATG.APL", Internal Report, 1983-84, Department of Electrical Engineering, University of Saskatchewan, Saskatoon.

[32] Sachdev M. S. and Giray M. M. "Amplitude and Frequency Measurements by Discrete Fourier Transform and Least Error Squares Techniques", *Transactions of the Engineering & Operating Division Canadian Electrical Association,* Vol. 24, Part 3, 1985, No. 85-SP-150.

[33] *Data-Acquisition Databook 1982,* Volume 1, Integrated Circuits, Analog Devices.

[34] Schweitzer, E. O. and Aliaga Antenor, "Digital Programmable Time-Parameter Relay offers Versatility and Accuracy", *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-99, No. 1, Jan./Feb. 1980.

[35] Gilbert, J. G., Udren, E. A. and Sackin, M., "Evaluation of Algorithms for Computer Relaying", IEEE PES Summer Meeting, Mexico City, July 1977, pp. 1-8.

[36] Sidhu T. S., *Computer-Aided Design and Performance Evaluation of Digital Relays,* M.Sc. Thesis, August 1985, University of Saskatchewan, Saskatoon.

[37] Sachdev, M. S., Sidhu, T. S. and Wood, H. C., "Simulation of Digital Relays on a Main Frame Computer for Performance Evaluation", Proceeding of IEEE Electronicom '85, October 1985, Toronto.

[38] Westinghouse Electric Corporation, *Applied Protective Relaying,* a book, Newark, N. J. 1976.

[39] Brigham, E. Oran, *The Fast Fourier Transform,* Prentice Hall Inc., Englewood Cliffs, N.J.

[40] *APLSF Programmer's Reference Manual,* Digital Equipment Corporation, Maynard, Massachusetts, 1977.

[41] *VAX-11 APL Reference Manual,* Digital Equipment Corporation, Maynard, Massachusetts, 1983.

[42] Katzan, Harry Jr., *APL User's Guide,* Computer Science Series, Van Nostrand Reinhold Company.

Appendix A

FREQUENCY RESPONSE OF DIGITAL FILTERS

A.1 Frequency Response of a Digital Filter

The suitability of a filter can be determined by its frequency response. One of the methods used for this purpose is the Z-transform technique. The transfer function of a non-recursive filter in Z-plane can be represented as follows:

$$H(z) = \sum_{n=0}^{m} C_n Z^{-n} \tag{a.1}$$

where

C_n are the filter coefficients and
Z is a complex variable given by $e^{-j\omega\Delta T}$.

The $(m+1)$ filter coefficients are, generally, referred to as the impulse response of a digital filter. Substituting $e^{-j\omega\Delta T}$ for the complex variable, Z, in Equation a.1 leads to the following equation.

$$H(e^{j\omega\Delta T}) = \sum_{n=0}^{m} C_n e^{j\omega\Delta T} \tag{a.2}$$

In Equation a.2, $H(e^{j\omega\Delta T})$ may be written as follows:

$$H\ (e^{j\omega\Delta T}) = H_r\ (e^{j\omega\Delta T}) + j\ H_i(e^{j\omega\Delta T}) \tag{a.3}$$

or

$$H\ (e^{j\omega\Delta T}) = |H(e^{j\omega\Delta T})|\, e^{j\,\theta} \tag{a.4}$$

where

$$\theta = arg\ \{ H(e^{j\omega\Delta T}) \}$$

Using these equations and varying the frequency, ω, from 0 to $\frac{\omega_s}{2}$, the Nyquist frequency, the magnitude and phase response of a non-recursive filter can be calculated.

A.2 Composite Response of Orthogonal Filters

Normally a digital relaying algorithm comprises of a pair of orthogonal filters, viz., a sine filter and a cosine filter. The frequency response for an orthogonal pair of filters, can be combined to obtain an average composite response.

Consider that the input to a pair of orthogonal digital filters is $V_p\, sin\,(\omega t + \theta)$. The outputs of the filters at the selected frequency will be

$$V_s = K_s V_p \sin(\omega t + \theta - \theta_s) \tag{a.5}$$

$$V_c = K_c V_P \sin(\omega t + \theta - \theta_c) \tag{a.6}$$

where

V_s and V_c are the outputs of the two filters,
K_s and K_c are the gains of the two filters,
θ_s and θ_c are the phase delays of the two filters and
θ is the phase angle of the input signal.

The phase delays of orthogonal filters are displaced by $\frac{\pi}{2}$ radians at all frequencies and the outputs can, therefore, be represented as follows:

$$V_s = A sin\,(\omega t + \psi) \tag{a.7}$$

$$V_c = B cos\,(\omega t + \psi) \tag{a.8}$$

where

A replaces the product, $K_s V_p$,

B replaces the product, $K_c V_p$ and

ψ replaces the difference, $\theta - \theta_s$.

