Improving the efficiency and accuracy of spectral analysis with applications to harmonic analysis

Jiangtao Xi
University of Wollongong
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
IMPROVING THE EFFICIENCY AND ACCURACY OF SPECTRAL ANALYSIS WITH APPLICATIONS TO HARMONIC ANALYSIS

A thesis submitted in fulfilment of the requirements for the award of the degree of DOCTOR OF PHILOSOPHY

from THE UNIVERSITY OF WOLLONGONG

by JIANGTAO XI

B.E., Beijing Institute of Technology, China, 1982
M.E., TsingHua University, China, 1985

Department of Electrical
and Computer Engineering
1995
Dedicated to my family
Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

............................................................

Jiangtao Xi
Acknowledgements

I would like to first of all thank Associate Professor Joe F. Chícharo for his invaluable guidance throughout this research. Associate Professor J. F. Chícharo has been very generous with his time in helping me with various aspects of the research, I am extremely grateful to him.

My special thanks go to Australian International Development Assistant Bureau (AIDAB) for awarding me the EMSS scholarship throughout the work. In particular I appreciate the kind help from Ms. Valri Nunn for her role in extending the scholarship.

I sincerely thank Professor C. D. Cook for his assistance during my graduate studies. I also deeply appreciate the assistance and friendship of the people in the Department of Electrical and Computer Engineering at the University of Wollongong, particularly Dr. Mehdi T. Kilani and Mr. Ali Y. Varjani.

I wish to sincerely thank my wife Aidong, and my daughter Maomao, for their patience, support and understanding. During the first year of this research, they were left in China. After joining me for a short period, they went to New Zealand, where they experienced many difficulties. My wife gave birth to our little son, Xiaoyu in New Zealand while I was not there. I hope we can start a normal life after finishing this thesis.

Finally, I am deeply grateful to my parents, who have been a constant source of encouragement. I also wish to thank my sisters for their unfailing support.
Author's Publications

Much of the work in this thesis has been published or has been submitted for publication as academic papers. These papers are:


5. Jiangtao Xi and Joe F. Chicharo, "Computation of running DCT's and DST's based on their second order shift properties," Accepted for publication on *IEEE Trans. Signal Processing*

6. Jiangtao Xi and Joe F. Chicharo, "A time domain interpolation approach for DFT harmonic analysis," Accepted for publication on *Signal Processing* (Journal of European Signal Processing Society)


8. Jiangtao Xi and Joe F. Chicharo, "Computing running discrete Hartley transform and running discrete W transforms based on adaptive LMS algorithm" accepted for publication by *IEEE Trans. on Circuits and Systems* subject to revision.


During my doctoral studies, I also did some work on adaptive array signal processing. This research does not appear in this thesis, but is in the following publications:


Abstract

This thesis deals with spectral analysis. A number of new techniques are proposed which improve the computational efficiency of discrete orthogonal transform algorithms, and the accuracy of spectral analysis with applications to harmonic signal analysis.

The techniques for computing discrete orthogonal transforms using adaptive filtering are systematically investigated. Firstly, a general relationship between orthogonal transforms and adaptive filtering is established, which sets the foundation for these techniques. Secondly, the issue of computing block-based discrete orthogonal transforms using the adaptive Least Mean Square (LMS) algorithm is examined. Sufficient conditions for implementing block LMS-based discrete orthogonal transforms are proposed, and the performance of the techniques for computing block LMS-based discrete Walsh transforms and discrete cosine transforms is analysed. Thirdly, the thesis proposes LMS-based techniques for computing running orthogonal transforms, including running discrete Hartley transform, running discrete Cosine and Sine transforms, as well as running discrete W transforms. Finally, the thesis examines the possibility of orthogonal analysis using other adaptive processing algorithms. It is shown that the sample matrix inversion (SMI) algorithm can be used to compute all the discrete orthogonal transforms, while the adaptive Howells-Applebaum loop can be used to implement a spectral analyser for continuous signals.

The running computation of discrete orthogonal transforms based on their shift properties is studied in detail. A number of discrete orthogonal transforms, including the discrete Hartley transform, the discrete cosine and sine transforms, and the discrete W transforms are considered. The shift properties of these transforms are developed, which are in effect
recursive equations that connect the previous and updated transform coefficients. Both the first order shift properties and the second order shift properties are proposed for these transforms. As expected the first order shift properties are in the form of first order difference equations. These first order difference equations involve two transform coefficients of different transforms (for example, a discrete cosine transform and its corresponding discrete sine transform). This is a source of extra computational burden. For some transforms such as the discrete Hartley transform and the discrete W transform, this extra computational burden can be eliminated by using the reverse symmetrical properties of the transform coefficients. However, for the discrete cosine and sine transforms, the computation associated with the first order shift properties is not very efficient. The second order shift properties (that is, second order difference equations) are proposed, which can independently update transform coefficients thus reducing the computational burden. It is shown that for the discrete cosine and sine transforms, the computational burden associated with second order shift properties can be considerably reduced in comparison to the first order shift properties.

A time domain interpolation pre-processing algorithm is proposed in an effort to reduce leakage effects associated with the DFT analysis of periodic signals. The leakage effect refers to the spreading of energy from one frequency bin into adjacent ones. To avoid leakage, the sampling frequency should be an integer multiple of the signal frequency. The basic idea of the proposed time domain interpolation pre-processing algorithm is to modify the actual samples towards an ideal sample sequence whose sampling frequency is an integer multiple of the signal frequency. The algorithm is based on a first order approximation of the Taylor's series. It is shown that the proposed algorithm can reduce quite significantly both the DFT leakage and the truncation error associated with digital wattmeter power measurement.
Finally, frequency estimation techniques based on adaptive IIR notch filtering are considered. In order to improve the steady state error performance of existing techniques, a block-gradient based adaptive algorithm is proposed for adaptive IIR filtering. The input signal sequence is arranged into data blocks and the filter coefficients are kept constant within each block. The gradients are evaluated for each complete block of data and the filter coefficients are updated on a block by block basis. Application of the proposed block gradient algorithm to the problem of sinusoidal frequency estimation is studied. It reveals that the proposed algorithm is characterised by a lower steady state error as well as reduced computational complexity.
# Contents

## List of Symbols

## Chapter 1. Preliminaries

1.1 Introduction .................................................................................................................. 1  
1.2 Spectral Analysis: A Brief Historical Review .............................................................. 2  
1.3 Background to The Thesis ............................................................................................ 6  
  1.3.1 Discrete Orthogonal Transforms ........................................................................... 6  
    1.3.1.1 Fast Orthogonal Transforms ........................................................................ 7  
    1.3.1.2 Computing Orthogonal Transforms Based on Adaptive Filtering .......... 9  
    1.3.1.3 Computing Running Discrete Orthogonal Transforms .............................. 9  
  1.3.2 Harmonic Signal Analysis ....................................................................................... 11  
    1.3.2.1 Harmonic Analysis Using DFT ..................................................................... 11  
    1.3.2.2 Sinusoidal Frequency Estimation Based on Adaptive IIR Notch Filtering ............................................................................................................................................................................... 13  
1.4 Approach and Contributions of This Thesis ............................................................... 14  
  1.4.1 Adaptive Filtering-Based Orthogonal Analysis .................................................... 15  
  1.4.2 Computing Running Discrete Orthogonal Transforms Based on Their Shift Properties ................................................................................................................................................................................. 16  
  1.4.3 A Time Domain Interpolation Approach for Harmonic Signal Analysis Using the DFT ........................................................................................................................................................................... 17  
  1.4.4 Frequency Estimation Based on Adaptive IIR Notch Filtering ......................... 17  
1.5 Summary of Contributions in Order of Presentation .................................................. 18
Chapter 2. Orthogonal Analysis Based On Adaptive Filtering techniques

2.1 Introduction .................................................................................................................. 21
2.2 A General Relationship between Orthogonal Analysis and Adaptive Filtering .... 22
2.3 Sufficient Conditions for Implementing Block LMS-based Orthogonal Transforms 25
2.4 Performance of LMS-based Block Orthogonal Transforms ..................................... 32
   2.4.1 LMS-based Discrete Walsh Transform ............................................................. 32
   2.4.2 LMS-based Discrete Cosine Transform ........................................................... 34
   2.4.3 Other LMS-based Discrete Orthogonal Transforms ....................................... 36
2.5 LMS-based Running Orthogonal Transforms ............................................................ 38
   2.5.1 LMS-based Running Discrete Hartley Transforms ......................................... 38
   2.5.2 LMS-based Running Discrete Cosine/Sine Transforms .................................. 42
   2.5.3 LMS-based Running Discrete W Transforms .................................................. 50
2.6 Orthogonal Analysis Based on Other Adaptive Algorithms ..................................... 53
   2.6.1 SMI-based Discrete Orthogonal Transforms .................................................... 53
   2.6.2 Adaptive Howells-Applebaum Orthogonal Analyser ....................................... 54
2.7 Conclusions .................................................................................................................. 57

Chapter 3. Computing Running Discrete Orthogonal Transforms
Based on Their Shift Properties

3.1 Introduction .................................................................................................................. 60
3.2 Running DCT's and DST's Based on Their Shift Properties ..................................... 61
   3.2.1 First Order Shift Properties and Their Implementation ................................. 62
Chapter 4. A Time Domain Interpolation Algorithm For Periodic Signal processing

4.1 Introduction........................................................................................................... 96
4.2 The Time Domain Interpolation Algorithm ......................................................... 100
4.3 Performance Analysis.......................................................................................... 102
4.4 Simulation Results............................................................................................... 108
  4.4.1 Leakage Analysis .......................................................................................... 108
  4.4.2 Parameter Estimation .................................................................................. 110
4.4 Application of The Proposed Algorithm to Digital Wattemeters .................. 113
4.6 Conclusion ........................................................................................................... 115
Chapter 5. Frequency Estimation Based On IIR Filtering: A new block gradient-based algorithm

5.1 Introduction ........................................................................................................................................ 117
5.2 A Frequency Estimation Scheme Based on Adaptive IIR Notch Filtering ........................................ 120
5.3 Gradient-based Algorithm for Adaptive IIR Filtering ....................................................................... 122
5.4 A Novel Block Gradient-based Algorithm ......................................................................................... 129
5.5 Simulation Results for Sinusoid Frequency Estimation .................................................................. 133
5.6 Conclusion ........................................................................................................................................ 137

Chapter 6. Conclusions And Suggestions For Further Research

6.1 Conclusions ....................................................................................................................................... 139
6.2 Suggestions for Further Research ...................................................................................................... 142
   6.2.1 Orthogonal Analysis Based on Other Adaptive Algorithms ..................................................... 142
   6.2.2 Finite Word Length Effect on LMS-based Discrete Orthogonal Transforms ........................... 143
   6.2.3 Finite Word Length Effect Studies of Running Discrete Orthogonal Transforms Based on Shift Properties ................................ ......................................................... 143
   6.2.4 Improvement of The Time Domain Interpolation Algorithm ................................................... 144
   6.2.5 Other Potential Applications of The Time Domain Interpolation Algorithm .......................... 144

Appendices

Appendix A The second Order Shift Properties of DCT-I and DST-I ................................................. 146
Appendix B The second Order Shift Properties of DCT-III and DST-III ............................................. 149
Appendix C The second Order Shift Properties of DCT-IV and DST-IV ............................................ 152
Appendix D Shift Properties of DWT-II and DWT-II ............................................................................ 154
Appendix E Shift Properties of DWT-III and DWT-III ........................................... 158
Appendix F Shift Properties of DWT-IV and DWT-IV ........................................... 162

References.................................................................................................................. 166
List of Symbols

\( A_m, B_m, C_m, D_m, E_m, F_m \)

variable coefficients

\( \alpha, \alpha_{m,n}, \alpha'_{m,n} \)

variable coefficients

\( \beta_m(n), \beta'_m(n), \gamma_m(n), \gamma'_m(n) \)

variable coefficients

\( a_m(j), b_m(j) \)

coefficients of IIR filter

\( C_I(j, m) \)

mth component of running DCT-I

\( c_I(n, m) \)

kernels of DCT-I

\( \hat{C}_I(z, m) \)

z transform of \( C_I(j, m) \)

\( C_{II}(j, m) \)

mth component of running DCT-II

\( c_{II}(n, m) \)

kernels of DCT-II

\( \hat{C}_{II}(z, m) \)

z transform of \( C_{II}(j, m) \)

\( C_{III}(j, m) \)

the \( m \)th component of running DCT-III

\( c_{III}(n, m) \)

kernels of DCT-III

\( \hat{C}_{III}(z, m) \)

z transform of \( C_{III}(j, m) \)

\( C_{IV}(j, m) \)

the \( m \)th component of running DCT-IV

\( c_{IV}(n, m) \)

kernels of DCT-IV

\( \hat{C}_{IV}(z, m) \)

z transform of \( C_{IV}(j, m) \)

\( d(t), d(j) \)

signals to be analysed

\( d_m(j) \)

the \( m \)th harmonic component of \( d(j) \)

\( D(j, m) \)

the \( m \)th component of the running transform of \( d(j) \)

\( \hat{D}(j) \)

running transform vector of

\( \hat{D}(z) \)

z transform of \( d(j) \)

\( e(n), e'(n), e_m(n), e'_m(n) \)

error sequences

\( E(k), E'(k), E_m(k), E'_m(k) \)

The DFT of \( e(n), e'(n), e_m(n), e'_m(n) \)

\( \varepsilon \)

deviation between actual and ideal sampling period
Fr(j,m) Running DFT of d(j)

G_m, H_m, O_m, P_m, Q_m variable coefficients

H(j,m) the mth component of running DHT

h(n,m) kernels of DHT

\( \hat{H}(z, m) \) z transform of H(j,m)

I(t), I(n) Current signal in electrical power measurement

i Square root of -1 (that is, \( i = \sqrt{-1} \))

j time index

k index in transformed domain

\( \Delta \) Coefficient vector of orthogonal representation of signals

L the number of signal cycles with the observation interval

\( \lambda \) the step size of the LMS algorithm

M Number of harmonics

m index in transformed domain

N data block length

n time index

P(n), P'(n) the instantaneous electrical power

P, P', P_0 Electrical power

R auto-correlation matrix

L cross-correlation vector

s(j) sinusoidal signal

S_I(j,m) the mth component of running DST-I

s_I(n,m) kernels of DST-I

\( \hat{S}_I(z, m) \) z transform of \( S_I(j, m) \)

S_{II}(j,m) the mth component of running DST-II

s_{II}(n,m) kernels of DST-II
$\hat{S}_n(z, m)$
z transform of $S_n(j, m)$

$S_{ni}(j, m)$
the $m$th component of running DST-III

$s_{ni}(n, m)$
kernels of DST-III

$\hat{S}_{ni}(z, m)$
z transform of $S_{ni}(j, m)$

$S_{nv}(j, m)$
the $m$th component of running DST-IV

$s_{nv}(n, m)$
kernels of DST-IV

$\hat{S}_{nv}(z, m)$
z transform of $S_{nv}(j, m)$

$T_0, f_0$
Period and frequency of fundamental component

$T_s, f_s$
sampling period and frequency

$T_{so}, f_{so}$
ideal sampling period and frequency

$U(t), U(n)$
Voltage signal in electrical power measurement

$u_m(j), v_m(j)$
the $m$th element of LMS weight vectors

$W(j)$
the LMS weight vector

$w_m(j)$
the $m$th element of $W(j)$

$W_{e,i}(j, m)$
the $m$th component of running DCWT-I

$w_{e,i}(n, m)$
kernel of DCWT-I

$\hat{W}_{e,i}(z, m)$
z transform of $W_{e,i}(j, m)$

$W_{e,ii}(j, m)$
the $m$th component of running DCWT-II

$w_{e,ii}(n, m)$
kernel of DCWT-II

$\hat{W}_{e,ii}(z, m)$
z transform of $W_{e,ii}(j, m)$

$W_{e,iii}(j, m)$
the $m$th component of running DCWT-III

$w_{e,iii}(n, m)$
kernel of DCWT-III

$\hat{W}_{e,iii}(z, m)$
z transform of $W_{e,iii}(j, m)$

$W_{e,iv}(j, m)$
the $m$th component of running DCWT-IV

$w_{e,iv}(n, m)$
kernel of DCWT-IV

$\hat{W}_{e,iv}(z, m)$
z transform of $W_{e,iv}(j, m)$

$W_{e,iv}(j, m)$
z transform of $W_{e,iv}(j, m)$
$W_{s,1}(j, m)$ the $m$th component of DWT-I

$w_{s,1}(n, m)$ kernels of DWT-I

$\hat{W}_{s,1}(z, m)$ $z$ transform of $W_{s,1}(j, m)$

$W_{s,II}(j, m)$ the $m$th component of DWT-II

$w_{s,II}(n, m)$ kernel of DWT-II

$\hat{W}_{s,II}(z, m)$ $z$ transform of $W_{s,II}(j, m)$

$W_{s,III}(j, m)$ the $m$th component of DWT-III

$w_{s,III}(n, m)$ kernel of DWT-III

$\hat{W}_{s,III}(z, m)$ $z$ transform of $W_{s,III}(j, m)$

$W_{s,IV}(j, m)$ the $m$th component of DWT-IV

$w_{s,IV}(n, m)$ kernel of DWT-IV

$\hat{W}_{s,IV}(z, m)$ $z$ transform of $W_{s,IV}(j, m)$

$X(t), X(j), X_x(j), X_v(j)$ orthogonal signal vector

$x(n, m)$ kernel of orthogonal transforms

$x_a(j, m), x_v(j, m)$ the $m$th components of $X_a(j)$ and $X_v(j)$

$x(j)$ input of adaptive IIR filters

$y(j)$ output of adaptive IIR filters

$z_a(t)$ harmonic signal

$z(n), Z(n)$ actual sample sequence of $z_a(t)$ and its DFT

$z_0(n)$ ideal sample sequence resulting from synchronised sampling

$z'(n)$ modified sample sequence using the pre-processing algorithm

$Z'(k)$ the DFT of $z'(k)$

$z_{a,m}(t)$ the $m$th harmonic component of $z_a(t)$

$z_m(n), Z_m(k)$ the $m$th harmonic component of $z(n)$ and its DFT

$z_{0,m}(n), Z_{0,m}(k)$ the $m$th harmonic component of $z_0(n)$ and its DFT
Chapter 1  Preliminaries
1.1 Introduction

This thesis deals with spectral analysis, which is one of the advanced and specialised subfields of digital signal processing. In particular it focuses on improving the efficiency of the discrete orthogonal transform algorithms and the accuracy of harmonic signal analysis. The work includes performance evaluation of some existing techniques, and development of a number of new techniques.

One of the central themes in signal processing is to extract useful information from signals corrupted by noise. This task often falls within the domain of spectral analysis. Hence, spectral analysis is widely used in such application areas as speech [Owens (1993)], radar [Lewis, Kretschmer and Shelton (1986)] and sonar signals [Nielsen (1986)], communication systems [Betts (1970)], seismic traces [Robinson (1983)], market data [Brooks (1984)], and so on.

Research in spectral analysis has led to variety of techniques. One of the fundamental and most widely used approaches is orthogonal analysis based on discrete orthogonal transforms [Ahmed and Rao (1975)]. The issue of computational efficiency associated with these orthogonal transforms has gained considerable interest by the research community over the last three or four decades.

Periodic signal analysis arises from many application areas, such as electrical power systems, biomedical signal processing, and communication systems. A periodic signal can be modelled as a combination of harmonic components (sinusoids) in frequency domain. Useful information can be carried by the phase, amplitude and frequency of each harmonic component. The objective of periodic signal analysis is to estimate these parameters from the received signals, which are usually corrupted by noise. Spectral analysis has been a major solution for this problem.
The thesis considers two main issues. One deals with improving the efficiency of spectral analysis techniques in terms of both computational burden and hardware implementation. This involves orthogonal analysis based on adaptive filtering, and the running computation of discrete orthogonal transforms. The other issue aims at modifying a number of existing techniques [Perera, et al.(1993), Chicharo and Ng (1990)] in order to obtain more accurate results of harmonic signal analysis. These techniques include adaptive IIR notch filtering and discrete Fourier transform (DFT).

This Chapter is organised as follows: Section 1.2 gives a brief historical review of spectral analysis techniques. The background for the work in the thesis is given in Section 1.3. The approach and contributions of this thesis are described in Section 1.4. Finally, Section 1.5 lists the contributions in point form.

### 1.2 Spectral Analysis: A Brief Historical Review

Spectral analysis has developed in uneven steps over a long period of time. The great Greek mathematician Pythagoras (ca 600 B.C.) was the first to study a physical problem in which spectrum analysis made its appearance — the law of vibrations on a vibrating string. This problem has excited scientists since ancient days. Daniel Bernoulli [Bernoulli (1738), L. Euler [Euler (1755)] and J. L. Lagrange [Lagrange (1759)] discovered that solutions of vibrational problems often entail a combination of sine waves (harmonics). This lead to the establishment of a theoretical connection between vibration and spectrum. In 1822, a great innovation was made by Jean Baptiste Joseph de Fourier [Fourier (1822)], which was later referred to as the Fourier series. It was

---

1 Spectral analysis is a widely used term in many areas of science and technology. The focus of this thesis will be on the spectral analysis of time series. For a detailed review of spectrum analysis, see [Robinson (1982)].
found that an arbitrary function, defined over a finite interval by any rough and even discontinuous graph, could be represented as an infinite summation of cosine and sine functions. Throughout the nineteenth century and into the twentieth century, efforts were made to examine the properties of the Fourier series. It was recognised that any function has a spectrum and its nature can be investigated by spectral analysis. In addition, the Fourier series was then extended to other forms consisting of other orthogonal functions by Charles Sturm [Sturm (1836)] and Joseph Liouville [Liouville (1838)]. As a result various orthogonal functions have been developed, which forms the foundation for today's orthogonal analysis techniques.

The first numerical method of spectral analysis for empirical time series is the periodogram proposed by Sir Arthur Schuster [Schuster (1898)]. This approach calculated the periodogram on the basis of the time series, and the peak of the periodogram provides information about the spectrum. However, this periodogram approach was very erratic, since for many cases the periodogram did not have any dominant peak. To solve the problem, Norbert Wiener [Wiener (1930)] proposed a new method in his classic paper, entitled "Generalised Harmonic Analysis". In this paper Wiener proposed a spectral analysis method for determining the spectral presentation of the stationary random process. He defined the autocovariance function and the power spectrum of a stationary random process and proved that the connection between them is the Fourier transform. The proposed spectral analysis method consists of two steps, first the autocovariance function is computed, and then the power spectrum is determined using the Fourier Transform of the autocovariance function. This approach was widely used until 1949 even though in many cases the power spectra from empirical autocorrelation functions were still too erratic to be of any use in formulating physical hypotheses.
In 1949 a breakthrough for the analysis of short time series was made by J. W. Tukey [Tukey (1949a)]. Not only did Tukey show how to compute power spectra from empirical data, but he also established the statistical framework for the analysis of short-time series. Many well known techniques and terms such as "aliasing," "smoothing and decimation," "pering," "bispectrum," "complex demodulation," and "cepstrum", were also proposed by Turkey [Tukey (1949-1963)]. These methods [Tukey (1949-1963)] have been widely used in many application areas, such as the analysis of oceanographic time series records [Pierson and Tick (1957)], seismic data analysis [Wadsworth, et al. (1953)], economics [Granger (1964)], and many others.

The methods proposed by both Wiener [Wiener (1930)] and Tukey [Tukey (1949-1963)] require the computation of the DFT. However, this was a formidable task at that time, especially when the number of data samples was large. Another breakthrough in time series spectral analysis was seen in 1965 with the discovery of the Fast Fourier transform (FFT) by J.W.Cooley and J.W.Tukey [Cooley and Tukey (1965)]. The FFT provides a much more computationally efficient approach for computing the DFT. The discovery of the FFT led to a rapid development of spectral analysis techniques. On one hand, similar fast algorithms have been developed for other Discrete Orthogonal Transforms, such as the Discrete Cosine Transform (DCT) [Ahmed, et al.(1974)], the Fast Walsh Transform (DWalT) [Shanks (1969)], and so on. While on the other hand, various spectral estimation techniques have been developed which provide better performance than DFT in some practical situations [Childers, ed. (1978), Kesler, ed. (1986)]. The rapid development in spectral analysis techniques can be seen from the expansion of the technical literature in this area (For example, [Childers, ed. (1978)], Proceedings RADC Spectrum Estimation Workshops (1978, 1979, 1981, 1983), Proc. IEEE (no.9, 1982) and Proc. IEE-f (no.3, 1983), [Kesler, ed. (1986)]).
The spectral analysis techniques are classified into either parametric or nonparametric methods [Kesler, ed.(1986)]. Parametric methods are based on the signal modeling or parameter estimation. In other words, the signal to be analysed is assumed to be the output of a system, and the spectrum is obtained by working out the parameters or coefficients of the system [Kesler, ed. (1986)]. Signals are usually modeled by moving average (MA), autoregressive (AR), and autoregressive-moving average (ARMA) processes. A variety of parameter estimation techniques have been developed, which result in various spectral estimation techniques. For example, the maximum entropy method [Burg (1967)], adaptive notch filtering for frequency estimation [Chicharo and Ng (1990, 1992), Chicharo (1989)]. By contrast, nonparametric approaches directly compute the spectrum on the basis of the signal samples. Compared to parametric methods, nonparametric approaches are more direct, though not always simpler, in computing the spectrum. Discrete Fourier transforms and various other discrete orthogonal transforms are the central part of the nonparametric approaches. Also there have been some advanced nonparametric approaches such as the time-frequency distribution techniques and higher order statistical approaches.

1.3 Background To The Thesis

As mentioned in the previous sections spectral analysis is a very broad field and contains many topics of research. The work of this thesis deals with a small subsection of this area. It is appropriate therefore to provide an overview of the relevant topics in order to place the necessary background for the work of this thesis.

1.3.1 Discrete Orthogonal Transforms

Spectral analysis based on discrete orthogonal transforms is by far the most widely used
technique. The first part of this thesis deals with efficient computation of various discrete orthogonal transforms.

1.3.1.1 Fast Orthogonal Transforms

Apart from the Discrete Fourier Transform (DFT), there have been several types of orthogonal transforms such as the Discrete Cosine Transform (DCT), Discrete Sine Transform (DST), Discrete Hartley Transform (DHT), Discrete Walsh Transform (DWalT), Discrete W Transforms (DWT) and so on. Each transform has its own advantages and disadvantages. The selection of appropriate transform depends on the application area.

In transform coding systems for image processing, a block transform is performed on consecutive blocks of data. In these applications the DCT is usually chosen as the transform because its performance is close to the optimal Karhunen-Loève transform. Since the introduction of the DCT in 1974 [Ahmed, et al. (1974)], many fast algorithms for computing the DCT have been proposed. These algorithms are available for computing the DCT for specific blocks of data. For example, the algorithms in references [Chen, et al. (1977), Lee (1984), and Malvar (1986)] are for computing a 2\(^n\)-point DCT, while [Chan and Siu (1992)] is for Radix 3 and Radix 6 DCT.

The Hartley Transform was first published in 1942 [Hartley (1942)] as an alternative formulation of a harmonic functional transform similar to the Fourier identity. Many years later its discrete version, the Discrete Hartley Transform (DHT) was proposed by Bracewell [Bracewell (1983)]. The DHT has some advantages. Firstly, it is symmetric in terms of its forward and its inverse transforms. In other words, the kernel of forward DHT is the same as the kernel of the reverse DHT. Secondly, the Hartley spectrum of a real signal is also real due to its real kernel. For these reasons the DHT has found many
applications, such as in spectral analysis and fast convolution of real data [Meckelburg and Lipka (1985)], and adaptive filtering [Wong and Kwong (1991)]. As in the case for the DFT and DCT, fast algorithms have also been proposed for computing the DHT [Bracewell (1984), Meckelburg and Lipka (1985), Kwong and Shiu (1986), and Hou (1987)].

Walsh sequences have attractive characteristics in the sense that the coefficients values are either 1 or -1. The Discrete Walsh Transform (DWalT) has been efficiently used for spectral filtering [Zarowski and Yunik (1985)] and system analysis [Lewis and Mortzios (1987)]. A fast algorithm for computing DWalT was proposed by Shanks (1969), which is similar to the FFT and can reduce the computational burden. However, it is restricted to the case when the transform length is of power 2.

The discrete W transform (DWT) was developed by Wang [Wang, Z. (1981a, 1981b, 1981c)] as an approach for harmonic analysis. One advantage of the W transforms in harmonic analysis is that they are suitable for cases where the data sequence is composed not only of integer multiples of the fundamental frequency, but also of half-integer multiples that are odd multiples of the fundamental frequency divided by 2. A more efficient nesting DFT algorithm can be implemented by using the DWT [Wang, Z. (1992)]. For these reasons there has been a growing interest in DWT in recent years. There are also various fast algorithms [Wang, Z. (1984, 1985, 1989, 1992)] developed for its computation.

It is clear from the above that the conventional technique for computing Discrete Orthogonal Transforms (DOT) is to use fast algorithms similar to the FFT. When compared to the direct computation of the discrete orthogonal transforms, fast algorithms can significantly reduce the computational burden. However, there are still some difficulties associated with these fast algorithms. For instance, fast algorithms
usually require the data length to be equal to some specific value (for example, to the power 2).

1.3.1.2 Computing Orthogonal Transforms Based On Adaptive Filtering

The implementation of orthogonal analysers using adaptive filtering systems is an issue which has attracted much interest recently. Widrow, et al. [Widrow, et al. (1987)] established the relationship between the LMS adaptive algorithm and the Discrete Fourier Transform (DFT), which resulted in a new method for calculating the running DFT. Subsequent efforts have been directed at generalising Widrow et al's work in basically two directions. One approach has been concentrated on using LMS-based adaptive filters to perform various orthogonal analysis. This has resulted in the LMS-based discrete Hartley analyser proposed by Liu and Lin [Liu and Lin (1987a)], and the LMS-based orthogonal analyser proposed by Wang [Wang, S. S. (1991)]. Another approach [Xi (1990)] investigated the implementation of orthogonal analysers using other adaptive algorithms, such as the Sample Matrix Inversion (SMI) algorithm [Monzingo and Miller (1980)] and adaptive Howells-Applebaum loop [Applebaum (1976)].

1.3.1.3 Running Discrete Orthogonal Transforms

In many situations the computation of discrete transforms is required in real time for a stream of input data samples. Running discrete orthogonal transforms are often used for this purpose. The conventional approach is to use the fast algorithms similar to the FFT [Cooley and Tukey (1965)]. However, these fast algorithms were originally developed for computing block transforms, which never-the-less are still considered expensive in terms of computational burden. By necessity, the development of efficient running transform algorithms has been an active research area during the last few years. One
realisation is to use a bank of filters, whose inputs are the sequence to be transformed and whose outputs are the running discrete transform coefficients [Stuller (1984)]. These filters can be either of Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) in nature.

Another realisation of running orthogonal transforms which is considered in this thesis is based on the shift properties of the discrete transforms. The shift property of a discrete orthogonal transform is a set of recursive equations, which provides the recursive updating of the transform coefficients. The first order shift properties of DCT's and DST's were derived by Yip and Rao [Yip and Rao (1987)]. However, the shift properties of DCT's and DFT's are not efficient in terms of computational burden, since the coefficients of DCT's and DST's are dependent on each other. The process of updating the DCT (or DST) requires updating the corresponding DST (or DCT). To alleviate this problem, Murthy and Swamy [Murthy and Swamy (1992)] proposed alternative approaches for computing the running DCT-II, DST-II, DCT-IV and DST-IV, where each transform was represented as the real part of a complex function. The approach proposed by Murthy and Swamy updated these complex functions rather than the transform coefficients.

Liu and Lin [Liu and Lin (1988b)] proposed a real time DHT analyser, which has the ability to calculate the running reversed-order DHT. For the properly ordered running DHT, a lattice structure was proposed by Liu and Chiu (1993). By separating the DHT kernel into the summation of DCT-I like and DST-I like kernels, Liu and Chiu (1993)

---

2 The running reversed order DHT is defined as the DHT for signal block \([x(n), x(n-1), \ldots, x(n-N+1)]\), in which the order of samples in the signal block is reversed when compared to the properly ordered running DHT (See Equation (3.40)).
proposed a lattice structure, which dually generates DCT-I and DST-I, and the DHT is given by the summation of DCT-I and DST-I.

1.3.2 Harmonic Signal Analysis

For the cases where the signal to be analysed is periodic, the spectrum analysis problem is simplified to harmonic signal analysis. The second part of this thesis deals with harmonic signal analysis. The concept of harmonics comes from the Fourier series model of periodic signals. A harmonic signal consists of a combination of sinusoids, referred to as harmonic components, whose frequencies are integer multiples of a fundamental frequency. Each harmonic component is determined by three parameters: amplitude, frequency and phase.

