

PERFORMANCE ENHANCEMENT OF DIGITAL  
RELAYS FOR TRANSMISSION LINE  
DISTANCE PROTECTION

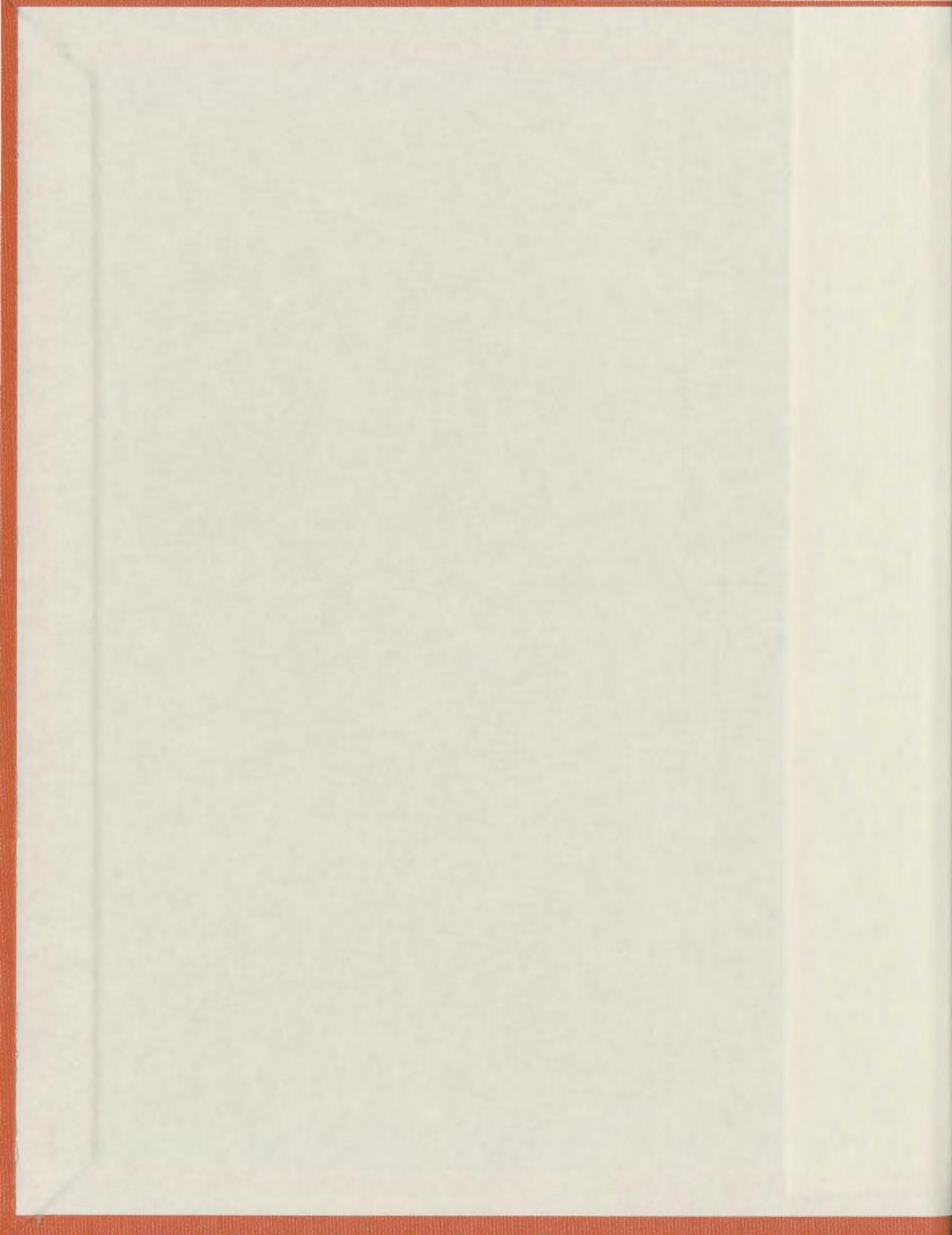
CENTRE FOR NEWFOUNDLAND STUDIES

---

**TOTAL OF 10 PAGES ONLY  
MAY BE XEROXED**

(Without Author's Permission)

FENG LIANG







National Library  
of Canada

Bibliothèque nationale  
du Canada

Acquisitions and  
Bibliographic Services

Acquisitions et  
services bibliographiques

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*

*ISBN: 0-612-89648-X*

*Our file* *Notre référence*

*ISBN: 0-612-89648-X*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

**Canada**



# **PERFORMANCE ENHANCEMENT OF DIGITAL RELAYS FOR TRANSMISSION LINE DISTANCE PROTECTION**

**By**

**© Feng Liang, B. Eng.**

**A thesis submitted to the School of Graduate**

**Studies in partial fulfillment of the**

**Requirements of the degree of**

**Master of Engineering**

**Faculty of Engineering and Applied Science**

**Memorial University of Newfoundland**

**January 2003**

**St. John's**

**Newfoundland**

**Canada**

## **Abstract**

This thesis investigates the performance enhancement of digital relays for transmission line distance protection from filtering (extracting the fundamental frequency components) point of view and relaying algorithm point of view.

Several Fourier based filtering algorithms are investigated, simulated, and compared in this thesis. Furthermore, a new wavelet transform based real-time filtering algorithm is proposed. The studies suggest that this method converges faster than Fourier based algorithms.

Two types of relaying algorithms are also examined in this thesis. One algorithm, which compensates the fault resistance, only needs to monitor the voltage and current signal at the relay end. The other algorithm makes use of the information from both ends of the transmission line and adaptively adjusts the relay operating characteristics. Both theoretical analysis and simulation results show that these algorithms can significantly enhance the performance of digital relays for transmission line distance protection.

## **Acknowledgments**

I would like to thank and express my indebtedness and heartiest gratitude to my supervisor Dr. B. Jeyasurya for his constant advice, encouragement and guidance during all stages of this research.

I would also like to express my sincere thanks to other professors in the faculty of Engineering at Memorial University of Newfoundland for many valuable discussions and useful suggestions.

I take this opportunity to express my profound gratitude to my wife for her constant support and encouragement during my study in Canada.

Thanks are also due to friends for all the useful discussions and suggestions.

Finally, I would like to gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada and Memorial University of Newfoundland, which made this research possible.



# Table of Contents

Abstract .....	i
Acknowledgments .....	ii
Table of Contents.....	iii
List of Figures .....	viii
List of Tables .....	xiii
Chapter 1      Introduction .....	1
1.1      Protective Relaying of Power Systems .....	1
1.1.1    Importance of Protective Relays in Power System .....	1
1.1.2    Common Features of Protective Relays .....	2
1.1.3    Evolution of Protective Relays.....	6
1.2      Aim of the Thesis .....	9
1.3      Organization of the Thesis .....	11
Chapter 2      Digital Protective Relay Configuration.....	12
2.1      Introduction .....	12
2.2      Signal Conditioning Subsystem .....	13
2.3      Conversion Subsystem .....	15
2.3.1    The Sampling Theorem.....	15

2.3.2	Sample and Hold Circuits .....	16
2.3.3	A/D Converter .....	16
2.3.4	Digital Multiplexer .....	17
2.4	Digital processing Subsystem .....	17
2.5	Communication Subsystem.....	18
2.6	Summary .....	18
Chapter 3	Filtering Algorithms for Distance Relays .....	20
3.1	Introduction .....	20
3.2	Theoretical Analysis for Fourier-based Algorithms.....	22
3.2.1	Discrete Fourier Transform Algorithms.....	23
3.2.2	One-Cycle Cosine Window with a Quarter-Cycle Delay Algorithm.....	25
3.2.3	One-Cycle Cosine Window with One Sample Delay Algorithm.....	27
3.3	Simulations for Fourier-Based Algorithms .....	29
3.3.1	Power System Model .....	30
3.3.2	Simulation Results.....	31
3.4	Comparison of the Algorithms .....	36
3.5	Wavelet Transform.....	40
3.5.1	Introduction .....	40
3.5.2	The Conception of Wavelet.....	42
3.5.3	Continuous Wavelet Transform and its Physical Explanations .....	43
3.5.4	Discrete Wavelet Transform .....	46
3.5.4.1	DWT.....	46

3.5.4.2	Maximal Overlap Discrete Wavelet Transform (MODWT).....	53
3.5.4.3	DWT versus MODWT .....	55
3.5.5	Practical Considerations.....	57
3.5.5.1	Choice of Wavelet.....	57
3.5.5.2	Requirements for Sampling.....	58
3.5.5.3	Boundary Conditions.....	59
3.5.6	Summary .....	59
3.6	Phasor extraction by wavelet transform .....	59
3.6.1	Introduction .....	59
3.6.2	Algorithm description .....	60
3.6.2.1	Selection of the Wavelet .....	60
3.6.2.2	Frequency Response Characteristics.....	62
3.6.2.3	Transient Characteristics .....	68
3.6.2.4	Data Pre-Processing .....	70
3.6.2.5	Boundary Issues .....	71
3.6.2.6	Block Diagram of MODWT Algorithm.....	73
3.6.2.7	MODWT and FCDFT Algorithm Comparison.....	74
3.7	Simulations for Wavelet-Based Algorithms .....	78
3.8	Summary .....	85
Chapter 4	Methods for Digital Relay Performance Enhancement.....	87
4.1	Introduction .....	87
4.2	Reactance Effect.....	88

4.3	Transmission Line Fault Location.....	90
4.3.1	Introduction of Fault Location .....	90
4.3.2	The Differences between Fault Location and Fault Detection.....	91
4.3.3	Fault Location Algorithms and Advanced Modern Digital Relays.....	93
4.3.4	Factors Affecting the Fault Location Accuracy .....	93
4.4	A One Terminal Data Based Fault Location Algorithm .....	95
4.4.1	Algorithm Description.....	95
4.4.2	Simulation Results.....	99
4.5	Adaptive Relaying.....	102
4.6	An Adaptive Boundary Algorithm .....	103
4.6.1	Algorithm Description.....	103
4.6.2	Simulation Results.....	104
4.7	Summary .....	106
Chapter 5	Conclusions and Future Work.....	108
5.1	Contributions of the Research.....	108
5.2	Suggestions for Future Work .....	111
Appendix	Simulation Tools for Power System Analysis .....	112
1.	Introduction .....	112
2.	PSCAD/EMTDC.....	113
3.	Pspice .....	116
4.	Matlab.....	117

5.	Simulation Example .....	120
6.	Summary .....	125
References	.....	127

## List of Figures

Figure 1.1.	Typical single line ac connections of a protective relay .....	2
Figure 1.2.	Logical representation of protective relays .....	2
Figure 1.3.	The protective zone coordination of a relay system.....	5
Figure 2.1.	Configuration of the signal conditioning subsystem.....	13
Figure 2.2.	Configuration of the conversion subsystem .....	15
Figure 2.3.	Configuration of a Digital processing subsystem .....	17
Figure 2.4.	Functional blocks of a digital relay .....	19
Figure 3.1.	Filtering windows used by standard DFT algorithm.....	24
Figure 3.2.	Data window used by a quarter-cycle delay algorithm .....	26
Figure 3.3.	Frequency responses of different windows for Fourier associated algorithm .....	27
Figure 3.4.	Computing scheme to determine phase angle ( $\phi$ ) .....	28
Figure 3.5.	Block diagram of a digital distance relay.....	29
Figure 3.6.	The single line diagram of the target system.....	30
Figure 3.7.	Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.75$ ) – the Fourier based algorithms.....	32
Figure 3.8.	Variation of relay impedance for a single-phase-to-ground fault ( $m=0.75$ ) – the Fourier based algorithms.....	32
Figure 3.9.	Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.90$ ) – the Fourier based algorithms.....	33

Figure 3.10.	Variation of relay impedance for a single-phase-to-ground fault ( $m=0.90$ ) – the Fourier based algorithms.....	33
Figure 3.11.	Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.75$ ) – the Fourier based algorithms.....	34
Figure 3.12.	Variation of relay impedance for a three-phase-to-ground fault ( $m=0.75$ ) – the Fourier based algorithms.....	35
Figure 3.13.	Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.90$ ) – the Fourier based algorithms.....	35
Figure 3.14.	Variation of relay impedance for a three-phase-to-ground fault ( $m=0.90$ ) – the Fourier based algorithms.....	36
Figure 3.15.	Relative errors of the Fourier based filtering algorithms for single-phase- to-ground faults .....	39
Figure 3.16.	Relative errors of the Fourier based filtering algorithms for three-phase- to-ground faults .....	40
Figure 3.17.	Flow diagram at scale level 1 .....	51
Figure 3.18.	Flow diagram at scale level $j$ (DWT).....	52
Figure 3.19.	Flow diagram at scale level $j$ (MODWT).....	54
Figure 3.20.	Effect of Shifts on DWT .....	56
Figure 3.21.	Effect of Shifts on MODWT .....	57
Figure 3.22.	Frequency Response at Scale 1 .....	65
Figure 3.23.	Frequency Response at Scale 2 .....	65
Figure 3.24.	Frequency Response at Scale 3 .....	66

Figure 3.25.	The Overall Frequency Response of the MODWT pyramid algorithm using Db(8).....	67
Figure 3.26.	The waveform of $x(t)$ .....	68
Figure 3.27.	The waveform of $x(t)$ and $x_{s3}$ in window1.....	69
Figure 3.28.	The waveform of $x(t)$ and $x_{s3}$ in window2.....	70
Figure 3.29.	The waveforms of $x(t)$ , 60 Hz signal and $x_{s3}$ .....	71
Figure 3.30.	The equivalent magnitudes for $x_{s3}$ .....	72
Figure 3.31.	Block diagram of the MODWT algorithm.....	74
Figure 3.32.	Block diagram of the FCDFT algorithm.....	75
Figure 3.33.	The frequency response for FCDFT.....	76
Figure 3.34.	Comparison of Convergence Characteristics.....	77
Figure 3.35.	Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.75$ ) – the MODWT algorithm vs. the FCDFT algorithm.....	79
Figure 3.36.	Variation of relay impedance for a single-phase-to-ground fault ( $m=0.75$ ) – the MODWT algorithm vs. the FCDFT algorithm.....	80
Figure 3.37.	Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.90$ ) – the MODWT algorithm vs. the FCDFT algorithm.....	80
Figure 3.38.	Variation of relay impedance for a single-phase-to-ground fault ( $m=0.90$ ) – the MODWT algorithm vs. the FCDFT algorithm.....	81
Figure 3.39.	Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.75$ ) – the MODWT algorithm vs. the FCDFT algorithm.....	82



Figure 3.40.	Variation of relay impedance for a three-phase-to-ground fault ( $m=0.75$ ) – the MODWT algorithm vs. the FCDFT algorithm .....	82
Figure 3.41.	Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.90$ ) – the MODWT algorithm vs. the FCDFT algorithm .....	83
Figure 3.42.	Variation of relay impedance for a three-phase-to-ground fault ( $m=0.90$ ) – the MODWT algorithm vs. the FCDFT algorithm .....	83
Figure 3.43.	Relative errors of the filtering algorithms for single-phase-to-ground faults – the MODWT algorithm vs. the FCDFT algorithm.....	84
Figure 3.44.	Relative errors of the filtering algorithms for three-phase-to-ground faults – the MODWT algorithm vs. the FCDFT algorithm.....	85
Figure 4.1.	Equivalent circuit for a fault on a transmission line using two-port model .....	88
Figure 4.2.	Influence of fault resistance on distance relay .....	89
Figure 4.3.	A faulted network and its equivalent decomposition .....	97
Figure 4.4.	Impedance Trajectory with Error Compensation (Fault at $m = 0.75$ )..	100
Figure 4.5.	Impedance Trajectory with Error Compensation (Fault at $m = 0.90$ )..	101
Figure 4.6.	Impedance Trajectory with Adaptive Boundary (Fault at $m = 0.75$ , Power flow into the system behind the relay) .....	105
Figure 4.7.	Impedance Trajectory with Adaptive Boundary (Fault at $m = 0.75$ , Power flow into the system in front of the relay).....	105
Figure A.1.	Single line diagram of a two bus system.....	121
Figure A.2.	EMTDC/PSCAD simulation scheme for the two bus system.....	122

Figure A.3.	EMTDC/PSCAD simulation result for the two bus system.....	122
Figure A.4.	PSPICE simulation scheme for the two bus system.....	123
Figure A.5.	PSPICE simulation result for the two bus system.....	123
Figure A.6.	Equivalent faulted circuit for the two bus system .....	124
Figure A.7.	Matlab simulation result for the two bus system.....	125

## List of Tables

Table 3.1	Parameters of the study system - 1 .....	30
Table A.1.	Parameters of the study system - 2 .....	120

# Chapter 1

## Introduction

### ***1.1 Protective Relaying of Power Systems***

#### ***1.1.1 Importance of Protective Relays in Power System***

In power systems, many kinds of faults and abnormal operating conditions may occur. The most common, and also the most harmful, are short circuits of different kinds. Short circuit current can damage equipment of power systems, reduce the voltage of relevant parts of the system, and even threaten the stability of the system.

Faults are not only caused by lightning, wind, ice, earthquake, fire, explosions, falling trees, flying objects, physical contact by animals and other natural events, but are also caused by equipment failure, incorrect operation etc.. Irrespective of the cause, faults must be isolated quickly and selectively. For this purpose, protective relays are installed at different parts of a power system. The main features are:

- Automatically, quickly, and selectively remove faulted components from power system in order to recover the rest of the parts to normal functioning quickly;

- Indicate the abnormal operation situation of relevant components and provide associated signals.

Nowadays, due to changes in power system operation like reactive power compensation, flexible ac transmission techniques, increased power transfers over transmission lines, protective relays play a much more important role in order to maintain a very high level of continuity of high quality power supply.

### 1.1.2 Common Features of Protective Relays

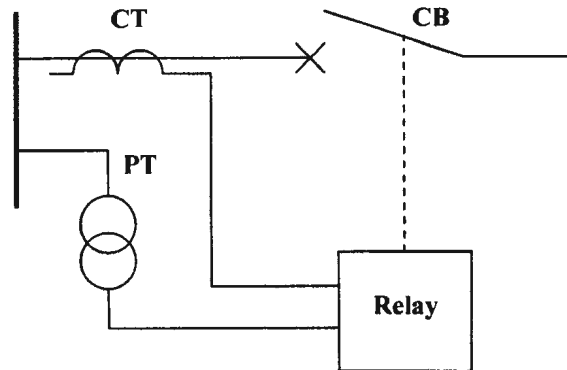


Figure 1.1. Typical single line ac connections of a protective relay

Figure 1.1 illustrates a typical simplified relationship between the Current Transformer (CT), Potential Transformer (PT), protective relay, and Circuit Breaker (CB). Protective relays take CT, PT signals as inputs and provide trip signals for the circuit breaker.

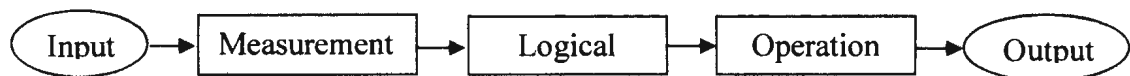


Figure 1.2. Logical representation of protective relays

In general, a typical logic representation of protective relays, such as electromechanical, solid state, or digital relays, is shown as in Figure 1.2. As it indicates, protective relays consist of measurement components, logical components, and operation components.

1. Measurement - it measures relevant quantities which are necessary for particular protective relays, such as voltage (from PT), and/or current (from CT), and compares with given setting point value to determine whether the relay should be started or not.
2. Logical - relaying decision is based on certain computations that use the measurement values.
3. Operation - relaying action is based on the decision in part 2. For example, in case of faults, trip the breaker; in case of abnormal situation, give signals; in case of normal condition, no action or just provide some self-monitoring functions (digital relays only).

Protective relays, consisting of three components indicated in Figure 1.2, are designed to satisfy the following four basic requirements.

1. Reliability

Reliability includes two aspects: protective relays must operate when intolerable transients and permanent faults occur within their protection area, and must not operate when any fault occurs outside their protection area or any kinds of tolerable transients.

Both incorrect operation and failure to operate are extremely harmful to the power system. However, it is difficult to avoid both unexpected operations at the same time. Due to different power systems, or different load conditions, incorrect operation and failure to operate can cause different effects to the power system. Therefore, different strategies should be taken depending on the operating conditions.

## 2. Selectivity

Selectivity (also known as relay coordination) of protective relays means, when protective relays operate, they only isolate the faulted element from the whole system to minimize the outage area and make the rest of power system maintain normal power supply.

The possibility of failure to operate and failure of protective relays and circuit breakers should be considered in determining the selectivity of protective relays. Therefore, backup relays are necessary. The coordination of primary relays and backup relays can be achieved by different operation zone and operation time delay. Normally, primary relays have a small operation zone but operate instantaneously, while backup relays have a large operation zone, namely overreached area, and operate with a particular time delay.

Figure 1.3 gives an illustration of the coordination of the different protective zone in a transmission line protective system. Zone 1 is the primary operation zone while Zone 2 is the backup zone. As can be seen, Zone 1 is set to protect 85% of the line without any time delay. In case the relay cannot operate properly, Zone 2, which is set to 150% of the

length of the transmission line, will trip this line with a particular time delay, say 0.2 second.

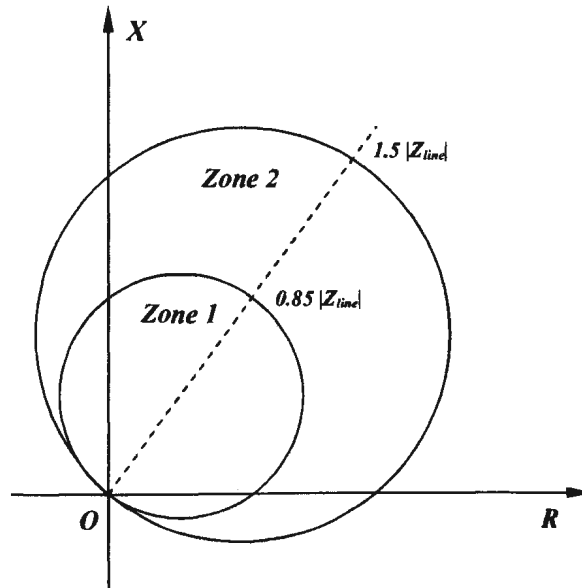


Figure 1.3. The protective zone coordination of a relay system

### 3. Sensitivity

Sensitivity is the response ability of protective relays to the faults or abnormal operation inside the protection zone.

Protective relays with good sensitivity can sense any faults within the protection zone with respect to different fault location, different fault types, and even different fault resistance. The sensitivity factor usually determines the sensitivity of protective relays. It depends on the parameters of protected elements and operation condition of the power system.



#### 4. Speed

Removing the faulted area (or elements) rapidly can significantly improve the stability of the power system, reduce outage duration, and minimize the damage of faulted elements. Therefore, in case a fault occurs, protective relays should identify the faults and operate quickly.

The total time to remove the fault is determined as the sum of operation time of protective relays and operation time of circuit breakers. Typically, high-speed relay can operate in less than 50 ms or less. Modern high-speed circuit breaker can operate in the range of 17 ms to 50 ms, [1]. Therefore, the total clearing time can be within 35 to 130 ms. However, high operation time is not always required especially in low-voltage systems for economic reasons.

Besides the four factors mentioned above, economics of protective relays is another important factor which should be considered. A “good” protective relay system should combine both features of maximum protection and minimum cost.

##### ***1.1.3 Evolution of Protective Relays***

Protective relays were introduced quite widely into power systems during the first two decades of the 20<sup>th</sup> century. From then on, their evolution mainly concerns three generations: electromechanical relay, solid-state relay, and digital protective relay.

## 1. Electromechanical protective relay [2]

In 1898, Andrews promoted a Reverse-current relaying scheme, in one of the earliest literatures to illustrate protective relay of electromechanical type. From then to the 1950s, electromechanical protective relays dominated this area.

Movement of armatures or discs of these relays is used to operate contacts, which in turn issue the tripping signal for circuit breakers. Some elaborate designs enable them to meet the system requirements even up to the present.

As the earliest protective equipment, each electromechanical relay has different and specified function, such as over-current relay, distance relay, signal relay, time relay, and so on. Therefore, it is necessary to organize several different relays to achieve a particular protective function, which in turn cause heavier burden to analyze, design, install, test, and maintain. However, its simplicity and high level of performance made it popular over many years.

## 2. Solid-state protective relay [2]

The later (from 1930s to 1950s) rapid progress in electronic technology, especially the invention of transducers, pushed the development of solid-state protective relay.

Compared to electromechanical relay, the size of solid-state relay reduced significantly. At the same time, some weakness of the former one did not appear in solid-state relay, such as slower resetting, overshooting and so on. Therefore, the relevant analysis, design, installation, testing, and maintenance could be simplified significantly.

However, it must be recognized that solid-state relay must have a reliable power supply and the effect of electrical interference must be taken into account.

### 3. Digital protective relay [3]

From the mid 1970s, digital hardware was used in commercial products. Almost at the same time, Rockefeller explored the possibility of implementing a digital technique in protective relaying area in 1968 [4]. In 1971, Ramamoorthy first proposed the discrete Fourier transform (DFT) algorithm [5], which is widely used at the present, to extract the fundamental frequency (60/50 Hz) components from sampled voltage/current data. The first commercial digital relay appeared in 1979. From then on, digital protective relay became the mainstream product and is widely used in power systems.

With reference to electromechanical relay and analog solid-state protective relay, the protective function of digital relay does not rely on the hardware any more, i.e., based on the same hardware, different algorithms can be easily implemented. In addition, it has the following advantages [6]:

- Self-monitoring: in case the main relay algorithms are not invoked, digital relays can run self-monitoring program to check both hardware and software.
- Remote data communication: it can exchange information with the central computer of the dispatch center in order to make its relay algorithm follow the different power system operation condition.
- Fault recoding: detailed fault data can be stored in digital relay and it can be easily uploaded or downloaded to other computer for further analysis.

- Standard products: it simplifies analysis, design, installation, testing, and maintenance; and also vastly reduces panel space and station battery power needs.
- Economics: standard products can significantly reduce cost.

With the changes in the structure of the electric power utility, the transmission lines carry significant load in an interconnected network. The security of transmission lines is very crucial to ensure that the customers receive the power demand reliably. The performance of the protective relays for transmission line protection is a significant factor in considering the security of the interconnected power system [7]. Therefore, from the customers' point of view, the performance of transmission line protection has to be enhanced in order to meet the power system operating requirement. Since the microprocessor is becoming more and more powerful, a digital relay can deal with more sophisticated and complicated algorithms for protective functions, thus making performance enhancements possible.

## **1.2 Aim of the Thesis**

The main aim of this thesis is to investigate possible means which can efficiently enhance the performance of digital relays for transmission line distance protection. The enhancements can be obtained using two approaches. The first approach concerns the real-time filtering, which extracts the fundamental frequency components of the voltage and currents signals measured by the relay. The second approach is based on the relaying algorithms, which make use of the fundamental components to calculate the (compensated) impedance and compare with some thresholds (operating zone).

In this thesis, several Fourier transform based filtering algorithms are investigated, simulated, and compared. Even though the Fourier based algorithms are widely implemented in current commercial digital relays and can achieve acceptable performance, they still have some drawbacks, such as the transient response characteristics. It is the motivation for many power system engineers to seek other possible filtering algorithms to improve the performance. In this thesis, the author proposes a new wavelet transform based real-time filtering algorithm. Single-phase-to-ground fault and three-phase-to-ground fault cases are considered. The performance of the new algorithm is evaluated with data obtained from the simulation of a two bus system. The results show that the new algorithm converges faster compared to the conventional Fourier based algorithms under different power system operating conditions, such as fault locations, fault resistance, load conditions, etc..

Fast and accurate locating of faults in a power transmission line depend not only on the filtering algorithms, but also on the relaying algorithms. Recently, many innovative and challenging ways have been investigated by power system engineers. Generally, all the algorithms can be classified into two categories. One category only uses the information at one terminal (the relay end) of the transmission line; the other utilizes the information from both terminals of the transmission line. Due to different operating requirements, both of the algorithms are widely implemented in the power system. The one-terminal relaying algorithm, which is investigated in this thesis, can compensate the effects caused by the fault resistance. The two-terminal relaying algorithm studied in this thesis can adaptively adjust its action boundary according to different load conditions.

Theoretical analysis and simulations show that both of the algorithms can significantly enhance the relay performance.

### ***1.3 Organization of the Thesis***

Chapter 2 of this thesis introduces the computer relay architecture. In general, any digital relay can be divided into signal conditioning subsystem, conversion subsystem, digital processing subsystem and communication subsystem.

Three Fourier based filtering algorithms, which include traditional full cycle discrete Fourier algorithm, one-cycle cosine window with a quarter delay algorithm, and one-cycle cosine window with one sample delay algorithm, are presented in Chapter 3. The simulation and comparison of these algorithms are presented. In addition, a new wavelet based real-time filtering algorithm, which follows the introduction of the concept of wavelet transform, is suggested, simulated and compared with the conventional Fourier algorithms.

A brief discussion on reactance effect is presented in Chapter 4. The concept of transmission line fault location is discussed. Then, a one terminal data based fault location algorithm, which compensates the fault resistance, is analyzed and simulated using a two bus power system. After a brief discussion of adaptive relaying, an adaptive boundary algorithm which uses two-terminal data is analyzed and simulated using the same power system.

The summary of the thesis highlighting the contribution of the research and suggestions for future research are outlined in Chapter 5.

## **Chapter 2**

# **Digital Protective Relay Configuration**

### ***2.1 Introduction***

Just as other real-time microprocessor based measurement, monitoring, and control system, digital protective relays have the same configuration, namely, analog input modules, digital input/output modules, Analog to Digital converters (ADC), Random Access Memory (RAM), Read Only Memory (ROM), Central Processor Unit (CPU), and other modules. However, the above-mentioned modules can usually be organized in the following subsystems according to different functions:

- Signal conditioning subsystem
- Conversion subsystem
- Digital processing relay subsystem
- Communication subsystem

Except the digital processing subsystem, all the other subsystems are generally similar for different types of digital protective relays. The former subsystem, i.e., the software, can be different so as to achieve different protective algorithms.

The following sections will discuss the details of the above subsystems. Section 2.2 investigates the signal conditioning subsystem, while Section 2.3 focuses on the conversion subsystem. Section 2.4 presents the signal processing subsystem, and Section 2.5 introduces the communication subsystem. Section 2.6 gives a summary.

## 2.2 Signal Conditioning Subsystem

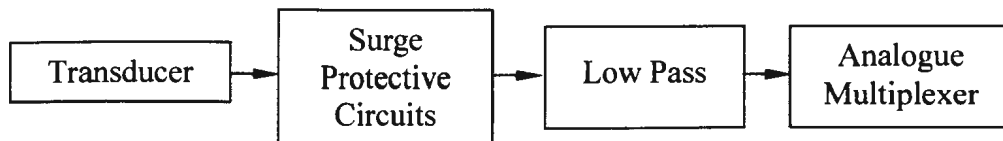


Figure 2.1. Configuration of the signal conditioning subsystem

Figure 2.1 gives the configuration of a signal conditioning subsystem, which includes a transducer, surge protective circuits, low pass filter, and analog multiplexer. All of the four components are connected in series.

### 1. Transducer

Usually, microprocessor-based systems can accept signals within  $\pm 5$  V or 4 to 20 mA, while the signals available from CT are 1A or 5A, and signals from PT are 110V or 120V. Therefore, a highly accurate transducer is necessary for digital protective relays to reduce CT and PT signals to suit the requirements of microprocessor based systems.

Both CT and current transducer can be saturated in case of short circuit so that the accuracy of measurement can be reduced. To minimize this effect, some compensation can be applied [6]. However, in most applications, this is not necessary.



## 2. Surge protective circuits

Surge protective circuits are connected between the transducers and low pass analog filters to avoid possible damage caused by surge current in the case of power system fault. Typically, surge circuits consist of well-organized capacitors and isolating transformers.