The output of the filters can be combined as follows:

$$V_o^2 = V_s^2 + V_c^2 \tag{a.9}$$

or

$$V_o^2 = \left\{ A sin\,(\omega t + \psi) \right\}^2 + \left\{ B cos\,(\omega t + \psi) \right\}^2 \tag{a.10}$$

Equation a.10 can be mathematically manipulated to get Equation a.11.

$$V_o^2 = \frac{1}{2}(A^2 + B^2) - \frac{1}{2}(A^2 - B^2) cos\, 2(\omega t + \psi) \tag{a.11}$$

In this equation, V_o is the time dependent output. However, the outputs will be time invariant if A and B are equal. The average composite frequency response at a selected frequency is represented as follows:

$$V_o = \frac{1}{\sqrt{2}} \left\{ A^2 + B^2 \right\}^{\frac{1}{2}} \tag{a.12}$$

Appendix B

ANALOG TO DIGITAL CONVERTER MODELLING

B.1 Introduction

The purpose of an analog to digital converter is to convert the electrical analog quantities to their corresponding digital representation. The parameters of the selected A/D converter are used to simulate the effect of the A/D converter. These parameters include the word size of the A/D converter, the range of the input signal that the converter can handle, and the quantization method i.e. whether the A/D converter truncates or rounds the inputs to the nearest integer values.

B.2 Modelling

The effect of an A/D converter on digital relays can be studied by developing a simulation model. If a b+1-bit A/D converter is selected with positive saturation voltage, $V_{ref\,(+)}$, and the negative saturation voltage, $V_{ref\,(-)}$, the full scale range [33], FSR, of the A/D converter can be expressed as

$$FSR = V_{ref\,(+)} - V_{ref\,(-)}. \tag{b.1}$$

If a voltage of value x volts is given as an input to the A/D converter model, the output would be an integer number that is obtained using Equation b.2.

$$(Z)_{10} = \left\{ TRN \text{ or } RND \; \frac{x + \frac{FSR}{2}}{FSR} 2^{b+1} \right\} - 2^b \tag{b.2}$$

where

$(Z)_{10}$ is an integer value with a base 10,

TRN represents the truncation and

RND represents the rounding.

The errors due to truncation from Equation b.2 are negative for both positive and negative numbers and the equation represents a b+1-bit A/D converter model that uses a two's complement representation for the negative inputs.

The saturation effect of an A/D converter can be simulated by selecting the input voltage levels such that positive inputs exceed $V_{eff\,(+)}$ and the negative inputs are less than $V_{eff\,(-)}$. The limits $V_{eff\,(+)}$ and $V_{eff\,(-)}$ define the effective range of A/D conversion. For two's complement arithmetic and b+1-bit A/D converters, the limits are defined by Equations b.3 and b.4.

$$V_{eff\,(+)} = V_{ref\,(+)}\left[\,1 - \frac{1}{2^b}\,\right] \qquad \text{(b.3)}$$

$$V_{eff\,(-)} = V_{ref\,(-)} \qquad \text{(b.4)}$$

Appendix C

BIT SHIFT APPROACH OF MULTIPLICATION

C.1 Introduction

The bit shift approach of multiplication is used in calculations for obtaining the product of an integer number with a decimal value. This approach of multiplication is implemented so that the user can demonstrate the accuracy of an algorithm when used with a microprocessor. The bit shift approach of multiplication is described below.

C.2 Mathematical Background

If d is a decimal number, it can be expressed in terms of inverse powers of two as follows:

$$d = \frac{m}{2^1} + \frac{n}{2^2} + \frac{p}{2^3} + \frac{q}{2^4} + \ldots \qquad (c.1)$$

where

m, n, p, q etc., are either 1 or 0.

In a digital computer, each of the above non-zero fractions when multiplied by an integer, is equivalent to carrying out right bit shift operations on the integer value. The number of shift operations is equal to the power of two in the denominator of the fraction. Using Equation c.1, it can be shown that

$$Id = I\left(\frac{m}{2^1} + \frac{n}{2^2} + \frac{p}{2^3} + \frac{q}{2^4} + \ldots\right) \qquad (c.2)$$

or

$$Id = I - I\left(1 - \left(\frac{m}{2^1} + \frac{n}{2^2} + \frac{p}{2^3} + \frac{q}{2^4} + \ldots\right)\right) \qquad (c.3)$$

where

I is the integer value.