The problem of analysing harmonic signals buried in noise has received a great deal of attention during the last few decades. This problem is a multifaceted one in that sometimes one parameter (such as frequency) of a harmonics is of interest, sometimes all parameters (frequencies, amplitudes and phases) are of interest. This thesis will consider two issues as described in the following subsections.

1.3.2.1 Harmonic Signal Analysis Using The DFT

The first topic in harmonic analysis which is considered in this thesis is the DFT. The DFT is one of the most widely used approaches in this area. There are generally three steps for performing the DFT analysis. Firstly, the signals are uniformly sampled and converted into discrete sequences. Then a block of data is constructed by looking at the actual signal for a finite period of time (referred to as the data window or the observation interval) and neglecting everything that happens before or after this period. Finally the DFT is computed on the basis of the data within the data window to get the
resulting spectrum. There are a number of requirements associated with these three procedures in order to achieve accurate results. First of all, the sampling frequency must be greater than or equal to the Nyquist frequency, which is twice of the highest frequency of interest. Secondly the sampling procedure must be synchronised with the signal to be analysed. There are two aspects associated with synchronisation. One is that the data window must cover a integer number of signal cycles. The other aspect requires that the length of the data window is also an integer multiple of the sampling period. There are some pitfalls associated with the use of the DFT in harmonic analysis. A particular problem arises when the sampling frequency is high enough to satisfy the Nyquist theorem but the sampling process is not synchronised with the signal. In this situation spectral leakage occurs and degrades the accuracy of DFT harmonic signal analysis [Girgis and Ham (1980), Perera, et.al (1993)]. Similar problems also exist in digital Wattmeters [Turgel (1974), Matouka (1982), and Srinivasan (1987)] whose accuracy suffers from truncation errors if the sampling procedure is not synchronised. Unfortunately, in practical situations it is often difficult for the sampling procedure to be exactly synchronised with the input signal. This is because both the sampling and signal frequency may vary with time due to many factors, such as oscillator variations and noisy signal environment.

Spectral leakage refers to the apparent spreading of energy from one frequency bin into adjacent ones. This effect is inherent in DFT for any finite block of data. It arises due to the truncation of time sequence, that is, only a fraction of cycle exists in the block of data that is subjected to the DFT. The leakage can be classified in terms of the range of the spill in frequency domain. The spectral leakage into most adjacent frequency bins is called short-range leakage. The spectrum spilled into far frequency bins is referred to as the long-range leakage.
There are two main conventional approaches for reducing the leakage in harmonic analysis. One method is through the use of time windows [Harris (1978)]. The other approach is based on interpolation between the DFT bins [Jain, et al (1979)]. These two approaches can be combined together [Grandke (1983)] to provide better reduction for both short-range and long-range leakage effects [Audria (1989)].

1.3.2.2 Sinusoidal Frequency Estimation Based On Adaptive IIR Notch Filtering

The second topic on harmonic signal analysis which is investigated in this thesis is harmonic frequency estimation. In particular, it considers the task of frequency estimation of harmonic signals buried in noise using adaptive Infinite Impulse Response (IIR) notch filtering techniques.

Adaptive notch filtering is an alternative approach for harmonic frequency estimation [Chicharo and Ng (1990, 1992)]. There are essentially two types of adaptive notch filters, Finite Impulse Response (FIR) and IIR adaptive notch filters [Shynk (1989)]. Compared to the FIR filters, the IIR filter is characterised by lower computational complexity.

There have been several types of adaptive IIR notch filter structures proposed in literature. The most recent one is the minimal parameter adaptive IIR notch filter proposed independently by Nehorai (1985) and Ng (1985, 1987), where $n$ notches can be characterised by $n$ parameters. This structure was later studied in more detail [Stoica and Nehorai (1988), Ng (1987), and Chicharo and Ng (1990)]. It is this structure that is considered in this thesis.
Adaptive Gradient-based (GB) algorithms are used extensively in adaptive IIR filtering [Shynk (1989)]. GB algorithms adjust filter coefficients at each instant of time to minimise the instantaneous square error. However, the gradient of the error (or output) can not be exactly evaluated due to the recursive nature of the filters. This makes it difficult to derive the desired gradient-based algorithms. For this reason an approximate gradient is used in practice. Several types of gradient-based algorithms were proposed which use different approximations, such as the Recursive Predict Error (RPE) algorithm [Ljung (1981)], simplified RPE algorithm [Shynk (1989)], and the Approximate Gradient (AG) algorithm (or peseudolinear regression algorithm) [Feintuch (1976)]. It has been shown that the more accurate the gradients, the better the steady state performance [Johnson (1984), Chicharo (1989)]. The RPE algorithm is the best in terms of steady state performance. This is due to the fact that the RPE algorithm uses more accurate estimate of the gradients than both the simplified RPE algorithm and AG algorithms. However, the RPE algorithm is computationally more expensive than the other two algorithms [Shynk (1989)].

1.4 Approach And Contribution Of This Thesis

This thesis attempts to improve the efficiency and accuracy of spectral analysis and frequency estimation using orthogonal transforms and adaptive filtering techniques with applications to periodic signal analysis. Specifically, the work in this thesis falls into two main categories. One is to develop some new approaches for computing orthogonal transforms in general which are more efficient in terms of computational burden and hardware implementation. The other is to produce new or improved techniques that improve the accuracy of periodic signal analysis and frequency estimation.
1.4.1 Adaptive Filtering-Based Orthogonal Analysis

Much has been done in the literature on the topic of implementing orthogonal analysers using adaptive filtering systems [Widrow, et al. (1987), Liu and Lin (1988a), Wang, S. S. (1991), Xi (1990)]. However, there are still a number of unresolved issues in this area. For example, LMS-based techniques are for computing block-based discrete orthogonal transforms in most cases except the DFT, where the running transforms can be computed using the LMS algorithm as well [Widrow, et al. (1987)]. An open question is whether it is possible to compute other running discrete transforms based on the LMS algorithm? Secondly, is it possible to implement orthogonal analysers by means of other adaptive algorithms? The motivation of Chapter 2 is to answer these two questions.

Chapter 2 establishes a general relationship between adaptive filtering and orthogonal transforms. A number of approaches for implementing orthogonal analysers are proposed on the basis of this connection. Firstly, the LMS-based discrete orthogonal analyser is studied and some sufficient conditions for computing block-based orthogonal transforms are presented. To illustrate the performance of the technique, an LMS-based discrete Walsh analyser and an LMS-based DCT analyser are investigated. Then new LMS-based running discrete orthogonal transforms are proposed, which include the LMS-based running DHT, DCT and DST, as well as DWT's. Finally the possibility of orthogonal analysis using two different adaptive algorithms is studied. This results in two new orthogonal analyser implementations. One is the discrete orthogonal analyser based on the adaptive Sampling Matrix Inversion (SMI) algorithm [Monzingo and Miller (1980)], while the other one is the spectral analyser based on the adaptive Howells-Applebaum loop [Applebaum (1976)].
1.4.2 Computing Running Discrete Orthogonal Transforms Based on Their Shift Properties

The DFT, including its shift properties, has been studied extensively. Chapter 3 complements this work by focusing on the shift properties of other discrete transforms, including DCT's and DST's, DHT, and DWT's. Two classes of recursive equations are developed for updating the transform coefficients.

One class of recursive equations are referred to as the first order shift properties since they are based on first order difference equations. These first order difference equations provide the connection between the updated transform coefficients at time $j$ and the transform coefficients at time $j-1$, and thus can be used to update the transform coefficients at every sample. Each of these first order difference equations usually involves two transform coefficients of different transforms (for example, a discrete cosine transform and its corresponding discrete sine transform). This means that two transforms must be updated at the same time. This is a source of extra computational burden. For some transforms such as the DHT and DWT's, this extra computational burden can be eliminated by using the reverse symmetrical properties of the transform coefficients. However, for other transforms such as DCT's and DST's, the computation based on the first order shift properties is not very efficient.

The other class of recursive equations are based on second order difference equations. These are referred to as the second order shift properties. The advantage of the second order shift properties is that each second order difference equations involves only one transform coefficient, thus enabling independent updating of every transform coefficient. For the transforms such as DCT's and DST's, which are not reverse symmetrical, the second order shift properties provide a more efficient technique in terms of computational burden. Compared to the approaches proposed by Yip and Rao [Yip and
Rao (1987)] as well as by Murthy and Swamy [Murthy and Swamy (1992)], the computational burden associated with the second order approach is significantly reduced.

### 1.4.3 A Time Domain Interpolation Approach For Harmonic Analysis Using The DFT

As discussed earlier, in order to avoid leakage in DFT analysis, there should be an exact integer number of cycles for each frequency component of an input signal within the data window. Chapter 4 presents a new approach for alleviating the problem of leakage. The basic idea of the proposed method is to modify the practical sample sequence towards an ideal sample sequence which in essence is synchronised with the signal. A simple formula for modifying the practical samples is derived on the basis of an interpolation approach. To illustrate the application of the proposed algorithm, two examples are considered; DFT analysis of harmonic signals and digital wattmeter measurement of electrical power. It is shown that the algorithm can reduce both the DFT leakage and the truncation error of the wattmeter. The proposed approach is continuously on-line and only modifies the sequence when synchronisation is lost. If the samples are properly synchronised with fundamental component, the algorithm has no effect.

### 1.4.4 Frequency Estimation Based On Adaptive IIR Notch Filtering

Frequency estimation techniques based on adaptive IIR notch filtering are considered in Chapter 5. The objective is to improve the steady state error performance.

Gradient-based (GB) algorithms have been widely used in the area of adaptive IIR filtering [Shynk (1989)]. The main reason for the popularity of GB algorithms is that
they are computationally simple to implement. This is particularly so in the case of Approximate Gradient-Based (AGB) algorithms which ignore the recursive gradient components [Chicharo and Ng (1990)]. Alternatively, the Recursive Prediction Error (RPE) algorithm is better than the AGB algorithm in terms of steady state error performance. This is because the RPE algorithm employs more accurate estimates of the gradients than the AGB algorithm [Shynk (1989)]. However, the RPE algorithm is computationally more expensive than the AGB algorithm. Because of the recursive nature of the IIR filtering, such gradient estimates are difficult to determine. In an attempt to remedy this problem, Chapter 5 presents a new adaptive procedure which provides more accurate estimates of the gradients. The result is a new block gradient-based adaptive algorithm. The input signal is divided into data blocks and the coefficients are kept constant within every block. For each block of data the gradients are evaluated and subsequently used to update the coefficients for the next block. In other words, the proposed algorithm updates the coefficients on a block by block basis. By contrast with the conventional Recursive Prediction Error (RPE) algorithm, the proposed approach is characterised by a reduction in steady state error and computational complexity, although the convergence rate is somewhat lower.

1.5 Summary of Contributions in Order of Presentation

- A general relationship is developed between adaptive filtering and orthogonal transforms, which is the theoretical foundation for implementing adaptive filtering based orthogonal transforms.
- Sufficient conditions are derived for implementing (1) block orthogonal transform and (2) running orthogonal transform based on adaptive LMS algorithm.
- The performance of LMS-based block DCT is investigated.
- The performance of the LMS-based DWalT is examined.
- LMS-based running DCT-II and DST-II are proposed.
• The LMS-based running DHT and Discrete W Transforms (DWT's) are proposed and analysed.

• An orthogonal analysis system based on adaptive Howells-Applebaum loop is proposed.

• The SMI-based discrete orthogonal transforms are proposed and analysed.

• The first order shift properties are derived for the DHT and DWT's.

• The second order shift properties are derived for the DCT's and DST's.

• Two structures for implementing the first and second order shift properties are proposed, which result in an architecture of real time orthogonal analysers.

• A time domain interpolation algorithm is derived for modifying the samples when synchronised sampling is desired.

• In order to reduce the spectrum leakage, the proposed time domain interpolation algorithm is applied to the signal sequence subject to the DFT. The performance is investigated by means of theoretical analysis and computer simulations.

• A new algorithm for improving the accuracy of digital wattmeters is proposed on the basis of the proposed time domain interpolation algorithm.

• A new block gradient based algorithm for adaptive IIR filtering is proposed.

• Sinusoidal frequency estimation using adaptive notch filter based on the proposed block gradient algorithm is investigated and its performance analysed.
Chapter 2 Orthogonal Analysis Based On Adaptive Filtering Techniques
2.1 Introduction

This chapter is concerned with the improvement in efficiency of discrete orthogonal transform algorithms. Two aspects of this issue are considered in this thesis. Firstly, we examine the computation of orthogonal transforms through the use of adaptive filtering techniques. Secondly, we consider running discrete orthogonal transforms based on their shift properties. The first aspect forms the main focus of this chapter and can be regarded as an extension or generalisation of the work by Widrow, et al. (1987), Lin and Liu (1988a), Xi (1991) and Wang [Wang, S. S. (1991)]. The second aspect will be considered in detail in Chapter 3.

Adaptive filtering based orthogonal analysis techniques relay on the underlying connection between adaptive filtering and orthogonal transforms. That is, both adaptive filtering and orthogonal transforms are based on least-square fitting criteria. However, this connection is not sufficient for implementing all different types of adaptive filtering based orthogonal analysers. This is because there are various discrete orthogonal transforms as well as different adaptive algorithms with differently characteristics. Hence each case needs to be considered independently. In this chapter, we will consider the following cases:

1. LMS-based techniques for computing block-based discrete orthogonal transforms, including the DCT's and DST's, DWalT's, DHT and DWT's.
2. LMS-based approaches for computing running discrete orthogonal transforms, including the DCT's and DST's, DHT and DWT's.
This chapter is organised as follows: Section 2.2 develops a generalised relationship between orthogonal analysis and adaptive filtering systems. In Section 2.3 the LMS-based discrete orthogonal analysers are investigated, and sufficient conditions for implementing these orthogonal analysers are presented. The performance of the LMS-based DWT and DCT analysers is investigated in Section 2.4. Section 2.5 studies the possibility of computing the running discrete orthogonal transforms using the LMS algorithm. Section 2.6 proposes an orthogonal analyser using two other adaptive algorithms: the SMI algorithm and the Howells-Applebaum loop. Finally, Section 2.7 concludes the chapter.

2.2 A Generalised Relationship Between Orthogonal Analysis And Adaptive Filtering

The purpose of this section is to establish a general connection between orthogonal analysis and adaptive filtering systems, and thus provide a theoretical foundation for implementing orthogonal analysers using adaptive filtering systems.

Consider the case where a time domain signal $d(t)$, for $0 < t < \infty$, is represented by a combination of complete orthogonal sets as follows [Franks (1969)]:

$$d(t) = \Lambda^T X(t)$$

(2.1)

where $X(t)$ is a column signal vector consisting of a complete orthogonal signal set, $\Lambda$ is a column vector representing the coefficients, while the superscript $T$ denotes the transpose. Equation (2.1) is valid provided that for any positive parameter value $\varepsilon$, however small, there exists $\Lambda$ such that
\[ \delta = \int_0^\infty (d(t) - \mathbf{\Lambda}^T \mathbf{X}(t))^2 \, dt < \varepsilon \]  

(2.2)

It can be shown that the coefficient vector which minimises \( \delta \) is:

\[ \mathbf{\Lambda} = \mathbf{R}^{-1} \mathbf{r} \]  

(2.3)

where \( \mathbf{R} = \int_0^\infty \mathbf{X}(t) \mathbf{X}^T(t) \, dt \) and \( \mathbf{r} = \int_0^\infty d(t) \mathbf{X}(t) \, dt \).

Similarly, a discrete signal \( d(j), j=0, 1, 2, ..., N-1, \) can be expressed as a combination of discrete orthogonal signal set:

\[ d(j) = \mathbf{\Lambda}^T \mathbf{X}(j) \]  

(2.4)

where \( \mathbf{X}(j) = [x(j,0) \ x(j,1) \ ... \ x(j,N-1)] \). Equation (2.4) is valid provided the following square error is zero:

\[ \delta = \sum_{n=1}^{N} [d(n) - \mathbf{\Lambda}^T \mathbf{X}(n)]^2 \]  

(2.5)

Clearly orthogonal analysis results in a decomposition of the signal into a weighted sum of \( \mathbf{X}(n) \) components with minimal square error. The optimal coefficient vector \( \mathbf{\Lambda} \) which minimises \( \delta \) is determined as follows:

\[ \mathbf{\Lambda} = \left[ \sum_{n=0}^{N-1} \mathbf{X}(n) \mathbf{X}^T(n) \right]^{-1} \sum_{n=0}^{N-1} d(n) \mathbf{X}(n) \]  

(2.6)

Equation (2.6) represents a general definition for various kinds of discrete orthogonal transforms.
Adaptive filtering algorithms can be used to determine the optimal coefficients in Equation (2.6). This is because adaptive filtering algorithms tend to also minimise a kind of the square error. For example, the LMS algorithm is based on minimising the Mean-Square-Error (MSE), the Least Square (LS) algorithm on minimising the time average of the square error, and so on. Figure 2.1 shows the adaptive filtering algorithm for finding the optimal coefficients in Equation (2.6). Note that the system depicted in Figure 2.1 is very similar to the LMS based spectral analyser proposed by Widrow, et al. (1987). However, in this case it is based on a general orthogonal analysis using adaptive algorithms rather than being restricted to the case of LMS algorithm and DFT analyser. In other words, Figure 2.1 shows a general relationship between adaptive filtering and orthogonal transforms.

The system depicted in Figure 2.1 can be considered as a generalised adaptive filtering based orthogonal analyser. The signal to be analysed serves as the desired input and the orthogonal signal vector $X(t)$ (or $X(j)$) serves as the input signals. The square error is minimised by using a suitable adaptive processing algorithm to adjust the coefficients ($w_l$, $l = 1, 2, ..., N-1$). The orthogonal transform of $d(t)$ is obtained when the coefficients ($w_l$) converge to the optimal values.

![Figure 2.1 Adaptive filtering based orthogonal analyser](image-url)
2.3 Sufficient Conditions for Implementing LMS-based Block Orthogonal Analysers

Consider the adaptive system depicted in Figure 2.2, when the LMS algorithm is employed to control the weights. The signal to be transformed is denoted as $d(j)$, and the input signal vector $X(j)$ is a set of orthogonal sequences defined by:

$$X(j) = [x(j,0) \quad x(j,1) \quad x(j,2) \ldots \quad x(j,N-1)]^T$$

(2.7)

where, as before, the superscript $T$ denotes the transpose and $N$ is the block length. For simplicity, we can express a discrete orthogonal transform of the block signal $d(n)$, where $n = j-N+1, j-N+2, \ldots, j$, in vector form as follows:

$$D(j) = [D(j,0) \quad D(j,1) \quad D(j,2) \ldots \quad D(j,N-1)]^T$$

(2.8)

Using the orthogonal property we have $\sum_{n=0}^{N-1} X(n)X^T(n) = I$, where $I$ is the identity matrix. Using Equation (2.6) we can rewrite an orthogonal transform as follows:

$$D(j) = \sum_{n=0}^{N-1} d(j - N + 1 + n)X(n)$$

(2.9)

The weights are controlled by the adaptive LMS algorithm [Widrow, et al. (1965)]:

$$W(j + 1) = W(j) + 2\lambda e(j)X(j)$$

(2.10)

where $W(j) = [w_0(j) \quad w_1(j) \ldots \quad w_{N-1}(j)]^T$ and $\lambda$ is the step size. The output error signal is given by $e(j) = d(j) - y(j)$, where $y(j) = W^T(j)X(j)$. Substituting these
expressions into Equation (2.10) leads to the following recursive expression for \( W(j+1) \):

\[
W(j+1) = \left[ I - 2\lambda X(j)X^T(j) \right] W(j) + 2\lambda d(j)X(j)
\]  
(2.11)

Consider the case where the initial weight vector is set to zero. Clearly under such circumstances Equation (2.11) becomes:

\[
W(1) = 2\lambda d(0)X(0)
\]  
(2.12)

For the next subsequent step, when \( j=1 \), we obtain:

\[
W(2) = 2\lambda [d(0)X(0) + d(1)X(1)] - 4\lambda^2 X(1)[X^T(1)X(0)]d(0)
\]  
(2.13)

![Diagram](image)

**Figure 2.2** LMS-based discrete orthogonal analyser

In order to obtain the connection between weight vector and the transform in Equation (2.9) we want the product term \( X^T(1)X(0) \) in Equation (2.13) to be zero, that is:
Equation (2.14) is satisfied provided the orthogonal kernel is unitary. In this case, Equation (2.13) can be simplified to yield the following:

\[ W(2) = 2\lambda \left[ d(0)X(0) + d(1)X(1) \right] \]  \hspace{1cm} (2.15)

Assuming that the above sufficient conditions apply, it can be shown that the general expression for \( W(j) \) is as follows:

\[ W(j) = 2\lambda \sum_{n=0}^{j-1} d(n)X(n), \hspace{0.5cm} j = 1, 2, ..., N \]  \hspace{1cm} (2.16)

Note that for the case when \( j = N \), Equation (2.16) becomes

\[ W(N) = 2\lambda \sum_{n=0}^{N-1} d(n)X(n) \]  \hspace{1cm} (2.17)

Clearly this is a direct relationship between \( W(N) \) and \( D(N-1) \). In other words, \( W(N) \) is proportional to \( D(N-1) \).

Now let us examine the weight vector for \( j > N \). From Equation (2.11) and (2.17) we have

\[ W(N+1) = \left[ I - 2\lambda \mathbf{X}(N) \mathbf{X}^T(N) \right] W(N) + 2\lambda d(N)X(N) \]
\[ W(N + 1) = 2\lambda \sum_{n=0}^{N} d(n)X(n) - 4\lambda^2 X(0)X(N) \]

\[ = 2\lambda \sum_{n=1}^{N} d(n)X(n) + 2\lambda(1 - 2\lambda)X(0)X(N) \]

\[ = 2\lambda \sum_{n=2}^{N+1} d(n)X(n) + 2\lambda(1 - 2\lambda)[d(1)X(1) + d(0)X(0)] \quad \text{ (2.21)} \]

This result can be generalised for \( j = N+1, N+2, \ldots, 2N \) as follows:

\[ W(j) = 2\lambda \sum_{n=j-N}^{j-1} d(n)X(n) + 2\lambda(1 - 2\lambda) \sum_{n=0}^{j-N-1} d(n)X(n) \quad \text{ (2.22)} \]
As discussed previously a general formula for the weight vector $W(j)$ can be derived which is applicable over all $j > 1$, that is:

$$W(j) = 2\lambda \sum_{n=j-N}^{j-1} d(n)X(n) + 2\lambda(1-2\lambda) \sum_{n=j-2N}^{j-N-1} d(n)X(n) +$$

$$+ 2\lambda(1-2\lambda) \sum_{n=j-3N}^{j-2N-1} d(n)X(n) ...$$  \hspace{1cm} (2.23)

Note that when using Equation (2.23) the valid ranges for the index $n$ at each summation are set by the upper and lower limits unless these limits are negative. For example, for $0 < j < N$ only the first term in Equation (2.23) exists, all the other terms are zero, while the first term is just the same as in Equation (2.18). Setting the step size to be $\lambda = \frac{1}{2}$, Equation (2.23) can be simplified as follows:

$$W(j) = \sum_{n=j-N}^{j-1} d(n)X(n)$$  \hspace{1cm} (2.24)

Let $j$ be any integer multiple of $N$, i.e., $j=mN$. The weight vector is:

$$W(mN) = \sum_{n=mN-N}^{mN-1} d(n)X(n) = \sum_{n=0}^{N-1} d(mN - N + n)X(mN - N + n)$$  \hspace{1cm} (2.25)

Using the periodic property $X(mN - N + n) = X(n)$, Equation (2.25) can be rewritten as:

$$W(mN) = \sum_{n=0}^{N-1} d(mN - N + n)X(n)$$  \hspace{1cm} (2.26)
From Equation (2.26) and (2.9) we obtain the result:

$$D(mN-1) = W(mN)$$  \hspace{1cm} (2.27)

Hence from the above analysis it is clear that, at the time $j=mN$, the weight vector for the adaptive system is equal to the discrete orthogonal transform of the block signal $d(n)$ for $n = mN-N, mN-N+1, ..., mN-1$.

It is also interesting to consider the case when the step size is set at some value other than $\lambda = \frac{1}{2}$. From Equation (2.23) it can be shown that the output of the system at the time $mN$ will be:

$$D_{out} = 2\lambda D((m-1)N,mN-1) + 2\lambda(1-2\lambda)D((m-2)N,(m-1)N-1) +$$
$$+ 2\lambda(1-2\lambda)^2 D((m-3)N,(m-2)N-1) + \ldots + 2\lambda(1-2\lambda)^{m-1} D(0,N-1)$$  \hspace{1cm} (2.28)

The resulting output is the geometrically weighted coherent sum of the transform coefficients for the past signal blocks. This is consistent with the work by Widrow, et al. (1987) dealing with the relationship between the LMS and the DFT.

Now we investigate the general conditions of using this LMS based spectral analyser to the other orthogonal transforms. It is important to note that the above derivation depends on the fact that the transform under consideration has a unitary kernel, that is:

$$\sum_{j=0}^{N-1} x(r,j)x(s,j) = \sum_{j=0}^{N-1} x(j,r)x(j,s) = \begin{cases} 1 & \text{for } r = s \\ 0 & \text{for } r \neq s \end{cases}$$  \hspace{1cm} (2.29)
We can examine different discrete transforms using this condition. The kernels of most discrete transforms, such as DFT, DWalT, DCT's, DST's, DHT and DWT's are unitary and so an LMS based spectral analyser can be implemented for these transforms.

It is also worthwhile investigating the required conditions for implementing a running type spectral analyser, which updates the results of orthogonal transforms at every data sample update. Such a steady flow type spectral analyser can be achieved if the kernel of orthogonal transforms is rotatable, that is:

\[ x(n_1 + n_2, k) = x(n_1, k)x(n_2, k) \quad \text{for} \quad k = 0, 1, ..., N-1 \]  

Equation (2.30) can be rewritten in vector form:

\[ X(n_1 + n_2) = P(n_1)X(n_2) \]  

where \( P(n_1) = \text{diag} \{ x(n_1,0), x(n_1,1), ..., x(n_1,N-1) \} \). In this case Equation (2.24) can be rewritten as:

\[ W(j) = \sum_{n=j-N}^{j-1} d(n)X(n) = \sum_{n=0}^{N-1} d(n+j-N)X(n+j-N) \]

\[ = \sum_{n=0}^{N-1} d(n+j-N)P(j-N)X(n) = P(j) \sum_{n=0}^{N-1} d(n+j-N)X(n) \]  

therefore we have:

\[ D(j-1) = P^{-1}W(j) \]

\[ = \begin{bmatrix} x^{-1}(j,0)w_0(j) & x^{-1}(j,1)w_1(j) & ... & x^{-1}(j,N-1)w_{N-1}(j) \end{bmatrix} \]  

(2.33)
Hence a steady data flow analyser can be obtained by multiplying $P^{-1}(j)$ by the weight vector $W(j)$ at any time instant. Examination of existing orthogonal transforms, however, reveals that only the kernel of DFT is found to be rotatable, and hence a steady flow LMS-DFT can be implemented [Widrow, et al. (1987)]. The kernels of all the other existing discrete transforms such as DWT, DHT and DCT are not rotatable, and so the steady flow structure can not be directly constructed for these transforms.

2.4 Performance of LMS-Based Block Orthogonal Transforms

As mentioned in Section 2.3 the LMS-based orthogonal analyser can be used to compute most of the existing discrete transforms. The objective of this section is to investigate the efficiency of this technique in terms of the computational burden and hardware implementation. As examples, we consider discrete Walsh transforms (DWalT) and Discrete Cosine Transforms (DCT) in this section.

2.4.1 LMS Based Discrete Walsh Transform

The Discrete Walsh Transform (DWalT) [Beauchamp (1975)] of a block signal $d(n)$ for $n = j-N+1, j-N+2, \ldots, j$ is defined as follows:

$$\text{Wal}(j,m) = \sum_{n=0}^{N-1} d(j - N + 1 + n) wal(n,m) \quad \text{for } m = 0, 1, 2, \ldots, N-1$$  \hspace{1cm} (2.34)

where $wal(n,m)$ are the Walsh sequences, given as follows [Ahmed and Rao (1975)]:

$$wal(n,m) = \sqrt{\frac{1}{N}} (-1)^{(n,m)}$$  \hspace{1cm} (2.35)
in which \( \langle n, m \rangle = \sum_{s=0}^{k-1} n_s m_s \), where \( n_s \) and \( m_s \) are the coefficients of the binary representations of \( n \) and \( m \), while \( k \) is the length of the binary representation of \( N \), given as \( k = \log_2 N \). It can be shown that the kernels of discrete Walsh transform in Equation (2.34) is orthogonal and unitary, which means that the systems depicted in Figure 2.1 can be used as LMS based DWalT analyser. The input signal vector is given as follows:

\[
X(n) = \begin{bmatrix} x(n,0) & x(n,1) & \ldots & x(n,N-1) \end{bmatrix}^T
\]  

(2.36)

where \( x(n,m) \) is the periodic extension of the transform kernel \( \text{wal}(n,m) \).

Using a similar mathematical derivation as presented in Section 2.3, the following result is obtained:

\[
\text{Wal}(mN-1) = W(mN)
\]  

(2.37)

This means that, at the time \( j=mN \), the weight vector of the adaptive system is equal to the DWalT of the block signal \( d(n) \) for \( n = mN-N, mN-N+1, \ldots, mN-1 \). Therefore the system in Figure 2.1 can be used for computing the DWalT. This approach is regarded as an LMS based DWalT (LMS-DWalT).

Now let us to compare the computational burden associated with the proposed LMS-DWalT, the fast algorithm [Shanks (1969)] and the direct computation. The results of comparison are depicted in Table 2.1. Note that the LMS-DWalT needs only \( 2N \) one-bit multiplications and \( 2N \) additions at the end of every signal block. This is because each new sample requires only one adaptation. Clearly from Table 2.1 the proposed LMS-DWalT is more efficient than the fast DWalT [Shanks (1969)] as well as the direction computation. In addition, the proposed LMS-DWalT is more flexible than the
fast DWaT [Shanks (1969)] as far as the signal block length is concerned. The fast DWaT usually requires that the transform length \( N \) is equal to an integer power of 2, while the LMS-DWaT can be implemented for any block length.

<table>
<thead>
<tr>
<th>TABLE 2.1 The comparison of computational burden</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direction computation by definition</td>
</tr>
<tr>
<td>Algorithm by Shanks (1969)</td>
</tr>
<tr>
<td>LMS-DWaT</td>
</tr>
</tbody>
</table>

2.4.2 LMS-Based Discrete Cosine Transform

This section investigates the performance of an LMS-based DCT analyser. A number of different DCT definitions have been proposed in literature [Wang, Z. (1984)]. The most popular one is the DCT-II, which was first defined by Ahmed, et al. (1974) as follows:

\[
c_{mn}(n,m) = \sqrt{\frac{2}{N}} P_m \cos\left(\frac{(n + \frac{1}{2}) m \pi}{N}\right), \quad \text{for } n, m = 0, 1, ..., N-1 \tag{2.38}
\]

where

\[
P_m = \begin{cases} 
1 & \text{if } m \neq 0, N \\
\frac{1}{\sqrt{2}} & \text{if } m = 0, N 
\end{cases} \tag{2.39}
\]

It can be verified that the above kernels satisfy Equation (2.29). In other words, they are orthogonal and unitary. Hence the system in Figure 2.2 can be used to implement the DCT. However, we also need to consider the input signal vectors for the LMS-based DCT. As mentioned previously, the input signal vector should be periodic, that is,
\[ X(n + lN) = X(n) \] (2.40)

From Equation (2.7) we note that the elements of \( X(n) \) are the transforms kernels. For the periodic condition to be satisfied, \( X(n) \) should be constructed as follows:

\[ x(n + lN, m) = c_n(n, m), \text{ for } l=0, 1, 2 \ldots \text{ and } n=0, 1, ..., N-1 \] (2.41)

Let us now examine the efficiency of the LMS-DCT. First of all it is important to note that the fast DCT algorithms proposed by Chen, et al. (1977), Lee (1984), and Malvar (1986) need the entire data block before execution can take place. By contrast, the proposed LMS-DCT distributes the computational burden at every sample update where only one adaptation is required. In other words, only \( 2N \) multiplications and \( 2N \) additions need to be performed during each sampling period, thus minimising the delay at the end of the data block. For illustration purposes, a comparison between the computational burden required at the end of each block for the proposed LMS-DCT and an alternative fast algorithm [Lee (1984)] is depicted in Table 2.2. It is evident that the number of multiplications and additions for the proposed LMS-DCT is higher than the other fast algorithms for small data blocks. However, as the block length increases the LMS-DCT becomes more efficient.

