Advanced Signal Processing Techniques for Pitch Synchronous Sinusoidal Speech Coders

Richard Edwards

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

University of Surrey
Centre for Communication Systems Research
School of Electronics and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

October 2007

© Richard Edwards 2007
Abstract

Recent trends in commercial and consumer demand have led to the increasing use of multimedia applications in mobile and Internet telephony. Although audio, video and data communications are becoming more prevalent, a major application is and will remain the transmission of speech. Speech coding techniques suited to these new trends must be developed, not only to provide high quality speech communication but also to minimise the required bandwidth for speech, so as to maximise that available for the new audio, video and data services.

The majority of current speech coders employed in mobile and Internet applications employ a Code Excited Linear Prediction (CELP) model. These coders attempt to reproduce the input speech signal and can produce high quality synthetic speech at bit rates above 8 kbps. Sinusoidal speech coders tend to dominate at rates below 6 kbps but due to limitations in the sinusoidal speech coding model, their synthetic speech quality cannot be significantly improved even if their bit rate is increased. Recent developments have seen the emergence and application of Pitch Synchronous (PS) speech coding techniques to these coders in order to remove the limitations of the sinusoidal speech coding model.

The aim of the research presented in this thesis is to investigate and eliminate the factors that limit the quality of the synthetic speech produced by PS sinusoidal coders. In order to achieve this innovative signal processing techniques have been developed. New parameter analysis and quantisation techniques have been produced which overcome many of the problems associated with applying PS techniques to sinusoidal coders. In sinusoidal based coders, two of the most important elements are the correct formulation of pitch and voicing values from the input speech. The techniques introduced here have greatly improved these calculations resulting in a higher quality PS sinusoidal speech coder than was previously available. A new quantisation method which is able to reduce the distortion from quantising speech spectral information has also been developed. When these new techniques are utilised they effectively raise the synthetic speech quality of sinusoidal coders to a level comparable to that produced by CELP based schemes, making PS sinusoidal coders a promising alternative at low to medium bit rates.

Key words: Pitch Synchronous, Speech Coding, Sinusoidal Speech Coding, Signal Processing

Email: r.edwards@eim.surrey.ac.uk
WWW: http://www.eps.surrey.ac.uk/
Acknowledgements

I would like to thank Professor Ahmet Kondoz for giving me the opportunity to pursue my doctorate in the DSP and Multimedia Group at the University of Surrey. I would also like to thank Doctor Stephane Villette for his help and guidance and fellow group members past and present who have helped me over the course of my studies.
3 Digital Speech Fundamentals

3.1 Introduction .................................. 17

3.2 Speech Analysis ................................ 17

3.3 LPC Analysis ................................. 20
   3.3.1 The Autocorrelation Method .................... 22
   3.3.2 The Covariance Method ....................... 23

3.4 Quantisation of LPC Coefficients .......... 24
   3.4.1 Line Spectral Frequencies .................... 24

3.5 Pitch Detection ............................. 27
   3.5.1 Time Domain ............................. 27
   3.5.2 Frequency Domain .......................... 28

3.6 Voicing Estimation ........................... 30
   3.6.1 Hard Decision Voicing ....................... 30
      3.6.1.1 Zero Crossing Ratio .................... 30
      3.6.1.2 Peakiness .......................... 31
      3.6.1.3 Energy to Peak Energy Ratio .......... 32
      3.6.1.4 Low Band to Full Band Energy Ratio .... 33
      3.6.1.5 Normalised Correlation ................. 33
      3.6.1.6 Pre-Emphasis Energy Ratio .......... 33
   3.6.2 Soft Decision Voicing ...................... 35

3.7 Quantisation ................................. 35
   3.7.1 Scalar Quantisation ........................ 35
   3.7.2 Vector Quantisation ........................ 36
      3.7.2.1 Split Vector Quantisation .......... 37
      3.7.2.2 Multi Stage Vector Quantisation ....... 39
   3.7.3 Codebook Training .......................... 41
      3.7.3.1 The LBG Algorithm .................... 42

3.8 Concluding Remarks ............................ 43
## 4 Sinusoidal Speech Coding

### 4.1 Introduction

### 4.2 Existing techniques

- **4.2.1 Channel Vocoder**
- **4.2.2 Cepstral Vocoder**
- **4.2.3 LPC Vocoder**
- **4.2.4 MELP Vocoder**
- **4.2.5 MBE Vocoder**
- **4.2.6 SB-LPC Vocoder**

### 4.3 Pitch Synchronous Speech Coding

- **4.3.1 Introduction**
- **4.3.2 Pitch Synchronous Multi Band Coder**
- **4.3.3 Waveform Interpolation Coding**
- **4.3.4 Pitch Synchronous MELP**

### 4.4 Overview of Pitch Synchronous SB-LPC

### 4.5 Detailed description of PS SB-LPC

- **4.5.1 Pitch Cycle Detection**
  - **4.5.1.1 Cycle Size Post Processing**
- **4.5.2 Pitch Cycle Based Analysis**
  - **4.5.2.1 LPC Analysis**
  - **4.5.2.2 Energy**
  - **4.5.2.3 Spectral Amplitudes**
  - **4.5.2.4 Voicing**
  - **4.5.2.5 Final Hard Decision Voicing Estimate**
  - **4.5.2.6 Soft Decision Voicing**
- **4.5.3 Quantisation**
  - **4.5.3.1 Pitch and Voicing Quantisation**
  - **4.5.3.2 Energy Quantisation**
  - **4.5.3.3 LSF Quantisation**
  - **4.5.3.4 Spectral Amplitude Quantisation**
- **4.5.4 Coder Evaluation**
- **4.6 Concluding Remarks**
## 5 Classifier Based Voicing

5.1 Introduction .................................. 74

5.2 Bit Stream Editor .................................. 74

5.3 Utilisation of Bit Stream Editor ....................... 76
  5.3.1 Pitch Cycle Position Editing .................... 78
  5.3.2 Voice Level Editing ............................ 82

5.4 Voicing Estimation .............................. 86
  5.4.1 Introduction .................................. 86

5.4.2 Classic Voicing Estimation Methods ................ 87
  5.4.2.1 MBE Mixed Voicing .................... 88
  5.4.2.2 MELP Mixed Voicing ................... 89
  5.4.2.3 SB-LPC Mixed Voicing .................. 90

5.4.3 PS SB-LPC Voicing Classifier .................... 93

5.5 Concluding Remarks ............................. 99

## 6 Pitch Cycle Detection

6.1 Introduction .................................. 100

6.2 Cycle Size Post Processing .......................... 100

6.3 Combined Sinusoidal and Waveform Coding ............... 102

6.4 Group Delay Method ............................. 104
  6.4.1 Algorithm Specifics .......................... 104

6.5 Closed Loop Pitch Cycle Detection ..................... 107
  6.5.1 Phase Removal in the PS SB-LPC ................ 109

6.5.2 Waveform Matching to Phase Removed Signal .......... 114
   6.5.2.1 Algorithm Specifics .................... 115
   6.5.2.2 Waveform Matching at Speech Onsets .......... 117
   6.5.2.3 Irregular Pitch Size Variation .............. 119
   6.5.2.4 Waveform Matching at Steady State Speech .... 120

6.5.3 Reducing Complexity of Full Search ............... 121

6.5.4 Reduced Search Solutions ...................... 124

6.6 Remaining Issues With Closed Loop Pitch Cycle Detection ...... 129
  6.6.1 Phase Removal ............................ 129

6.6.2 Waveform Matching ............................ 130

6.7 Concluding Remarks ............................. 132
### Contents

7 Spectral Amplitude Quantisation

7.1 Introduction .................................. 133
7.2 Background .................................. 133
7.3 Quantisation of Spectral Amplitudes ............. 135
  7.3.1 Algorithm Specifics ....................... 139
  7.3.2 MSVQ Experiments ......................... 141
  7.3.3 Interleaving of Spectral Amplitudes .......... 144
7.4 Bit Allocation of Coder ........................ 145
7.5 Concluding Remarks ............................. 146

8 Coder Quality Testing .......................... 147
8.1 Listening Tests ................................ 147
  8.1.1 Informal Listening Tests .................. 147
  8.1.2 Formal Listening Tests ..................... 148
8.2 Test Results Summary ........................... 150
8.3 Concluding Remarks ............................. 151

9 Conclusions .................................... 152
  9.1 Aims ...................................... 152
  9.2 Concluding Overview .......................... 153
  9.3 Future Work ................................. 153

A List of Publications ............................ 155
B List of Abbreviations ............................ 156

References ........................................ 1
Chapter 1

Introduction

1.1 Background

Human speech sounds must be efficiently transmitted and stored in order to be used effectively. Designers of mobile communication systems, wireless computer networks and multimedia systems are all searching for improved techniques for handling speech. Speech coding research has found applications in mobile and computer networks, automation, biomedical systems, consumer products and military applications and almost everywhere that digital communication is needed.

In order to transmit speech information digitally it must first be converted from its original analogue format. Once a digital representation has been achieved it can be compressed using digital signal processing techniques. The first implementation of a digital system began with Pulse Code Modulation (PCM) which offered the possibility of high-fidelity transmission and storage. Today many of the Public Service Telephone Networks (PSTN) are based on this method. PCM is a straightforward method for discrete time amplitude approximation of analogue waveforms. However it does not offer any mechanism for redundancy removal therefore new techniques were designed and implemented which exploited the natural redundancies in the speech signal.

The past few decades has seen substantial application and progress of speech coding to communication systems. Central to this progress has been the development of speech
coders capable of producing high quality speech at low to medium bit rates of 4 to 16 kbps. A number of these coders have been adopted in national and international telephone standards.

Many of the adopted standards used in mobile telephone networks operate in the range of 5 to 16 kbps. The great majority of these standards utilise Hybrid speech coders that use a Code Excited Linear Prediction (CELP) model. Hybrid coders aim to directly match the input speech waveform, this waveform matching is achieved by minimising the error in the speech domain by using closed loop techniques. However, the performance of these coders begins to degrade below 8 kbps as they fail to exploit the perceptual redundancy present in the speech signal.

Sinusoidal speech coders can produce good quality synthetic speech at bit rates of 6 kbps and below. Their model aims to produce perceptually intelligible speech without matching the speech waveform. They are able to operate at a lower bit rates than the hybrid type coders by not transmitting the speech information that is perceptually unimportant. These coders operate Time Synchronously (TS) as they extract the speech information to be coded at fixed points of the input speech signal. However, due mainly to the models lack of robustness of open loop speech parameter estimation and inadequate modeling of non-stationary speech segments, increasing their bit rate will not significantly improve their synthetic speech quality.

Recently developments have seen the emergence and application of Pitch Synchronous (PS) coding techniques to sinusoidal coders. These techniques aim to improve synthetic speech quality by decomposing the input speech signal into individual pitch cycles and estimating parameters for each cycle. The aim of the research presented in this thesis is to investigate and eliminate the factors that limit the quality of the synthetic speech produced by PS sinusoidal coders. Such coders would be a promising alternative to CELP based schemes and could be applied in many systems such as mobile and military telephony and voice Internet communications.
1.2 Outline of Thesis

The first part of this thesis presents the background to speech coding and sinusoidal speech coding in particular. The second part of this thesis presents the investigation and subsequent results of the research. The thesis is organised as follows:

Chapter 2 - A review of speech coding techniques is presented. The main types of speech coders are discussed and the criteria for speech coder design is introduced. The main applications of speech coders are also reviewed.

Chapter 3 - Presents a review of speech coding techniques, these techniques such as linear predictive coding, quantisation and pitch prediction are common to most speech coders.

Chapter 4 - Sinusoidal and pitch synchronous sinusoidal speech coding are presented, the PS Split Band Linear Predictive Coder (SB-LPC) which forms the basis of this thesis is discussed in detail.

Chapter 5 - The development of a novel voicing classifier technique for use in sinusoidal based speech coders is detailed, the application of a new speech coding analysis tool is also presented.

Chapter 6 - The development of open and closed loop pitch cycle detection and segmentation algorithms in a PS sinusoidal speech coder is presented in some detail.

Chapter 7 - Methods to improve the quantisation of spectral amplitude information in a PS sinusoidal speech coder are presented.

Chapter 8 - The PS processing techniques presented in this thesis are applied to the PS SB-LPC and the resulting synthetic speech is formally tested against other sinusoidal and CELP based speech coders.

Chapter 9 - A concluding review of the research presented in this thesis is given.

1.3 Original Achievements

- A speech analysis design tool has been developed that allows the user to determine the cause and effect of audio distortion in a sinusoidal speech coder. This tool has
1.3. *Original Achievements*

also been used to construct speech databases for use in novel classifier techniques.

- A novel voicing classification method has been developed. When compared to a established method used in a standard sinusoidal based speech coder clear improvements were found.

- A complete closed loop pitch detection and segmentation system has been established in a sinusoidal based speech coder. Methods to pitch synchronously produce an approximately zero phase signal from original speech have been developed, when utilised in the closed loop matching system, a good match of the pitch pattern in original speech has been established.

- A method to accurately quantise sets of pitch synchronous spectral amplitude information at a fixed rate has been developed.

- A pitch synchronous sinusoidal speech coder operating at 4.8 kbps has been developed. When formally compared to CELP based speech coders operating at similar and higher bit rates, good test results have been produced.
Chapter 2

Digital Speech Coding

2.1 Introduction

The past few years has seen huge growth in the field of communication techniques and applications. For the majority of these applications speech remains the most popular method of real time communication between humans. Most of these applications require that the speech signal is in digital format so that it may be processed, stored or transmitted. If uncompressed this digital speech signal has high bandwidth and storage requirements. Speech coding therefore is aimed at compressing this digital speech signal so that its transmission and storage requirements are significantly reduced.

This chapter gives an overview of digital speech coding. The criteria that forms the specifications of a speech coder are introduced and the many types of speech coders that have been designed and implemented are summarised.

2.2 Design Criteria

The application to which any speech coder is to be applied determines the majority of the speech coder design parameters. There are several design parameters to be considered, and improving one usually has the effect of causing degradation in another.
2.2. Design Criteria

2.2.1 Bit Rate and Quality

The two most important design parameters are the coder operational bit rate, and the decoded speech quality. When an appropriate algorithm needs to be selected, or a new speech coding algorithm is to be designed, one of these parameters must be specified. In conditions where the available bandwidth is limited, the bit rate of the coder is specified and the algorithm with the highest quality at that bit rate is selected. Each coding algorithm is suited to operation over a specific range of bit rates. There will exist a rate below which the obtainable quality from an algorithm will diminish significantly. An upper limit will also exist, above which little improvement in quality is observed despite the additional bit allocation.

Objective speech quality measures such as signal to noise ratio, do not account for the perceptual properties of the ear. Therefore subjective evaluations are required since the design of most low rate algorithms is based on perceptual criteria.