## 3. Low pass filter

Almost all the popular digital filter algorithms, such as Full Cycle Discrete Fourier Transform (FCDFT), cannot extract fundamental quantities directly and remove all of the high order harmonics from CT, PT signals. In addition, sampling theorem, which will be illustrated in Section 2.3, requires that the highest harmonics in analog signal should be less than a particular value to avoid errors caused by limited digital sampling frequency. Therefore, analog low pass filters are necessary to filter those unwanted high order harmonics by choosing suitable cut-off frequencies. Of course, this low pass filter can introduce some modifications to both magnitude and phase angle of CT, and PT signals. However, since the transform function of low pass filter can be easily specified, it is not difficult to extract the exact fundamental signal with both correct magnitude and phase angle [8].

Normally, well-designed first order or second order low pass filter can remove high order harmonics and give satisfactory output signals.

#### 4. Analog multiplexer

Since almost every digital protective relay has limited data processing channels (including sample and holding, analog to digital conversion, and so on), all or part of input analog signals have to be processed in a serial manner. An analog multiplexer is used to select a signal from a number of inputs and transfer it to suitable processing channels.

### 2.3 Conversion Subsystem

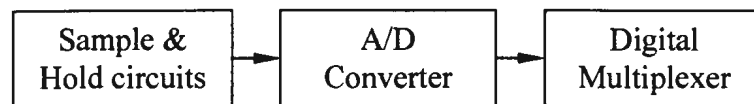


Figure 2.2. Configuration of the conversion subsystem

Conversion subsystems consist of three parts – sample & hold circuits, A/D converters, and digital multiplexers – in serial connection, as shown in Figure 2.2. The sampling theorem is an important basic theory which should be taken into account in this subsystem. Therefore, the sampling theorem and the three parts are explained below.

#### 2.3.1 The Sampling Theorem

The sampling theorem states that a band-limited signal can be uniquely specified by its sampled values if and only if the sampling frequency is at least twice the maximum frequency component contained within the original signal [3], [9], i.e.

$$f_s \geq 2f_m \quad (2.1)$$

$f_s$  is the sampling frequency,

$f_m$  is the maximum significant frequency, or Nyquist frequency.

If equation (2.1) is not satisfied, i.e., the sampling frequency is less than Nyquist frequency, aliasing error – overlap between high frequency signals and low frequency signals – can occur. Therefore, high frequency signals and low frequency signals cannot be clearly distinguished from each other.

In practise, using front-end low pass filters and/or increasing the sampling frequencies are suitable solutions to realize the sampling theorem so as to avoid aliasing error.

### ***2.3.2 Sample and Hold Circuits***

In general, sample and hold circuits are made of switch-capacitor combinations. Every sample can be held in this circuit until new sample is taken.

In this part of the conversion subsystem, synchronisation is extremely important for each sample in order to get precise calculation results. To achieve this target, accurate and uniform system clock is necessary and sampling rate should suit the changing system frequency.

### ***2.3.3 A/D Converter***

As the name indicates, A/D converters convert analog signals to digital signals so that they can be processed by a microprocessor. The most important parameters that characterize the performance of A/D converters are resolution, i.e., the accuracy, and conversion time. Both of these two parameters are extremely important to digital protective relays since they directly affect its total performance.

A/D converters can be classified as counter-controlled converters, dual-slope converter, and parallel comparator converters. Among them, parallel comparator

converters are the fastest ones, and at the same time, they are the most expensive ones to achieve the same precision.

#### **2.3.4 Digital Multiplexer**

Since every input signal has to be transferred to a microprocessor through the same data bus of the processor, digital multiplexers are another indispensable component that has a number of input ports and a single output. Controlled by particular time logic, microprocessors can accept and process every input signal in a desired sequence.

### **2.4 Digital processing Subsystem**

Figure 2.3 shows the configuration of a digital processing subsystem. As can be seen, this subsystem is actually the same as other computer systems, i.e., it contains CPU, memory, digital input and output.

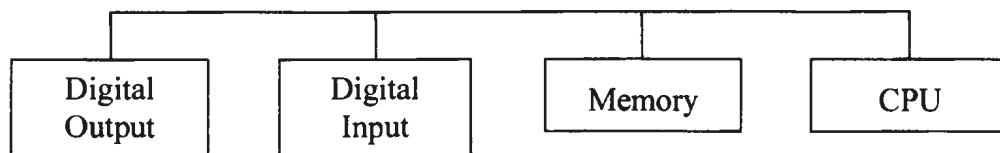


Figure 2.3. Configuration of a Digital processing subsystem

Digital input and output components work as input and output channels for digital signals, such as circuit breaker state signals – input signals, trip signals, and relay malfunction signals – output signals. Memory, including both RAM and ROM, stores digital relay monitoring programs, digital filter programs, protective relaying programs, and other programs. Among them, digital relay monitoring programs coordinate the operation of all other programs and also monitor the relay itself. Digital filter programs can extract fundamentals, high order harmonics, and also subharmonics in some cases

whose typical algorithms include DFT, Walsh transform, and so on. Protective relaying programs achieve different relaying algorithms.

CPU and other components have benefited from fast-developed computer technology and become simultaneously more powerful and cheaper. Therefore, more and more complicated algorithms can be implemented. Typical commercial digital relays today with 16-bit microprocessors can complete protective relay calculation and self-testing programs using a typical sampling rate of 20 samples per cycle of fundamental frequency.

## **2.5 Communication Subsystem**

Different from electromechanical relays and solid-state relays, another outstanding feature of digital relays is their communication ability. By means of computer technology, communication subsystem can easily achieve efficient computer-based communication with a Supervisory Control and Data Acquisition (SCADA) system and an Energy Management System (EMS). Recorded fault data, relay setting, and other data, can be transferred conveniently between local relays and remote control centers or other relays by Twisted Pair Metallic Cable, Coaxial Metallic Cable, Fiber Optic Cable, Power Line Carrier, Microwave Radio, and even Satellite [10].

## **2.6 Summary**

Figure 2.4 shows the functional organization of a digital relay. As can be seen, for different applications, no matter if they are transmission line protective relays, transformer protective relays, generator relays, or other kinds of relays, an identical hardware architecture consisting of signal conditioning subsystems, conversion

subsystems, digital processing relay subsystems, and communication subsystems can be implemented, and each subsystem can be updated and improved separately. This makes it possible for the manufacturer to use almost the same hardware platform to achieve different protective algorithms. Accordingly, the cost/performance ratio can be improved significantly.

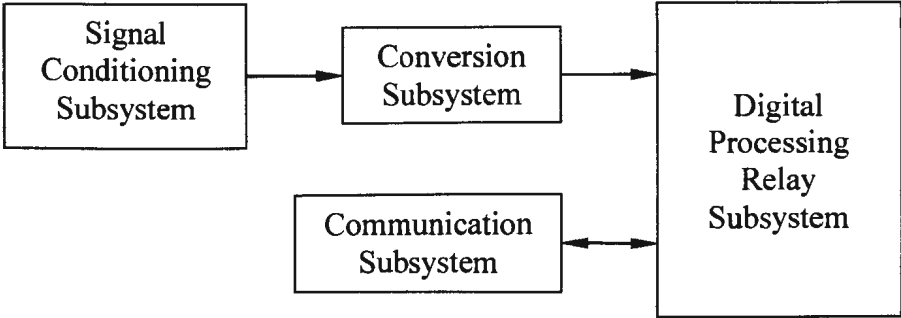


Figure 2.4. Functional blocks of a digital relay

## **Chapter 3**

# **Filtering Algorithms for Distance Relays**

### ***3.1 Introduction***

Power transmission and distribution lines are the vital links that achieve the essential continuity of service from generating plants to the end users. Microprocessor-based distance relays are widely used for transmission line protection. Innovative developments within microprocessor-based relays have created new ways of reacting to the data available and responding to the possible information that can be extracted from this data. Hidden within the distorted current and voltage signals available to a digital relay are the fundamental frequency voltage and current information.

All of the present-day microprocessor-based relays continuously take instantaneous samples of line voltage and current signals and convert them to binary values many times per cycle of the fundamental frequency [6]. Elaborate computations are performed in the microprocessor to convert the sample values to useful relaying measurements such as phasor. These are in turn combined with logical input and settings in the system logic to yield tripping, control and operator information outputs. Development and application of

microprocessor relays from the experimental fringe to the market mainstream has been dramatic during the last decade [11].

The transmission lines carry significant load in an interconnected network. The reliability and availability of transmission lines is a significant factor to ensure that the power demand of the customers is satisfied [7]. The performance of the protective relays for transmission line protection influences the reliability and availability of the interconnected power system. Due to the nature of the power system, the fault voltages and currents measured by protective relays include decaying dc offsets, harmonics and other interferences. The fundamental frequency voltage and current signals contained in this distorted signal can provide valuable information to the relay to respond to a fault.

For digital distance relays, which rely on precise fundamental quantities, the fundamental frequency components of the post fault voltages and currents need to be extracted as quickly and as accurately as possible. Filtering requirements for digital distance relays are very critical. A number of digital filtering algorithms have been proposed, such as Discrete Fourier Transform (DFT) based algorithms, Walsh transform based algorithms [12], Kalman filtering based algorithms [13], Artificial Neural Network (ANN) based algorithms [14] – as a matter of fact, ANN based techniques have very wide implementation in power system, including high impedance fault detection, fault direction discrimination, fault classification, adaptive reclosing, etc. [6]. Any filtering algorithm should converge fast and estimate the desired information quickly.

DFT (Discrete Fourier Transform) is widely used in digital distance relays for phasor calculations [6]. There has been considerable research in developing other



algorithms to provide a fast and accurate estimate of the transmission line impedance for relaying consideration, such as FCDFT (full cycle DFT) algorithms - which need a full cycle data in each processing session, and HCDFT (Half Cycle DFT) algorithms - which use half of a cycle data in a processing session. Since FCDFT algorithms have better transient response characteristics than HCDFT algorithms, and FCDFT algorithms are more widely used in commercial products, several full cycle Fourier-based algorithms are analyzed in Section 3.2, simulated in Section 3.3, and compared in Section 3.4. Furthermore, a wavelet-based approach is proposed and compared with the Fourier-based algorithms. As will be shown, the new approach, which takes the advantages of the good transient response properties of wavelet transform, functions correctly and converges rapidly. Section 3.5 introduces the basic concepts of wavelet transform. The new algorithm is proposed in Section 3.6. Simulations and the comparisons with Fourier based algorithms are presented in Section 3.7. Section 3.8 gives the summary.

### ***3.2 Theoretical Analysis for Fourier-based Algorithms***

The digital filtering algorithms for distance protection can be classified into two categories:

- i) Algorithms based on signal parameter estimation and,
- ii) Algorithms based on power system parameter estimation.

This study focuses on the first type of algorithms. Under normal operating conditions, the operating frequency of the power system can be considered stable at 60 (or 50) Hz. But faults introduce transients, dc offsets and other undesirable harmonic signals. Hidden within the distorted current and voltage signals are the fundamental

frequency voltage and current information. These can be effectively used by protective relays to determine the information regarding a possible fault. Algorithms based on signal parameter estimation extract the fundamental frequency voltage and current phasors [15][16]. DFT based algorithms are discussed first. These algorithms calculate the cosine and/or sine components of the phasor describing the input signal.

### 3.2.1 Discrete Fourier Transform Algorithms

The measurement  $M$ , voltage or current, of the digital distance relay is a time-domain signal. However, to calculate the impedance, its frequency-domain representation is needed. Suppose that the measured phasor is written in terms of the complex form  $a(n) + jb(n)$ . Using DFT,  $a(n)$  and  $b(n)$  can be obtained from

$$\begin{cases} a(n) = \frac{2}{N} \sum_{j=0}^{N-1} M[(n-N+1+j)\Delta t] \cos\left(\frac{2\pi}{N} j\right) \\ b(n) = -\frac{2}{N} \sum_{j=0}^{N-1} M[(n-N+1+j)\Delta t] \sin\left(\frac{2\pi}{N} j\right) \end{cases} \quad (3.1)$$

In equation (3.1),  $n$  denotes the sampling number;  $N$  is the sample numbers in one fundamental cycle;  $\Delta t$  is the sampling interval. To determine  $a(n)$  and  $b(n)$ , two orthogonal window functions, one cosine window and one sine window, with the length of  $N$  samples are needed. Figure 3.1 illustrates these two windows.

Equation (3.1) shows a Finite Impulse Response (FIR) filter characteristics, i.e., it uses the current sample as well as  $N-1$  previous samples to obtain the current output. This algorithm requires one cycle delay to extract the phasor.

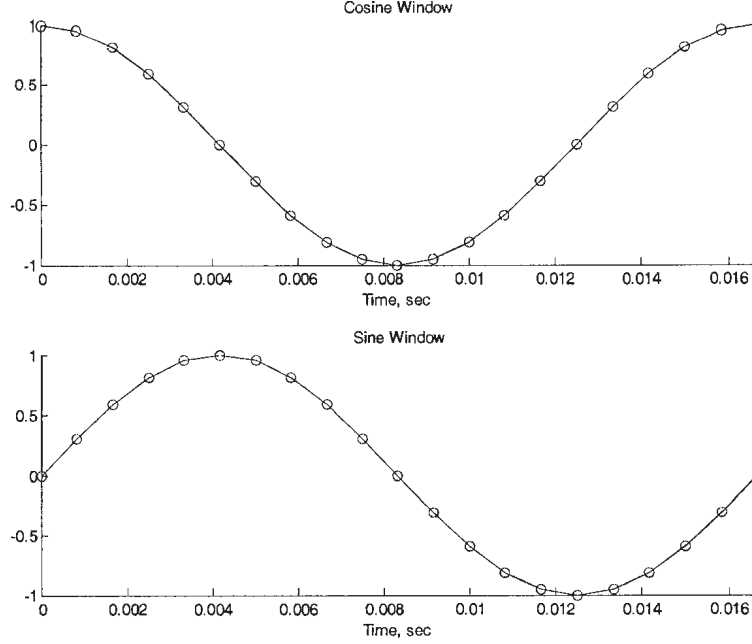


Figure 3.1. Filtering windows used by standard DFT algorithm

The voltage phasors and current phasors by means of equation (3.1) can be used to calculate the impedances seen by the relay based on different fault types. For a three-phase-to-ground fault, the measured impedance by phase A relay is

$$Z_A(n) = \frac{V_A(n)}{I_A(n)} \quad (3.2)$$

For a single-phase-to-ground fault, the measured impedance by phase A relay is [17]

$$Z_A(n) = \frac{V_A(n)}{I_A(n) + k_0 [I_A(n) + I_B(n) + I_C(n)]} \quad (3.3)$$

In equations (3.2) and (3.3),  $k_0 = \frac{z_0 - z_1}{3z_1}$ ;  $z_0$  and  $z_1$  are the zero sequence impedance and positive sequence impedance of the protected transmission line, respectively.  $V_A$  is the fundamental frequency phasor of the phase A voltage measured by

the relay and  $I_A$  is the fundamental frequency phasor of the phase A current measured by the relay. Similarly,  $I_B$  and  $I_C$  are the fundamental frequency phasor of the phase B and phase C current measured by the relay.

### 3.2.2 One-Cycle Cosine Window with a Quarter-Cycle Delay Algorithm

If the Fourier transform pair, as presented in equation (3.1), is further investigated, it is easy to see that the sine window is just a quarter-cycle delay of the cosine window. This implies that it is possible to develop an algorithm that just uses one function window, say the cosine window as showing in Figure 3.2, to extract the phasors [18].

The equations can be written as below

$$\begin{cases} a(n) = \frac{2}{N} \sum_{j=0}^{N-1} M[(n-N+1+j)\Delta t] \cos\left(\frac{2\pi}{N} j\right) \\ b(n) = a\left(n - \frac{N}{4}\right) \end{cases} \quad (3.4)$$

The phasor can be presented as  $a(n) + ja\left(n - \frac{N}{4}\right)$ . Using equations (3.2) and (3.3), the measured impedance for a three-phase-to-ground fault and for a single-phase-to-ground fault can be determined. Effectively, this method calculates the imaginary part of the phasor by using the cosine filter on data shifted by a quarter of a cycle. Hence this algorithm should be considered as requiring one and a quarter cycle data window. This algorithm still requires one cycle delay to obtain  $a(n)$ .

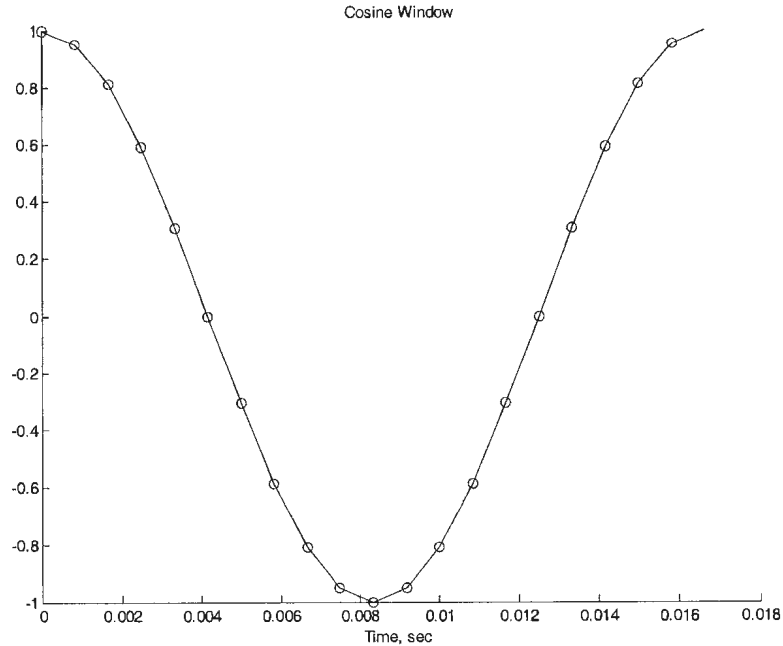


Figure 3.2. Data window used by a quarter-cycle delay algorithm

Frequency responses of these algorithms are shown in Figure 3.3. Although both the sine and cosine window filters reject dc and all harmonics, the cosine filter attenuates the signals above the fundamental frequency better.

Digital relays implement suitable logic for fault detection and faulted phase selection. Usually a trip decision is provided only after 3 or 4 consecutive impedance calculations confirm that the fault is within the zone of protection. Considering all these requirements, a relay which implements a one cycle data window filter may require nearly two cycles for the trip decision. If the cosine window filter is preferred due to its effectiveness in attenuating high frequency components, the overall delay may be more than two cycles. A new filtering algorithm is presented below which uses the cosine window but estimates the phasor information using one more set of voltage and current signal samples.

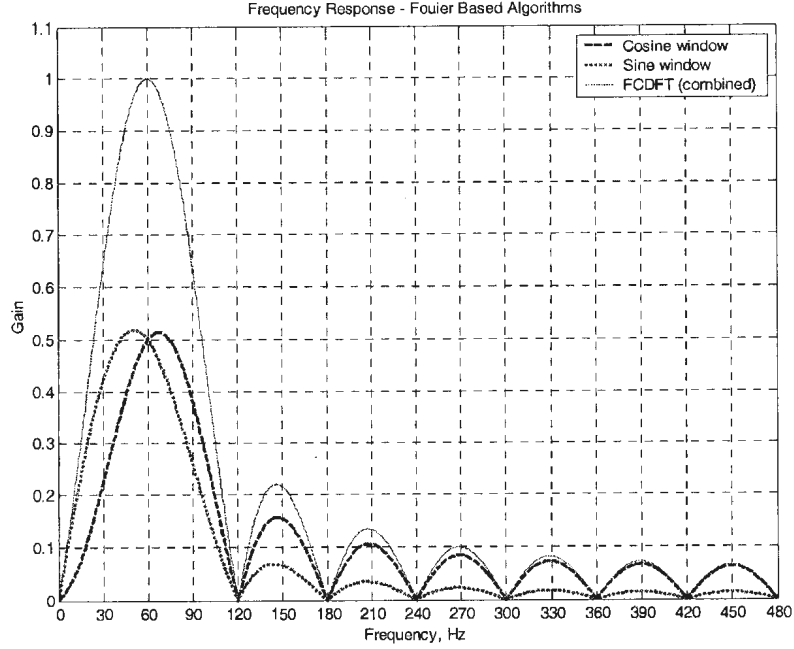


Figure 3.3. Frequency responses of different windows for Fourier associated algorithm

### 3.2.3 One-Cycle Cosine Window with One Sample Delay Algorithm

If the real part of the phasor is expanded in terms of the sampling time, it gives the time domain information of the phasor. It is possible to develop a new algorithm to obtain its frequency-domain representation using this property. This thesis proposes one approach.

Suppose that  $\{a(n), n \in \mathbb{Z}\}$  are a set of points which are sampled from a particular 60 Hz signal,  $v_a(t)$ . Assuming that within two sampling intervals, the magnitude (peak)  $V_A$  and the angle  $\phi$  of the phasor are fixed, equation (3.5) is satisfied.

$$\begin{cases} a(n) = V_A \times \cos[2\pi ft(n) + \phi] \\ a(n-1) = V_A \times \cos[2\pi ft(n-1) + \phi] \end{cases} \quad (3.5)$$

where,  $f = 60\text{Hz}$  is the fundamental frequency.

If  $t(n-1)$  is considered as the reference, it can be set to 0 and then  $t(n)$  can be simplified as the sampling interval, or  $2\pi/N$ .  $V_A$  and  $\tan\phi$  can be calculated from equation (3.5) as

$$\begin{cases} V_A = \frac{\sqrt{a(n)^2 + a(n-1)^2 - 2a(n)a(n-1)\cos(4\pi^2 f/N)}}{\sin(4\pi^2 f/N)} \\ \tan\phi = \frac{a(n) - a(n-1)\cos(4\pi^2 f/N)}{a(n-1)\sin(4\pi^2 f/N)} \end{cases} \quad (3.6)$$

Due to the fact that the period of tangent function is  $\pi$ , with respect to the different signs of  $a(n) - a(n-1)\cos(4\pi^2 f/N)$  and  $a(n-1)$  (since  $\sin(4\pi^2 f/N) > 0$ ), the exact value of  $\phi$  can be determined using the procedure shown in Figure 3.4.

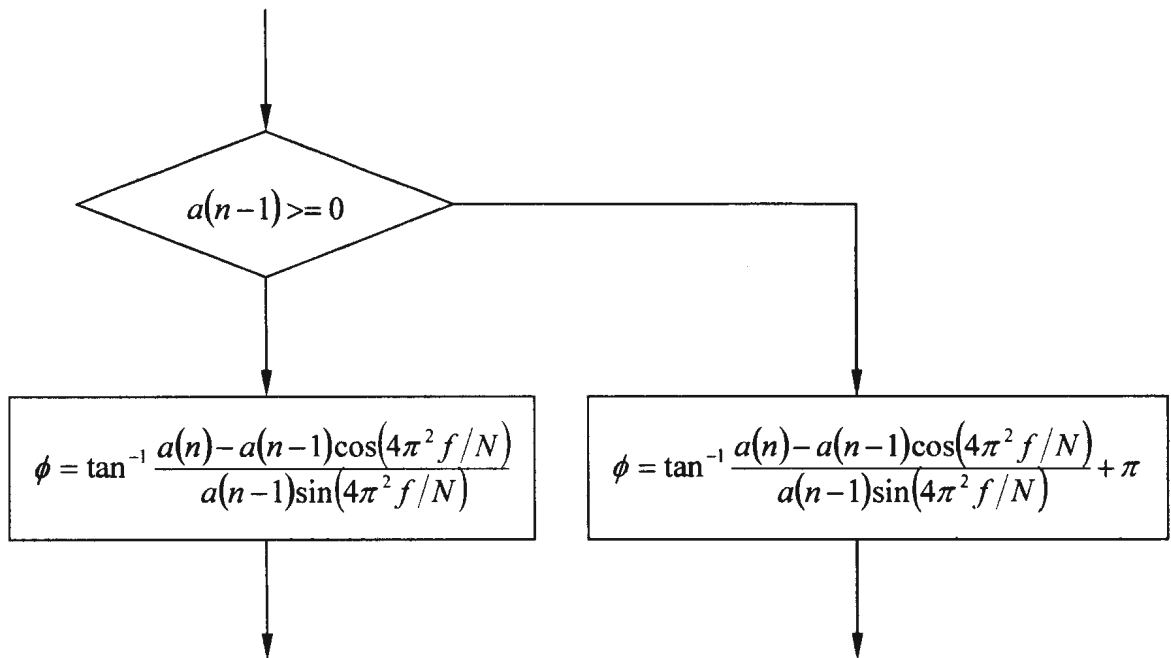


Figure 3.4. Computing scheme to determine phase angle ( $\phi$ )

Using this two-point algorithm, the phasors of voltages and currents can be obtained so that their 60 Hz representation is known. The total filtering window for this algorithm can be considered as one cycle augmented by one more sampled data at the fundamental frequency.

### 3.3 Simulations for Fourier-Based Algorithms

Figure 3.5 shows the block diagram of a digital distance relay. To avoid the possible aliasing problem which may occur within the whole digital signal processing, an analog low pass filter is always implemented between P.T. or C.T. and the digital relay systems. A 3<sup>rd</sup> order anti-aliasing Butterworth low pass filter with cutoff frequency of 540 Hz is implemented in the simulation.

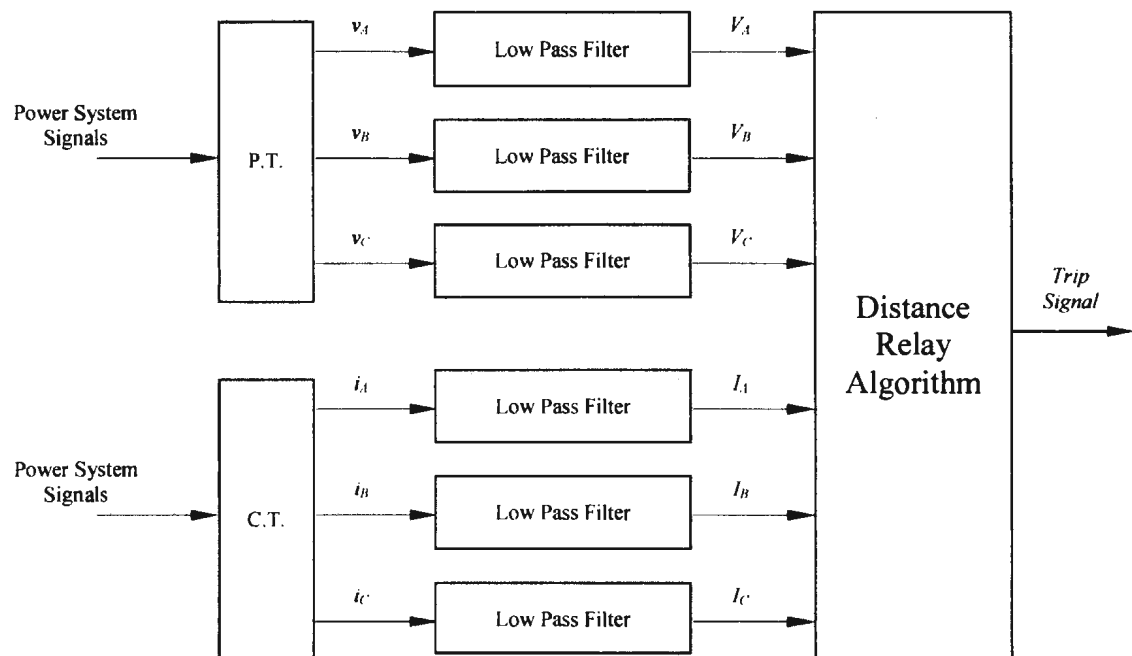


Figure 3.5. Block diagram of a digital distance relay



### 3.3.1 Power System Model

The performances of the three digital filters discussed above are evaluated for the distance protection of a 500kV, 200 miles long transmission line [19]. This transmission line is connected between two equivalent sources. Figure 3.6 shows the single line diagram of the system considered in this study.

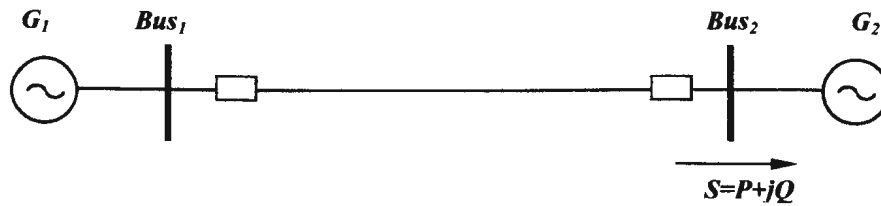


Figure 3.6. The single line diagram of the target system

The pre-fault loading and other system parameters are given in Table 3.1.

Table 3.1 Parameters of the study system - 1

Component	Parameters
$G_1$	Voltage: $V_{G1} = 500\angle 20^\circ \text{ kV}$
	Impedance: $17.177 + j45.529\Omega$ (equivalent system impedance)
$G_2$	Voltage: $V_{G2} = 500\angle 0^\circ \text{ kV}$
	Impedance: $15.310 + j45.925\Omega$ (equivalent system impedance)
Transmission Line	$Z_1 = 4.983 + j117.830\Omega$
	$Y_1 = j1.468 \times 10^{-3} \Omega^{-1}$
	$Z_o = 12.682 + j364.196\Omega$
	$Y_1 = j1.099 \times 10^{-3} \Omega^{-1}$
	$S = 433.63(MW) + j294.52(MVar)$

PSCAD/EMTDC [20] software is used to determine the transient voltage and current signals available at the relay locations. The fundamental frequency, voltage and current phases required by the relay are determined using the three filtering algorithms presented earlier. A sampling frequency of 1200 HZ is considered for this study. The relaying algorithm is simulated in Matlab [21]. The relay operating characteristic is a mho characteristic set to protect 85% of the transmission line in Zone I.

### **3.3.2 Simulation Results**

Consider a single-phase-to-ground (AG) fault with the fault occurring in Phase A occurs. Figure 3.7 and Figure 3.8 show the impedance trajectories determined by the three methods when the AG fault is inside the relay action zone, i.e., the per unit distance of the fault from the relay location  $m = 0.75$ , and the resistance of the fault path  $R_f = 15\Omega$ . Figure 3.9 and Figure 3.10 show the AG fault outside the relay action zone, i.e.,  $m = 0.90$ , and  $R_f = 20\Omega$ . The reasons to select  $R_f = 15\Omega$  and  $R_f = 20\Omega$  in this study are due to the facts that they are typical fault impedance values during overhead transmission line faults [1]. A “mho” circle characteristic, which is used most commonly, is adopted throughout this chapter in order to emphasize the characteristics of the filtering algorithms themselves.