<table>
<thead>
<tr>
<th>Algorithm by Lee (1984)</th>
<th>Number of Multiplications</th>
<th>Number of Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>((N/2)\log_2 N)</td>
<td>((3/2)N\log_2 N - N + 1)</td>
<td></td>
</tr>
<tr>
<td>LMS-DCT</td>
<td>(2N)</td>
<td>(2N)</td>
</tr>
</tbody>
</table>
2.4.3 Other LMS-Based Discrete Orthogonal Transforms

There are also some other discrete orthogonal transforms as defined in literature including: the four different definitions of DCT’s and DST’s [Wang, Z. (1984)], the Discrete Hartely Transform (DHT) [Bracewell (1983)], and Discrete W transforms [Wang, Z,(1981a), (1981b) and (1981c)]. The kernels of each of these transforms are given respectively as follows:

**DCT-I:** \( c_i(n,m) = \sqrt{\frac{2}{N}} P_m P_n \cos\left(\frac{n\pi}{N}m\right), \)

\[ n, m = 0, 1, ..., N \]  

(2.42)

**DCT-II:** \( c_{ii}(n,m) = \sqrt{\frac{2}{N}} P_m \cos\left(\frac{n + \frac{1}{2}}{N}m\right), \)

\[ n, m = 0, 1, ..., N-1 \]  

(2.43)

**DCT-III:** \( c_{iii}(n,m) = \sqrt{\frac{2}{N}} P_m \cos\left(\frac{n(m + \frac{1}{2})\pi}{N}\right), \)

\[ n, m = 0, 1, ..., N-1 \]  

(2.44)

**DCT-IV:** \( c_{iv}(n,m) = \sqrt{\frac{2}{N}} \cos\left(\frac{n + \frac{1}{2}}{N}(m + \frac{1}{2})\pi\right), \)

\[ n, m = 0, 1, ..., N-1 \]  

(2.45)

**DST-I:** \( s_1(n,m) = \sqrt{\frac{2}{N}} \sin\left(\frac{n\pi}{N}m\right), \)

\[ n, m = 1, 2, ..., N \]  

(2.46)

**DST-II:** \( s_{ii}(n,m) = \sqrt{\frac{2}{N}} P_m \sin\left(\frac{n - \frac{1}{2}}{N}m\pi\right), \)

\[ n, m = 1, 2, ..., N \]  

(2.47)
DST-III: \( s_{III}(n,m) = \sqrt{\frac{2}{N}} P_n \sin \left\{ n(m - \frac{1}{2}) \frac{\pi}{N} \right\} \)

\( n, m = 1, 2, ..., N \)

\( = \sqrt{\frac{2}{N}} P_n \sin \left\{ n(m + \frac{1}{2}) \frac{\pi}{N} \right\} \)

\( n, m = 0, 1, ..., N-1 \) (2.48)

DST-IV: \( s_{IV}(n,m) = \sqrt{\frac{2}{N}} \sin \left\{ (n + \frac{1}{2})(m + \frac{1}{2}) \frac{\pi}{N} \right\} \),

\( n, m = 0, 1, ..., N-1 \) (2.49)

Note that in the above Equations \( P_n \) is defined by Equation (2.29).

The kernel of the DHT is given as follows [Bracewell (1983)]:

\[
h(n,m) = \sqrt{\frac{1}{N}} \left[ \cos \frac{2\pi nm}{N} + \sin \frac{2\pi nm}{N} \right]
\]

(2.50)

For the discrete W transforms (DWT), the kernels are given as follows [Wang, Z. (1981a), (1981b) and (1981c)]:

DWT-I: \( w_I(n,m) = \sqrt{\frac{2}{N}} \sin \left( \frac{\pi}{4} + nm \frac{2\pi}{N} \right) \)

\( m, n = 0, 1, ..., N-1 \) (2.51)

DWT-II: \( w_{II}(n,m) = \sqrt{\frac{2}{N}} \sin \left( \frac{\pi}{4} + m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \)

\( m, n = 0, 1, ..., N-1 \) (2.52)
Chapter 2: Orthogonal Analysis based on Adaptive Filtering Techniques

DWT-III: \[ w_{\text{III}}(n,m) = \sqrt{\frac{2}{N}} \sin \left( \frac{\pi}{4} + n\left(m + \frac{1}{2}\right) \frac{2\pi}{N} \right) \]

\[ m, n = 0, 1, ..., N-1 \] (2.53)

DWT-IV: \[ w_{\text{IV}}(n,m) = \sqrt{\frac{2}{N}} \sin \left( \frac{\pi}{4} + (n + \frac{1}{2})(m + \frac{1}{2}) \frac{2\pi}{N} \right) \]

\[ m, n = 0, 1, ..., N-1 \] (2.54)

It can be verified that the kernels of all the above transforms satisfy the required orthogonal and unitary conditions as depicted in Equation (2.29). Hence an LMS based structure can be implemented for these transforms.

2.5 LMS-based Running Orthogonal Transforms

For situations where the signal consists of a stream of data samples, the running computation of discrete transforms is usually desired, and the transform coefficients need to be updated for every sample update. In other words, when a new data sample \( d(j+1) \) becomes available, the signal block of interest shifts one sample in time to include \( d(j+1) \) and exclude the sample \( d(j-N+1) \). However, the general consensus in current literature is that the LMS-based orthogonal analysers are generally limited to implementing block-based discrete transforms. The only exception is the DFT as proposed by Widrow, et al. (1987). The issue of implementing a steady flow or running LMS-based orthogonal analyser has not been investigated to the best of our knowledge for all other transforms. The motivation of the work in this section is to develop techniques for computing running orthogonal transforms based on the LMS algorithm.

2.5.1 LMS-Based Running Discrete Hartley Transform

The DHT for a signal block \([d(j-N+1), d(j-N+2), ..., d(j)]\) is given as follows [Bracewell (1983)]:
\[ H(j,m) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{2n\pi}{N} m \right), \quad \text{for } m = 0, 1, \ldots, N \] (2.55)

where \( N \) is the block length of the transform, and \( \cos(x) + \sin(x) \). For the case when the input signal is a stream of samples, \( H(j,m) \) is referred to as a running DHT. When a new sample \( d(j+1) \) becomes available, the signal block of interest becomes \([d(j-N+2), d(j-N+3), \ldots, d(j+1)]\).

Now consider the case where the input signal vector for system depicted in Figure 2.1 is given as follows:

\[ X_u(n) = [x_u(n,0), x_u(n,1), \ldots, x_u(n,N-1)]^T \] (2.56)

where \( x_u(n,m) = \sqrt{\frac{2}{N}} \cos \left( \frac{2\pi nm}{N} \right) \). It can be verified that \( X_u(n) = X_u(n+N) \), and \( x_u(n,m) \) satisfies Equation (2.29), that is:

\[ \sum_{n=0}^{N-1} x_u(n,r)x_u(n,s) = \sum_{n=0}^{N-1} x_u(r,n)x_u(s,n) = \begin{cases} 1 & \text{for } r = s \\ 0 & \text{for } r \neq s \end{cases} \] (2.57)

From Equation (2.24) the adaptive weights, denoted as \( u_m(j) \), for \( m = 0, 1, \ldots, N-1 \), are given as follows:

\[ u_m(j) = \sum_{n=j-N}^{j-1} d(n)x(n,m), \quad \text{for } m = 0, 1, \ldots, N-1 \] (2.58)

Equation (2.58) can be rearranged as follows:
\[ u_m(j) = \sqrt{\frac{2}{N}} \sum_{n=j-N}^{j-1} d(n) \cos \left[ \frac{2\pi nm}{N} \right] \]

\[ = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(n + j - N) \cos \left[ (n + j - N) \frac{2\pi m}{N} \right] \]

\[ = A_{m,j} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(n + j - N) \cos \left[ \frac{2nm\pi}{N} \right] - B_{m,j} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(n + j - N) \sin \left[ \frac{2nm\pi}{N} \right] \] (2.59)

where \( A_{m,j} = \cos \left( \frac{2jm\pi}{N} \right) \) and \( B_{m,j} = \sin \left( \frac{2jm\pi}{N} \right) \).

Now we consider another case where the input signal vector is given as follows:

\[ x_{v}(n) = [x_v(n,0) \quad x_v(n,1) \quad ... \quad x_v(n,N-1)]^T \] (2.60)

where \( x_v(n,m) = \sqrt{\frac{2}{N}} \sin \left[ \frac{2\pi nm}{N} \right] \). Similarly it can be verified that \( x_v(n,m) \) satisfies Equation (2.29) as well as being inverse periodic in nature. In this case the adaptive weights, denoted as \( v_m(j) \), for \( m = 0, ..., N-1 \), are obtained as follows:

\[ v_m(j) = B_{m,j} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(n + j - N) \cos \left[ \frac{2nm\pi}{N} \right] + A_{m,j} \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(n + j - N) \sin \left[ \frac{2nm\pi}{N} \right] \] (2.61)

From Equations (2.59) and (2.61) the DHT of \( d(n) \) for \( n = j - N, \quad j - N + 1, \quad ..., \quad j - 1 \) can be obtained by:

\[ D(j-1,m) = P_{m,j}u_m(j) + Q_{m,j}v_m(j) \] (2.62)
where $P_{m,j} = \frac{A_{m,j} - B_{m,j}}{2}$ and $Q_{m,j} = \frac{A_{m,j} + B_{m,j}}{2}$. Note that both $P_{m,j}$ and $Q_{m,j}$ are periodic, that is, $P_{m,j} = P_{m,j+N}$ and $Q_{m,j} = Q_{m,j+N}$.

Equation (2.57) can be used to implement a system as depicted in Figure 2.3 for computing the running DHT using the adaptive LMS algorithm. The outputs of the structure shown in Figure 2.3 are the running DHT coefficients for the applied input signal sequence.

Let us consider the efficiency of LMS-DHT in terms of the computational burden when compared with the FFT-like fast algorithms. Note that the FFT-like fast DHT algorithms
need the entire data block before execution can take place. By contrast, the proposed LMS-DCT distributes the computational burden at every sample update where only 2 adaptations are required. A comparison between the computational burden required at the end of each block of data for the proposed LMS-DCT and two fast algorithms [Meckelburg and Lipka (1985), and Sorensen, et al. (1985)] is depicted in Table 2.3. Note that both of the algorithms by Meckelburg and Lipka (1985) and Sorensen, et al. (1985) utilise butterfly computation similar to the FFT algorithm [Cooley and Tukey (1965)]. It is evident that the number of multiplications and additions for the proposed LMS-DHT is higher than the fast algorithms for small transform length. However, as the block length increases the proposed LMS-DHT becomes more efficient.

| Table 2.3 Computational burden of various DHT and LMS-DHT algorithms |
|--------------------------|------------------|------------------|
|                          | Number of Multiplications | Number of Additions |
| Meckelburg and Lipka (1985) | $N(\log_2 N - 1) + 4$ | $(3/2)N(\log_2 N - 1) + 2$ |
| Sorensen, et al. (1985)    | $(2N/3)\log_2 N - 19N/9 + 3$ | $(4N/3)\log_2 N - 14N/9 + 3$ |
| The proposed LMS-DHT       | $6N$             | $5N + 2$         |

### 2.5.2 LMS-Based Running Discrete Cosine/Sine Transforms

As stated previously there are a number of definitions for DCT and DST. Let us firstly consider the DCT-II and DST-II. For the signal block $[d(j - N + 1), d(j - N + 2), \ldots d(j)]$ the DCT-II and DST-II are given as follows [Wang, Z. (1984)]:

\[
\text{DCT-II: } C_n(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} P_m d(j - N + n + 1) \cos \left( (n + \frac{1}{2}) \frac{m\pi}{N} \right),
\]

\[m = 0, 1, \ldots, N-1\]  

\[(2.63)\]
DST-II: \[ S_{II}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} P_m d(j-N+n) \sin \left\{ \left( n - \frac{1}{2} \right) \frac{m \pi}{N} \right\} \]

\[ = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} P_m d(j-N+n) \sin \left\{ \left( n + \frac{1}{2} \right) \frac{m \pi}{N} \right\} \]

\[ m = 1, 2, ..., N \quad (2.64) \]

where \( N \) is the block length of the transform, and \( P_m \) is determined by Equation (2.39).

For the case when the input signal is a stream of samples, \( C_{II}(j,m) \) and \( S_{II}(j,m) \) are referred to as running DCT's and DST's at time \( j \).

Consider the case where the input signal vector for the system depicted in Figure 2.2 consists of the kernels of DCT-II, given as follows:

\[ X_u(n) = \begin{bmatrix} x_u(n,0) & x_u(n,1) & \ldots & x_u(n,N-1) \end{bmatrix}^T, \text{ for } n=0, 1, ..., j \quad (2.65) \]

where

\[ x_u(n,m) = \sqrt{\frac{2}{N}} P_m \cos \left[ m \left( n + \frac{1}{2} \right) \frac{\pi}{N} \right] \quad (2.66) \]

The input signals \( x_u(n,m) \) are constructed by extending the range of the time index \( n \) associated with the transform kernels in Equation (2.43) while keeping the same mathematical expression. Note that the input signals \( x_u(n) \) are different from those for the block based LMS orthogonal analyser in Section 2.4, in which the input signal samples for \( j > N+1 \) are the periodic extension of the first \( N \) samples for \( n = 0, 1, ..., N-1 \).

Clearly from Equation (2.65) \( X_u(n) = -X_u(n+N) \). In other words, \( X_u(n) \) is a reverse periodic vector function. It is necessary to re-examine the behaviour of the adaptive
LMS weights, since the results presented in Section 2.4 are based on the assumption that \( X_u(n) = X_u(n + N) \), which is clearly not valid in this case.

Let us examine the adaptive LMS weights in the situation where \( X_u(n) = -X_u(n + N) \). The LMS algorithm for updating the weights is as follows:

\[
W(j + 1) = [I - 2\lambda X(j)X^T(j)] + 2\lambda d(j)X(j)
\]

(2.67)

Consider the case where the initial weight vector is set to zero. It can be shown [Widrow, et al. (1987)] that:

\[
W(N) = 2\lambda \sum_{n=0}^{N-1} d(n)X(n)
\]

(2.68)

Now let us examine the weight vector for \( j > N \). From Equations (2.67) and (2.68) we have:

\[
W(N + 1) = [I - 2\lambda X(N)X^T(N)]W(N) + 2\lambda d(N)X(N)
\]

\[
= 2\lambda \sum_{n=0}^{N} d(n)X(n) - 4\lambda^2 X(N) \left[ X^T(N) \sum_{n=0}^{N-1} d(n)X(n) \right]
\]

(2.69)

Using the result where \( X^T(0)X(0) = 1 \) and the reverse periodicity of \( X(n) \), that is, \( X(0) = -X(N) \), the second term in Equation (2.69) can be evaluated as:

\[
X(N) \left[ X^T(N) \sum_{n=0}^{N-1} d(n)X(n) \right] = X(N) \left[ X^T(N) \left[ d(0)X(0) + \sum_{n=1}^{N-1} d(n)X(n) \right] \right]
\]

\[
= -X(0) \left\{ -X^T(0) \left[ d(0)X(0) + \sum_{n=1}^{N-1} d(n)X(n) \right] \right\} = X(0)d(0)
\]
Hence Equation (2.69) can be simplified as:

\[
W(N+1) = 2\lambda \sum_{n=0}^{N} d(n)X(n) - 4\lambda^2 d(0)X(N)
\]

\[
= 2\lambda \sum_{n=1}^{N} d(n)X(n) + 2\lambda(1-2\lambda)X(0)d(0)
\]  \hspace{0.5cm} (2.70)

Similarly the vector \(W(N+2)\) can be derived as:

\[
W(N+2) = 2\lambda \sum_{n=2}^{N+1} d(n)X(n) + 2\lambda(1-2\lambda)[d(1)X(1) + d(0)X(0)]
\]  \hspace{0.5cm} (2.71)

This result can be generalised for \(j = N+1, N+2, \ldots, 2N\) as follows:

\[
W(j) = 2\lambda \sum_{n=j-N}^{j-1} d(n)X(n) + 2\lambda(1-2\lambda) \sum_{n=0}^{j-N-1} d(n)X(n)
\]  \hspace{0.5cm} (2.72)

As discussed previously a general formula for the weight vector \(W(j)\) can be derived which is applicable over all \(j>I\), which is the same as Equation (2.23), that is:

\[
W(j) = 2\lambda \sum_{n=j-N}^{j-1} d(n)X(n) + 2\lambda(1-2\lambda) \sum_{n=j-2N}^{j-N-1} d(n)X(n) +
\]

\[
+ 2\lambda(1-2\lambda) \sum_{n=j-3N}^{j-2N+1} d(n)X(n) \ldots
\]  \hspace{0.5cm} (2.73)

Setting the step size to be \(\lambda = \frac{1}{2}\), Equation (2.73) can be simplified as follows:
\[ W(j) = \sum_{n=j-N}^{j-1} d(n)X(n) \]  

(2.74)

Now consider the adaptive LMS weight when the input signal vector is \( X_u(n) \) as defined by Equation (2.65). We use \( u_m(j) \) to denote the \( m \)th elements of the adaptive weights. From Equation (2.74) we have:

\[
u_m(j) = w_m(j) = \sum_{n=j-N}^{j-1} d(n)x_u(n)\]

\[
= \sqrt{\frac{2}{N}} P_m \sum_{n=j-N}^{j-1} d(n) \cos \left[ m \left( n + \frac{1}{2} \right) \frac{\pi}{N} \right] 
\]

\[
= \sqrt{\frac{2}{N}} P_m \sum_{n=0}^{N-1} d(n-i+N) \cos \left[ m \left( n - \frac{1}{2} - j + N \right) \frac{\pi}{N} \right] 
\]

\[
= A_{m,j} \sqrt{\frac{2}{N}} P_m \sum_{n=0}^{N-1} d(n-j+N) \cos \left[ m \left( n + \frac{1}{2} \right) N \right] 
- B_{m,j} \sqrt{\frac{2}{N}} P_m \sum_{n=0}^{N-1} d(n-j+N) \sin \left[ m \left( n + \frac{1}{2} \right) \frac{\pi}{N} \right] 
\]

(2.75)

where \( A_{m,j} = \cos \left( m\pi - \frac{mj\pi}{N} \right) \) and \( B_{m,j} = \sin \left( m\pi - \frac{mj\pi}{N} \right) \). Using definitions for the DCT-II and DST-II in Equations (2.63) and (2.64), we obtain:

\[
u_m(j) = A_{m,j} C_{\Pi}(j,m) - B_{m,j} S_{\Pi}(j,m), \text{ for } m = 0, 1, \ldots, N-1 \]  

(2.76)

Clearly the adaptive weights are the combination of the DCT-II and DST-II coefficients. In order to separate the DCT-II and DST-II from \( u_m(j) \), we need another expression, which is independent of Equation (2.76).
Consider the case where the input signal vector is given as follows:

\[
X_v(n) = [x_v(n,1) \ x_v(n,2) \ ... \ x_v(n,N)]^T \tag{2.77}
\]

where

\[
x_v(n,m) = \sqrt{\frac{2}{N}} P_m \cos \left[ m \left( n + \frac{1}{2} \right) \frac{\pi}{N} \right] \tag{2.78}
\]

Under such circumstances the adaptive weights, denoted as, \( v_m(j) \) for \( m=1, ..., N \), are as follows:

\[
v_m(j) = \sum_{n=j-N}^{j-1} d(n)x_v(n,m)
\]

\[
= B_{m,j} \sqrt{\frac{2}{N}} P_m \sum_{n=0}^{N-1} d(n-j+N) \cos \left[ m \left( n + \frac{1}{2} \right) \frac{\pi}{N} \right] +
\]

\[
+ A_{m,j} \sqrt{\frac{2}{N}} P_m \sum_{n=0}^{N-1} d(n-j+N) \sin \left[ m \left( n + \frac{1}{2} \right) \frac{\pi}{N} \right]
\]

\[
= B_{m,j} C_{II}(j,m) + A_{m,j} S_{II}(j,m) \tag{2.79}
\]

From Equations (2.76) and (2.79) we can obtain the following:

\[
C_{II}(j,m) = A_{m,j} u_m(j) + B_{m,j} v_n(j) \tag{2.80}
\]

and

\[
S_{II}(j,m) = A_{m,j} v_m(j) - B_{m,j} u_m(j) \tag{2.81}
\]
Hence the DCT-II and DST-II of $d(n)$ ($n=j-N, j-N+1, \ldots, j-1$) can be obtained by the unified system illustrated in Figure 2.3, where $D_m$ are the transform coefficients. The values of $P_{m,j}$ and $Q_{m,j}$ are dependent on the transforms. For DCT-II, $P_{m,j} = A_{m,j}$ and $Q_{m,j} = B_{m,j}$, and for DST-II, $P_{m,j} = B_{m,j}$ and $Q_{m,j} = A_{m,j}$.

Let us consider the possibility of extending the techniques derived above to other versions of DCT's and DST's [Wang, Z. (1984)], the kernels of which are defined in Equations (2.42) to (2.49).

As shown in Section 2.5.2 the input signals $x_{(n)}$ are constructed by extending the range of time index $n$ of the transform kernels while keeping the same mathematical expression.

For the above transforms, the input signal vector should be:

**DCT-I and DST-I:**

\[
x_{u}(n, m) = \sqrt{\frac{2}{N}} P_m P_n \cos\left(\frac{n\pi}{N} m\right)
\]

for $m = 0, 1, \ldots, N$ and $n = 0, 1, \ldots, j$ (2.82)

and

\[
x_{v}(n, m) = \sqrt{\frac{2}{N}} \sin\left(\frac{n\pi}{N} m\right)
\]

for $m = 1, 2, \ldots, N$ and $n = 0, 1, \ldots, j$ (2.83)

**DCT-III and DST-III:**

\[
x_{u}(n, m) = \sqrt{\frac{2}{N}} P_n \cos\left(\frac{n(m+1)}{2} \frac{\pi}{N}\right)
\]

for $m = 0, 1, \ldots, N-1$ and $n = 0, 1, \ldots, j$ (2.84)

and

\[
x_{v}(n, m) = \sqrt{\frac{2}{N}} P_n \sin\left(\frac{n(m+1)}{2} \frac{\pi}{N}\right),
\]

for $m = 0, 1, \ldots, N-1$ and $n = 0, 1, \ldots, j$ (2.85)
DCT-IV and DST-IV:

\[ x_a(n,m) = \sqrt{\frac{2}{N}} \cos \left\{ (n + \frac{1}{2})(m + \frac{1}{2}) \frac{\pi}{N} \right\}, \]

for \( m = 0, 1, \ldots, N-1 \) and \( n = 0, 1, \ldots, j \) \hspace{1cm} (2.86)

and

\[ x_v(n,m) = \sqrt{\frac{2}{N}} \sin \left\{ (n + \frac{1}{2})(m + \frac{1}{2}) \frac{\pi}{N} \right\}, \]

for \( m = 0, 1, \ldots, N-1 \) and \( n = 0, 1, \ldots, j \) \hspace{1cm} (2.87)

In order for Equation (2.23) to be satisfied, the input signal vector \( X(n) \) should be either periodic or inverse periodic, and the period should be equal to the transform length \((N)\). Investigation of Equations (2.84) to (2.87) reveals that neither the DCT-III/DST-III nor the DCT-IV/DST-IV are periodic, and consequently LMS-based running analysers cannot be implemented for these transforms. Let us consider DCT-I and DST-I. Note that the transform length for DCT-I is \( N+1 \) rather than \( N \). In this case the input signals given by Equations (2.82) and (2.83) should be periodic or inverse periodic with period \( N+1 \) in order that an LMS-based running DCT-I and DST-I can be implemented. Investigation of Equations (2.82) and (2.83) shows that they are inverse periodic. However, the period is \( N \), which is not the same as the transform length \( N+1 \). Consequently, an LMS-based running DCT-I and DST-I cannot be implemented either.

### 2.5.3 LMS-based Running Discrete W Transforms

Now we extend the techniques derived in Section 2.5.1 and 2.5.2 to the case of Discrete W Transforms [Wang, Z (1981a, 1981b, 1981c and 1984)], the kernels of which can be rearranged as follows:

DWT-I:

\[ w_i(n,m) = \sqrt{\frac{2}{N}} \sin \left( \frac{\pi}{4} + \frac{nm}{N} \frac{2\pi}{N} \right) \]

\[ = \sqrt{\frac{2}{N}} \left[ \sin \frac{\pi}{4} \cos \left( \frac{mn}{N} \frac{2\pi}{N} \right) + \cos \frac{\pi}{4} \sin \left( \frac{mn}{N} \frac{2\pi}{N} \right) \right] \]
Clearly the DWT-I is identical to the DHT as depicted in Equation (2.55). The running computation has been studied in Section 2.5.1.

For the remaining DWT’s we have:

DWT-II: \[ w_{II}(n,m) = \frac{2}{N} \sin \left( \frac{\pi}{4} + m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \]

\[ = \frac{1}{N} \left[ \cos \left( m(n + \frac{1}{2}) \frac{2\pi}{N} \right) + \sin \left( m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \right] \]

\[ m, n = 0, 1, ..., N-1 \tag{2.89} \]

DWT-III: \[ w_{III}(n,m) = \frac{2}{N} \sin \left( \frac{\pi}{4} + n(m + \frac{1}{2}) \frac{2\pi}{N} \right) \]

\[ = \frac{1}{N} \left[ \cos \left( (m + \frac{1}{2})n \frac{2\pi}{N} \right) + \sin \left( (m + \frac{1}{2})n \frac{2\pi}{N} \right) \right] \]

\[ m, n = 0, 1, ..., N-1 \tag{2.90} \]

DWT-IV: \[ w_{IV}(n,m) = \frac{2}{N} \sin \left( \frac{\pi}{4} + (n + \frac{1}{2})(m + \frac{1}{2}) \frac{2\pi}{N} \right) \]

\[ = \frac{1}{N} \left[ \cos \left( (m + \frac{1}{2})(n + \frac{1}{2}) \frac{2\pi}{N} \right) + \sin \left( (m + \frac{1}{2})(n + \frac{1}{2}) \frac{2\pi}{N} \right) \right] \]

\[ m, n = 0, 1, ..., N-1 \tag{2.91} \]

As discussed previously, the input signal should be constructed as follows:
For DWT-I: \( x_u(n,m) = \sqrt{\frac{2}{N}} \cos\left\{ m \left( n + \frac{1}{2} \right) \frac{2\pi}{N} \right\} \)

for \( m = 0, 1, \ldots, N-1 \) and \( n = 0, 1, \ldots, j \) \hspace{1cm} (2.92)

and \( x_v(n,m) = \sqrt{\frac{2}{N}} \sin\left\{ m \left( n + \frac{1}{2} \right) \frac{2\pi}{N} \right\} \)

for \( m = 0, 1, \ldots, N-1 \) and \( n = 0, 1, \ldots, j \) \hspace{1cm} (2.93)

Note that \( x_u(n,m) \) and \( x_v(n,m) \) are periodic with period \( N \). In addition, they satisfy Equation (2.29). Hence the LMS-based technique as depicted in Figure 2.3 can be implemented for computing the running DWT-II. The parameters are as follows:

\[
P_{m,j} = \frac{A_{m,j} - B_{m,j}}{2} \quad \text{and} \quad Q_{m,j} = \frac{A_{m,j} + B_{m,j}}{2}.
\]

where \( A_{m,j} = \cos\left( \frac{2jm\pi}{N} \right) \) and \( B_{m,j} = \sin\left( \frac{2jm\pi}{N} \right) \).

Let us consider DWT-III. The input signals should be as follows:

\[
x_u(n,m) = \sqrt{\frac{2}{N}} \cos\left\{ (m + \frac{1}{2})n \frac{2\pi}{N} \right\} \)

for \( m = 0, 1, \ldots, N-1 \) and \( n = 0, 1, \ldots, j \) \hspace{1cm} (2.95)

and \( x_v(n,m) = \sqrt{\frac{2}{N}} \sin\left\{ (m + \frac{1}{2})n \frac{2\pi}{N} \right\} \)

for \( m = 0, 1, \ldots, N-1 \) and \( n = 0, 1, \ldots, j \) \hspace{1cm} (2.96)

It can be shown that \( x_u(n,m) \) and \( x_v(n,m) \) satisfy Equation (2.29), and \( x_u(n,m) = x_u(n+N,m) \) and \( x_v(n,m) = -x_v(n+N,m) \). Hence an LMS-based running
DWT-III can be implemented as depicted in Figure 2.3, where the parameters are given by:

\[ \begin{align*}
P_{m,j} &= \frac{A_{m,j} - B_{m,j}}{2} \quad \text{and} \quad Q_{m,j} = \frac{A_{m,j} + B_{m,j}}{2} \\
\end{align*} \tag{2.97} \]

where \( A_{m,j} = \cos \left( \frac{2jm\pi}{N} \right) \) and \( B_{m,j} = \sin \left( \frac{2jm\pi}{N} \right) \).

Finally, we consider DWT-IV. The input signals are:

\[ \begin{align*}
x_s(n,m) &= \sqrt{\frac{2}{N}} \cos \left( \left( m + \frac{1}{2} \right) (n + \frac{1}{2}) \frac{2\pi}{N} \right) \\
&\quad \text{for } m = 0, 1, ..., N-1 \text{ and } n = 0, 1, ..., j \tag{2.98} \\
\end{align*} \]

and\[ \begin{align*}
x_v(n,m) &= \sqrt{\frac{2}{N}} \sin \left( \left( m + \frac{1}{2} \right) (n + \frac{1}{2}) \frac{2\pi}{N} \right), \\
&\quad \text{for } m = 0, 1, ..., N-1 \text{ and } n = 0, 1, ..., j \tag{2.99} \\
\end{align*} \]

It can be shown that the system depicted in Figure 2.3 can also be used for computing the running DWT-IV. The parameters are as follows:

\[ \begin{align*}
P_{m,j} &= \frac{A_{m,j} - B_{m,j}}{2} \quad \text{and} \quad Q_{m,j} = \frac{A_{m,j} + B_{m,j}}{2} \\
\end{align*} \tag{2.100} \]

where \( A_{m,j} = \cos \left( \frac{2jm\pi}{N} \right) \) and \( B_{m,j} = \sin \left( \frac{2jm\pi}{N} \right) \).
2.6 Orthogonal Analysis Based on Other Adaptive Algorithms

As mentioned in Section 2.2 it is also possible to implement orthogonal analysers based on other adaptive algorithms. This section deals with this issue. Two adaptive algorithms are studied which are widely used in adaptive array processing and adaptive filtering. These are the adaptive Sample Matrix Inversion (SMI) algorithm and the adaptive Howells-Applebaum loop.

2.6.1 SMI Based Discrete Orthogonal Transforms

The adaptive SMI algorithm [Monzingo and Miller (1980), Ch.5] has been widely used in adaptive array processing due to its independent convergence property in relation to the signal environment. Consider the case where the adaptive SMI algorithm is applied to control the weight vector. The weight vector can be directly computed as follows:

\[
W(j) = R^{-1} r = \left[ \sum_{n=j-N}^{j-1} X(n)X^T(n) \right]^{-1} \sum_{n=j-N}^{j-1} d(n)X(n)
\]  \hspace{1cm} (2.101)

where \( X(n) = [x(n,0) \ x(n,1) \ … \ x(n,N-1)]^T \). Since \( x(n,m) \) satisfies Equation (2.29), we have:

\[
\sum_{n=j-N}^{j-1} X(n)X^T(n) = I
\]  \hspace{1cm} (2.102)

As discussed previously the input signal vector is periodic, that is, \( X(n) = X(n+N,m) \). Consequently Equation (2.101) can be rewritten as:

\[
W(MN) = \sum_{n=0}^{N-1} d(n+(M-1)N)X(n)
\]  \hspace{1cm} (2.103)
Equation (2.103) is exactly the same as the definition in Equation (2.6). Therefore the SMI algorithm can be used to perform most discrete orthogonal transforms, such as DFT, DwalT, DCT and DST, DHT and DWT.

### 2.6.2 Adaptive Howells-Applebaum Orthogonal Analyser

Consider the systems depicted in Figure 2.4, where the $d(t)$ and $X(t)$ are continuous in time, and Howells-Applebaum adaptive loops [Applebaum (1976)] are used to adjust the adaptive weights. Assuming that $d(t)$ is a band-limited signal, and can approximately be expressed as a combination of narrow band signals as follows:

$$d(t) = \sum_{m=1}^{M} B_m e^{i(\omega_n t + \psi_m)}$$  \hspace{1cm} (2.104)

where $B_m$ and $\psi_m \ (m = 1, 2, ..., M)$ are the amplitude and phase of the $n$th narrow band signal. Further it is assumed that these parameters are slow time varying due to the nature of narrow band signal. The central frequencies $\omega_n \ (n = 1, 2, ..., N)$ are uniformly distributed within the frequency band of $d(t)$.