A subjective quality test that is used to assess coders at all rates is the Mean Opinion Score (MOS) [1]. The MOS test is a widely used procedure that has been standardised and used during the standardisation of many speech coders. In the MOS test subjects are required to score individual coded speech samples on a scale of one to five. The overall average score is used as the final score for a system. A MOS scale is depicted in Table 2.1.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Subjective Opinion</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Imperceptible</td>
<td>Transparent</td>
</tr>
<tr>
<td>4</td>
<td>Perceptible, but not annoying</td>
<td>Toll</td>
</tr>
<tr>
<td>3</td>
<td>Slightly annoying</td>
<td>Communication</td>
</tr>
<tr>
<td>2</td>
<td>Annoying</td>
<td>Synthetic</td>
</tr>
<tr>
<td>1</td>
<td>Very annoying</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2.1: The MOS speech quality scale

Toll quality speech is intelligible from the original with no distortion, Communication quality indicates some distortion present but the synthetic speech is still highly
2.2. Design Criteria

The three main delay components in a speech communication system are algorithmic, processing and the delay caused by the communications network. The algorithmic delay is caused as speech coders buffer in a frame of speech before encoding and they employ frame look ahead to remove redundancy in the speech thus improving compression. Processing delay is caused by the processing that must be carried out on each speech frame and may be reduced by using a lower complexity algorithm or faster processor. The communications delay of the network consists of signal propagation and multiplexing delay.

The end to end delay is an important factor for transmission applications. Communication delays of greater than 400 ms make full duplex conversation impossible. A delay of 250 ms is acceptable [2]. Delays of 50 ms introduce echo, requiring the need for echo cancellation either within the network or in the terminal equipment [3]; this impacts on communication system cost and complexity.

2.2.3 Complexity

The computational complexity and memory requirements of the coding algorithm determine the cost and power consumption of the Digital Signal Processor (DSP) used. Lower bit rates and higher quality can be achieved by increasing the complexity of an algorithm. However increasing coder complexity increases power consumption and cost, factors which are essential for mobile applications. For mass market applications fixed point processors are usually used due to their lower unit cost and lower power consumption though they can require greater programming effort. Although not comparable between CPU architectures a rough guide to processor speed is given in terms of Million Instructions Per Second (MIPS).
2.2. Design Criteria

2.2.4 Error Robustness

The received bit stream at the speech decoder can be corrupted by channel errors. Therefore the ability of the speech coder to be robust to varying channel conditions in a different communication environments is important. These environments may include PSTN, voice over Internet Protocol networks, mobile and satellite based communication. These channel errors can be classified as random and burst errors, burst errors are typical in mobile applications. Robustness against random errors can be guarded against by adding redundancy into the encoded information. Burst error detection schemes are used to classify transmitted frames as unusable and then remedial action can be taken to conceal the effects of the loss of a frame. Modern speech coders such as the Adaptive Multi Rate (AMR) coder [4] operate at different rates and different source-channel coding ratios according to channel conditions. This means that in poor channel conditions more bits can be allocated to preventing quality degradation due to channel errors.

Coders which use long term prediction are more sensitive to channel errors [5] as this causes errors in one speech frame to impact over several frames of speech. In order to improve the robustness of modern speech coders used in increasingly popular voice packet networks redundancy can be added in the packets at the encoder or the level of inter-frame dependencies at the encoder can be reduced [6]. Recently sinusoidal speech coders have been applied to voice packet networks [7], these coders do not utilise long term prediction as a consequence good packet loss results have been achieved when set against CELP based schemes.

2.2.5 Input Signal Requirements

Low bit rate speech coders are aimed at the compression of human speech, the ability to carry non-speech signals may be required. For example, the PSTN uses Dual Tone Multi Frequency (DTMF) system for signaling telephone digits and modem tones. The statistical properties of such voice-band data signals are quite different from those of speech, therefore the coder employed must be capable of processing both types.
2.3. Speech Coding Techniques

The input speech signal processed by the speech coder may be badly contaminated by background noise such as found in military situations or street noise in mobile applications. Therefore the speech analysis and parameter estimation process should be robust in the presence of background noise.

2.3 Speech Coding Techniques

In this section the three main classes of speech coder are summarised, they can be classed as Parametric, Waveform and Hybrid.

2.3.1 Parametric Coders

Parametric speech coders are based upon a set of model parameters, they are also known as vocoders (voice coders). In these coders, the extracted parameters are quantised and transmitted to the decoder where they are dequantised and used to produce synthetic speech. The speech production model does not attempt to match the original speech waveform to the synthetic speech and the model parameters are extracted by an open loop process. The speech quality does not converge to transparent speech quality with better quantisation due to the limitations of the speech model. As waveform similarity is not preserved the speech quality should be assessed subjectively.

2.3.1.1 Linear Prediction Coders

Linear Prediction (LP) coders model the vocal tract with a linear prediction filter. The excitation signal to the LP filter is provided by periodic pulses and random noise which represent glottal pulses and turbulent air flow respectively from the vocal cords, this speech production mechanism is introduced in Chapter 3. The main weakness of LP based coders is the binary voicing decision of the excitation which fails to model mixed type signals effectively, later LP based coders improved upon this process by making voicing decisions in the frequency domain.
2.3. Speech Coding Techniques

2.3.1.2 Sinusoidal Coders

Sinusoidal coders represent the speech as a sum of sinusoidal components. The model parameters such as the amplitudes, frequencies and phases are estimated at regular intervals from the speech spectrum. At low bit rates the phase information is not transmitted but is modelled at the decoder. Sinusoidal speech coders are discussed further in Chapter 4.

2.3.2 Waveform Coders

Waveform coders focus on representing the original speech waveform as accurately as possible. Modern waveform type coders (Hybrid coders) do this by minimising the error between the original and synthetic speech. Older waveform coders attempted to directly quantise the speech signal. International Telecommunications Union (ITU) G.711 [8] PCM employed logarithmic quantisation techniques to code speech at 64 kbps. Later standards such as G721 [9] at 32 kbps employed Adaptive Differential Pulse Coded Modulation (ADPCM) techniques to quantise the difference between the current sample and predicted value.

2.3.3 Hybrid Coders

Hybrid coders combine the advantages of parametric and waveform coding. They employ similar modeling techniques used in parametric coders (see Section 3.2) but they have a feedback loop that attempts to copy the input waveform (as in waveform coders) using a suitable error criterion. This is can be done by employing a closed loop Analysis by Synthesis scheme (AbS) to select the optimum sequence. The most widely used variant is the Code Excited Linear Prediction (CELP) [10] model which uses AbS to select the optimum sequence from one or more stored codebooks.

CELP coders usually offer better performance at the cost of greater computational complexity. Improvements in signal processing technology and the general increase in computational power have made complexity less important than before, and CELP coders now dominate in the field of mobile telephony and in related application areas.
2.4. Standardisation

The coder of choice in European 2nd and 3rd Generation (2G/3G) networks, the AMR coder and a popular coder for Voice over Internet Protocol (VoIP) applications, G729 [11] are both based on the CELP model.

2.3.3.1 Multi Mode Coders

Multi mode coders are classified here as hybrid coders. These coders can switch coding techniques according to the characteristics of the speech signal. Parametric based coders typically estimate the speech signal at regular intervals and interpolate between these points. This can produce high quality speech at periodic speech sections but at transitional speech sections where speech may not have strong periodic characteristics parametric based coders cannot reach the quality of waveform type coders that attempt to match the target waveform and can represent irregular features of speech well. These coders therefore combine parametric based coders at periodic speech sections with a waveform type coder that are more suited at transitional speech regions. These multi mode coders [12] and [13] need to use complex algorithms to ensure artifacts are not produced when switching between coder type.

A diagram showing the quality against bit rate of the various types speech coder is given in Figure 2.1.

2.4 Standardisation

Several international boards exist that have standardised speech coders. One such board is the International Telecommunication Union (ITU) that has released many popular standards commonly used in PSTN. Since the development of mobile phone technology many standardisation bodies have been set up. This standardisation process has been the main driving force behind speech coding research. One of the major standardisation bodies in Europe is the European Telecommunications Standards Institute (ETSI) which publishes standard coders for the European mobile network. There are similar standardisation bodies in North America and Japan. Table 2.2 shows the most
important speech coders developed in the past 30 years, the great majority of these in the mobile environment are based on the CELP model.

### 2.5 Speech Coder Applications

The major application of digital speech coding is telecommunication systems. This usage can be divided into terrestrial, satellite and Internet based communications.

#### 2.5.1 Terrestrial Based Communication

These include PSTN, Integrated Services Digital Networks (ISDN) and mobile radio systems. The initial PSTN international standard used 64 kbps compounded PCM. Rising consumer demand led to the adoption of a 32 kbps ADPCM speech coder the G.721 [9] as a standard for PSTN. Advances in speech coding led to the development G.728 at 16 kbps [15] and G.729 [11] at 8 kbps. Both produce near toll quality speech but are very complex compared to original PCM coders and have longer delays. They are used on networks where there is a high subscriber demand.

Integrated Services Digital Network (ISDN) is a circuit-switched telephone network
### Table 2.2: A table of primary speech coding standards.

Bit rate is in kbps with combined channel and source coding rate in brackets. The delay figures are in milliseconds [14] and [2].

<table>
<thead>
<tr>
<th>Standard</th>
<th>Year</th>
<th>Algorithm</th>
<th>Application</th>
<th>Bit Rate</th>
<th>MOS</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>1972</td>
<td>PCM</td>
<td>PSTN</td>
<td>64</td>
<td>4.3</td>
<td>0.125</td>
</tr>
<tr>
<td>G.721</td>
<td>1984</td>
<td>ADPCM</td>
<td>PSTN</td>
<td>32</td>
<td>toll</td>
<td>-</td>
</tr>
<tr>
<td>G.722</td>
<td>1984</td>
<td>SB-ADPCM</td>
<td>ISDN</td>
<td>64/56/48</td>
<td>toll</td>
<td>-</td>
</tr>
<tr>
<td>FS1015</td>
<td>1984</td>
<td>LPC-10</td>
<td>Military</td>
<td>2.4</td>
<td>synth</td>
<td>112.5</td>
</tr>
<tr>
<td>GSM FR</td>
<td>1989</td>
<td>RPE-LTP</td>
<td>Mobile</td>
<td>13(22.8)</td>
<td>3.7</td>
<td>20</td>
</tr>
<tr>
<td>IS54</td>
<td>1989</td>
<td>VSELP</td>
<td>Mobile</td>
<td>7.95</td>
<td>3.6</td>
<td>20</td>
</tr>
<tr>
<td>Inmarsat-M</td>
<td>1990</td>
<td>IMBE</td>
<td>Satellite</td>
<td>4.15</td>
<td>3.4</td>
<td>78.75</td>
</tr>
<tr>
<td>G726</td>
<td>1991</td>
<td>VBR-ADPCM</td>
<td>PSTN</td>
<td>16/24/32/40</td>
<td>toll</td>
<td>0.125</td>
</tr>
<tr>
<td>FR PDC</td>
<td>1991</td>
<td>VSELP</td>
<td>Mobile</td>
<td>6.17 (11.2)</td>
<td>comm.</td>
<td>20</td>
</tr>
<tr>
<td>FS1016</td>
<td>1991</td>
<td>CELP</td>
<td>Military</td>
<td>4.8</td>
<td>3</td>
<td>37.5</td>
</tr>
<tr>
<td>HR JDC</td>
<td>1993</td>
<td>PSI-CELP</td>
<td>Mobile</td>
<td>3.45 (5.6)</td>
<td>comm.</td>
<td>40</td>
</tr>
<tr>
<td>G.728</td>
<td>1994</td>
<td>LD-CELP</td>
<td>PSTN</td>
<td>16</td>
<td>4</td>
<td>4.625</td>
</tr>
<tr>
<td>GSM HR</td>
<td>1994</td>
<td>VSELP</td>
<td>Mobile</td>
<td>5.7 (11.4)</td>
<td>3.5</td>
<td>24.375</td>
</tr>
<tr>
<td>G.729</td>
<td>1995</td>
<td>CS-CELP</td>
<td>PSTN</td>
<td>8</td>
<td>4</td>
<td>15.5</td>
</tr>
<tr>
<td>G.723.1</td>
<td>1995</td>
<td>A/MP-MLQ CELP</td>
<td>IP based</td>
<td>5.3/6.3</td>
<td>toll</td>
<td>37.5</td>
</tr>
<tr>
<td>GSM EFR</td>
<td>1995</td>
<td>ACELP</td>
<td>Mobile</td>
<td>12.2 (22.8)</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>FS 2.4</td>
<td>1997</td>
<td>MELP</td>
<td>Military</td>
<td>2.4</td>
<td>3</td>
<td>45.5</td>
</tr>
<tr>
<td>GSM AMR</td>
<td>1998</td>
<td>Multi Rate ACELP</td>
<td>Mobile</td>
<td>4.75 to 12.2</td>
<td>-</td>
<td>40</td>
</tr>
<tr>
<td>SMV</td>
<td>2001</td>
<td>Multi Rate eX-CELP</td>
<td>Mobile</td>
<td>0.8 - 8.5</td>
<td>comm.</td>
<td>-</td>
</tr>
<tr>
<td>ILBC</td>
<td>2002</td>
<td>ACELP</td>
<td>IP based</td>
<td>13.967</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>AMR-WB</td>
<td>2002</td>
<td>Multi Rate ACELP</td>
<td>WB Mobile</td>
<td>6.6 to 23.85</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
system, designed to allow digital transmission of voice and data over ordinary telephone copper wires, resulting in better quality and higher speeds than that is available with the PSTN system. A 16 kHz sampling frequency speech coder the 64 kbps ITU G.722 Sub Band ADPCM is used for these applications [16]. Recently higher bandwidth technologies such as Asynchronous Digital Subscriber Line (ADSL) have reduced the relevance of ISDN.

Due to the high number of subscribers and limited radio bandwidth, the compression of speech is very important in mobile communications and the majority of recent speech coding research has been applied in this area. The Global System for Mobile communication (GSM) is a European mobile system that covers most of the continent. The first GSM standard phone was the 13 kbps GSM Full Rate (GSM FR) [17], followed by the 12.2 kbps GSM Enhanced Full Rate (GSM EFR) [18] and the 5.6 kbps GSM Half Rate (GSM HR) [19]. Due to the channel errors that can occur in mobile networks the GSM FR and GSM EFR coders have gross rates of 22.8 kbps, the GSM HR has 11.4 kbps with channel coding.