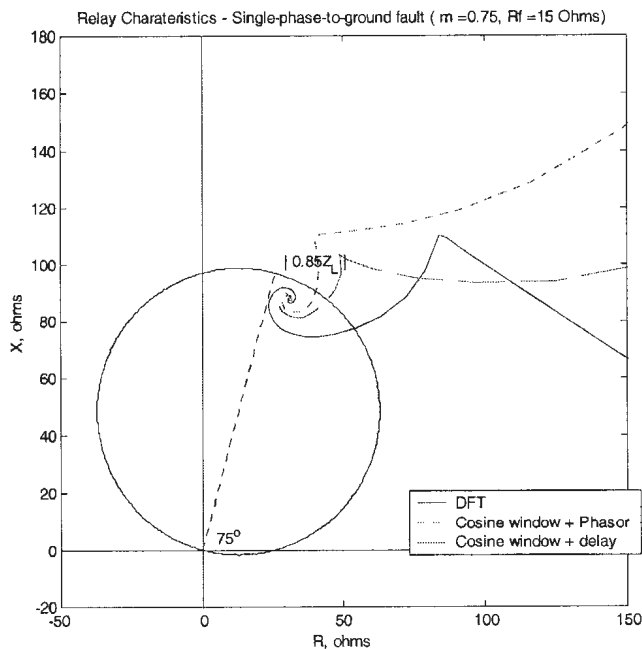


Figure 3.7. Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.75$ )  
 – the Fourier based algorithms

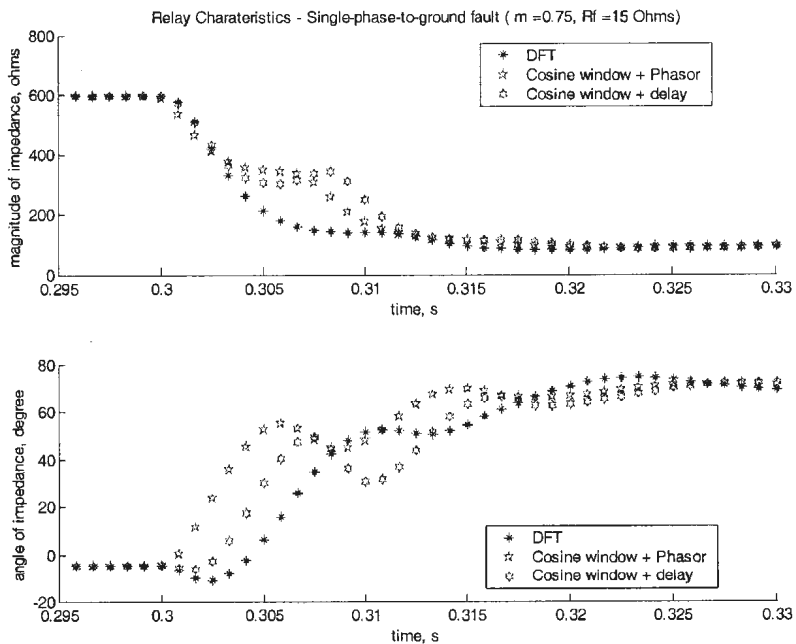


Figure 3.8. Variation of relay impedance for a single-phase-to-ground fault ( $m=0.75$ )  
 – the Fourier based algorithms

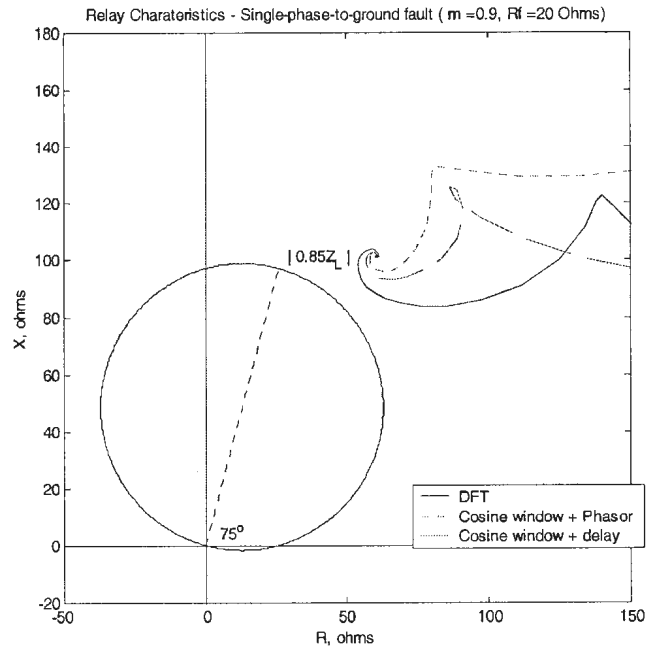


Figure 3.9. Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.90$ )  
– the Fourier based algorithms

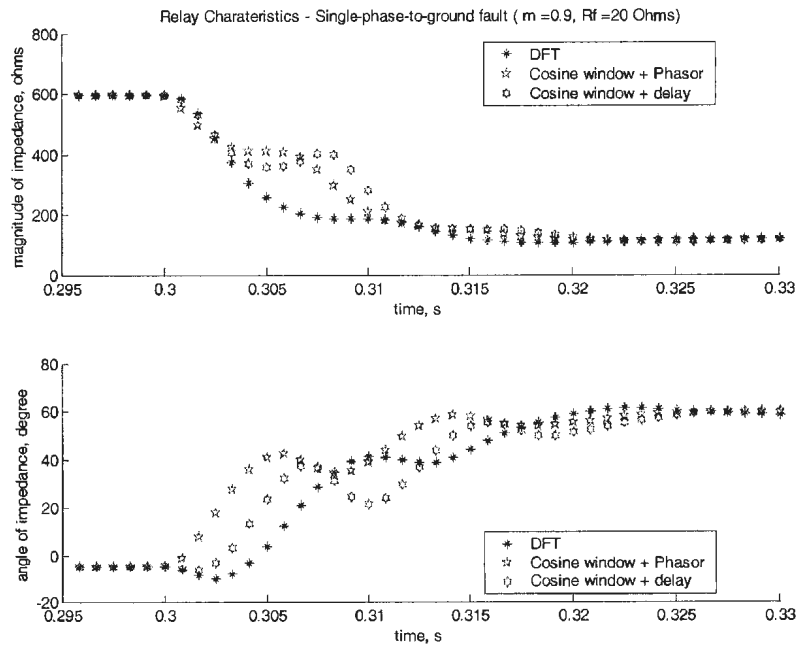


Figure 3.10. Variation of relay impedance for a single-phase-to-ground fault ( $m=0.90$ )  
– the Fourier based algorithms

Figure 3.7 to Figure 3.10 show the impedance trajectories determined by the three methods for different fault locations for phase A to ground faults. From these figures, it is easy to see that all the algorithms issue trip signals when the fault is inside the relay operating characteristics, and will not issue a trip signal when the fault is outside. If the fault resistance keeps increasing, algorithm 1 is likely to cause false operation of the relay. However, this is not a big challenge to algorithm 3 with a cosine window. This study indicates that algorithm 1 is likely to be affected by system operating conditions, while algorithm 3 is the most robust one.

A three-phase-to-ground (3P) fault is also considered. Figure 3.11 and Figure 3.12 show the impedance trajectories determined when a 3P fault is inside the relay action zone, i.e.,  $m = 0.75$ , and  $R_f = 15\Omega$ . Figure 3.13 and Figure 3.14 show a 3P fault outside the relay action zone, i.e.,  $m = 0.90$ , and  $R_f = 20\Omega$ .

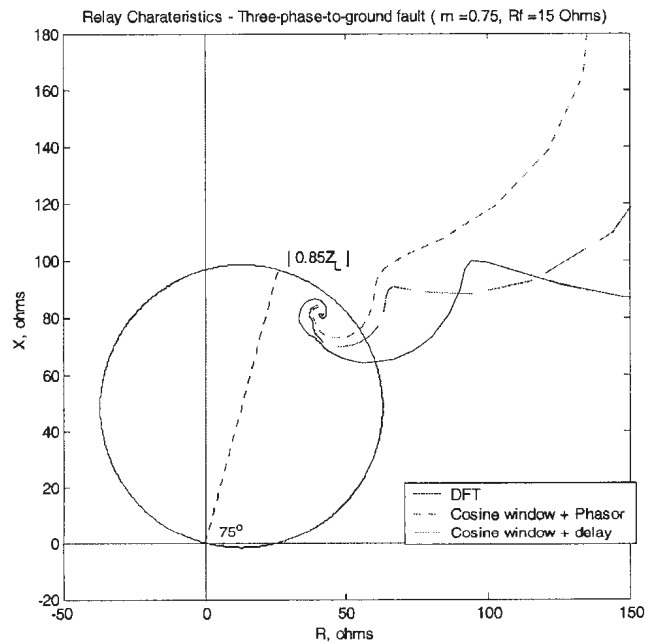


Figure 3.11. Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.75$ )  
– the Fourier based algorithms

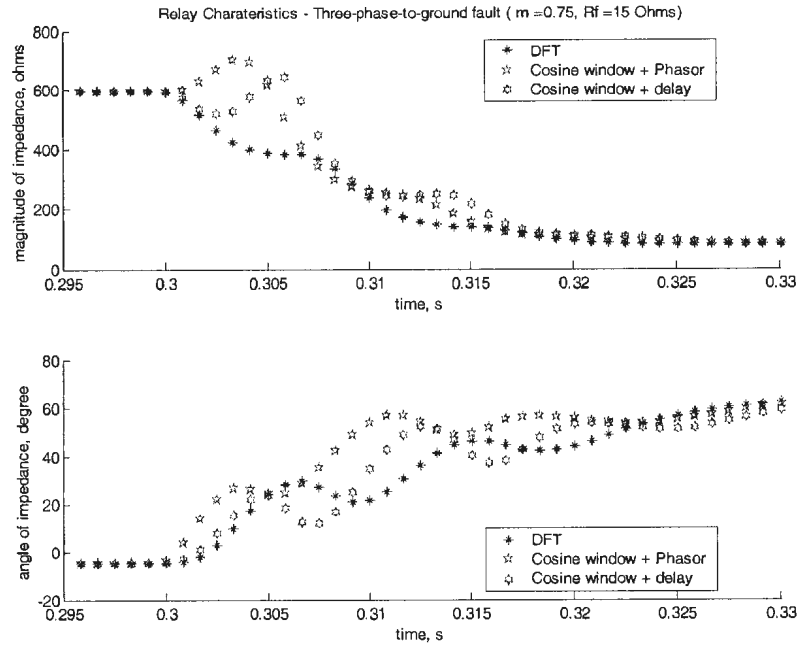


Figure 3.12. Variation of relay impedance for a three-phase-to-ground fault ( $m=0.75$ )  
– the Fourier based algorithms

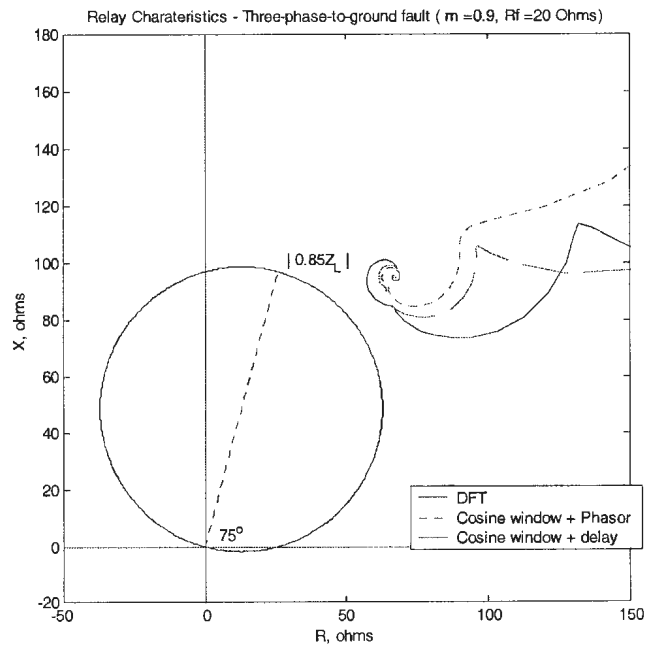


Figure 3.13. Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.90$ )  
– the Fourier based algorithms

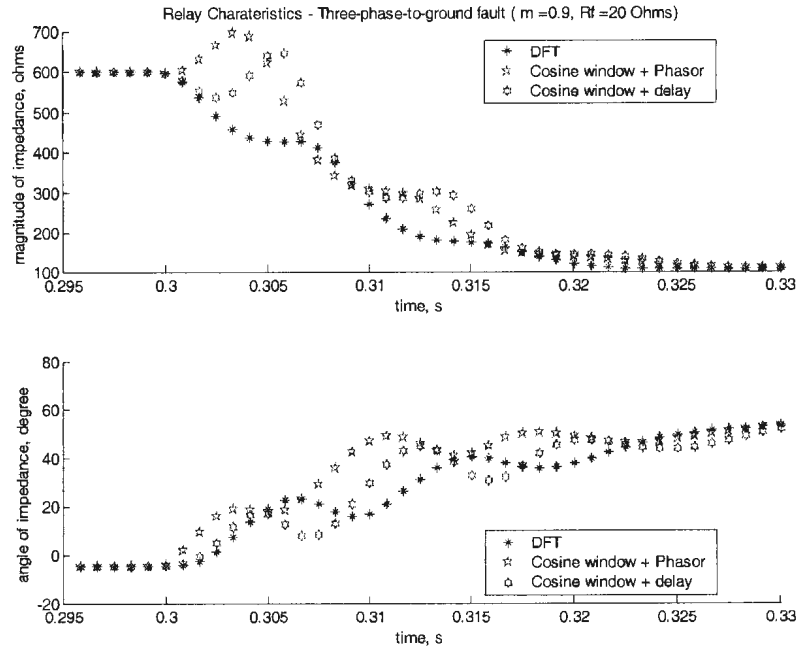


Figure 3.14. Variation of relay impedance for a three-phase-to-ground fault ( $m=0.90$ )  
– the Fourier based algorithms

Figure 3.11 to Figure 3.14 give the same illustrations for three phase to ground fault as AG faults, i.e., all the algorithms issue trip signals when the fault is inside the relay operating characteristics, and will not issue a trip signal when the fault is outside.

### 3.4 Comparison of the Algorithms

In the discussion to follow, algorithm 1 refers to the fully cycle Fourier algorithm using both sine and cosine window; algorithm 2 refers to cosine window which requires  $1\frac{1}{4}$  cycles of data; algorithm 3 refers to the proposed short data window cosine filter.

Algorithm 1 uses two window functions with the length of one cycle data. Therefore, with one cycle delay, the phasors can be ready and then the impedance can be obtained immediately. Considering the speed of convergence, this algorithm is the fastest one

among all the three algorithms. On the other hand, during the period of the fault transient, both cosine window and the sine window will introduce errors, especially when the windows contain both prefault fault samples and post fault samples. The measured impedance can be significantly different from its true value. In short, this algorithm has poor transient response.

Algorithm 2 using cosine window employs data available in one cycle of fundamental frequency. Considering its frequency response, the transient response is expected to be better than algorithm 1, which uses sine and cosine window. However, in this algorithm, additional one-quarter cycle delay is necessary to obtain a full representation of a phasor. Therefore, its equivalent window length expands to 1.25 cycles. This will influence the overall speed of operation of the relay.

Algorithm 3 with the cosine filter also requires one data cycle of data samples. But the phasor information is available using one additional sample. Overall, this algorithm combines the merits of the first two algorithms, i.e., fast convergence and good transient response. This algorithm needs significant calculations to determine the magnitude and angle of the voltage and current signals. With the enhanced speed and capability of the present-day microprocessors, this calculation burden may not appear as a real challenge.

If the fault resistance keeps increasing, algorithm 1 is likely to cause false operation of the relay. However, this is not a big challenge to algorithm 3 with a cosine window. This study indicates that algorithm 1 is likely to be affected by system operating conditions, while algorithm 3 is the most robust one for three phase faults as well.



Generally, the faulted transmission line should be removed within 1 or 2 cycles, which is much shorter than the time that the transient vanishes, i.e., the distance relay issues tripping signal before the apparent impedance has converged. This also suggests that the transient characteristics of different filtering algorithms should be carefully examined. In order to investigate the transient characteristics of the above-mentioned algorithms, the relative error is calculated based on equation (3.7). The errors are calculated corresponding to different sampled data after the fault occurrence. Even though the apparent impedance calculated by the relay after about 25 samples may not be required for any relay decision, it can be used to evaluate the accuracy of the different algorithms.

$$\varepsilon(n) = \left| \frac{Z_m(n) - Z_c}{Z_c} \right| \times 100\% \quad (3.7)$$

where,  $\varepsilon(n)$  is the relative error at sample  $n$ ;

$Z_m(n)$  is the apparent impedance measured at sample  $n$ ;

$Z_c$  is the converged value of the apparent impedance.

As Figure 3.15 and Figure 3.16 indicate, algorithm 3 has less relative error at different samples compared to the other two algorithms, especially when the data window only contains fault data samples.

In general, the modified algorithm, algorithm 3, combines the advantages of Fourier transform based algorithms. This algorithm is expected to have good transient response and also provide fast tripping command during a fault. With the changes in the power

industry, transmission lines carry bulk power over long distances. The performance of distance relays for transmission line protection is critical for the secure operation of the power system. Including new features to enhance the performance of digital relays is one of the challenges facing a relay designer [22][23]. The study presented in this chapter is motivated by these considerations.

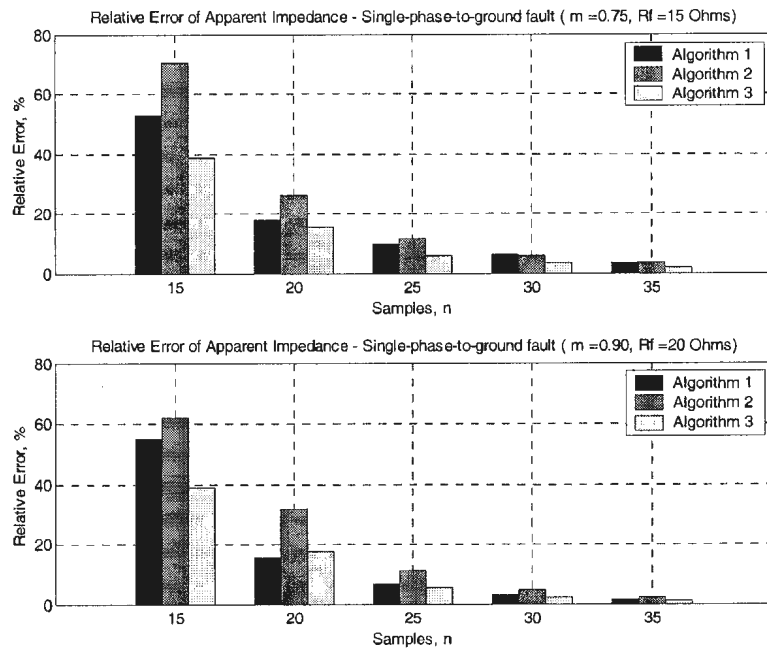


Figure 3.15. Relative errors of the Fourier based filtering algorithms for single-phase-to-ground faults

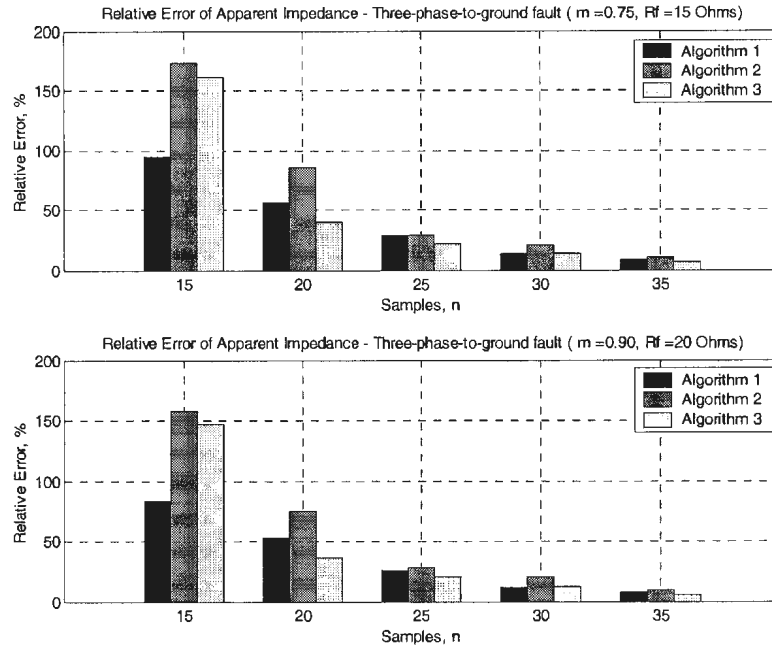


Figure 3.16. Relative errors of the Fourier based filtering algorithms for three-phase-to-ground faults

### 3.5 Wavelet Transform

#### 3.5.1 Introduction

Wavelet-based techniques are new powerful mathematical tools for Digital Signal Processing (DSP), which have become more and more popular since the 1980s [24]. Their applications are widely spread in many areas, such as time series analysis and image processing, due to their excellent transient response characteristics. Application of Wavelet transform for power system relaying has been investigated recently. The following section summarizes some of the research.

In 1999, Shyh-Jier Huang published a paper [25], which proposed a wavelet-based method to supervise power system disturbance. Since wavelet is sensitive to high frequency components when more samples are used, it can successfully detect and

localise different kinds of power system disturbances. In another paper [26], an application of Morlet wavelets to the analysis of high-impedance fault (HIF) generated signals was proposed. It is well known that the values of wavelet transform coefficients change significantly corresponding to different stages of the transient. Therefore, using carefully specified thresholds, the wavelet transform coefficients can indicate when the high-impedance faults occur.

Shi proposed a novel approach [27] to achieve accelerated trip using biorthogonal wavelet transform. The approach works for the case that when a fault occurs and another relay trips, the remaining relay can accelerate trip by detecting two transients. The first transient is easily observed but the second one is quite faint. Due to the excellent transient characteristics of wavelet-based techniques, it can sensitively discover both of the transients so as to accelerate the trip.

In Chi-kong Wong's paper [28], a multi-resolution analysis (MRA) of digital wavelet transform (DWT) associated algorithm is introduced. It implements the MRA characteristics of DWT so as to extract the phasors. In case the phasors are ready, different relay algorithms can be utilised to obtain precise fault location and fast relay response. However, the author uses a cosine wave as a reference to extract the magnitude and phase angle information, which is equivalent to DFT based algorithm.

Another wavelet-based technique is introduced by Meliopoulos in [29]. Instead of using frequency domain model of the power system, the power system is modeled using wavelet coefficients. The method is based on the wavelet series expansion and reconstruction. If more scales are considered, this presentation can give higher resolution

of the system presentation. Essentially, it is an alternative method for transient off-line analysis using wavelet series analysis.

### 3.5.2 *The Conception of Wavelet*

Wavelet, as its name suggests, means "small wave". "Small wave" will grow and decay in a particular limited time period, which is opposite to "big wave", such as the sine or cosine functions which go over  $(-\infty, +\infty)$  and never stop.

Quantitatively, any wavelet, a real-valued function  $\psi(t)$  must satisfy the following two properties [30]:

- (1) The integral of  $\psi(t)$  is zero

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (3.8)$$

- (2) The square of  $\psi(t)$  integrates to unity (which is normalized)

$$\int_{-\infty}^{+\infty} \psi^2(t) dt = 1 \quad (3.9)$$

Equation (3.8) shows that the average of  $\psi(t)$  should be zero, which suggests that the portion above zero is cancelled by the portion below zero, i.e., a wave intuitively. On the other hand, equation (3.9) indicates that the energy of  $\psi(t)$  is finite and can be normalized to unity. Furthermore, equation (3.10) can be derived from equation (3.9) as

$$\int_{-T}^T \psi^2(t) dt < 1 - \varepsilon \quad (3.10)$$

where  $T$  is a finite real value

$\varepsilon$  satisfies  $0 < \varepsilon < 1$ .

Equation (3.10) suggests that most of the energy of  $\psi(t)$  is concentrated on the interval  $[-T, +T]$  if  $\varepsilon$  is very close to zero, i.e.,  $\psi(t)$  grows and decays in a particular limited time period. Hence, the wavelet is well defined through equation (3.8) and (3.9).

### 3.5.3 Continuous Wavelet Transform and its Physical Explanations

The Continuous Wavelet Transform (CWT) for continuous-time domain signal  $x(t)$  can be written as

$$W(\lambda, t) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\lambda}} \psi\left(\frac{u-t}{\lambda}\right) x(u) du \quad (3.11)$$

In equation (3.11), the term  $\frac{1}{\sqrt{\lambda}} \psi\left(\frac{u-t}{\lambda}\right)$  usually is designated as  $\psi_{\lambda, t}(u)$  which is the shifted (by  $t$ ) and scaled (by  $\lambda$ ) version of a particular wavelet function  $\psi(u)$ .

To illustrate the physical meaning behind wavelet transform, which is defined by equation (3.11), Haar wavelet described by equation (3.12) is a good example.

$$\psi^{(H)}(u) = \begin{cases} -1/\sqrt{2}, & -1 < u \leq 0 \\ 1/\sqrt{2}, & 0 < u \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.12)$$

If it is shifted (by  $t$ ) and scaled (by  $\lambda$ ), Haar wavelet can be rewritten as

$$\psi_{\lambda,t}^{(H)}(u) = \frac{1}{\sqrt{\lambda}} \psi_{\lambda,t}^{(H)}\left(\frac{u-t}{\lambda}\right) = \begin{cases} -1/\sqrt{2\lambda}, & t-\lambda < u \leq t \\ 1/\sqrt{2\lambda}, & t < u \leq t+\lambda \\ 0, & \text{otherwise} \end{cases} \quad (3.13)$$

Hence, using equation (3.12) and (3.13), the wavelet transform of signal  $x(t)$  at time  $t_1$  and scale  $\lambda$  can be calculated as

$$\begin{aligned} W^{(H)}(\lambda, t_1) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{\lambda}} \psi^{(H)}\left(\frac{u-t_1}{\lambda}\right) x(u) du \\ &= \int_{t_1-\lambda}^{t_1+\lambda} \frac{1}{\sqrt{\lambda}} \psi^{(H)}\left(\frac{u-t_1}{\lambda}\right) x(u) du \\ &= \frac{1}{\sqrt{2}} \left( \frac{1}{\lambda} \int_{t_1-\lambda}^{t_1} x(u) du - \frac{1}{\lambda} \int_{t_1}^{t_1+\lambda} x(u) du \right) \end{aligned} \quad (3.14)$$

The terms,  $\frac{1}{\lambda} \int_{t_1-\lambda}^{t_1} x(u) du$  and  $\frac{1}{\lambda} \int_{t_1}^{t_1+\lambda} x(u) du$ , are the average values on the interval  $(t_1 - \lambda, t_1]$  and  $(t_1, t_1 + \lambda]$ , respectively. With respect to Haar wavelet, it is easy to see that the wavelet transform of  $x(t)$ , say,  $W^{(H)}(\lambda, t)$ , is the difference between adjacent averages on the interval  $(t - \lambda, t]$  and  $(t, t + \lambda]$ , which is factored by  $1/\sqrt{2}$ .

In a similar manner, the wavelet transforms based on other wavelets are just the difference between adjacent WEIGHTED averages (recall equation (3.8)) with proper scale factor, i.e.,

$$W^{(\psi)}(\lambda, t) = \int_{-\infty}^{\infty} \psi_{\lambda,t}(u) x(u) du \propto D^{(\psi)}(\lambda, t), \quad (3.15)$$

where,  $\psi$  is a particular wavelet;

$D^{(\psi)}(\lambda, t) = \bar{x}^{(\psi)}(t) \Big|_{t-\lambda}^{t+\lambda} - \bar{x}^{(\psi)}(t) \Big|_{t-\lambda}^t$  indicates the weighted average difference.

If the original signal  $x(t)$  satisfies  $\int_{-\infty}^{\infty} x^2(t) dt < \infty$ , i.e., has finite energy over the duration  $(-\infty, \infty)$ , then it can be reconstructed using the following equation.

$$x(t) = \frac{1}{C_\psi} \int_0^\infty \left[ \int_{-\infty}^\infty \frac{1}{\sqrt{\lambda}} \psi\left(\frac{t-u}{\lambda}\right) W(\lambda, t) du \right] \frac{d\lambda}{\lambda^2} \quad (3.16)$$

$C_\psi$  is a constant on the interval  $(0, \infty)$ , which only depends on the choice of wavelet and is determined by so-called admissibility condition

$$C_\psi = \int_0^\infty \frac{|\Psi(f)|^2}{f} df, \quad (3.17)$$

where  $\Psi(f) = \int_{-\infty}^\infty \psi(u) e^{-j2\pi fu} du$  is the Fourier transform of  $\psi(u)$ .

As the above derivation implies, CWT, which introduces another variable  $\lambda$ , besides the time variable  $t$ , provides a new presentation of the original time domain signal. If the scale factor  $\lambda$  is small, short-term changed “difference” will give more significant CWT value; if  $\lambda$  is large, long term changes for the signal will be reflected by CWT value. In other words, different scale factor, which corresponds to different data window length, can present different frequency components of the original signal. The most obvious reason, which supports that the wavelet transform is superior to other mathematical tools, is that CWT still keeps the time characteristics. Using  $\lambda$  and  $t$ , wavelet transform is quite



sensitive to the variation and can precisely locate it, i.e., it has very good transient response.

### **3.5.4 Discrete Wavelet Transform**

Two dimensional CWT presentations, which depend on only one-dimension signal, have a lot of redundancy especially when the scale factors increase. In addition, it is not easy for real-time computer-based applications due to its heavier computing burden.

Discrete wavelet transform (DWT), which plays a role analogous to that of the discrete Fourier transform in spectral analysis, is based on discrete time signals or samples and still preserves the key features of the CWT which describes the signal by different scales and time locations. As the following sections suggest, DWT can be easily implemented in real time by digital computers.

#### **3.5.4.1 DWT**

For ideal sampling, the sampling theorem requires that the minimal sampling rate should be at least twice as the maximum frequency which the signal contains. In other words, for a specified sampling rate, only the components with up to the half of the sampling frequency can be correctly sampled and reconstructed without any aliasing problems.

Therefore, if the original signal  $x(t)$  contains  $N = 2^n$  samples during the time interval  $t$ , the sampling interval or sampling frequency can be expressed as  $T_s = t/2^n$  or  $f_s = 2^n/t$ , respectively. The digital signal analysis on the frequency span  $[0, f_s/2]$  can

be progressed. If every 2 samples of  $x(t)$  are taken, i.e., totally there are  $2^{n-1}$  samples considered, then  $[0, f_s/2^2]$  can be analyzed. Similarly, if downsampling the original signal by  $2^{\lambda-1}$  ( $1 \leq \lambda < n$ , integer), i.e.,  $2^{n-\lambda+1}$  samples are chosen,  $[0, f_s/2^\lambda]$  can be processed.

Suppose  $f_s > 2f_M$ , where  $f_M$  is the maximum frequency of  $x(t)$ , and in every data treatment stage or scale  $\lambda$  when  $1 \leq \lambda < n$ , the components with frequency spanning the interval  $[f_s/2^{\lambda-1}, f_s/2^\lambda]$  or  $[0, f_s/2^{\lambda-1}]$  has been extracted out. Therefore,  $x(t)$  can be analyzed in different scale, which is equivalent to different frequency band. Actually, this data processing progress, or data multiresolution analysis (MRA), is the qualitative description of DWT.

Mathematically, a pyramid algorithm [30], which was introduced by Mallat in 1989, can be implemented to realize the above-described MRA progress.

