Efficient algorithms in speech coding

John Kostogiannis

University of Wollongong

Recommended Citation

Efficient Algorithms in Speech Coding

A thesis submitted in fulfilment of the
requirements for the award of the degree

Honours Masters of Engineering

from

UNIVERSITY OF WOLLONGONG

by

John Kostogiannis, B.E. (elec)

School of Electrical and Computer Engineering

1998
ACKNOWLEDGMENTS

I would like to thank my supervisor Professor Gary Anido for his support, guidance and encouragement. Dr Ian Burnett deserves special thanks for providing the specialised technical feedback on the material being presented. I am sincerely grateful to Dr David Rowe, who in the initial years gave me the direction to complete such a task. He not only was my co-supervisor in my academic pursuits but juggled the position of being my supervisor in my tenure as a Research Officer at the Department of Electrical and Computer Engineering.

Most importantly I would like to thank my family Vicki and Peter, and my partner Rima, for their love and understanding. Mum has been an example of what it means to never quit and for that I am truly grateful.

John Kostogiannis
ABSTRACT

The need for low complexity speech coding algorithms has emerged due to application driven requirements. This may be attributed to the power consumption constraints placed on hand held mobile communication systems and the Electromagnetic Interference emission requirements placed on all telecommunication products for both home and office use.

Electromagnetic Interference emissions are hardware design specific and become prevalent when faster rated hardware is used. Low complexity speech coding algorithms can be implemented on slower DSP processors, thereby making it easier to meet the emission requirements. Slower DSP processors consume less power than faster processors.

A literature review has revealed several algorithms that are applicable to speech coding and are characterised by high complexity. These have been selected to demonstrate the various complexity reduction techniques proposed by the author. These techniques include:

a) A decimation and interpolation process on a Pitch Determination Algorithm (PDA) which reduces the complexity of the algorithm by a factor of nine. The algorithm is shown to have greater accuracy (13% increase in its hit ratio) and is significantly less complex (four times) than the standard PDA, namely the autocorrelation method.

b) A procedure that enables adaptation of the weights of a filter structure for a portion of the time. This is implemented in an echo canceller and has the capacity to reduce its complexity by 30%, without compromising its performance.
c) The use of Chebyshev polynomial expansions as a method of eliminating trigonometric (cosine) functions which are inefficiently evaluated by DSP processors. This was presented by Kabal in the formulation of Line Spectral Pairs (LSP) and is included due to it being relevant to many other DSP algorithms including an efficient implementation of a Hanning window. In the case of the LSP method using Chebyshev polynomials, the complexity was reduced to a factor of one tenth compared to the conventional method with no implications on quality. The algorithm exhibited a spectral distortion of 0.48dB, which was well within the just noticeable difference (JND) of 1dB.
# TABLE OF CONTENTS

INTRODUCTION.............................................................................................................. 1

1.1 THESIS MOTIVATION.............................................................................................. 1

1.2 THESIS OVERVIEW.............................................................................................. 5

1.3 SUMMARY OF ORIGINAL CONTRIBUTIONS............................................................. 7

1.4 LIST OF PUBLICATIONS......................................................................................... 8

SPEECH CODING SYSTEMS: A REVIEW ........................................................................... 10

2.1 INTRODUCTION ......................................................................................................... 10

2.2 SPEECH CODING OVERVIEW ............................................................................... 11

2.2.1 Speech Quality vs Bit Rate .................................................................................... 12

2.2.2 Computational Complexity.................................................................................... 14

2.2.3 Coding Delay ....................................................................................................... 15

2.3 SPEECH PRODUCTION MODEL.................................................................................. 16

2.4 CHARACTERISTICS OF VOICED SIGNALS............................................................ 18

2.5 LINEAR PREDICTIVE CODING................................................................................ 19

2.6 EXISTING SPEECH CODING TECHNIQUES........................................................... 21

2.6.1 Code Excited Linear Prediction (CELP) ............................................................... 21
2.6.2 Sinusoidal Coding (Frequency domain) ................................................................. 24

2.7 Comparison of Coding Schemes ............................................................................. 26

2.8 Real Time Implementation ..................................................................................... 28

2.9 Issues Concerning the Selected Topics of This Research .................................... 30

  2.9.1 Pitch Estimation ............................................................................................... 30

  2.9.2 LPC Modeling .................................................................................................. 34

  2.9.3 Echo Cancellation .......................................................................................... 37

2.10 Conclusion ........................................................................................................... 41

EFFICIENT IMPLEMENTATION OF A SUPER RESOLUTION PITCH ESTIMATOR ................................................................. 43

3.1 Introduction ............................................................................................................. 43

3.2 Brief Background on the Medan Algorithm ......................................................... 45

3.3 Implementation of the Medan PDA in Speech Coding Applications .................. 49

3.4 Computational Complexity of the Medan Algorithm ......................................... 52

3.5 Solutions to the Complexity Issue ...................................................................... 54

  3.5.1 Decimation ...................................................................................................... 55

  3.5.2 Interpolation via Polyphase vectors ................................................................ 56

  3.5.3 Polynomial Interpolation ................................................................................ 61
3.6 INTRODUCTION OF A PITCH TRACKER ................................................................. 64

3.7 PERFORMANCE EVALUATION OF THE INTERPOLATION TECHNIQUES ............ 69

3.7.1 Defining the performance measurements...................................................... 71

3.7.2 Comparison between the original Medan PDA and the Autocorrelation Method .......................................................... 73

3.7.2 Performance Evaluation of the Modified Medan PDA using Polyphase Vectors ..................................................................................... 74

3.7.4 Performance Evaluation of the Modified Medan PDA using Polynomial Interpolation ........................................................................... 76

3.8 CONCLUSION ................................................................................................. 77

EFFICIENT IMPLEMENTATION OF LINE SPECTRAL PAIRS (LSP)............. 79

4.1 INTRODUCTION ............................................................................................ 79

4.2 LINE SPECTRUM PAIRS ............................................................................. 80

4.3 COMPUTATION OF THE LSP FREQUENCIES ........................................... 83

4.4 FORMULATION OF LSP FREQUENCIES USING CHEBYSHEV POLYNOMIAL EXPANSIONS ................................................................. 86

4.5 CONVERSION OF LSP COEFFICIENTS TO LPC COEFFICIENTS ............ 90

4.6 PERFORMANCE EVALUATION OF RECONSTRUCTING THE LPC COEFFICIENTS USING CHEBYSHEV POLYNOMIALS ............................. 91
4.6.1 Average Spectral Distortion ................................................................. 91

4.6.2 Objective Measure Results ................................................................... 92

4.7 ALGORITHMIC COMPLEXITY OF EVALUATING THE LSPS USING CHEBYSHEV
POLYNOMIALS ............................................................................................... 93

4.8 CONCLUSION .......................................................................................... 96

ECHO CANCELLATION ..................................................................................... 97

5.1 INTRODUCTION ......................................................................................... 97

5.2 ECHO CANCELLER IN A PSTN .............................................................. 99

5.3 ECHO CANCELLATION ALGORITHM ....................................................... 100

5.4 PERFORMANCE EVALUATION OF THE ECHO CANCELLER ................. 104

5.4.1 Methodology in Evaluating the Performance of an Echo Canceller ........ 104

5.4.2 Creating a database of hybrid losses ..................................................... 106

5.4.3 Effect of Gradient Step Adjustment on the Residual, Convergence and Stability .............................................................................................................. 107

5.4.4 Effect of the number of Filter Taps on the Residual and Convergence Time 112

5.4.5 Effect of speech input signals on the Residual ........................................ 114

5.5 OUTCOME OF THE PERFORMANCE EVALUATION OF THE LMS & NLMS
ALGORITHMS .................................................................................................. 115

5.6 PERIPHERALS OF THE ECHO CANCELLER ........................................... 116
## TABLE OF FIGURES

<table>
<thead>
<tr>
<th>Figure 2.1: Source-Filter Model of Speech Production</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.2: Male Voiced Segment: (a) Time Domain, and (b) Frequency Domain</td>
<td>18</td>
</tr>
<tr>
<td>Figure 2.3: LPC Modeling of Male Speech</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.4: CELP Decoder with Adaptive Codebook</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2.5: Telephone Network with respect to the Near-End Hybrid</td>
<td>38</td>
</tr>
<tr>
<td>Figure 3.1: Block diagram on the modified version of Medan's PDA</td>
<td>55</td>
</tr>
<tr>
<td>Figure 3.3: Illustration of the original $\rho(x,y)$ and the decimated version (solid line)</td>
<td>59</td>
</tr>
<tr>
<td>Figure 3.4: Illustration of the original $\rho(x,y)$ and the decimated version (solid line)</td>
<td>60</td>
</tr>
<tr>
<td>Figure 3.5: Comparison of the original $\rho(x,y)$ and an interpolated version using 2nd order polynomial (where $L = 3$)</td>
<td>63</td>
</tr>
<tr>
<td>Figure 3.6: Comparison of the original $\rho(x,y)$ and an interpolated version using 2nd order polynomial (where $L = 4$)</td>
<td>64</td>
</tr>
<tr>
<td>Figure 3.7: Illustration of a transition between unvoiced to voiced region with a pitch analysis window</td>
<td>65</td>
</tr>
<tr>
<td>Figure 3.8: Illustration of a speech signal that has prominent multiples of the pitch period</td>
<td>66</td>
</tr>
<tr>
<td>Figure 3.9: Illustration of a decaying periodicity within the true pitch period</td>
<td>66</td>
</tr>
</tbody>
</table>
Figure 3.10: Performance of the Medan PDA with pitch tracker compared to standard pitch contour................................................................. 68
Figure 3.11: Performance of the Medan PDA without a pitch tracker......................... 69
Figure 3.12: Histogram of the standard pitch contour database........................................ 70
Figure 4.1: Possible root locations for an even order $P'(z)$ and $Q'(z)$ ..................... 88
Figure 5.1: Speech Coder interfacing with the Near-end Hybrid............................. 99
Figure 5.2: Adaptive FIR filter structure................................................................. 101
Figure 5.3: Interface between the DSP Development card and the Speech Processing Testbed...................................................................................... 107
Figure 5.4: Effect of the Adaptation Constant, $\mu$, on the Convergence Time........... 108
Figure 5.5: Effect of the Adaptation Constant, $\mu$, on the Residual attenuation........... 109
Figure 5.6: Effect of the step size, $a$, on the Convergence Time of the NLMS algorithm......................................................................................... 110
Figure 5.7: Effect of the step size, $a$, on the Residual attenuation of the NLMS algorithm......................................................................................... 111
Figure 5.8: Effect of the Filter Tap Length on the Residual Attenuation for the LMS & NLMS Algorithms................................................................. 112
Figure 5.9: Effect of Tap Size on the Convergence Time using the LMS & NLMS Algorithms............................................................................. 114
Figure 5.10: Effect of speech input signals on the Residual Attenuation (dB) for the NLMS & LMS algorithms.................. 115

Figure 5.11: Echo Canceller Block Diagram............................................................... 117

Figure 5.12: Trajectory of the energy in the weights for the canceller and the input signal............................................................... 121

Figure 5.13: Trajectories of the energy in the weights of the modified NLMS and NLMS with hangover and the input signal. ......................... 123

Figure 5.14: Illustrates the movement of the energy in the weights of the canceller's filter when adapting to 100%, 50% and 10% of the input signal................. 130

Figure 5.15: Residual Attenuation as a function of the adaptation time............... 131

Figure 5.16: Convergence as a function of the Adaptation time. ......................... 132

Figure 5.17: Illustration of the complexity of the canceller as a function of adaptation time. ................................................................. 132
## LIST OF TABLES

Table 3.1 Performance evaluation of the original Medan PDA and the Autocorrelation based PDA against the standard pitch contour........................................................ 74

Table 3.2 Performance evaluation of the Modified Medan PDA (using polyphase vectors) and the Modified Medan PDA (with no interpolation techniques) against the standard pitch contour........................................................................ 75

Table 3.3: Comparison of the Modified PDA (using polyphase vectors) and the original Medan PDA........................................................................ 75

Table 3.4 Performance evaluation of the Modified Medan PDA (using polynomial interpolation) against the standard pitch contour ........................................ 76

Table 3.5 Comparison of the Modified PDA (using polynomial interpolation) and the original Medan PDA........................................................................ 77

Table 3.6: Summary of the performance results for all the PDAs against the standard pitch contour........................................................................ 77
CHAPTER 1

INTRODUCTION

1.1 Thesis Motivation

This thesis presents some useful techniques in digital signal processing (DSP), that allow algorithms, which would otherwise be impractical due to their high complexity, to become feasible for speech coding applications. The thesis also contains a significant element of engineering in an attempt to increase the computational efficiency of existing algorithms without compromising performance. It investigates selected algorithms that are crucial in maintaining and in certain cases improving the speech quality, but are characterised by high complexity. The algorithms are developed further, in relation to lowering the complexity, to warrant utilisation in actual commercial applications.

The advent of powerful and affordable DSP processors in recent years has contributed to the volume and diversity of digital voice compression technology that is available in the market place today. Voice compression technology is considered the solution to alleviating the demands placed on our bandwidth resource especially in the rapidly expanding mobile radio communications services. The replacement of our analogue communication systems with digital technology has facilitated the integration of data, voice and video. Many IT managers consider voice compression technology as a “free” option on their telecommunication (data) networks.

The level of research activity in low bit rate speech coding has increased recently due to:
a) standards being formulated for the introduction of global mobile communication services using Low Earth Orbiting satellites (LEOs) before the turn of the century;

b) "half rate" digital communication systems standards with respect to the current digital cellular services being introduced by Europe, Japan and North America;

c) military services wishing to upgrade their existing "old" 2.4kbit/s secure voice communication services to more "modern" 2.4kbit/s technology with promises of better speech quality;

Speech coding and, in the broader sense, source coding deals with the efficient representation of a signal using a finite digital alphabet. It strives to preserve the quality of the signal reconstructed from that finite representation. Historically this has involved the sampling and amplitude quantisation of a signal. This type of technique has resulted in representing speech (transparently) at bit rates of 64kbit/s and greater. A different philosophy is utilised to achieve lower bit rates (8-2.4kbit/s) as compared to the waveform matching approaches used in former. At the lower rates, the speech signal is analysed according to a speech production model, where model parameters are extracted and transmitted instead of the amplitude quantised speech samples. Hybrid approaches involving a combination of parametric and waveform matching techniques have been used to great effect for bit rates ranging between 16-4kbit/s.

Speech coders are characterised by their bit rate versus perceived speech quality attained. Much research has been undertaken in these two characteristics over the last two decades culminating in high quality 16kbit/s coders, such as the Low Delay Codebook Excited Linear Prediction (LDCELP) coder [17] and good quality 2.4-4kbit/s
coders, such as Prototype Waveform Interpolation (PWI) [32] and sinusoidal coders [19]. There are at least two other considerations that must be taken into account and are usually application dependent. These are implementation complexity and coding delay.

Recently, there has been a greater focus on the complexity of speech coding algorithms, due to the constraints placed on power consumption for hand held mobile applications and Electromagnetic Interference (EMI) emission requirements. Lower complexity algorithms can be implemented on slower DSP processors (less EMI) and consume less power.

There is a tired old adage that the lower the bit rate the higher the complexity for a given speech quality [40]. Much of the effort reported by this thesis concentrates on the complexity issue and opposes the above by stating that high quality at a low bit rate does not necessarily mean high complexity.

The selected algorithms of interest are:

a) A super resolution pitch determination algorithm, developed by Medan [27]: This algorithm is used to estimate the long term characteristics (pitch) of a speech signal. Accurate estimates of the pitch parameter are crucial to synthesising high quality, natural sounding speech in parametric coders. Unfortunately this algorithm is characterised by high complexity.

b) Line Spectral Pair (LSP) transformation of the LPC coefficients [13]: LPC coefficients are the weights of the Linear Prediction filter used to model the short term characteristics of speech, namely the spectral envelope. Transforming these coefficients to LSP frequencies enables the spectral parameters to be quantised efficiently, thereby reducing the bit rate. This
algorithm is characterised by trigonometric functions that are inefficiently evaluated on a DSP processor.

c) Line echo canceller: An essential feature for voice compression technology that is connected to the Public Switched Telephone Network (PSTN). The inherent delay characterised by most low bit rate speech coding algorithms distinguishes the echo from the normal sidetone of a telephone and degrades the quality of service. An echo canceller, using adaptive filtering, strives to model the echo and then eliminate it. The complexity of the canceller is dependent on the delay and dispersion of the echo it is attempting to cancel.

This thesis, as mentioned initially, provides some useful techniques that can be applied to the above algorithms to reduce their complexity without compromising their performance. These techniques are evaluated on their complexity reducing capabilities and their ability to maintain the performances of the original algorithms. Performances of the efficient algorithms are evaluated according to objective and subjective measurements. In cases that justify it, the algorithms are evaluated according to guidelines and standards set by governing bodies such as ITU (International Telecommunications Union) formerly known as CCITT. It is necessary that some type of efficiency measure be also introduced to allow a better indication of the degree of complexity. As these algorithms are to be implemented on a DSP processor, the complexity of an algorithm can be measured in the number of Multiply-Accumulates (MAC) it requires. Usually one MAC takes one instruction on a DSP processor.

The remainder of this introduction includes a thesis overview and a summary of original research contributions. It is important to note that most of the work carried out here has
been implemented on a real time low bit rate speech coder realised on a AT&T DSP32C DSP hardware platform which has enjoyed much commercial success [37].

1.2 Thesis Overview

Chapter 2 is concerned mainly with issues relating to the speech production model explaining problems associated with analysing the speech signal and extracting the model parameters. It describes the connection between the speech production model and the principal speech coding algorithms of today. There is a discussion on the trade-offs associated with the four considerations that characterise speech coding, namely quality, bit rate, complexity and delay. Readers with limited knowledge of speech coding but with a background in digital signal processing will be able to appreciate this chapter. It serves as background material assisting the reader in identifying where the algorithms investigated in this thesis can be incorporated in speech coding applications. It must be noted that some of the algorithms may be used in other speech processing applications.

Chapter 3 deals with the representation of the short term characteristics of the speech signal introduced in the previous chapter. This is the most crucial parameter set transmitted in certain speech coders and significantly affects the perceived speech quality in relation to intelligibility. The chapter analyses the traditional way of quantising this parameter set, namely the LPC coefficients transformed to Line Spectral Pair (LSP) frequencies [13]. It also describes the numerical techniques used in improving its implementation complexity [16].

Chapter 4 describes an existing Pitch Determination Algorithm (PDA) that has promised accurate evaluation of the pitch (long term) characteristics in speech signals.
The pitch parameter set in speech coding applications is important to the naturalness and speaker identification of the perceived synthesised speech quality. It is shown that the algorithm, in its current form, is impractical for use in speech coding applications due to its high complexity. The chapter describes various decimation and interpolation techniques implemented by the author to reduce the complexity of the PDA. It evaluates the performance of these techniques against the original PDA and a database consisting of pitch contours, which have been tabulated by observing the pitch of speech signals.

Chapter 5 deals with the problem of applying a low bit rate speech coder to the PSTN. The coding delay associated with speech coders increases the perception of the echo present in the network. The chapter explains clearly how the echo is generated in the network. An adaptive filtering approach is used to cancel the echo. The echo cancellation algorithm, based on a version of the Least Mean Square (LMS) adaptation algorithm and a Finite Impulse Response (FIR) filter structure, in conjunction with a novel technique developed by the author can potentially reduce the implementation complexity by half. A detailed analysis of the canceller's performances versus parameter variations is described.

The thesis concludes with a chapter in which the major results are summarised and original contributions are listed. The main objective of this thesis is to prove that algorithms that are characterised by high complexity can be reduced to a degree that enables them to become feasible options in speech coding applications. This is supported by the results presented throughout this dissertation and are summarised below:
a) The complexity of the PDA proposed by Medan was reduced by a factor of nine with the use of decimation and interpolation techniques.

b) Eliminating the trigonometric evaluations reduced the complexity of transforming LPC coefficients to LSPs and the complexity of the reconversion process. Each cosine function requires 24 instructions on the DSP32C floating point processor [49] and many more on a fixed point processor.

c) The echo canceller's complexity was reduced by 30% without compromising its performance, using a novel technique that enables adaptation of the weights of the FIR filter structure, only half the time.

1.3 Summary of Original Contributions

During the course of this research a number of contributions have been made, some detailed in previously published conferences papers. A brief description of original work is listed below:

a) A technique based on decimation and polynomial interpolation has enabled the complexity of a super resolution PDA developed by Medan [27] to be reduced. This technique outperforms the interpolation method of using polyphase vectors, proposed by Medan. Interpolating using polyphase vectors is ineffective when the initial pitch estimate (determined with a lower resolution) is not within the temporal resolution of the actual pitch period of the speech signal. The polynomial interpolation technique doesn’t rely on any initial pitch estimate.
b) A thorough investigation into the performance of a G.165 line echo canceller. Theoretical results associated with the LMS and NLMS adaptation algorithm and the FIR filter structure are supported by results achieved through practical experimentation. A novel technique is proposed which restricts the updating of the canceller's filter coefficients. This potentially reduces the implementation complexity by 30% and doesn't compromise the ITU Recommendations G.165.

c) The short term spectral parameters are the basis for most speech coding algorithms. The encoding procedure for these includes the construction of Line Spectral Pairs (LSP) from Linear Prediction Coefficients (LPC) due to their superior quantisation and robustness qualities. The procedure involves the extensive use of trigonometric evaluations. Trigonometric calculations undertaken by DSP processors can take significant portions of the computational power. An efficient method in evaluating trigonometric functions based on Chebyshev polynomials reported in [16] is adapted into the encoding of the short term spectral parameters.

d) Both b) and c) have been implemented in a successful commercial product based on the AT&T DSP32C processor. Its main commercial application has been in secure telephone communication systems [37].

1.4 List of Publications

Conference Paper based on the thesis

Conference papers referenced in the thesis


CHAPTER 2

SPEECH CODING SYSTEMS: A REVIEW

2.1 Introduction

This chapter endeavours to introduce the reader to various techniques utilised in low bit rate speech coding, ranging from 8kbits/s to 2kbits/s. These techniques have been selected on the basis that they are indicative of the research direction that has been undertaken by the speech coding community to date. They present and clarify to the reader instances of where the material examined and developed in this thesis can be applied.

Section 2.2 focuses on current speech coding issues; speech quality, computational complexity and coding delay. These issues determine the specifications and overall performance of speech coding algorithms. Two distinct methodologies in modeling speech have evolved over the years relevant to speech coding. They are the time domain technique used in Linear Predictive Coding (LPC) and its frequency domain counterpart utilised in Sinusoidal/Harmonic Coding. Section 2.6 briefly investigates two vocoder implementations, namely CELP [29] and the Sinusoidal coder [19]. Their implementations will offer insight to the computational complexity of speech coding algorithms as well as the procedures used in extracting their respective model parameters. Techniques proposed by the author in Section 2.9, offer solutions in improving the efficiency of selected extraction and transmission processes of certain model parameters. Section 2.9 also presents the problems associated with vocoder technology connected to the Public Switched Telephone Network (PSTN). The
vocoders introduce audible echo into the PSTN due to their coding delay. An efficient echo canceller is proposed which may be incorporated on the same hardware processor as the speech compression algorithm.

2.2 Speech Coding Overview

Parametric and waveform techniques are currently used in speech coding. Parametric or model based coders assume a speech production model and extract parameters from the speech signal that describe that model, updating the model parameters periodically as the characteristics of the speech signal change. A convenient speech production model is described in Section 2.3, namely the source filter model [39]. The analysis of the speech signal and extraction of model parameters can be open loop or closed loop. In closed loop analysis, the model parameters are selected on the criteria that they minimise the difference between the reconstructed speech signal and the original signal. This analysis procedure combines a synthesis stage and is commonly referred to as analysis by synthesis. This type of procedure is utilised in Code Excited Linear Predictive (CELP) coders [29], [17] and is illustrated in Section 2.6.1.