The AMR (Adaptive Multi-Rate) standard is a speech coding algorithm operating at variable bit rates in the range of 4.75 to 12.2 kbps. This technology was initially developed for GSM systems, the single most deployed 2G mobile telecommunication standard worldwide. This AMR narrow band codec was standardised by the ETSI and adopted by the 3rd Generation Partnership Project (3GPP) as the mandatory coder for 2.5G and 3G wireless systems based on the evolved GSM core network. A wide band implementation of this coder AMR-WB is the required coder in GSM and 3G networks for wide band speech and for multimedia services when wide band speech (with 16 kHz sampling frequency) is supported. The AMR-WB coder operates on nine speech coding rates between 6.6 and 23.85 kbps [20].

2.5.2 Satellite Based Communication

A satellite phone is a mobile phone that communicates directly with orbiting communications satellites. Depending on the architecture of a particular system, coverage may include the entire Earth, or only specific regions. Satellite communication systems
2.6. Concluding Remarks

use low bit rate speech coders as there is a limited bandwidth, satellite phone systems frequently suffer from high latency due to the distances involved and bursty channel errors due to fading from multi path effects and shadowing. Satellite systems such as Iridium use the 2.4 kbps Advanced Multi Band Excitation coder (AMBE) [21]. The Immarsat-M Improved Multi Band Excitation (IMBE) [22] is another coder used with a 4.15 kbps bit rate with channel coding this is increased to a rate of 6.4 kbps.

2.5.3 Internet Based Communication

One of the greatest growth developments for speech coding in recent years has been the increase in voice communication over the Internet known as Voice over Internet Protocol (VoIP). This initially was applied to PC based communication but recently this technology has been applied to mobile phones and is known as VoIP Mobile. VoIP allows voice communication between Internet users or an Internet user and a user on PSTN. Two popular coders used in VoIP applications are a 5.3/6.3 kbps coder called G.723.1 [23] and the 8 kbps coder G729, both are used in VoIP applications due to their relatively low bandwidth requirement.

The CELP based G729 in particular has become standard in VoIP applications due its toll quality speech performance at 8 kbps. However CELP based coders are sensitive to packet losses because of inter-frame dependencies in their predictor states [5]. A more recent CELP based coder is the Internet Low Bit Rate Coder (iLBC) [24]. This coder treats each packet individually removing memory and packet loss error propagation that is present in many standard CELP coders but at the increased bit rate of 13.697 kbps compared to G729 at 8 kbps.

2.6 Concluding Remarks

This chapter has given an overview of digital speech coding and its applications. Existing speech coders can be classed as parametric, waveform or hybrid based. The work presented here is based on a parametric coder employing a sinusoidal model.
The criteria that are involved in design of a speech coder has been presented. The relevant speech coding technologies that are currently employed in commercial use have been reviewed.

The next chapter will introduce the basic ideas that lie behind many of these speech coding technologies and standards.
Chapter 3

Digital Speech Fundamentals

3.1 Introduction

Most of the techniques used by low to medium bit rate speech coders to compress and code speech are based on the source filter model, which is a mathematical representation of the human speech production mechanism. The parameters required for this model are found by analysis, quantised and then transmitted for synthesis. These parametric modelling and quantisation techniques are employed in many of today's speech coding technologies, this chapter presents an overview of these fundamental techniques.

3.2 Speech Analysis

Figure 3.1 demonstrates the human speech production mechanism. The speech production mechanism is considered to be the result of excitation and modulation. The lungs and vocal cords produce the excitation. The vocal tract modulates the excitation by changing its shape.

Speech can be considered as quasi-stationary over short segments, typically 5-20 ms. It can be classified as voiced, unvoiced or mixed. The periodic excitation which is characterised by the fundamental or pitch frequency is used for voiced sounds and represents the airflow through the vocal cords as they vibrate. These air flow pulses
3.2. Speech Analysis

Figure 3.1: Human speech production mechanism [25]

Figure 3.2: A section of voiced speech and its frequency spectrum
then travel along the vocal tract and are shaped by the tongue, jaws, lips and teeth. Unvoiced sounds represent the noise created by forcing air past constrictions through the vocal tract. Mixed voiced sounds such as fricatives are produced by constrictions in the tongue, lips and teeth and may be accompanied by voiced excitation generated by the vocal cords. Plosives are generated by completely closing a part of the vocal tract and then by releasing the accumulated pressure.

Figures 3.2 and 3.3 show examples of voiced and unvoiced speech respectively. Voiced speech as shown is periodic in the time domain and has a harmonic structure in the frequency domain up to around 4 kHz whereas unvoiced speech lacks a clear structure. The peaks that are visible in the frequency domain plot of voiced speech are known as formants, they represent the resonant modes of the vocal tract and are very important perceptually. This formant structure is shown as the dotted line in Figure 3.2.

A simple speech production model is known as the Source Filter Model. The filter models the behaviour of the vocal tract, the filter parameters as such are modeled as Linear Predictor Coefficients. The model assumes that speech is produced by exciting the filter by random noise for unvoiced speech or an impulse train for voiced. The source filter model is based on a number of assumptions and simplifications. The main assumption is that there is no interaction between the vocal tract shape and larynx.
3.3 LPC Analysis

The second assumption is that the vocal tract has a linear transfer function. Although these assumptions are not strictly true the removal of redundancy has meant that the source filter model has been widely used in simple vocoders in the past.

![Source Filter Model Diagram](image)

Figure 3.4: Source Filter Model

3.3 LPC Analysis

Linear Predictive Coding is used to derive the coefficients of the vocal tract filter of the source filter model. The heart of the LPC is the linear predictor, the time varying filter which represents the combined effects of vocal tract, glottal flow and radiation of the lips. It can be represented by

$$ H(z) = \frac{S(z)}{X(z)} = \frac{G \left( 1 - \sum_{j=1}^{P} \beta_j z^{-j} \right)}{1 - \sum_{j=1}^{P} \alpha_j z^{-j}} \quad (3.1) $$

(3.1) contains both poles and zeros, however if the order of the denominator is high enough [26] an all pole approximation can be used. This assumption produces a synthetic spectral envelope that contains peaks that correspond to the formants in human speech. A conjugate pair of poles is required to model a single formant and as speech is assumed to contain one formant per kHz, a signal of bandwidth 4 kHz will assumed to have four formants. Therefore it is common to use a 10th order linear prediction filter and use the remaining poles to model other spectral characteristics. The all pole
3.3. LPC Analysis

The transfer function can be thus written as

\[ H(z) = \frac{G}{1 - \sum_{j=1}^{p} \alpha_j z^{-j}} = \frac{G}{A(z)} \]  

(3.2)

The z-domain transfer function of (3.2) can be written in the form of a difference equation in the time domain

\[ s(n) = Gx(n) + \sum_{j=1}^{p} \alpha_j s(n - j) \]  

(3.3)

(3.3) states that the present speech output \( s(n) \) can be obtained by summing the weighted present input, \( Gx(n) \) and a weighted sum of the most recent past \( p \) output samples. The gain term is usually set to unity; so the next step is to determine the coefficients of the predictor, i.e. \( \alpha_j \) for \( j = 1, 2, ..., p \) where \( p \) is the order of the filter assumed here to be \( 10^{th} \) order.

If the speech signal \( s(n) \) is filtered by the inverse of the predictor filter the output \( e(n) \) is called the residual signal

\[ e(n) = s(n) - \sum_{j=1}^{p} \alpha_j s(n - j) \]  

(3.4)

This residual signal is commonly used in speech coders to determine the pitch period. The objective is to find the predictor coefficients which minimise the residual energy \( E \)

\[ E\{e^2(n)\} = E\left\{ [s(n) - \sum_{j=1}^{p} \alpha_j s(n - j)]^2 \right\} \]  

(3.5)

Setting the partial derivatives of \( E \) with respect to \( \alpha_j \) to zero for \( j = 1, ..., p \) it can be shown that (3.5) reduces to

\[ \sum_{j=1}^{p} \alpha_j \phi_n(i, j) = \phi_n(i, 0), \quad \text{for } i = 1, 2, ..., p \]  

(3.6)

where

\[ \phi_n(i, j) = E\{s(n - i)s(n - j)\} \]  

(3.7)

A major assumption in the derivation of (3.6) is that the signal is stationary. This is not true for speech over long segments but for shorter segments is realistic. Consequently
our expectations in (3.7) are replaced by finite summations over a short length of speech samples. This is achieved by replacing the expectations of (3.6) by summations over finite limits, i.e.

\[
\phi_n(i, j) = E\{s(n - i)s(n - j)\} \quad (3.8)
\]

\[
= \sum_m s_n(m - i)s_n(m - j) \quad \text{for } i = 1, 2, \ldots, p, \quad j = 0, \ldots, p \quad (3.9)
\]

The two most popular methods that can be used to obtain the predictor coefficients are the autocorrelation and covariance methods.

### 3.3.1 The Autocorrelation Method

For the Autocorrelation Method (AM) the speech segment \( s_n(m) \) is assumed to be zero outside the interval \( 0 \leq n \leq N - 1 \) where \( N \) is the length of the sample sequence. Since for \( N \leq m \leq N + p \) the aim is to predict zero sample values (which are not zero) the prediction error for these samples will not be zero. Similarly, the beginning of the current frame will be affected by the same inaccuracy from the previous frame. The limits for (3.9) can be shown to be:

\[
\phi_n(i, j) = \sum_{m=0}^{N-1-|i-j|} s_n(m)s_n(m + |i - j|), \quad 1 \leq i \leq p, \quad 0 \leq j \leq p \quad (3.10)
\]

(3.10) can be reduced to the short time auto-correlation function as given by,

\[
\phi_n(i, j) = R_n(|i - j|), \quad \text{for } i = 1, 2, \ldots, p, \quad j = 0, \ldots, p \quad (3.11)
\]

where

\[
R_n(j) = \sum_{m=0}^{N-1-j} s_n(m)s_n(m + j) \quad (3.12)
\]

Using the AM (3.6) can be shown to be

\[
\sum_{j=1}^{p} \alpha_j R_n(|i - j|) = R_n(i), \quad 1 \leq i \leq p \quad (3.13)
\]
3.3. LPC Analysis

or in matrix form as

\[
\begin{pmatrix}
R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\
R_n(1) & R_n(0) & R_n(1) & \cdots & R_n(p-2) \\
R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
R_n(p-1) & R_n(p-2) & R_n(p-3) & \cdots & R_n(0)
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\vdots \\
\alpha_p
\end{pmatrix}
= \begin{pmatrix}
R_n(1) \\
R_n(2) \\
R_n(3) \\
\vdots \\
R_n(p)
\end{pmatrix}
\tag{3.14}
\]

Straight forward application of Gaussian elimination to solve such matrices is rather inefficient, with complexity \(O(n^3)\). Since all the elements on each diagonal of this symmetrical matrix are equal, i.e. it is a Toeplitz matrix, some efficient algorithms exist to solve the problem. The most widely used technique is a recursive algorithm known as the Levinson-Durbin [27] which has a complexity of \(O(n^2)\).

3.3.2 The Covariance Method

The covariance method (CM) differs from the autocorrelation method as it does not make any assumptions about the signal outside the range used in the calculation. The interval over which the mean square error is calculated is fixed.

\[
E = \sum_{m=0}^{N-1} e_n^2(m)
\tag{3.15}
\]

(3.9) can be written as

\[
\phi_n(i,j) = \sum_{m=0}^{N-1} s_n(m-i)s_n(m-j), \quad 1 \leq i \leq p, \quad 0 \leq j \leq p
\tag{3.16}
\]

Changing the summation index

\[
\phi_n(i,j) = \sum_{m=-i}^{N-i-1} s_n(m)s_n(m+i-j), \quad 1 \leq i \leq p, \quad 0 \leq j \leq p
\tag{3.17}
\]

(3.17) requires the use of samples in the interval \(-p \leq m \leq N - 1\). (3.16) is not a true autocorrelation function but a cross correlation between two very similar but not identical, finite length sample sequences. From (3.16) it can be shown that the original LPC equation (3.9) can be expressed as

\[
\sum_{j=1}^{p} \alpha_j \phi_n(i,j) = \phi_n(i,0), \quad 1 \leq i \leq p, \quad 0 \leq j \leq p
\tag{3.18}
\]

\(^1\)where \(O(.)\) is a measure of complexity in regard to the problem.
3.4 Quantisation of LPC Coefficients

in matrix form (3.18) becomes

\[
\begin{pmatrix}
\phi(1,1) & \phi(1,2) & \phi(1,3) & \ldots & \phi(1,p) \\
\phi(2,1) & \phi(2,2) & \phi(2,3) & \ldots & \phi(2,p) \\
\phi(3,1) & \phi(3,2) & \phi(3,3) & \ldots & \phi(3,p) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\phi(p,1) & \phi(p,2) & \phi(p,3) & \ldots & \phi(p,p)
\end{pmatrix}
\begin{pmatrix}
\alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\vdots \\
\alpha_p
\end{pmatrix}
= 
\begin{pmatrix}
\phi_n(1,0) \\
\phi_n(2,0) \\
\phi_n(3,0) \\
\vdots \\
\phi_n(p,0)
\end{pmatrix}
\]  

(3.19)

No assumptions are made therefore about the signal outside the calculation interval. The matrix here is not Toeplitz in structure, therefore the Levinson-Durbin algorithm cannot be used. A inversion method known as Cholesky Decomposition can be used [28]. Both the autocorrelation and covariance methods have been used in the research presented in this thesis.

3.4 Quantisation of LPC Coefficients

The spectral envelope defined by the LPC coefficients is sensitive to slight changes in coefficient values which may be introduced by the quantisation process. Quantisation may also affect the coefficients to such an extent that the stability of the filter may degrade. Therefore it is usual to transform the LPC filter coefficients into an alternative representation where small changes during quantisation will not have a dramatic effect on the LPC filter. There are several such representations available, the most popular of which being that of Line Spectral Frequencies (LSFs) [14].