At the first stage of pyramid algorithm, DWT takes the full-band series of  $x(i)$ ,  $0 \leq i \leq n$ , and transforms it into two series, which are called scaling and wavelet coefficients  $V(i)$ ,  $0 \leq i \leq n/2$  and  $W(i)$ ,  $0 \leq i \leq n/2$ , respectively.

$$\begin{cases} W_{1,t} = \sum_{l=0}^{L-1} h_l x(2t+1-l \bmod n), & t = 0, \dots, n-1 \\ V_{1,t} = \sum_{l=0}^{L-1} g_l x(2t+1-l \bmod n), & t = 0, \dots, n-1 \end{cases} \quad (3.18)$$

In equation (3.18),  $\{h_l : l = 0, \dots, L-1\}$  is a real-valued wavelet filter with length,  $L$ , which is determined by a particular wavelet and acts as a high-pass filter with a frequency band of  $[f_s/2^2, f_s/2]$ .  $\{g_l : l = 0, \dots, L-1\}$  is called a scaling filter which can be calculated by equation (3.19) and acts as a low-pass filter with the frequency band of  $[0, f_s/2^2]$ .

$$g_l = (-1)^{l+1} h_{L-1-l} \quad (3.19)$$

Note that the inverse relationship of equation (3.19) can be written as

$$h_l = (-1)^l g_{L-1-l} \quad (3.20)$$

The wavelet filter or scaling filter must satisfy the following properties

$$\begin{cases} \sum_{l=0}^{L-1} h_l = 0 \\ \sum_{l=0}^{L-1} h_l^2 = 1 \\ \sum_{l=0}^{L-1} h_l h_{l+2n} = \sum_{l=-\infty}^{\infty} h_l h_{l+2n} = 0 \end{cases} \quad (3.21)$$

Or

$$\begin{cases} \sum_{l=0}^{L-1} g_l = 0 \\ \sum_{l=0}^{L-1} g_l^2 = 1 \\ \sum_{l=0}^{L-1} g_l g_{l+2n} = \sum_{l=-\infty}^{\infty} g_l g_{l+2n} = 0 \end{cases} \quad (3.22)$$

And

$$\sum_{l=-\infty}^{\infty} g_l h_{l+2n} = 0 \quad (3.23)$$

In equation (3.21) and (3.22), the first equations are based on the notion of wavelet; while the others are collectively referred to as orthonormality properties. The orthonormality property implies that DWT has no data redundancy and maintains invariable energy before and after the transform. In addition,  $h_l$  and  $g_l$  are defined to zero for  $l < 0$  and  $l \geq L$ , i.e., only  $L$  nonzero values, so as to match the different length of  $x(t)$ .

Equation (3.18) also can be written in matrix form

$$\begin{cases} \mathbf{W}_1 = w_1 \mathbf{X} \\ \mathbf{V}_1 = v_1 \mathbf{X} \end{cases} \quad (3.24)$$

$$\text{where, } w_1 = \begin{bmatrix} h_1 & h_0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & h_l & \dots & h_3 & h_2 \\ h_3 & h_2 & h_1 & h_0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & h_l & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & h_l & \dots & h_3 & h_2 & h_1 & h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & h_l & \dots & h_3 & h_2 & h_1 & h_0 \end{bmatrix}_{N/2 \times N} \quad (3.25)$$

$$v_1 = \begin{bmatrix} g_1 & g_0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & g_l & \dots & g_3 & g_2 \\ g_3 & g_2 & g_1 & g_0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & g_l & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & g_l & \dots & g_3 & g_2 & g_1 & g_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & g_l & \dots & g_3 & g_2 & g_1 & g_0 \end{bmatrix}_{N/2 \times N} \quad (3.26)$$

As can be seen in equation (3.24), the length of  $\mathbf{W}_1$  or  $\mathbf{V}_1$  is  $N/2$ . Therefore, downsampling (or decimation, the operation of extracting every  $N^{\text{th}}$  sample) [31] by 2 is automatically carried out since the  $j^{\text{th}}$  row of matrices  $w_1$  and  $v_1$ , compared to the  $(j-1)^{\text{th}}$  row, is shifted by 2 for  $0 < j \leq N/2$ .

To synthesize  $x(i)$ , the following equation can be implemented

$$\begin{aligned} \mathbf{X} &= D_1 + S_1 \\ &= \begin{bmatrix} w_1^T & v_1^T \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{V}_1 \end{bmatrix} = w_1^T \mathbf{W}_1 + v_1^T \mathbf{V}_1 = w_1^T w_1 \mathbf{X} + v_1^T v_1 \mathbf{X} \end{aligned} \quad (3.27)$$

$$\text{where, } D_1 = w_1^T \mathbf{W}_1 = w_1^T w_1 \mathbf{X} \quad (3.28)$$

$$S_1 = v_1^T \mathbf{V}_1 = v_1^T v_1 \mathbf{X} \quad (3.29)$$

It can be seen that  $w_1^T$  and  $v_1^T$  are  $(N \times N/2)$  matrices, and  $\mathbf{W}_1$  and  $\mathbf{V}_1$  are  $(N/2 \times 1)$  vectors.  $D_1$  and  $S_1$ , therefore, become  $(N \times 1)$  vectors, i.e., equation (3.27) automatically carries out the upsampling (or interpolating, which is basically the reverse process to that of decimation or downsampling) [31] by 2. There is a second way to view the construction of  $D_1$  or  $S_1$ , which can clearly illustrate how the upsampling process is carried out

$$D_1 = w_1 \begin{bmatrix} 0 \\ W_{1,0} \\ \dots \\ 0 \\ W_{1,N/2-1} \end{bmatrix}_{1 \times N} \quad \text{and} \quad S_1 = v_1 \begin{bmatrix} 0 \\ V_{1,0} \\ \dots \\ 0 \\ V_{1,N/2-1} \end{bmatrix}_{1 \times N} \quad (3.30)$$

$$\text{where, } w_l = \begin{bmatrix} h_0 & h_1 & \dots & h_L & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & h_0 & h_1 & \dots & h_L & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_2 & \dots & h_L & 0 & 0 & 0 & 0 & \dots & h_0 & h_1 \\ h_1 & h_2 & \dots & h_L & 0 & 0 & 0 & \dots & 0 & h_0 \end{bmatrix}_{N \times N} \quad (3.31)$$

$$v_l = \begin{bmatrix} g_0 & g_1 & \dots & g_L & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & g_0 & g_1 & \dots & g_L & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_2 & \dots & g_L & 0 & 0 & 0 & 0 & \dots & g_0 & g_1 \\ g_1 & g_2 & \dots & g_L & 0 & 0 & 0 & \dots & 0 & g_0 \end{bmatrix}_{N \times N} \quad (3.32)$$

The analysis and synthesis progress corresponding to downsampling and upsampling by 2, namely scale level 1, can be illustrated using the following flow diagram

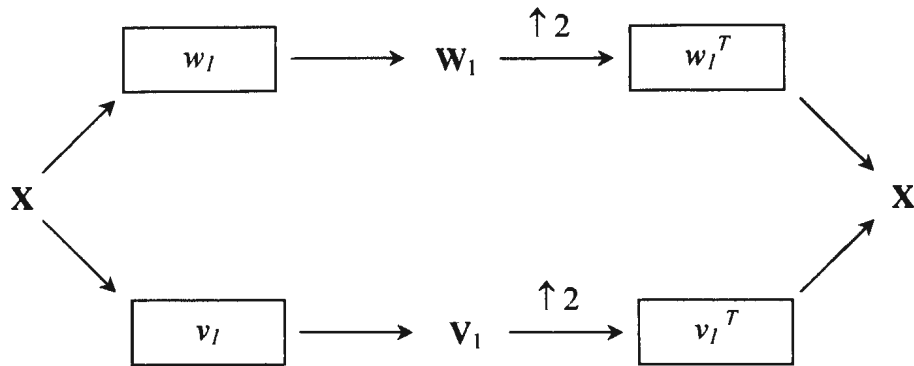


Figure 3.17. Flow diagram at scale level 1

If the following substitutions are implemented, i.e.,  $\mathbf{X} \rightarrow \mathbf{V}_{\lambda-1}$  or  $\mathbf{W}_{\lambda-1}$ ,  $\mathbf{V}_1 \rightarrow \mathbf{W}_\lambda$ ,  $\mathbf{V}_1 \rightarrow \mathbf{V}_\lambda$ ,  $w_{l(N/2 \times N)} \rightarrow w_{\lambda(N/2^{\lambda} \times N/2^{\lambda-1})}$ , and  $v_{l(N/2 \times N)} \rightarrow v_{\lambda(N/2^{\lambda} \times N/2^{\lambda-1})}$ , then Figure 3.17 becomes the flow diagram for the analysis and the synthesis at level  $\lambda$ .

Furthermore, if only the analysis and synthesis for a specified scale level, namely scale level  $j$  which downsamples and upsamples signal  $\mathbf{X}$  by  $2^j$  are of interest, then a

simplified flow diagram, which does not need level-by-level calculations, can be introduced

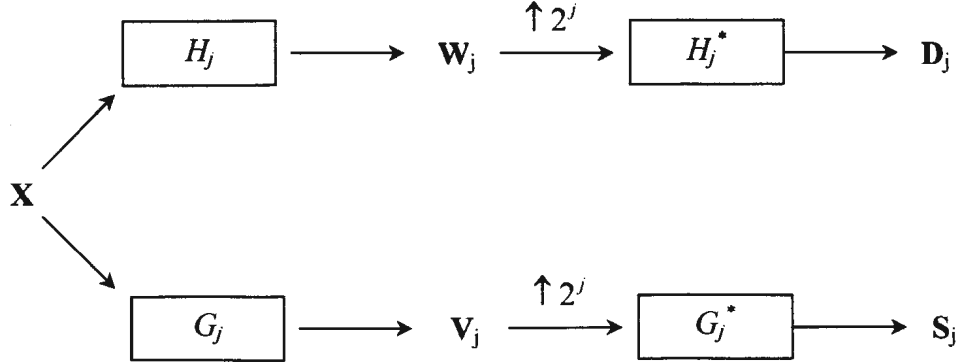


Figure 3.18. Flow diagram at scale level  $j$  (DWT)

In Figure 3.18,  $H_j = w_j v_{j-1} \dots v_2 v_1$ ,  $G_j = v_j v_{j-1} \dots v_2 v_1$ ,  $H_j^* = v_1 v_2 \dots v_{j-1} w_j$  and  $G_j^* = v_1 v_2 \dots v_{j-1} v_j$ . Therefore,

$$\begin{aligned} \mathbf{W}_j &= H_j \mathbf{X} = w_j v_{j-1} \dots v_2 v_1 \mathbf{X} \\ \mathbf{V}_j &= G_j \mathbf{X} = v_j v_{j-1} \dots v_2 v_1 \mathbf{X} \end{aligned} \quad (3.33)$$

$$\begin{aligned} D_j &= H_j^* \mathbf{W}_j^{\uparrow 2^j} = v_1 v_2 \dots v_{j-1} w_j \mathbf{W}_j^{\uparrow 2^j} \\ S_j &= G_j^* \mathbf{V}_j^{\uparrow 2^j} = v_1 v_2 \dots v_{j-1} v_j \mathbf{V}_j^{\uparrow 2^j} \end{aligned} \quad (3.34)$$

Since  $v_1, v_2, \dots, v_j$ , and  $w_1, w_2, \dots, w_j$  are constructed in the same fashion and their only differences are dimensions, the size of matrix  $\mathbf{W}_j$  or  $\mathbf{V}_j$  becomes  $(1 \times N/2^j)$ , using equation (3.33). However,  $H_j^*$  and  $G_j^*$  are  $(N \times N)$  matrices since  $v_1, v_2, \dots, v_j$ , and  $w_1, w_2, \dots, w_j$  are  $(N \times N)$  matrices. Therefore,  $\mathbf{W}_j$  and  $\mathbf{V}_j$  need to be upsampled

by  $2^j$ , i.e., inserting  $2^j$  zeros before each element of them, and then they are denoted as  $\mathbf{W}_j^{\uparrow 2^j}$  and  $\mathbf{V}_j^{\uparrow 2^j}$ , respectively.

The first row of  $v_j$  and  $w_j$  in equation (3.34) are respectively formed as

$$h_0, \underbrace{0, \dots, 0}_{2^{i-1} \text{ zeros}}, h_1, \underbrace{0, \dots, 0}_{2^{i-1} \text{ zeros}}, \dots, h_{l-2}, \underbrace{0, \dots, 0}_{2^{i-1} \text{ zeros}}, h_{l-1} \quad (3.35)$$

Or

$$g_0, \underbrace{0, \dots, 0}_{2^{i-1} \text{ zeros}}, g_1, \underbrace{0, \dots, 0}_{2^{i-1} \text{ zeros}}, \dots, g_{l-2}, \underbrace{0, \dots, 0}_{2^{i-1} \text{ zeros}}, g_{l-1} \quad (3.36)$$

The construction of the other rows is similar to the matrices  $v_1$  and  $w_1$  in equation (3.30).

#### 3.5.4.2 Maximal Overlap Discrete Wavelet Transform (MODWT)

MODWT [30] is another wavelet-based transform. The only difference between MODWT and DWT is that no downsampling progress is carried out when  $\tilde{\mathbf{V}}_1, \tilde{\mathbf{V}}_2, \dots, \tilde{\mathbf{V}}_j$ , and  $\tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2, \dots, \tilde{\mathbf{W}}_j$  are calculated. Therefore, MODWT is not orthonormal and redundant, which is in contrast to DWT.

In order to maintain the energy constant before and after MODWT, the MODWT wavelet filter  $\{\tilde{h}_l\}$  and scaling filter  $\{\tilde{g}_l\}$  have to be the scaled (by  $1/\sqrt{2}$ ) versions of  $\{h_l\}$  or  $\{g_l\}$ , respectively, i.e., equation (3.21) and (3.22) have to be rewritten as



$$\left\{ \begin{array}{l} \sum_{l=0}^{L-1} \tilde{h}_l = 0 \\ \sum_{l=0}^{L-1} \tilde{h}_l^2 = 1/\sqrt{2} \\ \sum_{l=0}^{L-1} \tilde{h}_l \tilde{h}_{l+2n} = \sum_{l=-\infty}^{\infty} \tilde{h}_l \tilde{h}_{l+2n} = 0 \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \sum_{l=0}^{L-1} \tilde{g}_l = 0 \\ \sum_{l=0}^{L-1} \tilde{g}_l^2 = 1/\sqrt{2} \\ \sum_{l=0}^{L-1} \tilde{g}_l \tilde{g}_{l+2n} = \sum_{l=-\infty}^{\infty} \tilde{g}_l \tilde{g}_{l+2n} = 0 \end{array} \right. \quad (3.37)$$

Since no downsampling is done, the dimension of  $\tilde{\mathbf{V}}_1, \tilde{\mathbf{V}}_2, \dots, \tilde{\mathbf{V}}_j$ , and  $\tilde{\mathbf{W}}_1, \tilde{\mathbf{W}}_2, \dots, \tilde{\mathbf{W}}_j$  is always  $(N \times N)$ , and in the reconstruction progress, there is no need to do upsampling. Therefore, Figure 3.18 is redrawn as Figure 3.19

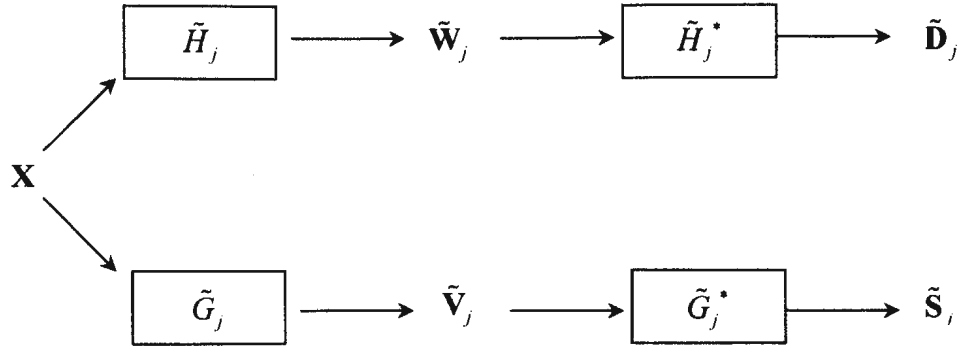


Figure 3.19. Flow diagram at scale level  $j$  (MODWT)

In Figure 3.19,  $\tilde{H}_j = \tilde{w}_j^T \tilde{v}_{j-1}^T \dots \tilde{v}_2^T \tilde{v}_1^T$ ,  $\tilde{G}_j = \tilde{v}_j^T \tilde{v}_{j-1}^T \dots \tilde{v}_2^T \tilde{v}_1^T$ ,  $\tilde{H}_j^* = \tilde{v}_1 \tilde{v}_2 \dots \tilde{v}_{j-1} \tilde{w}_j$  and  $\tilde{G}_j^* = \tilde{v}_1 \tilde{v}_2 \dots \tilde{v}_{j-1} \tilde{v}_j$ . Hence,

$$\begin{aligned} \tilde{\mathbf{W}}_j &= \tilde{H}_j \mathbf{X} = \tilde{w}_j^T \tilde{v}_{j-1}^T \dots \tilde{v}_2^T \tilde{v}_1^T \mathbf{X} \\ \tilde{\mathbf{V}}_j &= \tilde{G}_j \mathbf{X} = \tilde{v}_j^T \tilde{v}_{j-1}^T \dots \tilde{v}_2^T \tilde{v}_1^T \mathbf{X} \end{aligned} \quad (3.38)$$

$$\begin{aligned} \tilde{D}_j &= \tilde{H}_j^* \tilde{\mathbf{W}}_j \\ &= \tilde{v}_1 \tilde{v}_2 \dots \tilde{v}_{j-1} \tilde{w}_j \tilde{\mathbf{W}}_j \\ &= \tilde{v}_1 \tilde{v}_2 \dots \tilde{v}_{j-1} \tilde{w}_j \tilde{w}_j^T \tilde{v}_{j-1}^T \dots \tilde{v}_2^T \tilde{v}_1^T \mathbf{X} \\ &= M_{\tilde{D}_j} \mathbf{X} \end{aligned} \quad (3.39)$$

$$\begin{aligned}
\tilde{\mathbf{S}}_j &= \tilde{\mathbf{G}}_j \tilde{\mathbf{V}}_j = \tilde{v}_1 \tilde{v}_2 \dots \tilde{v}_{j-1} \tilde{v}_j \tilde{\mathbf{V}}_j \\
&= \tilde{v}_1 \tilde{v}_2 \dots \tilde{v}_{j-1} \tilde{v}_j \tilde{v}_j^T \tilde{v}_{j-1}^T \dots \tilde{v}_2^T \tilde{v}_1^T \mathbf{X} \\
&= M_{\tilde{\delta}_j} \mathbf{X}
\end{aligned} \tag{3.40}$$

Similarly, the first row of  $\tilde{v}_j$  and  $\tilde{w}_j$  are respectively formed as

$$\tilde{h}_0, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, \tilde{h}_1, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, \dots, \tilde{h}_{l-2}, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, \tilde{h}_{l-1} \tag{3.41}$$

Or

$$\tilde{g}_0, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, \tilde{g}_1, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, \dots, \tilde{g}_{l-2}, \underbrace{0, \dots, 0}_{2^{j-1}-1 \text{ zeros}}, \tilde{g}_{l-1} \tag{3.42}$$

### 3.5.4.3 DWT versus MODWT

Intuitively, the most significant difference between DWT and MODWT is that MODWT takes every sample when it goes along every scale, while DWT uses every other sample to calculate its next scale. From a computation point of view, DWT is definitely superior to MODWT. However, the waveform of the reconstructed signal by DWT has much more distortion compared to MODWT. This is due to the fact that the former discards some samples, while the latter uses every sample.

In addition, for real-time applications, a moving data window is always implemented. Due to DWT's downsampling and upsampling properties, different samples may be used for DWT calculation, even though the adjacent two data windows contain the same samples except the very first and the last ones. The following figure illustrates this problem.

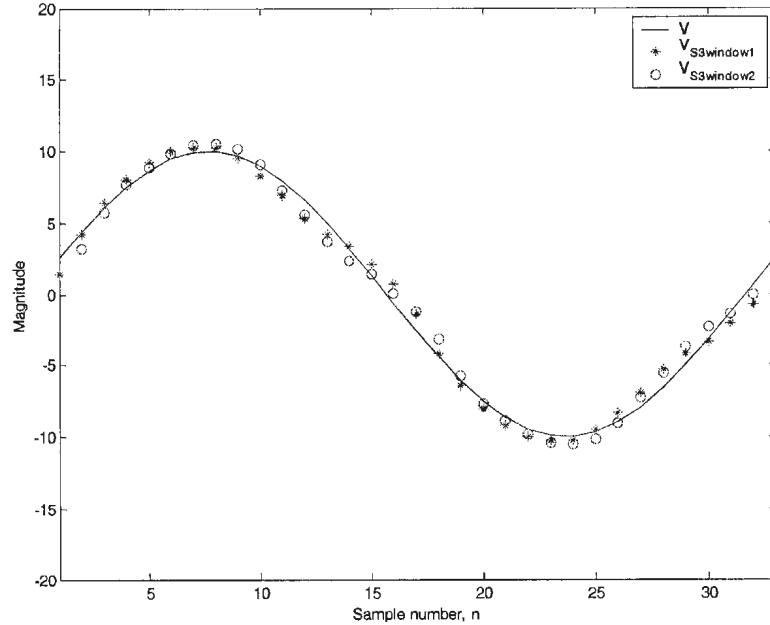


Figure 3.20. Effect of Shifts on DWT

As shown in Figure 3.20, signal  $V$  is a sine wave with 32 samples per cycle. The data window length for DWT contains 32 samples. Using *Daubechies*(6) or, in short  $Db(6)$  wavelet to implement DWT, the smooth  $S_3$ , which are obtained from adjacent windows ( $V_{S3window1}$  and  $V_{S3window2}$ ) are reconstructed. It can be seen that both  $V_{S3window1}$  and  $V_{S3window2}$  have some distortions to the original signal. In addition,  $V_{S3window1}$  and  $V_{S3window2}$  are different, i.e.,  $V_{S3window2}$  is not just a shifted version of  $V_{S3window1}$ .

Figure 3.21 shows another group of reconstructed  $V_{S3window1}$  and  $V_{S3window2}$  using MODWT under the same conditions. As the figure suggests, the wave forms match the original signal quite well and also, the difference between  $V_{S3window1}$  and  $V_{S3window2}$  can be ignored except the first point of  $V_{S3window1}$  and the last point of  $V_{S3window2}$ .

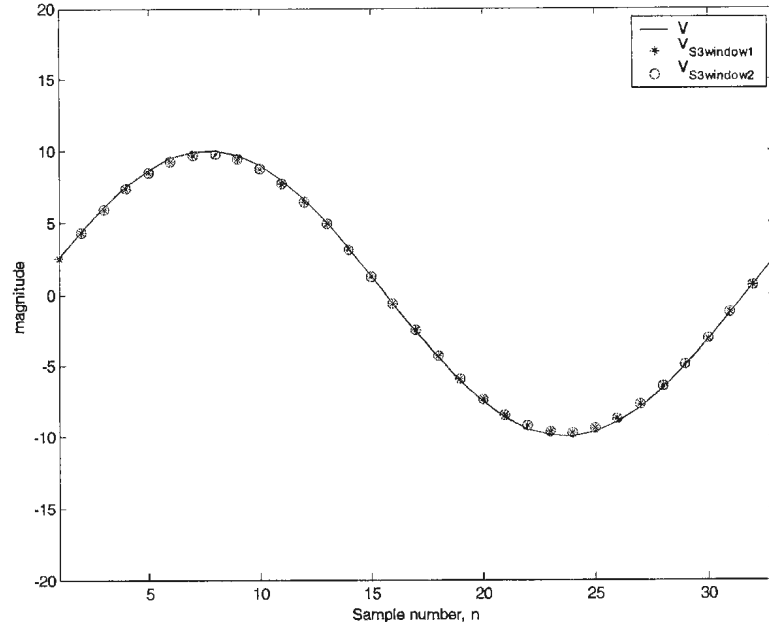


Figure 3.21. Effect of Shifts on MODWT

This analysis shows that MODWT can reconstruct the waveform more precisely than DWT. However, the computation time and resources required by MODWT are doubled compared to that of DWT since no downsampling progress is carried out for the former.

### 3.5.5 Practical Considerations

#### 3.5.5.1 Choice of Wavelet

There are lots of functions which satisfy equation (3.8) and (3.9). In other words, there are many types of wavelet families, such as Haar wavelet [30], Mexican hat wavelet [30], Daubechies (Db) wavelets [30], and so on. Each wavelet family has different characteristics, such as phase response, symmetric properties, transient responses, and so on. In a particular wavelet family, e.g.  $Db(n)$ ,  $n$  is the order. Normally, the order  $n$  determines the number of nonzero element  $L$  for DWT or MODWT based analysis. A

higher  $n$  (or  $L$ ) can provide a better frequency response characteristic. For the same  $L$ , the study in this thesis shows that wavelets, from different wavelet families, will give similar simulation results, especially based on MODWT. However, for a particular application, a careful choice of different wavelet is still necessary so as to match the requirements specific to the problem.

#### 3.5.5.2 Requirements for Sampling

It can be seen that equation (3.18) defines a circular convolution, i.e., for  $l = 2$ ,  $t = 0$ ,  $x(2t+1-l \bmod n) = x(-1) = x(n-1)$ . Therefore, a periodic extension is automatically implemented. It implies that the data window, which defines the samples to be processed, should contain integer cycles of samples for the signal  $x(n)$ . In the case that  $x(n)$  has significant transient components, the assumption may not be correct, which can cause the boundary problems as shown in Section 3.5.5.2.

In addition, unlike Fourier-based data window techniques, which assume that the magnitude and angle of time series signal are constant and provide relevant information in frequency domain, wavelet-based techniques give the time domain information under a particular scale, i.e., they are time-frequency presentations. In other words, the outputs are essentially “waveforms”, which imply that a higher sampling rate can give a more accurate reconstructed result. However, the sampling rate has to be compromised considering the available sampling devices and economic constraints.

### 3.5.5.3 Boundary Conditions

Wavelet-based techniques, which make use of circular filtering, are implemented based upon the windowed signal. By definition, the techniques treat the signal as if it were a portion of a periodic sequence. Its period is the same as the length of the windowed data. Therefore, at the boundary of the data window, some error will be introduced if the original signal is not periodic or it varies rapidly. However, there is virtually no difference in the Wavelet-based analysis except at the very beginning and end of windowed data. To cope with the possible negative effects caused by the circularity, the data processing, which is followed by wavelet-based analysis, can be carried out using the central portion of the windowed data.

### 3.5.6 Summary

Wavelet-based techniques are powerful digital processing tools due to their excellent transient response. Using the so-called “pyramid algorithm”, each scale level of the original signal can be easily analyzed. In real-time applications, the boundary condition of the associated problem should be carefully considered.

## 3.6 Phasor extraction by wavelet transform

### 3.6.1 Introduction

Under normal operating conditions, the steady state of power system uses 60 Hz frequency (or 50 Hz). In other words, the power system is modeled using 60 (or 50) Hz parameters. However, in some special cases, such as power system faults, transients and harmonics will occur in the power system. These unwanted or “bad” components usually

cannot be ignored and the fundamental 60 (or 50) Hz voltages and currents phasors are always hidden among them. On the other hand, power system distance relay needs these phasors to calculate the impedance so as to determine if the fault location is within its protection area or not. Therefore, to have a good understanding about the system operating, or faulted, conditions, it is always extremely important and necessary to extract the phasors correctly and rapidly.

Almost all the contemporary digital relaying applications rely on DFT, such as FCDFE or HCDFT, to extract those phasors. In the following sections, a wavelet-based approach is proposed and compared with the Fourier-based algorithms. As will be shown, the new approach, which takes the advantages of the good transient response properties of wavelet transform, functions correctly and converges rapidly.

### ***3.6.2 Algorithm description***

The voltages and currents measured by relays during a fault suffer from transients, and transmission line distance relays only can properly function using 60 (or 50) Hz fundamentals. As illustrated in Section 3.5.4.3, wavelet-based filtering techniques can be implemented for this purpose, and MODWT is more suitable to extract the phasors. A real-time MODWT algorithm is introduced below.

#### **3.6.2.1 Selection of the Wavelet**

Due to economical considerations, the sampling rate of 16 or 20 samples per cycle is quite popular for the current commercial distance relays. However, wavelet analysis needs higher sampling rate to obtain good output waveform. It should be reasonable to

assume that 32 samples per cycle will not cause significant additional cost to meet real-time processing requirements.

Assume that the sampling rate is fixed on 32 samples per cycle, i.e., 1920 Hz, then the data window length is 32 samples. The maximal nonzero element  $L$  (even integer) and the maximal scale  $\lambda$  of the wavelet can be determined according to equation (3.41).

Suppose  $L = 8$ , at each scale, the first rows of  $\tilde{v}_\lambda$  are

$$\begin{aligned}
 \lambda = 1, & \quad \left[ \tilde{h}_0, \tilde{h}_1, \dots, \tilde{h}_6, \tilde{h}_7, \underbrace{0, \dots, 0}_{24 \text{ zeros}} \right]_{1 \times 32} \\
 \lambda = 2, & \quad \left[ \tilde{h}_0, 0, \tilde{h}_1, 0, \dots, \tilde{h}_6, 0, \tilde{h}_7, \underbrace{0, \dots, 0}_{17 \text{ zeros}} \right]_{1 \times 32} \\
 \lambda = 3, & \quad \left[ \tilde{h}_0, 0, 0, 0, \tilde{h}_1, 0, 0, 0, \dots, \tilde{h}_6, 0, 0, 0, \tilde{h}_7, 0, 0, 0 \right]_{1 \times 32}
 \end{aligned} \tag{3.43}$$

Equation (3.43) shows that at scale 3, i.e.,  $\lambda = 3$ , the first row of  $\tilde{v}_3$  is fully used by the eight nonzero elements and relevant zeros.

If  $L$  is chosen as 6, the first rows of  $\tilde{v}_\lambda$  at each scale are

$$\begin{aligned}
 \lambda = 1, & \quad \left[ \tilde{h}_0, \tilde{h}_1, \dots, \tilde{h}_4, \tilde{h}_5, \underbrace{0, \dots, 0}_{26 \text{ zeros}} \right]_{1 \times 32} \\
 \lambda = 2, & \quad \left[ \tilde{h}_0, 0, \tilde{h}_1, 0, \dots, \tilde{h}_4, 0, \tilde{h}_5, \underbrace{0, \dots, 0}_{21 \text{ zeros}} \right]_{1 \times 32} \\
 \lambda = 3, & \quad \left[ \tilde{h}_0, 0, 0, 0, \tilde{h}_1, 0, 0, 0, \dots, \tilde{h}_4, 0, 0, 0, \tilde{h}_5, \underbrace{0, \dots, 0}_{11 \text{ zeros}} \right]_{1 \times 32}
 \end{aligned} \tag{3.44}$$



In equation (3.44), the maximal scale MODWT can reach is still 3. it is easy to see that  $6 \times 2^3 = 48$  samples per cycle, which is not the assumed sampling rate, are required if  $\lambda = 4$ .

Therefore, the relation between nonzero numbers  $L$  of a particular wavelet, say  $Db(n)$ , and the maximal scale  $\lambda_{\max}$  the DWT or MODWT can reach - under 32 samples per cycle - is

$$\begin{aligned}
 & \dots \\
 & L = 4, \quad \lambda_{\max} = 4; \\
 & L = 6, \quad \lambda_{\max} = 3; \\
 & L = 8, \quad \lambda_{\max} = 3; \\
 & L = 10, \quad \lambda_{\max} = 2; \\
 & \dots
 \end{aligned} \tag{3.45}$$

Since a higher  $L$  produces better frequency response, the length  $L = 8$  under  $\lambda_{\max} = 3$  is used for MODWT analysis. Since MODWT is not sensitive for different wavelets,  $Db(8)$  is chosen.

### 3.6.2.2 Frequency Response Characteristics

As shown in Section 3.6.2.1, for  $Db(8)$  and 32 samples per cycle, the maximal scale MODWT can reach is 3. Using the methods illustrated in Section 3.5.4.2, the transfer matrices  $M_{\hat{D}_j}$  and  $M_{\hat{S}_j}$  for  $j = 1, 2, 3$  can be computed.