Waveform coders assume no model, but attempt to minimise the error between the original and reconstructed speech waveforms. These include the A-Law and $\mu$-Law companders (ITU G.711). Many low bit rate speech coders use a combination of parametric and waveform matching techniques, and are thus known as hybrid coders (ITU G.726 [59], G.728 [17], G.729 [57]). CELP coders are often referred as hybrid coders since they use an analysis by synthesis procedure.

Signal processing techniques used in speech coding may be based on time or frequency domain processing. Speech signal statistics are non-stationary but can be considered
quasi-stationary over 5-30ms segments. Thus for analysis purposes, speech signals are buffered into blocks of samples, typically 20-30ms long and are analysed on a block by block basis. This allows the coding techniques to exploit redundancies across a block of speech samples to improve coding efficiency, at the expense of introducing delay.

2.2.1 Speech Quality vs Bit Rate

An important consideration in speech coding is speech quality versus bit rate. These two factors directly conflict with each other. Essentially the lower the bit rate (or signal compression ratio), the greater the degradation in speech quality exhibited by the vocoder. The main focus associated with speech coding research today, is to lower the bit rate required for a given level of speech quality. This is witnessed by the development of G.729 to supersede G.728. In this instance, the bit rate is reduced by 50% while maintaining toll quality synthesised speech. Toll quality is defined as containing no audible distortions. In past times, (late 1980’s and early 1990’s), vocoders having bit rates between 4-8kbit/s, such as G.723[60], IS-54 [31], FS1016 CELP [24] and IMBE [43], were usually identified with communications quality speech output, while lower bit rate vocoders such as FS1015 LPC10e [25], running at 2.4kbit/s, exhibited synthetic quality. Communications quality speech is defined as highly intelligible but with just noticeable distortion. The quality of the reconstructed speech is such that different speakers using the system can be identified. This is unlike synthetic quality systems where speakers cannot be easily distinguished due to the large amounts of distortion present. Table 2.1 summarises the speech quality, bit rate and complexity of speech coding systems [50].
Table 2.1: Summary of Speech Coding Systems

<table>
<thead>
<tr>
<th>Compression Standard</th>
<th>Bit Rate (kbit/s)</th>
<th>Fixed Point DSP (MIPS)</th>
<th>Floating Point DSP (MIPS)</th>
<th>Quality</th>
<th>Typical Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>64</td>
<td>0.5</td>
<td>Toll</td>
<td>Toll</td>
<td>Network Telephony</td>
</tr>
<tr>
<td>G.726</td>
<td>24,32,40</td>
<td>23</td>
<td>8.5</td>
<td>considerable distortion to Toll</td>
<td>Network Telephony</td>
</tr>
<tr>
<td>G728</td>
<td>16</td>
<td>37</td>
<td>20</td>
<td>Toll</td>
<td>Network Telephony</td>
</tr>
<tr>
<td>G729</td>
<td>8</td>
<td>34</td>
<td>Toll</td>
<td>Toll</td>
<td>Network Telephony/Mobile Communications</td>
</tr>
<tr>
<td>FR-GSM</td>
<td>13</td>
<td>4.7</td>
<td>perceptible distortion</td>
<td></td>
<td>Mobile Communications</td>
</tr>
<tr>
<td>TIA IS-54</td>
<td>7.95</td>
<td>18.6</td>
<td>15</td>
<td>perceptible distortion</td>
<td>Mobile Communications &amp; Voice Storage</td>
</tr>
<tr>
<td>G.723</td>
<td>5.3/6.4</td>
<td>28.9/30.3</td>
<td>same as G.726</td>
<td></td>
<td>Video Conference &amp; multi-media applications</td>
</tr>
<tr>
<td>USFS CELP 1016</td>
<td>4.8,7,2,9,.6</td>
<td>17</td>
<td>14.5</td>
<td>perceptible distortion</td>
<td>mobile &amp; secure communications</td>
</tr>
<tr>
<td>MELP</td>
<td>2.4</td>
<td>20.5</td>
<td>same as USFS 1016</td>
<td></td>
<td>mobile communications</td>
</tr>
<tr>
<td>USFS LPC10e</td>
<td>2.4</td>
<td>8.7</td>
<td>8.7</td>
<td>synthetic</td>
<td>mobile &amp; secure communications</td>
</tr>
</tbody>
</table>

Recent research efforts have also concentrated on improving the speech quality for a given bit rate. This is evident by the development of the 2.4kbit/s Mixed Excitation Linear Predictive (MELP) coder [52], exhibiting communications quality speech, as compared to LPC10e’s synthetic speech quality.
2.2.2 Computational Complexity

Lowering the bit rate while maintaining quality is often achieved at the cost of increased complexity. This is illustrated in Table 2.1, where the 2.4kbit/s MELP coder is implemented using 20.5 MIPS of a floating point DSP processor, while the 4.8kbit/s FS1016 CELP coder utilises 14.5 MIPS. Note that MIPS is an abbreviation for million instructions per second, and that one DSP processor’s instruction is equivalent to a Multiply-Accumulate (MAC).

A complex algorithm requires powerful DSP hardware that may be expensive and consume large quantities of power. Until the late 1980's, many speech coding algorithms were not implemented in real time due to the lack of sufficiently powerful DSP hardware. Powerful DSP chips now exist, however recent speech compression algorithms push their capabilities to the limit. Future algorithms are expected to demand more powerful DSP devices, judging by the evolution of speech coding depicted in Table 2.1. A useful rule of thumb is that a decreased bit rate at a constant perceived quality requires exponential increases in complexity [40]. Note the MIPS associated with the compression algorithms in Table 2.1 have been estimated on TMS320C3x floating point DSP processor and the TMS320C5x fixed point processor, both supplied by Texas Instruments (TI). These are two of the most widely used DSP processors of late. Future speech compression algorithms may be implemented as ASIC designs with DSP cores effectively making them a commodity item, such as the recently released G.729/G.723 by Rockwell.

DSP hardware consumes significant amounts of power due to the high clock rates required (typically 30-100MHz). This is a concern for services requiring speech compression for hand held or portable terminals. The reduction in complexity of speech
coding algorithms would enable lower power consumption through the use of less powerful DSP hardware. Less powerful DSP hardware would also reduce cost. This may involve using fixed point DSP processors, inherently lower power consuming devices, instead of floating point versions. Hardware that runs on a lower supply voltages (3.3V) may be an alternative. The recent focus by telecommunication governing bodies on reducing Electromagnetic Interference (EMI) reinforces the issue. The pros and cons associated with each of these devices is advocated in Section 2.8.

2.2.3 Coding Delay

The coding delay of a speech transmission system consists of algorithmic, computational and transmission delay. As mentioned earlier many speech coders buffer speech for analysis purposes. This introduces delay into the system defined as algorithmic delay. For communications quality speech coders, a buffer (frame) length of 20-30ms is common. The time taken to process the buffer of speech is considered computational delay. This is usually less than or equal to the algorithmic delay for real-time systems. The end to end delay (total delay of the encoding/decoding process) of the speech coder is a multiple of the frame (buffer) length (transmission delays exempted). Total coding delays of 80-120ms are common. Delay may also be introduced by specific features of the speech coding algorithm. Low bit rate speech coding algorithms may examine speech information in future frames before coding information in the current frame. This is implemented in a causal system by introducing delay. Also some coding schemes such as the Pan-European mobile (GSM) system initiate transmission of encoding spectral parameters before all the model parameters are available in the encoder [41]. These delays associated with the speech processing system may be
compounded further by the transmission medium used. Satellite communications systems, in the geosynchronous orbit, exhibit a round trip delay greater than 200ms.

Delay becomes a problem for several reasons. Firstly, speech coders are often interfaced to the PSTN via four to two wire converters or "hybrids". A side effect of using these devices is that a proportion of the output signal from the decoder is fed back into the input of the encoder. Due to coding delays, this introduces echo where the user hears an echo of their own voice returned at multiples of 80-120ms. The introduction of the G.728, alleviated this echo problem to some extent, as the coding delay was less than 2.5ms. Proposed future toll quality lower bit rate (4kbit/s) coders will unfortunately have greater coding delays. The second problem is associated with the coding delay being coupled with long transmission delays such as those encountered with transmission via satellite systems. In this case a total delay of over 300ms may be encountered, making actual conversation difficult. Delays over 150ms can be perceived as an impairment for certain highly interactive conversations [44]. Thus the minimisation of coding delay is crucial to maintaining good speech quality.

2.3 Speech Production Model

Speech signals are non-stationary by nature; as their characteristics evolve slowly over time. Due to this slow evolution, the speech signals can be approximated as stationary over short periods (5-30ms).

A suitable model for speech production is the source-filter model [39] illustrated in Figure 2.1.
Speech production is modeled as a filtering operation, where a filter, formed by the vocal tract, is excited by a sound source. Note that characteristics of the source and filter are assumed to be time varying outside of these intervals (5-30ms) and thus must be updated accordingly.

Speech essentially consists of voiced and unvoiced sounds. Voiced sounds, such as vowels, are physiologically produced by air from the lungs being periodically interrupted by the vocal cords. Unvoiced sounds, such as consonants and fricatives, are produced by air passing through constrictions in the vocal tract. The excitation signal in the speech production model is depicted as either an impulse train for voiced speech or random noise for unvoiced speech.

The filtering operation performed by the vocal tract enhances some frequencies and attenuates others. Accurate modeling of the vocal tract is important in retaining the intelligibility of coded speech, while effective modeling of the excitation source produces natural sounding speech.
2.4 Characteristics of Voiced Signals

Figure 2.2 is a time and frequency domain plot of a segment of voiced male speech. The speech was bandpass filtered between 300Hz and 3300Hz and sampled at 8kHz. Note there are two recurring waveform cycles within the segment. The waveform in this illustration has a period of approximately $P = 90$ samples where $P$ is known as the pitch period.

![Time Domain](image_a.png)

![Frequency Domain](image_b.png)

Figure 2.2: Male Voiced Segment: (a) Time Domain, and (b) Frequency Domain
This periodicity is termed as the long term characteristic of a voiced speech signal. Speech coding algorithms use various techniques to model this characteristic in their analyses. These are considered in Section 2.6 and 2.9. The pitch period is related to the fundamental frequency $F_0$ (units Hz) by:

$$F_0 = \frac{F_s}{P}$$  \hspace{1cm} (2.1)

where $F_s$ is the sampling frequency in Hz.

The fundamental frequency $F_0$, in human speech, ranges between 50-500Hz. Male speakers range between 50-160Hz and 100-500Hz for females. As the speech signal depicted is periodic, its magnitude spectrum consists of harmonics of the fundamental frequency. The amplitude of the harmonic series is modulated by the filtering effect of the vocal tract. The peaks in the spectrum correspond to resonances in the vocal tract and are known as formants. The number, frequency location and bandwidth of these resonances are time varying, changing as sounds are articulated.

### 2.5 Linear Predictive Coding

The source-filter model was introduced previously as a suitable way to model speech production. A popular method of modeling the vocal tract filter is Linear Predictive Coding (LPC) [30]. This technique uses an all pole filter to model the vocal tract.

Consider the speech production (source-filter) model in the z-domain. The excitation source, $X(z)$, drives a vocal tract filter $H(z)$, to generate synthetic speech $\hat{S}(z)$ such that:

$$\hat{S}(z) = X(z)H(z)$$  \hspace{1cm} (2.2)

where $H(z)$ is defined as an all-pole filter of the form:
\[
H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{G}{A(z)} \tag{2.3}
\]

where \( \{a_k\} \) for \( k = 1, 2, \ldots, p \) is a set of \( p \) linear prediction coefficients (LPC coefficients), and \( G \) is a scalar gain factor. The LPC model parameters are derived from the time domain speech signal [30]. The LPC coefficients are updated regularly due to the previously mentioned non-stationary nature of speech.

Figure 2.3 shows an example of LPC modeling applied to the speech segment presented previously in Figure 2.2. The dashed line is the magnitude spectrum of the original speech segment. The solid line represents the magnitude spectrum of a 10\textsuperscript{th} order LPC model.

![Figure 2.3: LPC Modeling of Male Speech](image)

LPC is normally used to represent the vocal tract filtering and thus enough poles must be present in the model to represent the number of formants. There are usually four formants in distinctly voiced speech segments, each formant requiring 2 poles to be adequately represented. Thus model orders of 10-16 are common for speech coding (extra poles are used to represent the valleys in the spectrum).
2.6 Existing Speech Coding Techniques

As mentioned in the previous section, LPC provides an effective method in representing the short term characteristics of speech. This section explains several techniques in coding the excitation signal by expanding on the two most popular approaches utilised in speech coding implementation.

2.6.1 Code Excited Linear Prediction (CELP)

The CELP coder, introduced in [29], is capable of attaining communications quality speech at bit rates of 4-8 kbit/s. It forms the basis of the US Federal standard FS1016 for operating voice at a rate of 4800 bps [24]. The CELP encoder uses codebooks and an analysis-by-synthesis procedure in extracting its excitation model parameters. Since this encoding process requires a decoder, it may prove useful to initially describe the CELP decoder (Figure 2.4).

In Figure 2.4, a stochastic and adaptive codebook excites a LPC synthesis (all-pole) filter in order to reconstruct speech. The (binary) stochastic codebook consists of

![Figure 2.4: CELP Decoder with Adaptive Codebook](image-url)
vectors with randomly distributed values (of unity and zero magnitude). This codebook might be considered as sparse, as the values of unity magnitude are sparsely distributed throughout the codebook [55]. The stochastic codebook contribution is defined by the entry \((i)\) and the gain \((\alpha)\).

Long term periodicity existing in the excitation signal and evident in speech characterised by voiced segments can be modeled by the current excitation segment as a weighted version of a previous excitation segment. This may be interpreted as either a filtering operation or as a vector quantisation operation with an adaptive codebook [44 pg 95]. The filtering operation is defined by the pitch delay or lag \((L)\), and the pitch gain, \((\beta)\) given as

\[
\frac{1}{P(z)} = \frac{1}{1 - \beta z^{-L}}
\]  

(2.4)

The adaptive codebook consists of codebook entries \(N_{sf}\) samples long (number of speech samples in a subframe). Thus each possible value of \(L\), has a codebook entry of \(N_{sf}\) samples. It is an overlapping codebook where adjacent entries share common components (first and last samples) [56]. The adaptive codebook contents are time varying, and are updated every subframe from the composite (sum of adaptive and stochastic) excitation. Thus the adaptive codebook tends to build up a good approximation of an excitation waveform over time, for continuous voiced segments of speech. Both the stochastic and adaptive codebooks have overlapping structures to reduce the memory and complexity requirements.

Both the stochastic and adaptive codebooks are time varying. They are updated every subframe of \(N_{sf}\) samples, where \(N_{sf}\) is usually a submultiple of the LPC frame size, \(N\) (eg. \(N = 4N_{sf}\)).
The LPC synthesis filter models the short term periodicity (vocal tract filtering), the pitch synthesis filter models the long term periodicity (pitch structure), and the stochastic codebook models the random component (remaining modeling errors). The stochastic codebook predominantly models the excitation for unvoiced speech, as it is characterised by random components.

The CELP encoder searches the adaptive and stochastic codebooks to determine the optimum excitation vectors and gains. The synthesis filter response to each possible excitation vector is determined. This is then compared to the target (original) speech vector in a weighted mean-square error sense. The excitation vector and gain that minimises the weighted mean square error for the current subframe is transmitted to the decoder. Thus the codebooks are searched using an analysis by synthesis or closed loop procedure and a synthesis stage is required to analyse the speech signal.

Thus the source signal from the speech production model in the CELP coder is represented by two gain shaped codebooks. One containing codevectors with a purely random component while the other containing codevectors with both a random and periodic component. The level of each component is dependent on the recent characteristics of the speech signal being analysed.

It must be noted that the LPC coefficients are determined in an open loop manner. Speech is initially subjected to an analysis window such as a Hanning window of size $N$. An autocorrelation procedure is applied to the windowed speech, producing coefficients which are subsequently used in determining the LPC coefficients by a recursive algorithm such as the Levinson-Durbin [30].
2.6.2 Sinusoidal Coding (Frequency domain)

Voiced speech, in terms of the source-filter model described in Section 2.3, is viewed in the time domain, as a periodic pulse train convolved with the impulse response of the vocal tract filter. The spacing of this pulse train is the pitch period, $P$. In the frequency domain, it corresponds to a pulse train multiplied by the frequency response of the vocal tract filter (Duality Rule). The spacing of this pulse train is the fundamental frequency, $F_0$, of the speech. The voiced excitation model illustrated in Figure 2.1, may be represented by the sum of sinusoidal oscillators. Ideally when the excitation is periodic, it may be represented by a Fourier series decomposition in which each harmonic component corresponds to a single sine wave. Thus passing this sine wave representation of the excitation through the vocal tract results in the following synthesised speech waveform:

$$s(n) = \sum_{m=1}^{M} A_m \cos(\omega_m n + \theta_m)$$  \hspace{1cm} (2.5)

where the parameters $\{A_m\}$, $\{\omega_m\}$, $\{\theta_m\}$ represent the magnitudes, frequencies, and phases of the sinusoids. To determine the frequency of each sinusoid, simple peak-picking of the high resolution Discrete Fourier Transform (DFT) magnitude spectrum may be used. The magnitude and phase of each sinusoid is then obtained by sampling the high resolution DFT at these frequencies.

The model parameters are updated at regular intervals, as the speech characteristics are non-stationary. Parameter update intervals (frames lengths) of 10-30ms are common. Note that $M$ is time varying, as the number of peaks from frame to frame varies. A limit is placed on the number of possible peaks where selection is based on their magnitude.
To synthesise speech using the sinusoidal model, the decoder generates $M$ sinewaves of the estimated magnitude, frequency, and phase. However, caution must prevail to ensure continuity of the sinewaves at the frame boundaries. This is achieved by slight adjustment (interpolation) of the model parameters to ensure smooth evolution of the synthesised speech signal across frame boundaries.

The sinusoidal coder can be considered as a parametric coder, as it describes the speech signal using a set of model parameters, where no attempt is made to reproduce the original speech waveform exactly. There is no closed loop analysis where the input speech is the desired signal. Instead it relies on the model assumptions to produce good quality synthesised speech.

The sinusoidal model is capable of representing both voiced and unvoiced speech. Peak picking the short term DFT magnitude spectrum of unvoiced speech will produce model parameters that tend to be randomly distributed.

The authors of [28] report that for frame lengths of less than 10ms, the speech signal reconstructed by the sinusoidal model is perceptually indistinguishable from the original. It was noted earlier that the amplitude, frequency and phase for each sinewave is determined on a frame by frame basis. As speech evolves frame to frame, different sets of parameters will be obtained. To obtain high quality speech, the sinewaves from one frame must evolve smoothly to the next frame. Larger frame lengths make the process of smoothing the discontinuities associated with the sinewaves at the frame boundaries difficult. This is the case when the overlap-add interpolator is used to reconstruct the speech signal. It assumes the parameters are stationary over a window twice the frame length. A frame length of 15-20 ms would require parameters to be...
stationary for greater than 30ms. This is beyond the limits of the speech production model and distortion will occur.

One of the problems with the sinusoidal coder is the time varying number of parameters, which after quantisation leads to a varying bit rate. During voiced speech, the sinusoid frequencies \( \{\omega_m\} \) will be multiples of the fundamental. Thus \( \{\omega_m\} \) can be efficiently modeled as multiples of the fundamental frequency for the current frame. Therefore (2.5) becomes the *harmonic* sinusoidal model:

\[
s(n) = \sum_{m=1}^{M} A_m \cos(\omega_0 m n + \theta_m)
\]  

(2.6)

where \( \omega_m = m \omega_0 \) and \( M = \frac{\pi}{\omega_0} \). Only the fundamental frequency, \( \omega_0 = 2\pi F_0 \), is required for reconstruction of the harmonic frequencies at the synthesis stage.

Due to the non-stationary characteristics of speech signals and the varying nature of the fundamental frequency with respect to time, the value of \( M \) changes frame by frame thus leading to the quantisation of varying numbers of amplitudes \( A_m \). This makes quantisation of these parameters using a uniform bit rate impossible. The use of LPC modeling has alleviated this problem by eliminating the need to transmit these amplitudes. The LPC parameters are used to reconstruct the spectral envelope of the speech signal at the decoder where the amplitudes are determined at multiples of the fundamental frequency.

### 2.7 Comparison of Coding Schemes

CELP based coders are capable of coding communications quality speech at bit rates down to 5-4kbit/s [24], [42]. Typically, 75% of the bit rate is allocated to the excitation (codebook and gain parameters). This is attributed to the excitation and gain parameters
being updated at a faster rate than the spectral parameters. These parameters are updated every 7.5ms, compared to the vocal tract filtering parameters being updated every 30ms. Below 4kbit/s, the quality sharply degrades, as not enough bits are available to adequately represent the excitation. Also the update intervals have to be lengthened, thus violating the stationary assumption for speech using the speech production model. Most of the computational complexity is in the codebook searching algorithm, and is proportional to the codebook sizes and update rates of the excitation parameters.

The adaptive codebook contributes significantly to the quality of CELP due its ability to build up a good model of the excitation. As CELP uses a waveform matching criteria to code the input speech, it is capable of faithfully coding background noise. No annoying artifacts are introduced with non-speech inputs.

Parametric coders (sinusoidal and harmonic) can provide communications quality speech below 5kbit/s [43], [19]. Unlike CELP, they require model parameters such as pitch and voicing to be extracted from the speech signal. Parametric coders have a longer frame rate than CELP for the excitation information (typically 20ms compared with 5ms for CELP).

The large time varying number of parameters in sinusoidal coders can make them difficult to quantise. Fortunately there is usually a large amount of correlation present. For instance adjacent harmonics often have similar magnitudes. This can be exploited using techniques such as LPC to code and transmit the magnitude information.
2.8 Real Time Implementation

Any speech coding algorithm may be implemented using available DSP chip technology, but the cost of the implementation will increase rapidly with the use of faster DSP processors or the increase in the number of "slower" DSP processors used. It is also important to consider power consumed by the implementation, especially for hand-held mobile telephony applications. The availability of powerful DSP processors has led to the implementation of highly complex speech coding algorithms. Five years ago, the G.728 16kbit/s Low Delay CELP implementation required two DSP (AT&T WE-DSP32C-80) processors to run in real time, while today it can be easily implemented using one processor (TMS320C31-50MHz). The fixed point processors are faster, have lower power consumption and are generally cheaper compared to their floating point counterparts. Unfortunately they are cumbersome to program and have less precision. This is generally the case in trying to implement floating point oriented speech coding algorithms. There has been a growing trend, for speech coding standards, to move away from heavily oriented floating point algorithms, as witnessed by the new G.729 8kbit/s standard being published in fixed point pseudo code. The floating-point processor simplifies algorithm development because it requires no scaling, normalisation and overflow handling, effects usually associated with fixed-point processors.

The selection of a DSP processor for implementing a speech coding algorithm is not determined solely on the instruction cycle of the DSP, but also on the suitability of its instruction set in implementing the main blocks of the coding algorithms. These include processes such as FFT computations, convolutions and correlation type evaluations. Much discussion has been made on evaluating a DSP processor on how well it
implements these type of processes, thus advocating for a list of processes to be used as benchmarks as well as considering the raw MIPS associated with the DSP processor. Other features that may be taken into account when selecting a DSP processor include: the amount of on chip RAM/ROM (single access memory fetches), overall memory address space, on-chip peripherals such as serial ports, timers, host interface ports, addressing modes, on chip Direct Memory Access (DMA), boot loading capabilities, internal and external interrupts, programmable wait states, on-chip emulation ports for debugging, power down capability and overall power consumption.