3.4.1 Line Spectral Frequencies

LSFs are directly computed from Line Spectral Pair (LSP) coefficients. LSFs measured in Hz are related to LSPs by the relation

\[
LSFi = \frac{f_s}{2\pi} \cos^{-1}(LSPi)
\]

(3.20)
where $f_s$ is the sampling frequency. The LSPs can be defined as follows: the all pole $p^{th}$ filter $H(z)$ can be derived from the linear prediction analysis given by:

\[
H(z) = \frac{1}{1 + \sum_{j=1}^{p} \alpha_j z^{-j}} = \frac{1}{A(z)}
\]  \hspace{1cm} (3.21)

For an even value of $p$ and a stable filter, the polynomial $A(z)$ can be decomposed as even and odd parts $P(z)$ and $Q(z)$. $A(z)$ has complex roots anywhere within the unit circle but $P(z)$ and $Q(z)$ have the very useful property of only having roots on the unit circle. Since they exist as palindromic polynomials it is possible to reduce the degree of the polynomial by two.

\[
P(z) = A(z) + z^{-(p+1)}A(z^{-1})
\]  \hspace{1cm} (3.22)

and

\[
Q(z) = A(z) - z^{-(p+1)}A(z^{-1})
\]  \hspace{1cm} (3.23)

wherein

\[
A(z) = \frac{P(z) + Q(z)}{2}
\]  \hspace{1cm} (3.24)

$P(z)$ and $Q(z)$ can be written as

\[
P(z) = A(z) + z^{-(p+1)}A(z^{-1})
= \left(1 - \sum_{j=1}^{p} a_j z^{-j}\right) + z^{-(p+1)}\left(1 - \sum_{j=1}^{p} a_j (z^{-1})^{-j}\right)
= 1 + (a_p - a_1)z^{-1} + \ldots + (a_1 - a_p)z^{-p} + z^{-(p+1)}
\]  \hspace{1cm} (3.25)

\[
P(z) = z^{-(p+1)} \prod_{j=0}^{p+1} (z - \alpha_j)
\]  \hspace{1cm} (3.26)

\[
Q(z) = z^{-(p+1)} \prod_{j=0}^{p+1} (z - \beta_j)
\]  \hspace{1cm} (3.27)

The roots of $P(z)$ and $Q(z)$ lie on the unit circle and occur in complex conjugate pairs with the exception of the roots at $z = -1$ in the case of $P(z)$ and $z = 1$ for $Q(z)$. Therefore there are $p$ unknowns to find which are the arguments of $\alpha_j$ and $\beta_j$, the roots of $P(z)$ and $Q(z)$. Since the roots lie on the unit circle, the all pole models $P^{-1}(z)$
3.4. Quantisation of LPC Coefficients

and $Q^{-1}(z)$ will have infinite values at these locations. In terms of the spectrum, these roots can be seen as vertical lines at frequencies corresponding to the angle of each root; the Line Spectral Frequencies, see Figure 3.5.

Various methods exist for solving the roots including the real root method, ratio filter method and Chebyshev series method [29]; this project used the real root method to solve the polynomials as described further in [30].

![Figure 3.5: LSFs and LPC frequency response for voiced (left) and unvoiced speech (right)](image)

LPC filter stability is guaranteed provided that the LSFs are monotonically increasing and are bound between the limits 0 Hz and $f_s/2$ where $f_s$ is the sampling frequency, i.e. $0 < f_1 < f_2 < \ldots < f_p < f_s/2$. As shown in Figure 3.5 closely grouped LSFs indicate formant presence also the closer formants are to each other the smaller the formant bandwidth. Any error that occurs in a particular LSF is localised to that LSF region and will not affect the spectrum as whole.
3.5 Pitch Detection

Low bit rate sinusoidal speech coders rely heavily on extracting the correct speech parameters from a given speech signal. The correct estimation of the signal's pitch is vitally important, pitch conveys information such as speaker identification, emotional state and intonation. Other parameters such as spectral amplitudes also rely on accurate pitch estimation. Therefore the reliability of the Pitch Detection Algorithm (PDA) has a dramatic effect on the quality of the synthesised speech [2].

The pitch period is defined to be the time between glottal pulses generated by the opening and closing of the glottis (vocal cords). During steady voiced speech this value usually varies little but it can become irregular at some speech sections such as transitions. During unvoiced speech sections, the pitch period is irrelevant and can be discarded. Analysis to find the pitch period or pitch frequency (separation between harmonic frequencies) is usually achieved in PDAs by windowing the signal and assuming the pitch value to be stationary under the analysis window. Pitch determination can be carried out in time or frequency domain. A review of some pitch detection algorithms will now be described.

3.5.1 Time Domain

A PDA which employs time domain methods uses periodic similarity to identify the pitch. This is achieved by comparison of the speech signal with a delayed version of itself, the delay with the highest similarity is identified as the pitch value. Two methods to achieve this are the Autocorrelation method [2] and the Average Magnitude Difference Function [27]. The autocorrelation method is a measure of similarity, the correct value of delay will correspond with a maximum in the autocorrelation signal. The autocorrelation function can be defined as

\[ R(\tau) = \sum_{n=0}^{N-1} s(n)s(n - \tau) \]  

where \( N \) is the analysis frame length and \( \tau \) is the delay. Energy normalisation is often employed in this method as errors can occur during rapid energy fluctuations that can
occur within the analysis frame.

The AMDF was a popular PDA, it measured signal disagreement and so gave the minimum value in the AMDF function corresponding to the pitch value.

\[
A(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} |s(n) - s(n - \tau)|
\]

where \( \tau \) is the delay. AMDF was popular before current DSP technology became widespread because of its simpler computational structure.

### 3.5.2 Frequency Domain

Frequency domain pitch determination algorithms extract the fundamental frequency from the harmonics of the speech spectrum. Sinusoidal Speech Model Matching PDA (SSMM-PDA) is a technique that was developed by [31] and modified by [32]. This method assumes the speech to be composed of a sum of sinusoidal components with no assumption on frequency or phase, given by:

\[
s(n) = \sum_{l=1}^{L} A_l e^{j(\omega_l n + \phi_l)}
\]

where \( A_l, \omega_l \) and \( \phi_l \) are the spectral amplitude, frequency and phase of the \( l^{th} \) sinusoidal component respectively. Then a synthetic signal \( \hat{s}(n, \omega_0) \), is generated, composed of entirely harmonically related sinusoids, as follows:

\[
\hat{s}(n, \omega_0) = \sum_{l=1}^{K(\omega_0)} \hat{A}_l e^{j(n\omega_0 + \phi_l)}
\]

where \( \hat{A}_l \) represents the spectral amplitude of the synthetic spectrum for the \( l^{th} \) harmonic. The fundamental frequency of the pitch is then obtained by finding the value of \( \omega_0 \) which minimises the MSE between the spectra of \( s(n) \) and \( (\hat{s}, \omega_0) \).

\[
\epsilon(\omega_0) = \frac{1}{N} \sum_{n=0}^{N-1} |s(n) - \hat{s}(n, \omega_0)|^2
\]
3.5. Pitch Detection

Direct evaluation of (3.34) for all the possible $\omega_0$ and choosing the $\omega_0$ corresponding to the minimum $\epsilon(\omega_0)$ is a computationally intensive process. [31] simplified the search procedure based on a number of assumptions [32]:

1. The spectra of $s(n)$ and $\hat{s}(n, \omega_0)$ are well resolved and can be approximated by sinc functions at each component frequency and scaled by $A_l$ for $s(n)$ and located at each harmonic of the candidate fundamental frequency $\omega_0$, scaled by the spectral envelope at the harmonic frequencies for $\hat{s}(n, \omega_0)$.

2. The matching of one sinusoidal component from $s(n)$ to another from $\hat{s}(n, \omega_0)$ is given by the product of their harmonic amplitudes, i.e. $A_l$ and $\tilde{A}_l$ weighted with a distance function $D(\omega_l - k\omega_0)$. The value of which may be pre computed and stored in a look up table.

3. Each sinusoidal component of $s(n)$ is represented by only one of its counterparts from $\hat{s}(n, \omega_0)$ the one which has the greatest matching, usually the closest.

4. Minimising the error $\sum_{k=0}^{N} [x(k) - y(k, a, b, c, ..)]^2$ over the variables $a, b, c, ..$ is equivalent to maximising the function $\sum_{k=0}^{N} y(k, a, b, c, ..)[x(k) - \frac{1}{2} y(k, a, b, c, ..)]$.

After simplifications, the fundamental frequency can be determined by maximising $\rho(\omega_0)$ with respect to $\omega_0$ where $\rho(\omega_0)$ is given by:

$$\rho(\omega_0) = \sum_{k=1}^{K(\omega_0)} \tilde{A}(k\omega_0) \left( \max_l [A_l D(\omega_l - k\omega_0)] - \frac{1}{2} \tilde{A}(k\omega_0) \right)$$

(3.35)

where $K(\omega_0) = [\pi/\omega_0]$ is the number of harmonics for the given pitch value. $\omega_l$ and $A_l$ are the frequency and amplitude of the $l^{th}$ peak in the original spectrum. $\omega_0 = \frac{2\pi}{P}$ is the fundamental frequency corresponding to the pitch candidate $P$. $\tilde{A}(k\omega_0)$ is the amplitude of the $k^{th}$ harmonic in the synthetic spectrum. $D(\omega_l - k\omega_0)$ is a distance measure defined as

$$D(\omega_l - k\omega_0) = \begin{cases} \frac{\sin(x)}{x} & \text{with} \quad x = 2\pi \frac{\omega_l - k\omega_0}{\omega_0} \quad x \leq \pi \\ 0 & \text{with} \quad x > \pi \end{cases}$$

(3.36)
3.6. **Voicing Estimation**

The amplitudes $\tilde{A}(k\omega_0)$ of the $k^{th}$ harmonic in the synthetic spectrum are approximated by the spectral envelope of the original spectrum at the corresponding frequency. This envelope can be approximated by interpolating between the peaks in the spectrum. This pitch detection process forms the basis of the pitch detection method used in [33] and [30].

3.6 **Voicing Estimation**

Most sinusoidal coders classify input speech according to the source of the excitation. Voiced speech is produced by the vocal cords which produces a periodic signal, unvoiced by turbulent air flow resulting in a random like signal. The correct voicing classification and the degree to which speech is voiced is crucial for producing quality synthesised speech. Voiced speech classified as unvoiced produces rough and intelligible speech, unvoiced declared as voiced produces speech with a metallic characteristic. Modern sinusoidal coders use a mixed excitation mode which allows speech to consist of both types. They use a hard decision to classify input speech as voiced or unvoiced; a soft decision is then made to further classify the degree of voicing present in speech classified as voiced to improve the synthetic speech quality.

3.6.1 **Hard Decision Voicing**

Some of the main methods to discriminate between voiced and unvoiced speech are described.

3.6.1.1 **Zero Crossing Ratio**

The number of times the speech signal changes sign from one sample to the next is a good indication of voicing classification [2]. The random nature of unvoiced speech means its Zero Crossing ratio ($Z_c$) will be higher than for voiced speech which will change sign less from sample to sample over the length of the signal.
3.6. Voicing Estimation

3.6.1.2 Peakiness

The energy of voiced speech is usually concentrated around the main pulse in the pitch cycle, the energy of unvoiced speech tends to be spread out over the pitch cycle as there are no pitch pulses. This effect is more pronounced in the LP residual signal where voiced speech tends to be peaky whereas unvoiced speech is not. If the signal is voiced i.e. contains large peaks the Peakiness ($P_k$) ratio given as (3.37) will be high, typically higher than 1.5 for voiced speech

$$P_k = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} r^2(n)}$$

$\frac{1}{N} \sum_{n=0}^{N-1} |r(n)|$

where $r(n)$ is the LP residual and $N$ is the length of the signal. Figure 3.6 shows a plot of Peakiness ($P_k$) and Zero Crossing Ratio ($Z_c$) with suggested voiced/unvoiced

![Figure 3.6: A section of speech and the Zero Crossing ratio (middle) and Peakiness (top). Suggested threshold values are shown (dashed)](image)
threshold.

![Graph showing voicing estimation thresholds](image)

**Figure 3.7:** A section of speech and the Energy to Peak Energy Ratio $E/E_p$ and Low Frequency Band to Full Frequency Band Ratio $LF/FF$. Suggested threshold values are shown (dashed)

### 3.6.1.3 Energy to Peak Energy Ratio

Voiced speech is usually higher in energy than unvoiced. As the dynamic range of speech can vary the energy of speech is not a very reliable method. Therefore a comparison of energy to tracked peak energy $E_{max}$ is used. When voiced the energy will be closer to the tracked peak energy, when unvoiced it will be lower. Energy to Peak Energy Ratio ($E/E_p$) is defined as

$$
E/E_p = \frac{2E_o}{E_o + E_{max}}
$$

(3.38)

where $E_o$ is the energy of the current signal. This tracking is detailed in [34].
3.6. Voicing Estimation

3.6.1.4 Low Band to Full Band Energy Ratio

Voiced speech usually follows the pattern of having more energy in the lower part of the spectrum. Unvoiced speech does not follow this pattern as it tends to have a flat spectrum with even energy across the band of frequencies. The energy ratio of the signal between 0 and 2 kHz and the energy of the speech spectrum gives a good indication of the voicing content of the speech signal. Voiced speech will give a high LF/FF ratio close to unity whereas unvoiced will give a lower LF/FF ratio. Figure 3.7 shows a plot of Energy to Peak Energy Ratio $E/E_P$ and Low Band to Full Band Energy Ratio $LF/FF$.

3.6.1.5 Normalised Correlation

The PCWs of voiced speech are very similar to their neighbours, the structure of unvoiced PCWs are considered to be random in nature with no obvious repetitions of the signal. This factor is given by

$$R_P = \frac{\sum_{n=0}^{N-1} s(n)s(n - N)}{\sqrt{\sum_{n=0}^{N-1} s(n)^2} \sqrt{\sum_{n=0}^{N-1} s(n - N)^2}}$$

(3.39)

where $N$ is the length of the signal and $s(n)$ is the current speech sample. The Normalised Correlation is calculated between the current pitch cycle and the two neighbouring cycles. Shifts of five samples either side are used to prevent incorrect values due to errors in the exact location of the pitch cycles. The maximum value from the correlation with the two neighbouring cycles is selected.

3.6.1.6 Pre-Emphasis Energy Ratio

A higher sample to sample correlation is present in voiced speech than unvoiced. For voiced speech the normalised pre-emphasis energy ratio will be low, unvoiced speech with its low sample to sample correlation will have a high normalised Pre-Emphasis Energy Ratio. The ratio therefore exploits first order correlations and is defined as
3.6. Voicing Estimation

Figure 3.8: A section of speech and the correlation values Normalised Correlation $R_P$ and Pre-Emphasis Energy Ratio $P_e$. Suggested threshold values are shown (dashed)

\[
P_e = \frac{\sum_{n=1}^{N-1} s(n)}{\sum_{n=0}^{N-1} |s(n) - s(n-1)|}
\]  

(3.40)

where $s(n)$ is current speech sample. Figure 3.8 shows a plot of Normalised Correlation $R_P$ and Pre-Emphasis Ratio $P_e$.

Each of the parameters described above provides a good indication of whether the speech section under consideration is voiced or unvoiced. Most parameters will give the same decision but frequently one or more of the parameters gives the incorrect indication. Therefore a final weighting decision is made based on all the parameters, this usually involves the construction of thresholds which are frequently determined perceptually. This decision logic is described further in 4.5.2.5 in the case of a hard decision used in the PS SB-LPC.
3.6.2 Soft Decision Voicing

Soft decision voicing is employed to make a decision when speech is considered to be a mixture of voiced and unvoiced speech. This is typically carried out in the frequency domain by separating the spectrum into several bands and making a decision for each frequency band. This is described in more detail in Sections 4.5.2.6 and 5.4.2.1.