The input signal vector $X(t)$ consists of a set of narrow band signals which cover the frequency band of $d(t)$. The $nth$ component of $X(t)$ is:

$$x_n(t) = A_m e^{i(\omega_n t + \phi_m)} \hspace{1cm} m = 1, 2, ..., M$$  \hspace{1cm} (2.105)

where $\phi_m$ is the phase of $x_m(t)$. Using Howells-Applebaum loops, the weight vector is controlled by the following equation:
\[ W(t) = \int_{0}^{\infty} x(t) e(t) dt = \int_{0}^{\infty} x(t) [d(t) - W^T(t) x(t)] dt \quad (2.106) \]

Equation (2.106) can be rewritten as:

\[ W(t) + X(t) X^T(t) W(t) = X(t) d(t) \quad (2.107) \]

Since all the components of \( X(t) \) are assumed to be narrow band, the solution of Equation (2.107) is given by:

\[ W(t) = [W(0) - W(\infty)] e^{-ig} + W(\infty) \quad (2.108) \]

where the steady state parameter vector \( W(\infty) \) is determined as follows:

\[ W(\infty) = R^{-1} r \quad (2.109) \]

where \( R = \langle X(t) X^T(t) \rangle \), \( r = \langle d(t) X(t) \rangle \), and \( \langle \ldots \rangle \) denotes the integration in Howells-Applebaum loop. From Equations (2.104) and (2.105) \( R \) and \( r \) can be evaluated as:

\[ R = \text{diag}\{A_1^2, A_2^2, \ldots, A_N^2\} \quad (2.110) \]

and

\[ r = \begin{bmatrix} A_1 B_1 e^{i(\psi_1 - \phi_1)} & A_2 B_2 e^{i(\psi_2 - \phi_2)} & \cdots & A_N B_N e^{i(\psi_N - \phi_N)} \end{bmatrix}^T \quad (2.111) \]

Hence the steady state weight vector is obtained as:

\[ W(\infty) = \begin{bmatrix} B_1 e^{i(\psi_1 - \phi_1)} / A_1 & B_2 e^{i(\psi_2 - \phi_2)} / A_2 & \cdots & A_N B_N e^{i(\psi_N - \phi_N)} / B_N \end{bmatrix}^T \quad (2.112) \]

The steady state output vector of the system is:
\[ Y(t) = [w_1(\infty)x_1(t) \quad w_2(\infty)x_2(t) \quad \ldots \quad w_N(\infty)x_N(t)]^T \]

\[ = [B_1 e^{j(\omega_1\tau + \psi_1)} \quad B_2 e^{j(\omega_2\tau + \psi_2)} \quad \ldots \quad B_N e^{j(\omega_N\tau + \psi_N)}]^T \]  \hspace{1cm} (2.113)

Therefore the system works as a comb filter which decomposes \( d(t) \) into \( N \) narrow band signals. Each of the narrow band signals corresponds to a Fourier component of \( d(t) \). Consequently, the system depicted in Figure 2.4 can be used as a spectral analyser for continuous signals.

\[ d(t) \]

\[ x_1(t) \quad W_1 \quad y_1(t) \]
\[ x_2(t) \quad W_2 \quad y_2(t) \]
\[ x_N(t) \quad W_{N-1} \quad \Sigma \]

Figure 2.4 The spectral analyser using adaptive Howells-Applebaum loop
2.7 Conclusions

This chapter considered adaptive filtering based techniques for orthogonal signal analysis. Firstly, the criteria for implementing LMS-based block orthogonal analysers were established. It was shown that a sufficient condition is that the kernels of orthogonal transforms are unitary. The performance of LMS-based techniques for computing some block discrete transforms was investigated, including the discrete Walsh transform, discrete cosine/sine transforms, discrete Hartley transform, and discrete W transforms. In addition, LMS-based techniques for computing running discrete orthogonal transforms were proposed. The LMS-based running DHT, DCT-II/DST-II, and DWT were developed and their performance was analysed.

The proposed LMS-based technique can be more efficient in terms of computational burden than the FFT-like fast algorithms. Considering DCT as an example, only $6N$ multiplications and $5N+2$ additions need to be performed during each sampling period, thus minimising the delay at the end of the data block. Although the number of multiplications and additions for the proposed LMS-DCT is higher than fast algorithms considered for small transform length, it is more efficient for larger block length.

A significant advantage of the proposed LMS-based discrete orthogonal analysers is the inherent parallel structure which makes it suitable for efficient VLSI implementation. As suggested by Widrow, et al. (1987), when parallel properties are exploited, the transform length, $N$, does not affect the computational speed. Note also that the proposed running LMS-DCT is applicable for arbitrary transform length, $N$, which is in direct contrast with some other fast algorithms [Chen, et al. (1977), Lee (1984)] that can only be applied for specific values of $N$. 
Two other orthogonal analysis techniques were also proposed. One is the SMI-based orthogonal analyser, which can be used for calculating all the existing discrete transforms. The other is a spectral analyser using the adaptive Howells-Applebaum loop. The later is applicable to signals which are continuous in time. It works as a comb filter and decomposes the input signal to be analysed into $N$ narrow band components.
Chapter 3 Computing Running Discrete Orthogonal Transforms Based On Their Shift Properties
3.1 Introduction

This chapter considers the issue of computing running discrete orthogonal transforms based on their shift properties. The shift properties are recursive equations which relate the updated transform coefficients with the previous transform coefficients. As an example, consider the $N$ point DFT of signal $d(n)$, where $n=j-N+1, j-N+2, ..., j$, given as follows:

$$Fr(j,m) = \sum_{n=0}^{N-1} d(j-N + n)W_N^{nm}, \text{ for } m=0, 1, ..., N-1$$  \hspace{1cm} (3.1)

where $W_N^{nm} = e^{-\frac{2\pi nm}{N}}$. When a new sample $d(j+1)$ becomes available, the signal block of interest shifts one sample in time to include $d(j+1)$ and exclude the sample $d(j-N+1)$. In other words, the signal block of interest becomes $d(n), n = j-N+2, j-N+3, ..., j, j+1$, the DFT of which is as follows:

$$Fr(j+1,m) = \sum_{n=0}^{N-1} d(j-N + 2 + n)W_N^{nm}, \text{ for } m=0, 1, ..., N-1$$  \hspace{1cm} (3.2)

It can be shown that

$$Fr(j+1,m) = W_N^{-m}[Fr(j,m) + d(j+1) - d(j-N+1)], \text{ for } m=0, 1, ..., N-1$$ \hspace{1cm} (3.3)

Equation (3.3) is the shift property of DFT coefficients. Clearly Equation (3.3) can be used to update the transform coefficients based on the previous transform coefficients and the new data sample $d(j+1)$ as well as the old sample $d(j-N+1)$. When compared with the FFT algorithm, Equation (3.3) is much more efficient in terms of computational
burden, since only $N$ complex multiplications and $2N$ complex additions are required for updating all the transform coefficients.

The shift properties of DFT are rather simple. However, for other discrete transforms such as DHT, DCT's, DST's and DWT's, the shift properties are more complex. The objective of this chapter is to investigate the shift properties for these other transforms, with a view to developing new techniques for computing the running transforms.

The rest of this chapter is organised as follows: Section 3.2 investigates the shift properties of DCT's and DST's and their implementation. Then the DHT and DWT are examined in Section 3.3 and Section 3.4 respectively. Finally Section 3.5 contains the conclusion.

### 3.2 Running DCT's and DST's Based on Their Shift Properties

This section investigates the shift properties of discrete cosine and sine transforms and their implementation as real time DCT/DST analysers. As mentioned in Chapter One, the first order shift properties of DCT's and DST's were initially derived by Yip and Rao [1987]. In this section similar first order shift properties are derived for the DHT and DWT's. The structures for implementing these first order shift properties are also proposed. In order to further reduce computational burden associated with the first order shift properties, we propose a new set of recursive equations, called Second Order Shift (SOS) properties. The implementation of the SOS properties is also investigated.
3.2.1 First Order Shift Properties and Their Implementation

The kernels for the family of DCT's and DST's were given in Equations (2.42)-(2.49). Based on these kernels the DCT's and DST's for a signal block \([d(j-N+1), \ d(j-N+2), \ ..., \ d(j)]\) are given as follows:

**DCT-I:**

\[
C_I(j,m) = \frac{2}{\sqrt{N}} \sum_{n=0}^{N} P_m \sum_{n=0}^{N} d(j-N+n) \cos\left(\frac{nm\pi}{N}\right),
\]

\[m = 0, 1, ..., N\]  \hspace{1cm} (3.4)

**DCT-II:**

\[
C_{II}(j,m) = \frac{2}{\sqrt{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos\left(\frac{m\pi n}{N} + \frac{1}{2}\right),
\]

\[m = 0, 1, ..., N-1\]  \hspace{1cm} (3.5)

**DCT-III:**

\[
C_{III}(j,m) = \frac{2}{\sqrt{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos\left(\frac{nm\pi}{N} + \frac{1}{2}\right),
\]

\[m = 0, 1, ..., N-1\]  \hspace{1cm} (3.6)

**DCT-IV:**

\[
C_{IV}(j,m) = \frac{2}{\sqrt{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos\left(\frac{(m+1/2)n\pi}{N} + \frac{1}{2}\right),
\]

\[m = 0, 1, ..., N-1\]  \hspace{1cm} (3.7)

**DST-I:**

\[
S_I(j,m) = \frac{2}{\sqrt{N}} \sum_{n=1}^{N} d(j-N+n) \sin\left(\frac{nm\pi}{N}\right),
\]

\[m = 1, 2, ..., N\]  \hspace{1cm} (3.8)

**DST-II:**

\[
S_{II}(j,m) = \frac{2}{\sqrt{N}} P_m \sum_{n=0}^{N} d(j-N+1+n) \sin\left(\frac{m\pi n}{N} + \frac{1}{2}\right),
\]

\[m = 1, 2, ..., N\]  \hspace{1cm} (3.9)
Chapter 3: Running Discrete Transforms Based On Their Shift Properties

DST-III: \[ S_{m}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} P_{n} d(j - N + 1 + n) \sin \left\{ n\left( m + \frac{1}{2} \right) \frac{\pi}{N} \right\}, \]
\[ m = 1, 2, ..., N \] (3.10)

DCT-IV: \[ S_{iv}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n) \sin \left\{ (n + \frac{1}{2})(m + \frac{1}{2}) \frac{\pi}{N} \right\}, \]
\[ m = 0, 1, ... N-1 \] (3.11)

where \( P_{m} \) is defined by Equation (2.39). The running DCT’s and DST’s at the instant \( j+1 \)
is determined by the following recursive equations [Yip and Rao (1987)]:

DCT-I and DST-I:

\[ C_{i}(j+1,m) = A_{m} C_{i}(j,m) + B_{m} P_{n} S_{i}(j,m) + \]
\[ + \sqrt{\frac{2}{N}} P_{m} \left\{ \left(-\frac{1}{\sqrt{2}}\right) A_{m} d(j - N) + \left(\frac{1}{\sqrt{2}} - 1\right) d(j - N + 1) \right\} + (-1)^{m} \left(1 - \frac{1}{\sqrt{2}}\right) A_{m} d(j) + (-1)^{m} \frac{1}{\sqrt{2}} d(j + 1) \right\} \]
\[ m = 0, 1, ... N-1 \] (3.12a)

and

\[ S_{i}(j+1,m) = A_{m} S_{i}(j,m) - B_{m} C_{i}(j,m) + \]
\[ + \sqrt{\frac{2}{N}} B_{m} \left\{ \frac{1}{\sqrt{2}} d(j - N) - (1 - \frac{1}{\sqrt{2}})(-1)^{m} d(j) \right\} \]
\[ m = 0, 1, ... N-1 \] (3.12b)
DCT-II and DST-II:

\[
C_{II}(j+1,m) = A_mC_{II}(j,m) + B_mC_{II}(j,m) + \frac{2}{N} P_m C_m \left[ (-1)^m d(j) - d(j-N) \right]
\]

\[m = 0, 1, \ldots N-1\] (3.13a)

and

\[
S_{II}(j+1,m) = A_m S_{II}(j,m) - B_m C_{II}(j,m) + \frac{2}{N} P_m D_m \left[ d(j-N) - (-1)^m d(j) \right]
\]

\[m = 0, 1, \ldots N-1\] (3.13b)

DCT-III and DST-III:

\[
C_{III}(j+1,m) = E_mC_{III}(j,m) + F_m S_{III}(j,m) + \frac{2}{N} \left\{ \left( -\frac{1}{\sqrt{2}} \right) E_m d(j-N) + \left( \frac{1}{\sqrt{2}} - 1 \right) d(j-N+1) + (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) F_m d(j) \right\}
\]

\[m = 0, 1, \ldots N-1\] (3.14a)

and

\[
S_{III}(j+1,m) = E_m S_{III}(j,m) - F_mC_{III}(j,m) + \frac{2}{N} \left\{ \left( \frac{1}{\sqrt{2}} \right) F_m d(j-N) + (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) E_m d(j) + \frac{(-1)^m}{\sqrt{2}} d(j+1) \right\}
\]

\[m = 0, 1, \ldots N-1\] (3.14b)

DCT-IV and DST-IV:

\[
C_{IV}(j+1,m) = E_mC_{IV}(j,m) + F_m S_{IV}(j,m) + \frac{2}{N} \left\{ -G_m d(j-N) + (-1)^m H_m d(j) \right\}
\]

\[m = 0, 1, \ldots N-1\] (3.15a)
and

\[ S_{rv}(j+1,m) = E_{m}S_{rv}(j,m) - F_{m}C_{rv}(j,m) + \sqrt{\frac{2}{N}} \{ H_{m}d(j-N) + (-1)^{m} G_{m}d(j) \} \]

\[ m = 0, 1, \ldots, N-1 \] (3.15b)

In Equations (3.12a) to (3.15b) \( A_{m} = \cos \frac{m\pi}{N} \), \( B_{m} = \sin \frac{m\pi}{N} \), \( C_{m} = \cos \frac{m\pi}{2N} \), \( D_{m} = \sin \frac{m\pi}{2N} \), \( E_{m} = \cos \frac{(2m+1)\pi}{2N} \), \( F_{m} = \sin \frac{(2m+1)\pi}{2N} \), \( G_{m} = \cos \frac{(2m+1)\pi}{4N} \), and \( H_{m} = \sin \frac{(2m+1)\pi}{4N} \).

Clearly Equations (3.12a) to (3.15b) can be used to update the DCT and DST coefficients. Now we investigate the structure for their implementations. Note from Equations (3.12a) to (3.15b) that the shift properties for a DCT and its corresponding DST are a pair of first order difference equations, which correspond to a first order linear shift invariant system with one input \( d(j) \) and two outputs, corresponding to the DCT and DST. Hence the transform operation can be viewed as passing the signal samples through a linear filter, whose outputs are the running transform coefficients. Consider the case of DCT-II and DST-II. Taking the z transform of Equations (3.13a) and (3.13b) yields:

\[ z\hat{C}_{H}(z,m) = A_{m}\hat{C}_{H}(z,m) + B_{m}\hat{S}_{H}(z,m) + \sqrt{\frac{2}{N}} P_{m}C_{m}[(-1)^{m} - z^{-N}]\hat{D}(z) \] (3.16a)

and

\[ z\hat{S}_{H}(z,m) = A_{m}\hat{S}_{H}(z,m) - B_{m}\hat{C}_{H}(z,m) + \sqrt{\frac{2}{N}} P_{m}D_{m}[z^{-N} - (-1)^{m}]\hat{D}(z) \] (3.16b)
where $\hat{C}(z,m)$, $\hat{S}(z,m)$, and $\hat{D}(z,m)$ are the z transforms of $C(j,m)$, $S(j,m)$ and $d(j)$ respectively. From Equations (3.16a) and (3.16b) we can obtain:

$$\hat{C}(z,m) = \frac{B_m z^{-1}}{1 - A_m z^{-1}} S(z,m) + \frac{\sqrt{2N} P_m \{(1)^m - z^{-N} \} z^{-1} C_m}{1 - A_m z^{-1}} \hat{D}(z)$$

(3.17a)

and

$$\hat{S}(z,m) = \frac{-B_m z^{-1}}{1 - A_m z^{-1}} C(z,m) + \frac{\sqrt{2N} P_m \{z^{-N} - (1)^m \} z^{-1} D_m}{1 - A_m z^{-1}} \hat{D}(z)$$

(3.17b)

The real time DCT-II and DST-II analyser depicted in Figure 3.1 can be implemented according to Equations (3.17a) and (3.17b). A similar lattice architecture is applied to other DCT's and DST's.

The architecture depicted in Figure 3.1 has a lattice like structure, in which the DCT-II and DST-II can be updated at the same time in parallel. Consequently, this structure provides high efficiency in terms of processing time. Hence, the architecture shown in Figure 3.1 is suitable for the situations where the computation of both the DCT-II and DST-II is required. However, it is not very efficient in terms of computational burden for the cases where only one running transform (DCT-II or DST-II) is required. This is because the updated DCT is related to its corresponding DST coefficients, and updating the DCT also requires updating its corresponding DST. In other words, there is an inherent dependence between DCT-II and DST-II which is a source of excessive computational burden.
Figure 3.1 Real time DCT-II and DST-II analyser structure, where
\[ O_m = \sqrt{\frac{2}{N}} P_m C_m \]
and \[ S_m = -\sqrt{\frac{2}{N}} P_m D_m \]

### 3.2.2 Second Order Shift Properties and Their Implementation

In order to reduce the computational burden associated with the first order shift properties, it is desirable to have recursive equations which enable the independent updating of DCT's and DST's respectively. This is possible when we note that the first order shift property for a DCT and its corresponding DST can be expressed as a pair of first order two variable recursive equations which can be equivalent to two independent second order recursive equations. The objective of this sub-section is to derive these second order recursive equations.
We present the derivation for DCT-II and DST-II in detail. Similar derivations can be performed for the remaining transforms, and the final results will be listed without proof in the interest of brevity (For detailed proofs refer to Appendix A-C).

Substituting (3.17b) into (3.17a) and after some manipulation we have:

\begin{equation}
(1 - 2A_mz^{-1} + z^{-2}) \hat{C}_n(z, m) = \sqrt{\frac{2}{N}} P_m C_m \left[ (-1)^m z^{-1} - (-1)^m z^{-2} - z^{-(N+1)} + z^{-N-2} \right] \tag{3.18a}
\end{equation}

and

\begin{equation}
(1 - 2A_mz^{-1} + z^{-2}) \hat{S}_n(z, m) = \sqrt{\frac{2}{N}} P_m C_m \left[ (-1)^m z^{-1} - (-1)^m z^{-2} + z^{-(N+1)} + z^{-N-2} \right] \tag{3.18b}
\end{equation}

Taking the inverse z-transform of Equations (3.18a) and (3.18b) yields the second order shift properties of the DCT-II and DST-II respectively:

\begin{equation}
C_n(j + 1, m) = 2A_mC_n(j, m) - C_n(j - 1, m) + \\
+ \sqrt{\frac{2}{N}} P_m C_m \left[ (-1)^m d(j) - (-1)^m d(j - 1) - d(j - N) + d(j - N - 1) \right] \tag{3.19}
\end{equation}

and

\begin{equation}
S_n(j + 1, m) = 2A_mS_n(j, m) - S_n(j - 1, m) + \\
+ \sqrt{\frac{2}{N}} P_m D_m \left[ (-1)^m d(j) - (-1)^m d(j - 1) + d(j - N) + d(n - N - 1) \right] \tag{3.20}
\end{equation}

Clearly Equations (3.19) and (3.20) can independently update the DCT-II or DST-II respectively.
The second order shift properties for other DCT's and DST's can be obtained using a similar approach (See Appendices A-C). Before listing the results, we introduce the following coefficients:

\[ E_m = \cos \frac{(2m+1)\pi}{2N}, \quad F_m = \sin \frac{(2m+1)\pi}{2N}; \quad \text{and} \]
\[ G_m = \cos \frac{(2m+1)\pi}{4N}, \quad H_m = \sin \frac{(2m+1)\pi}{4N}. \quad (3.21) \]

The second order shift properties of the DCT and DST family are listed as follows (See Appendix A-C):

DCT-I:

\[ C_{j}(j+1,k) = 2A_mC_{j}(j,m) - \left[A_m^2 + P_mB_m^2\right]C_{j}(j-1,m) + T_{11}^{m}d(j+1) + T_{12}^{m}d(j) + T_{13}^{m}d(j-1) - T_{14}^{m}d(j-N+1) + T_{15}^{m}d(j-N) + T_{16}^{m}d(j-N-1) \quad (3.22) \]

where

\[ T_{11}^{m} = (-1)^m P_m \sqrt{\frac{1}{N}}; \]
\[ T_{12}^{m} = (-1)^m (1 - \sqrt{2}) A_m P_m \sqrt{\frac{2}{N}}; \]
\[ T_{13}^{m} = (-1)^m (1 - \sqrt{2}) P_m \sqrt{\frac{2}{N}}; \]
\[ T_{14}^{m} = (\sqrt{\frac{1}{2}} - 1) P_m \sqrt{\frac{2}{N}}; \]
\[ T_{15}^{m} = (1 - \sqrt{2}) A_m P_m \sqrt{\frac{2}{N}}; \quad \text{and} \]
\[ T_{1,6}^m = P_m \sqrt{\frac{1}{N}} \] (3.23)

Note that in Equation (3.22) the coefficient, \[ A_m^2 + P_m B_m^2 \], is always equal to 1 when \( m \neq 0 \) and \( m \neq N \). This means that this coefficient introduces a multiplication only when \( m = 0 \) or \( m = N \).

**DST-I:**

\[
S_I(j + 1, m) = 2 A_m S_I(j, m) - \left[ A_m^2 + P_m B_m^2 \right] S_I(j - 1, m) + U_{1,2}^m d(j) + U_{1,3}^m d(j - 1) + U_{1,4}^m d(j - N) + U_{1,5}^m d(j - N - 1) \] (3.24)

where

\[
U_{1,1}^m = (-1)^m B_m \left( \frac{1 - P_m}{\sqrt{2}} \right) \sqrt{\frac{2}{N}};
\]

\[
U_{1,2}^m = (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) A_m (1 - P_m) B_m \sqrt{\frac{2}{N}};
\]

\[
U_{1,3}^m = B_m \left( \frac{1}{\sqrt{2}} + (1 - \frac{1}{\sqrt{2}}) P_m \right) \sqrt{\frac{2}{N}}; \quad \text{and}
\]

\[
U_{1,4}^m = A_m B_m (P_m - 1) \sqrt{\frac{1}{N}} \] (3.25)

**DCT-II:**

\[
C_H(j + 1, m) = 2 A_m C_H(j, m) - C_H(j - 1, m) + T_{2,1}^m \left( (-1)^m d(j) - (-1)^m d(j - 1) - d(j - N) + d(j - N - 1) \right) \] (3.26)
where

\[ T_{2,1}^{\text{m}} = \sqrt{\frac{2}{N}} P_m C_m \]  

(3.27)

DST-II:

\[ S_H^{\text{m}} (j + 1, m) = 2 A_{m} S_H^{\text{m}} (j, m) - S_H^{\text{m}} (j - 1, m) + \]
\[ + U_{2,1}^{\text{m}} \left[ (-1)^m d(j) - (-1)^m d(j - 1) + d(j - N) + d(j - N - 1) \right] \]  

(3.28)

where

\[ U_{2,1}^{\text{m}} = \sqrt{\frac{2}{N}} P_m D_m \]  

(3.29)

DCT-III:

\[ C_{\text{III}}^{\text{m}} (j + 1, m) = 2 E_m C_{\text{III}}^{\text{m}} (j, m) - C_{\text{III}}^{\text{m}} (j - 1, m) + \]
\[ + T_{3,1}^{\text{m}} d(j) + T_{3,2}^{\text{m}} d(j - N + 1) + T_{3,3}^{\text{m}} d(j - N) + T_{3,4}^{\text{m}} d(j - N - 1) \]  

(3.30)

where

\[ T_{3,1}^{\text{m}} = (-1)^m \sqrt{\frac{2}{N}} F_m ; \]
\[ T_{3,2}^{\text{m}} = (1 - \sqrt{\frac{1}{2}}) \sqrt{\frac{2}{N}} ; \]
\[ T_{3,3}^{\text{m}} = (1 - \sqrt{2}) E_m \sqrt{\frac{2}{N}} ; \] and

\[ T_{3,4}^{\text{m}} = \frac{1}{\sqrt{N}} . \]  

(3.31)
DST-III:

\[ S_{III}(j+1, m) = 2E_mS_{III}(j, m) - S_{III}(j-1, m) + \]
\[ + U_{3,1}^m d(j+1) + U_{3,2}^m d(j) + U_{3,3}^m d(j-1) + U_{3,4}^m d(j-N) \]  \hspace{1cm} (3.32)

where

\[ U_{3,1}^m = (-1)^m \sqrt{\frac{1}{N}} ; \]

\[ U_{3,2}^m = (-1)^m \left[ (1 - \frac{1}{2}) F_m - \frac{1}{2} E_m \right] \sqrt{\frac{2}{N}} ; \]

\[ U_{3,3}^m = (-1)^m (1 - \frac{1}{2}) \sqrt{\frac{2}{N}} (F_m^2 + E_m F_m) ; \text{ and } \]

\[ U_{3,4}^m = \sqrt{\frac{2}{N}} F_m \]  \hspace{1cm} (3.33)

DCT-IV:

\[ C_{IV}(j+1, m) = 2E_mC_{IV}(j, m) - C_{IV}(j-1, m) + \]
\[ + T_{4,1}^m[d(j) - d(j-1)] + T_{4,2}^m[d(j-N) - d(j-N-1)] \] \hspace{1cm} (3.34)

where

\[ T_{4,1}^m = (-1)^m H_m \sqrt{\frac{2}{N}} ; \text{ and } \]

\[ T_{4,2}^m = -G_m \sqrt{\frac{2}{N}} . \] \hspace{1cm} (3.35)

DST-IV:

\[ S_{IV}(j+1, m) = 2E_mS_{IV}(j, m) - S_{IV}(j-1, m) + \]
\[ + U_{4,1}^m[d(j) - d(j-1)] + U_{4,2}^m[d(j-N) - d(j-N-1)] \]  \hspace{1cm} (3.36)
where

\[
U_{4,1}^m = (-1)^m G_m \sqrt{\frac{2}{N}}; \text{ and }
\]

\[
U_{4,2}^m = H_m \sqrt{\frac{2}{N}}. \tag{3.37}
\]

Real time DCT/DST analysers can also be implemented on the basis of the second order shift properties as described by Equations (3.22)-(3.37). Note that Equations (3.22)-(3.37) are second order shift invariant difference equations. Consequently a transform operation can be considered as a linear shift invariant system, which transforms the input signal sequence into the transform coefficients. Consider the DCT-II and DST-II as an example. The transfer function of the system can be obtained from Equations (3.18a) and (3.18b):

\[
\hat{C}_{Ii}(z,m) = \sqrt{\frac{2}{N}} P_m C_m \left[ (-1)^m - z^{-N} \right] z^{-1} \frac{1 - z^{-1}}{1 - 2A_m z^{-1} + z^{-2}} \tag{3.38}
\]

and

\[
\hat{S}_{Ii}(z,m) = -\sqrt{\frac{2}{N}} P_m D_m \left[ (-1)^m - z^{-N} \right] z^{-1} \frac{1 + z^{-1}}{1 - 2A_m z^{-1} + z^{-2}} \tag{3.39}
\]

Each of Equations (3.38) and (3.39) corresponds to a system consisting of a shift register array and a second order IIR filter. The corresponding architecture is illustrated in Figure 3.2. This structure is also applicable for other DCT's and DST's.
Figure 3.2 The real time DCT-II analyser based on its second order shift property,

where $O_m = T_{2,1} = \sqrt{\frac{2}{N}} P_m C_m$, $S_m = -T_{2,2} = -\sqrt{\frac{2}{N}} P_m D_m$.

### 3.2.3 Performance Analysis

The techniques for computing running DCT's and DST's based on second order shift properties provide certain advantages over the first order approach as well as the algorithm proposed by Murthy and Swamy (1992). Firstly, the computational burden is significantly reduced. A comparison between the computational burden associated with the proposed second order shift properties and first order shift properties by Yip and Rao (1987) and Murthy and Swamy (1992) is illustrated in Table 3.1. Clearly in all cases the computational burden associated with the proposed approaches is significantly reduced when compared to both references [Yip and Rao (1987)] and [Murthy and Swamy (1992)]. For the DCT-II case, the proposed second order approach can reduce the number of multiplications by 66% when compared with the first order approach.
Table 3.1 Comparison of computational burden

<table>
<thead>
<tr>
<th></th>
<th>Multiplications</th>
<th>Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT-I</td>
<td>10N</td>
<td>7N</td>
</tr>
<tr>
<td>DCT-II</td>
<td>6N</td>
<td>5N</td>
</tr>
<tr>
<td>DCT-III</td>
<td>10N</td>
<td>5N</td>
</tr>
<tr>
<td>DCT-IV</td>
<td>8N</td>
<td>6N</td>
</tr>
<tr>
<td>DST-I</td>
<td>10N</td>
<td>5N</td>
</tr>
<tr>
<td>DST-II</td>
<td>6N</td>
<td>5N</td>
</tr>
<tr>
<td>DST-III</td>
<td>10N</td>
<td>5N</td>
</tr>
<tr>
<td>DST-IV</td>
<td>8N</td>
<td>6N</td>
</tr>
</tbody>
</table>

3.3 Running DHT Based On Its Shift Properties

This section presents a novel structure for computing the running DHT based on its shift properties. Both the first order and second order shift properties are studied and the structures for their implementation as real time DHT analysers are presented. The architecture based on the proposed first order shift property is characterised by lower computational burden when compared with the approach by Liu and Chiu (1993).

3.3.1 First Order Shift Property And Its Implementation

The DHT for the block signal \([d(j-N+1), d(j-N+2), \ldots d(j)]\) is defined as follows:
Chapter 3: Running Discrete Transforms Based On Their Shift Properties

$$H(j,m) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n) \operatorname{cas}(\frac{2\pi nm}{N})$$  \hspace{1cm} (3.40)$$

where \(\operatorname{cas}(\frac{2\pi nm}{N}) = \cos(\frac{2\pi nm}{N}) + \sin(\frac{2\pi nm}{N})\) and \(N\) is the block length. For the case where the input signal is a stream of samples, \(H(j,m)\) is regarded as the running DHT at time instant \(j\). The running DHT at time instant \(j+1\) is given as follows:

$$H(j+1,m) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} d(j - N + 2 + n) \operatorname{cas}(\frac{2\pi nm}{N})$$  \hspace{1cm} (3.41)$$

substituting \(n' = n+1\) into Equation (3.41) yields:

$$H(j+1,m) = \sqrt{\frac{1}{N}} \sum_{n=1}^{N} d(j - N + 1 + n) \operatorname{cas}(\frac{2\pi (n'-1)m}{N})$$

$$= \cos(\frac{2\pi m}{N}) \sqrt{\frac{1}{N}} \sum_{n=1}^{N} d(j - N + 1 + n) \operatorname{cas}(\frac{2\pi n'm}{N}) -$$

$$- \sin(\frac{2\pi m}{N}) \sqrt{\frac{1}{N}} \sum_{n=1}^{N} d(j - N + 1 + n) \operatorname{cas}(\frac{-2\pi n'm}{N}) +$$

$$+ \sqrt{\frac{1}{N}} \left[ \cos(\frac{2\pi m}{N}) - \sin(\frac{2\pi m}{N}) \right] [d(j+1) - d(j - N + 1)]$$

$$= A_m H(j,m) - B_m H(j,-m) + \sqrt{\frac{1}{N}} (A_m - B_m) [d(j+1) - d(j - N + 1)],$$

\[m = 0, 1, ..., N-1\]  \hspace{1cm} (3.42)
where \( A_m = \cos \left( \frac{2\pi m}{N} \right) \) and \( B_m = \sin \left( \frac{2\pi m}{N} \right) \). Similarly we can derive an equation for \( H(j+1,-k) \) as follows:

\[
H(j+1,-m) = A_m H(j,-m) + B_m H(j,m) + \frac{1}{N} \left( A_m - B_m \right) [d(j+1) - d(j - N + 1)],
\]

\[m=0, 1, ..., N-1\] \hspace{1cm} (3.43)

Clearly \( 4N \) multiplications and \( 6N \) additions are required for every new sample update.