The parameters  $g_{Db}(8) = \{g_0, g_1, \dots, g_7\}$  of the scaling filter of  $Db(8)$  are constant and easy to be obtained from literatures, such as [30], which are

$$\begin{aligned}
g_0 &= 0.2303778133074431, \\
g_1 &= 0.7148465705484058, \\
g_2 &= 0.6308807679358788, \\
g_3 &= -0.0279837694166834, \\
g_4 &= -0.1870348117179132, \\
g_5 &= 0.0308413818353661, \\
g_6 &= 0.0328830116666778, \\
g_7 &= -0.0105974017850021.
\end{aligned}
\tag{3.46}$$

Therefore

$$v_1 = \begin{bmatrix} g_0 & g_1 & \dots & g_7 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & g_0 & g_1 & \dots & g_7 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_2 & \dots & g_7 & 0 & 0 & 0 & 0 & \dots & g_0 & g_1 \\ g_1 & g_2 & \dots & g_7 & 0 & 0 & 0 & \dots & 0 & g_0 \end{bmatrix}_{32 \times 32}
\tag{3.47}$$

$$v_2 = \begin{bmatrix} g_0 & 0 & g_1 & \dots & 0 & g_7 & 0 & \dots & 0 & 0 \\ 0 & g_0 & 0 & g_1 & \dots & 0 & g_7 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_1 & \dots & 0 & g_7 & 0 & 0 & 0 & \dots & g_0 & 0 \\ 0 & g_1 & \dots & 0 & g_7 & 0 & 0 & \dots & 0 & g_0 \end{bmatrix}_{32 \times 32}
\tag{3.48}$$

$$v_3 = \begin{bmatrix} g_0 & 0 & 0 & 0 & g_1 & \dots & g_7 & 0 & 0 & 0 \\ 0 & g_0 & 0 & 0 & 0 & g_1 & \dots & g_7 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & g_1 & \dots & g_7 & 0 & 0 & 0 & g_0 & 0 \\ 0 & 0 & 0 & g_1 & \dots & g_7 & 0 & 0 & 0 & g_0 \end{bmatrix}_{32 \times 32}
\tag{3.49}$$

Using equation (3.20), the relevant parameters  $h_{j,b}(8) = \{h_0, h_1, \dots, h_7\}$  can be calculated as

$$\begin{aligned}
h_0 &= -0.0105974017850021, \\
h_1 &= -0.0328830116666778, \\
h_2 &= 0.0308413818353661, \\
h_3 &= 0.1870348117179132, \\
h_4 &= -0.0279837694166834, \\
h_5 &= -0.6308807679358788, \\
h_6 &= 0.7148465705484058, \\
h_7 &= -0.2303778133074431.
\end{aligned} \tag{3.50}$$

Similarly,

$$w_1 = \begin{bmatrix} h_0 & h_1 & \dots & h_7 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & h_0 & h_1 & \dots & h_7 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_2 & \dots & h_7 & 0 & 0 & 0 & 0 & \dots & h_0 & h_1 \\ h_1 & h_2 & \dots & h_7 & 0 & 0 & 0 & \dots & 0 & h_0 \end{bmatrix}_{32 \times 32} \tag{3.51}$$

$$w_2 = \begin{bmatrix} h_0 & 0 & h_1 & \dots & 0 & h_7 & 0 & \dots & 0 & 0 \\ 0 & h_0 & 0 & h_1 & \dots & 0 & h_7 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_1 & \dots & 0 & h_7 & 0 & 0 & 0 & \dots & h_0 & 0 \\ 0 & h_1 & \dots & 0 & h_7 & 0 & 0 & \dots & 0 & h_0 \end{bmatrix}_{32 \times 32} \tag{3.52}$$

$$w_3 = \begin{bmatrix} h_0 & 0 & 0 & 0 & h_1 & \dots & h_7 & 0 & 0 & 0 \\ 0 & h_0 & 0 & 0 & 0 & h_1 & \dots & h_7 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & h_1 & \dots & h_7 & 0 & 0 & 0 & h_0 & 0 \\ 0 & 0 & 0 & h_1 & \dots & h_7 & 0 & 0 & 0 & h_0 \end{bmatrix}_{32 \times 32} \tag{3.53}$$

Hence,  $M_{\tilde{D}_j}$  and  $M_{\tilde{S}_j}$  can be calculated using equation (3.39), i.e.,

$$\begin{aligned}
M_{\tilde{D}_j} &= \tilde{w}_j \tilde{w}_j^T \\
M_{\tilde{S}_j} &= \tilde{v}_j \tilde{v}_j^T
\end{aligned} \tag{3.54}$$

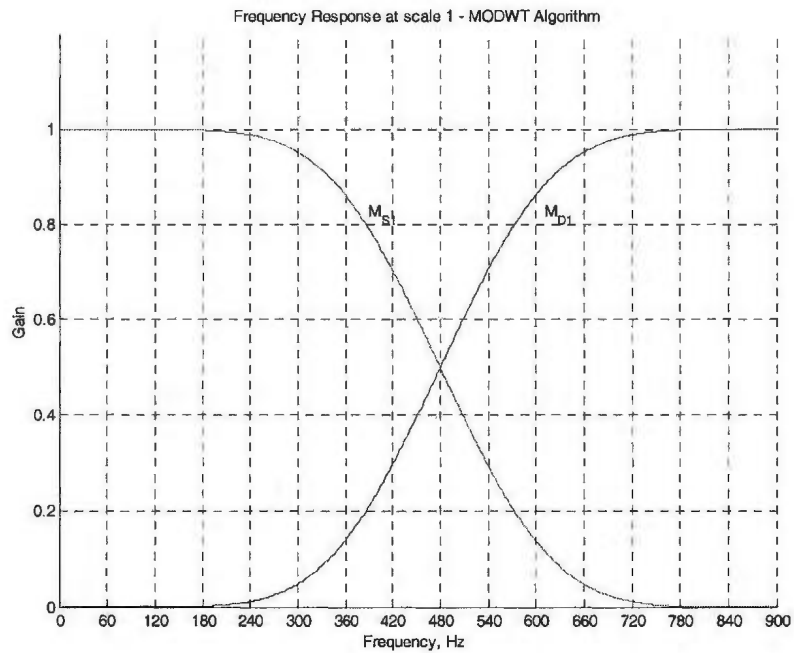


Figure 3.22. Frequency Response at Scale 1

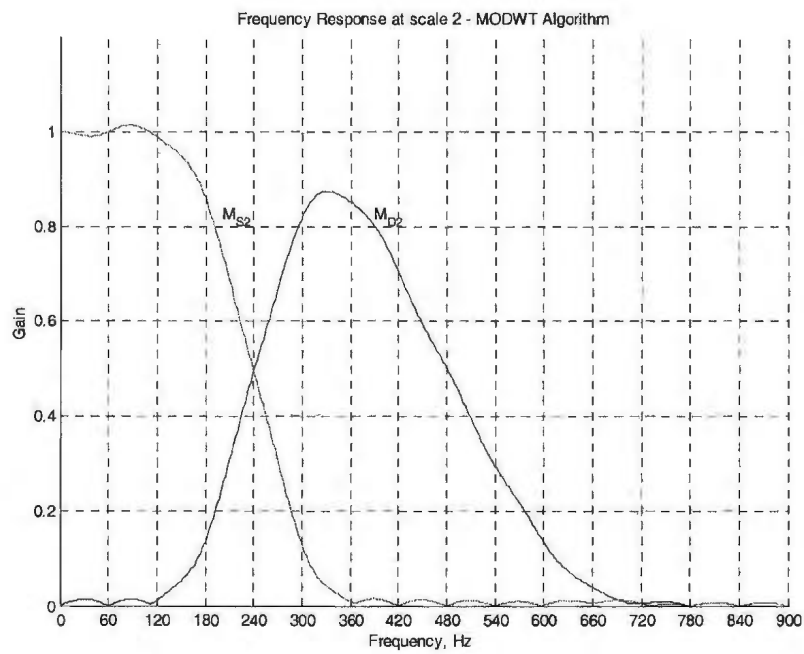


Figure 3.23. Frequency Response at Scale 2

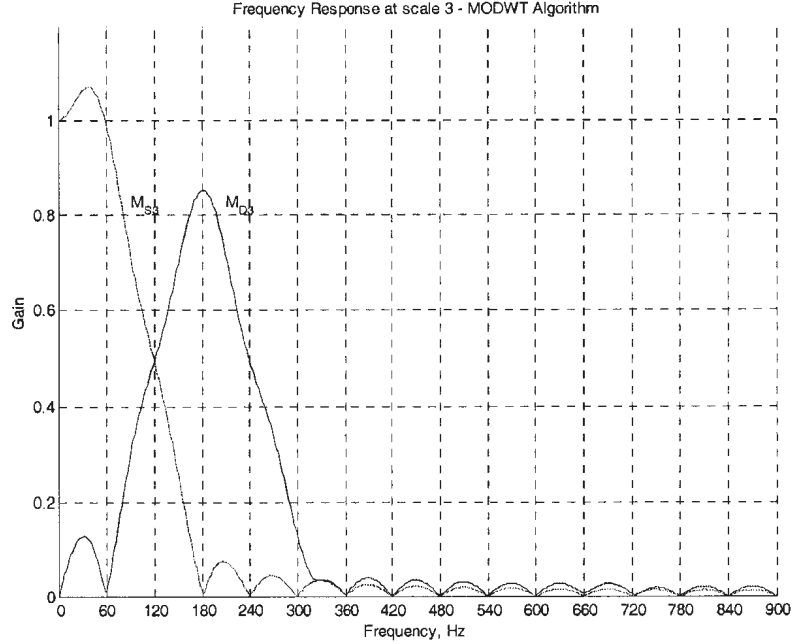


Figure 3.24. Frequency Response at Scale 3

$$\begin{aligned}
 M_{\tilde{D}_2} &= \tilde{v}_1 \tilde{w}_2 \tilde{w}_2^T \tilde{v}_1^T \\
 M_{\tilde{S}_1} &= \tilde{v}_1 \tilde{v}_2 \tilde{v}_2^T \tilde{v}_1^T
 \end{aligned} \tag{3.55}$$

$$\begin{aligned}
 M_{\tilde{D}_3} &= \tilde{v}_1 \tilde{v}_2 \tilde{w}_3 \tilde{w}_3^T \tilde{v}_2^T \tilde{v}_1^T \\
 M_{\tilde{S}_3} &= \tilde{v}_1 \tilde{v}_2 \tilde{v}_3 \tilde{v}_3^T \tilde{v}_2^T \tilde{v}_1^T
 \end{aligned} \tag{3.56}$$

$M_{\tilde{D}_j}$  and  $M_{\tilde{S}_j}$  work as filters corresponding to different scales in equation (3.39).

Therefore, the matrices can be investigated from their frequency response point of view. Since the matrices function as circular filters, each row in the matrix basically has identical frequency response. Due to the boundary issues, which will be shown in 3.6.2.5, only the 23<sup>rd</sup> row of each matrix will be examined.

Using Matlab, the frequency response of the matrices for  $j=1, 2,$  and  $3$  are shown in Figure 3.22, Figure 3.23, and Figure 3.24, respectively. As the figures imply,  $M_{\tilde{S}_3}$  should be the best choice since it works as a low pass filter which has zero gain for the 3<sup>rd</sup> and

above harmonics. However, the gain for DC signal is unity and for the 2<sup>nd</sup> harmonic is around 0.5, which is undesired for the equivalent filter of  $M_{\bar{s}_3}$ . This phenomenon indicates that some data pre-processing is necessary, which will be mentioned in Section 3.6.2.3. Note that the frequency response only can illustrate the steady-state characteristics of a particular filter. In other words, it does not show any transient characteristics.

If  $M_{\bar{D}_1}$ ,  $M_{\bar{D}_2}$ ,  $M_{\bar{D}_3}$  and  $M_{\bar{s}_3}$  are put together, they show the overall frequency analysis using different scales, which can be seen in Figure 3.25. As the figure shows, there are some significant overlaps among different scales. This can be improved by increasing the nonzero element  $L$  of a particular wavelet. However, the sampling rate should be increased at the same time, which will demand higher processing ability of the hardware and also increase the cost.

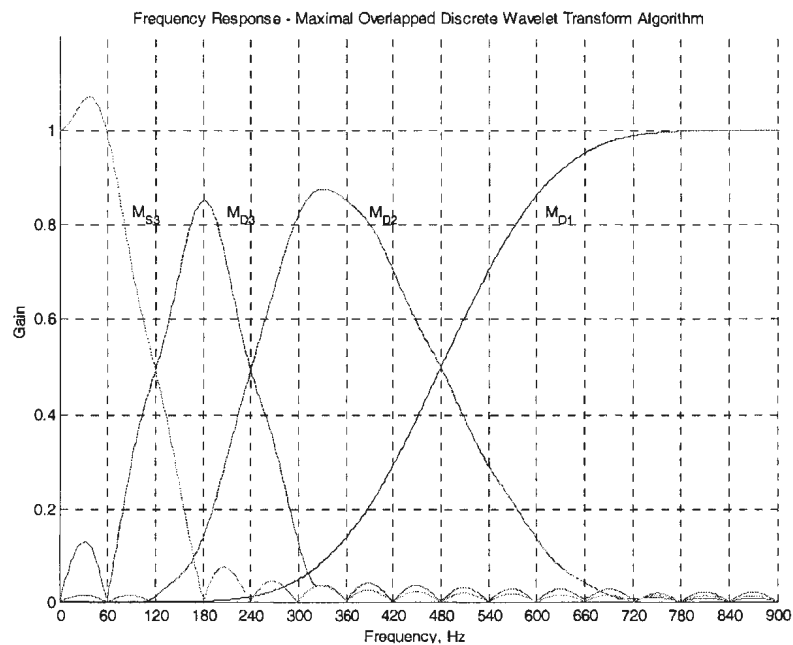


Figure 3.25. The Overall Frequency Response of the MODWT pyramid algorithm using Db(8)

### 3.6.2.3 Transient Characteristics

To investigate the transient characteristics of the MODWT algorithm, the following time series signal is used as an example.

$$x(t) = \begin{cases} 5 \sin(120\pi t), & t \leq 0.0167 \\ 10 \sin(120\pi t), & t > 0.0167 \end{cases}$$

To monitor the signal  $x(t)$  in real time, a moving window can be used. As shown in Figure 3.26, before  $t = 0.0167$  sec., the magnitude of  $x(t)$  is 5; after that moment, it changes to 10. Even though  $x(t)$  is made up of pure sinusoidal signals, it still contains significant harmonics, especially at  $t = 0.0167$  sec. when sharp changes occur.

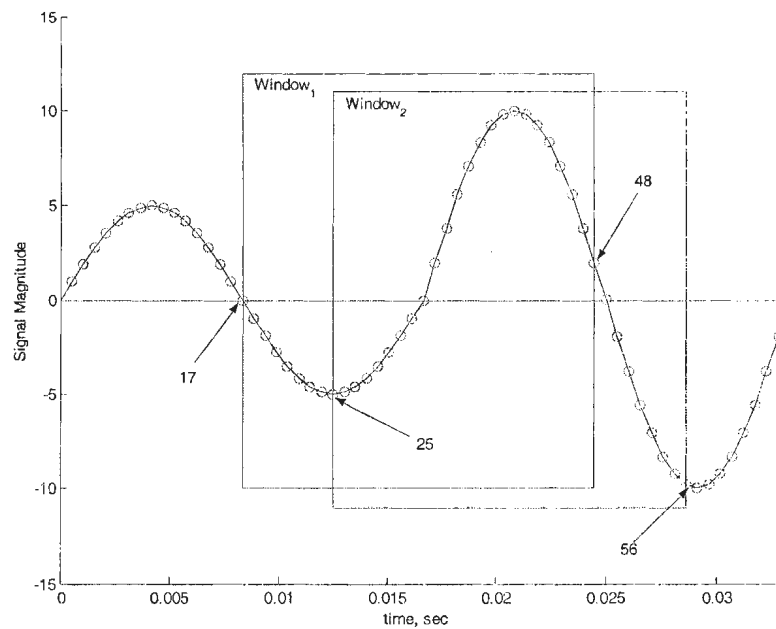


Figure 3.26. The waveform of  $x(t)$

When the data window moves to  $t = 0.0245$  sec., i.e., window<sub>1</sub>, which contains sample 17 to 48 in Figure 3.26, half of the data can be described using  $5 \sin(120\pi t)$  and

the other half follows  $10\sin(120\pi t)$ . Using the algorithm introduced in the previous sections, the smooth at scale 3 under  $Db(8)$  can be expressed as

$$x_{s3} = M_{\tilde{s}_3} \cdot x(n)_{17}^{48} \quad (3.57)$$

where  $x(n)_{17}^{48}$  denotes 32 discrete samples of  $x(t)$  in window<sub>1</sub>

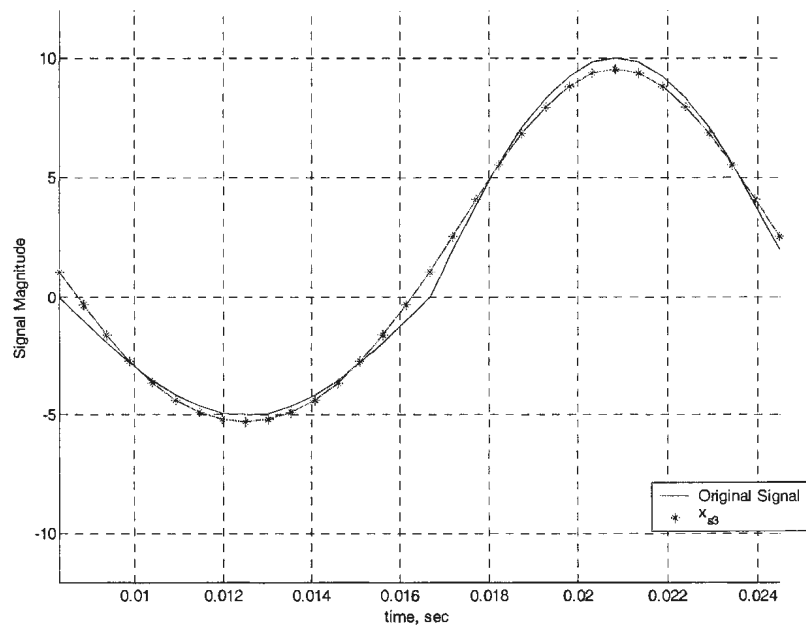


Figure 3.27. The waveform of  $x(t)$  and  $x_{s3}$  in window<sub>1</sub>

Figure 3.27 shows  $x_{s3}$ , together with the original  $x(t)$ . As can be seen, the reconstructed waveform  $x_{s3}$  is very close to  $x(t)$ .

Similarly, Figure 3.28 shows  $x_{s3}$  and  $x(t)$  in windows<sub>2</sub> using sample 25 to 56, windows<sub>2</sub> only contains a quarter signals of  $5\sin(120\pi t)$  at  $t = 0.0286$  sec. It also can be seen that  $x_{s3}$  can estimate  $x(t)$  quite well, especially for the middle portions.



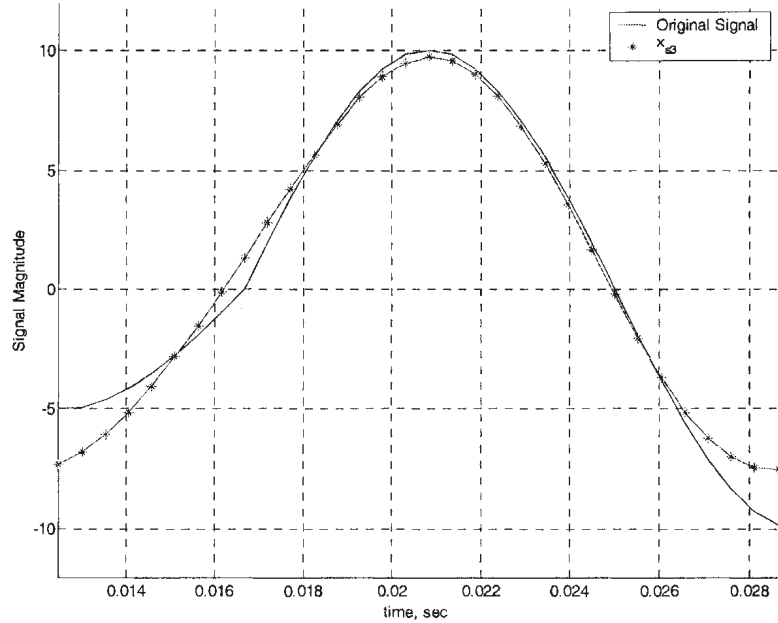


Figure 3.28. The waveform of  $x(t)$  and  $x_{3,3}$  in window2

#### 3.6.2.4 Data Pre-Processing

As illustrated in Section 3.6.2.2, the equivalent gain of MODWT for DC signal is unity and for 2<sup>nd</sup> harmonic is around 0.5.

It is well known that transmission line protective relay may encounter overreach in the presence of DC components in faulted current waveforms. Many algorithms [31][33], have been proposed to remove the DC offsets. Since the study target here is just to investigate the performance of the Wavelet-based algorithm, it can be assumed that the DC components have been removed before the Wavelet analysis.

In general, during power system faults, the 2<sup>nd</sup> harmonic is not significant, and usually can be ignored, during power system faults. Therefore, the proposed MODWT algorithm can work correctly and can be effectively used to extract the phasors. If the

transients contain significant 2<sup>nd</sup> harmonic, the MODWT algorithm cannot extract the fundamentals precisely.

### 3.6.2.5 Boundary Issues

Due to the fact that MODWT works as a circular filter, it automatically treats the samples within a data window as periodic signals. Therefore, in steady-state, i.e., the data window contains “pure” pre-fault data or faulted data, its periodic character can treat the data without any errors. If the data window contains both pre-fault data and faulted data, then the signal loses periodic characters. So, some special treatment should be carried out. In other words, some “boundary” output by MODWT should be discarded.

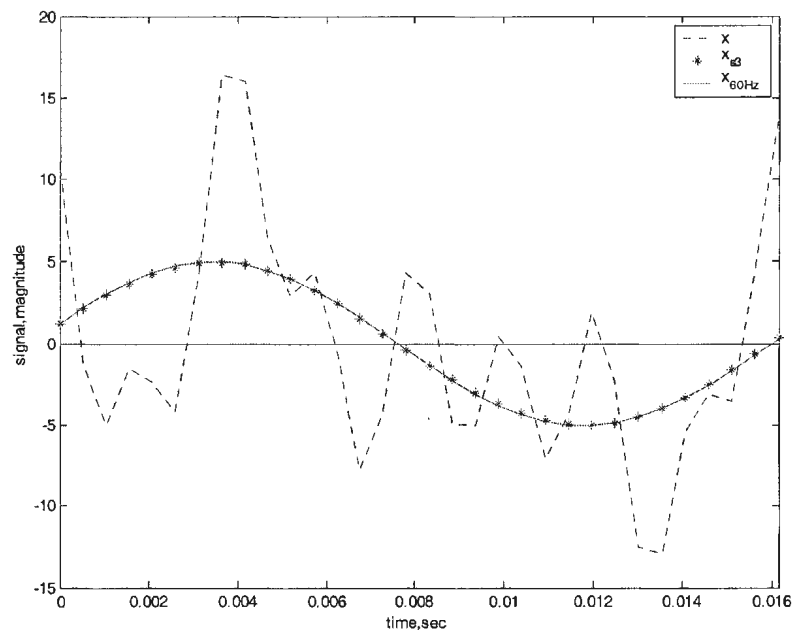


Figure 3.29. The waveforms of  $x(t)$ , 60 Hz signal and  $x_{s,3}$

When a moving data window is used, actually, one output is necessary after the whole window of data is processed. The “best” output should be selected and used as the output of MODWT window. Theoretically, any “middle” outputs can be used as the

outputs. However, to realize a realistic real-time monitoring function, some outputs which are close to the latest boundary should be selected. The following example will illustrate how to choose the “best” output.

Suppose the original signal is described by

$$x(t) = 5 \sin(120\pi t) + 5 \sin(360\pi t) + 5 \sin(480\pi t) + 5 \sin(600\pi t) \quad (3.58)$$

At  $t = 0.0161$ , i.e., when the 32<sup>nd</sup> sample is obtained,  $x_{s3}$  can be computed as shown in Figure 3.29.

Using the two-sample algorithm illustrated in equation (3.6), the equivalent magnitude for  $x_{s3}$  is shown in Figure 3.30.

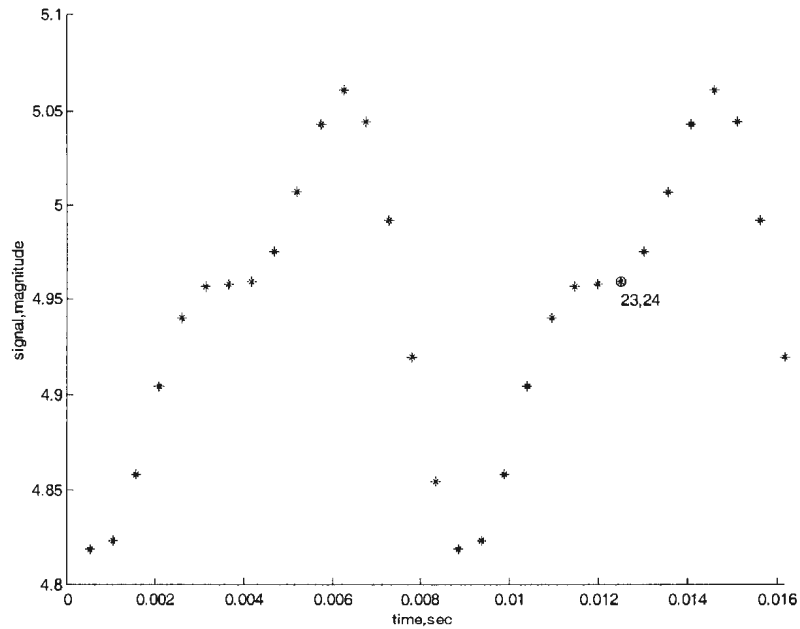


Figure 3.30. The equivalent magnitudes for  $x_{s3}$

Figure 3.30 shows that the equivalent magnitudes for  $x_{s3}$  are varied even in a narrow range, which means that  $x_{s3}$  contains some other harmonics. The reason can be

due to the frequency response characteristics of  $Db(8)$  based MODWT matrix  $M_{s_3}$ . As the figure shows, the magnitude calculated from the 23<sup>rd</sup> and 24<sup>th</sup> elements of  $x_{s_3}$  are close to their adjunct neighbors, which suggests the 23<sup>rd</sup> and 24<sup>th</sup> elements are stable. In addition, these two elements are not far from the right-side boundary, which implies they are sensitive to the transients of the input signal. Therefore, it should be reasonable to choose them as the output of the MODWT window and used to compute the frequency-domain presentation of a time-domain signal.

In addition, since only these two elements are required, the two rows of  $M_{s_3}$  are necessary to be stored in hardware, which can reduce the calculation burden of the relay system.

#### 3.6.2.6 Block Diagram of MODWT Algorithm

Based on the above analysis, a real-time MODWT algorithm, with the block diagram as shown in Figure 3.31, is proposed.

As Figure 3.31 illustrates, the MODWT algorithm realizes the real-time monitoring function by picking the coming sample and discarding the oldest sample. Since the outputs of the moving window, i.e., the equivalent magnitude and phase angle, require two elements (element 23 and 24) of  $x_{s_3}$ , only two rows of matrix calculations are necessary. Note that, the corresponding time tags for the element 23 and 24 of  $x_{s_3}$  lag 9 and 8 sample intervals, roughly 1/4 cycles, compared to  $x(32)$ . Even though, as will be seen in 3.6.2.7, MODWT algorithm is superior to traditional DFT algorithms.

Implementing this algorithm to compute  $V_A$ ,  $V_B$ ,  $V_C$ ,  $I_A$ ,  $I_B$ , and  $I_C$ , respectively, all the phasors can be extracted when new samples arrive.

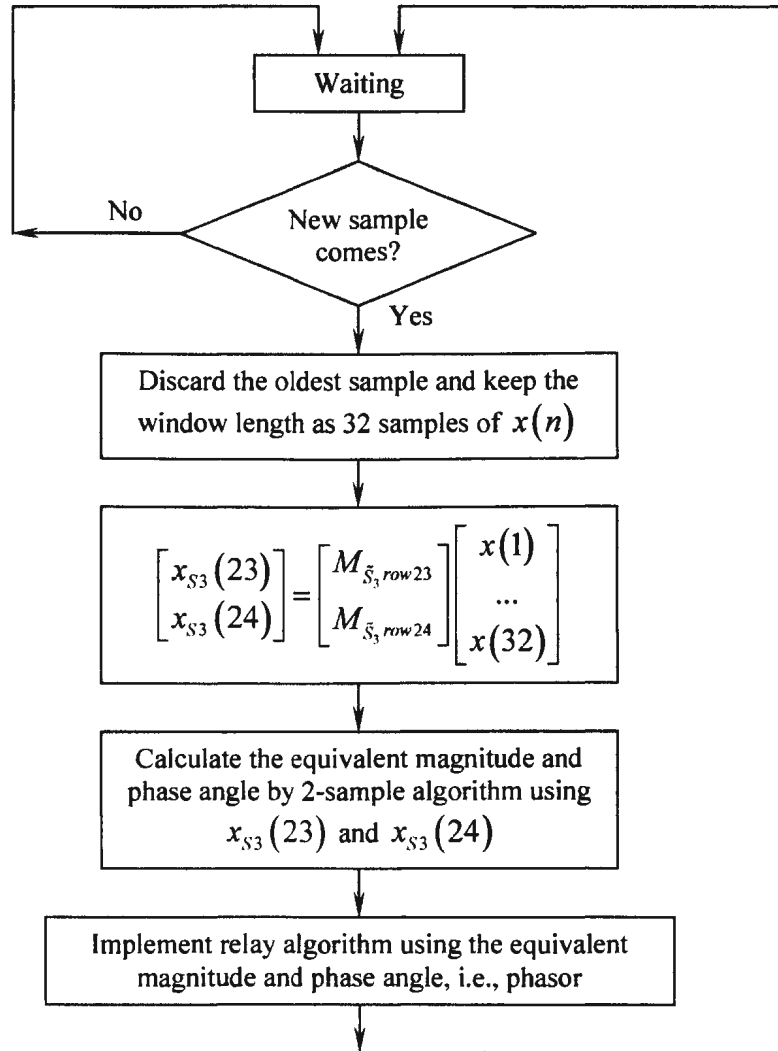


Figure 3.31. Block diagram of the MODWT algorithm

### 3.6.2.7 MODWT and FCDFE Algorithm Comparison

FCDFE algorithm is a well-known technique which is widely used by current commercial digital relays. Figure 3.32 shows its block diagram.

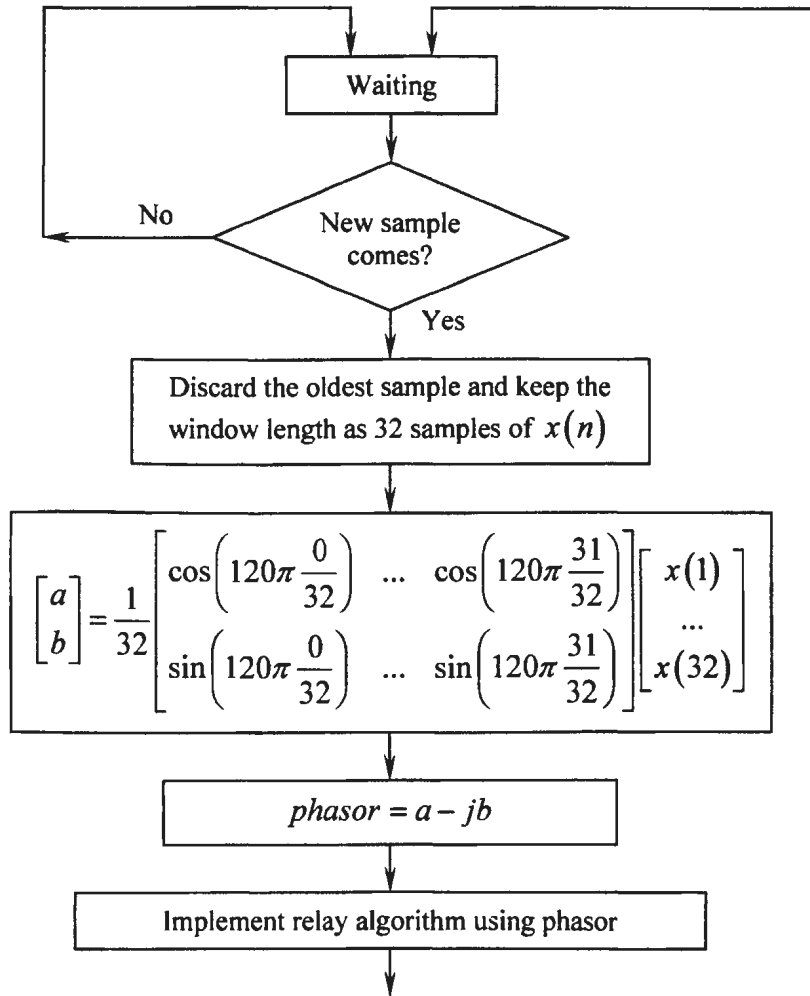


Figure 3.32. Block diagram of the FCDFT algorithm

As can be seen, the only differences from Figure 3.32 to Figure 3.31 are the filtering techniques and phasor calculations. For the filtering techniques, both MODWT and FCDFT need exactly the same numbers of multiplications and additions, since all the coefficients of MODWT and FCDFT can be calculated in advance and then saved in the memory of the relay systems. The only differences appear when the equivalent magnitude and phase angle (MODWT) or the phasor (FCDFT) are calculated. Since MODWT needs the two-sample algorithm, the calculation burden is obviously higher

than FCDFT. However, those contemporary computer-based digital relays can handle the extra calculations in real-time without too much difficulty.