3.7 Quantisation

In general, a digital speech signal is obtained by sampling and quantising a low-pass filtered version of the continuous speech signal. Ideal sampling of a band-limited signal is a process in which no information is lost. Digital speech is obtained by quantising and coding the sampled speech signal, thus representing each sample with finite precision. Quantisation is a lossy process and distortion always occurs, quantised speech can generally be modeled as a perfect, infinite resolution sampled signal plus a quantisation error. The goal therefore of any quantisation process must be to minimise this error. There are two fundamental quantisation techniques: scalar and vector quantisation.

3.7.1 Scalar Quantisation

Scalar quantisation maps each value to the nearest quantiser level from an infinite set of levels. The number of bits $B$ required to represent $l$ levels is given by

$$B = \log_2(l)$$

For a uniform quantiser the quantisation step size is given by $D/l$ where $D$ is the dynamic range of the input signal. Uniform quantisers assume that the distribution of the input values is even across their dynamic range. In speech coding this is rarely the case so due to their poor performance they are hardly used. Quantiser performance can be improved by adding more levels across sections of the input signal which are more densely populated therefore matching the statistical properties of the input signal. These nonuniform quantiser schemes are generally based on compounding logarithmic sampling such as A-law and $\mu$-law PCM [35].
3.7. Quantisation

As speech is a non-stationary signal whose characteristics change with time, adaptive quantisers whose characteristics dynamically change to match the properties of the speech signal were introduced. The basic idea is to vary the step size of the quantiser to match the variance of the input signal. Adaptive quantisers have been introduced that significantly increase the signal to noise ratio when compared to nonuniform quantisers at the same bit rate. Adaptive Differential PCM [9] operates on narrow band speech at 32 kbps at quality equal to logarithmic PCM at 64 kbps [8].

Scalar quantisation is a simple process that requires little storage space, is robust to channel errors and low in computational complexity. A more efficient quantisation method used in speech coding is known as Vector Quantisation. Vector quantisation (VQ) considers a entire set of values as an entity and allows for direct minimisation of the quantisation distortion. As many of the parameter sets used in speech coding such as LPCs exist in inter related sets, VQ of these sets exploits many of the intra-set correlations.

3.7.2 Vector Quantisation

Vector quantisation uses the basic idea to code values from a multidimensional vector space into values from a discrete subspace of lower dimension. The lower space vector requires less storage and the data is thus compressed. The transformation into the subspace is achieved by the use of a codebook. The K dimensional input vector of amplitude levels \([x_1, x_2, ..., x_k]\) is compressed by choosing the nearest vector (codeword) from a set of N dimensional vectors \([y_1, y_2, ..., y_n]\). All possible combinations of the N dimensional vector form the codebook. The codebooks used within a vector quantiser are precomputed and require a training process in order to be populated.

The operation of a vector quantiser is summarised in Figure 3.9. The VQ encoder takes the input vector \(x\) and outputs the index \(i\) of the codeword from the N dimensional codebook that offers the lowest distortion. The lowest distortion is found by evaluating a measure such as the Mean Squared Error (MSE) distance between the input vector and each codeword in the codebook. Once the closest codeword is found, the index of
that codeword is transmitted to the VQ decoder. When the VQ decoder receives the index of the codeword, it replaces the index with the associated codeword \( \hat{x} \).

\[
\begin{align*}
\text{VQ Encoder} & \quad \text{VQ Decoder} \\
\text{Channel} & \\
\text{Codebook} & \quad \text{Codebook}
\end{align*}
\]

Figure 3.9: A block diagram of vector quantisation system operation

Vector quantisation can produce good results though the subsequent complexity and memory requirements are prohibitively high. Because of this sub-optimal VQ routines have been investigated one of the most popular is known as Split Vector Quantisation (SVQ).

3.7.2.1 Split Vector Quantisation

Split vector quantisation can reduce the complexity and memory requirements of VQ by splitting the vector into several parts and quantising the sub-vectors with separate codebooks. In a SVQ system, an input vector \( x \) can be represented by a vector \( \hat{x} \) given by

\[
\hat{x} = \{ \{ y_0^0(0), \ldots, y_0^0(N_0) \}, \ldots, \{ y_{K-1}^i(0), \ldots, y_{K-1}^i(N_{K-1}) \} \}
\]

where the vector \( x \) is divided into \( K \) sub-vectors each of length \( N_k \). \( y_k^i(m) \) is the \( n^{th} \) element of the \( j^{th} \) codewector from the \( k^{th} \) codebook, \( i_k \) is the codebook index for the \( k^{th} \) sub-vector.

The complexity of a full vector search process requires one multiply add instruction to compare each of the \( N \) vector elements in the vector, the complexity of the search in a codebook with \( L \) vectors, with \( L = 2^B \) with \( B \) the number of bits, is given by

\[
C = N L = N 2^B
\]
The memory location to store the codebook assuming each codevector requires \( N \) locations is given by

\[
M = N L = N 2^B
\]  
(3.44)

The complexity and memory requirements for SVQ are the sum of the requirements of the various sub-vector quantisers. Splitting the vector into \( K \) sub-vectors of lengths \( \{N_1, N_2, ..., N_K\} \) and quantised with codebook sizes \( \{2^{B_1}, 2^{B_2}, ..., 2^{B_k}\} \) will have the following complexity and memory requirements

\[
C = \sum_{k=1}^{K} N_k 2^{B_k}, \quad M = \sum_{k=1}^{K} N_k 2^{B_k}
\]  
(3.45)

Complexity and memory storage for VQ and SVQ is given in Table 3.1 for a 10 element vector.

<table>
<thead>
<tr>
<th>Sub-vectors</th>
<th>Vector Sizes</th>
<th>Bits</th>
<th>Complexity</th>
<th>Memory storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>24</td>
<td>1.67 \times 10^8</td>
<td>1.67 \times 10^8</td>
</tr>
<tr>
<td>2</td>
<td>5, 5</td>
<td>12, 12</td>
<td>40960</td>
<td>40960</td>
</tr>
<tr>
<td>3</td>
<td>4, 3, 3</td>
<td>8, 8, 8</td>
<td>2560</td>
<td>2560</td>
</tr>
</tbody>
</table>

Table 3.1: Complexity and memory requirement comparison between various SVQ schemes

The complexity is measured in terms of required operations. A comparison between two vectors of length \( N_k \) requires \( N_k \) operations, therefore in a full search codebook of size \( 2^{B_k} \) the search requires \( 2^{B_k} N_k \) operations. Memory storage is in terms of memory words, assuming one codebook entry can be stored using one memory word.

Typically these need to be performed every 20ms in a speech coder, i.e. 50 times per second. The complexity and storage for the case of 1 sub-vector are clearly impractical, keeping in mind that typical complexity figures for existing standard coders are in the 20-50 MIPS range [14]. The case for 2 sub-vectors is still rather complex, as typically 2 MIPS would be used for complexity which is a large chunk out of 20 MIPS, and the memory requirement is high at 40k words just for the quantisation tables when
typically the Read Only Memory (ROM) used in speech coders is normally less than 5-6k words [14].

A major disadvantage of SVQ is that the correlations between elements in different sub-vectors are not exploited and the quantisation efficiency decreases as the size of sub-vectors reduce. Multi-stage vector quantisation (MSVQ) is an attractive alternative as it has a better ability to exploit the correlation advantage and has been shown to outperform the split codebook approach [36].

3.7.2.2 Multi Stage Vector Quantisation

Multi Stage Vector Quantisation (MSVQ), first proposed for the quantisation of LPC coefficients by [37] is a technique that reduces memory as well as computational complexity. The encoding error of a vector quantiser is formed by taking the difference between the original and quantised vectors and then feeding it into a second stage vector quantiser. The process can be repeated by feeding the second stage error into a third stage vector quantiser and so on. Therefore during MSVQ the input vector $x$ is quantised in several stages by a number of codebooks and the quantised vector $\hat{x}$ is formed by summing the vectors for the various codebooks

$$\hat{x} = y_0^{i_0} + y_1^{i_1} + \ldots + y_{K-1}^{i_{K-1}}$$

(3.46)

where $K$ is the number of stages, $y_k^j$ is the $j^{th}$ code vector from the $k^{th}$ codebook and $i_k$ is the codebook index for the $k^{th}$ stage. The codebooks at each stage can be relatively small, reducing storage requirements. This approach leads to a reduction in both complexity and memory storage compared to single stage VQ. The general memory requirement of a MSVQ is

$$M = N \sum_{k=1}^{K} 2^{B_k}$$

(3.47)

where $K$ is the number of stages, each of $B_k$ bits and $N$ is the length of the input vector. As the entire vector is quantised at each stage the gain from intra vector correlation is not lost though memory and computational complexity can be higher than SVQ.
During VQ each vector in each codebook is searched once, there are several search strategies available such as the sequential and full search [2]. In the sequential search strategy the input vector \( x \) is quantised by the first codebook \( Y_0 \) with the vector \( y^{i_0}_0 \) of index \( i_0 \) chosen to minimise the quantisation error. The index of the first codebook \( i_0 \) is then fixed and the quantisation error \( x - y^{i_0}_0 \) is then computed. The error from this stage is then used as the input to the next stage and is quantised. This strategy is then repeated for each stage in the codebook, the complexity of this search is equal to the complexity of a full search through each codebook and is given by

\[
C = N \sum_{k=1}^{K} 2^{B_k} \tag{3.48}
\]

The sequential method is sub-optimal as there is no guarantee that the set of codebooks vectors giving the lowest overall distortion will also give the lowest intermediate distortion. The full search strategy is optimal as it performs a full search on all codebooks jointly. Every combination of codebook vectors \( \hat{x} = y^{i_0}_0 + y^{i_1}_1 + ... + y^{i_{K-1}}_{K-1} \) is tested against the original input vector. This optimisation comes at the cost of greater complexity of

\[
C = N \sum_{k=1}^{K} 2^{B_k} \tag{3.49}
\]

A reduced complexity method which keeps many of the full search advantages is known as M-best tree search. A M-best trees search operates by exploring a certain number of M paths in the quantiser tree. At each stage the M-best vectors are chosen. M quantisation errors are computed and each one is passed to the second stage. The second stage is searched M times one for each of the error vectors. After the second stage the M vectors that achieve the lowest overall distortion at the end of the second stage are kept. This procedure is repeated for each stage of the codebook. At the last stage the path with the lowest overall distortion is selected.

This process is illustrated in Figure 3.10. M here is equal to 2. At the the first stage two paths are selected from codebook \( Y_0 \), at the second stage codebook \( Y_1 \) is searched to find the M vectors matching the best \( X \). At the third stage this process is repeated and the final M path is shown, of these the path with the lowest distortion is chosen.

The complexity of this tree search method can be shown to be
3.7. Quantisation

For $M = 1$ the complexity is the same as the sequential search. As the $M$ factor does not apply to the first stage this can be exploited when designing the codebook structure. This method has been shown to give good results [34] and is widely used in this project.

3.7.3 Codebook Training

The codebooks used in VQ are stored for use and require a period of training in order to fill the codebooks. A popular method to train the codebooks used in vector quantisation is the Linde Buzo Gray (LBG)[38]. This is an iterative algorithm that will produce a codebook to minimise the total quantisation error over the whole of the training database. During the training process it designs a $L$ level codebook by partitioning the $N$ dimensional space into $L$ non overlapping cells of $C_i$. Each cell being assigned a vector $y_i$. If $x$ is found to belong within $C_i$ during quantisation then it is represented
3.7. Quantisation

by \( y_i \). LBG is a relatively fast and can give good results; it has been widely used in this project.

3.7.3.1 The LBG Algorithm

A detailed description of the LBG algorithm follows:

1. **Initialisation**: An initial one entry codebook is chosen \( C_1 \) with the first codevector \( C_1(0) \) computed as the average of the \( M \) vectors \( x_m \) in the training database

\[
C_1(0) = \frac{1}{M} \sum_{m=1}^{M} x_m
\]

This is design stage \( N = 1 \).

2. **Splitting**: Each vector in the codebook \( C_N \) is split in two, generating codebook \( C_{N+1} \)

\[
C_{N+1}(k) = (1 + \epsilon)C_N(k) \quad (3.52)
\]
\[
C_{N+1}(2N-1 + k) = (1 - \epsilon)C_N(k) \quad (3.53)
\]

where \( \epsilon \) is a value very small in magnitude.

3. **Optimisation**: Step A) The training vectors are partitioned into clusters, each cluster being associated to a codevector \( C_N(k) \). Each training vector \( x_m \) is allocated to the cluster corresponding to the codevector \( C_N(k) \) which minimises a MSE measure. Step B) Each codevector is updated as the average of the training vectors present in the corresponding cluster. This reduces quantisation error in the cluster. Steps A) and B) are then repeated until there is no significant improvement in quantisation error.

4. Steps 2 and 3 above are repeated until the codebook of desired size has been found.

**MSVQ codebook training**

The LBG algorithm is not designed for the multistage codebooks as used in MSVQ routines. However it has been adapted for MSVQ codebook training. Initially a sequential optimisation technique was designed. This LBG based technique first designs
a codebook for stage 1 of MSVQ, the quantisation errors for the training database are then found and the codebook for stage 2 MSVQ is trained over them. This is then repeated for each stage giving the final set of codebooks.

A better technique is known as iterative sequential optimisation. Using sequential optimisation an initial set of MSVQ codebooks are chosen for each stage. Each codebook is then optimised by assuming all the other stages are fixed and known, i.e. the quantisation error using all the other stages except the current one is computed and an updated version of the current codebook is found. The process is iterated until the codebooks have converged.

A method to jointly optimise all codevectors of all stages after each iteration using simultaneous joint codebook design was proposed in [36]. Although highly complex it results in better performance than iterative sequential optimisation and has been used in this project to train the MSVQ codebooks.

### 3.8 Concluding Remarks

In this chapter the main parameters required to accurately code the speech signal have been introduced. The pitch, voicing and LPC parameters must be accurately represented if speech is to be successfully reproduced synthetically. In order to be used in a practical system these values must be accurately quantised. Many of these concepts have been applied during the course of this research and they form the basis of the following chapters. The next chapter will discuss sinusoidal speech coding in general and give a detailed discussion on the PS SB-LPC which forms the basis of this work.
Chapter 4

Sinusoidal Speech Coding

4.1 Introduction

This chapter introduces sinusoidal speech coders and in particular PS sinusoidal speech coders. Vocoder s are parametric coders that model the main features of the human speech production mechanism; vocal tract, pitch period and voicing status. Sinusoidal coders [39] treat speech as a sum of sine waves but can be considered to be a variant of vocoders as they make use of these three elements described above to reduce their bit rate.