Programming technique is crucial in optimising the algorithms run on a DSP processor (reducing the implementation complexity). Implementation complexity takes into account the limitations of the DSP processor in accommodating the algorithm. This is not to be confused with general algorithmic complexity, which basically determines the number of Multiply-Accumulates (MAC) required by the algorithm. The extra operations required when implementing a particular algorithm may include all or some of the following:

i) Initialisation overhead involved in setting up the algorithm in a subroutine. This is dependent on the type of DSP processor and the number of arguments being passed in the subroutine.

ii) Wait states due to the pipeline architecture of the chip and conflict wait states, which are automatically inserted by the chip for consecutive fetches from the same physical memory. The conflict wait states may be overcome with careful program and data memory management. This is accomplished by putting program and data buffers in different memory banks. The wait states due to the pipelining effect may be reduced by taking into consideration the latency effects
of the individual DSP instruction. This is accomplished by making sure the result of an instruction is made available before it is required by a subsequent instruction.

iii) The scaling of particular vectors to maintain resolution, normalisation and overflow handling in fixed point DSP processors.

iv) Divide and square root operations. These are usually contained in the DSP processor’s library as subroutines [49]. In cases where subroutines are embedded in a repeating sequence (loop) it may be optimal to avoid calling the subroutine as it contains initialisation overhead. Instead incorporate the particular function of the subroutine in the loop.

2.9 Issues concerning the selected topics of this research

Having introduced the reader to the various methodologies associated with speech coding it is now necessary to present the selected topics of this research in a form where one can identify some if not all of the following:

i) The selected topic’s relevance in the field of speech coding.

ii) The advances made in each topic to this date in terms of increased system performance.

iii) The improvements proposed as a result of the research undertaken by the author.

2.9.1 Pitch Estimation

The pitch parameter is crucial in creating the excitation in the speech production model, described in Section 2.3, that synthesises voiced speech. Pitch estimation is an essential
component in speech analysis-synthesis systems that utilise this speech production model. Accurate pitch estimation is particularly useful in low bit rate speech coding systems such as CELP and sinusoidal based coders. The pitch parameter may be represented in the time domain as the pitch period (lag) or in the frequency domain as the fundamental frequency, \( F_0 \).

An accurate pitch estimate is essential to the synthesis of high quality speech within the harmonic sinusoidal coder’s framework. This is depicted in the harmonic coder’s synthesis stage, defined in (2.6), where the amplitudes, \( A_m \), and their position, are dependent on the fundamental frequency.

In the CELP coder, pitch is characterised by searching an adaptive gain-shape codebook containing past excitation information. The best innovation candidate, from the adaptive codebook, that minimises the difference between the original and synthesised speech is selected and its index is considered the pitch lag or delay. Pitch being modeled in a closed loop format means high computational complexity, as the search is conducted for an entire codebook. An alternative approach, requiring less computations, would be to consider using a low complexity pitch determination algorithm (PDA) to obtain a pitch estimate in an open loop format and then search the codebook within the neighbourhood of the estimated pitch delay, instead of the entire codebook [21].

These applications reinforce the importance of accurate pitch detectors in terms of improving the speech quality and to some extent decreasing computational complexity.

Trade-offs are associated with determining a pitch parameter that satisfies the speech production model requirements and the instantaneous pitch period (fundamental frequency) of the speech segment in question. The speech production model is rigid in
its interpretation of voiced speech, modeling it as a pulse train convolved by the impulse response of the vocal tract filter. The distance between pulses in the excitation signal is defined by the pitch period parameter. The speech signal varies both in fundamental frequency and in the finer structure of the waveform within a period. In certain instances, where the formants are changing rapidly, the pitch period can be difficult to detect due to the changing structure of the glottal waveform. In other words, the glottal waveform is not a perfect train of periodic impulses (assumption made in the speech production model). Therefore finding the pitch period is not a straightforward procedure.

There is also some concern over PDAs distinguishing between unvoiced speech and low level voiced speech or transitions between voiced and unvoiced speech. In frame based analysis of speech signals, feature extraction is carried out on the current frame of data and a decision is given at the end of the frame. Frame based methods are incapable of tracking rapid changes in signal characteristics and transitions between unvoiced and voiced can affect the decision. In addition, 'quantisation errors' in the pitch estimates can be introduced since the sampling of the speech signal is performed independently to the instantaneous pitch period, which may lead to audible distortions.

Over the years a wide variety of PDAs have been developed including time domain detectors, such as the autocorrelation type detector [22], and frequency domain detectors such as the Cepstral method [26]. All, to some extent, have suffered from the above mentioned problems.

Most PDAs use the concept of waveform similarities in determining the pitch parameter, whether it's the pitch period or fundamental frequency. The concept compares the original signal with shifted versions and the shifted distance exhibiting the
greatest similarity is declared the pitch period. This is prevalent in the time domain
detectors, such as the Autocorrelation method and Average Magnitude Difference
Function (AMDF) [45]. In the frequency domain, spectrum similarity methods have
been used as opposed to waveform similarity methods, as in the case with the MBE
vocoder proposed in [46]. In this case, the pitch parameter is determined by comparing
the pitch dependent reconstructed spectrum with the original speech spectrum.

Both the time and frequency domain detectors are computationally expensive. The time
domain detectors have correlation type evaluations dependent on the number of speech
samples being analysed (usually twice the length of the maximum pitch period). The
frequency domain detectors operate on the speech spectrum and thus require DFT
transformations on the speech signal.

A super resolution PDA proposed by Medan has been developed based on a similarity
model for the voice excitation [27]. The model takes into account the similarity
between two successive non-overlapping pitch intervals with the only difference being a
modulation or gain constant with respect to time. The author claims that the algorithm
offers a robust, high resolution scheme that is capable of avoiding the problems
associated with common pitch based speech coding techniques. Unfortunately the
computational complexity of the PDA is too high to warrant incorporation with existing
speech coding algorithms. Chapter 3 focuses on introducing procedures that will reduce
the complexity of the algorithm yet not jeopardise its accuracy. These procedures based
on decimation and interpolation techniques are compared with the original PDA and the
efficient method proposed by Medan. In addition all procedures are compared with a
database that contains manually estimated pitch periods of a speech database consisting
of two male and two female utterances.
Chapter 3 illustrates that the PDA developed by Medan compares favourably with existing methods such as the Autocorrelation method. The method exhibits similar complexity characteristics to the normalised Autocorrelation method [45], where the energy of the shifted version of the signal is used as the normalising factor. The decimation procedure to be introduced in Chapter 3 allows for a reduction in the complexity of the PDA by a factor of $D^2$, where $D$ is the decimation factor. This procedure consists of removing redundancy in the speech signal (usually bandpassed filtered between 300-3600Hz and sampled at 8 ksamples/s) in relation to determining the pitch parameter (usually in the range [50Hz,500Hz]). This is at the expense of the pitch estimate's resolution. Interpolation procedures endeavour to bridge this compromise. The interpolation procedure using polyphase vectors, recommended in [27], is shown to perform poorly in comparison to the polynomial interpolation procedure proposed by this dissertation.

2.9.2 LPC Modeling

In any Linear Predictive based coder [29] [31] [32], an all-pole synthesis filter is utilised to model an accurate estimate of the short term spectral parameters of speech. The weights of the filter are known as LPC coefficients and are determined by subjecting speech to Linear Prediction. The philosophy behind Linear Prediction is that the present speech sample can be reconstructed by summing the previous $P$ weighted speech samples [30]. In effect there is much correlation between speech samples.

Direct quantisation of these LPC coefficients is inappropriate due to their large dynamic range (8-10 bits/coefficient). Also there is no direct way of ensuring filter stability, which is perceptually important to the speech quality. Thus for implementing low bit
rate, Linear Predictive based speech coders, representations of the spectral parameters with better encoding characteristics must be employed. These include forms such as the lattice form reflection coefficients, otherwise known as Log Area Ratios (LAR) [12], which are used in VSELP [31]. Alternatives to the reflection coefficients are the Line Spectral Pairs (LSP) [13]. These LSP coefficients lend themselves to robust and efficient quantisation of the spectral parameters [14].

The short term spectral information can be adequately represented by a 34 bit LSP scalar quantiser [24] as compared to the non-linear quantisation of the reflection coefficients, encoded using a 43 bit LAR quantiser. With the utilisation of powerful Vector Quantisation (VQ) methods, the short term spectral information represented by LSPs can now be coded using 24 bits [23], effectively lowering the bit rate of Linear Predictive based coders by up to 500 bit/s, assuming update intervals of 20 ms.

Unfortunately the transformation of LPC coefficients to LSP frequencies and its reconversion back to LPC coefficients in the synthesis stage of the coding process increases the complexity. This can be attributed to the many evaluations needing computations of trigonometric functions, usually inefficiently evaluated in DSP processors. A simple iterative method based on [16] is proposed in Chapter 4. It utilises Chebyshev polynomials to eliminate trigonometric evaluations. The trigonometric evaluations can also be eliminated in the reconversion process by mapping the LSP frequencies in the cosine frequency (real) domain during the conversion process, instead of the conventional frequency (z) domain.

Chapter 4 presents the procedure in calculating the LSP frequencies from a set of LPC coefficients, assuming they characterise a stable and even ordered filter, \( A(z) \). Essentially, the procedure is a root finding algorithm, where the roots correspond to the
LSP frequencies and the polynomials in question are the difference and sum between $A(z)$ and its conjugate functions. It investigates both the Real root \[14\] and Ratio filter method \[15\] in determining the LSP frequencies, illustrating that the latter requires twice as many trigonometric evaluations. The Real root method in combination with the Chebyshev polynomials represents the most accurate and efficient method in determining the LSP frequencies from a set of LPC coefficients.

Another method that may be used includes using the LSP frequency scalar quantisers (eg. from the FS1016 standard) as the potential roots of the LSP polynomials. Instead of using the Discrete Cosine Transform (DCT) at points determined by a fine grid, as in conventional LSP root finding methods \[14\], the points are determined by actual LSP frequency quantiser values. This eliminates the need to quantise the LSP frequencies (required when using the generic conventional LSP finding procedures). Effectively the resolution or accuracy of the LSP frequencies is governed by the accuracy of the LSP frequency quantisers. Thus using a fine grid in determining the roots of the polynomials might prove wasteful. This may be known as the “Quantised-search Kabal” \[51\] and has a complexity of 642 MAC operations. A new method of computing the LSP parameters from LPC coefficients was presented in \[51\]. The grid associated with the Kabal method of determining zero crossings (LSPs) is eliminated by a procedure which separates the interval $[-1, 1]$ into five sections, each containing one zero crossing. The positions of these zero crossings are refined by five successive bisections and a final interpolation, giving a total of 60 polynomial evaluations instead of 140 using Kabal’s method (see Section 5.7). Unfortunately the procedure is reported in \[51\] to have less accuracy than the conventional Kabal method and require 664 MAC operations plus overhead including divide and square root functions.
2.9.3 Echo Cancellation

Echo is the circumstance where a delayed and distorted version of an original signal is reflected back to its source. This phenomenon is evident in telephone networks as a result of impedance mismatching in the hybrid transformer and is known as electronic echo (created by an electronic circuit. It was mentioned previously that delay exhibited by vocoders connected to a PSTN, lead to the introduction of echo in the telephone network. Thus it is crucial that an echo cancellation mechanism be incorporated with the vocoders.

Hybrid transformers are connection points between two wire links (subscriber loop), where both directions of transmission are carried on a single wire pair, and four wire links (typical of carrier systems), where the two directions of transmission are physically separated. Figure 2.5 illustrates a typical telephone network with reference to the near-end hybrid. Near-end echo can be described as the signal leakage from the near-end hybrid resulting in the far-end talker hearing a delayed version of his or her own speech (talker echo).

In normal telephone networks, with the exception of long haul transmission systems such as satellite links, the near-end echo is perceptually insignificant due to the small round trip delay. In the case of interfacing low bit rate speech coders into the four-wire path of the network, the delay is increased (up to 120 ms) and the near-end echo leaked by the hybrid transformer becomes annoying.
The echo path is assumed not to vary during a connection, thereby allowing an adaptive filtering approach [2] [4] to be a suitable candidate in canceling echo resulting from leakage in a hybrid transformer. Various types of adaptive filter structures can be used to model the echo path exhibited by the hybrid. These include Finite Impulse Response (FIR) [2], Infinite Impulse Response (IIR) [6], Lattice [8] and Frequency Domain [7] filter structures. There are two basic algorithms for adapting the weights of the filter structure, namely the Least Mean Square (LMS) [4] and the Least Squares (LS) [9] or Fast Kalman (Recursive Least Squares) method.

There are various trade-offs in using a particular filter structure and adaptation algorithm configuration in terms of echo attenuation, speed of convergence and implementation complexity. These are briefly mentioned below, but the main objective in this research is implementing an efficient echo canceller satisfying ITU recommendation G.165 that integrates with a vocoder on a single DSP processor.

The echo canceller must accurately estimate the echo path and adapt quickly to any variations. Therefore it not only needs to be accurate in replicating the echo path, but it must converge quickly. The advantages of using a FIR filter as opposed to other filter structures in the canceller include [8];
a) It is a simple tapped delay line filter structure, as opposed to the more complex lattice filter structure.

b) It exhibits quicker convergence, as compared to the IIR filter that converges slower with possibilities of converging to a local minimum.

c) There is no need for stability testing to confine the poles within the unit circle, as is the case with the IIR filter structure.

d) No need for transform operations that are necessary for a frequency domain filter structure. The frequency domain filter structure becomes viable for filter tap sizes $N \geq 64$. It has been reported in [33] that for a tap size, $N = 64$, the complexity reduces to 70% of the time domain version of LMS. Unfortunately to obtain such reductions a Fast Fourier Transform (FFT) is used confining the tap sizes to powers of two. These transform operations have a higher complexity when implemented on a fixed-point processor (due to scaling) as compared to the number of MACs depicted by their algebraic formula.

e) Easy implementation of the Least Mean Squares (LMS) adaptation algorithm. The Fast Kalman method has a complexity in the order of $7N$ compared to $N$ using LMS adaptation. The LS approach takes $\frac{N^3}{6}$ MACs using the basic Gauss elimination and backward substitution [9].

The convergence speed of the LMS adaptation procedure is dependent on the input signal to the FIR filter structure. The normalised version of the LMS procedure (NLMS) eliminates this dependence and results in faster convergence [34]. The NLMS adaptation procedure requires $2N$ MAC operations (twice the complexity of the LMS
The research undertaken in this thesis and described in Chapter 5 compares the performance of the LMS and NLMS adaptation algorithms in terms of echo attenuation levels attained and convergence speeds.

The block update method may also be used as an adaptation procedure [54, Eq. 34 page 415]. This involves updating the coefficients less frequently with a thinning ratio, $M$. In this case the adaptation algorithm differs in its implementation compared to the standard LMS algorithm. The block update method during each sample period updates $\frac{N}{M}$ weights using the correlation of the past $M$ residual (error) and input samples corresponding to the weight index. The standard LMS procedure updates $N$ weights using the correlation of the past residual and input signal, where $M = 1$. The tap size of the filter structure is represented by $N$. This procedure has been reported to perform better than the standard LMS algorithm in terms of attenuation level and convergence speed attained. Concerns of instability raised in [54] recommend smaller step sizes which compromise its performance below that of the standard LMS adaptation procedure. The complexity of the block update method in [54] is equivalent to the standard LMS procedure.

A novel procedure, relating to the time the LMS adaptation process (or its normalised version) is enabled is also presented. This procedure decreases the complexity of the cancellation algorithm while compromising the convergence time of the adaptation algorithm. The reductions in complexity are in proportion to the increases in the convergence time.

An alternative method used to reduce the complexity involves avoiding the correlation computations in the LMS adaptation procedure defined in Chapter 5 by (5.5). This is
accomplished by using the signs of the input signal (far-end talker) and the residual signal to compute the updates. This “sign” algorithm is reported in [54] to degrade significantly, with a 50% decrease in the convergence rate and an increase in degradation of the residual echo due to interfering near-end speech, compared to the conventional LMS adaptation procedure.

2.10 Conclusion

This chapter has provided an overview on issues relating to current speech coding algorithms, namely speech quality, bit rate, coding delay and complexity. Essentially low bit rate speech coding algorithms use a speech production model, thereby classing them as parametric. Two of the more popular speech coding techniques have been described covering both time and frequency domain techniques. This has introduced the reader to a variety of techniques used in estimating parameters such as the long term characteristics of speech (pitch) and its short term characteristics (spectral envelope). Their accuracy is essential in synthesising high quality speech. Implementation complexity of an algorithm was impressed as an important issue that is sometimes overlooked in striving for high quality low bit rate yields. The chapter introduced to the reader selected techniques that estimate pitch and spectral parameters and cancel echo when vocoders are connected to the telephone network. These techniques were chosen due to their accuracy in estimating parameters or their quantisation characteristics and, in relation to the echo cancellation technique, its popularity.

The rest of the dissertation is concerned with introducing techniques that reduce the complexity of the selected techniques, namely LPC modeling using LSPs, the PDA developed by Medan and the NLMS echo canceller. The efficiency of these techniques is quantified in terms of determining the number of MAC operations required in
implementing the selected algorithms. Any compromise in the performance of these algorithms subjected by these techniques is investigated. It will be shown that these techniques have a varying degree of impact on the performance of the algorithms. The PDA developed by Medan is reduced by a factor of the decimation factor squared. It will be shown in Chapter 3 that this result is characterised by a slight degradation in performance. The reduction in complexity associated with the modified PDA facilitates possibilities for incorporating this technique in speech coding algorithms, as it still out performs current PDAs such as the autocorrelation technique. The technique in Chapter 4 used in reducing the complexity of the LPC to LSP transformation and vice-versa will be shown to have no impact on its performance.

The echo cancellation algorithm described in the dissertation embodies an adaptive 60 tap Finite Impulse Response (FIR) filter structure, where the weights of the filter are adapted using the Normalised Least Mean Square (NLMS) algorithm. Various trade-offs with respect to its performance are described with the end result being a reliable and effective near end echo canceller satisfying ITU Recommendation G.165 in terms of attenuation (dB) and convergence. The echo canceller is implemented on a single AT&T DSP32C floating point DSP processor and utilises less than 10 percent of the chip's computational power. This is accomplished by a novel procedure, relating to the time the adaptation process is enabled. This procedure decreases the complexity of the cancellation algorithm by 30% while having no impact on the convergence time of the adaptation algorithm. This is quantified in Chapter 5 depicting a critical point where the technique violates ITU recommendations G.165.
CHAPTER 3

EFFICIENT IMPLEMENTATION OF A SUPER RESOLUTION PITCH ESTIMATOR

3.1 Introduction

The chapter investigates the impact of decimation and interpolation techniques on the accuracy and complexity of the PDA developed by Medan [27].

It was mentioned in Section 2.9.1 that accurate pitch estimation is essential in producing high quality speech for Sinusoidal [19] [28] based coders. Low complexity PDAs also prove useful in lowering the complexity of the Long Term Predictor search in CELP [17] [29] coders. A possible candidate for accurate pitch estimation was proposed in Section 2.9.1, namely the PDA developed by Medan [27].

The PDA has been developed based on a similarity model for the voice excitation. The model takes into account the likeness between two successive non-overlapping pitch intervals with the only difference being a modulation or gain constant with respect to time. This is treated in more detail in Section 3.2.

Medan claims that the algorithm offers a robust, high resolution scheme that is capable of avoiding the problems associated with common pitch estimation techniques. This is investigated by comparing performances between the autocorrelation method [22] and the PDA developed by Medan. Essentially, the former is used as a benchmark in assessing the performance of the Medan PDA. The results in Section 3.7.2 illustrate that the PDA developed by Medan exhibits superior performance than the former.
Unfortunately the computational complexity of this PDA, determined in Section 3.4, is too high to warrant incorporation in existing speech coding algorithms.

This research focuses on introducing procedures that will reduce the complexity of the algorithm, yet not jeopardise its accuracy. Procedures based on decimation and interpolation techniques have been utilised in the PDA to reduce the complexity, compared to the original PDA proposed by Medan. The decimation procedure reduces the complexity of the algorithm by $D^2$, where $D$ is the decimation factor. This is accomplished by decimating the speech samples used by the PDA. It also compromises the accuracy of the pitch estimate, as the possible pitch candidates available are reduced (lower resolution). The interpolation technique attempts to compensate for the reduced pitch resolution. The decimation and interpolation procedures are described in Section 3.5. Their respective performances and method of evaluation are described in Section 3.7. All procedures are compared with a database consisting of manually estimated pitch periods of a speech database consisting of two male and two female utterances.

Medan in [27] offers an interpolation procedure to lower the complexity of his super resolution PDA. This is based on polyphase vectors and described in Section 3.5.2. This technique has a fundamental flaw in that it assumes the lower resolution pitch estimate (determined from the decimated speech samples) is within the temporal resolution of the correct estimate (original resolution). It will be shown in Section 3.5.2 that this assumption does not always hold. This is reinforced in Section 3.7.3 where this interpolation procedure produces a sub-standard performance similar to having no interpolation procedure present. The polynomial interpolation procedure proposed by this dissertation exhibits superior performance compared to the former technique and makes no such assumptions. The performance of the modified Medan PDA (utilising a
decimation factor, $D = 3$ and the polynomial interpolation technique) is also shown to
outperform the autocorrelation method. This combination results in an efficient PDA
that may be used in speech coding algorithms. The complexity of this PDA will be
shown to be one part in four compared to the widely used autocorrelation method.

3.2 Brief Background on the Medan Algorithm

A formal treatment in deriving the Medan PDA is presented here, based on [27]. It
includes both continuous and discrete time representations. The latter representation is
a prelude to the implementation of the PDA in speech coding systems described in
Section 3.3.

Medan defines two adjacent signals of duration, $\tau$, as

$$x_{\tau}(t, t_0) = s(t)w_{\tau}(t - t_0)$$  \hspace{1cm} (3.1)

$$y_{\tau}(t, t_0) = s(t + \tau)w_{\tau}(t - t_0)$$  \hspace{1cm} (3.2)

where $s(t)$ denotes the speech signal and $w_{\tau}(t)$ is a rectangular window of length $\tau$
seconds. The two signals are non-zero only inside the interval. $[t_0, t_0 + \tau]$.

Consider a speech frame starting at $t = t_0$ and consisting of exactly two pitch periods of
duration $\tau = T_0$ where $x_{T_0}(t, t_0)$ is the first period, and $y_{T_0}(t, t_0)$ is the subsequent period.

It is assumed that successive pitch periods are similar and are amplitude modulated
versions of each other.

$$x_{T_0}(t, t_0) = a(t_0)y_{T_0}(t, t_0) + e(t, t_0)$$  \hspace{1cm} (3.3)

where $a(t_0)$ is an unknown, positive amplitude gain at time $t_0$. The $e(t, t_0)$ reflects the
differences between the two periods.
To maximise similarity between the two segments \( x_\tau(t,t_0) \) & \( y_\tau(t,t_0) \), the time interval \( \tau = T_0 \) for which \( e(t,t_0) \) is minimised over the interval \([t_0, t_0 + \tau]\), is defined as the pitch period at the time instant \( t = t_0 \). Minimising the normalised squared error yields the following

\[
T_0 = \arg \min_{\tau, a(t_0) > 0} \left\{ J = \frac{\int_{t_0}^{t_0+\tau} [x_\tau(t,t_0) - a(t_0)y_\tau(t,t_0)]^2 \, dt}{\int_{t_0}^{t_0+\tau} [x_\tau(t,t_0)]^2 \, dt} \right\}
\]

(3.4)

where \( \tau \) is restricted to \([T_{0_{\text{min}}}, T_{0_{\text{max}}}]\).