When the steady-state frequency response characteristics are examined, FCDFT will show its excellent filtering ability, compared to MODWT, for the fundamentals (see Figure 3.33). It automatically filters out DC component and any order harmonics. The only frequency component which can pass through the filter without any losses is 60 Hz fundamentals, which is much better than MODWT-based techniques shown in Figure 3.25.

Due to its excellent frequency response characteristics, FCDFT can process the “raw” data which comes directly from CTs or PTs. However, in most cases, an anti-aliasing low pass filter and a high pass filter which rejects the DC offsets are always set up between CTs (or PTs) and those digital devices.

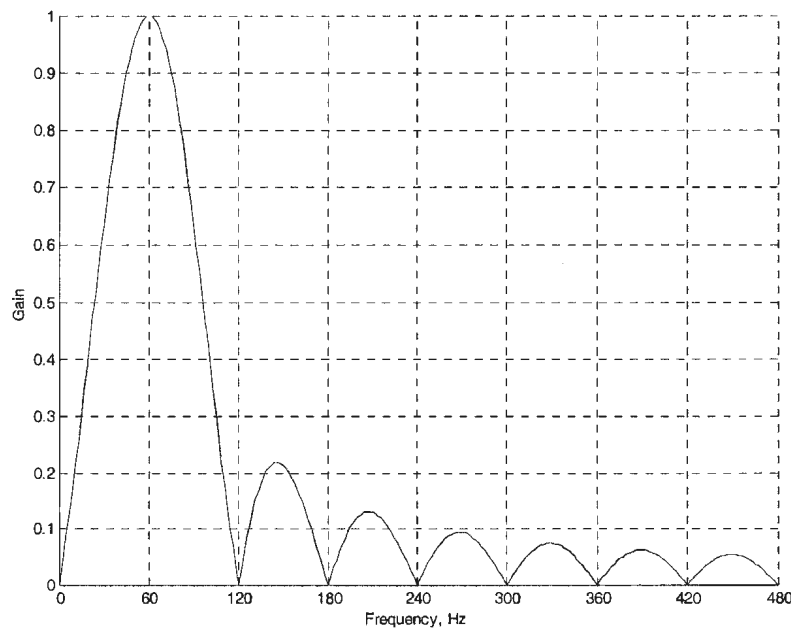


Figure 3.33. The frequency response for FCDFT

However, the transient characteristics of FCDFT are quite poor since Fourier-based techniques lose all the time domain information and only contain frequency domain information during its data window.

The example in 3.6.2.3 can be used to support the statement. Using window 1 and 2 which are defined in 3.6.2.3, the FCDFT outputs are  $-1.4632+j7.3559=7.5\angle(101.25^\circ)$  and  $-8.5819-j2.5080=8.9408\angle(-163.7096^\circ)$ . It is easy to see that MODWT has better results, which are  $9.5278\angle(-89.8462^\circ)$  and  $9.9350\angle(-88.3546^\circ)$ , respectively. Comparing the magnitude, it can be seen that MODWT converges much faster than FCDFT, since the magnitude of the original signal is 10 when  $t > 0.0167$  seconds.

Figure 3.34 shows the magnitude converging progress by FCDFT and MODWT algorithms. For comparison purpose, the original signal is also illustrated. From this figure, it is easy to see that MODWT converges faster.

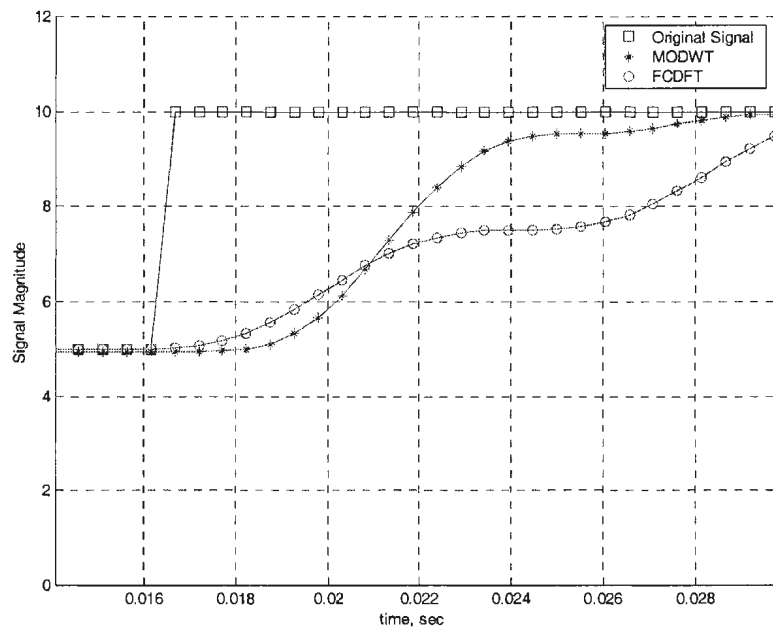


Figure 3.34. Comparison of Convergence Characteristics



### **3.7 Simulations for Wavelet-Based Algorithms**

The two-bus power system, which is modeled in Section 3.3.1, is also used here to study wavelet based algorithm. PSCAD/EMTDC software [20] is used to determine the transient voltage and current signals available at the relay. The relaying performances are simulated in Matlab [21].

The MODWT algorithm, together with the standard DFT algorithm, is investigated for the distance relay using the same power system model introduced in Section 3.3.1. The relay operating characteristic is a mho characteristic set to protect 85% of the transmission line in Zone I. A sampling frequency of 1920 *Hz* is considered for this study.

Figure 3.35 and Figure 3.36 show the impedance trajectories determined by the two methods when a single-phase-to-ground fault occurs inside the relay action zone, i.e., the per unit distance of the fault from the relay location  $m = 0.75$ , and the resistance of the fault path  $R_f = 15\Omega$ . To remove the unwanted DC offset, a 4<sup>th</sup> order Butterworth high pass filter with 30 *Hz* cut-off frequency is implemented.

As can be seen in these figures, both MODWT and FCDFT algorithm can properly function, i.e., the apparent impedance is located inside the relay action zone. In addition, as shown in Figure 3.35, the time tags show that MODWT algorithm converges faster than the FCDFT algorithm. Figure 3.36, which illustrates the impedance trajectories in a different way, shows the convergence progress of the impedance magnitude and phase angle. It also shows the improved convergence feature of the MODWT algorithm, i.e., even though both MODWT and FCDFT algorithms take the same time span to converge

to the steady-state point, the outputs of MODWT algorithm during the transient are closer, than FCDFT algorithm, to the steady-state point.

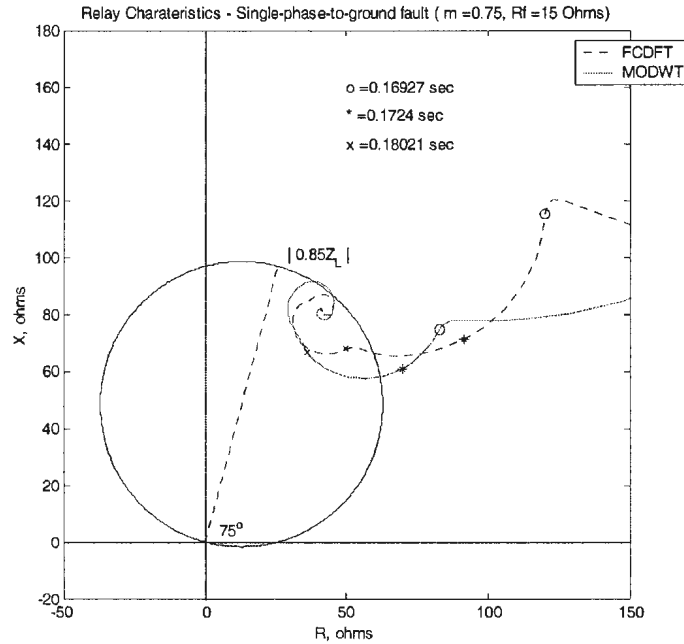


Figure 3.35. Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.75$ ) – the MODWT algorithm vs. the FCDFT algorithm

Figure 3.37 and Figure 3.38 show the impedance trajectories when a single-phase-to-ground fault occurs outside the relay action zone, i.e., the per unit distance of the fault from the relay location  $m = 0.90$ , and the resistance of the fault path  $R_f = 20\Omega$ . Figure 3.37 shows that the apparent impedance calculated by either MODWT or FCDFT correctly converges to the point out of the relay action zone. Therefore, both of the algorithms will not make the relay issue trip signal. The magnitude and phase angle converging progress can be seen in Figure 3.38. This figure indicates that the MODWT converges faster than FCDFT.

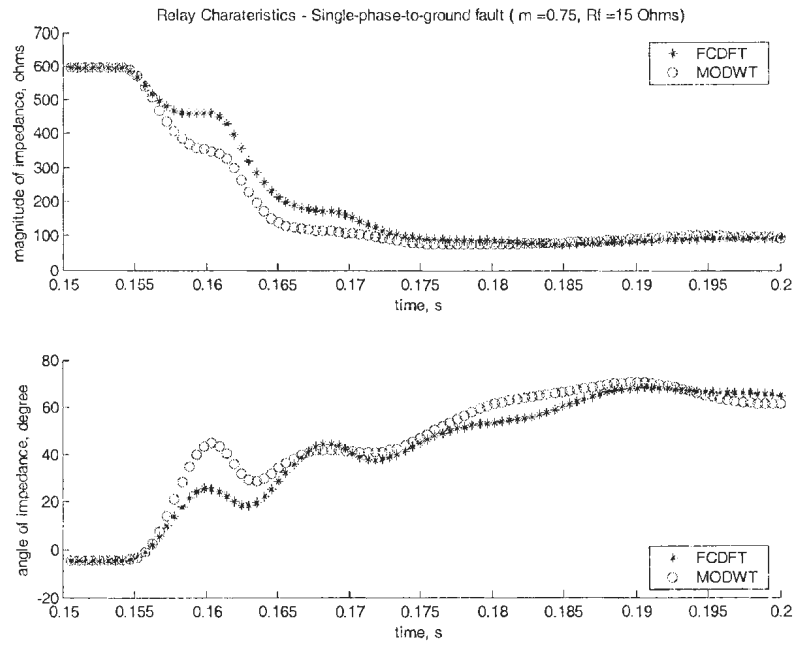


Figure 3.36. Variation of relay impedance for a single-phase-to-ground fault ( $m=0.75$ )

– the MODWT algorithm vs. the FCDFT algorithm

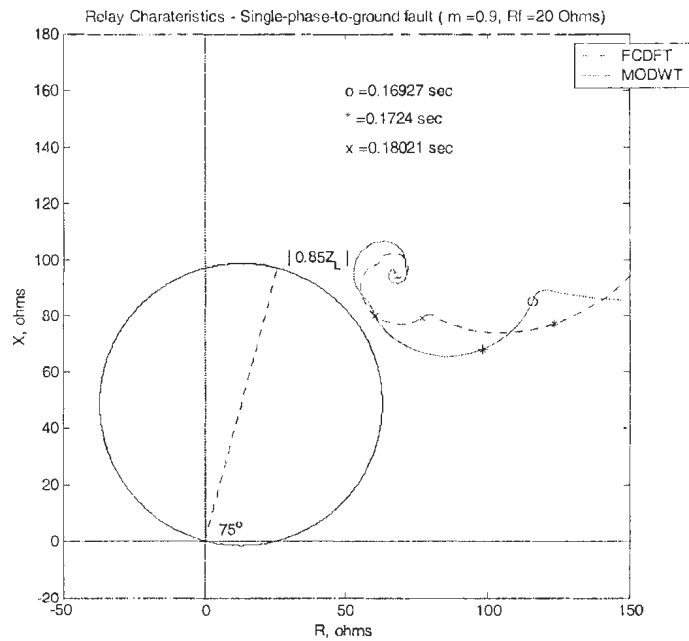


Figure 3.37. Apparent impedance trajectory of a single-phase-to-ground fault ( $m=0.90$ )

– the MODWT algorithm vs. the FCDFT algorithm

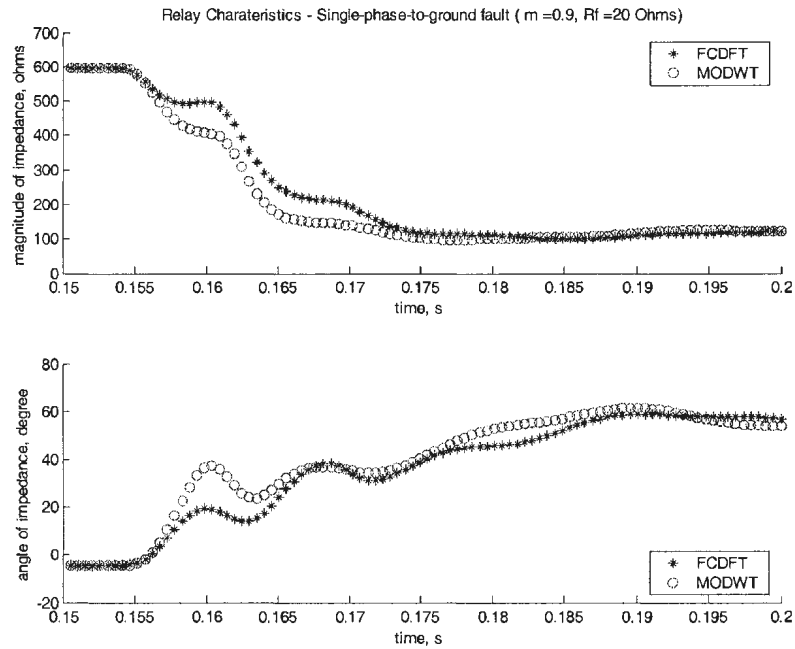


Figure 3.38. Variation of relay impedance for a single-phase-to-ground fault ( $m=0.90$ ) – the MODWT algorithm vs. the FCDFT algorithm

Figure 3.39 and Figure 3.40 show the impedance trajectories determined by the MODWT and FCDFT algorithm when a three-phase-to-ground fault occurs inside the relay action zone. The per unit distance of the fault from the relay location is set as  $m=0.75$ , and the resistance of the fault path is  $R_f=15\Omega$ . Figure 3.41 and Figure 3.42 show the impedance trajectories when a three-phase-to-ground fault occurs out of the relay action zone, i.e., the per unit distance of the fault from the relay location  $m=0.90$ , and the resistance of the fault path  $R_f=20\Omega$ . As can be seen in these figures, both MODWT and FCDFT algorithms function properly. MODWT algorithm converges faster than the FCDFT algorithm.

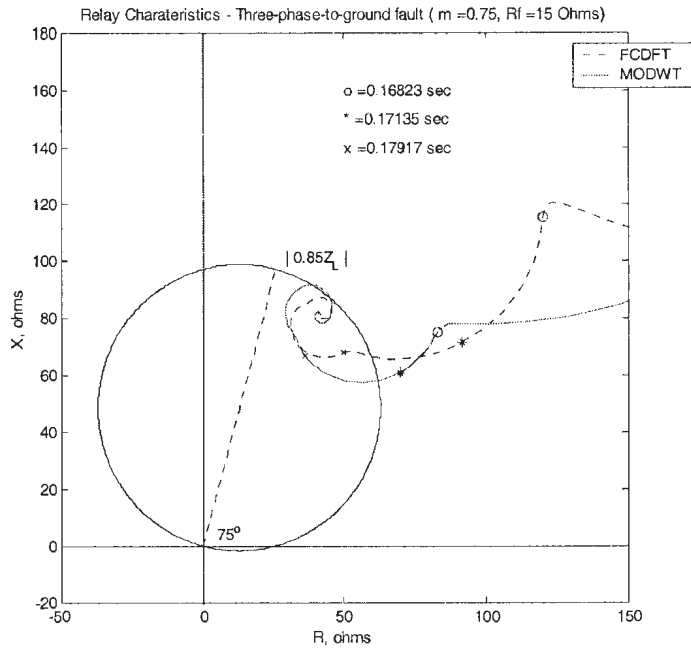


Figure 3.39. Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.75$ ) – the MODWT algorithm vs. the FCDFT algorithm

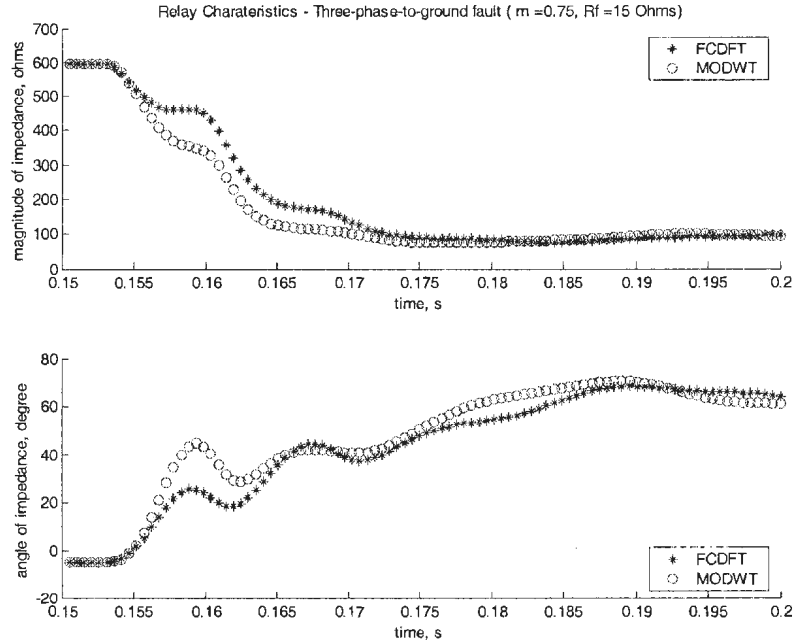


Figure 3.40. Variation of relay impedance for a three-phase-to-ground fault ( $m=0.75$ ) – the MODWT algorithm vs. the FCDFT algorithm

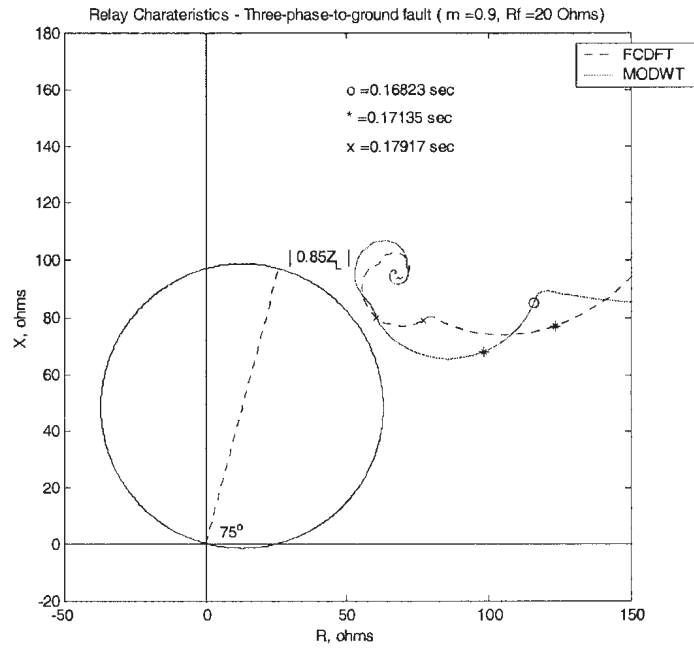


Figure 3.41. Apparent impedance trajectory of a three-phase-to-ground fault ( $m=0.90$ ) – the MODWT algorithm vs. the FCDFT algorithm

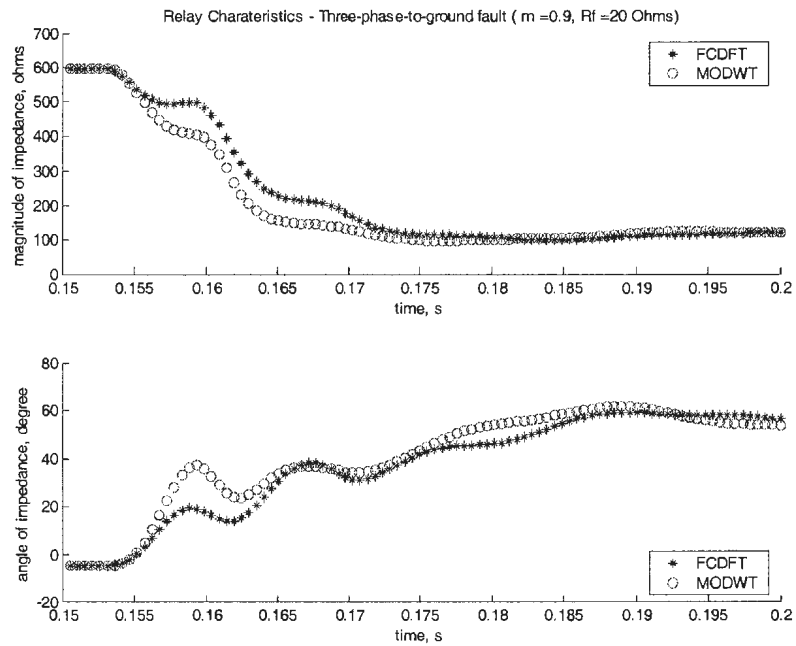


Figure 3.42. Variation of relay impedance for a three-phase-to-ground fault ( $m=0.90$ ) – the MODWT algorithm vs. the FCDFT algorithm

The difference in convergence time for the two methods as presented in this study is small. However, in real-time application, faster convergence is desirable since it will enhance the overall performance of the protection scheme.

In order to investigate the transient characteristics of the above mentioned algorithms, the relative error, defined by equation (3.7), is implemented. As Figure 3.43 and Figure 3.44 indicate, MODWT algorithm has less relative error at different samples compared to the FCDFT algorithm, especially when the data window contains both pre-fault and post-fault data samples.

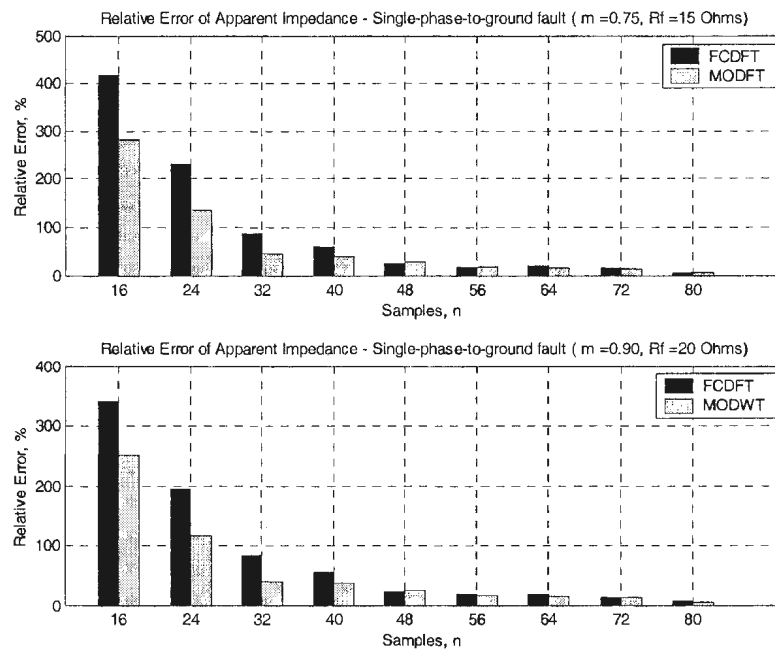


Figure 3.43. Relative errors of the filtering algorithms for single-phase-to-ground faults – the MODWT algorithm vs. the FCDFT algorithm

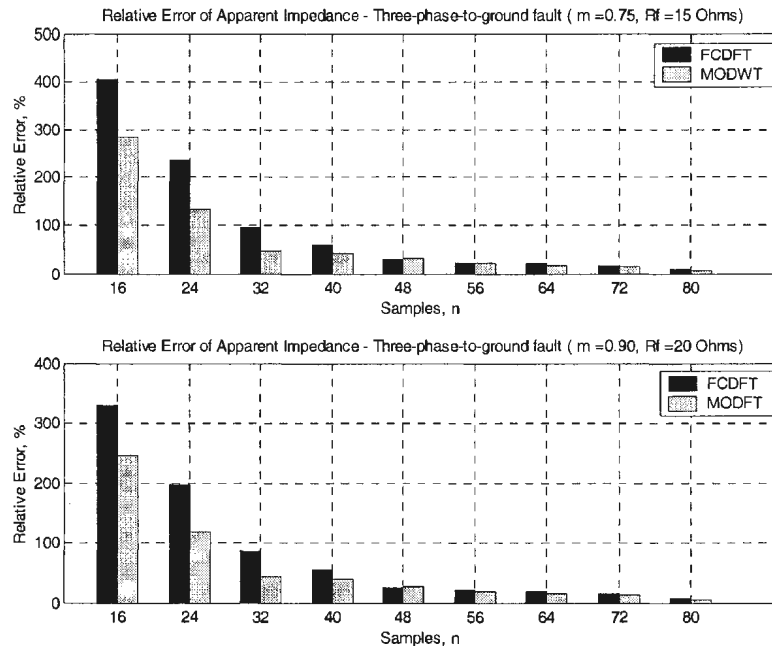


Figure 3.44. Relative errors of the filtering algorithms for three-phase-to-ground faults – the MODWT algorithm vs. the FCDFT algorithm

### 3.8 Summary

In this chapter, three Fourier-based filtering algorithms and a Wavelet-based algorithm are introduced, compared, and simulated. As can be seen, all of these algorithms can be implemented in real time, and all of them reach the converged points at the same time when their data windows contain a full cycle of steady-state measurements. Due to the excellent transient characteristics, the wavelet-based pyramid algorithm can give a better transient characteristic since its every output is closer, compared to the other algorithms, to the converged point. Simulation results also show that the wavelet transform based algorithm converges faster than the Fourier transform based algorithm, such as FCDFT algorithm, for digital distance relay applications. The computational requirement for the wavelet-based method is high. However, with the capability of the



present generation microprocessors, the algorithm can be easily implemented in a digital distance relay.

## **Chapter 4**

# **Methods for Digital Relay Performance**

## **Enhancement**

### ***4.1 Introduction***

Apparent impedance, calculated by the digital distance relays using voltage and current phasors, can be used to determine whether a fault is within or outside the operating zone of the relay and the relays provide a trip signal if the computations indicate that a fault is within the operating zone [17]. In most cases the characteristics of the relays themselves are fixed and do not vary with the power system operating condition. In addition to the fault resistance, in feed into the fault from remote source influences the accuracy of the relay computations. It is possible to implement different computational strategies to overcome the possible errors in the relaying decision. With the capability of the present-day microprocessors the operating characteristics of the relays can be adapted considering the changing power system conditions.

The objective of this chapter is to present some methods that can be incorporated to enhance the performance of digital distance relays for transmission line protection.

The organisation of the chapter is as follows. Section 4.2 investigates the reactance effect which is caused by fault resistance. Section 4.3 provides the basic concepts of transmission line fault location. Section 4.4 presents error compensation method for digital distance protection. Section 4.5 provides an overview of adaptive relaying. Section 4.6 explains how the relay operating characteristics can be changed with power system loading.

## 4.2 Reactance Effect

Reactance effect is defined as the combined effect of the fault resistance and the load current in case that a fault occurs. Figure 4.1 gives a detailed illustration.

In Figure 4.1,  $E_S$ ,  $E_U$ ,  $Z_S$ ,  $Z_U$ , and  $Z_E$  constitute a two-port equivalent of a particular power system, and  $Z_L$  is the impedance of the protected transmission line. A real number,  $h$  (between 0 and 1), indicates the distance from fault location to relay R, and  $R_F$  is the fault resistance. The measurements can be the voltages,  $V_R$  and  $V_Q$ , at relay R and Q respectively, and current,  $I_R$  and  $I_Q$ , through them respectively. The polarities of the voltages and the directions of the currents are shown in the same figure.

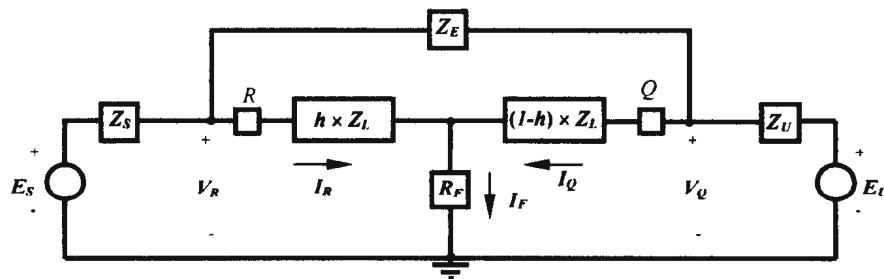


Figure 4.1. Equivalent circuit for a fault on a transmission line using two-port model

Suppose that relay R and Q are distance relays and a three-phase fault occurs.  $V_R$

$V_Q$ ,  $I_R$  and  $I_Q$  satisfy the following equations.

$$\begin{cases} V_R = I_R \times h \times Z_L + I_F \times R_F \\ V_Q = I_Q \times (1-h) \times Z_L + I_F \times R_F \end{cases} \quad (4.1)$$

where,  $I_F$  is the current through  $R_F$  and

$$I_F = I_R + I_Q \quad (4.2)$$

Dividing by  $I_R$ ,  $I_Q$  respectively, equation (4.1) can be written as

$$\begin{cases} \frac{V_R}{I_R} = h \times Z_L + \frac{I_R + I_Q}{I_R} \times R_F = h \times Z_L + \left(1 + \frac{I_Q}{I_R}\right) \times R_F \\ \frac{V_Q}{I_Q} = (1-h) \times Z_L + \frac{I_R + I_Q}{I_Q} \times R_F = (1-h) \times Z_L + \left(1 + \frac{I_R}{I_Q}\right) \times R_F \end{cases} \quad (4.3)$$

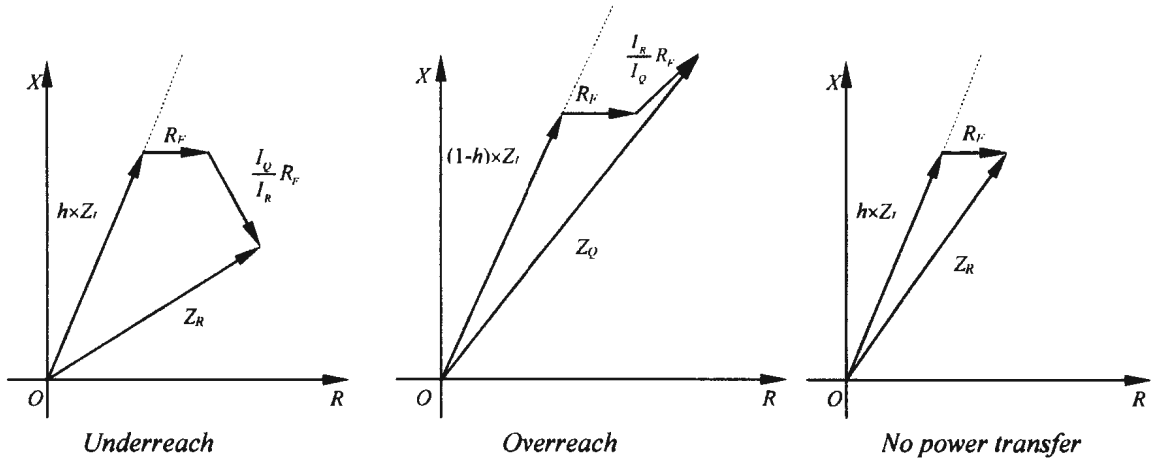


Figure 4.2. Influence of fault resistance on distance relay

Equation (4.3) shows that the apparent impedance, which is obtained directly by  $\frac{V}{I}$ , equals the sum of the line impedance and the fault resistance which is scaled by a complex factor ( $I_R$ , and  $I_Q$  are not in phase). Therefore, from the point view of a distance relay, fault resistance  $R_f$  is seen as a complex quantity, i.e., "reactance effect".