The first part of this chapter summarises the main work that has been produced in relevant low bit rate coders in the past. The second half of this chapter introduces the Pitch Synchronous Split Band LPC (PS SB-LPC) [40] a PS sinusoidal coder that was used as a basis of the work presented in this thesis.

4.2 Existing techniques

4.2.1 Channel Vocoder

The first practical vocoder was demonstrated in 1939 by Homer Dudley [41]. In a channel vocoder the speech spectrum magnitudes are modeled by channels of contiguous
4.2. Existing techniques

variable-gain bandpass filters. Unvoiced speech was modeled with random noise and
voiced speech with a pulse train generator, the period of which is set by a derived
pitch. The excitation is then scaled at the decoder by the magnitudes of the frequency
channels. Due to technological limitations the synthetic speech produced was poor and
also a large number of filter channels (with consequent high digit rate) were required.
Later implementations with more modern technology produced speech at rates between
2.4 and 1.2 kbps [42].

4.2.2 Cepstral Vocoder

This coder separates the speech waveform, which is assumed to have been produced
by the convolution of the excitation spectrum $E(w)$ and vocal tract frequency response
$H(w)$, using time frequency domain relationships. Applying a log function to the speech
magnitude spectrum separates the speech spectrum into the sum of two functions

$$
\log|S(w)| = \log|E(w)H(w)| = \log|E(w)| + \log|H(w)|
$$

(4.1)

Applying an inverse Fourier Transform transfers the signal into an alternative time
domain where the excitation and vocal tract occupy different regions. The vocal tract
response varies slowly whereas the excitation varies quickly thereby occupying different
regions in this new domain. The vocal tract information may be obtained by firstly
multiplying the signal by a rectangular window (lifter) of unit height and of a length
long enough to contain all the low frequency information pertaining to just the vo-
cal tract. A process known as Cepstral or Homomorphic deconvolution. This vocal
tract information is then quantised and transmitted and the process is reversed at the
decoder. These coders can produce good quality at low rates [43]. However the per-
formance of the separation process of (4.1) degrades when the speech contains noise
as the speech spectrum is no longer simply the multiplication of excitation and vocal
tract information.

4.2.3 LPC Vocoder

Linear Predictive Coding (LPC) coders model the vocal tract by using a linear pre-
dictive filter and model the excitation using a pulse generator, using the source filter
4.2. Existing techniques

model as described in Section 3.2. LPC modeling allows the speech spectral shape to be accurately represented and efficient parameter quantisation is possible. Also the extraction of the LPC values is relatively straightforward. Many coders are based on this model including the Mixed Excitation LP (MELP) and the Split Band LPC (SB-LPC) both of which are introduced in this chapter.

4.2.4 MELP Vocoder

The MELP [44] coder is a United States Department of Defense speech coding standard used mainly in military applications and satellite communications, secure voice, and secure radio devices. MELP is based on the traditional LPC sinusoidal model but includes additional excitation modes to produce more natural sounding speech as these additional modes can represent a richer ensemble of possible speech characteristics. These are mixed excitation, aperiodic pulses, pulse dispersion and adaptive spectral enhancement. The mixed excitation makes the MELP coder robust in difficult background noise environments making it popular in military applications.

The mixed excitation is implemented using a multi-band mixing model simulating the frequency dependent voicing strength by using a filter bank, this reduces the buzz associated with simplistic voiced/unvoiced LPC coders. Aperiodic pulses model speech transitions where erratic pitch pulses can occur, pulse dispersion is used to spread the energy of the pulse within a pitch period. Adaptive spectral enhancement is applied to enhance the formant structure in the synthetic speech. A block diagram of the MELP

![Figure 4.1: Block diagram of MELP decoder](image-url)
4.2. Existing techniques

decoder is given in Figure 4.1.

4.2.5 MBE Vocoder

The Multi Band Excitation (MBE) vocoder [45] is a sinusoidal coder that models speech as a linear combination of sinusoidal waveforms with time varying amplitudes, phases and frequencies. It transmits a voicing decision for several frequency bands allowing the synthetic speech to be a mixture of periodic and noise like signals. The sinusoids are assumed to be all harmonics of a single fundamental frequency, given by the pitch of the input speech. The pitch is extracted by synthetic spectral matching. The speech is then separated into a number of frequency bands and the voicing of each band determined. The magnitudes and phases of the harmonics are extracted from the short time Fourier Transform. The decoder uses a bank of sinusoidal oscillators to generate the voiced part of speech and utilises the amplitudes and phases to produce synthetic speech. Spectrally shaped random noise is used to generate the unvoiced part of speech. For voiced parts the parameters are interpolated across the frame with a harmonic birth and death process to provide for voicing changes which cause harmonics to appear or disappear. The Improved Multi Band Excitation (IMBE) [46] coder does not transmit

![Figure 4.2: Block diagram of IMBE coder](image-url)
4.2. Existing techniques

the phases as synthetic phase is used at the decoder. A block diagram of the IMBE coder is given in Figure 4.2.

4.2.6 SB-LPC Vocoder

The SB-LPC is a high quality speech coder which has been developed at the University of Surrey [30] and [34]. The PS SB-LPC model of Section 4.5 is based upon the SB-LPC. The SB-LPC uses a sinusoidal model to synthesise an excitation sequence, this sequence is then used to stimulate an LPC filter which models the vocal tract. It can be considered to be an amalgam of a sinusoidal and LPC coder. However whereas the MBE coder uses a voicing decision for every harmonic or group of harmonics the SB-LPC assumes all bands to be voiced from DC to a unvoiced cut off frequency. It is this Split Band Voicing hypothesis which gives the coder its name. A block diagram of SB-LPC decoder operation is given in Figure 4.3.

![Figure 4.3: Block diagram of SB-LPC decoder operation](image)

The SB-LPC encoder operates on either narrow band speech at 8 kHz or wide band speech sampled at 16 kHz. Parameters are extracted every 10 ms or 20 ms depending upon the mode. The parameters used to represent the speech are:

- LPC coefficients, 10\textsuperscript{th} order for narrow band (8 kHz sampled speech) and 16\textsuperscript{th} order for wide band (16 kHz sampled speech).
4.3 Pitch Synchronous Speech Coding

- Pitch Period.
- Voicing cut off frequency.
- Spectral Amplitudes.
- Speech Energy.

In this section the main vocoders and sinusoidal based vocoders have been summarised. The next section describes PS coding in more detail and the motivations behind it.

4.3 Pitch Synchronous Speech Coding

Traditional TS speech coders such as the SB-LPC and MELP cannot currently produce speech of toll quality owing to the inaccurate modeling of perceptually important speech transitions and the lack of accurate speech parameter analysis [12] and [47]. In order to improve the quality of TS speech coders, several PS speech coders have been proposed. By extracting and analysing speech on a smaller pitch cycle basis rather than at regular discreet points PS coders can overcome the disadvantages of the TS coders.

4.3.1 Introduction

The limitations of the model employed by TS coders means that even with an increase in the bit rate the decoded speech quality does not increase significantly. Previous analysis of the SB-LPC showed that a major weakness lies in the model's lack of ability to accurately reproduce non-stationary sections of speech in the original speech waveform; these sections are only a small percentage of the original speech signal but convey a lot of information and their faithful reproduction seems to be very important perceptually [48]. During sections of regular speech where the original speech signal is stationary the decoded speech produced by the SB-LPC is of the highest quality.

The 4 kbps SB-LPC extracts parameters every 10 ms. A large window centred at an analysis point on the speech waveform is used to extract the parameter data. These windows need to be long enough to capture enough speech information but not so long that temporal resolution is seriously degraded. This model assumes that speech is
stationary during this extraction, however speech is not stationary at all speech sections such as onsets, offsets and transitions, as a result smoothing at these speech sections can occur.

Figure 4.4 demonstrates the smoothing that can occur between the 10 ms extracted parameter sets at the decoder. In the figure (top) a extracted speech segment can fall under three analysis windows at the analysis stage. Interpolation between these segments at the decoder (bottom) can therefore affect up to 40 ms of speech which results in a smoothing of the speech waveform. This interpolation at the decoder is done to remove the steps between the 10 ms extracted parameter sets at analysis. Figure 4.5 shows a section of speech synthesised by the SB-LPC. The SB-LPC smooths the onset and the fine detail in the original speech has been lost. This smoothing is the result of the large analysis window used to extract the parameters and secondly, as the decoder interpolates between the parameters extracted at 10 ms intervals as shown in Figure 4.4. This is also illustrated in Figure 4.6 which shows a speech transition. The SB-LPC has failed to model the two cycles of speech starting at 640 samples which have noisy content in the original speech.

Figure 4.7 compares the synthetic speech produced by a TS coder and the PS SB-LPC. Four areas of speech, namely onsets and offsets, are highlighted. Highlighted sections
4.3. Pitch Synchronous Speech Coding

Figure 4.5: A speech onset. Original speech (top) and synthetic speech produced by the SB-LPC (bottom). The dashed vertical lines indicate the TS analysis points. [40]

Figure 4.6: A speech transition. Original speech (top) and synthetic speech produced by the SB-LPC (bottom). The dashed vertical lines indicate the TS analysis points. [40]

Figure 4.8 highlights three speech transitions where the speech modeling of the PS SB-LPC is superior to that provided by a TS coder - the SB-LPC. In the TS coder at section
4.3. Pitch Synchronous Speech Coding

Figure 4.7: Comparison of synthetic speech produced by the SB-LPC (middle) and PS SB-LPC (bottom)

Figure 4.8: Comparison of synthetic speech produced by the SB-LPC (middle) and PS SB-LPC (bottom)

...the speech transition is smeared and at sections b and c the cycles do not contain cycles with the correct voicing characteristics. The following section describes PS coding of
speech in more detail. Several PS coders discussed and a detailed description of the PS SB-LPC which forms the basis of this project is presented.

4.3.2 Pitch Synchronous Multi Band Coder

In [49] the Pitch Synchronous Multi Band (PSMB) coding of speech was presented. The PSMB coder uses an MBE speech model to generate a PCW that is coded and transmitted as a representation of a speech frame. The MBE model is used to generate the PCW at the encoder, the PCW is then synthesised and replicated to produce a frame of synthetic speech. If the PCW is similar to the PCW from the previous frame it is encoded using a Length Converted Excitation (LCE) codebook and a stochastic codebook. A Band Limited Single Pulse excitation (BPSE) codebook and stochastic codebook are used if the PCW is different from the previous frame. An AbS procedure is used to determine whether the PCWs from different frames are related. The general structure of the PSMB encoder is presented in Figure 4.9.

![Figure 4.9: Block diagram of the PSMB encoder](image)

The choice of excitation from one of the two codebooks and the entry from the stochastic codebook is then used at the decoder to excite a LPC filter producing frames of synthetic speech.
4.3.3 Waveform Interpolation Coding

Waveform Interpolation (WI) coding was first introduced by [50], the first version was called Prototype Waveform Interpolation (PWI) coding. PWI encoded voiced segments only, unvoiced segments were coded with other schemes. PWI coding assumes that speech evolves slowly and instead PCWs are transmitted at regular intervals and at the decoder the PCWs recovered through interpolation. In [51] PWI became WI coding which can code both voiced and unvoiced speech segments. WI coding separates the Characteristic Waveforms (CW) into a Slowly Evolving Waveforms (SEW) and Rapidly Evolving Waveform (REW). The SEW describes the periodic component of the speech signal, the REW describes the noise component of the speech signal. Since these two waveform types are different perceptually they can be processed differently thus enhancing coding efficiency.

LP analysis is carried out on the speech to produce the residual signal. Then the pitch is estimated and the residual is decomposed into a series of CWs. The CWs are subsequently aligned and normalised in power so that they can accurately represent a 2-D surface describing the evolution of the waveforms. The quantiser carries out the SEW-REW decomposition and parameter quantisation, at the decoder the parameters are dequantised and the CWs are reconstructed from the transmitted SEWs and REWs. The residual signal is then reconstructed from the CWs and passed to the synthesis filter where the speech is reconstructed. The general structure of a WI decoder is presented in Figure 4.10.

4.3.4 Pitch Synchronous MELP

In [52] a PS MELP coder was proposed, known as the I-MELP. This coder uses an improved sub frame based correlation method to estimate the pitch (PEA). The low pass filtered residual is used to find pitch boundaries. The normalised correlation based algorithm searches a range of pitch lags around the average pitch of the frame. When the estimated largest pitch lag correlation is found, the PEA is repeated on N times the up sampled signal around the integer lag estimate to find the cycle length in 1/N sample resolution. N is set to 10 in the I-MELP. This procedure was found to improve
4.4 Overview of Pitch Synchronous SB-LPC

In the PS SB-LPC harmonic analysis is carried out on individual pitch cycle waveforms (PCWs) rather than using a large window. The PCWs are segmented using an initial pitch detection algorithm (PDA) [34] and an algorithm that analyses the low pass filtered rectified speech residual energy as described in [53]. Once the speech has been segmented, the PCW parameter information is estimated for each PCW. The parametric information is quantised using a Joint quantisation Interpolation (JQI) scheme [54]. An overview of PS SB-LPC operation is given as Figure 4.11.

The speech is segmented into separate PCWs by analysis of the cycle energies. The algorithm used to segment the speech produces jitter and size errors in the PCW lengths and therefore a smoothing algorithm at the analysis stage is used to prevent artefacts being produced in the synthesised speech.

Once the PCWs have been found, for each cycle of speech the following parameters are extracted:

- Linear Prediction Filter Coefficients
4.5 Detailed description of PS SB-LPC

The following section describes work that has been previously carried out in the department and is detailed in [40]. This research presented here forms the basis of the work carried out in the following chapters.

The quantisation techniques employed aim to efficiently quantise the data whilst capturing the evolution of the parameters within the frame. No parameter smoothing is included in the decoder, any necessary interpolation is included in the quantisation routines. The encoder passes parametric information for each pitch cycle to the quantiser, the decoder receives the parametric information for each pitch cycle and treats them as if they were directly from the analysis stage. At the decoder of the PS SB-LPC the dequantiser passes parameters one cycle at a time to the synthesiser.

Figure 4.11: Basic operation of the PS SB-LPC

- Cycle size.
- Voicing cut off frequency.
- Spectral Amplitudes.
- Cycle Energy.
4.5. Detailed description of PS SB-LPC

4.5.1 Pitch Cycle Detection

The PS SB-LPC is reliant on a pitch cycle detection algorithm to segment the speech into PCWs. Incorrect identification of individual PCWs may result in either roughness in the synthesised speech or an abrupt change in pitch which causes artefacts in the synthesised speech. This section presents the method used and then describes a post processing algorithm that eliminates many of the effects of incorrect cycle detection. The method used to carry out the segmentation is known as the Trapezoidal Search, this operates on the modified low pass filtered LPC residual known here as Modified Time Envelope (MTE).