One can easily verify the reverse symmetrical property of DHT, given as follows:

\[
H(j,-m) = H(j,N-m), \quad m=0, 1, ..., N-1
\] \hspace{1cm} (3.44)

Hence Equations (3.42) and (3.43) can be equivalent to the following recursive formula:

\[
H(j+1,0) = H(j,0) + \frac{1}{N} \left[ d(j+1) - d(j - N + 1) \right]
\] \hspace{1cm} (3.45a)

and

\[
H(j+1,N/2) = -H(j,N/2) - \frac{1}{N} \left[ d(j+1) - d(j - N + 1) \right], \quad \text{for even } N
\] \hspace{1cm} (3.45b)

and

\[
H(j+1,m) = A_m H(j,m) - B_m H(j,-m) + \frac{1}{N} \left( A_m - B_m \right) [d(j+1) - d(j - N + 1)]
\]

\[
H(j+1,N-m) = A_m H(j,N-m) + B_m H(j,m) + \frac{1}{N} \left( A_m - B_m \right) [d(j+1) - d(j - N + 1)]
\]

\[k=1, ..., N/2-1 \quad \text{(when } N \text{ is even)} \quad \text{or} \quad k=1, ..., (N-1)/2 \quad \text{(when } N \text{ is odd)}\] \hspace{1cm} (3.45c)

Making use of the common term \([d(j+1) - d(j - N + 1)]\), Equations (3.45a) to (3.45c) can be rewritten with reduced computational burden as follows:
Chapter 3: Running Discrete Transforms Based On Their Shift Properties

\[ y(j + 1) = \sqrt{\frac{1}{N}} [d(j + 1) - d(j - N + 1)] \]  
(3.46)

and

\[ H(j + 1, 0) = H(j, 0) + y(j + 1); \]  
(3.47a)

and

\[ H(j + 1, N/2) = -H(j, N/2) - y(j + 1), \text{ for even } N \]  
(3.47b)

and

\[ H(j + 1, m) = A_m H(j, m) - B_m H(j, -m) + (A_m - B_m) y(j + 1) \]
\[ H(j + 1, N - m) = A_m H(j, N - m) + B_m H(j, m) + (A_m - B_m) y(j + 1) \]

\text{for } k = 1, ..., N/2 - 1 (when } N \text{ is even) or } k = 1, ..., (N - 1)/2 (when } N \text{ is odd)  
(3.47c)

It is also worthwhile comparing the proposed algorithm as described by Equations (3.47a)-(3.47c) with the results achieved by Liu and Chiu (1993). The dual updated pair in Liu and Chiu's paper (1993) are DCT-I like and DST-I like transforms. The summation of these two gives the DHT. In other words, each dual updated pair corresponds to one DHT coefficient. By contrast, the dual updated pair in Equation (3.47c) are \( H(j, m) \) and \( H(j, N - m) \). This means that each dual updated pair gives two DHT coefficients. For computing \( H(j + 1, m), m = 1, 2, ..., N - 1 \), Liu and Chiu's approach [Liu and Chiu (1993)] requires \( N - 1 \) dual updated pairs, while the proposed approach requires only \( N/2 - 1 \) when \( N \) is even or \( (N - 1)/2 \) when \( N \) is odd. Note that the time-recursive equations by Liu and Chiu (refer to Equations (48)-(51), in [Liu and Chiu...
(1993)) for updating the DCT-I and DST-I pair require similar computational burden as Equation (3.47c). Hence the computational burden associated with the proposed algorithm depicted in Equations (3.47a)-(3.47c) is about only half when compared with the approach by Liu and Chiu (1993).

At this point it is possible to construct the discrete Hartley filter based on Equations (3.46) and (3.47a)-(3.47c). Taking the z-transform of Equations (3.46), (3.47a)-(3.47c) and rearranging the results we have:

\[
Y(z) = \sqrt{\frac{1}{N}} (1 - z^{-N}) \hat{D}(z) \quad (3.48)
\]

and

\[
\hat{H}(z, 0) = \frac{1}{1 - z^{-1}} Y(z) \quad (3.49a)
\]

\[
\hat{H}(z, N/2) = -\frac{1}{1 + z^{-1}} Y(z), \quad \text{for even } N \quad (3.49b)
\]

\[
\begin{aligned}
&\hat{H}(z, m) = \frac{-B_m z^{-1}}{1 - A_m z^{-1}} \hat{H}(z, N - m) + \frac{(A_m - B_m)}{1 - A_m z^{-1}} \hat{Y}(z) \\
&\hat{H}(z, N - m) = \frac{B_m z^{-1}}{1 - A_m z^{-1}} \hat{H}(z, m) + \frac{(A_m + B_m)}{1 - A_m z^{-1}} \hat{Y}(z)
\end{aligned}
\]

\[
\text{for } k=1, \ldots, N/2-1 \text{ (when } N \text{ is even) or } k=1, \ldots, (N-1)/2 \text{ (when } N \text{ is odd)} \quad (3.49c)
\]

The system corresponding to Equations (3.48) and (3.49c) is illustrated in Figure 3.3, which has a lattice like structure and is similar to the DCT/DST as depicted in Figure 3.1.
Figure 3.3 The real time DHT based on its first order shift property, where

\[ A_m = \cos \frac{2\pi m}{N}, \quad B_m = \sin \frac{2\pi m}{N}, \quad O_m = \sqrt{\frac{2}{N}}(A_m - B_m) \quad \text{and} \]

\[ S_m = \sqrt{\frac{2}{N}}(A_m + B_m) \]

### 3.3.2 Second Order Shift Property And Its Implementation

The second order shift property can be derived by solving \( \hat{H}(z,m) \) from Equation (3.49c). The result is as follows:

\[
(1 - 2A_m z^{-1} + z^{-2}) \hat{H}(z,m) = (A_m - B_m - z^{-1}) \hat{Y}(z)
\]

Using Equation (3.48) we have

\[
(1 - 2A_m z^{-1} + z^{-2}) H(z,m) = \frac{1}{\sqrt{N}} \left[ (A_m - B_m)(1 - z^{-N}) - (z^{-1} - z^{-(N+1)}) \right] \hat{X}(z)
\]
taking the inverse z transform of Equation (3.51) gives

$$H(j+1,m) = 2A_m H(j,m) - H(j-1,m) + \frac{1}{N} \left( (A_m - B_m) [x(j+1) - x(j-N+1)] - [x(j) - x(j-N)] \right)$$

$$m = 0, 1, \ldots, N-1$$

(3.52)

Clearly Equation (3.52) can independently update the transform coefficients. A real time DHT analyser can also be implemented on the basis of the above second order shift property. The architecture is illustrated in Figure 3.4, which consists of a shift register array followed by a second order IIR filter.

Figure 3.4 Real time DHT based on second order shift properties, where $A_m = \cos \frac{2\pi m}{N}$,

$$O_m = \sqrt{\frac{1}{N} (A_m - B_m)}$$ and $$S_m = \sqrt{\frac{1}{N}}.$$
3.3.3 Performance Analysis

Let us consider the performance of the proposed approaches in terms of the computational burden. Table 3.2 shows the comparison between the lattice structure suggested by Liu and Chiu (1992), the proposed first order shift property described by Equations (3.46)-(3.47), and the proposed second order shift properties as depicted in Equation (3.52). It is evident that the proposed approach based on the first order shift property is characterised by the least computational burden when compared with the other two approaches.

<table>
<thead>
<tr>
<th></th>
<th>The approach by Liu and Chiu (1992)</th>
<th>The Proposed 2nd order Shift Property</th>
<th>The Proposed 1st order Shift Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplications</td>
<td>4N</td>
<td>3N-2</td>
<td>3N-6 for even N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3N-3 for odd N</td>
</tr>
<tr>
<td>Additions</td>
<td>5N-2</td>
<td>4N+1</td>
<td>2N-1 for even N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2N for odd N</td>
</tr>
</tbody>
</table>

3.4 Running DWT's Based on Their Shift Properties

The running discrete W transforms for the input sequence 
\[ d(j - N + 1) \quad d(j - N + 2) \quad \ldots \quad d(j) \] are as follows [Wang, Z. (1984)]:

\[
W_{s,l} (j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n) \sin \left( \frac{\pi}{4} + nm \frac{2\pi}{N} \right) \\
m = 0, 1, \ldots, N-1
\] (3.53)
Chapter 3: Running Discrete Transforms Based On Their Shift Properties

DWT-II: \[ W_{s,II}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \sin \left( \frac{\pi}{4} + m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \]
\[ m = 0, 1, ..., N-1 \] (3.54)

DWT-III: \[ W_{s,III}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \sin \left( \frac{\pi}{4} + m\left(n + \frac{1}{2}\right) \frac{2\pi}{N} \right) \]
\[ m = 0, 1, ..., N-1 \] (3.55)

DWT-IV: \[ W_{s,IV}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \sin \left( \frac{\pi}{4} + \left(n + \frac{1}{2}\right) \left(m + \frac{1}{2}\right) \frac{2\pi}{N} \right) \]
\[ m = 0, 1, ..., N-1 \] (3.56)

where \( N \) is the block length of the transform. Note that the DWT-I is identical to the discrete Hartley transform (DHT) in Equation (3.40).

3.4.1 First Order Shift Properties and Their Implementation

In order to derive the shift properties of the DWT's, we define a set of new transforms by changing the sine functions in Equations (3.53)-(3.56) into cosine functions. The new transforms are referred to as the Discrete Cosine W Transforms (DCWT's), which are given as follows:

DCWT-I: \[ W_{c,I}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{\pi}{4} + nm \frac{2\pi}{N} \right) \]
\[ m = 0, 1, ..., N-1 \] (3.57)

DWT-II: \[ W_{c,II}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{\pi}{4} + m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \]
\[ m = 0, 1, ..., N-1 \] (3.58)
DWT-III: \[ W_{c,iii}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{\pi}{4} + n(m+\frac{1}{2}) \frac{2\pi}{N} \right) \]
\[ m = 0, 1, ..., N-1 \]  \hspace{1cm} (3.59)

DWT-IV: \[ W_{c,iv}(j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{\pi}{4} + (n+\frac{1}{2})(m+\frac{1}{2}) \frac{2\pi}{N} \right) \]
\[ m = 0, 1, ..., N-1 \]  \hspace{1cm} (3.60)

One can easily verify that the transform kernel matrix of the above four versions of DCWT are orthogonal and unitary. Considering the fact that \( \cos \frac{\pi}{4} = \sin \frac{\pi}{4} = \frac{\sqrt{2}}{2} \), we have the following relations between the kernels of DWT's and corresponding DCWT's:

\[ \sin \left[ \frac{\pi}{4} + n(N-m) \frac{2\pi}{N} \right] = \cos \left[ \frac{\pi}{4} + nm \frac{2\pi}{N} \right]; \]  \hspace{1cm} (3.61)

and

\[ \sin \left[ \frac{\pi}{4} + n(N-m-1+\frac{1}{2}) \frac{2\pi}{N} \right] = \cos \left[ \frac{\pi}{4} + n(m+\frac{1}{2}) \frac{2\pi}{N} \right]; \]  \hspace{1cm} (3.62)

and

\[ \sin \left[ \frac{\pi}{4} + (n+\frac{1}{2})(N-m) \frac{2\pi}{N} \right] = -\cos \left[ \frac{\pi}{4} + (n+\frac{1}{2})m \frac{2\pi}{N} \right]; \]  \hspace{1cm} (3.63)

and

\[ \sin \left[ \frac{\pi}{4} + (n+\frac{1}{2})(N-m-1+\frac{1}{2}) \frac{2\pi}{N} \right] = -\cos \left[ \frac{\pi}{4} + (n+\frac{1}{2})(m+\frac{1}{2}) \frac{2\pi}{N} \right] \]  \hspace{1cm} (3.64)
Based on Equations (3.61) to (3.64) we can obtain the relationship between DWT and DCWT as follows:

\[ W_{s,l}(j, N-m) = W_{c,l}(j, m) \text{ or } W_{c,l}(j, N-m) = W_{s,l}(j, m) \quad (3.65) \]

and

\[ W_{s,ll}(j, N-m-1) = W_{c,ll}(j, m) \text{ or } W_{c,ll}(j, N-m-1) = W_{s,ll}(j, m) \quad (3.66) \]

and

\[ W_{s,lll}(j, N-m) = -W_{c,lll}(j, m) \text{ or } W_{c,lll}(j, N-m) = -W_{s,lll}(j, m) \quad (3.67) \]

and

\[ W_{s,lV}(j, N-m) = -W_{c,lV}(j, m-1) \text{ or } W_{c,lV}(j, N-m-1) = -W_{s,lV}(j, m) \quad (3.68) \]

Now we consider the running DWT at time \( j+1 \). Due to the similar nature of each of the four versions and in the interest of brevity, we will derive the DWT-I in detail and list the results for remaining transforms (See Appendices D-F for proof). Consider the DWT-I:

\[
W_{s,l}(j + 1, m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 2 + n) \sin \left( \frac{\pi}{4} + m \frac{2\pi}{N} \right) \quad (3.69)
\]

Let \( n' = n + 1 \), Equation (3.69) becomes:

\[
W_{s,l}(j + 1, m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \sin \left( \frac{\pi}{4} + (n'-1)m \frac{2\pi}{N} \right) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \sin \left( \frac{\pi}{4} + n'k \frac{2\pi}{N} - \frac{2\pi k}{N} \right) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \sin \left( \frac{\pi}{4} + n'm \frac{2\pi}{N} \cos \left( \frac{2\pi m}{N} \right) \right) - \sqrt{\frac{2}{N}} \sum_{n=1}^{N} x(j - N + 1 + n') \cos \left( \frac{\pi}{4} + n'm \frac{2\pi}{N} \right) \sin \left( \frac{2\pi m}{N} \right) \quad (3.70)
\]
Consider the definitions of DWT in Equation (3.53) and DCWT-I in Equation (3.57), it follows that

\[ W_{s,t}(j+1,m) = \]

\[ A_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n') \sin \left( \frac{\pi}{4} + n'm \frac{2\pi}{N} \right) + A_m \sqrt{\frac{1}{N}} \left[ d(j+1) - d(j - N + 1) \right] \]

\[ - B_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n') \cos \left( \frac{\pi}{4} + n'm \frac{2\pi}{N} \right) - B_m \sqrt{\frac{1}{N}} \left[ d(j+1) - d(j - N + 1) \right] + \sqrt{\frac{1}{N}} \left[ d(j+1) - d(j - N + 1) \right] \left[ A_m - B_m \right] \]

\[ (3.71) \]

where \( A_m = \cos \frac{2\pi m}{N} \) and \( B_m = \sin \frac{2\pi m}{N} \). Similarly for DCWT-I we have

\[ W_{c,t}(j+1,m) = A_m W_{s,t}(j,m) + B_m W_{c,t}(j,m) + \sqrt{\frac{1}{N}} \left[ d(j+1) - d(j - N + 1) \right] \left[ A_m + B_m \right] \]

\[ (3.72) \]

The results for the remaining transforms can be obtained using the similar approaches (See Appendices D-F). The results are listed as follows:

DWT-II:

\[ W_{s,ll}(j+1,m) = A_m W_{s,ll}(j,m) - B_m W_{c,ll}(j,m) + \sqrt{\frac{2}{N}} \left[ d(j+1) - d(j - N + 1) \right] \sin \left( \frac{\pi}{4} - \frac{m\pi}{N} \right) \]

\[ (3.73) \]

DCWT-II:

\[ W_{c,ll}(j+1,m) = A_m W_{c,ll}(j,m) + B_m W_{s,ll}(j,m) + \sqrt{\frac{2}{N}} \left[ d(j+1) - d(j - N + 1) \right] \cos \left( \frac{\pi}{4} - \frac{m\pi}{N} \right) \]

\[ (3.74) \]
DWT-III:

\[ W_{s,III}(j+1, m) = C_m W_{s,III}(j, m) - D_m W_{c,III}(j, m) \]

\[-\sqrt{\frac{2}{N}} [d(j+1)+d(j-N+1)] \sin \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{2\pi}{N} \right) \]  

(3.75)

DCWT-III:

\[ W_{c,III}(j+1, m) = C_m W_{c,III}(j, m) + D_m W_{s,III}(j, m) \]

\[-\sqrt{\frac{2}{N}} [d(j+1)+d(j-N+1)] \cos \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{2\pi}{N} \right) \]  

(3.76)

DWT-IV:

\[ W_{s,IV}(j+1, m) = C_m W_{s,IV}(j, m) - D_m W_{c,IV}(j, m) \]

\[-\sqrt{\frac{2}{N}} [d(j+1)+d(j-N+1)] \sin \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{\pi}{N} \right) \]  

(3.77)

DCWT-IV:

\[ W_{c,IV}(j+1, m) = C_m W_{c,IV}(j, m) + D_m W_{s,IV}(j, m) \]

\[-\sqrt{\frac{2}{N}} [d(j+1)+d(j-N+1)] \cos \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{\pi}{N} \right) \]  

(3.78)

The recursive Equations (3.71)-(3.78) can be used to update the transform coefficient at every sample update. Note that the coefficients of DWT and the corresponding DCWT are dependent on each other. The process of updating the DWT (or DCWT) requires updating the DCWT (or DWT). This is a source of excessive computational burden. Fortunately, this problem can be alleviated by using the relationships between DWT and
DCWT's as depicted in Equations (3.65) to (3.68). For example, using the relation $W_{s,t}(j,N-m) = W_{c,t}(j,m)$ and substituting into Equations (3.70) and (3.71) yields the following recursive equations to update DWT-I:

\[
W_{s,t}(j+1,m) = A_m W_{s,t}(j,m) - B_m W_{s,t}(j,N-m) + \frac{1}{N} \left[ d(j+1) - d(j-N+1) \right] \left[ A_m - B_m \right]
\]
\[
W_{s,t}(j+1,N-m) = A_m W_{s,t}(j,N-m) + B_m W_{s,t}(j,m)
\]
\[
+ \sqrt{\frac{1}{N}} \left[ d(j+1) - d(j-N+1) \right] A_m + B_m
\]

where $m = 1, 2, ..., N/2-1$ for even $N$ and $m = 1, 2, ..., (N-1)/2$ for odd $N$ \hspace{1cm} (3.79)

Similar results can be obtained for the remaining transforms (See Appendices D-F). In the interest of brevity the results are listed below for other DWT definitions:

DWT-II:

\[
W_{s,II}(j+1,m) = C_m W_{s,II}(j,m) - D_m W_{s,II}(j,N-m-1)
\]
\[- \sqrt{\frac{1}{N}} \left[ d(j+1) + d(j-N+1) \right] C_m - D_m \]
\[
W_{s,II}(j+1,N-m-1) = C_m W_{s,II}(j,N-m-1) + D_m W_{s,II}(j,m) -
\]
\[- \sqrt{\frac{1}{N}} \left[ d(j+1) + d(j-N+1) \right] C_m + D_m \]

where $m = 1, 2, ..., N/2-1$ for even $N$ and $m = 1, 2, ..., (N-1)/2$ for odd $N$ \hspace{1cm} (3.80)

DWT-III:

\[
W_{s,III}(j+1,m) = A_m W_{s,III}(j,m) + B_m W_{s,III}(j,N-m)
\]
\[+ \sqrt{\frac{2}{N}} \left[ d(j+1) - d(j-N+1) \right] \sin \left( \frac{\pi}{4} - \frac{m\pi}{N} \right) \]
Chapter 3: Running Discrete Transforms Based On Their Shift Properties

\[ W_{s,III}(j+1,N-m) = A_m W_{s,III}(j,N-m) - B_m W_{s,III}(j,m) - \sqrt{\frac{2}{N}} \left[ d(j+1) - d(j-N+1) \right] \cos \left( \frac{\pi}{4} - \frac{m\pi}{N} \right) \]

where \( m = 1, 2, ..., N/2-1 \) for even \( N \) and \( m = 1, 2, ..., (N-1)/2 \) for odd \( N \)  \( (3.81) \)

DWT-IV:

\[ W_{s,IV}(j+1,m) = C_m W_{s,IV}(j,m) + D_m W_{s,IV}(j,N-m-1) - \sqrt{\frac{2}{N}} \left[ d(j+1) + d(j-N+1) \right] \sin \left( \frac{m\pi}{N} + \frac{\pi}{2} - \frac{\pi}{4} \right) \]

\[ W_{s,IV}(j+1,N-m-1) = C_m W_{s,IV}(j,N-m-1) - D_m W_{s,IV}(j,m) + \sqrt{\frac{2}{N}} \left[ d(j+1) + d(j-N+1) \right] \cos \left( \frac{m\pi}{N} + \frac{\pi}{2} - \frac{\pi}{4} \right) \]

where \( m = 1, 2, ..., N/2-1 \) for even \( N \) and \( m = 1, 2, ..., (N-1)/2 \) for odd \( N \)  \( (3.82) \)

Note that the computational burden for updating a DWT using Equations (3.79) to (3.82) is only about half of that required for Equations (3.71) to (3.78).

Clearly real time DWT analysers can be implemented on the basis of the first order shift properties as per Equations (3.79) to (3.82). As an example, we consider the case of DWT-I. Taking the z-transform of Equations (3.79):

\[ z \hat{W}_{s,l}(z,m) = A_m \hat{W}_{s,l}(z,m) - B_m \hat{W}_{s,l}(z,N-m) + \frac{1}{N} (A_m - B_m) z(1-z^{-N}) \hat{D}(z) \]

\[ z \hat{W}_{s,l}(z,N-m) = A_m \hat{W}_{s,l}(z,N-m) + B_k \hat{W}_{s,l}(z,k) + \frac{1}{N} (A_m + B_m) z(1-z^{-N}) \hat{D}(z) \]

where \( m = 1, 2, ..., N/2-1 \) for even \( N \) and \( m = 1, 2, ..., (N-1)/2 \) for odd \( N \)  \( (3.83) \)
where $\hat{W}_{s,l}(z,m)$, $\hat{W}_{s,l}(z,N-m)$ and $\hat{D}(z)$ are the z transforms of $W_{s,l}(j,m)$, $W_{s,l}(j,N-m)$ and $d(j)$ respectively. From Equations (3.83) we obtain:

$$\hat{W}_{s,l}(z,m) = \frac{-B_m z^{-1}}{1 - A_m z^{-1}} \hat{W}_{s,l}(z,N-m) + \frac{1}{N} \frac{1 - z^{-N}}{1 - A_m z^{-1}} \hat{D}(z)$$

$$\hat{W}_{s,l}(z,N-m) = \frac{B_m z^{-1}}{1 - A_m z^{-1}} \hat{W}_{s,l}(z,m) + \frac{1}{N} \frac{1 - z^{-N}}{1 - A_m z^{-1}} \hat{D}(z)$$

where $m = 1, 2, ..., N/2-1$ for even $N$ and $m = 1, 2, ..., (N-1)/2$ for odd $N$  (3.84)

Equation (3.79) can be implemented as the lattice structure depicted in Figure 3.5. A parallel combination of such lattice filter structures can be implemented to provide a real-time DWT-I analyser.

### 3.4.2 Second Order Shift Properties And Their Implementation

As indicated in previous subsections for DCT/DST and DHT, second order shift properties are capable of updating each transform coefficient independently, thus reducing the computational burden. The purpose of this subsection is to develop the similar second order shift properties for DWT's.

Once again we present the derivation of the DWT-I and DCWT-I in detail and list the final results for the other transform members. Taking the z-transform of Equations (3.71) and (3.72) and after some manipulation we have:
Figure 3.5. The real time DWT-I, where \( O_m = \sqrt{\frac{2}{N}} (A_m - B_m) \), \( S_m = \sqrt{\frac{2}{N}} (A_m + B_m) \),

\[
A_m = \cos \left( \frac{2\pi m}{N} \right) \text{ and } B_m = \sin \left( \frac{2\pi m}{N} \right).
\]

\[
\hat{\hat{W}}_{s,l}(z,m) = \frac{-B_m z^{-1}}{1 - A_m z^{-1}} \hat{\hat{W}}_{c,l}(z,m) + \frac{\sqrt{\frac{1}{N} [1 - z^{-N} [A_m - B_m]}}{1 - A_m z^{-1}} \hat{D}(z)
\]

and

\[
\hat{\hat{W}}_{c,l}(z,m) = \frac{B_m z^{-1}}{1 - A_m z^{-1}} \hat{\hat{W}}_{s,l}(z,m) + \frac{\sqrt{\frac{1}{N} [1 - z^{-N} [A_m + B_m]}}{1 - A_m z^{-1}} \hat{D}(z)
\]

Substituting (3.86) into (3.85) we have:
(1 - 2A_m z^{-1} + z^{-2}) \hat{W}_{s,t}(z, m) = \frac{1}{N} \left[ (A_m - B_m) - z^{-1} - (A_m - B_m) z^{-N} + z^{-N+1} \right] \hat{D}(z)

(3.87)

Further, substituting Equation (3.85) into Equation (3.86) results in the following:

(1 - 2A_m z^{-1} + z^{-2}) \hat{W}_{c,t}(z, m) = \frac{1}{N} \left[ (A_m + B_m) - z^{-1} - (A_m + B_m) z^{-N} + z^{-N+1} \right] \hat{D}(z)

(3.88)

Taking the inverse z-transform of Equations (3.87) and (3.88) yields:

\begin{align*}
W_{s,t}(j+1, m) &= 2A_m W_{s,t}(j, m) - W_{s,t}(j-1, m) + \\
&\quad + \frac{1}{N} \left\{ (A_m - B_m) [d(j+1) - d(j-N+1)] - [d(j) - d(j-N)] \right\} 
\end{align*}

(3.89)

and

\begin{align*}
W_{c,t}(j+1, m) &= 2A_m W_{c,t}(j, m) - W_{c,t}(j-1, m) + \\
&\quad + \frac{1}{N} \left\{ (A_m + B_m) [d(j+1) - d(j-N+1)] - [d(j) - d(n-N)] \right\} 
\end{align*}

(3.90)

Clearly Equations (3.89) and (3.90) can be used to independently update the DWT-I or DCWT-I respectively. Note that the recursive equations are of second order and thus are referred to as second order shift properties.

The second order shift properties for the remaining transforms can also be derived as follows (See Appendices D-F):
DWT-II/DCWT-II:

\[ W_{s,II}(j + 1, m) = 2A_m W_{s,II}(j, m) - W_{s,II}(j - 1, m) + \]
\[ + \sqrt{\frac{2}{N}} \left\{ \sin \left( \frac{\pi}{4} - \frac{m\pi}{N} \right) [d(j + 1) - d(j - N + 1)] - \sin \left( \frac{\pi}{4} + \frac{m\pi}{N} \right) [d(j) - d(j - N)] \right\} \]

(3.91)

and

\[ W_{c,II}(j + 1, m) = 2A_m W_{c,II}(j, m) - W_{c,II}(j - 1, m) + \]
\[ + \sqrt{\frac{2}{N}} \left\{ \cos \left( \frac{\pi}{4} - \frac{m\pi}{N} \right) [d(j + 1) - d(j - N + 1)] + \cos \left( \frac{\pi}{4} + \frac{m\pi}{N} \right) [d(j) - d(j - N)] \right\} \]

(3.92)

DWT-III/DCWT-IV:

\[ W_{s,III}(j + 1, m) = 2C_m W_{s,III}(j, m) - W_{s,III}(j - 1, m) - \]
\[ - \sqrt{\frac{2}{N}} \left\{ (C_m - D_m) [d(j + 1) + d(j - N + 1)] - [d(j) + d(j - N)] \right\} \]

(3.93)

and

\[ W_{c,III}(j + 1, m) = 2C_m W_{c,III}(j, m) - W_{c,III}(j - 1, m) - \]
\[ - \sqrt{\frac{2}{N}} \left\{ (C_m + D_m) [d(j + 1) + d(j - N + 1)] - [d(j) + d(j - N)] \right\} \]

(3.94)

DWT-IV/DCWT-IV:

\[ W_{s,IV}(j + 1, m) = 2C_m W_{s,IV}(j, m) - W_{s,IV}(j - 1, m) - \]
\[ - \sqrt{\frac{2}{N}} \left\{ \sin \left( \frac{\pi}{4} - \frac{m + 1}{2} \frac{\pi}{N} \right) [d(j + 1) + d(j - N + 1)] - \right\]
\[ - \sin \left( \frac{\pi}{4} + \frac{m + 1}{2} \frac{\pi}{N} \right) [d(j) + d(j - N)] \right\} \]

(3.95)

and
\[ W_{c,l} (j+1,m) = 2C_m W_{c,l} (j,m) - W_{c,l} (j-1,m) - \]
\[ -\sqrt{2/N} \left[ \cos \left( \frac{\pi}{4} - \left( m + \frac{1}{2} \right) \frac{\pi}{N} \right) \left[ d(j+1) + d(j-N+1) \right] - \right. \]
\[ - \cos \left( \frac{\pi}{4} + \left( m + \frac{1}{2} \right) \frac{\pi}{N} \right) \left[ d(j) + d(j-N) \right] \]
\[ (3.96) \]

As for the previous cases considered, real time DWT analysers can be implemented using the above second order shift properties. Consider the DWT-I as an example. A real time DWT-I analyser can be implemented on the basis of Equation (3.87), which is the z-transform of the second order shift property described by Equation (3.89). The architecture of the resulting system is depicted in Figure 3.6.

Figure 3.6. The real time DWT-I analyser based on its second order shift property,

where \( O_m = \sqrt{\frac{2}{N}} (A_m - B_m) \), \( S_m = \sqrt{\frac{2}{N}} \)
3.4.3 Performance Analysis

A comparison between the computational burden associated with the fast algorithm as proposed by Wang, Z (1984), and the first and second order shift properties proposed above is illustrated in Table 3.4. Clearly both the first order and second order approaches are more efficient in terms of computational burden than the fast algorithm given by Wang, Z. (1984) especially for the case when $N$ is large. The second order shift property approach is the most efficient method in terms of reduced computational burden.

### Table 3.4 Comparison of computational burden

<table>
<thead>
<tr>
<th>Transforms</th>
<th>Multiplication</th>
<th>Additions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWT-I</td>
<td>$(N/4)(3\log_2 N - 13) + 4\log_2 N - 2$</td>
<td>$3N - 4$ for even $N$</td>
</tr>
<tr>
<td>DWT-II</td>
<td>$(3/4)N\log_2 N - 3N/2 + 4$</td>
<td>$3N - 4$ for even $N$</td>
</tr>
<tr>
<td>DWT-III</td>
<td>$(3/4)N\log_2 N - 3N/2 + 4$</td>
<td>$3N - 4$ for even $N$</td>
</tr>
<tr>
<td>DWT-IV</td>
<td>$3N\log_2 N - 3N/4$</td>
<td>$3N - 4$ for even $N$</td>
</tr>
</tbody>
</table>

3.5 Conclusions

In this chapter the shift properties of a number of discrete orthogonal transforms, including discrete Hartley transform, discrete cosine and sine transforms, and discrete W transforms were studied. Both the first order shift properties and the second order shift
properties were also developed and their implementation as real time orthogonal analysers were presented.

The first order shift properties are in the form of first order difference equations. Each of these first order difference equations usually involves two transform coefficients belonging to different transforms (for example, a discrete cosine transform and its corresponding discrete sine transform). This means that two transforms must be updated at the same time. This is a source of extra computational burden. For some transforms such as the DHT and DWT's, this extra computational burden can be eliminated by using the reverse symmetrical properties of the transform coefficients. However, for other transforms such as DCT's and DST's, the computational burden associated with the first order approach is not very efficient.

The second order shift properties are in the form of second order difference equations. The advantage of the second order shift properties is that each second order difference equations involves only one transform coefficient, thus enabling independent updating of every transform coefficient. For transforms such as DCT's and DST's, which are not reverse symmetrical, the second order shift properties provide a more efficient technique in terms of computational burden as well as the hardware implementation.

The system architecture for implementing the shift properties was also considered. The first order shift properties can be realised as a lattice structure, and the second order shift properties correspond to a shift register array structure followed by an IIR filter.
Chapter 4  A Time Domain Interpolation Algorithm For Harmonic Signal Analysis Using DFT
4.1 Introduction

This chapter is concerned with improving the accuracy of harmonic signal analysis. As indicated in Chapter one harmonic analysis is a multifaceted problem in that sometimes all of the parameters for all harmonic components are of interest, sometimes only one or two parameters are of interest. In particular, this chapter considers the harmonic signal analysis using DFT, and the reduction of spectrum leakage associated with the non-synchronous sampling.