The optimal modulation gain is obtained by taking the derivative of \( J \) with respect to \( a(t_0) \) and equating it to zero giving the following:

\[
a(t_0) = \frac{(x,y)_\tau}{|y|^2_\tau}
\]

(3.5)

where \((x,y)_\tau \) is the inner product and \(|y|^2_\tau \) represents the energy in the segment. By substituting the optimal value of \( a(t_0) \) into (3.4), it can be expressed as

\[
J = 1 - \rho_\tau^2(x,y)
\]

(3.6)

where \( \rho_\tau(x,y) \) is the cross correlation coefficient between the \( x \) and \( y \) segments and defined as

\[
\rho_\tau(x,y) = \frac{(x,y)_\tau}{|x|_\tau |y|_\tau}
\]

(3.7)

The pitch period, \( T_0 \) at time \( t_0 \) can be computed by finding the maximum \( \rho_\tau(x,y) \) where \( \tau \) is in the range \([T_{0_{\text{min}}}, T_{0_{\text{max}}}]\).
By sampling speech uniformly with a sampling interval $T$, a realisable solution of (3.7) can be obtained. The pitch period can be determined with a finite resolution, dictated by the sampling interval. This will be labeled as integer pitch. In the Medan algorithm, speech samples are prefiltered using a low pass filter in order to remove high frequency components, which are not necessary for tracking the pitch.

The $x$ and $y$ segments are now replaced by a sampled version of the segments, namely two $n$-dimensional vectors $x_n(i_0)$ and $y_n(i_0)$, given by

$$x_n(i_0) = s[1:n] \text{ and } y_n(i_0) = s[n+1:2n]$$

(3.8)

where $i_0$ indicates the sample index associated with $t_0$. The length of these vectors, $n$, is the hypothesised value of the integer pitch.

Based on (3.4), an optimal integer pitch period, $N_0$, at time $t = t_0$, minimises the following normalised discrete squared error function:

$$N_0 = \arg \min_{n,a_{(t_0)} > 0} \left\{ J = \frac{\sum_{j=1}^{n} (x_j - a(t_0)y_j)^2}{\sum_{j=1}^{n} [x_j]^2} \right\}$$

(3.9)

for the range $[N_{\min}, N_{\max}]$ of feasible integer pitch values corresponding to $[T_{0\min}, T_{0\max}]$ in (3.4). The optimisation of (3.9) leads to

$$N_0 = \arg \max_n \rho_n(x(i_0), y(i_0))$$

(3.10)

s.t. $N_{\min} \leq n \leq N_{\max}$

$$\rho_n(x(i_0), y(i_0)) = \frac{\sum_{j=1}^{n} x_jy_j}{\sqrt{\sum_{j=1}^{n} [x_j]^2} \sqrt{\sum_{j=1}^{n} [y_j]^2}}$$

(3.11)
The minimisation of (3.9) can be carried out by evaluating $\rho_n(x, y)$ for the full range $[N_{\min}, N_{\max}]$ and picking its maximum.

The integer pitch period, $N_0$, is estimated with a resolution dependent on the sampling rate. The exact pitch period is not normally an integer number (multiple of the sampling rate). It may be expressed as $N_0 + \beta$, where $\beta$ is the difference between the exact and integer pitch ranging between $0 \leq \beta < 1$. Thus by definition of (3.3) and the cross correlation definition, the attainment of the maximum cross correlation between the adjacent segments only occurs for pitch periods equal to an integer number. So to solve for non-integer pitch periods, the $y(i_0)$ segment is synchronised with the pitch period using linear interpolation. This may be used, as the bandwidth of the low-pass speech signal is much smaller than the sampling rate. Therefore $y_{N_0}(i_0 + \beta)$ can be approximated by a linear combination of two available segments, described in (3.12) to obtain the true maximum cross correlation.

\[
y_{N_0}(i_0 + \beta) = (1 - \beta)y_{N_0}(i_0) + \beta y_{N_0}(i_0 + 1)
\]  

(3.12)

Once the integer pitch estimate, $N_0$, has been determined by (3.10) and (3.11), substituting $y_{N_0}(i_0 + \beta)$ in (3.12) for $y_{N_0}(i_0)$ in (3.10) and (3.11) gives a value $\beta^*$ that properly aligns the two ($x_{N_0}(i_0)$ and $y_{N_0}(i_0)$) segments with respect to the pitch period, and determines the exact pitch estimate. This can be adequately expressed by (3.13) where $\beta^*$ is added to the integer pitch estimate to obtain the 'exact' pitch estimate.

\[
\beta^* = \arg\max_{\beta} \rho_{N_0}(x(i_0), y(i_0 + \beta))
\]

s.t. \[
y_{N_0}(i_0 + \beta) = (1 - \beta)y_{N_0}(i_0) + \beta y_{N_0}(i_0 + 1)
\]

(3.13)
Medan describes a low computational procedure for determining the 'exact' (super resolution) pitch estimate using empirical formulae based on the orthogonal projection theorem [27]. The evaluation of the integer pitch estimate is the most computational expensive procedure in the Medan PDA. Thus the focus for this chapter is in alleviating the computational complexity in determining the integer pitch estimate.

3.3 Implementation of the Medan PDA in Speech Coding Applications

The Medan PDA may be applied to parametric based speech coders to estimate pitch. Speech coders utilise speech production models, where model parameters are updated at 20-30ms intervals and will be known as analysis frames. This is attributed to the quasi-stationary nature of speech at these intervals. Note the pitch period may vary within these intervals for voiced speech. The estimation of pitch, voiced/unvoiced decisions, short term spectrum characteristics and gain factors, in these coders, are updated at these intervals. An interpretation of the Medan algorithm can be implemented to estimate the integer pitch, $N_0$, of each 20-30ms segment of speech.

As expressed earlier in (3.11), the algorithm calculates the cross correlation coefficient for adjacent segments $x(i_0)$ and $y(i_0)$, for hypothesised integer pitch values, $n$, ranging $[N_{\min}, N_{\max}]$. The $x(i_0)$ and $y(i_0)$ segments (successive intervals) are two $n$-dimensional vectors. The pitch analysis window size, $N_p$, must contain at least two pitch periods of a sampled speech signal to satisfy (3.3). Thus the total number of $x(i_0)$ and $y(i_0)$ vectors in the pitch analysis window varies between $[2, N_p/n]$ and is dependent on the hypothesised integer pitch estimate, $n$, currently being evaluated. Furthermore the number of cross correlation coefficients for each hypothesised integer pitch to be evaluated throughout the pitch analysis window is also dependent on $N_p/n$. An average
value for the cross correlation coefficient for the whole pitch analysis window must be calculated for each hypothesised integer pitch estimate, \( n \). By determining the maximum average cross-correlation coefficient for the range \([N_{\text{min}}, N_{\text{max}}]\), the integer pitch estimate can be found for a particular pitch analysis window. As the pitch analysis window size is twice the size of the coder's analysis frame (20-30ms), overlapping with the previous coder's analysis frame occurs.

The above procedure can be described by the following:

\[
N_o = \arg \max_n \frac{1}{N_k} \sum_{k=1}^{N_p} \rho_k(x_k(i_0), y_k(i_0))
\]

s.t. \( N_{\text{min}} \leq n \leq N_{\text{max}} \) \hspace{1cm} (3.14)

\[
\rho_k(x_k(i_0), y_k(i_0)) = \frac{\sum_{j=1}^{n} x_{kj} y_{kj}}{\left[\sum_{j=1}^{n} [x_{kj}]^2\right]^{1/2} \left[\sum_{j=1}^{n} [y_{kj}]^2\right]^{1/2}}
\]

where

\[
y_{kj} = x_{k+1, j}.
\]

\[
x_{kj} = k^{th} \text{ interval of the } x \text{ vector},
\]

\[
y_{kj} = j^{th} \text{ element of the } y \text{ vector in the } k^{th} \text{ interval},
\]

\[
N_p = \text{number of samples in pitch analysis window},
\]

\[
n = \text{hypothesised integer pitch period}.
\]
Thus the Medan PDA may be implemented in a vocoder in the following manner. Assuming a speech sampling rate of 8ksamples/s and the vocoder’s update interval of 20ms, the number of speech samples in the vocoder’s analysis frame, \( N_f = 160 \). The fundamental frequency of speech can vary between 50Hz to 500Hz, signifying pitch periods as long as 160 samples. Consequently, to satisfy (3.3), the pitch analysis window requires, \( N_p = 320 \), where 160 speech samples are from the vocoder’s current analysis frame while another 160 samples are overlapped from the previous one. The frame is divided into intervals of \( x(i_0) \) and \( y(i_0) \) segments containing \( n \) speech samples, where \( n \) is in the pitch range \([20,148]\). Note that the maximum pitch of 148 is selected as it allows for a 7 bit pitch quantiser, coinciding with the number of bits set aside by CELP and Sinusoidal vocoders. Depending on the value of the integer pitch period, \( n \), being analysed, the number of intervals varies from 16 when \( n = 20 \), to 2 when \( n = 148 \).

The energy of each adjacent interval \( x_{kj} \) and \( y_{kj} \) is calculated and the average cross correlation coefficient is evaluated by accumulating the cross correlations among each of the intervals throughout the pitch analysis window and dividing it by the number of intervals in the window (see 3.14).

The algorithm performs the above procedure for each hypothesised integer pitch period, \( n \), in the range \([20,148]\). The integer pitch period, \( n \), with the largest average cross correlation within this range is selected as the pitch estimate for the coder’s analysis frame.
3.4 Computational Complexity of the Medan Algorithm

The Medan algorithm described in equations (3.14) and (3.15), can be computational expensive. Basing all computations on Multiply-Accumulates (MAC) operations (the standard single DSP processor instruction), the following observations are noticed on this implementation of the Medan algorithm:

i) Three MAC operations are executed for each speech sample, in each $x(i_0)$ and $y(i_0)$ interval, of every pitch analysis window. This considers the inner product and energy calculations performed in (3.11) to evaluate the cross correlation coefficient. This is the most computational intensive task associated with the algorithm.

ii) Extra overhead such as initialisation, divide and square root functions needed to compute (3.12) and (3.13) add to the complexity. The complexity of these special functions is dependent on the DSP processor used and its software libraries.

Therefore the number of MAC operations necessary to perform the Medan PDA can be approximated by

$$3PN_p + P \times \text{overhead}$$

(3.16)

where

$P = \text{number of pitch periods searched},$

$N_p = \text{number of speech samples in the pitch analysis window},$

$\text{overhead} = \text{special functions (2 divides \& 2 square roots)}.$
Certain modifications to the algorithm can reduce the computational load. These include taking advantage of the fact that the $\|x\|$ (norm of the $x$ vector) in $(k+1)^{\text{th}}$ interval is equal to the $\|y\|$ (norm of the $y$ vector) in the previous interval,

$$\|y\|_k = \|x\|_{k+1}. \tag{3.17}$$

Subsequently by replacing $\|x\| \& \|y\|$ with an array of $\|x\|_k$ where $k$ ranges from $[1,N_r/n]$, all the energy calculations for the $y$ vectors are removed. This modification also removes one of the square root functions, as $\|y\|^{1/2}$ doesn't have to be computed. Therefore the number of MAC operations for the modified version reduces to approximately

$$2PN_p + P \times \text{overhead} \tag{3.18}$$

where

$$P = \text{number of pitch periods searched},$$

$$N_p = \text{number of speech samples in the pitch analysis window},$$

$$\text{overhead} = \text{special functions(2 divides & 1 square roots).}$$

This reduces the computational load by 30%, as compared to the original PDA. Yet based on a speech coder's update rate of 20ms, a pitch period range of [20,148] and a pitch analysis window of $P = 320$, the number of MAC operations is still considered computationally intensive (approximately 80,000 MAC operations).

The autocorrelation method is defined as
\[ R(\tau) = \sum_{n=0}^{p-1} s(n)s(n+\tau) \]  

(3.19)

where the variable, \( \tau \), is called the lag or delay and represents the pitch period range [20,148] while \( s(n) \) represents the speech signal. The pitch period is estimated by finding the lag that gives the maximum autocorrelation coefficient, \( R(\tau) \), within the pitch period range. Thus the number of MAC operations required to implement this algorithm is defined by \( PN_p \). Given the same dimensions for the pitch period range and pitch analysis window as the Medan PDA the autocorrelation method requires over 40,000 MAC operations.

3.5 Solutions to the Complexity Issue

To alleviate the above complexity, the number of speech samples needed to evaluate the cross-correlations will have to be reduced. If the speech signals are pre-filtered by a low pass filter, the speech samples can be decimated, thereby reducing the number of speech samples and cross-correlations evaluations in determining the integer pitch period, \( N_0 \). This decreases the resolution of the pitch estimate by the decimation factor, \( D \). Various forms of interpolation can be implemented to reconstruct an estimate of the cross-correlation values comparable to the "original" resolution. These include:

i) The interpolation technique suggested by Medan using polyphase vectors [27].

ii) Polynomial interpolation, which endeavours to model the non-linear cross-correlation characteristics.

Once the cross-correlation coefficients of "original" resolution have been reconstructed, the integer pitch estimate can be determined by finding the pitch period, \( n \), that maximises the cross-correlation coefficient \( \rho_s(x,y) \).
3.5.1 Decimation

Decimating the speech samples by a factor $D$ reduces the number of cross-correlation evaluations needed within $[N_{\text{min}}, N_{\text{max}}]$ by a factor of $D$. Also, the number of samples in $x(i_0)$ and $y(i_0)$ vectors are reduced by a factor $D$. Thus the complexity of the PDA, expressed in (3.18), can be reduced by a factor of $D^2$.

As mentioned earlier the speech samples are pre-filtered by a low-pass filter and in this case, the cutoff frequency is at 900Hz. By the Nyquist theorem, the sampling rate must be at least twice the bandwidth of the signal, $f_{ds} = 1800Hz$. This allows for a decimation factor, up to $D = 4$, without jeopardising loss of information in the speech signal previously sampled at 8ksamples/s. However, decimation of the speech samples reduces the pitch period resolution of the PDA. Previously, an integer pitch period had a resolution of $125\mu s$, whereas now it would have a resolution of $D \times 125\mu s$.

Assuming a decimation factor $D = 3$, the approximate number of instructions needed in (3.18) would now be around 9000 as compared to over 80000. This significantly
reduces complexity only if the interpolation techniques, implemented to reconstruct the full complement of cross-correlation coefficients within the pitch period range \([N_{\text{min}}, N_{\text{max}}]\), are relatively inexpensive in computational complexity. Subjecting the speech samples to this decimation factor is unwarranted in terms of the Nyquist theorem but it provides a greater resolution for the integer pitch period estimates compared to \(D = 4\). This ultimately, assists the interpolation techniques by providing more cross-correlation values to determine the pitch estimate, thereby allowing greater accuracy in determining a pitch estimate.

### 3.5.2 Interpolation via Polyphase vectors

The use of polyphase vectors is claimed by Medan to compensate for the reduced sampling rate caused by decimating the speech signal and may provide accurate pitch estimates. The procedure initially evaluates the integer pitch estimate, \(N_L\), using (3.10), with the \(x(i_0)\) and \(y(i_0)\) vectors containing a decimated set of speech samples. Unfortunately the integer pitch period is also evaluated on a decimated set of cross-correlation coefficients. This decreases the resolution and therefore the accuracy of the pitch estimate. Thus an integer pitch estimate with original resolution can be expressed as

\[
N_0 = N_L L + k^* \tag{3.20}
\]

where

\(L = \) the interpolation factor and equal to the decimation factor,

\(N_L = \) pitch estimate based on the decimated set of cross-correlation coefficients,

\(k^* = \) is an integer ranging from \([0, L - 1]\).
The formula (3.13) used in evaluating the fractional portion of the pitch estimate in Medan's PDA can be adapted to estimating $k^*$, the difference between integer pitch estimate comparable to the original resolution, $N_0$, and the pitch estimate based on the decimated cross-correlation coefficients, $N_L$. It can be empirically expressed as

$$k^* = \arg \max_k \rho_{N_L} (x(i_0), y(i_0 + k/L)) \quad k = 0, \ldots, L - 1 \quad (3.21)$$

Though the vectors $x(i_0)$ and $y(i_0)$ are decimated, the vectors $y(i_0 + k/L)$ are available by using different sets of the original speech samples and are called polyphase vectors. Thus no interpolation of the $y(i_0 + k/L)$ is necessary.

The value of $k^*$ which maximises the cross-correlation values in (3.21) is substituted in (3.20) to obtain an approximation of the pitch estimate with a resolution comparable to the original sampling rate. In other words, the decimated cross-correlation coefficients are reconstructed to their former resolution within the neighbourhood of $N_L$, using the polyphase vectors. Subsequently the interpolated position of the pitch period estimate, $N_0$, which maximises cross-correlation coefficient is determined.

This interpolation technique requires only $L$ cross-correlation evaluations in its calculation of (3.21). The procedure also takes advantage of the fact that the components of the polyphase vectors are attained at the original sampling rate. Thus the number of MAC operations necessary to implement this technique can be expressed as

$$2LN_p + L \times \text{overhead} \quad (3.22)$$

where the pitch period range, $P$, in (3.16) has been substituted by the interpolation factor, $L$. 

57
This particular method assumes that $N_0$ is within the temporal resolution of $N_L L$. Also the interpolation procedure only looks at pitch estimates ahead of the initial pitch estimate. Due to the resampling (decimation) of the cross-correlations, the maximum cross-correlation may be found at a lower pitch period than the initial estimate. Thus due to the decimation process and the numerous local maxima characteristics of the cross-correlation coefficients within $[N_{\min}, N_{\max}]$, the true maximum cross-correlation coefficient that would be evaluated under original sampling conditions is sometimes unnoticed. This can be seen in the following scenario. Figure 3.2 illustrates speech samples (sampled at 8ksamples/s) in an arbitrary pitch analysis window. The observed pitch period is estimated at 32 samples.

The Medan PDA with original resolution is applied to the pitch analysis window. The cross-correlation coefficients obtained by the PDA indicate that the integer pitch period is 63 samples. In effect the PDA has succumbed to pitch doubling. The solution to this problem is described in detail in Section 3.6. By applying a lower resolution to the Medan PDA, by the use of decimation, where $D = 3$, the integer pitch period is evaluated as 62 samples. In this case, the polyphase interpolation procedure described in (3.21) and substituted in (3.20) will exhibit results comparable to the PDA with original resolution, as the initial maximum chosen, $N_L L$, was within the temporal resolution of the maximum chosen by the PDA with original resolution.

Figure 3.3 illustrates the cross-correlation calculations for the above speech samples evaluated under original sampling conditions (dashed line) and under the lower resolution, $D = 3$ (solid line). The cross-correlation values under original sampling condition have comparable local maxima to the absolute. This may lead to selecting a local maximum instead of the absolute when using a lower resolution.
Figure 3.2: Illustration of speech samples in an arbitrary pitch analysis window.

By applying an even lower resolution to the Medan PDA, where $D = 4$, the integer pitch period is evaluated as 32 samples. Coincidentally the integer pitch period evaluated here is equivalent to the observed pitch period estimate in Figure 3.2, but it also depicts how decimation can affect the estimation of the integer pitch period, especially in instances where the PDA has calculated cross-correlations with many close local maxima.

Figure 3.3: Illustration of the original $\rho(x,y)$ and the decimated version (solid line).
Figure 3.4 illustrates the cross-correlation calculations for the above speech samples evaluated under original sampling conditions (dashed line) and under the lower resolution, $D = 4$ (solid line). In this case, the polyphase interpolation procedure described in (3.21) and substituted in (3.20) will not exhibit results equivalent to the PDA with original resolution as the integer pitch period selected is not within the temporal resolution of the pitch period estimated using original resolution.

Thus the interpolation technique proposed by Medan is sensitive to the decimation factor being applied to the speech samples. The decimated cross-correlations (ie. the cross-correlations sampled at a lower resolution) must have their maxima within the temporal resolution, $L$ of the absolute maxima of the cross-correlations evaluated under original sampling conditions. This, as depicted above, is not always the case and leads to degradation in the performance of the PDA. This is reinforced in Section 3.7.3, where performance evaluations are conducted on the above technique.

Figure 3.4: Illustration of the original $\rho(x,y)$ and the decimated version (solid line).
3.5.3 Polynomial Interpolation

Another type of interpolation can be in the form of an $M^{th}$ order polynomial described below as

$$y_n = \sum_{i=0}^{M} a_i n^{M-i}$$

(3.23)

where the interpolated value, $y_n$, is evaluated by summing $M+1$ weighted points raised to a specific power, defined as $n^{M-i}$.

A system of $M+1$ equations is necessary to determine the $M+1$ unknown weights, $\{a_i\}$ in (3.23). $M+1$ known points adjacent to each other and positioned by $n$ are substituted into (3.23) to form the system of equations. This can be expressed in matrix form as

$$y^T = A a^T$$

(3.24)

where $A$ represents an $x 2 \times i - 1$ matrix, formed by the $M+1$ known points and defined as

$$A = \begin{bmatrix}
(n - ML/2)^M & (n - ML/2)^{M-1} & \ldots & (n - ML/2) & 1 \\
. & . & . & . & . \\
. & . & . & . & . \\
(n - L)^M & (n - L)^{M-1} & \ldots & (n - L) & 1 \\
n^M & n^{M-1} & \ldots & n & 1 \\
(n + L)^M & (n + L)^{M-1} & \ldots & (n + L) & 1 \\
. & . & . & . & . \\
. & . & . & . & . \\
(n + ML/2)^M & (n + ML/2)^{M-1} & \ldots & (n + ML/2) & 1
\end{bmatrix}$$

(3.25)

where $L$ is the interpolation factor, $a^T = \{a_0, a_1, a_3, \ldots, a_{M-1}, a_M\}^T$ and $y^T = \{y(n - \frac{M}{2} L), \ldots, y(n - L), y(n), y(n + L), \ldots, y(n + \frac{M}{2} L)\}^T$. Note that the matrix $A$
is dependent only on the interpolation factor, \( L \), which is equal to the decimation factor, \( D \), and the polynomial’s order, \( M \).

The origin of the interval containing the known points is depicted as \( n = 0 \) and corresponds to the \( \frac{M}{2} + 1 \) point.

The above linear system of equations can be solved for \( a^T \), if the matrix \( A \) is invertible.

\[
a^T = A^{-1} y^T
\]  

(3.26)

Once the weights \( \{a_i\} \) are solved, the interpolated points within the range \( \pm \frac{L-1}{2} \) of the origin point can be determined by (3.21).

The polynomial interpolator reconstructs the full complement of cross-correlation coefficients by

a) initially taking intervals of \( M+1 \) decimated cross-correlation coefficients to being the known points.

b) determining the weights, \( \{a_i\} \), for the particular interval of concern.

c) evaluating (3.23) for \( \frac{-L+1}{2} \leq n \leq \frac{L-1}{2} \) to determine \( L-1 \) interpolated cross-correlation coefficients, \( y_n \), around the origin point of the concerned interval.

d) shifting the origin point of the interval by one place and repeating steps a) to d) until all the decimated cross-correlations within the interval \([N_{\text{min}}, N_{\text{max}}] \) have been interpolated.

e) choosing the pitch estimate that maximises the cross-correlation function.
The accuracy in restoring the reduced resolution of the integer pitch estimates using the polynomial interpolator is dependent on its order $M$. The cross-correlation coefficients have a non-linear characteristic as illustrated in Figure 3.3. Thus polynomials of $2 \leq M \leq 6$, where $M$ is even, will suffice.

Note that the computational complexity of the interpolator is proportional to the order $M$ of the polynomial. The number of MAC operations, necessary to implement the interpolator can be expressed as

$$\frac{N_{\text{max}} - N_{\text{min}}}{L} \left[(M + 1)^2 - 2(M + 1)\right].$$

This interpolation technique is not as sensitive to the decimation factor as was the case with the polyphase vectors.