C.3 Algorithm

This section describes the algorithm used to simulate the procedure of bit shift multiplication in APL software.

(1) <exec> Multiply the decimal argument by 2^{b+1}, where b+1 is the number of bits to be used.

(2) <exec> Find the binary equivalent of the integer portion of the value obtained in step 1.

(3) <if> The number of ones are more than the number of zeros in binary representation:

<then> Complement the binary representation.

<else> Continue.

(4) <exec> For each one in the binary representation, multiply its inverse value by the integer argument and then truncate or round-off the results.

(5) <exec> Add all the results obtained in step 4.

(6) <if> The binary representation used in step 4 was the one obtained by complementing in step 3:

<then> Subtract the sum obtained in step 5 from the integer argument to get the final product, *DI*.

<else> The sum obtained in step 5 is the final product.

(7) <return >

Step 4 of the algorithm represents right bit shift operations on the integer argument, I. Shifting of a binary number to the right may not divide the number exactly by two. If the integer, I, is divided by two using a bit shift operation, the result would be $\frac{I}{2}+\varepsilon$, ε represents the error. For multiple shift operations, the errors continue to accumulate. Therefore, if the number of ones in the binary representation are more than the number of zeros, the complement of the representation is used. The multiplication described until now is the ordinary form of the bit shift approach. The errors due to truncation or rounding can further be reduced by implementing the extended form of bit shift multiplication. The extended form is similar to ordinary bit shift multiplication except that the integer number is first multiplied with the largest number that can be stored in the word of the processor, 2^{b+1}. The result is multiplied with a decimal value using the bit shift approach described above. The end product is then divided by the number with which the integer was multiplied. The process of multiplying the integer argument by 2^{b+1} is equivalent to making the integer argument two b+1-bit words long with the lower word filled with zeros. This eliminates truncation or rounding errors during the right shift operations. The error due to truncation or rounding takes place only when the final product is divided by 2^{b+1} which is equivalent to ignoring the lower b+1-bit word of the product.

Appendix D

AMPLITUDE ESTIMATOR

D.1 Introduction

Most of the relaying applications require the computations of the amplitudes of phasors from their real and imaginary components. This process requires two squaring functions and one square root function to be performed. These functions are computationally inefficient to implement on micro-processors. However, these calculations can be replaced by a piecewise linear approximation [34]. The approximation requires only addition and multiplication by constants and is much more efficient than the general purpose multiplication of two unknowns. The error in the estimate can be made arbitrarily small. This method of estimation is described below.

D.2 Mathematical Background

Consider that R and I are the real and imaginary components of a phasor having magnitude, M. These components represent a point in a complex plane. The amplitude of the phasor is the distance from the origin to that point. The amplitude is unaffected by the signs of R and I, and points (R, I) and (I, R) are equidistant from the origin. Let

$$U = \max(|R|, |I|) \tag{d.1}$$

$$V = \min(|I|, |R|) \tag{d.2}$$

Then (U, V) is a point in the first octant at a distance, M, from the origin. As a first approximation, let

$$\hat{M} = aU + bV \tag{d.3}$$

Where a and b are the coefficients or the multipliers and can be found to minimize the error $\hat{M} - M$.

This is the case when only one approximation region is considered. Additional accuracy is easily obtained by subdividing the first octant in multiple regions and using different approximations in each region. But the average processing time increases as the number of regions is increased. In general, for n regions there will be a set of value of a and b, (a_1, b_1), (a_2, b_2),...,(a_n, b_n). Each set of coefficients a and b, is valid for a particular range of $\frac{V}{U}$ ratios. The values for each set of coefficients can be calculated using least error squares fit of the data (of a particular region for which values of a and b are being evaluated) to a linear equation of the form of Equation d.3. If m is the number of data points used to fit the data from the n^{th} region, then in matrix form it can be written as

$$\underset{m\times 1}{[M]} = \underset{m\times 2}{[U \mid V]} \underset{2\times 1}{\begin{bmatrix} a_n \\ b_n \end{bmatrix}} . \qquad \text{(d.4)}$$

Values of unknowns a_n and b_n can be calculated using Equation d.5.

$$\underset{2\times 1}{\begin{bmatrix} a_n \\ b_n \end{bmatrix}} = [\underset{2\times m}{[U \mid V]^T} \underset{m\times 2}{[U \mid V]}]^{-1} \underset{2\times m}{[U \mid V]^T} \underset{m\times 1}{[M]} \qquad \text{(d.5)}$$

The coefficients were calculated for one, two, three, four, five, six, eight, ten and sixteen regions of approximations. All these approximations have been programmed in the developed software. The choice to select an approximation is made by the user.