Firstly we will endeavour to gain some insight into nature of leakage. Consider a harmonic signal \( z_a(t) \) with period \( T_0 \). Such a signal can be expressed in the form of Fourier series as follows:

\[
Z_a(t) = \sum_{m=0}^{M-1} A_m e^{i2\pi mf_0 t}
\]  

(4.1)

where \( f_0 \) is the fundamental frequency, and \( A_m \) is the amplitude of the \( m \)th harmonic component. Assume that \( z_a(t) \) in Equation (4.1) is sampled with an adequately high sampling rate to avoid aliasing. The resulting discrete signal samples are given by:

\[
z(n) = z_a(nT_s) = \sum_{m=0}^{M-1} A_m e^{i2\pi mf_0 nT_s}, \text{ for } n = 0, 1, 2, ..., N-1
\]

(4.2)

\(^1\text{Note that Equation (4.1) defines an one side Fourier series, which corresponds to a complex signal. In the cases where } d(t) \text{ is real, there is the following expression:}

\[
d_a(t) = \sum_{m=-M+1}^{M-1} A_m e^{i2\pi mf_0 t},
\]

where \( A_m \) is equal to the conjugate of \( A_{-m} \), that is, \( A^*_m = A_{-m} \).
where \( T_s \) and \( f_s \) are the sampling period and the signal frequency respectively. \( N \) is the number of samples during the observation interval or data window. We assume that the observation interval encompasses \( L \) signal cycles. The corresponding \( N \)-point DFT of the sequence \( z(n) \) is given as follows:

\[
Z(k) = \sum_{n=0}^{N-1} z(n)e^{-\frac{2\pi nk}{N}}, \text{ for } k = 0, 1, 2, ..., N-1 \tag{4.3}
\]

Due to the relationship between the DFT and the Fourier series [Oppenheim and Schafer (1975)], the DFT coefficients correspond to the spectrum of \( z(n) \) at the frequencies of integer multiples of \( \frac{f_s}{N} \). In other words, \( Z(0) \) corresponds to the DC component, \( Z(1) \) corresponds to the spectrum at frequency \( \frac{f_s}{N} \), \( Z(2) \) corresponds to the spectrum at frequency \( \frac{2f_s}{N} \), and so on. On the other hand, since the data window covers \( L \) cycles of \( x(n) \), \( Z(1) \) also corresponds to the spectrum of \( z(n) \) at frequency \( \frac{f_0}{L} \), \( Z(2) \) corresponds to \( \frac{2f_0}{L} \), and so on. As a result, the parameters of \( z(n) \) for each harmonic component can be determined by the DFT coefficients subject to the following conditions:

\[
\frac{f_0}{L} = \frac{f_s}{N} \text{ or } LT_0 = NT_s \text{ or } \frac{LT_0}{T_s} = N \tag{4.4}
\]

Equation (4.4) means that the length of data window is an integer multiple of the fundamental period as well as an integer multiple of the sampling period. Equation (4.4) ensures that for each harmonic component, there is one DFT coefficient whose frequency is identical to this harmonic frequency. For example, the fundamental component corresponds to \( Z(L) \), the second harmonic component corresponds to \( Z(2L) \), and so on.
When conditions described by Equation (4.4) are valid, the magnitude of the harmonic component can be determined by the DFT coefficients as follows:

\[
Z(k) = \begin{cases} 
NA_m & \text{for } k = mL, \quad m = 1,2,\ldots, M \\
0 & \text{for other } k 
\end{cases} \quad (4.5)
\]

However, if Equation (4.4) is not satisfied, the result in Equation (4.5) will not be valid. The value of \(Z(mL)\) will not be equal to \(A_m\), and the other DFT coefficients will not be zero, which means that some energy has spilled from the frequency bins where \(k=mL\) into other frequency bins. This undesirable effect is referred to as leakage.

Leakage effect is one of the major difficulties associated with the DFT. Some of the approaches used to reduce the leakage in harmonic analysis include the use of time windows [Harris (1978)], interpolation between the DFT bins [Jain, et al (1979)], and the combination the windowing and interpolation techniques [Grandke (1983), and Audria (1989)].

The motivation of this chapter is to propose an alternate approach to alleviate the leakage problem. One of the desirable features of the proposed approach is that it is continually on-line and only modifies the sequence when synchronisation is lost. In other words, if the samples are properly synchronised with fundamental component then the algorithm has no effect.

The rest of this Chapter is organised as follows: Section 4.2 describes the sampling procedure of a periodic signal and then develops the new algorithm. Section 4.3 presents the performance of the proposed algorithm with application to DFT analysis of harmonics, and Section 4.4 gives the simulation results for the proposed algorithm. In Section 4.5 an alternate application of the proposed algorithm is demonstrated. It is
shown that the proposed algorithm can be used in digital wattmeters to reduce the truncation error. Finally, Section 4.6 concludes the chapter.

### 4.2 The Time Domain Interpolation Algorithm

In this section we consider the derivation of an algorithm to modify the samples when Equation (4.4) is not satisfied. Suppose \[ \left[ \frac{LT_0}{T_s} \right] = N, \] where \([ \ ]\) denotes the operation which yields the closest integer value. Clearly \(N\) represents the number of samples within the data window. We assume that \(N\) is known. Our approach is to modify the practical samples \(z(n)\) towards an ideal sampling sequence \(z_0(n)\) whose sampling period satisfies the condition depicted in Equation (4.4). The ideal sampling frequency is denoted as \(f_{s0}\) and the ideal sampling period is expressed as \(T_{s0}\). Given \(\left[ \frac{LT_0}{T_s} \right] = N\) and \(\frac{LT_0}{T_s} = N\), the ratio \(\frac{f_s}{f_{s0}}\) (or \(\frac{T_{s0}}{T_s}\)) should be close to one for a large \(N\). In this case we assume:

\[
T_s - T_{s0} = \varepsilon \quad \text{and} \quad |\varepsilon| < \frac{T_{s0}}{2N} \quad (4.6)
\]

Without loss of generality it is further assumed that initially \(z(0) = z_0(n)\), which means that the first actual sample is identical to the desired sample. For subsequent samples, there is a time deviation between \(z(n)\) and \(z_0(n)\) due to their different sampling periods, and the deviation increases with \(n\). The time deviation for the second sample is given by Equation (4.6) as \(T_s - T_{s0} = \varepsilon\). For the third sample it is \(2\varepsilon\), and for the \(nth\) sample it will be \((n-1)\varepsilon\). Consequently we have the following relation:

\[
z_0(n) = z_a(nT_{s0}) = z_a(nT_s - n\varepsilon) \quad (4.7)
\]
Expanding Equation (4.7) into a series and neglecting the higher order components, we have:

\[ z_0(n) \approx z_a(nT_s) - n\epsilon \nabla z_a(nT_s) \quad (4.8) \]

where \( \nabla z_a(nT_s) = \left. \frac{dz_a(t)}{dt} \right|_{t=nT_s} \). Considering the periodicity of \( z_a(t) \), the following relationships apply:

\[
\begin{align*}
  z(n-N) &= z_a((n-N)T_s) = z_a[(n-N)(T_{s0} + \epsilon)] = z_a(nT_{s0} - NT_{s0} + n\epsilon - N\epsilon) \\
  &= z_a(nT_{s0} + n\epsilon - N\epsilon) = z_a[n(T_{s0} + \epsilon) - N\epsilon] = z_a(nT_s - N\epsilon) \quad (4.9)
\end{align*}
\]

From the above, the gradient can be approximately evaluated as follows:

\[
\nabla z_a(nT_s) \approx \frac{z_a(nT_s) - z_a(nT_s - N\epsilon)}{N\epsilon} = \frac{z_a(nT_s) - z_a((n-N)T_s)}{N\epsilon} \\
= \frac{1}{N\epsilon} [z(n) - z(n-N)] \quad (4.10)
\]

Substituting Equation (4.10) into Equation (4.8) gives the following formula which can be used to modify the input samples:

\[
z'(n) = z(n) + \frac{n}{N} [z(n-N) - z(n)] , \quad n = 0, 1, \ldots, N-1 \quad (4.11)
\]

It is evident from Equation (4.11) that for each sample, the algorithm only requires one addition and one multiplication. This means that the proposed algorithm is characterised by low computational burden thus making it suitable for on-line implementation.
4.3 Performance Analysis

Now we investigate the case where the proposed algorithm described by Equation (4.11) is applied to the harmonic signal as depicted in Equation (4.1). For simplicity and without loss of generality, we consider an arbitrary $m$th harmonic component of Equation (4.1). The ideal samples corresponding to the $m$th harmonic component are given by:

$$z_{0,m}(n) = z_{a,m}(nT_0) = A_m e^{\frac{2\pi nmL}{N}}$$  \hspace{1cm} (4.12)

from Equation (4.2) the actual samples are obtained as follows:

$$z_m(n) = z_{a,m}(nT_0) = A_m e^{\frac{2\pi nmL}{N}} = A_m e^{\frac{2\pi nmT_s}{T_0}} = A_m e^{\frac{2\pi nmL}{N T_0}} = A_m e^{\frac{2\pi nmL}{N (1 + \frac{e}{T_0})}}$$  \hspace{1cm} (4.13)

We define a sampling error sequence, which is the difference between the actual signal sequences and the ideal signal sequence. Hence the error between the two signal sequences is given by:

$$e_m(n) = A_m \left[ e^{\frac{2\pi nmL}{N (1 + \frac{e}{T_0})}} - e^{\frac{2\pi nmL}{N}} \right] = A_m e^{\frac{2\pi nmL}{N T_0}} \left[ e^{\frac{2\pi nmL}{N T_0}} - 1 \right] = \alpha_{m,n} e^{\frac{2\pi nmL}{N}}$$  \hspace{1cm} (4.14)

where $\alpha_{m,n} = A_m [\beta_m(n) - 1]$ and $\beta_m(n) = e^{\frac{2\pi nmL}{N T_0}}$. For simplicity, we define the sampling error for $m$th harmonic component as follows:

$$\gamma_m = \frac{\alpha_{m,n}}{A_m} = [\beta_m(n) - 1]$$  \hspace{1cm} (4.15)
As indicated in Equation (4.8), \(|e| < \frac{T_{s0}}{N}\), that is, \(|\epsilon| < \frac{1}{2N}\). Assuming \(N >> ML\), we have

\[
\frac{2\pi nmL\epsilon}{NT_{s0}} < \frac{\pi nmL}{N^2} << 1.\]

In this case \(\beta_m(n)\) can be evaluated as:

\[
\beta_m(n) = e^{\frac{2\pi nmL\epsilon}{NT_{s0}}} \approx \cos\left(\frac{2\pi nmL\epsilon}{NT_{s0}}\right) \approx 1 - \frac{1}{2!}\left(\frac{2\pi nmL\epsilon}{NT_{s0}}\right)^2
\]

Hence

\[
\gamma_m(n) = |\beta_m(n) - 1| \approx \frac{1}{2!}\left(\frac{2\pi nmL\epsilon}{NT_{s0}}\right)^2
\]

Now we evaluate the sampling error after the proposed algorithm is applied. The modified samples are given by:

\[
\dot{z}_m'(n) = \dot{z}_m'(n) + \frac{n}{N} [\dot{z}_m'(n - N) - \dot{z}_m'(n)]
\]

\[
= A_m e^{\frac{2\pi nmL}{N} e^{\frac{2\pi nmL}{NT_{s0}}}} \left\{1 + \frac{n}{N} \left[1 - e^{\frac{2\pi nmL}{T_{s0}}}\right]\right\}
\]

Hence the sampling error sequence after modification is given as follows:

\[
e_m'(n) = A_m e^{\frac{2\pi nmL}{N}} \left\{e^{\frac{2\pi nmL}{NT_{s0}}} \left[1 + \frac{n}{N} \left(1 - e^{\frac{2\pi nmL}{T_{s0}}}\right)\right] - 1\right\} = \alpha_m e^{\frac{2\pi nmL}{N}}
\]
where \( \alpha'_{m,n} = A_m[\beta'_{m}(n) - 1] \) and \( \beta'_{m}(n) = e^{\frac{2\pi m L E}{NT_0}} \left[ 1 + \frac{n}{N} \left( 1 - e^{\frac{2\pi m L E}{T_0}} \right) \right]. \) Similarly to Equation (4.15), the relative value of sampling error is defined as:

\[
\gamma'_{m}(n) = \frac{\alpha'_{m,n}}{A_m} = [\beta'_{m}(n) - 1]
\]

Similarly to Equation (4.15), the relative value of sampling error is defined as:

\[
\gamma'_{m}(n) = \frac{\alpha'_{m,n}}{A_m} = [\beta'_{m}(n) - 1]
\]

\( \beta_{m}(n) \) can be approximately evaluated as follows:

\[
\beta_{m}(n) = e^{\frac{2\pi m L E}{NT_0}} \left[ 1 + \frac{n}{N} \left( 1 - e^{\frac{2\pi m L E}{T_0}} \right) \right] = \left[ 1 - \frac{1}{2!} \left( \frac{2\pi m L E}{NT_0} \right) \right] \left[ 1 + \frac{1}{2!} \frac{n}{N} \left( \frac{2\pi m L E}{T_0} \right) \right]
\]

(4.21)

It can be shown from Equation (4.21) that \( \beta'_{m}(n) \) decreases with \( n \). Hence the value of \( \beta'_{m}(n) \) is bounded with the range given by:

\[
\beta'_{m}(N) \leq \beta'_{m}(n) \leq \beta'_{m}(0)
\]

(4.22)

or

\[
\left[ 1 - \frac{1}{(2!)^2} \left( \frac{2\pi m L E}{NT_0} \right)^4 \right] < \beta'_{m}(n) \leq 1
\]

(4.23)

Therefore we have:
Comparing Equation (4.17) and (4.24) we have

\[ 0 < |\beta'_m(n) - 1| \leq \frac{1}{(2!)^2} \left( \frac{2\pi nmL}{NT_{s0}} \right)^4 \approx |\beta_m(n) - 1|^2 \]  

(4.25)

and

\[ \frac{\gamma'_m(n)}{\gamma_m(n)} = \frac{|\beta'_m(n) - 1|}{|\beta_m(n) - 1|} \leq \frac{1}{2!} \left( \frac{2\pi nmL}{NT_{s0}} \right)^2 \]  

(4.26)

Given that \( \frac{|e|}{T_{s0}} < \frac{1}{2N} \), and assuming that \( m << N \) we have

\[ \frac{\gamma'_m(n)}{\gamma_m(n)} \leq \frac{1}{2!} \left( \frac{2\pi nmL}{NT_{s0}} \right)^2 \leq \frac{1}{2!} \left( \frac{\pi nmL}{N^2} \right)^2 << 1 \]  

(4.27)

In other words,

\[ \gamma'_m(n) << \gamma_m(n) \]  

(4.28)

Clearly the residual error after the proposed algorithm has been applied is much smaller than the original error. In other words, the modified samples are much closer to the ideal samples.
It is also interesting to investigate the ideal case where $\varepsilon = 0$. In this case the modified sample sequence is given by:

$$z_i(n) = z(n) = z_0(n), \quad n = 0, 1, 2, \ldots, N-1$$  \hspace{1cm} (4.29)

This means that the proposed algorithm has no effect on the actual sample. This result is very important, since in such cases the actual sample sequence is the same as the ideal sequence, and no modification should be made.

Now we can examine and compare the DFT of the error sequences before and after modification. Firstly, the DFT of $e_m(n)$ is given by:

$$E_m(k) = \sum_{n=0}^{N-1} e_m(n)e^{\frac{2\pi nk}{N}} = A_m \sum_{n=0}^{N-1} \gamma_m(n)e^{\frac{2\pi nk}{N}(Lm-k)}$$  \hspace{1cm} (4.30)

and the DFT of the error sequence after modification is:

$$E'_m(k) = \sum_{n=0}^{N-1} e'_m(n)e^{\frac{2\pi nk}{N}} = A_m \sum_{n=0}^{N-1} \gamma'_m(n)e^{\frac{2\pi nk}{N}(Lm-k)}$$  \hspace{1cm} (4.31)

At this point it is necessary to use the following results:

$$E_m(k) = Z_{0,m}(k) - Z_m(k)$$  \hspace{1cm} (4.32)

and

$$E'_m(k) = Z_{0,m}(k) - Z'_m(k)$$  \hspace{1cm} (4.33)

and

$$Z_{0,m}(Lm) = A_m$$ and $Z_{0,m}(k) = 0$ for $k \neq Lm$  \hspace{1cm} (4.34)
where $Z_m(k)$, $Z'_m(k)$ and $Z_{0,m}(k)$ denote the DFT of $z_m(n)$, $z'_m(n)$ and $z_{0,m}(n)$ respectively. Let us first consider the case where $k \neq Lm$. From Equations (4.32)-(4.34) we have:

\[
Z_m(k) = E_m(k) = A_m \sum_{n=0}^{N-1} \gamma_m(n) e^{\frac{2\pi in}{N}(Lm-k)}
\]  
(4.35)

\[
Z'_m(k) = E'_m(k) = A'_m \sum_{n=0}^{N-1} \gamma'_m(n) e^{\frac{2\pi in}{N}(Lm-k)}
\]  
(4.36)

where $E_m(k)$ and $E'_m(k)$ represent the spectral leakage of $m$th harmonic component before and after the modification respectively. From Equation (4.28) it is expected that the summation of Equation (4.36) will yield a smaller value than Equation (4.35) thus effectively reducing the level of the leakage.

Let us now investigate the case where $k=Lm$. From Equations (4.30)-(4.34) we have:

\[
Z_m(Lm) = A_m + E_m(Lm) = A_m \sum_{n=0}^{N-1} \gamma_m(n)
\]  
(4.37)

and

\[
Z'_m(Lm) = A'_m + E'_m(mL) = A'_m + A_m \sum_{n=0}^{N-1} \gamma'_m(n)
\]  
(4.38)

Hence $A_m \sum_{n=0}^{N-1} \gamma_m(n)$ and $A_m \sum_{n=0}^{N-1} \gamma'_m(n)$ represent the error for estimating $A_m$ by DFT analysis. From Equation (4.28) it is clear that the estimation error can be reduced by the proposed modification algorithm.
4.4 Simulation Results

In the above sections we have proposed a time domain interpolation algorithm, depicted in Equation (4.11), for modifying the signal sequence when the synchronisation is lost. Theoretical analysis shows that it is capable of reducing the leakage. In order to investigate the actual performance of the proposed algorithm, computer simulations have been performed with a number of specific examples. The results are presented in this section.

4.4.1. Leakage Analysis

In order to demonstrate the effect of the proposed algorithm on the leakage, consider the electrical power supply signal given as follows:

\[ z(t) = 6\cos(2\pi ft + 0.1) \tag{4.39} \]

The fundamental frequency is supposed to be 50Hz, the sampling frequency is 1400Hz, and the data window covers only one signal cycle. In this case \( N=28 \). We investigate the situation where the fundamental frequency changes slightly by ±0.5 Hz.

The amount of leakage for the above sine signal can be evaluated by the leakage coefficient proposed in [Danielson and Lanczos (1942)] and is given as follows:

\[
V = \frac{\sum_{k=0, k \neq k_{\max}}^{D-1} |Z(k)|}{|Z_{\max}|} - \frac{\sum_{k=0}^{D-1} |Z(k)| - |Z_{\max}|}{|Z_{\max}|} \tag{4.40}
\]

where \( Z_{\max} \) refers to the maximum DFT bin and \( k_{\max} \) refers to the DFT index where the DFT coefficient exhibits the biggest magnitude, that is \( Z(k_{\max}) = Z_{\max} \). For the given
example $Z(1) = Z_{\text{max}}$. Note that Equation (4.40) represents the portion of spectrum spilled from the actual sinusoid frequency bin into other frequency bins. In order to have deeper insight into the leakage of spectrum to adjacent bins, we introduce the leakage coefficients, given as follows:

$$V_k = \frac{|Z(k)|}{Z_{\text{max}}}, \text{ where } k \neq k_{\text{max}}$$  \hspace{1cm} (4.41)

Clearly $V_k$ represents the portion of spectrum spilled into the $k$th frequency bin. For the given signal in Equation (4.39), we have calculated the leakage coefficients associated with the DFT of the sample sequences before and after modification. The results are illustrated in Figure 4.1 and Figure 4.2 respectively, where $V$ denotes the overall leakage coefficient, $V_0$ is the leakage coefficient at DC component, $V_2$ is the leakage coefficient at the second harmonic frequency bin, and $V_3$ represents the leakage coefficient at the third harmonic frequency bin. It is seen that the proposed algorithm is capable of considerably reducing the leakage in all the cases considered.

![Figure 4.1 Leakage coefficients associated with the DFT of data before modification.](image)
4.4.2. Parameter Estimation

One of the aims of DFT harmonic analysis is to estimate the amplitudes of harmonic signal based on the calculated DFT coefficients. Let us now demonstrate the application of the proposed algorithm to this problem. Consider the following example:

\[ z(n) = 6.0 \cos(2\pi f_0 t) + 1.0 \cos(6\pi f_0 t) \]  

(4.42)

The signal consists of the fundamental component and a third harmonic component. The nominal frequency is still assumed to be 50Hz, the sampling frequency is 1400Hz, the data window still covers one cycle of the fundamental component (that is, N=28). The 28-point DFT is computed and the amplitudes of each of the harmonic components are evaluated on the basis of Equation (4.5). Table 4.1 depicts the results obtained for the amplitude estimates of the various sinusoidal components for both the modified and unmodified cases. Note that \( A_0, A_1, A_2 \) and \( A_3 \) represent the DC, fundamental component, the second harmonic component and the third harmonic component.
respectively. From Equation (4.42) the true values for $A_0$, $A_1$, $A_2$ and $A_3$ are 0.0, 6.0, 0.0 and 1.0 respectively. The results clearly show that the proposed algorithm can improve the estimation accuracy for all of the components.

In order to show the improvement of the estimation accuracy more clearly, we define the estimation error in dB as follows:

$$(\text{Estimation error}) = 10 \log_{10} \frac{|(\text{Estimated value}) - (\text{Actual value})|}{|\text{Actual value}|}$$

The comparison between the estimation errors for $A_1$ and $A_3$ is shown in Figure 4.3 and Figure 4.4 respectively. It is seen that the proposed algorithm can reduce the error by at least 15 dB.

<table>
<thead>
<tr>
<th>Table 4.1 Parameter Estimation Using DFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>f(Hz)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>49.5</td>
</tr>
<tr>
<td>49.8</td>
</tr>
<tr>
<td>50.0</td>
</tr>
<tr>
<td>50.2</td>
</tr>
<tr>
<td>50.5</td>
</tr>
</tbody>
</table>
Figure 4.3 Estimation error of $A_1$ in dB vs fundamental frequency

Figure 4.4 Estimation error of $A_3$ in dB vs fundamental frequency
4.4 Application of the Proposed Algorithm to Digital Wattmeters

Microprocessor based or digital wattmeters [Turgel (1974), Matouka (1982), and Srinivasan (1987)] offer improved accuracy and speed of response over electromechanical instruments for electrical power measurement. Consider the case where the voltage and current signals are given by:

\[
U(t) = \sum_{m=1}^{p} U_m \cos(2\pi f_0 t + \phi_m) \tag{4.43}
\]

and

\[
I(t) = \sum_{m=1}^{p} I_m \cos(2\pi f_0 t + \phi_m) \tag{4.44}
\]

The average power is evaluated as follows:

\[
P_0 = \frac{1}{T_0} \int_{0}^{T_0} U(t)I(t)dt = \frac{1}{2} \sum_{m=1}^{p} U_m I_m \cos(\phi_m - \phi_m) \tag{4.45}
\]

where \(T_0 (=1/f_0)\) is the period of the voltage or current signals. In sampling wattmeters the voltage and current waveforms are regularly sampled at a rate given by \(f_s=1/T_s\). A discrete equation is used to calculate the average power as follows:

\[
P = \frac{1}{N} \sum_{n=0}^{N-1} P(n) \tag{4.46}
\]

where \(P(n) = U(n)I(n)\), while \(U(n) = U(nT_s)\) and \(I(n) = I(nT_s)\). Hence, the calculated average power is given as:

\[
P = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{i=1}^{p} \sum_{m=1}^{p} U_i I_m \cos\left(\frac{2\pi i f_0}{f_s} n + \phi_i\right) \cos\left(\frac{2\pi m f_0}{f_s} n + \phi_m\right) \tag{4.47}
\]
If \( f_s/f_0 \) is an integer, the calculated average power, \( P \), is equal to the actual average power in Equation (4.45). However, when \( f_s/f_0 \) is not a integer, \( P \) will not be equal to \( P_0 \). The error in this case is called truncation error and is given by:

\[
(\text{Truncation error}) = \frac{|P - P_0|}{P_0}
\]  

(4.48)

Since the instantaneous power \( U(t)I(t) \) is also a periodic function, we can derive the following algorithm for reducing the truncation error:

\[
P'(n) = P(n) + \frac{n}{N}(P(n-N) - P(n)), \quad \text{for } n = 0, 1, \ldots, N-1
\]  

(4.49)

Substituting Equation (4.49) into Equation (4.46) we obtain the new algorithm to calculate the average power:

\[
P' = \frac{1}{N} \sum_{n=0}^{N-1} P'(n) = \frac{1}{N} \sum_{n=0}^{N-1} \left\{ P(n) + \frac{n}{N}(P(n-N) - P(n)) \right\}
\]  

(4.50)

In order to investigate the performance of the proposed algorithm, computer simulations are performed for a specific example which is described as follows. The voltage signal is given by:

\[
U(t) = 100 \sin(100\pi t + 35^\circ) + 15 \sin(300\pi t + 7^\circ) + 10 \sin(500\pi t + 15^\circ)
\]  

(4.51)

and the current signal is as follows:

\[
I(t) = 100 \sin(100\pi t + 25^\circ) + 8 \sin(300\pi t + 14^\circ) + 4 \sin(500\pi t + 7^\circ) + 2 \sin(700\pi t + 15^\circ)
\]  

(4.52)

Where \( f=50Hz \) and \( f_s=6400Hz \), and so \( N=128 \). Consider the case when the signal frequency deviates slightly from 50Hz. The results of truncation error for the proposed algorithm are illustrated in Table 4.2. As can be seen, the proposed algorithm can
significantly reduce the truncation error when compared to the direct computation approach.

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Truncation error without the proposed pre-processing algorithm</th>
<th>Truncation error with the proposed pre-processing algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>49.5</td>
<td>0.47</td>
<td>0.20</td>
</tr>
<tr>
<td>49.8</td>
<td>0.16</td>
<td>0.03</td>
</tr>
<tr>
<td>50.0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>50.2</td>
<td>0.13</td>
<td>0.08</td>
</tr>
<tr>
<td>50.5</td>
<td>0.24</td>
<td>0.23</td>
</tr>
</tbody>
</table>

4.6. Conclusions

This chapter has proposed a preprocessing algorithm for harmonic signals, where synchronised sampling is desired. The proposed algorithm intends to modify the actual signal sequence towards an ideal signal sequence which results from synchronised sampling. The modification is based on the first order interpolation technique. Theoretical analysis indicated that the modified sequence resulting from the proposed pre-processing algorithm yields significantly improved result. To illustrate the application of the proposed algorithm, DFT analysis of harmonic signals and digital wattmeter for electrical power were considered. It was clearly shown by computer simulations that the proposed algorithm is capable of reducing both the DFT leakage in harmonic analysis and the truncation error in digital wattmeters.
Chapter 5  Frequency Estimation Of Harmonic Signals Based On Adaptive IIR Filtering
5.1 Introduction

This chapter is concerned with the issue of harmonic signal analysis from a filtering point of view. In particular, it considers the task of frequency estimation of harmonic signal buried in noise using adaptive Infinite Impulse Response (IIR) filtering. The objective is to present a new block-gradient based adaptive algorithm in order to improve the accuracy of harmonic frequency estimation.

Generally speaking notch filters are characterised by a constant gain (for example, unity gain) at all frequencies, except at the notch frequencies where the gain is zero. In other words, the amplitude frequency response of a notch filter has narrow and deep "nulls" at the notch frequencies. This property enables the notch filter to eliminate the sinusoids embedded in a broadband signal.

Adaptive notch filtering has been an active research area during the last two decades. The early work on adaptive notch filtering dealt mostly with the adaptive Finite Impulse Response (FIR) notch filtering. For example, the now famous paper by Widrow et al. (1975) proposed a structure for adaptive notch filter, as an application of the adaptive noise canceller proposed in the paper. The structure has only one notch and so can only be applied to eliminate one sinusoid interference. Glover (1977) extended the adaptive notch filter structure proposed by Widrow et al. (1975) to multiple sinusoidal interference case. These adaptive notch filters are FIR transversal filter consisting of an N-stage tapped delay line. This FIR scheme is very robust from a stability viewpoint. However, it often requires a large filter length in order to achieve good performance.

During the last ten years adaptive IIR notch filtering has attracted increasing interest due to its potential benefits, such as the simplicity of its structure and the reduction of computational burden. The work on adaptive IIR notch filters has been focused on a
number of issues including: optimal filter parametisation, adaptive algorithms and their performance. Friedlander and Smith (1984) proposed an IIR notch filter based on the general prediction error method discussed by Ljung (1981) and Soderstrom (1983). The proposed notch filter structure forced the zeros to lie on the unit circle while placing soft constraints on the poles. The recursive maximum likelihood algorithm was used to update the filter coefficients. The proposed adaptive notch filter needs $2n$ adaptive parameters for $n$ notches. At about the same time Rao and Kung (1984) independently proposed a constrained IIR notch filter structure for the retrieval of sinusoids in noise. This contribution was also characterised by $2n$ parameters for $n$ notches. An minimal parameter adaptive notch filter structure was proposed independently by both Nehorai (1985) and Ng (1985,1987). Using the proposed structure, $n$ notches can be characterised by $n$ parameters instead of $2n$ parameters as required by previous designs. This thesis is concerned with the minimal parametisation structure.

The other issue which requires consideration when implementing adaptive IIR filtering is the choice of adaptive algorithm. Since the development of adaptive IIR notch filters many adaptive algorithms have been proposed, such as the Stochastic Gauss Newton (SGN) algorithm proposed by Rao and Kung (1984), the Recursive Maximum Likelihood (RML) algorithm used by Friedlander and Smith (1984) as well as Nehorai (1985), and various gradient-based adaptive algorithms [Shynk (1989)]. The performance of these algorithms was investigated and compared by Chicharo and Ng (1992).

The gradient-based algorithm is one of the most widely used approaches in adaptive IIR filtering [Shynk (1989)]. The steady state error and convergence rate of a gradient-based adaptive algorithm depend to a large on the gradient estimates. In other words, for frequency estimation based on the adaptive IIR filtering using the gradient-based adaptive algorithm, the estimation accuracy depends upon the accuracy of the gradient estimates. In this chapter a new block gradient-based adaptive algorithm for IIR filtering is
proposed. The proposed algorithm estimates the gradients based on sampled data blocks, within which the filter coefficients are kept constant. By contrast with other existing algorithms, such as the Recursive Predict Error (RPE) algorithm [Ljung and Soderstrom (1983)], simplified RPE algorithm [Shynk (1989)], and the Approximate Gradient (AG) algorithm (or pseudo-linear regression algorithm) [Feintuch (1976), Landau (1976)], the proposed algorithm employs more accurate gradient estimates. Consequently, the proposed approach is capable of improving the accuracy of harmonic frequency estimation.

This chapter is organised as follows: Section 5.2 formulates the problem and gives a brief review of a frequency estimation scheme proposed by Chicharo (1989). The gradient-based adaptive algorithm is described in Section 5.3. Section 5.4 presents the new block gradient-based algorithm. Computer simulation results for frequency estimation using the proposed algorithm are given in Section 5.5. Finally, Section 5.6 concludes this chapter.

### 5.2 A Frequency Estimation Scheme Based On Adaptive IIR Notch Filtering

This section gives a brief review of the fundamentals of frequency estimation using adaptive IIR filtering. Consider the signal given as follows:

\[ x(j) = s(j) + n(j) \quad \text{for} \quad j = 0, 1, 2, ... \tag{5.1} \]

where \( n(j) \) is an white noise with zero mean and variance \( \sigma^2 \), while \( s(j) \) is a sum of sinusoids specified by:
Chapter 5: Frequency Estimation based on Adaptive IIR Notch Filtering

\[ s(j) = \sum_{m=1}^{M} A_m \sin(\omega_m j + \phi_m) \]  

(5.2)

where \( \omega_m \) is the normalised angular frequency given by \( \omega_m = \frac{2\pi f_m}{f_s} \); where \( f_m \) is the sinusoidal frequency and \( f_s \) is the sampling frequency. The adaptive notch filter which is considered to estimate the frequencies of the signal in Equation (5.1) is the minimal parameter structure proposed by Nehorai (1985) and Ng (1985, 1987), given as follows:

\[ H(z) = \frac{A(z^{-1})}{A(\alpha z^{-1})} \]  

(5.3)

where

\[ A(z^{-1}) = \prod_{m=1}^{M} (1 + a_m z^{-1} + z^{-2}) \]  

(5.4)

where the relationship between \( a_m \) and notch frequencies is as follows:

\[ a_m = -2 \cos \omega_m, \quad \text{for } m = 1, 2, ..., M \]  

(5.5)

Clearly \( A(z^{-1}) \) has complex conjugate roots which lie on the unit circle in the \( z \)-plane. It can be shown that [Chicharo (1989)] \( H(z) \) will have unity gain everywhere except at the sinusoidal frequencies where the gain are zero for ideal notch filter characteristics (that is; \( \alpha \rightarrow 1 \)).

There are mainly two notch filter model structures which can be used for frequency estimation [Chicharo, 1989] based on the filter parameterization depicted in Equation (5.3). The first one is for applications where two signal sources with correlated noise are available (See Figure 5.1). The second one is the single input model and can be used in situations where only one composite signal source is available (See Figure 5.2). In this
chapter we consider the first structure, since the two input source structure performs much better than the signal input model when the two signal sources have correlated broadband components [Chicharo (1989,1990)]. This frequency estimation scheme is illustrated in Figure 5.1. The system has two inputs, a primary input and a reference input. The primary input signal is composed of the sinusoids buried in noise as depicted in Equation (5.1). The reference input $n_z(j)$ is a sequence which is partially correlated with $n_1(j)$. The desired signal is generated by passing $n_z(j)$ through an adaptive notch filter with the same structure as the one in the primary input signal channel.