Figure 3.5: Comparison of the original $\rho(x,y)$ and an interpolated version using 2nd order polynomial (where $L = 3$).

The following scenario illustrates this. The speech samples (see Figure 3.2) and decimation factors used are the same as with the polyphase vector technique. Note that the Medan PDA using original resolution exhibited an integer pitch period of 63 samples. Using a 2nd order polynomial on the decimated cross-correlation coefficients
(both for $D = 3$ and $D = 4$) allows accurate reconstruction of the cross-correlation coefficients of original resolution. This is illustrated by Figure 3.5 and Figure 3.6.

The solid line is the reconstructed set of cross-correlation coefficients and the dashed line the original cross correlation coefficients. In this scenario, the interpolation technique exhibited the same result for both decimation factors and equivalent to the Medan PDA using original resolution.

Figure 3.6: Comparison of the original $\rho(x,y)$ and an interpolated version using 2nd order polynomial (where $L = 4$).

A more detailed evaluation of the performance of this technique can be found in Section 3.7.4.

3.6 Introduction of a Pitch tracker

One of the problems associated with PDAs, based on correlation or similarity models, is that for a periodic signal, the cross-correlation function may exhibit several identical maxima at multiples of the pitch period within the range $[N_{\text{min}}, N_{\text{max}}]$ as witnessed in Figure 3.3.
In practice, due to the length of the pitch analysis window, speech signals demonstrate non-stationary attributes such as transitions between voiced and unvoiced segments, making it difficult to ascertain a valid pitch estimate for that specific interval, as depicted in Figure 3.7.

![Figure 3.7: Illustration of a transition between unvoiced to voiced region with a pitch analysis window](image)

In this case, the analysis window is in a transition between unvoiced to voiced speech. The PDA would find it difficult to accurately estimate the pitch of the voiced region given that only a portion of the voiced speech signal lies in the analysis window.

In some circumstances, certain multiples of the pitch period may be more prominent and may produce a multiple (doubling) pitch period estimate. This is illustrated in Figure 3.8 where the true pitch period is 41 samples in duration but the PDA estimates it as 82 samples long.

Also speech signals may have a decaying periodicity within the true pitch period or a dominant first formant, shaped by the vocal tract, which cause PDAs to choose a reduced pitch period estimate.
Figure 3.8: Illustration of a speech signal that has prominent multiples of the pitch period.

In Figure 3.9, the pitch period is 151 samples long, but the PDA estimates the pitch period to be 20 samples.

Figure 3.9: Illustration of a decaying periodicity within the true pitch period.

These underlying characteristics of speech signals make it difficult for PDAs based on the similarity (correlation) model to only select the pitch period estimate on the basis of locating the maximum cross-correlation function and still maintain high degrees of accuracy.
Pitch tracking has been proposed in [27] to combat the characteristics of pitch doubling and halving evaluated using Medan's PDA. This includes taking a selection of pitch estimates that attain a certain cross-correlation threshold and choosing the largest candidate pitch estimate. This eliminates the probability of choosing pitch estimates that are a submultiple of the actual pitch. This can occur from subperiodicities within the pitch period.

Pitch tracking procedures may also involve delayed decisions on the pitch period estimate based on the previous analysis window's pitch estimate and in some cases previous and future analysis window's pitch estimate. The former takes advantage of the slow varying pitch characteristic of voiced speech (±10% of the pitch value and never exceeds 25%) [27].

One such method that can be implemented on Medan's PDA involves delayed decisions based on the previous pitch period estimate's cross correlation function, $\rho_s(x,y)$. It providing bias towards the previous pitch estimate, as compared to the current, pending a previous frame's cross-correlation value greater than a specified threshold. Initially the procedure evaluates the current frame's pitch estimate, $n_c$, determined by finding the maximum cross-correlation coefficient for the interval $[N_{\text{min}}, N_{\text{max}}]$. Then it searches for the maximum cross-correlation coefficient evaluated on the current pitch analysis window within a ±20% interval around the previous window's pitch estimate. The pitch period corresponding to this maximum correlation will be denoted by $n_b$. If the pitch period estimate of the previous window, $n_p$, has a cross-correlation coefficient, $\rho_{n_p}(x,y) > 0.85$ then the pitch tracker exhibits bias towards the pitch estimate $n_b$. The threshold has been obtained from experimental results signifying a strong voiced region.
The pitch estimate for the current window is determined by selecting the largest cross-correlation coefficient from $\rho_{n_h}(x,y)$ and $\rho_{n_c}(x,y)$.

Figure 3.10 illustrates the performance of the Medan PDA with a pitch tracker. The solid line depicts the standard pitch contour, representing the observed pitch period estimates for an Australian female speaker. The dashed line represents the pitch estimate from the Medan PDA. The PDA matches the standard pitch contour for 89% of the utterance.

Figure 3.10: Performance of the Medan PDA with pitch tracker compared to standard pitch contour.

Figure 3.11 illustrates the performance of the Medan PDA without a pitch tracker. The solid line depicts the same standard pitch contour.
Figure 3.11: Performance of the Medan PDA without a pitch tracker.

The dashed line represents the pitch estimate from the Medan PDA. The PDA matches the standard pitch contour for 78.8% of the utterance. There is evidence of pitch doubling in Figure 3.11 that is eliminated by the pitch tracker in Figure 3.10. The evaluation of the various PDA will be undertaken without the use of pitch trackers, as this chapter is concerned with reducing the complexity of the Medan PDA and not evaluating the performance of particular pitch tracking algorithms. While this decision degrades the performance of the PDA it also gives a true indication of the PDA’s performance without any post processing. Post processing can be incorporated in a later stage of the PDA development to obtain better performance.

3.7 Performance Evaluation of the Interpolation Techniques

To evaluate the performance of the interpolation techniques a comparative investigation is undertaken among the original version of the Medan PDA, the Modified Medan PDAs which incorporate the decimation and interpolation techniques and a set of pitch estimates that have been manually estimated by observing the speech signal. The observed pitch estimates, which will be referred as the standard pitch contour, have been collated from the speech database that is applied to each PDA. The speech
database consists of eight utterances spoken by two Australian males and two Australian females.

A histogram depicting the range of the pitch lags in the contour associated with these eight utterances is illustrated in Figure 3.12. Most of the pitch lags are represented adequately except for the very high pitch lags associated with very low pitched male utterances. These limitations exist due to the lack of a substantial pitch contour database at the time of writing.

![Histogram of the standard pitch contour database.](image)

Figure 3.12: Histogram of the standard pitch contour database.

In evaluating the performance of the modified Medan PDA using the interpolation techniques, the following procedure is adhered:

a) Initially some performance benchmark must be set with respect to the original Medan PDA. This is achieved by comparing it with a PDA such as the autocorrelation method. The performance evaluation is made by comparing the integer pitch estimates from the respective PDAs with the standard pitch contour.
b) Performance evaluations of the Modified Medan PDAs are accomplished by comparing their pitch estimates with the pitch estimates from the original version of the Medan algorithm and the standard pitch contour.

c) The results of the performance evaluations on the PDAs are collated into a summary table.

3.7.1 Defining the performance measurements

The performance indicators consist of a set of measurements made that quantify the difference between the standard pitch contour and the pitch estimates from the various PDAs. Included among these are:

a) The number of gross errors in the pitch period (compared to the standard pitch contour) occurring throughout the whole speech database.

b) The number of fine errors in the pitch period, occurring in the speech database, compared to the standard pitch contour.

c) Hit ratio, defined as the occurrence of no gross or fine errors and is calculated as a percentage. This indicates the accuracy of the PDA. Both the fine and gross errors are calculated as a percentage of the overall pitch estimates in the database.

The difference between the standard pitch contour and pitch estimates from a PDA can be defined as being

\[ e(i) = |P_s(i) - P_e(i)| \] (3.28)

where
\[ P_s(i) = \text{the } i^{th} \text{ estimate from the standard pitch contour}, \]

\[ P_p(i) = \text{the } i^{th} \text{ pitch estimate from the PDA}. \]

A gross error in a pitch period is defined as a difference, \( e(i) \geq 8 \), between the standard pitch contour and the estimated pitch from a PDA [38]. This is equivalent to 1ms for speech sampled at 8kHz. For such cases, the pitch detector is deemed to have failed in estimating the pitch period. Possible causes for this are pitch doubling or halving and an inability to suppress the formant structure in speech, which can give misleading results.

A fine error in a pitch period is defined as a difference ranging from \( 3 \leq e(i) < 8 \). The value of 3 is chosen as the lower bound due to the fact that the modified Medan PDAs will have a decimation factor \( D = 3 \) applied to them as well as the uncertainty in the accuracy of the standard pitch contour. The decimation factor lowers the resolution of the pitch estimator, so the confidence in attaining the correct pitch estimate is compromised.

All the Modified Medan PDAs will be subjected to decimated speech signals having been low-pass filtered to 900Hz. The original Medan algorithm and the autocorrelation algorithm will be subjected to speech at the original sampling rate. The hypothesised integer pitch estimates for all PDAs will range between [20,146]. The pitch analysis window, \( N_p = 320 \) is the same for all the PDAs.
3.7.2 Comparison between the original Medan PDA and the Autocorrelation Method.

The Autocorrelation method consists of evaluating the autocorrelation function over a range of lags that span the pitch period range \([N_{\text{min}}, N_{\text{max}}]\) and choosing the lag, which attains the maximum autocorrelation value. The speech signal is spectrally flattened to de-emphasise the formant structure of voiced speech. This can be attained by subjecting the speech to a LPC analysis filter. It is an all-zero filter that uses the Linear Prediction model. Linear prediction strives to predict the current speech sample from a weighted version of the past \(P\) samples [30].

The LPC analysis filter can be described in the z-transform domain as

\[
H(z) = \sum_{i=0}^{P} a_i z^{-i}
\]  

(3.29)

where

\[a_i = \text{the } i^{th} \text{ LPC coefficient,}\]

\[z^{-i} = \text{the backward } z \text{ operator.}\]

The LPC coefficients are determined by the Levinson Durbin algorithm [30], which strives to minimise the mean square error between the original speech sample and the predicted version.

The pitch estimates from both the Autocorrelation method and the original Medan algorithm are compared to the standard pitch contour for gross and fine errors. Also a hit ratio is given, indicating the accuracy of the PDA. Hit results shown in Table 3.1 indicate that the original Medan algorithm performs better overall. It exhibits
significantly less fine errors, as well as a superior hit ratio. The Autocorrelation method has a tendency to perform adequately for female utterances, but significantly degrades for male utterances. This can be attributed to the formant structure of male speech interfering with the determination of the fundamental frequency. The Medan PDA is not dependent on speaker characteristics.

Table 3.1 Performance evaluation of the original Medan PDA and the Autocorrelation based PDA against the standard pitch contour.

<table>
<thead>
<tr>
<th></th>
<th>Original Medan</th>
<th>Autocorrelation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit Ratio</td>
<td>75.0</td>
<td>61.0</td>
</tr>
<tr>
<td>Fine Errors</td>
<td>1.5</td>
<td>14.6</td>
</tr>
<tr>
<td>Gross Errors</td>
<td>23.5</td>
<td>23.9</td>
</tr>
</tbody>
</table>

3.7.2 Performance Evaluation of the Modified Medan PDA using Polyphase Vectors

This Modified PDA is subjected to speech signals that have been decimated by a factor $D = 3$. The speech signals have been low-pass filtered to 900Hz. The polyphase vectors endeavour to increase the resolution of the pitch estimate comparable to the original sampling rate. Pitch estimates from the modified PDA are compared to the standard pitch contour and estimates from the original Medan algorithm, for gross errors, fine errors and a hit ratio. Pitch estimates are also evaluated for the modified PDA with the interpolation procedure being inhibited.

Hit ratio results in Table 3.2 show that the modified PDA with the polyphase vectors, performs comparably to the modified PDA with no interpolation.
Table 3.2: Performance evaluation of the Modified Medan PDA (using polyphase vectors) and the Modified Medan PDA (with no interpolation techniques) against the standard pitch contour.

<table>
<thead>
<tr>
<th></th>
<th>Polyphase</th>
<th>No Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit Ratio</td>
<td>67.1</td>
<td>66.8</td>
</tr>
<tr>
<td>Fine Errors</td>
<td>1.5</td>
<td>1.9</td>
</tr>
<tr>
<td>Gross Errors</td>
<td>31.4</td>
<td>31.2</td>
</tr>
</tbody>
</table>

This indicates that the polyphase vector interpolation scheme has very little effect in estimating the pitch period with increased resolution. The gross errors are similar indicating that there is limited improvement. This can be attributed to the fact that the initial integer pitch estimate found from the decimated samples is assumed to be within the temporal resolution of the correct estimate, which may not always be the case. The polyphase vectors can only be effective if the above assumption holds. This was expanded in Section 3.5.2.

Table 3.3: Comparison of the Modified PDA (using polyphase vectors) and the original Medan PDA.

<table>
<thead>
<tr>
<th>Deviation from Original Medan PDA</th>
<th>Polyphase</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>55.9</td>
</tr>
<tr>
<td>1 sample</td>
<td>25.9</td>
</tr>
<tr>
<td>Gross</td>
<td>13.9</td>
</tr>
</tbody>
</table>

Table 3.3 indicates that the polyphase vectors exhibit a deviation of not more than 1 sample, compared to the original Medan PDA, in over 80% of speech database. This table also indicates that gross errors (compared to the original Medan algorithm) occur in only 13.9% of the database.
3.7.4 Performance Evaluation of the Modified Medan PDA using Polynomial Interpolation

The Modified Medan PDA with the polynomial interpolator is also subjected to speech samples that have been decimated by a factor of 3. Three different ordered polynomials, where \( M = 2, 4, 6 \), are implemented as potential interpolators. The pitch estimates from these are compared to the standard pitch contour and the pitch estimates from the original Medan PDA. The hit ratio results, in Table 3.4, show that all of the different ordered polynomial interpolation schemes compare favourably with the original Medan PDA in terms of matching the pitch estimates against the standard pitch contour.

This is reinforced by results tabulated in Table 3.5, showing there is no deviation between the pitch estimate of the original Medan PDA and the modified version for over 75% of the speech database. Also illustrated is that the modified PDA's pitch estimates deviate significantly from the original Medan's pitch estimates on only 8% of the speech database. There are approximately half as many gross deviations compared to the polyphase vector technique illustrated in Table 3.3.

Table 3.4 Performance evaluation of the Modified Medan PDA (using polynomial interpolation) against the standard pitch contour.

<table>
<thead>
<tr>
<th>Polynomial</th>
<th>( M = 2 )</th>
<th>( M = 4 )</th>
<th>( M = 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit Ratio</td>
<td>74.2</td>
<td>73.2</td>
<td>73.2</td>
</tr>
<tr>
<td>Fine Errors</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Gross Errors</td>
<td>24.8</td>
<td>25.3</td>
<td>25.3</td>
</tr>
</tbody>
</table>

In terms of approximating the original Medan PDA, the polynomial interpolation scheme of \( M = 4 \) is the proposed candidate. The effectiveness of the polynomial
technique is illustrated in Table 3.5. The pitch period estimates between the original Medan PDA and the Modified Medan using the above mentioned interpolation technique digress occasionally.

Table 3.5  Comparison of the Modified PDA (using polynomial interpolation) and the original Medan PDA.

<table>
<thead>
<tr>
<th>Deviation from Original Medan PDA</th>
<th>$M = 2$</th>
<th>$M = 4$</th>
<th>$M = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>74.4</td>
<td>77</td>
<td>75.8</td>
</tr>
<tr>
<td>1 sample</td>
<td>16.9</td>
<td>15.0</td>
<td>15.6</td>
</tr>
<tr>
<td>Gross</td>
<td>8.1</td>
<td>7.5</td>
<td>8.0</td>
</tr>
</tbody>
</table>

A summary of the performances exhibited by all the PDAs is tabulated in Table 3.6.

Table 3.6: Summary of the performance results for all the PDAs against the standard pitch contour.

<table>
<thead>
<tr>
<th></th>
<th>Autocorrelation Method</th>
<th>Medan Original</th>
<th>Medan Polyphase</th>
<th>Medan Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit Ratio</td>
<td>61.0</td>
<td>75.0</td>
<td>67.1</td>
<td>73.2</td>
</tr>
<tr>
<td>Fine Errors</td>
<td>14.6</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Gross Errors</td>
<td>23.9</td>
<td>23.5</td>
<td>31.4</td>
<td>25.3</td>
</tr>
</tbody>
</table>

Table 3.6 indicates that the modified Medan PDA algorithm using the polynomial interpolation technique performs similarly the original Medan PDA algorithm. It also performs significantly better than either the Autocorrelation method or the modified Medan PDA using polyphase vectors as its interpolation technique.

3.8 Conclusion

This chapter presented an alternative approach in reducing the complexity of the PDA developed by Medan to the one proposed in [27]. Reducing the complexity was
achieved by decimating the speech signal. It potentially reduces the complexity by a factor of $D^2$, where $D$, is the decimation factor. Unfortunately this decimation also reduces the resolution in determining the pitch period estimate. Medan suggested the use of polyphase vectors to compensate for this reduction. This research found limitations in the technique, clearly exposing it as having very little effect in improving the reduced resolution pitch period estimate. The chapter offered a polynomial interpolation technique to solve the reduced resolution pitch period estimate. For a decimation factor of $D = 3$, the technique was found to exhibit a pitch period estimate equal to the original resolution for over 77% of the database it was subjected to. It also attained a deviation of $\leq 1$ sample of the original pitch period estimate, for over 92% of the database. This is in comparison to the polyphase vector technique, which only had a 56% success rate.

This relates to effectively reducing the complexity of the original PDA developed by Medan nine fold with little compromise in performance. The complexity of the polynomial interpolation technique determined in (3.27) was found to be insignificant to the reductions achieved by decimation.

The performance of the modified Medan PDA (utilising a decimation factor, $D = 3$ and the polynomial interpolation technique) also out performed the autocorrelation method. Table 3.6 indicated that the autocorrelation method had a 60% hit ratio with the standard pitch contour, while the former had a 73.2% hit ratio. It was shown that the complexity of the autocorrelation method is four times that of the modified Medan PDA.

This chapter has presented evidence to suggest that the Modified Medan PDA is a possible candidate in low complexity speech coding algorithms.
CHAPTER 4

EFFICIENT IMPLEMENTATION OF LINE SPECTRAL PAIRS (LSP)

4.1 Introduction

This chapter focuses on the Chebyshev polynomial expansion technique, initially reported in [16] as a method of eliminating the computational expensive trigonometric evaluations, associated with transforming the LPC coefficients into LSP frequencies and its reconversion.

Chapter 2, "Speech Coding Systems: A Review" associated trigonometric evaluations as undesirables in real time implementations as they are inefficiently evaluated by DSP processors. The adaptation of the Chebyshev polynomial expansions in computing the LSP frequencies is presented in Section 4.4. This technique replaces a cosine evaluation with a single MAC (DSP instruction) defined by (4.27). The trigonometric evaluations are also eliminated in the reconversion process by representing the LSP frequencies in the cosine frequency domain ($x$ domain, where $x = \cos \omega$) instead of the conventional frequency domain ($\omega$-domain). In other words, LSP frequencies which are normally located on the upper semi-circle of the z-plane (illustrated in Figure 4.1) are now mapped to the real interval $[+1,-1]$. Section 4.5 details how the prediction filter's reconstruction polynomials require cosine evaluations of the LSP frequencies but are not required when using the LSP coefficients mapped to the real interval.

The formulation of LSP frequencies from the Linear Prediction filter, $A_m(z)$ is described in Section 4.2 to assist in understanding the overall complexity of the LSP
transformation. This section also presents the methodologies in obtaining the LSP frequencies from a set of polynomials, which reconstruct the Linear Prediction filter, namely the phase response method [15] and the magnitude method [14]. In describing both methods, the phase response method is shown to require greater trigonometric evaluations in obtaining the LSP frequencies than the later. Section 4.3 presents the polynomials required to solve for LSP frequencies, using the magnitude method, in the conventional frequency domain. The implementation complexity of solving the roots of these polynomials will be compared to solving the roots of the polynomials using the Chebyshev expansions in Section 4.7. The roots of the polynomials are the LSP frequencies.

It is crucial that the proposed technique does not compromise the accuracy in representing the spectral shape of speech by adding unwanted distortion. This is investigated in Section 4.6, with results indicating that the performance of the proposed technique of using Chebyshev polynomial expansions compares favourably with the conventional method. Section 4.6 illustrates that the proposed method achieves spectral transparency.

4.2 Line Spectrum Pairs

LSP frequencies can be derived by decomposing the Prediction filter, $A_m(z)$ into symmetric and anti-symmetric polynomials $P(z)$ and $Q(z)$. The polynomials can be expressed as

$$P(z) = A_{m+1}^+(z) = A_m(z) + z^{-(m+1)}A_m(z^{-1}) \quad (4.1)$$

$$Q(z) = A_{m+1}^-(z) = A_m(z) - z^{-(m+1)}A_m(z^{-1}) \quad (4.2)$$
where

\[ A_m(z) = \sum_{i=0}^{m} a_i z^{-i} \]  \hspace{1cm} (4.3)

The roots of these two polynomials determine the LSP frequencies. The Prediction (LPC analysis) filter, \( A_m(z) \), may be reconstructed using these two polynomials in the following manner

\[ A_m(z) = \frac{P(z) + Q(z)}{2} \]  \hspace{1cm} (4.4)

The polynomials \( P(z) \) and \( Q(z) \) may be described in factored form,

\[ P(z) = (1 + z^{-1}) \prod_{i=1}^{m/2} (1 - 2 \cos(\omega_{2i-1}) z^{-1} + z^{-2}) \]  \hspace{1cm} (4.5)

\[ Q(z) = (1 - z^{-1}) \prod_{i=1}^{m/2} (1 - 2 \cos(\omega_{2i}) z^{-1} + z^{-2}) \]  \hspace{1cm} (4.6)

where \( \omega_{2i-1} \) and \( \omega_{2i} \) are the LSP frequencies.

Equations (4.1) and (4.2) can be expressed as

\[ P(z) = A_m(z) \left[ 1 + z^{-(m+1)} \frac{A_m(z^{-1})}{A_m(z)} \right] \]  \hspace{1cm} (4.7)

\[ Q(z) = A_m(z) \left[ 1 - z^{-(m+1)} \frac{A_m(z^{-1})}{A_m(z)} \right] \]  \hspace{1cm} (4.8)

By defining \( H_m(z) = z^{-(m+1)} \frac{A_m(z^{-1})}{A_m(z)} \), \( P(z) \) and \( Q(z) \) can be described as
\[ P(z) = A_m(z)[1 + H_m(z)] \]  \hspace{1cm} (4.9) \\
\[ Q(z) = A_m(z)[1 - H_m(z)] \]  \hspace{1cm} (4.10)

The all-pass system \( H_m(z) \) can be factorised in the form of

\[ H_m(z) = z^{-1} \prod_{i=1}^{m} \frac{z^{-1} - c_i^*}{(1 - c_i z^{-1})} \]  \hspace{1cm} (4.11)

where \( A_m(z) \) is minimum phase with zeroes, \( c_i = r_i e^{j \omega_i} \), located inside the unit circle, \( r_i < 1 \).

The roots of \( P(z) \) and \( Q(z) \) are determined when these polynomials solve to zero. The only solution for \( P(z) \) and \( Q(z) \) to equal zero is for \( H_m(z) \) to equal -1 or 1 respectively. This only occurs when \( |z|=1 \), requiring all zeroes of the LSP polynomials to be on the unit circle. With the zeroes on the unit circle, the root finding procedure becomes easier, as \( P(z) \) and \( Q(z) \) need only be evaluated on the unit circle, as described in [14].