Due to different load conditions, underreach or overreach may occur at the relay. Figure 4.2 gives a graphical illustration under a particular system operating condition that underreach occurs at relay R and overreach may occur at relay Q.  $Z_R$  and  $Z_Q$  are the apparent impedances at relay R and relay Q, respectively. In addition, even for the simplest case, i.e., the prefault current is zero, the apparent impedance is significantly affected by fault resistance. In other words,  $V/I$  does not give the exact impedance between the relay location and the fault point, even for the simplest three-phase fault [34].

### **4.3 Transmission Line Fault Location**

#### **4.3.1 Introduction of Fault Location**

Transmission lines suffer short-circuit faults and are usually isolated by protective relays and power circuit breakers. Faults are mainly caused by severe weather conditions or by arcs initiated by foreign objects. While the majority of the faults are temporary, a permanent fault requires inspection by a maintenance crew before the line can be manually re-energized. Knowing the fault location helps in fast repairs and restoration of the service. Even in the case of temporary faults with successful reclosing, accurate fault location provides the utilities with valuable information to discover trouble spots on the

line. Microprocessor technology has enabled automated and accurate calculation of fault location [19]. The voltage and current waveform values at the line terminals during the fault are the raw data from which the physical location of the fault can be determined. Calculations for fault location are fundamentally different from those used for relaying. Relaying schemes are designed to make correct and fast tripping decision based on the measurements computed in real time. Fault location is needed only after the fault and hence fault location programs are executed after the fault using stored fault data. The different fault location techniques that have been proposed may be broadly classified in two categories: fault location using data from one terminal of the line and fault location using data from two terminals of the line.

#### ***4.3.2 The Differences between Fault Location and Fault Detection***

Although the distance relays usually use the same data, including voltage values and current values, to calculate the apparent impedance for fault detection purpose, the different requirements between fault detection and fault location are obvious.

##### **1. Speed requirement**

In order to remove the faulted transmission lines as soon as possible, the relevant calculation of fault detection should be launched immediately and completed as quickly as possible. Normally, protective relay can accomplish the calculation within 10 to 50 milliseconds [6].

Since the calculation results of fault location are used by the repair crews, the calculation can be triggered after the faults are detected or even when the faults are

isolated. Therefore, a longer response time, which can be in seconds or minutes, is acceptable.

## 2. Accuracy requirement

A typical characteristic of a protective relay can be an operating zone with a particular shape. In case that the calculation of fault detection is located inside the operating zone, a tripping signal should be issued. Therefore, it is unnecessary to implement accurate calculation for fault detection purpose.

On the other hand, because the purpose of fault location is to save time and cost for the repair crews in finding the fault spot, high accuracy is obviously required.

## 3. Algorithm requirement

Compared to the algorithms used for fault detection, the algorithms for fault location are more complicated and elaborate since they should provide higher accuracy. Usually, lots of compensation techniques are implemented, such compensation for fault resistance, compensation for prefault load conditions, and so forth. In addition, using digital filtering techniques, more accurate phasor calculation can be achieved. All these techniques can improve the fault location estimation. However, they are not always suitable for fault detection purpose, because they may be too complicated and may introduce unacceptable delay.

#### 4. Communication requirement

The communications, which are required by relay applications, are always high-speed data exchanging. Due to the off-line characteristic of the fault location techniques, low-speed data transmitting can be accepted.

##### ***4.3.3 Fault Location Algorithms and Advanced Modern Digital Relays***

As illustrated in 4.3.2, fault location techniques can provide accurate fault spot and, at the same time, need more information of system operating condition and are time-consuming. However, innovative developments within microprocessor-based relays have created new ways of reacting to the data available and responding to the possible information that can be extracted from this data. This tendency in turn makes it possible that more and more fault location algorithms can be completed in real time. Due to the inclusion of fault location algorithms into real-time fault detection, the performance of the digital relays can be significantly improved and associated Security/Dependability can be better balanced.

##### ***4.3.4 Factors Affecting the Fault Location Accuracy***

With reference to equation (4.3) and Figure 4.2, it can be easily seen that several factors can affect the accuracy of the fault location.

#### 1. Reactance effect

The majority of the faults on overhead transmission lines are ground faults, and the value of fault resistance is usually quite high. Considering the prefault load together,



therefore, reactance effect cannot be avoided and contribute the most part of the fault location error.

## 2. Measurement errors

The transient response characteristics of Capacitor Coupled Voltage Transformer (CCVT), the iron core saturation of Current Transformer (CT), and the constrictions caused by some other subsystem (such as bit resolution of A/D converter, unsynchronized system clock, the system frequency variation, and so forth) can introduce errors into the relay measurements, voltages and currents.

## 3. Reactance compensation methods

In order to increase the power transfer capability, different compensation techniques are implemented more and more frequently, such as parallel and shunt reactors and capacitors, even FACTS. Due to those applications,  $Z_l$  in equation (4.3) will not simply mean the impedance of the protected transmission line.

## 4. Transmission line model

In equation (4.3), the protected transmission line is simply modelled as an impedance  $Z_l$ , i.e., a resistance in series with a reactance. Actually, A  $\pi$  model or a distributed model can give a more precise description of the line. The insufficient line model of course can introduce fault location error.

## 5. Zero-sequence quantities

Since the above-mentioned analysis is based on a three-phase fault, only the positive sequence quantities are used. As a matter of fact, the most common faults in power system concerns ground faults, and zero-sequence quantities should be taken into account.

At first, it is usually difficult to obtain accurate zero-sequence impedance, because zero-sequence impedance can be affected by soil resistivity, which varies for a number of reasons. Secondly, mutual effects can introduce zero sequence components in some applications. Therefore, zero-sequence quantities can also influence the accuracy of the fault location significantly.

### **4.4 A One Terminal Data Based Fault Location Algorithm**

#### **4.4.1 Algorithm Description**

Distance relays use sample values of voltage and current signals available at the relay location to estimate the impedance of the line from the location of the relay up to the fault point. This apparent impedance differs from the actual impedance of the line. Several factors affect the accuracy of this computation. The combined effect of the load current and fault resistance has a significant influence on the accuracy of impedance calculation. The apparent impedance is compared with the operating characteristics of the relay and the decision to trip or not to trip is taken by the relay. However, it is possible that the influence of load current and fault resistance can cause a significant error in the estimation of the transmission line impedance. The relay may over-reach or under-reach in these conditions and the wrong decision as to trip or not to trip is taken by

the relay. It is proposed in this research to compensate for this error as a part of the relaying algorithm itself. Any addition to the relaying algorithm will increase the computational requirement within each sampling interval. The proposed method uses a simple one terminal transmission line fault location algorithm as part of distance relaying computations. Details of this approach are discussed below.

Consider a faulted network as shown in (i) of Figure 4.3. Suppose that a three-phase fault with fault resistance  $R_F$  occurs at the location from which the p.u. distance to relay R is  $h$  ( $0 \leq h \leq 1$ ). Using superposition theorem, the network (i) can be replaced by a pre-fault network (ii) and a pure-fault network (iii). Then,

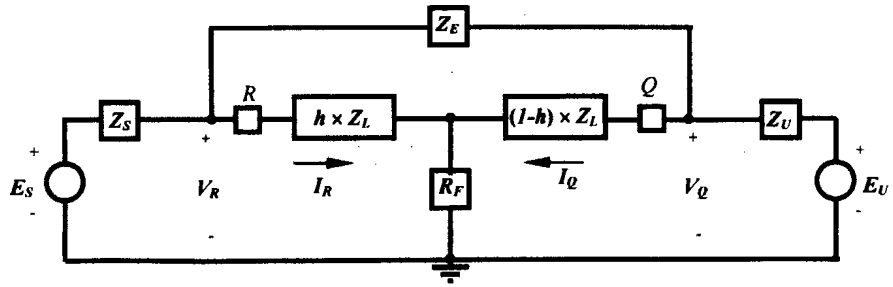
$$V_F = -R_F (I_{FR}'' + I_{FQ}'') \quad (4.4)$$

where,  $V_F$  is the voltage vector across  $R_F$

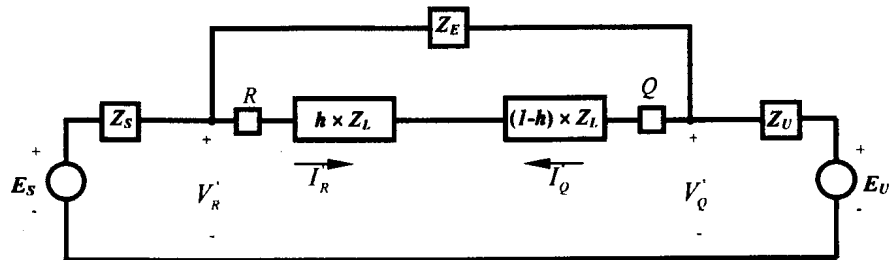
$I_{FR}''$  and  $I_{FQ}''$  are pure-fault currents which flow from fault location to relay R and relay Q side respectively.

A well-known one-terminal fault location algorithm is proposed by A. Wiszniewski in 1983 [35]. Again, to simplify the analysis, the equivalent circuit of a three-phase-faulted network, as shown in Figure 4.3, and a short line model of transmission line are implemented.

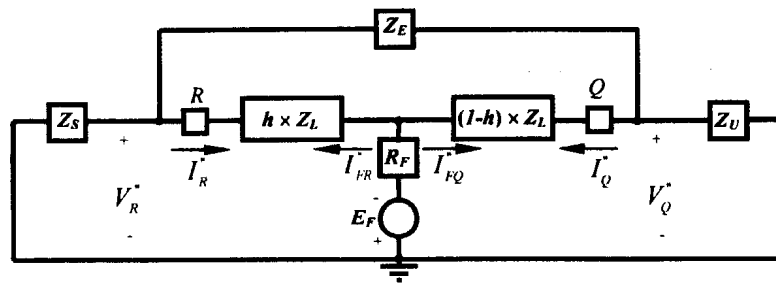
The apparent impedance measured by relay R can be expressed as,



(i). A faulted network



(ii). A prefault network



(iii). A pure-fault network

Figure 4.3. A faulted network and its equivalent decomposition

$$Z_K = \frac{V_R}{I_R} = hZ_L + \frac{I_F}{I_R} R_F \quad (4.5)$$

where,  $Z_R = R_R + jX_R$  (4.6)

$$I_F = I''_R + I''_Q \quad (4.7)$$

$$I_R = I'_R + I''_R \quad (4.8)$$

Using KVL for (iii) of Figure 4.3

$$\begin{cases} I_1(Z_S + hZ_L + R_F) - I_2R_F - I_3hZ_L = E_F \\ -I_1R_F + I_2(Z_U + (1-h)Z_L + R_F) - I_3(1-h)Z_L = -E_F \\ -I_1hZ_L - I_2(1-h)Z_L + I_3(Z_L + Z_E) = 0 \\ I_R'' = I_1 - I_3 \\ I_F = I_1 - I_2 \end{cases} \quad (4.9)$$

Obviously, the pure-fault current  $I_R''$  is a part of the current  $I_F$ , i.e.,  $I_R'' = \mathbf{k}I_F$ , where  $\mathbf{k}$  is a complex number. Solving equation (4.9)

$$\mathbf{k} = \frac{I_R''}{I_F} = \frac{((1-h)Z_L + Z_U)Z_E + (Z_S + Z_U)(1-h)Z_L}{(Z_S + Z_U + Z_L)Z_E + (Z_S + Z_U)Z_L} = ke^{j\lambda} \quad (4.10)$$

Therefore, equation (4.5) can be rewritten as

$$\begin{aligned} R_R + jX_R &= h(R_L + jX_L) + \frac{I_F}{I_R} R_F \\ &= h(R_L + jX_L) + \frac{I_R''}{I_R \cdot ke^{j\lambda}} R_F \\ &= h(R_L + jX_L) + \frac{I_R - I_R'}{I_R \cdot ke^{j\lambda}} R_F \\ &= h(R_L + jX_L) + \frac{I_R - I_R'}{I_R e^{j\lambda}} \cdot \frac{R_F}{k} \end{aligned} \quad (4.11)$$

The real and imaginary parts can be expressed separately as

$$\begin{cases} hR_L = R_R - (R_F/k)a \\ hX_L = X_R - (R_F/k)b \end{cases} \quad (4.12)$$

$$\text{where, } a = \text{Re} \left( \frac{I_R - I_R'}{I_R e^{j\lambda}} \right) \quad (4.13)$$

$$b = \text{Im} \left( \frac{I_R - I_R'}{I_R e^{j\lambda}} \right) \quad (4.14)$$

Due to the fact that the reactor is much greater than the resistance of transmission line, i.e.,  $X \gg R$ , the phase angle  $\lambda$  is assumed to be zero. Actually,  $\lambda$  generally does not exceed  $10^\circ$ . In addition, the ratio of  $X_L/R_L$  is always known and equals to  $\tan \phi_L$  since the parameter of the protected line is known. Therefore,

$$(R_F/k) = \frac{R_R \tan \phi_L - X_R}{a \tan \phi_L - b} \quad (4.15)$$

Accordingly,

$$\begin{cases} hR_L = R_R - \frac{R_R \tan \phi_L - X_R}{\tan \phi_L - (b/a)} \\ hX_L = X_R - \frac{R_R \tan \phi_L - X_R}{(a/b) \tan \phi_L - 1} \end{cases} \quad (4.16)$$

In equation (4.16),  $Z_R = R_R + jX_R$  is the measured apparent impedance, and the others are system parameters. Therefore, the fault location  $h$  can be determined.

This algorithm can provide fairly accurate result of the fault location. The only factor that can affect the accuracy is the assumption that  $\lambda$  is zero.

#### 4.4.2 Simulation Results

The method for dynamic error compensation is investigated for distance protection of the same system in Section 3.3.1. Single line to ground fault with a fault resistance of 40 ohms is applied at a location 150 miles from the relay location. The relay operating characteristic is a mho characteristic set to protect 85% of the transmission line in Zone I. The fundamental frequency, voltage and current phases required by the relay are determined using full cycle Fourier Algorithm. A sampling frequency of 1200 HZ is

considered for this study. The relaying algorithm is simulated in Matlab [21]. After the occurrence of the fault, the apparent impedance determined by the relay changes as it approaches the operating characteristics of the relay. PSCAD/EMTDC software [20] is used to determine the transient voltage and current signals available at the relay.

Figure 4.4 shows the performance of the method which compensates for the error as a part of the distance relaying algorithm. The trajectory converges to the fault-point. For comparison, the trajectory obtained by determining the impedance based on the traditional approach, without error compensation is also shown. This trajectory converges to a point outside the exact fault point. The fault will be identified as in Zone II and this will delay the operation of the protective relay which is not desirable.

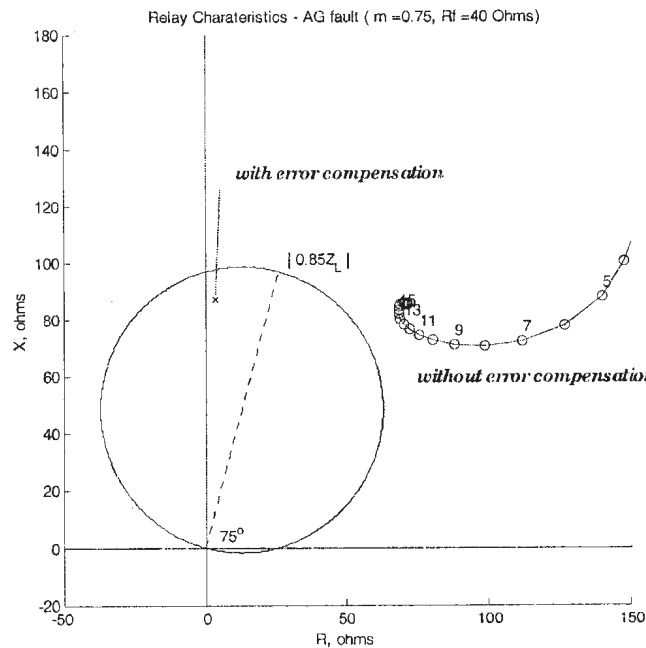


Figure 4.4. Impedance Trajectory with Error Compensation (Fault at  $m = 0.75$ )

Figure 4.5 shows the performance of this method for a different loading condition and for a fault at 180 miles from the relay location. The fault is outside zone I and is

located correctly by compensating for the error. The computational requirements of the proposed method are higher than the method which does not compensate for the error. However, it is within the capability of present-day microprocessors.

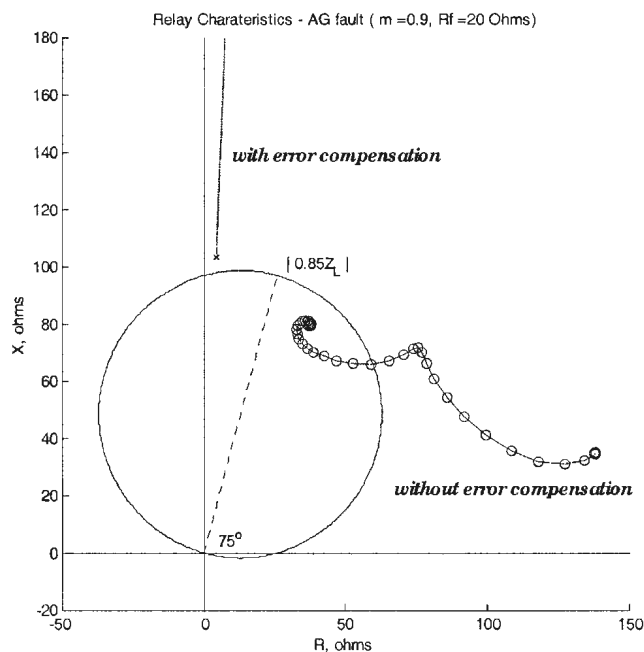


Figure 4.5. Impedance Trajectory with Error Compensation (Fault at  $m = 0.90$ )

The converged value of the apparent impedance is obtained approximately one cycle after the fault inception. Full cycle Fourier algorithm is relatively slow as the data window should contain sampled values of current and voltage signals for one cycle of the fundamental frequency. The apparent impedance calculation including compensation for the fault resistance also converges after one cycle. For the results presented here it was observed that suitable logic has to be included so that the relaying decision is made after the apparent impedance converges [13][16].



## **4.5 Adaptive Relaying**

Adaptive protection is a philosophy which permits and makes adjustments to protection functions automatically for making the protection more attuned to the prevailing power system conditions [36]. Digital relays contain two important features that are vital to the adaptive relaying concept; their functions are determined through software, and they have a communication capability which can be used to alter the software in response to higher level supervisory software, or under commands from a remote control center. With these features, microprocessor-based relays can adjust their operating characteristics to reduce the impact of faults and disturbances on power systems and their equipment. Some of the specific areas that can benefit from incorporating adaptive features in protection are:

- Balancing security and dependability
- Changing relay settings as system configuration changes
- Out-of Step protection
- Compensating for pre-fault load flow

The research presented in this thesis uses the concept of adjusting the operating boundary of a distance relay. Recently the concept of adaptive protection has been expanded to include preventive and emergency control features [37]. One of the goals is to position the protection system to be more robust in the event of a threat or hidden failures. Another possible feature is to modify the protection system to defend against future events in case of a component failure. In addition, the rich resource of information

obtained by digital relays makes it possible to verify the fundamental frequency power system model that was used to determine the relay application and settings [38]. These were previously only observable either by special monitoring equipment connected for specific tests, or by expensive fault recording equipment.

## **4.6 An Adaptive Boundary Algorithm**

### **4.6.1 Algorithm Description**

Microprocessor-based distance relays are usually designed with fixed relay setting. However, if the system conditions vary over a wide range and faults occur through high resistance the relay may lose selectivity. In order to provide coverage for high resistance faults, the relay operating characteristics should be adjusted. It has been shown [39] that this ideal operating region of a distance relay is mainly governed by the active and reactive power flow in the line. One method of adjusting the operating characteristic is to calculate the operating region within the normal range of active and reactive power flow offline for a specific system configuration. This can be stored and the relay can use the characteristic that is close to the existing operating condition. However, the loading patterns and system configuration can change frequently from the assumed patterns and this will affect the accuracy of the relaying decision.

It is proposed to determine the operating characteristics periodically in the energy management system and communicated to the microprocessor-based relay. The communication capability of the relays makes this feasible. For large power systems a two-port equivalent model of the protected system can be used to determine the operating characteristics. The real-time data available in the Energy Managing Systems has enough

information on the voltages, power flow in the system as well as the topology of the power system. A two-port equivalent model can be derived easily to determine the ideal operating characteristics of the relay [40]. Thus the operating characteristic of the relays can be adjusted to be in tune with the power system.

#### **4.6.2 Simulation Results**

Under the same operating conditions of the power system which is used in 4.4.2, the ideal operating characteristic method is investigated for the protection of the two-terminal high voltage system. Figure 4.6 shows the ideal characteristic (boundary 1) for real and reactive power flow into the system behind the relay. A single line to ground fault at 75 % of the line from the relay location, with a fault resistance of 40 ohms is considered. The trajectory of the apparent impedance converges to a point outside the mho characteristics but inside the new operating region and thus protects the line when the power flows into the system behind the relay.

Figure 4.7 shows the ideal characteristic (boundary 2) for real and reactive power flow into the system in front of the relay. A single line to ground fault at 75 % of the line from the relay location, with a fault resistance of 40 ohms is considered. Correct operation of the relay is obtained with the updated boundary.

Figure 4.6 and Figure 4.7 indicate that the boundaries change significantly with the loading condition. By adjusting the relay characteristics the relay responds to the fault correctly. For the study presented here, the boundaries of the ideal operating characteristics (4 segments) are calculated for fault-resistance varying from 0 to 50 ohms and for fault location from 0 to 85 % of the line from the relay location.

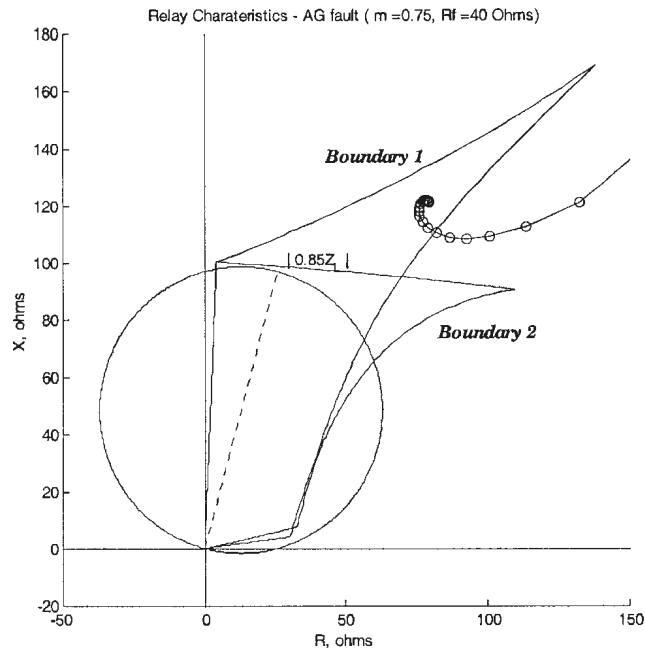


Figure 4.6. Impedance Trajectory with Adaptive Boundary (Fault at  $m = 0.75$ , Power flow into the system behind the relay)

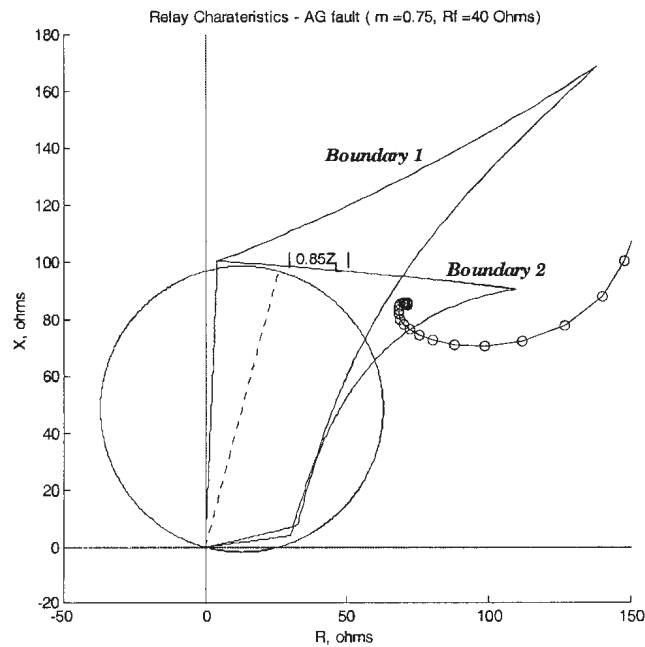


Figure 4.7. Impedance Trajectory with Adaptive Boundary (Fault at  $m = 0.75$ , Power flow into the system in front of the relay)

## **4.7 Summary**

The results presented illustrate that both the concepts (error compensation and adaptive boundary) can be useful for accurate relaying decision. The requirements to implement both the methods are significantly different. The algorithm to compensate for error can be easily implemented in any stand-alone microprocessor distance relay. The algorithm for the apparent impedance calculation has to be modified based on equation (4.5). The existing tripping characteristics can be used.

For the method which changes the operating boundary, suitable communication features are required. The Energy Management System determines the operating region of the relay regularly and updates the information stored in the relay. Any algorithm, which determines the apparent impedance, is adequate for this method. Even though the computational requirements of the relays to implement the proposed concepts are different from conventional microprocessor-based relays, they can be achieved using the capability of present day microprocessors.

It is extremely important for utilities to maintain quality power supply for customers. After a fault occurs, accurate fault location can significantly reduce the recovery time and aid rapid fault analysis. On the other hand, more and more fault location techniques can be implemented as real-time fault detection tools by present powerful digital relays, which can also significantly improve the immunity of the digital relays to the dramatically changed system operating condition.

Among the factors, which can affect the accuracy of fault location, reactance effect seems to be one of the key factors. However, elaborately designed fault location

algorithms, including one-terminal algorithms and two-terminal algorithms, and well-organized estimation scheme can efficiently eliminate or reduce the influence of those factors.

This chapter has presented two concepts that can be implemented in microprocessor-based relays for transmission line distance protection. The error in conventional distance relay computations due to fault resistance can be dynamically compensated as part of the relaying algorithm. By adjusting the operating logic of the relay, the relay can be in tune with the changing power system loading and provide accurate relaying decisions. With the increasing speed and capability of the available microprocessors it is possible to incorporate these innovative features in microprocessor-based relays.

## **Chapter 5**

### **Conclusions and Future Work**

#### ***5.1 Contributions of the Research***

The rapid progress in electrical power technology has made it possible to construct economic and reliable power systems capable of satisfying the continuing growth in the demand for electrical energy. Power system protection plays a significant part and progress in the field of power system protection is a vital prerequisite for the efficient operation and continuing development of power supply systems as a whole. Fast and accurate locating and isolating faults in an electrical transmission line has become increasingly important, as transmission lines are a vital link between the generating system and distributing system.

Due to the fact that power system is modeled using fundamental frequency quantities, such as currents, voltages, impedances, real powers, reactive powers, etc., many power transmission line protective relays use phasors to realize different kinds of relay algorithms. It is extremely important for the relays to extract the phasors from faulted currents and voltages quickly and accurately.

To achieve this aim, Fourier based filtering algorithms, including the Full Cycle Discrete Fourier Transform (FCDFT) algorithm, play very important roles, and they are widely implemented in commercial distance protective relays. However, the FCDFT algorithm needs one cosine window and one sine window to extract the phasors. From the required computation, accuracy, and convergent speed points of views, the FCDFT algorithm has obvious drawbacks. One-cycle cosine window with a quarter delay algorithm only requires one cosine window since the sine window is just a quarter-cycle delay of the cosine window. It may introduce fewer errors compared to the FCDFT algorithm. However, this algorithm needs extra a quarter-cycle delay to obtain the phasors. From the speed point view, theoretical analysis and simulation results show that it is not suitable for real-time application. With the implementation of the two sample algorithm, which is proposed in this thesis, it is possible to extract the phasors only using the cosine window with one sample delay, i.e., the one-cycle cosine window with one sample delay algorithm. Simulation results illustrate that this algorithm converges faster and has better accuracy compared to the others. For this algorithm, the computation requirements are higher but can be easily implemented in present generation microprocessors.

The sine window and the cosine window which are implemented in Fourier based algorithms are not suitable for transient analysis since sine function and cosine function are steady-state functions. On the other hand, wavelet transform based techniques have excellent transient characteristics because wavelet functions themselves are transient ones. With wavelet based techniques becoming popular in the area of power system, protective relays are experiencing improvements related to shorter decision time, as well



as in being accurate. The research proposes and demonstrates a real-time based pyramid filtering algorithm for transmission line protective relays with the use of Maximal Overlap Discrete Wavelet Transform (MODWT). The simulation using PSCAD/EMTDC and Matlab shows that the MODWT algorithm converges faster compared to the conventional Fourier based algorithms. From the computation point of view, the new algorithm has similar requirements as the one-cycle cosine window with one sample delay algorithm.

Two additional relaying algorithms are also investigated in this thesis. Due to the reactance effect of the fault resistance, under-reach or over-reach may occur under different power system operating conditions. An one-terminal data based fault location algorithm, which compensates the fault resistance, is investigated and simulated in this thesis. Simulation results show that the error in conventional distance relay computations due to fault resistance can be dynamically compensated. An adaptive boundary algorithm, which uses two-terminal data, is also investigated here. By adjusting the operating logic of the relay according to different load conditions, the relay can be in tune with the changing power system loading and provide accurate relaying decisions. With the increasing speed and capability of the available microprocessors it is possible to incorporate innovative features studied in the thesis in microprocessor-based relays.

It should be mentioned here that this thesis has not specifically focussed on the computational requirements of different relay algorithms. The new algorithms investigated in this research require additional computations compared to the conventional distance relay algorithms. However, with capabilities of present-generation

microprocessors, the author of this thesis does not foresee the computing demand as a challenge.