In order to accurately estimate sinusoidal frequencies, it is desired that $a_m$ can be adaptively adjusted so that the notches are aligned to the sinusoidal frequencies. The relationship between the mean square output error and $a_m$ (referred to as the error surface) has been investigated [Stoica and Nehorai (1988), Chicharo and Ng (1990)]. It has been shown [Chicharo and Ng (1990)] that when there is a single sinusoid, the error surface is unimodal while for the multiple sinusoids buried in noise, the error surface is multimodal and has multiple minima corresponding to each of the sinusoidal frequencies. Examples of error surface for a single second order notch filters are illustrated in Figures 5.3 and 5.4. Note that the error surface minimum corresponds to the input sinusoidal frequencies. Hence an adaptive algorithm can be used which minimises the system mean square error by adjusting the parameters $a_m$ of the notch filter.

### 5.3 Gradient Based Adaptive Algorithm For Adaptive IIR Filtering

This section presents a brief review of the gradient-based algorithm for adaptive IIR filtering. The objective is to demonstrate the important role that the estimation of gradients play on the convergence rate as well as the steady state error of the IIR filters.
Figure 5.1 Single Input Source Adaptive IIR Notch Filter Model Structure [Chicharo (1989), Chicharo and Ng (1990)].

Figure 5.2 The Adaptive IIR notch filter proposed in [Chicharo and Ng (1990)].
Consider the output-error adaptive IIR filter depicted in Figure 5.5, which is characterised by the following recursive difference equation [Shynk (1989)]:

$$y(j) = \sum_{m=1}^{N-1} a_m(j) y(j-m) + \sum_{m=1}^{N-1} b_m(j) x(j-m)$$  \hspace{1cm} (5.6)

where \( x(j) \) and \( y(j) \) are the input and output of the adaptive filter respectively. \( a_m(j) \) and \( b_m(j) \) are the filter coefficients, which are adaptively updated to minimise at each instant of time the instantaneous square error given by \( \zeta(j) = e^2(j) \).

One of the simplest and most widely used approaches for performing this minimisation process is a gradient-based adaptive algorithm which is defined as follows [Shynk (1989)]:

$$\theta(j+1) = \theta(j) - \lambda \nabla \zeta(j) = \theta(j) - 2\lambda e(j) \nabla e(j) = \theta(j) - 2\lambda e(j) \nabla y(j)$$  \hspace{1cm} (5.7)

where \( \nabla \) is the gradient operator, \( \lambda \) is the step size and \( \theta(j) \) is the coefficient vector given by:

$$\theta(j) = [a_1(j) \hspace{0.5cm} a_2(j) \hspace{0.5cm} \cdots \hspace{0.5cm} a_N(j) \hspace{0.5cm} b_0(j) \hspace{0.5cm} b_1(j) \hspace{0.5cm} \cdots \hspace{0.5cm} b_{M-1}(j)]^T$$  \hspace{1cm} (5.8)

where the superscript "T" denotes the transpose. From Equation (5.6) it is clear that we need to evaluate the following gradient:

$$[\nabla y(j)]^T = \left[ \frac{\partial y(j)}{\partial a_1(j)} \hspace{0.5cm} \cdots \hspace{0.5cm} \frac{\partial y(j)}{\partial a_N(j)} \hspace{0.5cm} \frac{\partial y(j)}{\partial b_0(j)} \hspace{0.5cm} \cdots \hspace{0.5cm} \frac{\partial y(j)}{\partial b_{M-1}(j)} \right]$$  \hspace{1cm} (5.9)
Figure 5.3 Error Surface for the second order notch filter model with a single sinusoid in additive noise. (Figure 4.2 on p.78 [Chicharo (1989)])
Figure 5.4 Error Surface for a second order notch filter model with two sinusoids in additive noise. (Figure 4.8 on p.99 [Chicharo (1989)])
Taking the derivative of both sides of the Equation (5.6) with respect to $a_k(j)$ gives:

$$\frac{\partial y(j)}{\partial a_k(j)} = y(j-k) + \sum_{m=1}^{N-1} a_m(j) \frac{\partial y(j-m)}{\partial a_k(j)}$$ (5.10)

Similarly, the derivative with respect to $b_k(j)$ is:

$$\frac{\partial y(j)}{\partial b_k(j)} = x(j-k) + \sum_{m=1}^{N-1} a_m(j) \frac{\partial y(j-m)}{\partial b_k(j)}$$ (5.11)

Since the previous outputs depend on previous coefficients, which in turn are related to the current coefficients via successive updates of the algorithm, the partial derivatives on the right-hand side of the Equations (5.10) and (5.11) are:

$$\frac{\partial y(j-m)}{\partial a_k(j)} = \sum_{m=1}^{N-1} \frac{\partial y(j-m)}{\partial a_m(j-m)} \frac{\partial a_m(j-m)}{\partial a_k(j)} + \sum_{m=1}^{N-1} \frac{\partial y(j-m)}{\partial b_m(j-m)} \frac{\partial b_m(j-m)}{\partial a_k(j)}$$ (5.12)
and

\[
\frac{\partial y(j-m)}{\partial b_k(j)} = \sum_{m=1}^{N-1} \frac{\partial y(j-m)}{\partial a_m(j-m)} \frac{\partial a_m(j-m)}{\partial b_k(j)} + \sum_{m=1}^{N-1} \frac{\partial y(j-m)}{\partial b_m(j-m)} \frac{\partial b_m(j-m)}{\partial b_k(j)}
\]  

(5.13)

It is difficult to evaluate the derivatives \(\frac{\partial a_m(j-m)}{\partial a_k(j)}\), \(\frac{\partial a_m(j-m)}{\partial b_k(j)}\), \(\frac{\partial b_m(j-m)}{\partial a_k(j)}\) and \(\frac{\partial b_m(j-m)}{\partial b_k(j)}\) due to the recursive nature of successive coefficients. The Recursive Prediction Error (RPE) algorithm [Ljung and Soderstrom (1983)] was developed based on the assumption that the coefficients vary slowly and can be considered as fixed during a period of time \([j-L-1, j-L \ldots j]\), that is:

\[
a_k(j) = a_k(j-1) = \ldots = a_k(j-L) \quad \text{and} \quad b_k(j) = b_k(j-1) = \ldots = b_k(j-L)
\]

(5.14)

Under such circumstances, Equations (5.10) and (5.11) can be rewritten as:

\[
\frac{\partial y(j)}{\partial a_k(j)} = y(j-k) + \sum_{m=1}^{N-1} a_m(j) \frac{\partial y(j-m)}{\partial a_k(j-m)}
\]

(5.15)

similarly, the derivative with respect to \(b_k(j)\) is:

\[
\frac{\partial y(j)}{\partial b_k(j)} = x(j-k) + \sum_{m=1}^{N-1} a_m(j) \frac{\partial y(j-m)}{\partial b_k(j-m)}
\]

(5.16)

Using the recursive properties of Equations (5.15) and (5.16), the gradients can be derived as follows:
\[
\frac{\partial y(j)}{\partial a_k(j)} = y(j-k) + \sum_{m=1}^{N-1} a_m(j) \left\{ y(j-m-k) + \sum_{m=1}^{N-1} a_m(j-m) \frac{\partial y(j-m-m_k)}{\partial a_k(j-m-m_k)} \right\}
\]
\[
= y(j-k) + \sum_{m=1}^{N-1} a_m(j) \left\{ y(j-m-k) + \sum_{m=1}^{N-1} a_m(j-m) \left[ y(j-m_{M-1})+...+ \right. \right.
\]
\[
\left. \left. \sum_{m_{M-1}=1}^{N-1} a_{m_{M-1}}(j-m_{M-1}-m_{M-2}) y(j-m_{M-1})+... \right] \right\} \quad (5.17)
\]

and

\[
\frac{\partial y(j)}{\partial b_k(j)} = x(j-k) + \sum_{m=1}^{N-1} a_m(j) \left\{ x(j-m-k) + \sum_{m=1}^{N-1} a_m(j-m) \frac{\partial y(j-m-m_k)}{\partial b_k(j-m-m_k)} \right\}
\]
\[
= x(j-k) + \sum_{m=1}^{N-1} a_m(j) \left\{ x(j-m-k) + \sum_{m=1}^{N-1} a_m(j-m) \left[ x(n-m_{M-1})+...+ \right. \right.
\]
\[
\left. \left. \sum_{m_{M-1}=1}^{N-1} a_{m_{M-1}}(j-m_{M-1}-m_{M-2}) x(j-m_{M-1})+... \right] \right\} \quad (5.18)
\]

Clearly Equations (5.17) and (5.18) represent the gradients for the RPE algorithm. However, Equations (5.17) and (5.18) can not produce the exact values of the gradients, since the coefficients are actually time varying due to the adaptive procedure. Hence it is expected that there will be some performance degradation when the assumption that is inherent in Equation (5.14) is not valid.

### 5.4 A Novel Block Gradient-Based Algorithm

Block-based adaptive algorithms have been proposed for Finite Impulse Response (FIR) filtering whereby the Fast Fourier Transform (FFT) has efficiently performed the filter convolution and the gradient correlation [Dentino, et al. (1978), Clark, et al. (1980) and
(1981), Ferrara, Jr. (1980), Mansour and Gray (1982), Clark, et al. (1983)). These block-based adaptive algorithms reduce the computational complexity because the filter output and the adaptive weights are computed after a block of data has been accumulated [Shynk (1992)]. In addition, for the block based Least Means Square (LMS) algorithm, a more accurate estimate of the true gradient can be obtained [Shynk (1992)]. However, for adaptive IIR filtering, block-based gradient updating strategies have not appeared in literature to the best of our knowledge.

This section develops a block updating strategy for adaptive IIR filtering in order to improve the convergence properties. Consider the case where the input signals are divided into time blocks and the filter coefficients are kept constant within every block. The gradients are evaluated on the basis of the data within each block. These gradients are then used to calculate the new filter coefficients for the next block. In other words, the coefficients are adjusted on a block by block basis. Assuming that the block length is \( L \), then the following relations apply: \( a_k(j) = a_k(j-1) = \ldots = a_k(j-L) = a_k \) and \( b_k(j) = b_k(j-1) = \ldots = b_k(j-L) = b_k \). In this case Equation (5.6) can be rewritten as:

\[
y(j) = \sum_{m=1}^{N-1} a_m y(j-m) + \sum_{m=1}^{N-1} b_m x(j-m)
\]  

(5.19)

Taking the derivative of Equation (5.19) with respect to \( a_k \) gives:

\[
\frac{\partial y(j)}{\partial a_k} = y(j-k) + \sum_{m=1}^{N-1} a_m \frac{\partial y(j-m)}{\partial a_k}
\]  

(5.20)

and the derivative with respect to \( b_k \) is:

\[
\frac{\partial y(j)}{\partial b_k} = x(j-k) + \sum_{m=1}^{N-1} a_m \frac{\partial y(j-m)}{\partial b_k}
\]  

(5.21)
Note that Equations (5.20) and (5.21) are recursive and the derivatives \( \frac{\partial y(j-m)}{\partial a_k} \) and \( \frac{\partial y(j-m)}{\partial b_k} \) can be calculated by substituting \( j \) with \( (j-m) \) in Equations (5.20) and (5.21).

Hence we have:

\[
\frac{\partial y(j)}{\partial a_k} = y(j-k) + \sum_{m=1}^{N-1} a_m \left\{ y(j-m-k) + \sum_{m_1=1}^{N-1} a_{m_1} \frac{\partial y(j-m-m_1)}{\partial a_k} \right\} \tag{5.22}
\]

and

\[
\frac{\partial y(j)}{\partial b_k} = x(j-k) + \sum_{m=1}^{N-1} a_m \left\{ x(j-m-k) + \sum_{m_1=1}^{N-1} a_{m_1} \frac{\partial y(j-m-m_1)}{\partial b_k} \right\} \tag{5.23}
\]

Continuing this procedure we obtain:

\[
\frac{\partial y(j)}{\partial a_k} = y(j-k) + \sum_{m=1}^{N-1} a_m \left\{ y(j-m-k) + \sum_{m_1=1}^{N-1} a_{m_1} \left[ y(j-m-m_1-k) + \ldots + \right. \right.
\]

\[
\ldots + \sum_{m_{M-1}=1}^{N-1} a_{m_{M-1}} \left( y(j-m-m_1-\ldots-m_{(M-1)}) + \sum_{m_M=1}^{N-1} a_{m_M} \frac{\partial y(n-m-m_1-\ldots-m_M)}{\partial a_k} \right) \right\} \tag{5.24}
\]

and

\[
\frac{\partial y(j)}{\partial b_k} = x(j-k) + \sum_{m=1}^{N-1} a_m \left\{ x(j-m-k) + \sum_{m_1=1}^{N-1} a_{m_1} \left[ x(j-m-m_1-k) + \ldots + \right. \right. \right.
\]

\[
\ldots + \sum_{m_{M-1}=1}^{N-1} a_{m_{M-1}} \left( x(j-m-m_1-\ldots-m_{(M-1)}) + \sum_{m_M=1}^{N-1} a_{m_M} \frac{\partial y(j-m-m_1-\ldots-m_M)}{\partial b_k} \right) \right\} \tag{5.25}
\]
Note that the delayed components of $x(j)$ and $y(j)$ in Equations (5.24) and (5.25) should not exceed the range of the block itself. We assume that the last term of Equations (5.24) and (5.25) can be neglected. This is reasonable if we assume that the filter is stable and the data of previous blocks have little effect on the gradients of the current block. It is evident that the gradients in Equations (5.24) and (5.25) are different from the gradients of the RPE algorithm given by Equation (5.17) and (5.18), although they have similar expressions. In deriving the gradients in Equations (5.17) and (5.18), the following assumptions were made:

$$\frac{\partial y(j-m)}{\partial a_k(j)} = \frac{\partial y(j-m)}{\partial a_k(j-m)}$$ and $$\frac{\partial y(j-m)}{\partial a_k(j)} = \frac{\partial y(j-m)}{\partial a_k(j-m)}$$ (5.26)

The above assumptions are in fact not entirely valid since the coefficients are actually time varying due to the adaptive procedure as well as the noisy environment. However, in deriving the gradients in Equations (5.24) and (5.25), the conditions in Equation (5.26) are true, because all the filter coefficients are constant within each block. Hence the gradients in Equations (5.24) and (5.25) are closer to the true gradients than the RPE gradients in Equations (5.17) and (5.18). Theoretically, in the case of very low noise level and stationary signals, more accurate gradients could be achieved by increasing the block length. However, there are two problems associated with the issue of increasing the block length. Firstly, the required computation grows linearly with the block length as well as the number of coefficients. Secondly, the convergence rate reduces, since increasing the block length results in a lower rate of coefficient update. Consequently, a compromise must be made in choosing the block length and its impact on the computational burden and convergence rate.

Now we consider the computational burden associated with the proposed algorithm. Close examination of Equations (5.24) and (5.25) reveals that the proposed algorithm
needs $2LN$ multiplications to compute all of the gradients and $2L$ memory locations are needed to store the previous input and output signals. The computational burden for each update is the same as for the RPE algorithm. However, since the proposed algorithm updates the coefficients only once every $L$ samples, the effective computational burden is only $\frac{1}{L}$ of that required by the RPE algorithm. The proposed BGB algorithm is summarised in Table 5.1.

<table>
<thead>
<tr>
<th>Table 5.1 Summary for the proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Initialise the filter coefficients, inputs and outputs, and step size;</td>
</tr>
<tr>
<td>2. For each new input $x(n), d(n)$</td>
</tr>
<tr>
<td>If the time $n$ is not the end of a block</td>
</tr>
<tr>
<td>Keep the coefficients constant, calculate the output by Equation (5.19).</td>
</tr>
<tr>
<td>If the time $n$ is the end of a block</td>
</tr>
<tr>
<td>Keep the coefficients constant, calculate the output by Equation (5.19).</td>
</tr>
<tr>
<td>Calculate the gradients by Equations (5.24) and (5.25), and then calculate the new coefficients for the next block by Equation (5.7).</td>
</tr>
</tbody>
</table>

5.5 Simulation Results for Sinusoid Frequency Estimation

This section examines the performance of the proposed block gradient-based algorithm for frequency estimation. For illustration purposes, we use the IIR notch filter structure (See Figure 5.2) given in [Chicharo and Ng (1990)], which is characterised as follows:

$$H(z) = \frac{1 + az^{-1} + z^{-2}}{1 + \alpha az^{-1} + \alpha^2 z^{-2}}$$  (5.27)

where $a = -2\cos \frac{2\pi f_0}{f_s}$ is the adaptive parameter which controls the position of the notch frequency while $\alpha$ ($0 \leq \alpha < 1$) is a fixed parameter which determines the notch bandwidth. The primary input signal consists of a sinusoid buried in white Gaussian noise, that is:
The reference input $n_{ij}(j)$ is a Gaussian unit variance sequence which is partially correlated with $n_{i}(j)$. The desired signal is generated by passing $n_{ij}(j)$ through an adaptive notch filter with the same structure as the one in the primary input signal channel (refer to Chicharo and Ng (1990) for more details). The signal-to-noise ratio at the primary channel is 6dB. For the simulations presented here, the parameter $\alpha = 0.9$. The normalised frequency of the sinusoidal is $f_0 = 0.25$, which corresponds to the optimal parameter value $a = 0.0$. Different lengths of data blocks were used in our simulations. It was found that when the block length is greater than 2, the convergence rate reduced considerably while only providing a small improvement in steady state variance.

The proposed algorithm was compared with the RPE algorithm for the case when both methods used the same adaptive step size factor (that is, $\lambda = 0.015$). Table 5.2 shows the statistical results based on 100 independent simulations, where the block length is 2.

<table>
<thead>
<tr>
<th>Time</th>
<th>Estimate variance of the proposed algorithm</th>
<th>Estimate variance of the RPE algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 400$</td>
<td>$2.2 \times 10^{-4}$</td>
<td>$0.7 \times 10^{-3}$</td>
</tr>
<tr>
<td>$n = 500$</td>
<td>$2.9 \times 10^{-5}$</td>
<td>$0.8 \times 10^{-3}$</td>
</tr>
<tr>
<td>$n = 600$</td>
<td>$1.1 \times 10^{-5}$</td>
<td>$1.1 \times 10^{-3}$</td>
</tr>
<tr>
<td>$n = 700$</td>
<td>$1.2 \times 10^{-5}$</td>
<td>$0.9 \times 10^{-3}$</td>
</tr>
<tr>
<td>$n = 800$</td>
<td>$1.5 \times 10^{-5}$</td>
<td>$1.8 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

The Cramer-Rao lower bound for this notch structure was investigated by Chicharo (1989). For this above case, the Cramer-Rao lower bound for variance could be less than
140dB (that is, less than $10^{-7}$). From Table 5.2 it is evident that the proposed algorithm is much closer to the Cramer-Rao lower bound when compared with the RPE algorithm. A typical parameter estimate trajectory is illustrated in Figure 5.6. Note that in the condition of the same step size the convergence rate of the proposed approach is slower than the RPE algorithm as expected since updates are only performed on a block by block basis.

Figure 5.6. Parameter estimation trajectories for adaptive notch filtering: Comparison between the proposed algorithm and the RPE algorithm in the case of the same step size.
It can be argued that the above comparison of the steady state variance is not fair since the convergence rates are different for both methods. For this reason the simulations were repeated for the case where the RPE algorithm exhibits the same convergence rate as the proposed algorithm. This was achieved by reducing the adaptive step size factor of the RPE algorithm until the desired convergence rate was approximately achieved (that is, $\lambda = 0.0075$). The statistical results are illustrated in Table 5.3, while a typical parameter estimate trajectory is shown in Figure 5.7. Clearly the variance of the proposed algorithm is still significantly lower than the RPE algorithm.

<table>
<thead>
<tr>
<th>Time</th>
<th>The estimate variance of the proposed algorithm</th>
<th>The estimate variance of the RPE algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 500</td>
<td>$2.2 \times 10^{-4}$</td>
<td>$0.3 \times 10^{-3}$</td>
</tr>
<tr>
<td>n = 500</td>
<td>$2.9 \times 10^{-5}$</td>
<td>$0.1 \times 10^{-3}$</td>
</tr>
<tr>
<td>n = 600</td>
<td>$1.1 \times 10^{-5}$</td>
<td>$0.3 \times 10^{-3}$</td>
</tr>
<tr>
<td>n = 700</td>
<td>$1.2 \times 10^{-5}$</td>
<td>$0.3 \times 10^{-3}$</td>
</tr>
<tr>
<td>n = 800</td>
<td>$1.5 \times 10^{-5}$</td>
<td>$0.5 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 5.3 The steady state variance comparison
5.6 Conclusions

This chapter investigated the use of adaptive IIR filtering in sinusoidal frequency estimation. A new block gradient-based algorithm for adaptive IIR filtering is proposed in order to improve the convergence properties. The proposed algorithm has certain advantages over the conventional RPE algorithm. Firstly, it uses more accurate estimates of the gradients, which means that it provides better steady state error performance. Secondly, because the filter coefficients are updated on a block by block basis, the computational burden is considerably less than the RPE algorithm. It was also shown that for the same convergence rate, the proposed algorithm yields a reduction in steady state variance when compared to the RPE algorithm.
Chapter 6 Conclusions And Suggestions For Further Research
6.1 Conclusions

The main focus of this thesis was aimed at improving the efficiency and accuracy of some spectral analysis techniques and their applications. Extensive work has been performed which included the performance evaluation of some existing techniques, and the development of a number of new techniques. In particular, we considered two main issues. One dealt with improving the efficiency of discrete orthogonal transforms in terms of both computational burden and hardware implementation. This included developing techniques for computing various discrete orthogonal transforms based on adaptive filtering (Chapter 2), and the running computation of discrete orthogonal transforms based on their shift properties (Chapter 3). The other main aspect considered by the thesis dealt with improving the accuracy of the spectral analysis of harmonic signals. This was achieved by reducing the spectral leakage associated with the DFT harmonic analysis (Chapter 4) through the development of a pre-processing algorithm. In addition, a block gradient-based adaptive IIR notch filtering algorithm was proposed for sinusoidal frequency estimation (Chapter 5) which yields improved steady state error performance and lower computational burden when compared with the RPE algorithm.

Chapter 2 concentrated on computing orthogonal transforms using adaptive filtering techniques. A number of approaches for implementing orthogonal analysers were investigated. Firstly, the LMS-based techniques for computing the block based discrete orthogonal transforms were studied. It was shown that a sufficient condition for implementation is that the transform kernel matrix should be orthogonal and unitary. This means that most existing discrete transforms, such as the DFT, DCT, DST, DHT, and DWT can be computed using the LMS algorithm. For illustration purpose, the performance of two LMS based discrete transforms, the LMS-based DCT and the-LMS based discrete Walsh transform, were studied. Secondly, the issue of computing
running discrete orthogonal transforms based on the LMS algorithm was studied. The LMS-based running DHT, DCT, DST and DWT were proposed.

Chapter 2 also examined orthogonal analysers based on other adaptive filtering algorithms. The first one considered was the adaptive Sampling Matrix Inversion (SMI) algorithm [Monzingo and Miller, 1980], which was found to be suitable for performing all orthogonal transforms. The second method considered was the adaptive Howells-Applebaum loop [Applebaum, 1976], which was shown to be applicable for spectral analysis of time continuous signals.

Chapter 3 was concerned with computing the running discrete transforms based on their shift properties. The shift properties of these transforms were developed based on recursive equations that connect the previous and updated transform coefficients. A number of discrete orthogonal transforms, including DHT, DCT's/DST's, and DWT were examined. Two classes of recursive equations were derived for each of these transforms.

The first class of recursive expressions were based on the first order difference equations, which are referred to as the first order shift properties. These first order difference equations provide the connection between the updated transform coefficients at time $j$ and the transform coefficients at time $j-1$, and thus can be used to update the transform coefficients for every new sample. However, each of these first order difference equations usually involves determining two transform coefficients of two different transforms (for example, a discrete cosine transform and its corresponding discrete sine transform). This means that two transforms must be updated at the same time. This is a source of extra computational burden. For some transforms such as the DHT and DWT's, this extra computational burden can be eliminated by using the reverse symmetrical properties of the transform coefficients. However, for other transforms
such as DCT's and DST's, the computational burden associated with the first order shift properties is not very efficient.

The second class of recursive expressions developed were based on the second order difference equations, which are referred to as the second order shift properties. The advantage of the second order shift properties is that each second order difference equations involves only one transform coefficients, thus enabling independent updating of every transform coefficient. The implementation of the shift properties was also investigated. For transforms such as DCT's and DST's, which are not reverse symmetrical, the second order shift properties provide a more efficient technique in terms of computational burden as well as simpler hardware implementation.

The system architecture for implementing the shift properties was also discussed in Chapter 3. It was shown that the first order shift properties can be realised as a lattice structure, and that the second order shift properties correspond to a shift register array followed by an IIR filter structure.

Chapter 4 presented a new approach for remedying the problem of leakage effects associated with DFT-based periodic signal analysis. The basic idea of the proposed method is to modify the practical sample sequence towards an ideal sample sequence whose sampling frequency is an integer multiple of the signal fundamental frequency. A simple formula for modifying the practical samples was derived using time domain interpolation based on the first order Taylor's series. The proposed approach can be used as a real time pre-processing algorithm which only modifies the sequence when synchronisation is lost. If the samples are properly synchronised with fundamental component, the pre-processing algorithm has no effect. Computer simulations of the proposed algorithm with application to DFT analysis of harmonic signals and digital wattmeter for electrical power measurement were performed. The results showed that
the algorithm can reduce both the DFT leakage and the truncation error associated with digital wattmeters.

Sinusoidal frequency estimation techniques based on adaptive IIR notch filtering were considered in Chapter 5. The objective was to improve the steady state error performance. A new adaptive procedure was proposed and referred to as a block gradient-based algorithm, which provides more accurate estimates of the gradients. The input signals are divided into data blocks and the coefficients are kept constant within every block. The gradients are evaluated based on the data of each block and subsequently used to update the coefficients for the next block. In other words, the proposed algorithm updates the coefficients on a block by block basis. Computer simulations were presented using the notch filter structure initially proposed by Chicharo and Ng [1990] together with the proposed block gradient-based algorithm. It is shown that, by contrast with the conventional Recursive Prediction Error (RPE) algorithm, the proposed approach is characterised by less steady state error and lower computational complexity.

6.2 Suggestions for Further Research

This section outlines some of the issues which need further investigation. Naturally the list of issues outlined below is not exhaustive but represents instead a number of interesting areas which deserve further research effort.

6.2.1 Orthogonal Analysis Based on Other Adaptive Algorithms

In this thesis three adaptive algorithms (the LMS, the SMI and the Howells-Applebaum loop) have been considered for implementing orthogonal transforms or orthogonal analysers. Orthogonal analysers based on other adaptive algorithms are a open issue
which need to be addressed. For example, the least square algorithm and recursive least square algorithms are also possible candidates for implementing orthogonal analysers.

### 6.2.2 Finite Word Length Effects on LMS Based Discrete Orthogonal Transforms

Numerical issues such as finite word length implementation are one of the major factors which affect the accuracy of the transforms. This problem has not been addressed in this thesis. In order to investigate the problem, it will be necessary to employ the approaches and/or results in the literature on the limited-precision error analysis of the LMS algorithms [Cioffi (1987), Ardalan (1986)].

### 6.2.3 Finite Word Length Effect Studies of Running Discrete Orthogonal Transforms Based on Shift Properties

As mentioned in Chapter 3 the shift properties are recursive expressions, whose implementation corresponds to IIR filters. The limited-precision error due to the finite work length may seriously affect the accuracy of the running transforms. This is an issue which needs further consideration.

### 6.2.4 Improvement of The Time Domain Interpolation Algorithm

The time domain interpolation algorithm proposed in Chapter 4 may be improved further as follows: Firstly, it may be possible to improve the algorithm by using a higher order estimation of the gradient. In other words, a higher order interpolation algorithm may be possible which provides better performance. The second possible improvement of the algorithm is to use a time varying step size factor, $\mu(n)$, in the algorithm as follows:
\[ x_j(n) = x(n) + \mu(n)[x(n + N) - x(n)] \quad (6.1) \]

The proposed algorithm given in Equation (4.11) is a special case of the above algorithm in Equation (6.1) when \( \mu(n) = \frac{n}{N} \), which means that the modification factor is linear function of \( n \). Better performance may be obtained for other choices of \( \mu(n) \) which actually minimise the leakage coefficient.

### 6.2.5 Other Potential Applications of The Time Domain Interpolation Algorithm

The proposed time domain interpolation algorithm may be applied to other periodic signal processing areas where synchronised sampling of the signals is required. One example is biomedical signal processing where many signals are almost periodic, such as electrocardiogram (ECG) [Tompkins (1993)]. The proposed time interpolation algorithm might be used to improve the analysis accuracy.

### 6.2.6 Application of the Proposed Block Gradient Based Adaptive Algorithm to Multiple Sinusoidal Frequency Estimation.