Therefore \( H_m(\omega) \), the frequency response of the all-pass filter, will alternately take on values 1 and -1, corresponding to zeroes in \( P(\omega) \) and \( Q(\omega) \), with the first zero appearing in \( Q(\omega) \) for \( \omega = 0 \), and the final in \( P(\omega) \) for \( \omega = \pi \). The zeroes of \( P(\omega) \) and \( Q(\omega) \) are interlaced with each other. The stability of \( A_m(z) \) is preserved after the quantisation of \( P(z) \) and \( Q(z) \), as long as the zeroes of the LSP polynomials are on the unit circle and are interlaced.

The phase function \( \phi(\omega) \) of the all pass system [11] is given as
\[
\phi(\omega) = -(m+1)\omega - \sum_{i=1}^{m} 2 \tan^{-1} \frac{r_i \sin(\omega - \omega_i)}{1 - r_i \cos(\omega - \omega_i)}
\]  

(4.12)

where it is shown in [14] that \(\phi(\omega)\) is a monotonic decreasing function with \(\phi(0) = 0\) and \(\phi(2\pi) = -2\pi(m+1)\). Now \(P(\omega)\) and \(Q(\omega)\), the frequency response of \(P(z)\) and \(Q(z)\) equal zero when the phase response, \(\phi(\omega)\), is a multiple of \(\pi\). This is shown in [15] as an alternative procedure to evaluating the LSP frequencies and shall be called the phase response method.

4.3 Computation of the LSP Frequencies

When the prediction filter's order, \(m\), is at most 8, the LSP frequencies can be found analytically, as \(P(z)\) and \(Q(z)\) can be represented as 4th order polynomials. A prediction filter of tap length greater or equal to ten is generally needed to accurately represent the short term speech spectrum [30]. Voiced speech in the frequency domain, shown in Figure 2.3, has a number of formants (peaks) in its spectrum. These peaks often called resonances can be individually modeled by a 2nd order all-pole filter. Numerical methods are used to evaluate the LSP frequencies for these tap lengths. \(P(z)\) and \(Q(z)\) in (4.1 & 4.2) expand out to odd powers of \(z^{-1}\). These polynomials have a root at \(z = -1\) and \(z = 1\) respectively and remain fixed. Consequently these roots may be removed using polynomial division, to give symmetric polynomials of even powers of \(z^{-1}\). These can be expressed by the following

\[
P'(z) = \sum_{i=0}^{m} p_i z^{-i}
\]

(4.13)

\[
Q'(z) = \sum_{i=0}^{m} q_i z^{-i}
\]

(4.14)
where \( p_0 = q_0 = a_0 \) and due to the symmetric nature of the polynomials

\[
p_i = a_i + a_{m-i} - p_{i-1} \quad \text{for } i = 1, \ldots, \frac{m}{2} \tag{4.15}
\]

\[
p_i = p_{m-i} \quad \text{for } i = \frac{m}{2} + 1, \ldots, m \tag{4.16}
\]

\[
q_i = a_i - a_{m-i} + q_{i-1} \quad \text{for } i = 1, \ldots, \frac{m}{2} \tag{4.17}
\]

\[
q_i = q_{m-i} \quad \text{for } i = \frac{m}{2} + 1, \ldots, m \tag{4.18}
\]

The \( a_i \) are the coefficients of the prediction filter \( A_m(z) \). The frequency response for \( P'(z) \) and \( Q'(z) \) can be described as

\[
P'(\omega) = \sum_{i=0}^{m} p_i e^{-j\omega i} \tag{4.19}
\]

\[
Q'(\omega) = \sum_{i=0}^{m} q_i e^{-j\omega i} \tag{4.20}
\]

The factorisation of \( P'(\omega) \) and \( Q'(\omega) \) in (4.19) and (4.20) by the phase \( e^{-j(m/2)\omega} \) leads to

\[
P'(\omega) = e^{-\frac{m}{2}\omega} \sum_{i=0}^{m} p_i e^{-j\left(i \frac{m}{2}\omega\right)} \tag{4.21}
\]

\[
Q'(\omega) = e^{-\frac{m}{2}\omega} \sum_{i=0}^{m} q_i e^{-j\left(i \frac{m}{2}\omega\right)} \tag{4.22}
\]
By taking into consideration the symmetric qualities of \( p_i \) and \( q_i \) in (4.16 & 4.18), the above equations may be expressed as

\[
P'(\omega) = e^{-\frac{j\omega}{2}} \sum_{i=0}^{m-1} p_i (e^{j\frac{m-1}{2}\omega} + e^{-j\frac{m-1}{2}\omega}) + e^{-\frac{j\omega}{2}} \frac{p_m}{2} \quad (4.23)
\]

\[
Q'(\omega) = e^{-\frac{j\omega}{2}} \sum_{i=0}^{m-1} q_i (e^{j\frac{m-1}{2}\omega} + e^{-j\frac{m-1}{2}\omega}) + e^{-\frac{j\omega}{2}} \frac{q_m}{2} \quad (4.24)
\]

The terms between the square brackets of (4.23 & 4.24) can be expressed in the form of a cosine function. Consequently by disregarding the scalar phase factor, \( e^{-\frac{j\omega}{2}} \), due to it having no effect on the positioning of the roots, it leads to the following

\[
P'(\omega) = 2 \sum_{i=0}^{m-1} p_i \cos \left( \omega \left( \frac{m}{2} - i \right) \right) + \frac{p_m}{2} \quad (4.25)
\]

\[
Q'(\omega) = 2 \sum_{i=0}^{m-1} q_i \cos \left( \omega \left( \frac{m}{2} - i \right) \right) + \frac{q_m}{2} \quad (4.26)
\]

The procedure utilised in [14] evaluates the above equations on the unit circle using the discrete cosine transform at points determined by a fine grid. Sign changes at adjacent grid points isolate intervals, which contain the roots of the polynomials. Further bisections of these intervals give refined estimates of the actual root positions. An equispaced frequency grid with 100 to 150 points within the interval is sufficiently small to avoid missing sign changes. Four subsequent bisections guarantee an interval smaller than the difference between roots of \( P'(\omega) \) and \( Q'(\omega) \) [16]. The root finding
procedure begins by evaluating $P'(\omega)$ at points determined by the grid. Once a root has been isolated and the interval is refined, the root finding procedure is subjected to $Q'(\omega)$. The root finding procedure alternates between $P'(\omega)$ and $Q'(\omega)$ until all the roots are found.

The phase response method proposed in [15], as mentioned earlier, determines the roots of the polynomials when the phase response of the all-pass filter, given in (4.12), is a multiple of $\pi$. This is not appropriate for real-time implementation, as the evaluation of the phase response of the all-pass filter is significantly high in terms of complexity and storage. This is due to the numerous trigonometric functions needed to be evaluated and or trigonometric tables stored.

The Chebyshev method proposed in [16] requires no prior storage or calculation of trigonometric functions. The method utilises Chebyshev polynomial expansions and an efficient recursive numerical algorithm to evaluate (4.25 & 4.26) and subsequently find the roots. By subjecting the same Chebyshev polynomial expansion described in [16] on (4.25 & 4.26), the number of Multiply Accumulates (MAC) required to evaluate these LSP polynomials is $2m$. The formulation and evaluation of the Chebyshev polynomial expansion on $P'(\omega)$ and $Q'(\omega)$ is described in the next section.

4.4 Formulation of LSP Frequencies using Chebyshev Polynomial Expansions

Consider the frequency mapping $x = \cos(\omega)$, then $\cos(k\omega) = T_k(x)$, where $T_k(x)$ is the $k^{th}$ order Chebyshev polynomial as a function of $x$. Now the Chebyshev polynomials satisfy the order recursion

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$ (4.27)
where \( T_0(x) = 1 \) and \( T_1(x) = x \).

This recursion procedure can be computed using one MAC instruction. \( T_k(x) \), where \( k = m/2 - i \), is substituted for \( \cos((m/2 - i)\omega) \) in (4.25 & 4.26) giving

\[
P'(x) = 2 \sum_{i=0}^{m/2-1} p_i T_{m/2-i}^i(x) + p_m
\]

\[
Q'(x) = 2 \sum_{i=0}^{m/2-1} q_i T_{m/2-i}^i(x) + q_m
\]

(4.28) (4.29)

where the factor 2 is disregarded as it doesn't affect the root locations.

Now the root finding procedure is similar to the conventional method outlined in [14], except that the polynomials are now evaluated in the \( x \)-domain, as compared to \( \omega \)-domain. In other words, the polynomials are evaluated at points across the real interval of the \( z \)-plane, \( x \)-axis, as opposed to points around the unit circle, depicted in Figure 4.1.

As the roots appear in complex conjugate pairs, shown in (4.5 & 4.6), a root located on the upper half of the unit circle corresponds to another root mirrored onto the bottom half. Thus only points within the interval \([0, \pi]\) are required to locate all the roots. In the \( x \)-domain the interval corresponds to \([1,-1]\). The \( x \)-axis on the unit circle is divided into 100 equidistant grid points to eliminate the probability of two roots connected with the same polynomial existing in the same grid interval. The subsequent four bisections guarantee a degree of refinement on the location of the root. This also decreases the probability of missing a root or interchanging its order in the process of switching the search of roots from one polynomial to the other. The worst case uncertainty in locating the roots in the \( x \)-domain is determined by the number of equidistant points on the \( x \)-
axis and remains constant. Unfortunately due to the non-linearity between the $x$ and $\omega$ domain, the uncertainty in the $\omega$-domain varies considerably between 1.9Hz at middle frequencies and 70Hz at low and high frequencies, assuming four bisections on an interval. But for a narrow bandpassed signal [300-3400Hz], the uncertainty remains less than 10Hz.

Figure 4.1: Possible root locations for an even order $P'(z)$ and $Q'(z)$.

The procedure for finding the roots of $P'(z)$ and $Q'(z)$ can be summarised as follows;

a) The initial spacing $\Delta x$ is determined by the number of grid points on the interval and the convergence threshold $\epsilon$, determined by the number of bisections in the interval. In the case of 100 points and 4 bisections, $\Delta x = 0.02$ and $\epsilon = 0.0015$. 
b) Initially \( \hat{x}_0 = 1 \) and \( P'(\hat{x}_0) \) is evaluated.

c) Increment \( k \) and set \( \hat{x}_k = \hat{x}_{k-1} - \delta x \). Evaluate the polynomial \( P'(\hat{x}_k) \).

d) Check for a sign change in the interval \([\hat{x}_{k-1}, \hat{x}_k]\), by evaluating \( P'(\hat{x}_k)P'(\hat{x}_{k-1}) < 0 \).

e) Repeat from c) if no sign change. If a sign change is detected, set \( x_l \) to equal \( \hat{x}_k \) and \( x_r \) to \( \hat{x}_{k-1} \). The zero is located in the interval \([x_l, x_r]\).

f) Bisect the interval \([x_l, x_r]\) to create sub-intervals \([x_l, x_m]\) and \([x_m, x_r]\) where \( x_m = (x_l + x_r)/2 \). Evaluate \( P'(x_m) \) and determine in which sub-interval the sign change takes place using a similar method to d). Replace the previous interval boundaries \([x_l, x_r]\) with the appropriate sub-interval boundaries.

g) If \( |P'(x_m)| < \varepsilon \), a root of the polynomial \( P'(x) \) has been found within the required accuracy. The root is given as the location of \( x_m \). If not, repeatedly bisect the interval and evaluate g). Maximum bisection is limited to four times.

h) Once a root from \( P'(x) \) is found, the root procedure is exposed to the polynomial \( Q'(x) \). The procedure alternates between the polynomials as each root is found. The procedure starts by setting \( \hat{x}_k = x_m \), and repeating from c).

The process of quantisation is initiated once the LSP coefficients have been found. Different techniques of quantising the LSP have been studied over the years, including Scalar and Vector Quantisation [23]. In the FS1016 CELP standard [24], The LSP scalar quantisers are represented as frequencies. Thus quantising the LSP coefficients,
represented in the cosine frequency domain, requires extra trigonometric computations (arc-cosine evaluations) to express the roots in the conventional \( \omega \)-domain. Since these are a scalar set of quantisers, the conversion can be avoided by applying the cosine non-linearity to the cosine boundaries and output values of the LSP quantiser. This allows for the elimination of the procedure converting the \( x \)-domain LSP coefficients to frequencies just to accommodate the LSP quantisers [16].

4.5 Conversion of LSP Coefficients to LPC Coefficients

The conversion process from LSP frequencies to LPC coefficients is less computational expensive than deriving the LSP frequencies from LPC coefficients, as there is no root finding procedures involved. As seen in (4.5 & 4.6) the roots (LSP frequencies) appear as complex conjugate pairs and give rise to second order polynomials in the form of \( 1 - 2 \cos \omega \; z^{-1} + z^{-2} \). Successive polynomial multiplications of these, will facilitate the reconstruction of the \( P(z) \) and \( Q(z) \) polynomials. From this, the prediction filter \( A_m(z) \) may be reconstructed using (4.4). Again, the need to evaluate trigonometric functions can be eliminated by replacing \( \cos \omega \) with its \( x \)-mapped transformed value, giving

\[
P(z) = (1 + z^{-1}) \prod_{i=1}^{m/2} (1 - 2x_{2i-1}z^{-1} + z^{-2})
\]

\[
Q(z) = (1 - z^{-1}) \prod_{i=1}^{m/2} (1 - 2x_{2i}z^{-1} + z^{-2})
\]

where \( x_{2i-1} \) and \( x_{2i} \) are the LSP coefficients mapped in the \( x \)-domain.
In the latter stages of the previous section, it was discussed how the set of LSP frequency quantisers could be easily transformed into a set of LSP coefficient quantisers (x-domain). This allows for a simply substitution of the quantiser values into (4.30 4.31). The computational complexity of this conversion is constant, unlike the root finding procedure, which relies on the root location. The algorithm requires $2(m+2)$ MAC, where $m$ is the prediction filter's order.

4.6 Performance Evaluation of Reconstructing the LPC Coefficients using Chebyshev Polynomials

The performance of the proposed method is evaluated using the Average Spectral Distortion (SD) measure. This objective measure is used to evaluate the accuracy in subjecting the LPC model to the LSP transformation and its reconversion back to LPC coefficients. It is usually used to evaluate the accuracy in spectral magnitude quantisers. The objective measure is presented in Section 4.6.1. Results in Section 4.6.2 indicate that the proposed method models the spectral characteristics of speech (LPC modeling) transparently.

4.6.1 Average Spectral Distortion

The average spectral distortion measure can be used to evaluate the performance of LPC modeling, essentially used to characterise the short term spectral characteristics of speech signals in speech compression algorithms.

Spectral distortion is defined as the RMS difference between the original LPC log power spectrum and the quantised version averaged over a large number of frames. This may be expressed as
where $S(\omega)$ is the power spectrum evaluated from the unquantised LPC vector, $S_q(\omega)$ is the power spectrum from the quantised LPC vector and $N$ is the number of frames evaluated.

An Average Spectral Distortion (SD) of around 1 dB is usually accepted as the bound for spectral transparency with SD outliers of 2 dB in less than 2% of the database and none greater than 4 dB [23].

### 4.6.2 Objective Measure Results

The proposed method was subjected to an extensive database (2.5 minutes in duration) consisting of utterances from Australian, English and American male and female speakers. The utterances were in the form of 16 bit linear samples and sampled at 8 ksamples/s. They were subjected to a non-overlapping Hanning window of 20ms. LPC analysis was performed using the Levinson-Durbin algorithm [30]. The LPC coefficients obtain from the above algorithm where bandwidth expanded by 15Hz. The LPC coefficients were transformed to LSP coefficients and back to LPC coefficients using the proposed method. The distortion exhibited by this transformation was measured to have an average SD of 0.48 dB.

This indicates that the transformation to and from LSP coefficients in the cosine domain maintains spectral transparency.
4.7 Algorithmic Complexity of Evaluating the LSPs using Chebyshev Polynomials

As mentioned previously 100 equidistant points are necessary to eliminate the probability of missing a sign change in each polynomial and four subsequent bisections to guarantee minimum spacing between roots, taken from each polynomial. The root finding algorithm in its initial search for intervals containing roots associated with polynomials $P'(x)$ and $Q'(x)$ requires approximately $N_g$ evaluations, where $N_g$ is the number of points on the fine grid. In the worst case scenario (final root located near -1 of the real axis of the unit circle), assuming $N_g = 100$, 100 evaluations will be needed.

In most cases all the roots are found prior to reaching the final points on the grid, which will decrease the number of evaluation needed.

After each sign change has been isolated, four subsequent bisections refine the root position to an acceptable uncertainty. Therefore the root finding algorithm in its refinement of the root locations uses $4m$ evaluations, where $m$ equals the predictor filter tap length. Assuming that $m = 10$, 40 evaluations are needed. At worst, a total of 140 polynomial evaluations are needed to find all the roots (LSP coefficients).

In using the conventional method of evaluating the polynomials on the unit circle, seen (4.25 & 4.26), the complexity of the algorithm consists of $m$ MAC and $m-2$ cosine evaluations. The implementation complexity differs slightly to the algebraic complexity, being defined as the number of MAC operations needed to implement the algorithm on a DSP processor. In this case the implementation complexity of the whole algorithm is expressed as

$$ (N_g + 4m) \times (m + (m - 2)) \times (N_{\text{dig}}) + N_{\text{over}} $$

(4.33)
where

\[ m = \text{the order of the predictor filter.} \]

\[ N_g + 4m, \text{ is the number of polynomial evaluations required (140).} \]

\[ N_{\text{trig}} = \text{special trigonometric function (cosine).} \]

The number of MAC required to implement the cosine function is dependent on the DSP processor. It takes 24 instruction cycles to implement the cosine function on the AT&T DSP32C-80 floating-point processor [49].

The overhead, \( N_{\text{over}} \), can include wait states due to the pipelining of the DSP architecture, set up instructions, such as memory pointers, and saving contents of memory pointers before subroutine computations. Trigonometric functions are available from the DSP's software library.

In the proposed method, where Chebyshev polynomial expansions are used in evaluating the LSP polynomials, seen in (4.27) and (4.28 & 4.29), the algebraic complexity consists of \( 2m - 2 \) MAC operations. The implementation complexity of the algorithm may be expressed as

\[ (N_g + 4m) \times (2m - 2 + N_{\text{over}}) \]

By comparing both algorithms, the latter algorithm's implementation complexity is approximately equal to its algebraic complexity, except for the overhead, whereas the former is dependent on the trigonometric evaluations. As most trigonometric computations take more than one MAC instruction, the proposed method is much more efficient. Given that there are 100 points on the grid and ten LSPs, and the cosine
function is implemented using 24 instructions (MAC), the conventional method requires 28280 MACs per 20ms frame (1.4MIPS). Note the overhead is assumed to be zero for simplicity. The proposed algorithm requires 2520 MACs per 20ms frame (0.126MIPS). Thus the proposed method requires less than one tenth of the processing power that has to be afforded to the conventional method of finding the LSP frequencies. The reduction in complexity is solely due to the elimination of the cosine function evaluation.

In the reconstruction of the predictor $A_m(z)$, the implementation complexity varies depending on whether the conventional method or the proposed method is used. In the previous section, the algebraic complexity of the proposed method was calculated as being $2(m+2)$ MAC operations. The implementation complexity is approximately equal to the algebraic complexity, except for the DSP processor's overhead. Now in the conventional method, illustrated by equations (4.4), (4.5) & (4.6), the implementation complexity can be expressed as

$$2(m+2) + m \times N_{\text{dig}} + N_{\text{over}}$$

(4.35)

This depicts that representing the LSP coefficients in the cosine frequency domain (4.30 & 4.31) reduces the complexity of reconstructing the Prediction filter, compared to the conventional method, as it doesn't rely on the DSP's software library to compute trigonometric functions. The conventional method requires 264 MAC per 20ms frame while the proposed method requires 24 MAC per frame. The complexity associated with the reconversion process is insignificant compared to the LSP finding algorithm.
4.8 Conclusion

This chapter presented an alternative method in computing the LSP coefficients from the prediction filter's coefficients (LPC coefficients) and vice-versa. This method based on the work presented in [16] allowed for an efficient evaluation of the LSP parameters by eliminating the computation of trigonometric evaluations. This was achieved by computing the LSP coefficients in the cosine frequency domain and using Chebyshev polynomial expansions (4.27), (4.28) & (4.29). Conventional methods [14] [15] evaluated the LSP coefficients in the frequency domain (4.25) & (4.26) thereby requiring evaluations of the cosine function. The process of converting the LSP coefficients back to LPC coefficients (4.30) & (4.31) also required no trigonometric evaluations as the LSP coefficients are mapped in the cosine frequency domain.

The proposed method allows the process of computing the LSP coefficients to be independent of trigonometric evaluations (eliminating the need to use DSP software library tools to evaluate them on a DSP). Results by objective measures indicate that the proposed method maintains spectral transparency ($SD = 0.48dB$).

The savings in implementation complexity, offered by the proposed techniques, are proportional to the number of MAC operations required to compute trigonometric functions by a DSP processor. Usually a fixed-point processor requires a greater number of MAC operations compared to a floating-point version. It was demonstrated that the proposed method, if implemented on a DSP32C processor, requires less than 10% of the computational power afforded to the conventional method.
CHAPTER 5

ECHO CANCELLATION

5.1 Introduction

This chapter presents a low complexity line echo canceller that may be integrated with voice compression technology on a single DSP processor. Issues covered in implementing the echo canceller include:

a) Performance evaluations on the echo attenuation and convergence speed of various canceller configurations.

b) Peripherals, such as double talk and far-end talk detectors, used in signaling the canceller when to adapt and essential in real world applications.

c) Techniques in reducing the complexity of the echo canceller.

The motivation behind this research was revealed in Chapter 2 “Speech Coding Systems: A Review”. It was mentioned in Section 2.2.3, that echo (as a result of impedance mismatching in the hybrid) is more noticeable and annoying to listeners with the introduction of low bit rate speech coders in the telephone network. This is attributed to the extra delay introduced to the transmission path by the speech compression algorithms. This delay makes the echo distinguishable from the normal sidetone of a telephone. The benefits of using bandwidth efficient voice technology (compression algorithms) in the PSTN can only be realised with the advent of low cost (low complexity) echo cancellers. This low complexity can facilitate the integration of the canceller and voice compression technology on a single DSP, thereby, reducing cost and power consumption (lowering the chip count) as well as increasing the quality of
service to users by reducing the echo present. Section 5.2 describes briefly the echo canceller’s integration with the telephone network infrastructure and more importantly the vocoder.

This chapter focuses on the FIR filter structure incorporating Least Mean Square (LMS) adaptation procedure and its normalised variant (NLMS) as potential echo canceller candidates. The reasons behind their selection were treated in Section 2.9.3. A formal treatment of the algorithms is presented in Section 5.3. This will serve as background material giving insight to their respective complexity. Section 5.4 compares the performance of the algorithms, investigating the effects of tap size and step adjustment on the echo attenuation level and convergence speed. The methodology applied to characterise the performance of the canceller is closely modeled on the recommendations set out in ITU G.165 and the new standard for echo cancellers in digital networks (ITU G.168).

The NLMS algorithm can at times diverge in environments where it has been set up for fast adaptation and when the energy in the filter taps are low due to silence in conversations [2]. The effects of these environments and the proposed solution involving the effective switching from the NLMS to LMS adaptation in such conditions are detailed in Section 5.6.

A novel technique is introduced in Section 5.7, which reduces the complexity of the echo canceller. The canceller can be categorised essentially as a filtering operation and the adaptation of the filter taps. The complexity reduction technique involves enabling the adaptation procedure for only a portion of time as opposed to it being permanently enabled. In instances where the adaptation procedure is disabled, the canceller reduces to a filtering operation where the complexity is proportional to the number of taps in the
filter structure. Its performance with respect to reducing the complexity and its effect on the convergence time of the canceller is considered.