## **5.2 Suggestions for Future Work**

The work reported in this thesis can be extended in the following areas:

- The wavelet function which is implemented in the proposed wavelet filtering algorithm is chosen as Daubechies (8) wavelet. Further work can examine other wavelet functions in order to find the most suitable wavelet functions for digital distance relaying schemes.
- The phasors calculated by the wavelet algorithm is based on the 23<sup>rd</sup> and 24<sup>th</sup> samples of the wavelet pyramid algorithm output. This decision is made through some simplified computations. The detailed derivation can be carried out in future work.
- All the simulations in this thesis only concern a two bus system. Considering the complexity of the real power system, further investigation can be carried out using a larger power system. In addition, more fault patterns other than single-phase-to-ground fault or three-phase-to-ground fault should be investigated for all the proposed algorithms.

# Appendix

## Simulation Tools for Power System Analysis

### *1. Introduction*

The electric power system is very complex. Its defining mathematical expressions become more intricate and challenging. To get deep understanding about the power system operating characteristics, especially the power system transients, powerful simulation tools are extremely important. Currently, several simulation tools are widely used. In this chapter, PSCAD/EMTDC, Pspice, and Matlab are introduced. In addition, a small two-bus power system is simulated using these software tools.

The rest of the chapter is organized as follows. Section 2 introduces a widely used power system simulation software, PSCAD/EMTDC. Section 3 investigates some features of Pspice, which is frequently adopted by electrical engineers. Matlab is introduced in Section 4. A simple two-bus system is examined in Section 5 using PSCAD/EMTDC, Pspice, and Matlab, respectively. Section 6 gives a summary.

## **2. PSCAD/EMTDC**

PSCAD/EMTDC, which was developed by Manitoba HVDC Research Centre, is one of the most popular power system transient simulation tools. It contains two parts. One is EMTDC [41], and the other is PSCAD [42].

EMTDC, Electro-Magnetic Transients in DC system, is a transient simulator which has been evolving since the mid-1970s [43]. Its development has always been completely independent of Electromagnetic Transient Program (EMTP) [44] and its many derivatives. The first early versions of EMTDC were run at Manitoba Hydro on mainframe computers using IBM punched cards. Present EMTDC is a versatile tool to study AC as well as DC power systems problems. Today, the powerful personal computers allow detail and depth in simulation that could not even be imagined 30 years ago. EMTDC is most suitable for simulating the time domain instantaneous responses, also popularly known as electromagnetic transients of electrical systems.

The following are some of the common components used in systems studied using EMTDC [41]:

- Resistors (R), inductors (L), capacitors (C)
- Mutually-coupled windings such as transformers
- Distributed frequency dependent transmission lines and cables
- Current and voltage sources
- Switches, breakers

- Diodes, thyristors, GTO's
- Analog and digital control functions
- AC machines, exciters, governors, stabilizers and inertial models
- Meters and measuring functions
- Generic DC and AC controls
- HVDC, SVC, and other FACTS devices

Typically, EMTDC is used by engineers from utilities, manufactures, consultants, research and academic institutions alike. It is used in planning, operation, design, commissioning, preparation of tender specifications, teaching and advanced research. Some typical examples are listed as the following.

- Contingency studies of ac networks consisting of rotating machines, exciters, governors, turbines, transformers, transmission lines, loads
- Relay coordination
- Insulation coordination of transformers, breakers and arrestors
- Impulse testing of transformers
- Subsynchronous resonance (SSR) study of networks with machines, transmission lines, HVDC
- Filter design and harmonic analysis
- Control system design and coordination of FACTS and HVDC

- Optimal design of controller parameters
- Investigation of new circuit and control concepts

PSCAD, Power System Computer Aided Design, is a graphical user interface which facilitates the simulation studies on EMTDC. It is a family of tools designed to help simulate power systems. With the emergence of Windows operating systems for personal computers and their expanding capabilities, the Manitoba HVDC Research Centre has published the latest version, known as PSCAD/EMTDC.

Some features of PSCAD Version 3 are listed as the following [42]:

- **Totally Integrated:** the run-time plots can be put alongside of the circuit or arranged on a separate page. Circuits, plots and descriptive comments can all be printed together.
- **Modular:** Electrical systems can be split into different modules (or pages) without having to connect them using transmission lines. Control systems can be modeled in separate modules.
- **Hierarchical:** Circuits assembled using basic building blocks can be contained inside modules (also called pages) which can in turn contain more such modules. Double-clicking on one of these modules opens the circuit inside.
- **Graphical Component Design Tool:** new PSCAD components can be designed in a completely graphical environment called Component Workshop. This is the tool used to write/edit all the components in the Master library.

### **3. Pspice**

SPICE stands for Simulation Program with Integrated Circuit Emphasis. The University of California, Berkeley, developed the original SPICE program in the mid 1970s. Since then, many other companies have developed commercial versions. MicroSim originated PSpice in 1985. In 1998 OrCAD acquired MicroSim. OrCAD then more or less integrated PSpice with their own Capture schematic entry software [45].

OrCAD PSpice can be counted on for accurate simulation results. That accuracy has been proven by thousands of engineers. Since 1985, it has been continuously improved, and structured to use the latest hardware and operating systems. Each generation has addressed technological advances in electronics, keeping designers current with the capabilities they need. The current version, OrCAD PSpice Release 9, features OrCAD's new Interchange™ Architecture. Now the world's most popular analog simulator is part of a complete information-sharing solution for any design team.

OrCAD PSpice is a full-featured simulator for serious analog designers. Simulations cover the range from high-frequency systems to low-power IC designs with its sophisticated internal models. Designers can draw on its large library of models for off-the-shelf parts, or create models for new devices from their data sheets.

OrCAD PSpice A/D is a sophisticated, native mixed-signal simulator and a superset of OrCAD PSpice. It is capable of simulating mixed-signal designs of any size, containing analog and digital parts ranging from IGBTs and pulse width modulators to DACs and ADCs. The simulation results, both analog and digital, can be viewed in the same window and on the same time axis.

OrCAD PSpice A/D Basics is an entry-level mixed-signal simulation tool. It's ideal if a friendly environment for simulating simple or basic analog and mixed-signal designs is needed. It imposes no limits on circuit size, performs functional simulations of digital parts in mixed signal simulations, and all of the basic PSpice analyses can be performed.

PSpice Optimizer automates the iterative process of re-running simulations and fine-tuning the design. It can be used in conjunction with OrCAD PSpice or OrCAD PSpice A/D. PSpice Optimizer calculates the optimal parameter values for the circuit under consideration.

#### **4. Matlab**

MATLAB<sup>®</sup> [46], which is one of the most widely used computer-based research tools, is a high-performance language for technical computing developed by MathWorks, Inc. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building



MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows researchers to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow one to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system consists of five main parts:

#### 1. Development Environment

This is the set of tools and facilities that helps one to use MATLAB functions and files. Many of these tools are graphical user interfaces. The development environment includes the MATLAB desktop and Command Window, a command history, and browsers for viewing help, the workspace, files, and the search path.

#### 2. The MATLAB Mathematical Function Library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

#### 3. The MATLAB Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

#### 4. Handle Graphics®

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow fully customized appearance of graphics, as well as complete graphical user interfaces on the MATLAB applications.

## 5. The MATLAB Application Program Interface (API)

This is a library that allows users to write C and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

### 5. Simulation Example

- *A Two-Bus Power System*

To illustrate how the above-mentioned tools can be used to simulate power system operating characteristics, a two-bus power system is implemented. The single line diagram of the system is shown in Figure A.1.

Using the base voltage value of 345 KV, the component parameters are defined in the Table A.1.

Table A.1. Parameters of the study system - 2

Component	Parameters
$G_1$	300 MVA, $X_d'' = X_d' = X_d = 0.15$ p.u.
$G_2$	300 MVA, $X_d'' = X_d' = X_d = 0.015$ p.u.
$T_1, T_2$	300 MVA, $X = 0.10$ p.u.
$T_1$	300 MVA, $X = 0.10$ p.u.
Transmission Line	$L = 0.8$ mH/km, $R = 0.036$ $\Omega$ /km, $C = 0.0112$ $\mu$ F/km, length = 130 km

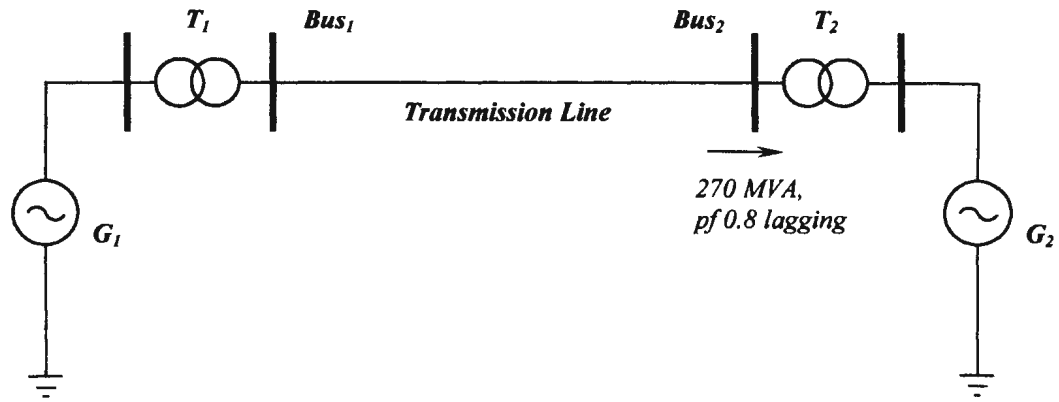


Figure A.1. Single line diagram of a two bus system

Assume that the voltage and the complex power at Bus<sub>2</sub> are 325 KV (line-to-line voltage) and 270 MVA with 0.8 power factor lagging. At  $t = 0.3$  sec., a three-phase-to-ground fault occurs at the location 78 km from Bus<sub>1</sub>. The fault resistance is zero. The simulation goal is to investigate the phase A current and voltage at Bus<sub>1</sub> during the fault period.

The steady-state, i.e., pre-fault, phase voltage of G<sub>1</sub> and G<sub>2</sub> are:

$$\begin{cases} V_{G1} = 222.7892 \sin(376.99t + 21.0248^\circ) \\ V_{G2} = 172.0344 \sin(376.99t) \end{cases} \quad (\text{A.1})$$

- **Simulation by EMTDC/PSCAD**

Figure A.2 and Figure A.3 show the EMTDC/PSCAD simulation scheme and result, respectively. As can be seen in Figure A.3, the phase A current and voltage at Bus<sub>1</sub> change drastically before and after 0.3 seconds, which is due to the effect of the three-phase-to-ground fault.

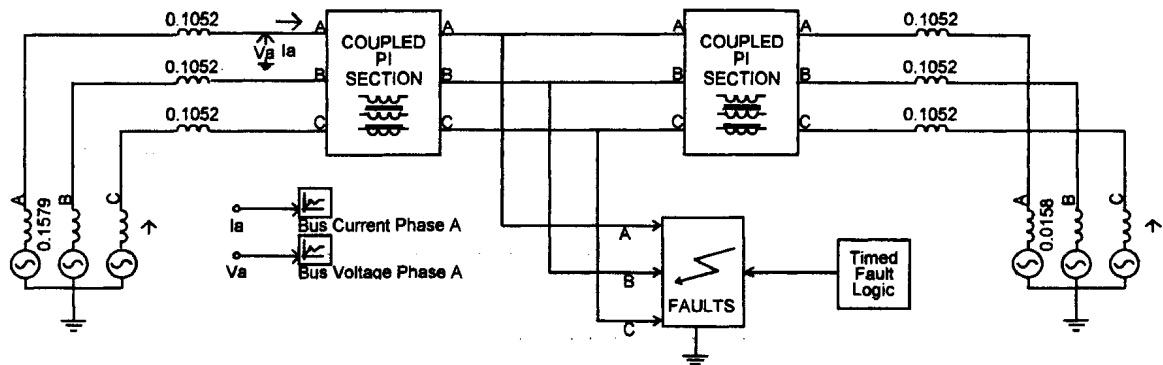


Figure A.2. EMTDC/PSCAD simulation scheme for the two bus system

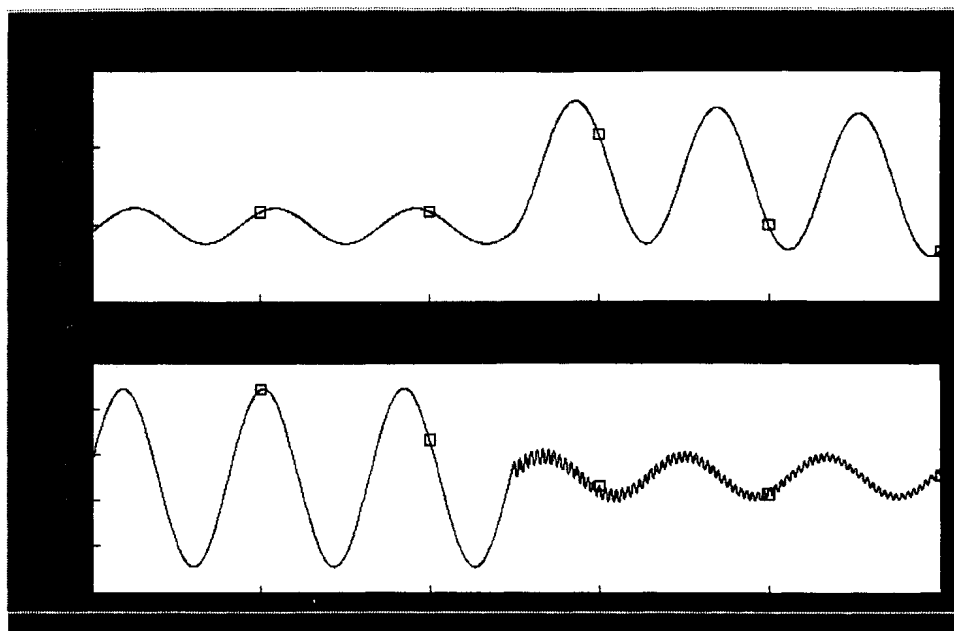


Figure A.3. EMTDC/PSCAD simulation result for the two bus system

- **Simulation by Pspice**

Since only three-phase-to-ground fault is investigated, a single line representation can be used so as to simplify the problem. Figure A.4 shows the system schematic in Pspice simulation, which uses the same parameters given in Table A.1.

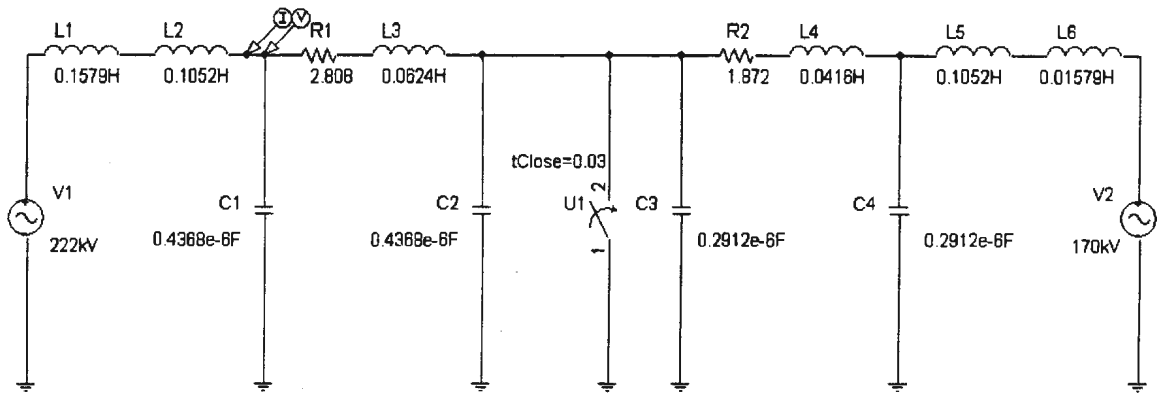


Figure A.4. PSPICE simulation scheme for the two bus system

As long as the schematic is well defined, Pspice will automatically convert it to a set of differential equations and provide the numerical solutions. Figure A.5 shows the relevant simulation result.

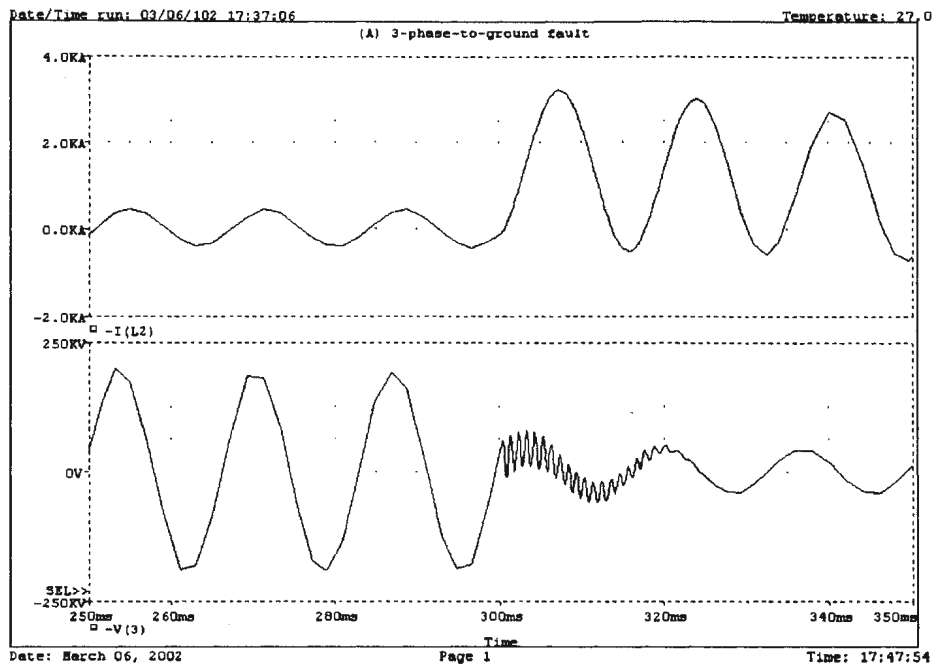


Figure A.5. PSPICE simulation result for the two bus system

- **Simulation by Matlab**

Mathematically, the system described in Figure A.1 is an RLC circuit and can be described using differential equations. Since the prefault analysis is straightforward, only the faulted equivalent circuit, i.e.,  $t \geq 0.3$  second, is illustrated in Figure A.6.

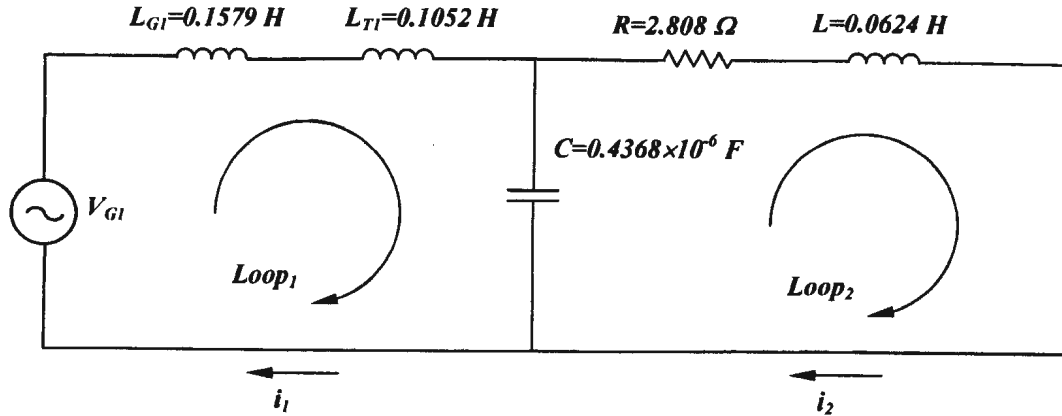


Figure A.6. Equivalent faulted circuit for the two bus system

Using the notation in Figure A.6, the differential equations can be written as:

$$\begin{cases} V_{G1} = (L_{G1} + L_{T1}) \frac{di_1(t)}{dt} + \frac{1}{C} \int_{0.3}^t (i_1(\tau) - i_2(\tau)) d\tau \\ Ri_2(t) + L \frac{di_2(t)}{dt} + \frac{1}{C} \int_{0.3}^t (i_2(\tau) - i_1(\tau)) d\tau = 0 \end{cases} \quad \text{for } t \geq 0.3. \quad (\text{A.2})$$

Therefore, the differential equations in (App.2) can be solved by Matlab using ODE23. The initial conditions can be easily obtained according to the steady-state of the prefault system.

The simulation result can be seen in Figure A.7.

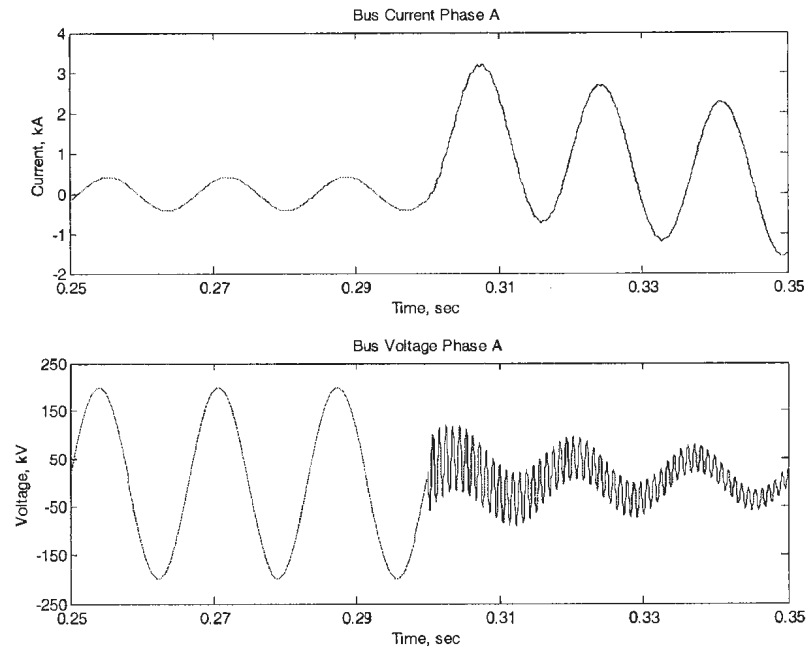


Figure A.7. Matlab simulation result for the two bus system

## 6. Summary

As can be seen, all of the simulation tools work efficiently and correctly. Since those tools emphasize different aspects of the simulated systems, the simulation results are slightly different.

EMTDC/PSCAD is most suitable for power system analysis. Different power system parameters, such as the time constant for the electrical machines, the zero sequence impedance of the transmission lines, etc., can be specified. In general, since this software is designed by power system engineers, the power system operating characteristics have been thoroughly considered. Therefore, it is a popular simulation tool for power system analysis. It is chosen in the study of digital relay performance investigated in this thesis.



PSPICE is more suitable to simulate analog and digital electric circuits. Different IC chips, power suppliers, and other electronic components, have been selected in the libraries. Since it is not specially designed for power system analysis, even a simple generator has to be constructed by three power supply sources, which means every phase angle or other parameters of each phase have to be specified. In addition, the zero sequence impedances of transmission lines are also very difficult to model. Hence PSPICE is not suitable for power system simulations.

Matlab is a well-known data analysis software. It provides not only plenty of built-in functions but also very flexible data processing approaches.

In the study of this thesis, the power system operating data is provided by EMTDC/PSCAD, and all the relay algorithms are simulated in Matlab. This approach can achieve the highest simulation efficiencies by effectively using the features of all these tools.

## References

- [1] J. L. Blackburn, *Protective Relaying principle and Applications*, Marcel Dekker, INC., 1987.
- [2] C. Christopoulos, A. Wright, *Electrical Power System Protection*, Kluwer Academic Publishers, 1999.
- [3] A. T. Johns, S. K. Salman, *Digital Protection for Power Systems*, Peter Peregrinus Ltd., 1995.
- [4] G. D. Rockefeller, "Fault Protection with a Digital Computer", *IEEE Transactions*, Vol. PAS-88, No. 4, April 1969, pp. 438-464.
- [5] M. Ramamoorthy, "A Note on Impedance Measurement Using Digital Computers", *IEEE-IERE Proceedings*, Institute of Electronic & Radio Engineers, Bangalore, India, Nov.-December 1971, Vol. 9, No. 6, pp. 243-247.
- [6] M. S. Sachdev (Coordinator), *Advancements in Microprocessor Based Protection and Communication*, IEEE Tutorial Course, IEEE, Tutorial Course, Publication No. 97TP120-0, NJ, USA.
- [7] C. W. Taylor, "Improving Grid Behavior", *IEEE Spectrum*, Vol. 36, No. 6, June 1999, pp. 40-45.

- [8] Sun-Li, Jyh-Cheng Gu, "Removal of Decaying DC in Current and Voltage Signals Using a Modified Fourier Filter Algorithms", *IEEE Transactions on Power Delivery*, Vol. 16, No. 3, July 2001, pp 373-379.
- [9] A. G. Phadke, J. S. Thorp, *Computer Relaying for Power Systems*, Research Studies Press Ltd., London, and John Wiley & Sons, New York, 1988.
- [10] D. J. Marihart, "Communications Technology Guidelines for EMS/SCADA Systems", *IEEE Transactions on Power Delivery*, Vol. 16, No. 2, April 2001, pp 181-188.
- [11] SEL Product Catalogue, Schweitzer Engineering Laboratories, Inc., Pullman, WA, USA, 2000.
- [12] H. Altuve, I. Diaz, E. Vázquez, "Fourier and Walsh Filtering Algorithms for Distance Protection", *IEEE Transactions on Power Systems*, Vol. 11, No. 1, February 1996, pp. 457-462.
- [13] A.A. Girgis, W. Chang, E.B. Makram, "Analysis of High-Impedance Fault Generated Signals Using a Kalman Filtering Approach", *IEEE Transactions on Power Delivery*, Vol. 5, No. 4, October 1990, pp. 1714-1724.
- [14] L.L. Lai, W.L. Chan, C.T. Tse, A.T.P. So, "Real-time Frequency and Harmonic Evaluation Using Artificial Neural Networks", *IEEE Transactions on Power Delivery*, Vol. 14, No. 1, January 1999, pp. 52-59.

- [15] T. S. Sidhu, D. S. Ghotra, M. S. Sachdev,, "A Fast Distance Relay Using Adaptive Data Window Filters", *IEEE Power Engineering Society Summer Meeting Proceedings*, July 2000.
- [16] E. Rosolowski, J. Izykowski, B. Kasztenny, "A New Half-Cycle Adaptive Phasor Estimation Immune to the Decaying DC Component for Digital Protective Relaying", *Proceeding NAPS 2000*, University of Waterloo, Canada, October 2000, pp. 2-17- 2-24.
- [17] S. H. Horowitz, A.G. Phadke, *Power System Relaying*, Research Studies Press Ltd., England, 1995.
- [18] E O. Schweitzer.III, Daqing Hou, "Filtering for Protective Relays", *47th Annual Georgia Tech. Protective Relaying Conference*, Atlanta, Georgia, April 28-30, 1993.
- [19] D. Novosol, et.al, "Unsynchronized Two-Terminal Fault Location Estimation", *IEEE Transactions on Power Delivery*, Vol. 11, No. 1, January 1996, pp. 130-138.
- [20] PSCAD/EMTDC Simulation Software Program, Version 3, Manitoba HVDC Research Centre, Manitoba, Canada.
- [21] Matlab Version 12, The Mathworks, Inc, Natick, MA, USA, September 2001.
- [22] B. Jeyasurya, Feng Liang, "Intelligent Relaying Concepts for Transmission Line Digital Distance Protection", *2001 North American Power Symposium*, Texas, USA, Oct. 2001.

- [23] K. K. Li, L.L. Lai, A.K. David, "Stand Alone Intelligent Digital Distance Relay", *IEEE Transactions on Power Systems*, Vol. 15, No. 1, February 2000, pp. 137-142.
- [24] Lokenath Debnath, editor, *Wavelet Transforms & Time-Frequency Signal Analysis*, Birkhäuser Boston, c/o Springer-Verlag New York, Inc., New York, NY, USA, 2001.
- [25] Shyh-Jier Huang, Cheng-Tao Hsieh, Ching-Lien Huang, "Application of Morlet Wavelets to Supervise Power System Disturbances", *IEEE Transactions on Power Delivery*, Vol. 14, Issue 1, Jan. 1999, pp. 235-243.
- [26] Shyh-Jier Huang, Cheng-Tao Hsieh, "High-Impedance Fault Detection Utilizing a Morlet Wavelet Transform Approach", *IEEE Transactions on Power Delivery*, Vol. 14, Issue 4, Oct. 1999, pp. 1401-1410.
- [27] T.H. Shi, H. Zhang, P. Liu, D.J. Zhang, Q.H. Wu, "Accelerated Trip of Power Transmission Line Based on Biorthogonal Wavelet Analysis", *IEEE Power Engineering Society Summer Meeting*, 2000, Vol. 3, 2000, pp. 1333-1337.
- [28] Chi-kong Wong, Ieng-tak Leong, Chu-san Lei, Jing-tao Wu, Ying-duo Han, "A Novel Algorithm for Phasor Calculation Based on Wavelet Analysis", *IEEE Power Engineering Society Summer Meeting*, 2001, Vol. 3, 2001, pp. 1500-1503.
- [29] A.P.S. Meliopoulos, Chien-Hsing Lee, "An Alternative Method for Transient Analysis via Wavelets", *IEEE Transactions on Power Delivery*, Vol. 15, Issue 1, Jan. 2000, pp. 114-121.

- [30] D. B. Percival & A. T. Walden, *Wavelet Methods for Time Series Analysis*, Cambridge University Press, New York, NY, USA, 2000.
- [31] A. V. Oppenheim, A. S. Willsky, I. Young, *Signals and Systems*, Prentice-Hall, Inc., New Jersey, USA, 1983.
- [32] G. Benmouya, "Removal of DC-Offset in Current Waveforms Using Digital Mimic Filtering", *IEEE Transactions on Power Delivery*, Vol. 10, Issue 2, April 1995, pp. 621-630.
- [33] Jyh-Cherng Gu, Sun-Li Yu, "Removal of DC Offset in Current and Voltage Signals Using a Novel Fourier Filter Algorithm", *IEEE Transactions on Power Delivery*, Vol. 15, Issue 1, Jan., 2000, pp. 73-79.
- [34] J. Roberts, A. Guzman, E. O. Schweitzer. III, "Z = V/I Does not Make a Distance Relay", *20th Annual Western Protective Relay Conference*, October 1993, Spokane, Washington, USA.
- [35] A. Wiszniewski, "Accurate Fault Impedance Locating Algorithm", *IEE Proceedings*, Vol. 130, Pt. C, No. 6, Nov. 1983, pp. 331-314.
- [36] IEEE Power System Relaying Committee, "Feasibility of Adaptive Protection and Control", *IEEE Transactions on Power Delivery*, Vol. 8, No. 3, July 1993, pp. 975-983.
- [37] M. J. Damborg, et.al., "Adaptive Protection as Preventive and Emergency Control", *IEEE Power Engineering Society Summer Meeting Proceedings*, July 2000.

- [38] C. F. Henville, "Digital Relay Reports Verify Power System Models", *IEEE Transactions on Power Delivery*, Vol. 13, No. 2, April 1998, pp. 386-393.
- [39] K. K. Li, L. L. Lai, "Ideal Operating Region of Digital Distance Relay under High Resistance Resistance Earth Fault", *Electric Power Systems Research*, Vol. 43, 1997, pp. 215-219.
- [40] P.M. Anderson, *Power System Protection*, IEEE Press, 1999.
- [41] EMTDC user manual, Manitoba HVDC Research Centre, 1988.
- [42] PSCAD user manual, Manitoba HVDC Research Centre, 1994.
- [43] <http://www.pscad.com/main/pscad/history.html>.
- [44] <http://www.emtp.org>.
- [45] <http://pcb.cadence.com/Product/Simulation/PSpice/>
- [46] [http://www.mathworks.com/access/helpdesk/help/base/relnotes/r12/relnotes\\_tocframe.shtml](http://www.mathworks.com/access/helpdesk/help/base/relnotes/r12/relnotes_tocframe.shtml)