In Chapter 5 only the frequency estimation of a single sinusoid is considered. It remains an open issue to consider the application of the proposed block gradient based algorithm to other adaptive filter structures. For example, it would be interesting to apply block gradient based adaptive algorithm to the parallel structure proposed by Kilani and Chicharo (1994)) which is used to estimate multiple sinusoidal signals.
Appendices
Appendix A

Second Order Shift Properties of DCT-I and DST-I

The first order shift properties of DCT-I and DST-I are given in Equations (3.12a) and (3.12b) as follows:

\[
C_i(j+1,m) = A_m C_i(j,m) + B_m P_m S_i(j,m) + \sqrt{\frac{2}{N}} P_m \left\{ \left( -\frac{1}{\sqrt{2}} \right) A_m d(j-N) + \left( \frac{1}{\sqrt{2}} - 1 \right) d(j-N+1) \right\} \]
\[
+ (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) A_m d(j) + (-1)^m \frac{1}{\sqrt{2}} d(j+1) \}
\]

for \( m = 0, 1, \ldots, N-1 \) \( (A-1) \)

and

\[
S_i(j+1,m) = A_m S_i(j,m) - B_m C_i(j,m) + \sqrt{\frac{2}{N}} B_m \left\{ \frac{1}{\sqrt{2}} d(j-N) - (1 - \frac{1}{\sqrt{2}})(-1)^m d(j) \right\}
\]

for \( m = 0, 1, \ldots, N-1 \) \( (A-2) \)

Taking the z transform of Equations (A-1) and (A-2) yields:

\[
z \hat{C}_i(z,m) = A_m \hat{C}_i(z,m) + B_m P_m \hat{S}_i(z,m) + \sqrt{\frac{2}{N}} P_m \left[ (-1)^m \frac{1}{\sqrt{2}} z + (-1)^m (1 - \frac{1}{\sqrt{2}}) A_m z^{-N+1} - \frac{1}{\sqrt{2}} A_m z^{-N} - z^{-N} \right] \hat{D}(z)
\]

\( (A-3) \)

and
\[ z \hat{S}_f(z,m) = A_m \hat{S}_f(z,m) - B_m \hat{C}_f(z,m) + \sqrt{\frac{2}{N}} B_m \left[ (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) - \frac{1}{\sqrt{2}} z^{-N} \right] \hat{D}(z) \]  

(A-4)

where \( \hat{C}_f(z,m), \hat{S}_f(z,m), \) and \( \hat{D}(z,m) \) are the z transforms of \( C_f(j,m), S_f(j,m) \) and \( d(j) \) respectively. From Equations (A-3) and (A-4) we can obtain:

\[
\hat{C}_f(z,m) = \frac{B_m P_m z^{-1}}{1 - A_m z^{-1}} \hat{S}_f(z,m) + \sqrt{\frac{2}{N}} \frac{P_m z^{-1}}{1 - A_m z^{-1}} \times
\[
\times \left[ (-1)^m \frac{1}{\sqrt{2}} z + (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) A_m + \frac{1}{\sqrt{2}} - 1 \right] \hat{D}(z)
\]

(A-5)

and

\[
\hat{S}_f(z,m) = \frac{-B_m z^{-1}}{1 - A_m z^{-1}} \hat{C}_f(z,m) + \sqrt{\frac{2}{N}} \frac{B_m z^{-1}}{1 - A_m z^{-1}} \times
\]

\[
\times \left[ \frac{1}{\sqrt{2}} z^{-N} - (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) \right] \hat{D}(z)
\]

(A-6)

Substituting (A-6) into (A-5) and after some manipulation we have:

\[
[1 - 2 A_m z^{-1} + (A_m^2 + B_m^2 P_m) z^{-2}] \hat{C}_f(z,m) = \sqrt{\frac{2}{N}} P_m \left[ (-1)^m \frac{1}{\sqrt{2}} - (-1)^m (1 - \sqrt{2}) A_m z^{-1} - \right.
\]

\[
\left. - (1 - \frac{1}{\sqrt{2}}) (-1)^m z^{-2} - (\frac{1}{\sqrt{2}} - 1) A_m z^{-N+1} + \frac{1}{\sqrt{2}} z^{-N-2} \right] \hat{D}(z)
\]

(A-7)

Substituting (A-5) into (A-6) yields:

\[
[1 - 2 A_m z^{-1} + (A_m^2 + B_m^2 P_m) z^{-2}] \hat{S}_f(z,m) = \sqrt{\frac{2}{N}} B_m \left[ (-1)^m \left( \frac{1}{\sqrt{2}} - 1 - \frac{P_m}{\sqrt{2}} \right) z^{-1} + (-1)^m (1 - \frac{1}{\sqrt{2}}) A_m \left( 1 - P_m \right) z^{-2} + \right.
\]

\[
\left. + \frac{1}{\sqrt{2}} (1 - P_m) + P_m \right] z^{-N-1} - \frac{1}{\sqrt{2}} A_m (1 - P_m) z^{-N-2} \right] \hat{D}(z)
\]

(A-8)
Taking the inverse z-transform of Equations (A-7) and (A-8) yields the second order shift properties of the DCT-I and DST-I respectively:

\[ C_l(j+1,m) = 2A_mC_l(j,m) - [A_m^2 + P_mB_m^2]C_l(j-1,m) + \frac{2}{N}P_m \left[ (-1)^m \sqrt{\frac{1}{2}} d(j+1) - \right. \]

\[ -(-1)^m(1-\sqrt{2})A_md(j) - (-1)^m(1-\sqrt{2})d(j-1) + \left( \frac{1}{\sqrt{2}} - 1 \right)d(j - N + 1) - \]

\[ -\left( \sqrt{2} - 1 \right)A_md(j - N) + \frac{1}{\sqrt{2}} d(j - N - 1) \]

(A-9)

and

\[ S_l(j+1,m) = 2A_mS_l(j,m) - [A_m^2 + P_mB_m^2]S_l(j-1,m) + \frac{2}{N}B_m \left[ (-1)^m \left( \frac{1}{\sqrt{2}} - 1 \right) d(j) + \right. \]

\[ +(-1)^m(1-\sqrt{2})A_m(1-P_m)d(j-1) + \left( \frac{1}{\sqrt{2}} - (\frac{1}{\sqrt{2}} - 1)P_m \right)d(j - N) - \]

\[ -\frac{1}{\sqrt{2}}A_m(1-P_m)d(j - N - 1) \]

(A-10)
Appendix B

The Second Order Shift Properties of DCT-III and DST-III

The first order shift properties of DCT-III and DST-III are given in Equations (3.14a) and (3.14b) as follows:

\[ C_{\text{III}}(j+1,m) = E_mC_{\text{III}}(j,m) + F_mS_{\text{III}}(j,m) + \]
\[ + \sqrt{\frac{2}{N}} \left\{ \left( -\frac{1}{\sqrt{2}} \right) E_m d(j-N) + \left( \frac{1}{\sqrt{2}} - 1 \right) d(j-N+1) + (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) F_m d(j) \right\} \]
\[ \text{for } m = 0, 1, \ldots, N-1 \] (B-1)

and

\[ S_{\text{III}}(j+1,m) = E_mS_{\text{III}}(j,m) - F_mC_{\text{III}}(j,m) + \]
\[ + \sqrt{\frac{2}{N}} \left\{ \frac{1}{\sqrt{2}} F_m d(j-N) + (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) E_m d(j) + \frac{(-1)^m}{\sqrt{2}} d(j+1) \right\} \]
\[ \text{for } m = 0, 1, \ldots, N-1 \] (B-2)

Taking the z transform of Equations (B-1) and (B-2) yields:

\[ z\hat{C}_{\text{III}}(z,m) = E_m\hat{C}_{\text{III}}(z,m) + F_m\hat{S}_{\text{III}}(z,m) + \]
\[ + \sqrt{\frac{2}{N}} \left[ \frac{-1}{\sqrt{2}} E_m z^{-N} + \left( \frac{1}{\sqrt{2}} - 1 \right) z^{-N+1} + (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) F_m \right] \hat{D}(z) \] (B-3)

and

\[ z\hat{S}_{\text{III}}(z,m) = E_m\hat{S}_{\text{III}}(z,m) - F_m\hat{C}_{\text{III}}(z,m) + \]
\[ + \sqrt{\frac{2}{N}} \left[ \frac{1}{\sqrt{2}} F_m z^{-N} + (-1)^m \left( 1 - \frac{1}{\sqrt{2}} \right) E_m + \frac{(-1)^m}{\sqrt{2}} \right] \hat{D}(z) \] (B-4)
where $\hat{C}_{iii}(z,m)$, $\hat{S}_{iii}(z,m)$, and $\hat{D}(z,m)$ are the $z$ transforms of $C_{iii}(j,m)$, $S_{iii}(j,m)$ and $d(j)$ respectively. From Equations (B-3) and (B-4) we can obtain:

$$\hat{C}_{iii}(z,m) = \frac{F_m z^{-1}}{1-E_m z^{-1}} \hat{S}_{iii}(z,m) + \sqrt{\frac{2}{N}} \frac{z^{-1}}{1-A_m z^{-1}} \times$$

$$\times \left[ \frac{-1}{\sqrt{2}} E_m z^{-N} + \left( \frac{1}{\sqrt{2}} - 1 \right) z^{-N+1} + \left( -1 \right)^m \left( 1 - \frac{1}{\sqrt{2}} \right) F_m \right] \hat{D}(z) \quad (B-5)$$

and

$$\hat{S}_{iii}(z,m) = \frac{-F_m z^{-1}}{1-A_m z^{-1}} \hat{C}_{iii}(z,m) + \frac{\sqrt{2}}{N} \frac{z^{-1}}{1-A_m z^{-1}} \times$$

$$\times \left[ \frac{1}{\sqrt{2}} F_m z^{-N} + \left( -1 \right)^m \left( 1 - \frac{1}{\sqrt{2}} \right) E_m + \left( -1 \right)^m \frac{(-1)^m}{\sqrt{2}} z \right] \hat{D}(z) \quad (B-6)$$

Substituting (B-6) into (B-5) and after some manipulation we have:

$$[1 - 2E_m z^{-1} + z^{-2}] \hat{C}_{iii}(z,m) =$$

$$= \sqrt{\frac{2}{N}} \left[ \left( -1 \right)^m F_m z^{-1} - \left( \frac{1}{\sqrt{2}} \right) \left( 1 - 1 \right) z^{-N} - \left( \sqrt{2} - 1 \right) E_m z^{-N+1} + \frac{1}{\sqrt{2}} z^{-N-2} \right] \quad (B-7)$$

Substituting (B-5) into (B-6) yields:

$$[1 - 2E_m z^{-1} + z^{-2}] \hat{S}_{iii}(z,m) = \sqrt{\frac{2}{N}} \left[ \left( -1 \right)^m \frac{1}{\sqrt{2}} +$$

$$+( -1 )^m \left( 1 - \frac{1}{\sqrt{2}} \right) F_m - \frac{1}{\sqrt{2}} E_m \right) z^{-1} + \left( -1 \right)^m \left( 1 - \frac{1}{\sqrt{2}} \right) \left( F_m^2 + E_m F_m \right) z^{-2} + \frac{1}{\sqrt{2}} z^{-N-1} \right] \quad (B-8)$$

Taking the inverse $z$-transform of Equations (B-7) and (B-8) yields the second order shift properties of the DCT-III and DST-III respectively:
\[
C_{\mu \mu}(j + 1, m) = 2 E_n C_{\mu \mu}(j, m) - C_{\mu \mu}(j - 1, m) + \sqrt{\frac{2}{N}} \times
\left[ (-1)^m F_n d(j) - \frac{1}{\sqrt{2}} d(j - N + 1) - (\sqrt{2} - 1) E_n d(j - N) + \frac{1}{\sqrt{2}} d(j - N - 1) \right] \quad (B-9)
\]

and

\[
S_{\mu \mu}(j + 1, m) = 2 E_n S_{\mu \mu}(j, m) - S_{\mu \mu}(j - 1, m) + \sqrt{\frac{2}{N}} \left[ (-1)^m \frac{1}{\sqrt{2}} d(j + 1) + \right.
\left. + (-1)^m \left( (1 - \frac{1}{\sqrt{2}}) F_n - \frac{1}{\sqrt{2}} E_n \right) d(j) - (-1)^m (1 - \frac{1}{\sqrt{2}}) (F_n^2 + E_n F_n) d(j - 1) + F_n d(j - N) \right] \quad (B-10)
\]
Appendix C

The Second Order Shift Properties of DCT-IV and DST-IV

The first order shift properties of DCT-IV and DST-IV are given in Equations (3.15a) and (3.15b) as follows:

\[
C_{IV}(j + 1, m) = E_m C_{IV}(j, m) + F_m S_{IV}(j, m) + \sqrt{\frac{2}{N}} \left[ -G_m d(j - N) + (-1)^m H_m d(j) \right]
\]
\[
\text{for } m = 0, 1, \ldots, N-1 \tag{C-1}
\]

and

\[
S_{IV}(j + 1, m) = E_m S_{IV}(j, m) - F_m C_{IV}(j, m) + \sqrt{\frac{2}{N}} \left[ H_m d(j - N) + (-1)^m G_m d(j) \right]
\]
\[
\text{for } m = 0, 1, \ldots, N-1 \tag{C-2}
\]

Taking the z transform of Equations (C-1) and (C-2) yields:

\[
z \hat{C}_{IV}(z, m) = E_m \hat{C}_{IV}(z, m) + F_m \hat{S}_{IV}(z, m) + \sqrt{\frac{2}{N}} \left[ -G_m z^{-N} + (-1)^m H_m \right] \hat{D}(z)
\]
\[
\text{for } m = 0, 1, \ldots, N-1 \tag{C-3}
\]

and

\[
z \hat{S}_{IV}(z, m) = E_m \hat{S}_{IV}(z, m) - F_m \hat{C}_{IV}(z, m) + \sqrt{\frac{2}{N}} \left[ H_m z^{-N} + (-1)^m G_m \right] \hat{D}(z)
\]
\[
\text{for } m = 0, 1, \ldots, N-1 \tag{C-4}
\]

where \( \hat{C}_{IV}(z, m) \), \( \hat{S}_{IV}(z, m) \), and \( \hat{D}(z, m) \) are the z transforms of \( C_{IV}(j, m) \), \( S_{IV}(j, m) \) and \( d(j) \) respectively. From Equations (C-3) and (C-4) we can obtain:

\[
\hat{C}_{IV}(z, m) = \frac{F_m z^{-1}}{1 - E_m z^{-1}} \hat{S}_{IV}(z, m) + \sqrt{\frac{2}{N}} \frac{z^{-1}}{1 - E_m z^{-1}} \left[ -G_m z^{-N} + (-1)^m H_m \right] \hat{D}(z)
\]
\[
\text{for } m = 0, 1, \ldots, N-1 \tag{C-5}
\]
\[ \hat{S}_{iv}(z, m) = \frac{-E_m z^{-1}}{1 - E_m z^{-1}} \hat{C}_{iv}(z, m) + \sqrt{\frac{2}{N}} \frac{z^{-1}}{1 - E_m z^{-1}} \left[H_m z^{-N} + (-1)^m G_m \right] \hat{D}(z) \] (C-6)

Substituting (C-6) into (C-5) and after some manipulation we have:

\[ [1 - 2 E_m z^{-1} + z^{-2}] \hat{C}_{iv}(z, m) = \sqrt{\frac{2}{N}} \left[(-1)^m H_m(z^{-1} - z^{-2}) - G_m(z^{-N-1} - z^{-N-2}) \right] \] (C-7)

Substituting (C-5) into (C-6) yields:

\[ [1 - 2 E_m z^{-1} + z^{-2}] \hat{S}_{iv}(z, m) = \sqrt{\frac{2}{N}} \left[(-1)^m G_m(z^{-1} - z^{-2}) + H_m(z^{-N-1} - z^{-N-2}) \right] \] (C-8)

Taking the inverse z-transform of Equations (C-7) and (C-8) yields the second order shift properties of the DCT-IV and DST-IV respectively:

\[ C_{iv}(j+1, m) = 2 E_m C_{iv}(j, m) - C_{iv}(j-1, m) + \sqrt{\frac{2}{N}} \left[(-1)^m H_m [d(j) - d(j-1)] - G_m [d(j-N) - d(j-1)] \right] \] (C-9)

and

\[ S_{iv}(j+1, m) = 2 E_m S_{iv}(j, m) - S_{iv}(j-1, m) + \sqrt{\frac{2}{N}} \left[(-1)^m G_m [d(j) - d(j-1)] + H_m [d(j-N) - d(j-1)] \right] \] (C-10)
Appendix D
Shift Properties of DWT-II/DCWT-II

First Order Shift Properties

The running DWT-II/DCWT-III for the sequence \([d(j-N+1), d(j-N+2), \ldots, d(j)]\) is defined as follows:

\[
W_{s,H}(j, m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \sin \left( \frac{\pi}{4} + m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \\
\quad m = 0, 1, \ldots, N-1 \tag{D-1}
\]

\[
W_{c,H}(j, m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{\pi}{4} + m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \\
\quad m = 0, 1, \ldots, N-1 \tag{D-2}
\]

Now we consider the running DWT-II at time \(j+1\):

\[
W_{s,H}(j+1, m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+2+n) \sin \left( \frac{\pi}{4} + m(n + \frac{1}{2}) \frac{2\pi}{N} \right) \tag{D-3}
\]

let \(n'=n+1\), Equation (D-3) becomes:

\[
W_{s,H}(j+1, m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \sin \left( \frac{\pi}{4} + (n'-1 + \frac{1}{2}) m \frac{2\pi}{N} \right) \\
= \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \sin \left( \frac{\pi}{4} + (n' + \frac{1}{2}) m \frac{2\pi}{N} - \frac{2\pi m}{N} \right)
\]
\[
\begin{align*}
= & \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \sin\left(\frac{\pi}{4} + (n' + \frac{1}{2})m\frac{2\pi}{N}\right) \cos\left(\frac{2\pi m}{N}\right) - \\
- & \sqrt{\frac{2}{N}} \sum_{n=1}^{N} x(j - N + 1 + n') \cos\left(\frac{\pi}{4} + (n' + \frac{1}{2})m\frac{2\pi}{N}\right) \sin\left(\frac{2\pi m}{N}\right)
\end{align*}
\] (D-4)

Denoting \( A_m = \cos \frac{2\pi m}{N} \) and \( B_m = \sin \frac{2\pi m}{N} \), it follows that,

\[
W_{s,II}(j + 1, m) = A_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n') \sin\left(\frac{\pi}{4} + (n' + \frac{1}{2})m\frac{2\pi}{N}\right) + \\
+ A_m \sqrt{\frac{2}{N}} \sin\frac{\pi}{4} + \frac{m\pi}{N}\left[d(j+1) - d(j - N + 1)\right] \\
- B_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n') \cos\left(\frac{\pi}{4} + (n' + \frac{1}{2})m\frac{2\pi}{N}\right) - \\
- B_m \sqrt{\frac{2}{N}} \cos\frac{\pi}{4} + \frac{m\pi}{N}\left[d(j+1) - d(j - N + 1)\right] \\
= A_m W_{s,II}(j, m) - B_m W_{c,II}(j, m) + \sqrt{\frac{2}{N}} \left[d(j+1) - d(j - N + 1)\right] \sin\frac{\pi}{4} - \frac{m\pi}{N}
\] (D-5)

Similarly for DCWT-I we have

\[
W_{c,II}(j + 1, m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \cos\left(\frac{\pi}{4} + (n' - 1 + \frac{1}{2})m\frac{2\pi}{N}\right)
\]
Appendices

Second Order Shift Properties

Taking the z-transform of Equations (D-5) and (D-6) and after some manipulation we have:

\[ \hat{W}_{s,t} (z, m) = -\frac{B_m z^{-1}}{1 - A_m z^{-1}} \hat{W}_{c,t} (z, m) + \sqrt{\frac{2}{N}} \left[ 1 - z^{-N} \right] \sin \left( \frac{\pi}{4} - \frac{m \pi}{N} \right) \hat{D}(z) \]  

(D-7)

and

\[ \hat{W}_{c,t} (z, m) = \frac{B_m z^{-1}}{1 - A_m z^{-1}} \hat{W}_{s,t} (z, m) + \sqrt{\frac{2}{N}} \left[ 1 - z^{-N} \right] \cos \left( \frac{\pi}{4} - \frac{m \pi}{N} \right) \hat{D}(z) \]  

(D-8)

Substituting (D-8) into (D-7) we have:

\[ (1 - 2 A_m z^{-1} + z^{-2}) \hat{W}_{c,t} (z, m) = \sqrt{\frac{2}{N}} \left[ \sin \left( \frac{\pi}{4} - \frac{m \pi}{N} \right) - \sin \left( \frac{\pi}{4} - \frac{m \pi}{N} \right) z^{-1} - \sin \left( \frac{\pi}{4} - \frac{m \pi}{N} \right) z^{-N} + \sin \left( \frac{\pi}{4} + \frac{m \pi}{N} \right) z^{-(N+1)} \right] \hat{D}(z) \]  

(D-9)
Further, substituting Equation (D-7) into Equation (D-8) results in the following:

\[(1-2A_mz^{-1}+z^{-2})\hat{W}_{c,II}(z,m) = \left[ \frac{2}{N} \cos \left( \frac{\pi - m\pi}{4} \right) - \cos \left( \frac{\pi + m\pi}{4} \right) \right] z^{-N} + \cos \left( \frac{\pi - m\pi}{4} \right) \right] \hat{D}(z) \]

(D-10)

Taking the inverse z-transform of Equations (D-9) and (D-10) yields:

\[W_{s,II}(j+1,m) = 2A_mW_{s,II}(j,m) - W_{s,II}(j-1,m) +\]
\[+ \sqrt{\frac{2}{N}} \left\{ \sin \left( \frac{\pi - m\pi}{4} \right) \left[ d(j+1) - d(j-N+1) \right] - \sin \left( \frac{\pi + m\pi}{4} \right) \left[ d(j) - d(j-N) \right] \right\} \]

(D-11)

and

\[W_{c,II}(j+1,m) = 2A_mW_{c,II}(j,m) - W_{c,II}(j-1,m) +\]
\[+ \sqrt{\frac{2}{N}} \left\{ \cos \left( \frac{\pi - m\pi}{4} \right) \left[ d(j+1) - d(j-N+1) \right] - \cos \left( \frac{\pi + m\pi}{4} \right) \left[ d(j) - d(j-N) \right] \right\} \]

(D-12)

Clearly Equations (D-11) and (D-12) can be used to independently update the DWT-II or DCWT-II respectively.
Appendix E
Shift Properties of DWT-III/DCWT-III

First Order Shift Properties

The running DWT-III/DCWT-III for the sequence \(d(j-N+1), d(j-N+2), \ldots, d(j)\) are given as follows:

\[
W_{s,\text{III}} (j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \sin \left( \frac{\pi}{4} + n(m+\frac{1}{2}) \frac{2\pi}{N} \right) \\
m = 0, 1, \ldots, N-1
\] (E-1)

\[
W_{c,\text{III}} (j,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{\pi}{4} + n(m+\frac{1}{2}) \frac{2\pi}{N} \right) \\
m = 0, 1, \ldots, N-1
\] (E-2)

Now we consider the running DWT-III at time \(j+1\):

\[
W_{s,\text{III}} (j+1,m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+2+n) \sin \left( \frac{\pi}{4} + n(m+\frac{1}{2}) \frac{2\pi}{N} \right) \\
\] (E-3)

let \(n'=n+1\), Equation (E-3) becomes:

\[
W_{s,\text{III}} (j+1,m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \sin \left( \frac{\pi}{4} + (n'-1)(m+\frac{1}{2}) \frac{2\pi}{N} \right) \\
= \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \sin \left( \frac{\pi}{4} + n'(m+\frac{1}{2}) \frac{2\pi}{N} - (m+\frac{1}{2}) \frac{2\pi}{N} \right)
\]
\[
W_{s,III}(j+1, m) = C_m \frac{2}{N} \sum_{n=1}^{N} d(j - N + 1 + n') \sin \left( \frac{\pi}{4} + n'(m + \frac{1}{2}) \frac{2\pi}{N} \right) \cos \left( m + \frac{1}{2} \frac{2\pi}{N} \right) - \\
- C_m \sqrt{\frac{2}{N}} \sin \left( \frac{\pi}{4} \right) \left[ d(j + 1) + d(j - N + 1) \right] - \\
- D_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j - N + 1 + n') \cos \left( \frac{\pi}{4} + n'(m + \frac{1}{2}) \frac{2\pi}{N} \right) + \\
+ D_m \sqrt{\frac{2}{N}} \cos \left( \frac{\pi}{4} \right) \left[ d(j + 1) + d(j - N + 1) \right] \\
= C_m W_{s,III}(j, m) - D_m W_{s,III}(j, m) - \sqrt{\frac{2}{N}} \left[ d(j + 1) + d(j - N + 1) \right] \sin \left( \frac{\pi}{4} - (m + \frac{1}{2}) \frac{2\pi}{N} \right) \\

(E-5)
\]

Similarly for DCWT-III we have
\[
W_{s,III}(j+1, m) = \frac{2}{N} \sum_{n=1}^{N} d(j - N + 1 + n') \cos \left( \frac{\pi}{4} + n'(m + \frac{1}{2}) \frac{2\pi}{N} \right) \\
= \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \cos \left( \frac{\pi}{4} + n'(m + \frac{1}{2}) \frac{2\pi}{N} \right) - (m + \frac{1}{2}) \frac{2\pi}{N} \right) \\
= \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \cos \left( \frac{\pi}{4} + n'(m + \frac{1}{2}) \frac{2\pi}{N} \right) \cos \left( m + \frac{1}{2} \frac{2\pi}{N} \right) + \\
+ \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j - N + 1 + n') \sin \left( \frac{\pi}{4} + n'(m + \frac{1}{2}) \frac{2\pi}{N} \right) \sin \left( m + \frac{1}{2} \frac{2\pi}{N} \right) \
\]
Second Order Shift Properties

Taking the z-transform of Equations (E-5) and (E-6) and after some manipulation we have:

\[
\hat{W}_{c,III}(z, m) = C_m W_{c,III}(j, m) + D_m W_{s,III}(j, m) - \\
\sqrt{\frac{2}{N}} [d(j+1) + d(j-N+1)] \cos \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{2\pi}{N} \right) 
\]

(E-6)

Substituting (E-8) into (E-7) we have:

\[
\hat{W}_{s,III}(z, m) = \frac{-D_m z^{-1}}{1-C_m z^{-1}} \hat{W}_{c,III}(z, m) - \frac{\sqrt{\frac{2}{N}} [1+z^{-N}] \sin \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{2\pi}{N} \right)}{1-C_m z^{-1}} \hat{D}(z) 
\]

(E-7)

and

\[
\hat{W}_{c,III}(z, m) = \frac{D_m z^{-1}}{1-C_m z^{-1}} \hat{W}_{s,III}(z, m) - \frac{\sqrt{\frac{2}{N}} [1+z^{-N}] \cos \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{2\pi}{N} \right)}{1-C_m z^{-1}} \hat{D}(z) 
\]

(E-8)

Substituting (E-8) into (E-7) we have:
Further, substituting Equation (E-7) into Equation (E-8) results in the following:

\[(1 - 2C_mz^{-1} + z^{-2})\hat{W}_{s,III}(z,m) = -\sqrt{\frac{2}{N}}\left[ (C_m - D_m) - z^{-1} + (C_m - D_m)z^{-N} - z^{-(N+1)} \right] \hat{D}(z) \]

(E-9)

Taking the inverse z-transform of Equations (E-9) and (E-10) yields:

\[ W_{s,III}(j+1,m) = 2C_m W_{s,III}(j,m) - W_{s,III}(j-1,m) + \]
\[ -\sqrt{\frac{2}{N}} \left\{ (C_m - D_m) \left[ d(j+1) + d(j-N+1) \right] - \left[ d(j) + d(j-N) \right] \right\} \]

and

\[ W_{c,III}(j+1,m) = 2C_m W_{c,III}(j,m) - W_{c,III}(j-1,m) + \]
\[ -\sqrt{\frac{2}{N}} \left\{ (C_m + D_m) \left[ d(j+1) + d(j-N+1) \right] - \left[ d(j) + d(j-N) \right] \right\} \]

(E-12)

Clearly Equations (E-11) and (E-12) can be used to independently update the DWT-III or DCWT-III respectively.
Appendices

Appendix F

Shift Properties of DWT-IV/DCWT-IV

First Order Shift Properties

The running DWT-IV/DCWT-IV for the sequence \([d(j-N+1), d(j-N+2), \ldots, d(j)]\) are defined as follows:

\[
W_{s,IV}(j, m) = \frac{2}{N} \sum_{n=0}^{N-1} d(j-N+1+n) \sin \left( \frac{\pi}{4} + \frac{1}{2}(m+\frac{1}{2}) \frac{2\pi}{N} \right)
\]

\[
m = 0, 1, \ldots, N-1
\]

(F-1)

\[
W_{c,IV}(j, m) = \frac{2}{N} \sum_{n=0}^{N-1} d(j-N+1+n) \cos \left( \frac{\pi}{4} + \frac{1}{2}(m+\frac{1}{2}) \frac{2\pi}{N} \right)
\]

\[
m = 0, 1, \ldots, N-1
\]

(F-2)

Now we consider the running DWT-III at time \(j+1\):

\[
W_{s,IV}(j+1, m) = \frac{2}{N} \sum_{n=0}^{N-1} d(j-N+2+n) \sin \left( \frac{\pi}{4} + \frac{1}{2}(m+\frac{1}{2}) \frac{2\pi}{N} \right)
\]

(F-3)

let \(n'=n+1\), Equation (F-3) becomes:

\[
W_{s,IV}(j+1, m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \sin \left( \frac{\pi}{4} + \frac{1}{2}(n'-1+\frac{1}{2})(m+\frac{1}{2}) \frac{2\pi}{N} \right)
\]

\[
= \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \sin \left( \frac{\pi}{4} + \frac{1}{2}(n'+\frac{1}{2})(m+\frac{1}{2}) \frac{2\pi}{N} - (m+\frac{1}{2}) \frac{2\pi}{N} \right)
\]
\[
\frac{2}{N} \sum_{n=1}^{N} \cos (m + -) \cos \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right) - \\
\sqrt{2} \sum_{n=1}^{N} \sin (m + -) \sin \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right)
\]

Let \( C_m = \cos \left( m + \frac{1}{2} \frac{2\pi}{N} \right) \) and \( D_m = \sin \left( m + \frac{1}{2} \frac{2\pi}{N} \right) \). Consider the definitions of DWT-IV in Equation (F-1) and DCWT-IV in Equation (F-2), it follows that,

\[
W_{s,iv}(j+1,m) = C_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n') \cos \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right)
\]

\[
- C_m \sqrt{\frac{2}{N}} \sin \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right) \left[ d(j+1) + d(j-N+1) \right]
\]

\[
- D_m \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} d(j-N+1+n') \cos \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right)
\]

\[
+ D_m \sqrt{\frac{2}{N}} \cos \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right) \left[ d(j+1) + d(j-N+1) \right]
\]

\[
= C_m W_{s,iv}(j,m) - D_m W_{c,iv}(j,m) - \sqrt{\frac{2}{N}} [ d(j+1) + d(j-N+1) ] \sin \left( \frac{\pi}{4} \left( m + \frac{1}{2} \frac{2\pi}{N} \right) \right)
\]

Similarly for DCWT-IV we have

\[
W_{c,iv}(j+1,m) = \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \cos \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right)
\]

\[
= \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \cos \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right)
\]

\[
+ C_m \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \cos \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right)
\]

\[
+ D_m \sqrt{\frac{2}{N}} \sum_{n=1}^{N} d(j-N+1+n') \sin \left( \frac{\pi}{4} \left( n + \frac{1}{2} \right) \right)
\]
\[ C_n \sqrt{\frac{2}{N} \sum_{n=0}^{N-1} d(j-N+1+n') \cos \left( \frac{\pi}{4} + (n' + \frac{1}{2})(m + \frac{1}{2}) \frac{2\pi}{N} \right)} + \]

\[-C_m \sqrt{\frac{2}{N} \cos \left( \frac{\pi}{4} + (m+\frac{1}{2}) \frac{\pi}{N} \right) \left[ d(j+1) + d(j-N+1) \right]} \]

\[+D_m \sqrt{\frac{2}{N} \sum_{n=0}^{N-1} x(j-N+1+n') \sin \left( \frac{\pi}{4} + (n' + \frac{1}{2})(m + \frac{1}{2}) \frac{2\pi}{N} \right) \]

\[-D_m \sqrt{\frac{2}{N} \sin \left( \frac{\pi}{4} + (m+\frac{1}{2}) \frac{\pi}{N} \right) \left[ d(j+1) + d(j-N+1) \right]} \]

\[W_{c,iv}(j+1, m) = C_n W_{c,iv}(j, m) + D_m W_{s,iv}(j, m)\]

\[-\sqrt{\frac{2}{N} \left[ d(j+1) + d(j-N+1) \right] \cos \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{\pi}{N} \right)} \]

\[\text{(F-6)}\]

**Second Order Shift Properties**

Taking the z-transform of Equations (F-5) and (F-6) and after some manipulation we have:

\[\hat{W}_{s,iv}(z, m) = -\frac{D_m z^{-1}}{1-C_m z^{-1}} \hat{W}_{c,iv}(z, m) - \frac{\sqrt{\frac{2}{N} [1+z^{-N}] \sin \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{\pi}{N} \right)}}{1-C_m z^{-1}} \hat{D}(z) \]

\[\text{(F-7)}\]

and

\[\hat{W}_{c,iv}(z, m) = \frac{D_m z^{-1}}{1-C_m z^{-1}} \hat{W}_{s,iv}(z, m) - \frac{\sqrt{\frac{2}{N} [1+z^{-N}] \cos \left( \frac{\pi}{4} - (m+\frac{1}{2}) \frac{\pi}{N} \right)}}{1-C_m z^{-1}} \hat{D}(z) \]

\[\text{(F-8)}\]

Substituting (F-8) into (F-7) we have:
(1 - 2C_m z^{-1} + z^{-2}) \hat{W}_{s,iv}(z,m) = \\
= -\sqrt{\frac{2}{N}} \left[ \sin \left( \frac{\pi}{4} - (m + \frac{1}{2}) \frac{\pi}{N} \right) - \sin \left( \frac{\pi}{4} + (m + \frac{1}{2}) \frac{\pi}{N} \right) \right] \hat{D}(z)

(F-9)

Further, substituting Equation (F-7) into Equation (F-8) results in the following:

(1 - 2C_m z^{-1} + z^{-2}) \hat{W}_{c,iv}(z,m) = \\
= -\sqrt{\frac{2}{N}} \left[ \cos \left( \frac{\pi}{4} - (m + \frac{1}{2}) \frac{\pi}{N} \right) - \cos \left( \frac{\pi}{4} + (m + \frac{1}{2}) \frac{\pi}{N} \right) \right] \hat{D}(z)

(F-10)

Taking the inverse z-transform of Equations (F-9) and (F-10) yields:

\begin{align*}
W_{s,iv}(j + 1,m) &= 2C_m W_{s,iv}(j,m) - W_{s,iv}(j - 1,m) - \\
& - \sqrt{\frac{2}{N}} \left[ \sin \left( \frac{\pi}{4} - (m + \frac{1}{2}) \frac{\pi}{N} \right) \left[ d(j + 1) + d(j - N + 1) \right] - \sin \left( \frac{\pi}{4} + (m + \frac{1}{2}) \frac{\pi}{N} \right) \left[ d(j) + d(j - N) \right] \right]
\end{align*}

(F-11)

and

\begin{align*}
W_{c,iv}(j + 1,m) &= 2C_m W_{c,iv}(j,m) - W_{c,iv}(j - 1,m) - \\
& - \sqrt{\frac{2}{N}} \left[ \cos \left( \frac{\pi}{4} - (m + \frac{1}{2}) \frac{\pi}{N} \right) \left[ d(j + 1) + d(j - N + 1) \right] - \cos \left( \frac{\pi}{4} + (m + \frac{1}{2}) \frac{\pi}{N} \right) \left[ d(j) + d(j - N) \right] \right]
\end{align*}

(F-12)

Clearly Equations (F-11) and (F-12) can be used to independently update the DWT-IV or DCWT-IV respectively.
References


Bernoulli, D. (1738), *Hydrodynamics*, Basel, Switzerland, 1738,


Euler, L. (1755), Institutiones Calculi Differentialis, St. Petersburg, Russia, 1755.