5.2 Echo Canceller in a PSTN

In the telephone network the near-end echo canceller (adaptive filter) is placed between the hybrid and the speech coder, as illustrated in Figure 5.1.

![Figure 5.1: Speech Coder interfacing with the Near-end Hybrid.](image)

The echo canceller is implemented on the four-wire portion of the telephone network. Being a digital echo canceller, PCM codecs (A/D-D/A converters) are used to sample the signals. The echo (reference) signal from the hybrid can be replicated by applying the far-end talker's (input) signal to an adaptive transversal filter [11]. The output signal from the filter is then subtracted from the near-end talker's (reference) signal, to produce the error (residual) signal. The weights of the filter are adapted to the echo transfer function to minimise the residual.

Peripheral functions are also essential to realise an echo canceller. A double talk detector is implemented to prevent the echo canceller adapting to the near-end talker's signal while canceling the hybrid echo. A far-end talker silence detector is also
included to prevent the echo canceller adapting to only telephone line noise. These are usually based on energy thresholds placed after the A/D and before the D/A.

In Figure 5.1 only half of a full duplex speech coder interfacing with a near-end hybrid is shown. By placing the near-end echo canceller between the hybrid and the speech coder (four-wire connection) for both sides of the telephone connection, talker echo in the telephone network can be eliminated.

It is important to note that by placing the echo canceller between the speech coder and the hybrid, the inherent delay of the speech coder has no effect on the propagation delay between the hybrid and the echo canceller. As the propagation transmission delay between the hybrid and the echo canceller is typically small, a FIR filter structure with a small tap size is sufficient to replicate the echo and ultimately attenuate it to the level prescribed in [1]. Under the recommendations set out in [1], the residual signal must be attenuated by at least 27dB, with reference to the input signal.

5.3 Echo Cancellation Algorithm

The echo canceller (as mentioned in the introduction to the chapter) can be described as a filtering operation and an adaptation procedure whereby the weights of the filter vary to replicate a desired signal (echo). A formal treatment of the filter structure and adaptation procedure (including equations) will now be presented in order to give the reader an insight to the complexity of an echo canceller. Note that this treatment is presented for completeness and is not original. The complexity of the echo canceller will be treated in Section 5.7.

The echo canceller must accurately estimate the echo path and adapt quickly to any variations. Therefore it not only needs to be accurate in replicating the echo path, but it
must converge quickly. The FIR filter offers distinct advantages as a vehicle in modeling the echo path over other structures. These were stated earlier in Chapter 2.

![Figure 5.2: Adaptive FIR filter structure.](image)

The Finite Impulse Response (FIR) filter structure in Figure 5.2 comprises a tapped delay line, whose input signal \( x_j \), at the delay line taps is weighted and subsequently summed.

The output from the adaptive FIR filter can be expressed as:

\[
z_j = \sum_{i=0}^{n-1} h_i^j x_{j-i}
\]

(5.1)

where \( h_i^j \) is the \( i^{th} \) element of weight vector at time \( j \).

The residual (error) can be described as:

\[
\varepsilon_j = y_j - z_j
\]

(5.2)

where \( y_j \) denotes the desired response (received echo) and \( z_j \) the echo replica defined in (5.1).
The weights vector, \( h_i \), is adapted to minimize the residual, \( e_j \), in the mean square error sense.

The Least Mean Squares (LMS) algorithm [3][2] [4] is used to adapt the weights (taps) of the FIR filter structure. It is an implementation of the steepest descent algorithm described by:

\[
h_{i+1}^j = h_i^j + \mu(-\hat{\delta}_i^j) \tag{5.3}
\]

where \( \mu \) is the adaptation constant and \( \hat{\delta}_i^j \) the gradient estimate. The gradient, \( \hat{\delta}_i^j \), is estimated by differentiating the squared error in (5.2) resulting in:

\[
\hat{\delta}_i^j = \frac{d[e_j]^2}{dh_i^j} = -2e_jx_{j-i} \tag{5.4}
\]

Therefore the LMS recursive tap adaptation algorithm in (5.3) becomes:

\[
h_{i+1}^j = h_i^j + 2\mu e_jx_{j-i} \tag{5.5}
\]

Only when the input signal, \( x(j) \), is not correlated, does the gradient estimate, \( \hat{\delta}_i^j \), equal the true gradient and the weights, \( h_i \), converge to their optimum solution [4]. In the case of a highly correlated input signal, such as speech, the weights of the filter structure do not converge to the optimum solution. This results in a higher residual and slower convergence speed. Whitening the input signal can increase the convergence speed.

The adaptation constant, \( \mu \), influences the accuracy and adaptation speed of the weights in modeling the echo response. The effects of this parameter will be treated in Section 5.4.3. The input signal and the number of taps in the adaptive FIR filter, dictate the
maximum bound for the adaptation constant, $\mu$, which ensures the system remains stable [2]. As a rule of thumb the following must be adhered:

$$0 < \mu < \frac{1}{n\sigma_j^2} \tag{5.6}$$

where $n$ denotes the tap size of the filter structure and $\sigma_j^2$ average power of the input signal sample.

Therefore as the number of taps increase, or the input signal power increases, a lower adaptation constant is required to satisfy stability concerns. As the input signal power varies considerably in the case of speech signals, a varying adaptation constant may be considered. One solution is to use the Normalised LMS method [5] [8] where the energy contained in the weights of the filter are used to normalise the gradient estimate. Thus (5.5) becomes

$$h_i(j+1) = h_i(j) + \frac{d[e(j)x(j-i)]}{\sum_{k=0}^{N-1}x^2(j-k)} \quad 0 < a < 2 \tag{5.7}$$

The NLMS algorithm defined in (5.7) has a relaxed stability criterion compared to the LMS algorithm. The criterion is independent of the input signal. The step size, $a$, determines the rate of convergence as well as the accuracy in replicating the echo. The Normalised LMS algorithm is computational more expensive than the basic LMS algorithm described in (5.5). Sophisticated techniques to significantly reduce this complexity are described in Section 5.7.

The advantage of using a NLMS adaptation algorithm as compared to others such as the Least Squares (LS) algorithm is the small number of computations required. The LS algorithm [9] determines the weights that minimise the squared error summed over
time. To fade out past data, exponentially decreasing weights are assigned to the past errors. To obtain the optimum weights, the algorithm requires solution to an inverse matrix, which requires a large number of computations.

5.4 Performance Evaluation of the Echo canceller

The performance of the canceller is dependent on essentially two parameters, namely the number of taps in the filter structure and the step size adjustment in the adaptation procedures. Thus it is crucial that these parameters are optimised for a well performed echo canceller. The following section will describe the methodology used to evaluate the performance of the canceller. Essentially it is based on ITU guidelines [1]. A database has been created consisting of telephone hybrid responses. These simulate the echo present in a telephone connection and will be used to assess the performance of the canceller. A description on how the database was created is presented.

A comparison is made of the two adaptation procedures (LMS and NLMS), illustrating their respective advantages. The effects of step size adjustment and filter tap length on both algorithms are investigated. This comparison will give the reader an insight to the performance of a hybrid of the LMS and NLMS adaptation procedures on a FIR filter structure. This is proposed by the author in Section 5.6, as a solution to possible divergence in NLMS based echo cancellers when set up for fast convergence.

5.4.1 Methodology in Evaluating the Performance of an Echo Canceller

The steady state residual echo level and the convergence speed are important parameters that determine the performance of the echo canceller. These have been specified in the ITU G.165 Recommendations. The steady state residual echo level is defined as the level of echo signal remaining after the output of the echo canceller’s filter has been
subtracted from the reference signal. Thus the level of attenuation in the residual may be defined as:

\[ R_{db} = 10 \sum \log_{10} \left( \frac{|x_j|^2}{|e_j|^2} \right) \]  

(5.8)

The convergence test determines the interval between the instance a test signal is applied to the input of the canceller (with the weights of filter set to zero) and when the residual echo level reaches an attenuation of 27dB. For measurement convenience and repeatability, the guidelines set out in [1] rely on a band-limited white noise test signal for performance evaluation.

Similar tests to [1] have been developed to carry out performance evaluations on the FIR filter using the LMS and NLMS algorithm as its adaptation process. To allow for easy and accurate measurements the tests are implemented in software where the FIR filter structure and the adaptation algorithms are simulated on a PC. The test signal database consists of bandlimited white noise and its associated echo response from a hybrid. As the adaptation algorithms converge faster on an uncorrelated source, a separate database containing speech signals and their respective echo is also utilised. This is to illustrate that the algorithms can properly synthesise and cancel the echo path from a correlated source. Each database contains a test signal of 24 seconds that has been sampled at 8kHz with 15 bits of resolution. The procedure undertaken to create the various echo paths in the database is described in the next section.

The residual echo level and convergence tests will also illustrate the effects of varying the echo canceller's parameters, namely the adaptation constant and the filter structure's tap length. These parameters play a significant role in the performance of the canceller.
5.4.2 Creating a database of hybrid losses

A database was created containing echo paths with various hybrid losses. The echo paths consist of input signals and their respective reference signals. The following system was set up to record the hybrid losses.

A connection is made between two subscribers using standard telephones. Once the connection is made, one of the telephones is replaced by the Speech Processing Testbed [47] illustrated in Figure 5.3, while still maintaining the connection.

This board contains a telephone interface with a gain adjustable hybrid and phone I/O sockets for connection to the DSP development board sitting in a Host PC. The DSP32C development card replays an input signal through the telephone connection (hybrid) and records the reference signal. The hybrid can be adjusted to obtain different losses. The signals are stored on files on the PC ready to be used as echo paths for testing the echo canceller off-line. The input signals include white noise, speech and a train of impulses. The train of impulses allows for the estimation of hybrid loss, delay and dispersion in the echo path. It facilitates a quick estimation of the canceller’s required filter tap size. The white noise input signal was subjected to gain factor of 83.5 dB (comfortable listening level) to utilise the resolution of the A/D-D/A (16 bit linear PCM codec) converter (as white noise has unit power).
5.4.3 Effect of Gradient Step Adjustment on the Residual, Convergence and Stability

The gradient step adjustment is critical to the performance of the echo cancellation algorithm. The adaptation constant, $\mu$, the residual, $\varepsilon_j$, and the input signal, $x_j$, play a significant role in determining the gradient step adjustment used in correcting the weights of the adaptive filter. The gradient step adjustment for the LMS algorithm is defined as $2\mu\varepsilon_j x_{j-i}$.

In order for the algorithm to converge closely to the optimal solution, the gradient step adjustment must be small enough [8]. Larger gradient step adjustments lead to rapid convergence but to a bigger residual (smaller attenuation). These large adjustments do not allow the weights of the filter to converge close to the optimum solution. The greater the distance between the filter coefficients and the optimum solution, the greater the residual. As larger residuals and input signals contribute to larger gradient step adjustments, there is a spiraling effect between the two, possibly causing instability in
the echo canceller. It is important to ensure that the adaptation constant, $\mu$, observes (5.6). Theoretically smaller gradient step adjustments lead to the filter coefficients adapting closer to the optimum solution but with a longer convergence time.

Note that the white noise test signal has been subjected to a gain factor of 83.5dB (mentioned in the previous section). Thus in the experiments undertaken using the LMS algorithm, the adaptation constant is normalised to the average power of this test signal. In this case, the adaptation constant will be subjected to a factor of $5 \times 10^{-9}$, as compared to adaptation constants evaluated on test signals with unit power.

![Figure 5.4: Effect of the Adaptation Constant, $\mu$, on the Convergence Time.](image)

Figure 5.4 illustrates the convergence time for a 60 tap LMS based echo canceller subjected to a white noise test signal. The convergence time is the time taken for the residual attenuation level to reach 27dB, when the weights of the echo canceller are initially set to zero. As the adaptation constant varies between $[10^{-11}, 10^{-10}]$, the convergence time dramatically decreases by 800 percent, thereafter remaining constant until such a time that the adaptation constant violates the stability bound. Figure 5.4
suggests an adaptation constant varying between $[10^{-10}, 10^{-9}]$ for 15 bit resolution samples, is sufficient to meet ITU Recommendations [1] guidelines, which stipulate a convergence time of less than 500 ms.

Figure 5.5 illustrates the residual attenuation, $R_{db}$, as the adaptation constant is varied. The attenuation $R_{db}$ is calculated over a 3 second period when the weights of the canceller have converged. Again a 60 tap LMS based echo canceller, subjected to white noise, is used throughout this test.

![Figure 5.5: Effect of the Adaptation Constant, $\mu$, on the Residual attenuation.](image)

The attenuation level is similar for adaptation constants varying between $[10^{-11}, 10^{-10}]$. This illustrates that the weights do converge in close proximity for most adaptation constants that observe (5.6). As the adaptation constant increases further the residual attenuation decreases. In the instance where $\mu = 2 \times 10^{-9}$, the residual attenuation drops to a level of 25 dB. Thus the level of echo suppression will be almost halved compared to the case where $\mu$ varies between $[10^{-11}, 10^{-10}]$. Tests reveal that a
maximum adaptation constant, $\mu = 8 \times 10^{-10}$, maintains the prescribed level of attenuation (>27 dB) and a fast convergence (180ms). In the case of using the white noise test signal, there is evidence to suggest that further increases to the adaptation constant, $\mu > 2 \times 10^{-9}$ will lead to instabilities. This is directly dependent on the power of the input signal.

In comparison the NLMS adaptation algorithm has a gradient step adjustment defined by $\frac{a[e(j)x(j-i)]}{\sum_{k=0}^{N-1} x^2(j-k)}$. As the gradient step adjustment is normalised by the input signal's energy present in the taps of the filter, the residual attenuation and convergence time is primarily governed by the step size, $a$. Here a higher step adjustment also leads to faster convergence and lower residual attenuation. It is stated in [34], [35] that fastest convergence is attained by the NLMS algorithm when $a = 1$ but the residual attenuation decreases significantly as compared to smaller step sizes. This can be seen in Figure 5.7 where the lowest residual attenuation is found at step size, $a = 1$.

![Figure 5.6: Effect of the step size, $a$, on the Convergence Time of the NLMS algorithm.](image-url)
Figure 5.6 illustrates the convergence time for a 60 tap NLMS based echo canceller subjected to white noise. The NLMS algorithm satisfies ITU recommendations for step sizes varying between $[0.1, 1]$. The NLMS algorithm converges faster than LMS algorithm, having a fastest convergence time of 90ms, while its counterpart has a fastest convergence time of 180ms.

![Image of Figure 5.6: Convergence Time for NLMS Echo Canceller](image_url)

Figure 5.7: Effect of the step size, $a$, on the Residual attenuation of the NLMS algorithm.

At their respective fastest convergence times, the LMS algorithm has a higher residual attenuation compared to the NLMS algorithm [35]. This is supported by our experimental results depicting a 1.1dB difference. The residual attenuation for the NLMS algorithm as the step size is varied is illustrated in Figure 5.7.

![Image of Figure 5.7: Effect of Step Size on Residual Attenuation](image_url)

It is difficult to accurately compare the performance of both algorithms in terms of their step sizes (adaptation constants) as they are considerably different (ie. the LMS algorithm has a fixed step size while the NLMS has a step size normalised by the input signal).
5.4.4 Effect of the number of Filter Taps on the Residual and Convergence Time

Echo paths vary depending on the hybrid and telephone network transmission path. In attempting to cancel near-end echo, the tap length typically varies between 30-60 taps [10].

![Graph showing the effect of filter tap length on residual attenuation for LMS & NLMS algorithms.](image)

**Figure 5.8: Effect of the Filter Tap Length on the Residual Attenuation for the LMS & NLMS Algorithms.**

Figures 5.8 illustrates the effects of the filter tap length on the residual attenuation for the LMS and NLMS algorithms. Here the tap sizes are varied while their step sizes remain fixed for the whole exercise. The adaptation constant for the LMS algorithm is \( (5 \times 10^{-10}) \) while the NLMS algorithm has a step size of \( (0.5) \). The algorithms are subjected to the white noise test signal and the attenuation is calculated over a 3 second period when the weights have converged.

Figure 5.8 illustrates that the residual attenuation decreases as the filter tap length decreases. This is attributed to the algorithms not being able to replicate the echo using...
smaller tap lengths. A filter tap length of 60 ensures that both algorithms attain the recommended attenuation of 27dB.

The hybrid and the telephone network associated with these tests have influenced the results illustrated here. Therefore it can only give an indication on the effect of tap lengths on the residual for this specific echo path. The adaptation constants (step sizes) have been chosen to give similar performance from each algorithm, as the proposed echo canceller, investigated in Section 5.7, encapsulates both adaptation procedures.

The weights of the filter in an echo canceller do not converge to the theoretical optimum solution. Gradient noise will affect the adaptive process during initial transients and steady state conditions [4]. This random noise in the weights of the filter during adaptation causes an excess residual. This can be described as misadjustment, a dimensionless ratio defined as the average excess residual over the "minimum" residual error. The "minimum" residual is attained, when the weights converge to the optimum solution [4].

The performance of the adaptive filter may improve with an increase in the number of taps, but for a fixed rate of convergence, large number of weights increase misadjustment. Hence the convergence time increases (adaptation slows) as the number of taps increase for a given level of performance. Consequently increases in the convergence times with respect to the tap lengths (especially at a tap length of 80) in Figure 5.9 can be attributed to misadjustment, where an increase in the tap length increases the convergence time.
Figure 5.9: Effect of Tap Size on the Convergence Time using the LMS & NLMS Algorithms.

It may be assumed that the relationship between the power of the input signal and the tap length defined by (5.6) plays an insignificant role in this experimental set up, due to the fact that the NLMS adaptation procedure exhibits similar results to the LMS. The step sizes again remain fixed as in the previous case, while the tap lengths vary.

5.4.5 Effect of speech input signals on the Residual

The ITU G.165 recommendations state that the preceding tests on residual attenuation levels and convergence speeds attained by an echo canceller should be performed after the canceller has shown to properly synthesise a replica of the echo path impulse response from a speech input signal and its corresponding echo.

A comparison of the NLMS and LMS canceller’s performance using speech, as the number of taps varies, is illustrated in Figure 5.10.
The average residual attenuation is calculated over the database, discarding measurements where the hybrid loss is less than 6 dB. The adaptation constant, $\mu$, for the LMS is held at $10^{-9}$ and the step size, $a$, for the NLMS is held at 0.5. It can be seen that NLMS canceller outperforms the LMS canceller by approximately 2 dB. The 60 tap NLMS echo canceller performs at the required level for both white noise and speech input signals with the similar residual.

5.5 Outcome of the Performance Evaluation of the LMS & NLMS Algorithms

Section 5.4 has been devoted to the performance evaluation of both the NLMS and LMS adaptation procedures. The effects of step size adjustments and tap sizes on the residual attenuation levels and convergence speeds have shown that the NLMS algorithm, while having a faster convergence time than the LMS algorithm, can produce lower attenuation levels when it has been configured for fast adaptation. Figures 5.8 & 5.9 illustrated that the two algorithms, configured appropriately, perform equally well in terms of the attenuation levels attained and speed of convergence for a FIR filter.
structure varying in tap length. The results were obtained in a bandlimited white noise environment. The configuration details consisted of using an adaptation constant of \(5 \times 10^{-10}\) for the LMS algorithm, while the NLMS algorithm had a step size of 0.5.

The above configuration was set up for algorithms being subjected to speech as an input signal. This experimentation found that the adaptation constant used for the LMS algorithm did not facilitate adequate performance compared to the NLMS algorithm, or even results obtained previously with the white noise input signal (illustrated in Figure 5.8). This is explained by the fact that the performance of the LMS algorithm is dependent on the input signal's energy. A varying input signal unfavourably affects the rate of convergence and may cause filter instability [2]. The energy in the speech and white noise database is significantly different. While the white noise energy is constant, the speech energy varies and in the case of this experiment, is on average lower. This would not affect the NLMS algorithm, as it is independent of the input signal. On the conclusive evidence shown above, the NLMS algorithm is a better candidate for a line echo canceller.

5.6 Peripherals of the Echo Canceller

Previously the chapter had revealed how the performance of an echo canceller is dependent on the number of taps in the filter and the step adjustment. These performance evaluations have been undertaken in ideal conditions using white noise as the input signal to the hybrid. Unfortunately in real world applications such conditions do not exist. An echo canceller in the PSTN is subjected to speech signals and various signaling, including DTMF. In the case of speech, instances of double talk occur frequently during telephone conversations. Figure 5.11 illustrates the echo canceller with respect to the near-end hybrid.
The echo canceller strives to model the input signal response of the hybrid (reference signal). The output of the canceller is subtracted from the reference signal to obtain the residual. The residual is the difference between the input signal response of the hybrid and the input signal response of the adaptive FIR filter in the canceller. Thus if the weights of the filter have accurately modeled the characteristics of the hybrid, then the residual (echo) will be negligible.

As the interest here is to cancel echo from the near-end hybrid it is crucial that the reference signal is not corrupted by signals from the near end telephone in the instances when the weights of the filter are adapting to the desired response (reference signal). Reference signals corrupted by near end energy would consequently force the weights to diverge from the characteristics of the hybrid and subsequently increase the level of the echo present in the telephone connection. Thus adaptation should only occur when the reference signal contains the response of the hybrid to the input signal (far end speech). Adaptation should not occur when the reference signal contains near end energy, as this energy has no relation to the echo path response that the canceller is trying to adapt to. Thus the correct moment for adaptation by the canceller is when the far end speaker is talking and when the near end speaker is not. This guarantees that the
reference signal contains only the response of the hybrid to the input signal, and not any near end speech.

There are numerous talk detection algorithms that may be used to signal the canceller to adapt to the desired response (reference signal). One such method might be an energy threshold detector. This is usually performed by measuring the input and reference signal energies and follows the subsequent procedure below:

\[
\text{if } E_{\text{input}} > C_1 \text{ and } E_{\text{ref}} < C_2 \text{ then adapt} \quad (5.9)
\]

where:

\[
E_{\text{input}} = \sum_{n=0}^{N_z-1} \hat{s}^2(n) \quad (5.10)
\]

\[
E_{\text{ref}} = \sum_{n=0}^{N_z-1} s^2(n) \quad (5.11)
\]

and \( \hat{s}(n) \) is the input signal (far end speech), \( s(n) \) is the reference signal (near end speech plus hybrid response to far end speech), \( N_z \) is the frame length, \( C_1 \) and \( C_2 \) are experimentally derived constants. The frame length parameter is included due to the canceller having to integrate with voice compression technology.

Voice compression algorithms utilise parametric modeling, requiring speech to be buffered into frames of 160-240 samples (sampled at 8 ksamples/s). Consequently the input and reference signals accessed by the echo canceller are updated at the frame rate of the compression algorithm.

This talk detection logic switches on adaptation when the far end energy exceeds \( C_1 \) and the near end energy is smaller than \( C_2 \). This logic assumes that during adaptation the reference energy \( E_{\text{ref}} \) will not exceed \( C_2 \), a reasonable assumption for hybrids with
losses greater than 20 dB. If $E_{ref}$ exceeds $C_2$ it is assumed near end speech is present and adaptation stops.

The problem experienced with the talk detector logic in (5.9) occurs with low hybrid loss systems, 6 to 10 dB. In the presence of far end speech only, $E_{ref}$ could easily exceed $C_2$ due to the small amount of attenuation from the hybrid. Thus in many otherwise valid frames, adaptation is disabled, even though no near end speech is present. This results in noticeably slower adaptation for poor hybrids (around 6-10 dB loss), as compared to other systems with hybrid losses exceeding 20 dB.

Adjusting the thresholds could provide some relief to this problem, however the results would only be useful for a given hybrid loss. For example, if the system were optimised for 6 dB hybrids, it would perform poorly for 20 dB hybrids.