LPC parameters are extracted every 10ms using a 25 ms Hamming window. The residual is obtained by partly inverse filtering using LPC parameters interpolated at 4 points within each 10 ms frame. In some speech areas where the vocal tract is highly resonant, the excitation signal (residual) does not always contain any excitation pulses to identify. Therefore a limited amount of LP inverse filtering using a pole-zero filter based on the LP coefficients was used. This removes some of the effect of the vocal tract but leaves enough energy so that in highly resonant areas a large enough signal exists to allow the peaks to be identified. The transfer function of this chirped LPC filter used is shown in (4.2)

$$H(z) = \frac{A(z/\alpha)}{A(z/\beta)}$$  \hspace{1cm} (4.2)

When $\alpha = 1$ and $\beta = 0$ the filter has the same effect as the LP analysis filter, in that it removes the majority of the frequency domain shaping. If $\alpha = 1$ and $\beta = 1$ the filter has no effect. If $\alpha = 0$ and $\beta = 1$ the filter becomes an LP synthesis filter. In order to vary the amount of LP filtering $\alpha$ is set to 1 and $\beta$ is varied between 0.9 and 0.7. When the pitch value is below 30 samples the speech is more likely to be female and contain the highly resonant areas with little excitation. In such areas only a small amount of inverse filtering is required with $\alpha = 1$ and $\beta = 0.9$. Speech with a longer pitch value greater than 60 samples the value of $\beta$ is set to 0.7. This gives a large amount of LP inverse filtering and gives a signal that is closer to the excitation signal than the speech signal. For pitch values between 30 to 60 samples the values of $\beta$ are varied linearly.
4.5. Detailed description of PS SB-LPC

This low pass filtered rectified residual is then used to calculate the MTE, $s_{ENC}(n)$

$$s_{ENC}(n) = s(n) + 0.3s(n - T_0) + 0.1s(n + T_0)$$  \hspace{1cm} (4.3)

where $s(n)$ is the low pass filtered rectified residual. $T_0$ is the estimated pitch from the PDA.

![Figure 4.12: Original Speech (top), LP Residual (middle) and MTE signal (bottom)](image)

This filtering raises any peaks that have periodicity of $T_0$ and will increase the value of the troughs that do not have periodicity of $T_0$. This resultant signal is therefore periodic at the multiples of the estimated pitch period. $s_{ENC}$ is then time shifted to align it with the original speech signal. The original speech, LP residual and resultant MTE signals are shown in Figure 4.12.

Pitch cycles are then searched using a weighted cross correlation with a trapezoidal window of length $T$, where $T$ is estimated pitch from PDA. The effect of the trapezoidal window is to search for troughs separated by the estimated pitch value $T$. The correlation process begins at the end of the previously selected PCW. Circular cross correlations of the trapezoidal window and a segment of $T$ samples of the MTE signal is computed. The maximum weighted value of correlation identifies the PCW start.
4.5. Detailed description of PS SB-LPC

location. The signals used are shown in Figure 4.13. The PCW identified by the Trapezoidal Search contain large amounts of jitter in the cycle sample size. This is caused by the fact that the troughs of the MTE are wide and the identified PCW boundaries may vary slightly. Therefore the boundary locations are moved slightly to remove the cycle size jitter. In order to remove the cycle size jitter, $\vartheta$ in (4.4) is minimised

$$\vartheta = \alpha \frac{\sum_{i=1}^{N} |(p_{i+1} - p_{i}) - (p_{i} - p_{i-1})|}{N} + \beta \frac{\sum_{i=1}^{N} s_{ENC}(p_{i})}{N}$$  \hspace{1cm} (4.4)

where $p_i$ is the location of the start of the $i_{th}$ PCW, $N$ is the number of cycles in the frame and $s_n(p_{i})$ is the value of the input signal at the position of the PCW boundary. $\vartheta$ is minimised by allowing each boundary location $p_i$ to vary by two samples either side of its initial location. Therefore there are $5^{N+1}$ possible combinations searched. The values of $\alpha$ and $\beta$ were set experimentally to $\alpha = 3$ and $\beta = 1$.

4.5.1.1 Cycle Size Post Processing

The Trapezoidal Search routine correctly segments the speech in a high percentage of cases but occasional cycle detection errors occur which limit the synthetic speech quality. This is illustrated in Figure 4.14. At a speech transition the filtering operation

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.13.png}
\caption{Trapezoidal shape (top), MTE (middle) and Original speech (bottom)}
\end{figure}
4.5. Detailed description of PS SB-LPC

Figure 4.14: PCW sizes (top), Input speech (middle) and MTE (bottom). The dotted lines indicate pitch boundaries identified by Trapezoidal Search

has produced a uneven cycle leading to an incorrect variation in PCW sizes. This causes an audible distortion in the synthesised speech. The method chosen to solve this problem is to ignore or expand the partial cycles during synthesis. During analysis, PCWs following a cycle detection error are not modified. The erroneous cycle sizes are adjusted and the analysis size expanded to the adjusted cycle length. The partial cycles are expanded to the size of a complete cycle or discarded.

Three rules were designed by identifying changes in pitch cycle lengths that are likely to be due to pitch cycle detection errors. Firstly single cycles that significantly larger or smaller than the cycles either side of them are smoothed. $S_i$ the size of cycle $i$ is adjusted as follows (4.5):

$$\tilde{S}_i = \frac{S_{i-1} + S_{i+1}}{2} \quad \text{if} \quad \begin{cases} S_i > \alpha S_{i+1} \text{ and } S_i > \alpha S_{i-1} \\ \text{or } (S_i > \beta S_{i+1} \text{ and } S_i > \beta S_{i-1}) \end{cases}$$

(4.5)

Values of $\alpha = 1.1$ and $\beta = 0.83$ were found to give the best perceptual results. Secondly, smoothing cycle size errors in regions where the surrounding cycle sizes are constant or
smoothly evolving. Single cycles with sizes different to those around them are adjusted as follows (4.6) - (4.7):

\[ \bar{S}_i = \frac{S_{i-1} + S_{i+1} - 1}{2} \quad \text{if} \quad \begin{cases} (S_{i-1} > S_{i+1} \text{ and } S_{i+1} = S_{i+1}) \\ and \\ (S_i > S_{i-1} \text{ and } S_i > S_{i+1}) \text{ or } (S_i < S_{i-1} \text{ and } S_i < S_{i+1}) \end{cases} \] (4.6)

\[ \bar{S}_i = S_{i-1} \quad \text{if} \quad \begin{cases} S_{i-2} = S_{i-1} = S_{i+1} \text{ or } S_{i-1} = S_{i+1} = S_{i+2} \end{cases} \] (4.7)

Thirdly a rule was added to remove jitter cycle sizes caused by non-integer pitch values. Cycles that are only one sample longer or shorter in length than their neighbouring cycles are adjusted as follows (4.8) - (4.9):

\[ \bar{S}_i = S_{i-1} \quad \text{if} \quad \begin{cases} (S_i = S_{i-1} + 1 \text{ and } S_i >= S_{i+1}) \\ or \\ (S_i = S_{i-1} - 1 \text{ and } S_i < S_{i+1}) \end{cases} \] (4.8)

\[ \bar{S}_i = S_{i+1} \quad \text{if} \quad \begin{cases} (S_i = S_{i+1} + 1 \text{ and } S_i >= S_{i-1}) \\ or \\ (S_i = S_{i+1} - 1 \text{ and } S_i < S_{i+1}) \end{cases} \] (4.9)

These cycle size smoothing routines in conjunction with cycle size jitter removal in (4.4) significantly reduce the number of speech artefacts caused by pitch size errors. The next section details how the parametric information is taken from the identified cycle waveforms.

### 4.5.2 Pitch Cycle Based Analysis

This section describes the methods used to estimate the speech parameters from individual pitch cycle waveforms. Methods are described which estimate the LP coefficients, speech energy, spectral amplitudes and voicing cutoff frequency.
4.5.  Detailed description of PS SB-LPC

4.5.2.1  LPC Analysis

Many sinusoidal speech coders such as the SB-LPC extract LP speech parameters from fixed length portions of speech using the Autocorrelation Method (AM)\cite{26}. The AM method is used as the estimated parameters are guaranteed to be stable. In addition the method can be implemented with a relatively simple algorithm.

An alternative method for estimation of the optimum LP filter coefficients is the Covariance Method (CM)\cite{26}. Both of these techniques were described in Section 3.3. The AM method assumes that the signal is zero outside of the analysis period, to achieve this the signal is multiplied by a window that tapers to zero at each end. Therefore to obtain reliable LP coefficients the duration of the analysis period must be sufficiently long so that the tapering effect of the window has little influence. An analysis period of several pitch periods is needed to obtain reliable coefficients\cite{26}. This is in contradiction with the aim of PS analysis. The CM is better suited to PS analysis as no assumption is made about the signal outside of the analysis region and the LP parameters can be computed directly from a PCW.

In\cite{40} an investigation into the optimum method of extraction of LP coefficients was carried out. It was found that for analysis periods equal or less than one pitch cycle the CM outperformed the AM. Both the autocorrelation and covariance method have been implemented in this project for the PS extraction of LP parameters.

4.5.2.2  Energy

The PS SB-LPC speech model requires that the energy of each PCW be extracted. The energy value is used during speech synthesis to restore the relative amplitudes of the speech. In the PS SB-LPC an individual energy value is calculated for each PCW

\[ E_{ci} = \frac{1}{f_i - s_i} \left( \sum_{n=s_i}^{n<f_i} (s(n) - \bar{s})^2 \right)^{1/2} \]  

(4.10)

where \( E_i \) is the energy of the \( i^{th} \) PCW which starts at location \( s_i \) and finishes at location \( f_i \) within the speech signal \( s(n) \) which has a mean value of \( s(\bar{n}) \).
4.5. Detailed description of PS SB-LPC

4.5.2.3 Spectral Amplitudes

In the SB-LPC and PS SB-LPC encoders the LP filter models a large proportion of the spectral characteristics of the speech. Spectral amplitudes are used to transmit the remaining spectral characteristics. PS estimation of the amplitudes is carried out by taking the Discrete Fourier Transform (DFT) of a single PCW over its length \( N \) from 15 to 150 samples. The DFT \( S(k) \) of PCW \( s(n) \) is given by

\[
S(k) = \sum_{i=0}^{N-1} s(n) \exp^{-2\pi i nj/N} = \sum_{i=0}^{N-1} s(n) \left( \cos\left(\frac{2\pi nk}{N}\right) - j \sin\left(\frac{2\pi nk}{N}\right) \right)
\]

(4.11)

for \( k = 0,1,2,\ldots,N-1 \). Since \( s(n) \) is real, the amplitude of each harmonic \( A_k \) is calculated as \( |S(k)| \) and is given by

\[
A_k = |S(k)| = \left( \sum_{i=0}^{N-1} \left( s(n) \cos\left(\frac{2\pi nk}{N}\right) \right)^2 + \sum_{i=0}^{N-1} \left( s(n) \sin\left(\frac{2\pi nk}{N}\right) \right)^2 \right)^{1/2}
\]

(4.12)

assuming \( s(n) \) is a PCW speech residual found by carrying out LPC analysis as described in Section 3.3.

4.5.2.4 Voicing

The PS SB-LPC firstly uses hard decision techniques to classify the speech as voiced or unvoiced using the measurements introduced in Section 3.6. Then a time domain method based on peakiness is used to determine the correct cutoff frequency for individual pitch cycles which are voiced. This method is applied as many of the standard metrics used for analysis in TS coders such as the SB-LPC are based on periodicity which require several cycles of speech, these cannot be used with great accuracy for voicing analysis of single cycles which is required here.

4.5.2.5 Final Hard Decision Voicing Estimate

The hard decision voicing estimate uses a majority voicing scheme to classify the PCW voicing status. The six measurements are used with the threshold values in Table 4.1
4.5. Detailed description of PS SB-LPC

to classify each cycle as either

- Fully unvoiced
- Partially or Fully Unvoiced
- Uncertain

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Unvoiced threshold</th>
<th>Voiced threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_c$</td>
<td>$&gt; 0.36$</td>
<td>$&lt; 0.14$</td>
</tr>
<tr>
<td>$P_k$</td>
<td>$&lt; 1.0$</td>
<td>$&gt; 1.4$</td>
</tr>
<tr>
<td>$LF/FF$</td>
<td>$&lt; 0.51$</td>
<td>$&gt; 0.97$</td>
</tr>
<tr>
<td>$E/E_p$</td>
<td>$0$</td>
<td>$&gt; 0.67$</td>
</tr>
<tr>
<td>$R_p$</td>
<td>$&lt; 0.3$</td>
<td>$&gt; 0.9$</td>
</tr>
<tr>
<td>$P_e$</td>
<td>$&gt; 1.05$</td>
<td>$&lt; 0.5$</td>
</tr>
<tr>
<td>Combined</td>
<td>$&lt; 0.49$</td>
<td>$\geq 0.49$</td>
</tr>
</tbody>
</table>

Table 4.1: Voiced and unvoiced threshold values used for the six voicing measures and the combined measure when making final hard decision

If the number of measurements that classify a cycle as voiced is greater than the number as unvoiced, then it is voiced. The opposite case is also true. If all the measurements are indecisive then a decision is made from a combination of the values. To form the combined voicing measurement, each of the individual measurements are scaled so that the range between the threshold values is equal to one. The scaling is given by

$$
x = \begin{cases} 
TH_L & \text{if } x \leq TH_L \\
TH_U & \text{if } x \geq TH_U \\
(x - TH_L)/(TH_U - TH_L) & \text{otherwise}
\end{cases} \quad (4.13)
$$

where $x$ is the value of the voicing measure, $TH_L$ is the lower voicing threshold and $TH_U$ is the upper. The values are arranged such that values close to one indicate voiced speech and values close to zero indicated unvoiced. The arithmetic mean of these values is then used as the combined voicing measure. A threshold value of 0.49 was found to distinguish unvoiced from partially voiced speech.
4.5.2.6 Soft Decision Voicing

Voiced speech residual energy is usually concentrated around the glottal pulse resulting in higher peakiness values whereas unvoiced speech residual energy is spread, resulting in lower values. Using this information band-limited peakiness is used to determine the voicing cutoff frequency for those cycles declared as voiced by the hard voicing decision. If the PCW is fully voiced with the continued addition of higher frequency levels the voiced harmonics should continually raise the peakiness value. When the PCW is mixed voiced, the voiced harmonics should raise the peakiness value whereas the unvoiced should not, indicating that when the peakiness value levels or falls off the highest voiced level has been found.