The following adaptation logic rule is independent of hybrid loss as long as it meets or exceeds the 6 dB threshold:

$$\text{if } E_{\text{input}} > C_1 \text{ and } E_{\text{input}} \leq 2E_{\text{ref}} \text{ then adapt} \quad (5.12)$$

This logic requires the input energy to be at least twice the magnitude of the reference energy. This condition will be true with hybrids greater than 6 dB loss. This adaptation rule will suffice as it is stated in the ITU G.165 that the minimum hybrid loss required for an echo canceller to function must be 6 dB. This adaptation logic rule is loosely based on the algorithm by A.A. Geigel [48] consisting of declaring near-end speech whenever

$$|y_j| \geq \frac{1}{2} \max(|x_j|, |x_{j-1}|, |x_{j-2}|, ..., |x_{j-N}|) \quad (5.13)$$
where $|y_j|$ is the absolute magnitude of the reference signal and $|x_j|, |x_{j-1}|, |x_{j-2}|, \ldots, |x_{j-N}|$ are the absolute magnitudes of the samples in the FIR filter's memory. This essentially performs an instantaneous power comparison over a window spanning the echo path delay range (filter range). The preceding adaptation logic rule performs a comparison of the energies over a window spanning the frame length of the compression algorithm.

At some instance the adaptation logic rule will fail, as the near-end speech energy will be low enough to activate adaptation even though there is evidence of double talk. This may be solved by introducing a hangover period to the adaptation. Thus it continues to declare near-end speech present after initial detection for the duration of the hangover period, which is 75ms.

Another procedure in overcoming double talk is to have two separate echo path models, one in the background adaptively identifying the echo path transfer characteristics and the other for synthesising the echo replica to cancel out the echo [36]. The parameter values of the latter are refreshed by those of the background only when the latter is deemed to give a better echo path characteristics than the former. Unfortunately this procedure is computational expensive as it requires two filtering operations to be performed, instead of the standard one found in traditional cancellers. This increases the complexity of the canceller by 25%. The complexity of the canceller will be discussed in Section 5.7.

Thus the double talk detector defined in (5.12) is a suitable candidate for implementing the peripherals associated with a low complexity line echo canceller. It requires the
energy calculations of far-end and near end signals. A low complexity technique in calculating these energies is discussed in Section 5.7.

As mentioned previously, the canceller uses adaptive filtering to model the echo path. Thus the dynamics of the canceller may be evaluated by monitoring the movements of the energy in the weights of the filter. Particular movements of the weights can be traced over time and give indication of convergence, divergence and no adaptation.

![Graph showing energy in the weights over time.]

Figure 5.12: Trajectory of the energy in the weights for the canceller and the input signal.

It was noticed in experiments, carried out with speech as the input signal, that the echo canceller using the NLMS algorithm at times caused large levels of echo (divergence) when the step size had been tuned for fast adaptation \((a = 0.5)\). This divergence was audible and is better illustrated in the movement of the weights of the filter in Figure 5.12.
The energy in the weights of the NLMS canceller shoots up momentarily, indicating a sudden large adjustment in the weights. These sudden large adjustments correspond to divergence in the canceller (as the weights eventually adapt towards the optimum solution after this event).

This is indicated in Figure 5.12 by the three large spikes in the energy of the weights against a backdrop of relative input signal energy levels. Normally the trajectory of energy in the weights should rapidly ramp within the first 500ms and then hover around that energy for the rest of its operation. This corresponds to the canceller converging within the 500ms. Note the adaptation logic rule used for this illustration did not include hangover.

The NLMS algorithm can suffer from instabilities if the energy in the taps of the filter is low [2]. This condition causes large step adjustments and leads to coarse echo path modeling of the near-end hybrid. The adaptation process is an iterative approach and the large adaptation constants combined with larger residual signals, eventually cause filter instability (breakdown of the canceller). This is illustrated in Figure 5.12 where the large spikes are associated with low input signal energy.

For the NLMS algorithm to work effectively it is therefore necessary to monitor the energy in the taps of the filter constantly, similar to the algorithm in (5.13). This scheme would operate separately to the talk detection algorithm described in the previous section. The talk detection algorithm works on the principle of calculating the energy of the input and reference signal over a whole frame (20-30ms interval), and making the appropriate decision. It is a binary decision in which the canceller’s adaptation process is enabled or alternatively disabled.
Having a hangover period of 75-80ms on the far-end talker (input signal) detector will make sure that the input signal energy is not in a transition region (falsely enabling adaptation) and should alleviate these spikes. This is illustrated in Figure 5.13 where the large spikes, evident in Figure 5.12, associated with the NLMS algorithm having no hangover, have been eliminated. There is still some evidence of divergence in the energy in the weights after initial convergence, illustrated by minor spikes in the movement of the weights in Figure 5.13.

Figure 5.13: Trajectories of the energy in the weights of the modified NLMS and NLMS with hangover and the input signal.

The NLMS algorithm is sensitive to periods of silence in the input signal. The talk detection algorithm doesn’t detect instances of silence within the frame but makes a decision on whether to enable adaptation based on the overall energy in the frame. Frames containing a mixture of large magnitude samples and periods of silence (greater
in duration than the tap size of the filter) occasionally occur. These will enable the adaptation process, and lead to the above mentioned problem.

This may occur when transmitting DTMF tones. The tones being generated are for a specific period of time followed by silence. Over a particular 20ms coder frame one might find a mixture of tone and silence. Under these conditions, the talk detection algorithm will enable adaptation, and due to the uneven mixture of high and low energy in the frame the adaptation process will produce large step adjustments causing the echo levels to rise in the telephone connection momentarily.

The large step adjustments may also occur when the far-end talker is finishing an utterance, resulting in a frame with a mixture of low energy utterance and silence. These large step adjustments correspond to the spikes (dashed line) in Figure 5.13. The author's solution to this problem is to implement a tap energy estimator. If the energy in the taps is greater than a defined threshold, $C_3$, then the canceller adapts according to equation (5.7), else it adapts to a fixed adaptation constant $\mu_{\text{fix}}$:

$$\text{if } E_{\text{tap}} > C_3 \text{ then } \mu = \frac{a}{E_{\text{tap}}} \text{ else } \mu = \mu_{\text{fix}}$$

(5.14)

where:

$$E_{\text{tap}} = \sum_{k=0}^{N-1} x^2(j - k)$$

(5.15)

and $\mu_{\text{fix}} = \frac{a}{C_3}$. The energy threshold, $C_3$ is calculated by making sure it is less than $\frac{1}{\mu}$ where $\mu$ is calculated by (5.6) in order to prevent filter instability.
The modified NLMS algorithm is a hybrid of the LMS and NLMS algorithm. In instances of low energy or silence in the input signal and adaptation enabled (by the talk detection logic) the canceller adapts according to the LMS algorithm, otherwise, the canceller operates the NLMS algorithm. The LMS algorithm's step adjustment is independent of the input signal. Thus the low energy in the input signal has no effect on the step adjustment of the LMS algorithm. Thus the modified NLMS algorithm, as shown by the solid line in Figure 5.13, eliminates the spikes associated with the NLMS algorithm. Thus the divergence exhibited by the NLMS based canceller is removed. After initial convergence, the energy of the weights remain relatively constant (having a flat trajectory) and exhibit the normal characteristics of a canceller. Thus this hybrid canceller takes advantage of fast convergence associated with the NLMS algorithm and the robustness of the LMS algorithm.

Messerschmitt reported in [54], that echo cancellers should have the following fundamental requirements:

a) Rapid convergence on a new connection.

b) High residual attenuation during single talk (no near end signal).

c) Slow divergence when there is no input signal.

d) Rapid return of the echo level to residual if the echo path is interrupted.

e) Little divergence during double talk.

The modified NLMS algorithm using a FIR filter structure is the author's proposed solution for echo cancellation. It takes advantage of the fast adaptation of the NLMS algorithm and considers the real world problems, discussed in the previous two sections,
affecting the canceller. These include eliminating divergence when there is no (or low) input signals (problem with the standard NLMS algorithm) and when there is double talk.

5.7 Complexity of the Echo Canceller Algorithm

Estimating the algorithmic complexity of the canceller involves calculating the number of Multiply-Accumulates (MACs) necessary to implement the algorithm. The canceller using the NLMS adaptation algorithm consists of equations (5.1), (5.7) and (5.2) and requires $N$ filtering operations ($N$ MACs), $N$ filter coefficient adaptations ($N$ MACs), $N$ operations for calculating the energy in the taps of the FIR filter ($N$ MACs) and one operation for calculating the residual signal (1 MAC), resulting in an algorithmic complexity of $3N+1$ MAC operations for each input sample. This doesn't take into account the normalisation, which requires a divide operation. Divide operations are not single instruction operations on a DSP and take numerous instructions to implement them in software. Thus the complexity of the echo cancellation algorithm is dependent on the tap size, $N$, of the FIR filter structure.

It was mentioned earlier that the echo canceller requires double talk and far end talker detectors to function properly. These are predominantly energy threshold binary switches, which are governed by adaptation logic rules in (5.12). The energy in the input signal (5.10) and reference signal (5.11) must be determined, resulting in $2N_s$ MACs, where $N_s$ is the number of samples in a speech frame. Note that the echo canceller is integrated with a vocoder, and the term speech frame has been introduced for determining the energy levels. This is due to the voice compression algorithms buffering and reconstructing speech signals on a frame basis.
Most DSP manufacturers produce DSP processors in various speed grades to accommodate most signal processing applications. For instance, Texas Instruments (TI) produce the TMS320C5x (fixed point) processor rated at 20, 28.5, 40 and 100 MIPS (million instructions/s). The higher the speed grade the greater the cost. Figures based on TI’s software cooperative [50] indicate that a vocoder such as the FS1016 CELP requires 17 MIPS for implementation on the C5x processor, while the G.729 requires 34 MIPS. Thus the CELP could easily fit on the processor rated at 20 MIPS and the G.729 on the one rated at 40 MIPS. Not shown are the requirements for operating the peripherals such as the PCM codecs interfaced to the serial port of the processors. The common use of DTMF tones in our phone systems today (for dialing and interactive voice response applications) require the vocoders to pass DTMF tones (equivalent functionality to the standard telephone). A DTMF generator and detector requires between 1-1.5 MIPS. Thus integrating an echo canceller and DTMF facilities with any of the above two speech coding algorithms on a single DSP would require it to be implemented within 10-15% of the computational power. This would reduce the implementation cost and chip count. In the case of the CELP algorithm the canceller would be implemented within 1.5-2 MIPS. In the latter, the canceller has up to 6 MIPS at its disposal. Given the above scenario, it is important to reduce the complexity of the canceller without jeopardising its attenuation properties in order to meet constraints set by the processor’s computational power and the complexity of the vocoder. In most cases it is undesirable to use a higher rated processor in order to integrate an echo canceller with a vocoding algorithm, due to higher costs, concerns with Electromagnetic Interference (EMI), and power consumption for hand held applications.

One possible solution for this scenario is to reduce the tap size, $N$, of the FIR filter structure in the NLMS algorithm. The problem associated with this is that the residual
attenuation will decrease because there are insufficient weights in the filter to accurately replicate the transfer function of the echo path. This is illustrated in Figure 5.8 where the performance of the algorithm in terms of attenuation decreases as the tap size decreases. It is crucial that the canceller exhibits a residual attenuation level of not less than 27dB to satisfy CCITT Recommendation G.165 [1]. Thus from the experiments conducted on the hybrid used for this research, a 60 tap FIR filter structure is necessary to cancel the echo present. The weights of the echo canceller adapt according to (5.7) on a sample by sample basis. Therefore to implement an NLMS based echo canceller (with talk detectors) which is integrated to a vocoder would require \((3N + 3)N_s\) MAC operations for each speech frame, where \(N_s\) is the number of speech samples in the frame. If the speech is sampled at 8 ksamples/s then there are 160 samples in each (20ms) speech frame. This presents an algorithmic complexity of 29280 MACs for each frame. It would produce a canceller that requires nearly 1.5 MIPS. Note the normalisation (divide operation) is not included in the calculation.

As mentioned previously, calculating the energy in the taps requires \(N\) MACs/input sample. This may be reduced with respect to the speech frame by the following:

a) Calculate the energy in the taps at the start of each new speech frame.

b) As the weights are updated by the adaptation procedure for each new input sample, the energy is updated by subtracting the energy of the oldest input sample and adding the energy of the newest input sample. This can be considered as a FIFO effect, First In, First Out.

The energy in the taps is evaluated at the start of each speech frame to avoid discontinuities from frame to frame. Decisions whether to adapt or not, due to the talk
detectors, are made at the start of each new speech frame. Thus if the decision is not to adapt, only the filtering operation is undertaken. There is no need to calculate the tap energy as the updating of the weights in the filter is disabled. The above procedure reduces the complexity (MA) for each frame to:

\[ N + (2N + 5)N_s \]  

where the first \( N \) corresponds to the tap energy evaluation at the start of a frame, the \( 2N + 5 \) corresponds to the filtering, the adaptation, the updating the energy in the taps and the talk detectors. Based on the frame update rate and sampling rate used in the previous calculation, the complexity of the canceller would reduce to 1 MIPS (excluding normalisation).

An alternative solution in reducing the complexity of the algorithm is to utilise the adaptation algorithm for only a percentage, \( p \), of the input samples, while still maintaining the filtering operation for the entire input signal samples. By adapting to only \( p \) percent of the speech samples in the frame, the complexity (MACs) for each frame would reduce to:

\[ N + [(N + 5) + N(p / 100)]N_s \]  

where the first \( N \) corresponds to the tap energy evaluation at the start of a frame, \( N + 5 \) corresponds to the filtering, talk detectors and the updating of the energy in the taps. The rest of the complexity corresponds to the adaptation.

Figure 5.14 illustrates the movement of the energy in the weights of the canceller’s filter when adapting to 100%, 50% and 10% of the input signal. The canceller is subjected to a white noise input signal and adapts to a step size of (0.5). All configurations of the canceller are found to reach the 27 dB requirement, specified in [1], for echo attenuation.
and is illustrated in Figure 5.15. The difference in the configurations is the slower convergence time exhibited by adapting to only 10% of the input signal.

Figure 5.14: Illustrates the movement of the energy in the weights of the canceller's filter when adapting to 100%, 50% and 10% of the input signal

Figure 5.14 indicates that the energy in the weights of the canceller (with 10% adaptation) levels off after 500ms. This corresponds to the canceller having converged. It also illustrates that the convergence speed for the standard use of the canceller (100% adaptation) and adapting to 50% of the input signal are similar. Both give convergence speeds of around 90ms. This is also reinforced in Figure 5.16 which depicts the convergence time as a function of the adaptation time.

The performance of the echo canceller using this technique, in terms of reaching the residual level of attenuation specified by [1] (27dB), compares favourably to the full adaptation algorithm. This is illustrated in Figure 5.15 where the attenuation is constant for all adaptation times except when no adaptation takes place. In this case, the attenuation corresponds to the hybrid loss.
Figure 5.15: Residual Attenuation as a function of the adaptation time

Figure 5.16 illustrates the convergence speeds of the canceller as the adaptation is varied. The convergence time remains constant when adaptation is enabled for greater than 50% of the time, but increases steadily below that adaptation percentage. It violates the G.165 recommendations for convergence (500ms) when the adaptation is enabled for less than 12% of the time.

Note these convergence results are dependent on the step size configured for the canceller. In this case the step size is $a = 0.5$ (characteristic of fast adaptation). Lower step sizes (characteristic of slower adaptation) will violate the convergence requirement at a higher adaptation percentage. The complexity reducing technique is invariably bounded by the convergence rate of the canceller.
Figure 5.16: Convergence as a function of the Adaptation time.

Figure 5.17: Illustration of the complexity of the canceller as a function of adaptation time.

Figure 5.17 illustrates the complexity of the canceller using this technique. A designer wishing to use this type of technique would have to determine the processing power available and then find the corresponding adaptation time for the canceller to be enabled. Then the designer would have to investigate using Figure 5.16, whether the
adaptation time compromises the convergence speed or totally violates the convergence requirement of G.165.

The optimum solution based on the above two figures would be to adapt to only 50% of the input samples as convergence is not compromised. Thus a complexity reduction of approximately 30% (compared to full adaptation of the NLMS algorithm) can be achieved for a 60 tap echo canceller, adapting to only 50% of the input samples.

5.8 Conclusion

This chapter addressed the real world problem of near-end echo, associated with low bit rate speech coders integrated on a telephone network. A 60 tap FIR filter structure utilising a modified version of the NLMS adaptation algorithm was found to be suitable in canceling the echo. The effects of the tap size of the filter structure and the step size adjustments in both adaptation algorithms were illustrated in terms of the residual level attenuation and convergence time. Simulation results show that the echo cancellation algorithm satisfies the residual level attenuation and convergence tests specified in [1]. A novel technique in reducing the implementation complexity of the echo cancellation algorithm was described. This has the capacity to reduce the canceller's complexity by half while not compromising the convergence speed. The echo canceller algorithm, integrated with a dual bit rate speech coding algorithm has been implemented in real-time on a single DSP32C processor [37].
CHAPTER 6

CONCLUSION

6.1 Summary of Contributions

This thesis has covered a number of different topics in signal processing and coding. The main objective was to select algorithms of major importance in the area of speech coding and investigate techniques that may reduce the complexity of these algorithms. The complexity reduction techniques discussed ranged from the theoretical to the more practical and application oriented. A summary of the contributions contained in the thesis is listed with respect to the selected algorithms.

Efficient Implementation of the Super Resolution Pitch Determination Algorithm:

1. The technique of reducing the complexity of this PDA, using decimation (of speech signals) and polyphase filtering as the interpolator, proposed in [27], is proved to be ineffective. It assumes that the initial pitch estimate (lower resolution) found after the decimation process is within the temporal resolution of the correct pitch estimate (determined before decimation). This is a requirement for the interpolator to perform properly and estimate pitch with original resolution. This assumption doesn’t necessarily hold and is the contributing factor to its poor performance.

2. The polynomial interpolation technique proposed by the author attempts to directly reconstruct the correlation coefficients that have been determined by the decimated speech signals. This interpolation technique which doesn’t rely on any initial pitch estimate proves to be the more effective in estimating the pitch with original resolution.
Fast Implementation of Line Spectral Pairs (LSPs):

3. On a minor point, investigating the limitations placed upon it by the DSP processor may reduce the complexity of an algorithm. This is accomplished by endeavouring to eliminate special mathematical functions such as square root and trigonometric functions from the algorithms. These usually require many Multiply-Accumulates (MACs) to be implemented on a DSP processor.

This approach was undertaken for the fast implementation of the LSPs transformed from LPC coefficients (LPCs). By evaluating the LSPs on the real axis of the z-plane instead of on the unit circle, the cosine evaluations required previously by the root finding algorithm are replaced by the Chebyshev polynomial expansions as reported in [16]. These expansions require 1 MAC as opposed to the many required by a general purpose cosine function found in the software libraries of a DSP processor. These LSP frequencies have been mapped to the real interval of the z-plane (cosine frequency domain). In the reconversion process of transforming the LSPs to LPCs, there is no requirement of cosine evaluations as the LSPs have been determined in the cosine frequency domain as opposed to conventional frequency domain.

To a lesser extent, the cross-correlation evaluations proposed by Medan in [27] for the PDA, reduced by 30% just by eliminating a square root evaluation.

Echo Cancellation:

4. The integration of an echo canceller and a vocoder on a single DSP processor was discussed. A thorough investigation into the performance of a near-end echo canceller was undertaken using a FIR filter structure and LMS or NLMS adaptation procedure.
5. To test the performance of the canceller, a database of echo paths was sampled using the “Speech Processing Testbed” jointly developed by the author and other members of the Speech Coding and Transmission Group at the University of Wollongong. The database consisted of echo path responses from white noise, speech and impulses. The impulses were recorded to give an indication of delay on a telephone connection.

6. A modified NLMS algorithm was proposed by the author as the canceller to be integrated with the vocoder, after considering the LMS and the NLMS algorithms as potential adaptation procedures in terms of adaptation speed and echo attenuation levels attained. This modified NLMS algorithm is essentially a hybrid of the two schemes (NLMS & LMS) enabling fast adaptation and robustness to low input signals.

7. A novel technique developed by the author to reduce the complexity of the canceller provides options for a designer of the proposed system (integration of a canceller and vocoder on a single DSP processor). The designer initially takes into consideration the processing power available after the vocoder has been implemented and the degree of compromise in the convergence speed. The technique facilitated a complexity reduction of 50% with no compromise in the convergence speed.

This complexity reduction technique has been successfully implemented into a successful commercial product based on the AT&T DSP32C processor. Its main commercial application has been in secure telephone communication systems.

6.2 Future Work

The complexity reducing technique for the echo cancellation algorithm, discussed in this thesis, was based on enabling the adaptation of the filter’s weight for a portion of
the time. Most acoustic echo cancellers use a partial update of the weights, either sequential [54] or selective based [53]. The combining of the partial update techniques with the time based adaptation procedure illustrated in this thesis may prove fruitful, with further gains in reducing the complexity.

6.3 Conclusion

This thesis has presented a number of complexity reducing techniques that are particularly useful for speech coding applications. The research has been successful in offering solutions to some of the real world problems associated with implementing speech coding algorithms for commercial use. The techniques contain elements of academic merit and practical engineering. The major discussions and conclusions arising from the research are included in the relevant chapters and a summary of the contributions has been presented above.

It would be appropriate to draw some general conclusions here:

1. Issues in speech coding have been largely concerned with bit rate versus perceived quality, robustness, delay and complexity. All these issues have been addressed in research undertaken over the last 20 years. The complexity of the speech coding algorithm has taken a greater importance in recent years, as low bit rate speech coding products with the assistance of powerful DSP processors have entered the commercial markets as “off the shelf” components. The constraints of cost due to competition, Electromagnetic Interference (EMI) regulations and power consumption for hand held applications has placed much focus in low complexity implementations.

2. The complexity of the super resolution PDA developed by Medan was reduced by a factor of nine by using simple decimation and (polynomial) interpolation.
techniques. The important factor to maintaining the resolution of the PDA, was the accuracy of the interpolator in reconstructing the cross-correlation coefficients determined by a decimated set of speech signals. The low complexity interpolator had no significant effect on the complexity. Thus the complexity of determining the integer pitch estimate in the super resolution PDA was reduced from 90,000 MAC operations to less than 10,000 MAC operations. The technique was found to exhibit an integer pitch period estimate equal to the original resolution for over 77% of the database it was subjected to. It also attained a sample deviation of ≤1 with respect to the original integer pitch estimate for over 92% of the database. The one sample deviation is acceptable as the pitch period is estimated with a resolution dependent on the sampling rate. The exact pitch period is not normally a multiple of the sampling rate.

3. The fast implementation of the LPC coefficients being transformed to Line Spectral Pairs and its reconversion process is only possible if the trigonometric functions (inefficiently evaluated by DSP processors) are eliminated.

4. Echo cancellation is essential in low bit rate speech coding algorithms once integrated into the PSTN. The inherent delays of the coding algorithms make the echo more noticeable and annoying to listeners as is distinguished from the normal sidetone of a telephone. The benefits associated with using bandwidth efficient technology can only be realised with the advent of low cost echo cancellers. These cancellers maintain the quality of service demanded for network telephony. The proposed solution of using a modified NLMS based echo canceller with the option of using the complexity reduction technique discussed in Section 5.7 facilitates the implementation of voice coding and echo cancellation on a single DSP processor.
BIBLIOGRAPHY

[1] CCITT Recommendations G.165, "Echo Cancellers".


[58] ITU-T Recommendations G.168, "Digital Network Echo Cancellers".


[60] ITU-T Recommendations G.723.1, "Dual rate speech coder for multimedia communications transmitting at 5.3 & 6.3 kbit/s".