Voicing in PCWs of the PS SB-LPC are classified into one of seven levels ranging from 0 to 4000 Hz, with the highest level indicating the voicing level cut off. The PCWs are band-limited in the spectral domain so each contains the frequency content of

$$0Hz \text{ to } f(V_i^{Cut})Hz$$  \hspace{1cm} (4.14)

The peakiness of each of the seven signals is then computed. The following is then repeated for each of the seven candidate voicing cut off frequencies.

- The PCW LP residual is DFT transformed into the frequency domain. DFT length is equal to the length of the PCW.
- Frequency content above the candidate voicing cut off frequency $f(V_i^{Cut})$ is set to zero.
- The band-limited signal is inverse transformed by a IDFT back to the time domain.
- The peakiness of the band-limited LP residual is measured.

The voicing cutoff frequency is determined by analysing the change in the band-limited peakiness values. Figure 4.15 shows a section of speech that is partially voiced. The band-limited peakiness falls from level five onwards indicating that frequency content above this is unvoiced.
4.5. Detailed description of PS SB-LPC

The voicing cut off frequency should be equal to the value that gives the highest peakiness measurement but it was found that the band-limited peakiness measurement itself sometimes varies slightly from cycle to cycle. To counteract this a small amount of historical bias was added. The peakiness value for the cut off level selected as the previous PCW voicing level is multiplied by a factor of $1 + \epsilon$. A value of $\epsilon$ of 0.07 was found experimentally to remove rapid changes in the voicing level. This is discussed further in Section 5.3.2.

4.5.3 Quantisation

To implement the speech coder into a communications network the speech parameter data must be quantised. Unlike TS sinusoidal coders the PS SB-LPC does not extract parameters at a fixed rate but does so according to the pitch lengths of the speech signal. Direct quantisation of the PS parameter sets would therefore lead to a coder with a source rate that would vary according to these pitch values. The majority of communication systems use fixed rate coders therefore quantisation techniques were
used to allow the PS sinusoidal data to be quantised at a fixed rate.

4.5.3.1 Pitch and Voicing Quantisation

For a frame containing only voiced cycles, the first and last PCW sizes are jointly quantised using a 8-bit vector quantiser. Using the dequantised boundary PCW sizes and a 2-bit shape codebook all four possible sequences of PCW size are searched. The shape codebook is a four entry vector table containing interpolation factors. One entry is for linear interpolation and three entries for stepped interpolation.

For unvoiced frames the two PCWs are set to the same length and the length is quantised using a 8-bit linear quantiser. Smoothing causes the encoder and decoder to lose synchronisation, for this reason the unvoiced frames are adjusted. If the decoder is ahead of the encoder the unvoiced frame is reduced in length, the frame is extended if the encoder is running ahead of the decoder.

For mixed voicing mode, the voicing levels of the voiced PCWs are quantised by averaging and quantising with a 3-bit scalar quantiser. The lengths of the voiced PCW are averaged and quantised using a 7-bit non-linear quantiser. The three bits that are used to quantise the PCW length interpolation factor in the voiced mode quantiser are used here to quantise the unvoiced PCW size.

4.5.3.2 Energy Quantisation

Sets of PCW energies are quantised using a joint-quantisation interpolation scheme using 14-bits. The quantiser operates similarly to the PCW length quantiser in that two boundary values and a shape vector are quantised. The energies of the first and last PCW in the frame are quantised. These boundary values are quantised using a 8-bit logarithmic joint quantiser with second order moving average prediction. The intra frame evolution of the energy is quantised using a 6-bit shape codebook. A separate codebook is used for each possible number of PCW per frame, no codebook is needed for frames containing two PCWs. The codebooks were trained using the LBG algorithm detailed in Section 3.7.2.
4.5. Detailed description of PS SB-LPC

4.5.3.3 LSF Quantisation

The LSFs in the PS-SBLPC are quantised using a Joint Linear Interpolation Quantiser as described in [54]. An optimum interpolation function is selected to describe the variation of LSFs over the frame, secondly boundary sets of LSFs are selected to minimise the overall quantisation distortion. This non linear description is shown as Figure 4.16.

The boundary vectors $\hat{x}_1$ and $\hat{x}_2$ and the interpolation function are selected so as to minimise the overall quantisation distortion.

The quantised PS LSF values are calculated by taking a weighting combination of the optimum boundary parameter sets. The $k^{th}$ set of quantised PS LSF $Y_k^j$ is calculated as:

$$Y_k^j = (1 - \alpha_k)\hat{x}_{k-1}^j + \alpha_k\hat{x}_1^j \quad (4.15)$$

if the $k^{th}$ PCW falls within the first half of the frame, or:

$$Y_k^j = (1 - \beta_k)\hat{x}_1^j + \beta_k\hat{x}_2^j \quad (4.16)$$

if the $k^{th}$ PCW falls within the second half of the frame. $\hat{x}_{k-1}^j$ is the dequantised optimum set of $j$ LSF from the previous frame. $\hat{x}_1^j$ and $\hat{x}_2^j$ are the dequantised optimum sets.

Figure 4.16: Quantisation using non-linear interpolation. $\bullet$ are the estimated parameter values, $\times$ are the quantised parameter values.
4.5. Detailed description of PS SB-LPC

From the current frame, \( \alpha \) and \( \beta \) vary linearly from zero to one during the two halves of the frame. The restriction of the values of \( \alpha \) and \( \beta \) limits the parameter variation to linear.

To allow the PS LSF parameters to vary freely within a frame, the JQI defines the PS parameters as a weighted combination of \( x_1^j \) and \( x_2^j \) and three weights \( \gamma, \alpha \) and \( \beta \). The synthesised PS LSF parameters are then calculated as:

\[
Y_k^j = \gamma_k x_{-1}^j + \alpha_k x_1^j + \beta_k x_2^j
\]

In order to restrict the range of quantised PS LSF the values of the shape function (represented by the weightings \( \gamma, \alpha \) and \( \beta \)) are restrained such that \( \gamma + \alpha + \beta = 1 \). Hence \( \gamma = 1 - \alpha - \beta \). Therefore the synthesised PS LSF parameters are calculated as:

\[
Y_k^j = (1 - \alpha_k - \beta_k)x_{-1}^j + \alpha_k x_1^j + \beta_k x_2^j
\]

This is then used to find the total quantisation error \( E \), between the \( K \) sets of quantised LSF parameters \( Y_k^j \) and the original estimated parameter sets \( \lambda_k^j \). This error can be calculated as:

\[
E = \sum_{k=0}^{K-1} \left( \sum_{j=0}^{P-1} (\lambda_k^j - Y_k^j)^2 \right)
\]

The total quantisation error \( E \) can be minimised with respect to \( \alpha_k \) and \( \beta_k \) by taking \( \partial E / \partial \alpha_k = 0 \) and \( \partial E / \partial \beta_k = 0 \) and solving for \( \alpha_k \) and \( \beta_k \). As a result the optimum shape vectors \( \alpha_k \) and \( \beta_k \) are calculated to minimise the total quantisation error given \( x_{-1}^j \) and \( x_1^j \) and \( x_2^j \). \( x_{-1}^j \) is calculated for the previous frame and is fixed.

Once optimum \( \alpha_k \) and \( \beta_k \) have been calculated, the quantisation error \( E \) is then minimised in terms of the edge parameters \( x_1^j \) and \( x_2^j \). By defining the overall individual error on LSF \( j \) as:

\[
E^j = \sum_{k=0}^{K-1} (\lambda_k^j - Y_k^j)^2 = \sum_{k=0}^{K-1} (\lambda_k^j - (1 - \alpha_k - \beta_k)x_{-1}^j + \alpha_k x_1^j + \beta_k x_2^j)^2
\]

and minimising with respect to each set of LSF by setting \( \partial E^j / \partial x_1^j = 0 \) and \( \partial E^j / \partial x_2^j = 0 \) and solving for \( x_1^j \) and \( x_2^j \) [54], results in a optimum sets of LSF \( x_1^j \) and \( x_2^j \) (the edge...
4.5. Detailed description of PS SB-LPC

vectors) given a fixed shape vector \((\alpha_k, \beta_k)\). An iterative calculation of \(x_1^j, x_2^j, \alpha_k\) and \(\beta_k\) is therefore used to obtain a near optimum solution. This iterative calculation of the edge and shape vectors is summarised in the flow chart of Figure 4.17.

![Flow chart of iterative parameter calculation of LSF quantisation edge vectors \(x_1\) and \(x_2\) and shape vectors, \(\alpha_k\) (\(alpha_k\)) and \(\beta_k\) (\(beta_k\)).](image)

**Figure 4.17:** Iterative parameter calculation of LSF quantisation edge vectors \(x_1\) and \(x_2\) and shape vectors, \(\alpha_k\) (\(alpha_k\)) and \(\beta_k\) (\(beta_k\)).

The JQI scheme therefore requires the quantisation of two sets of LSF parameters (edge vectors) and two shape vectors of length \(K\). The two sets of LSF parameters are quantised using a 30-bit MSVQ. The shape vectors are jointly quantised as a single vector of length \(2K\) using the LBG algorithm with a codebook size of 6 bits. The shape vector is quantised using a separate vector codebook for each of the possible values of
4.5. Detailed description of PS SB-LPC

K. Nine codebooks are required for values of K from 3 to 11.

4.5.3.4 Spectral Amplitude Quantisation

When quantising spectral amplitude information the number of amplitudes to be quantised is dependent upon the pitch length P of the pitch cycle. The number of amplitudes N present is given by

$$N = \frac{P}{f_s} \times f_c = 0.4625 \times 3700$$

where $f_s$ is the sampling frequency of 8000 Hz and $f_c$ is the cut off frequency typically 3700 Hz.

The SB-LPC uses a amplitude peak picking algorithm to convert the vector to fixed length [34]. This process selects and quantise the perceptually important amplitudes. This algorithm was utilised in the PS SB-LPC in order to quantise the spectral amplitudes. This algorithm is described further in Section 7.3.1.

Peak picking will pick amplitudes from different areas of the speech spectrum during different pitch cycle waveforms during the frame. If the frequency of the pitch cycles is varying over the frame the shape vector would be attempting to model the amplitudes of different frequencies during different sections of the frame. Rather than using the respective LP spectrum for each PCW contained in a frame, amplitudes are selected using one LP spectrum per frame. The spectral amplitudes are selected from the LSF parameters found over a frame from

$$x^j = \sum_{k=0}^{K-1} w_k \hat{y}_k^j$$

$x^j$ is the $j^{th}$ LSF. $\hat{y}_k^j$ is the $j^{th}$ quantised LSF from the $k^{th}$ PCW. $w_k$ is the weighting of the $k^{th}$ PCW. The weighting function is defined as

$$w(k) = \frac{\sqrt{e_k}}{\sum_{i=0}^{K-1} \sqrt{e_i}}$$
where $\hat{e}_k$ is the quantised energy from the $k^{th}$ pitch cycle. This single set of LSF values is found for all PCW contained within the frame and then peak picking is performed for each PCW. If the PCW has less than fourteen harmonics all of them are selected. If not then the first two amplitudes are selected as LP modeling can be poor at lower frequencies. Next three amplitudes are selected around the four largest peaks in the LP spectrum for the frame. This results in fourteen spectral amplitudes being selected for each PCW. The peak picking algorithm was implemented in the PS SB-LPC. For each frame two sets of spectral amplitudes were quantised using JQI as described in the LSF quantiser. The 14 edge element amplitudes were jointly quantised using a 24-bit 3 stage MSVQ. In addition to the two edge vectors an an optimum shape vector was calculated. This shape vector was quantised using a 6-bit vector quantiser trained with the LBG algorithm.

4.5.4 Coder Evaluation

This results in a PS sinusoidal coder operating at 4.8 kbps. This coder segments the speech into constituent pitch cycles and then carries out parameter analysis on these cycles. The pitch cycle information is quantised and transmitted to the decoder. Because of errors in the pitch cycle detection and segmentation process the analysed cycle sizes have to be altered so pitch size artefacts are removed from the speech coding process.

Although this coder at certain sections clearly improves over the TS method as shown in Figures 4.7 and 4.8 when evaluated in its unquantised mode over several seconds of speech, artefacts were present that degraded the synthetic speech quality. This was believed to be mainly caused by errors in the analysis process namely pitch cycle detection, pitch cycle size smoothing and voicing estimation algorithms.

When the speech information was quantised there was further distortion, this was believed to be caused by the spectral amplitude quantisation method of Section 4.5.3.4. Also the artefacts present from the analysis process were boosted by the quantisation which further degraded the synthetic speech quality.
4.6 Concluding Remarks

The background to sinusoidal speech coding and current sinusoidal speech coders has been discussed. The motivations behind PS coding of speech has been demonstrated and several PS low bit rate speech coders have been introduced. The PS SB-LPC which forms the basis of this project has been described in detail. This coder produced high quality synthetic speech at most speech sections however it was not robust to all input speech categories and consequently perceptually displeasing artefacts were present in the synthetic speech which limited the final speech quality. The next chapter will detail the investigation and possible solutions found to improve this coder.
Chapter 5

Classifier Based Voicing

5.1 Introduction

The PS SB-LPC was introduced in chapter 4, this coder can produce high quality synthetic unquantised speech, however occasional artefacts are produced which limit the speech quality. It is believed that these problems are mainly caused by deficiencies in the pitch and voicing analysis algorithms. These algorithms utilise a number of heuristic measures to improve their performance. It is considered that these measures cannot accommodate all speech inputs thus causing a degradation in algorithm ability.

The first section of this chapter describes the actions taken to find and eliminate these artefacts in order to raise the quality of the synthetic speech produced by the PS SB-LPC. The second section and greater part of this chapter is concerned with the design of a voicing classifier intended to improve voicing decisions in the PS SB-LPC. Both of these parts utilise a graphical user interface (GUI) based tool called the Bit Stream Editor (BSE). This tool allows the user to manually set and save to file analysis values at the encoder of the PS SB-LPC and evaluate their effect upon the decoded speech.

5.2 Bit Stream Editor

The Bit Stream Editor has been developed with the Tcl and Tk programming languages and is based on Snack Toolkit [55] and previous work on the SB-LPC [56]. Tcl is a
scripting language and is used extensively in GUI and embedded applications, Tk is an open source library of basic elements for building a GUI; the combination of Tcl and Tk GUI toolkit is known as Tcl/Tk. Snack is a Tcl based tool which also allows executables to be called in the C programming language. Snack has functions for basic sound handling such as playback, recording, file and socket I/O. It has callable functions which allows users to open a input speech file, for example in several audio formats, from disk storage and then view and listen to the speech file. A summary of how Snack and Tcl/Tk elements slot into the BSE programming environment is shown in Figure 5.1.

![Figure 5.1: Bit Stream Editor programming environment](image)

A screen shot of the BSE is shown as Figure 5.2. The user options at the far left are basic functions for dealing with the waveform signal such as open, copy, save and zoom, etc. The call encoder button executes the encoder with initial pitch positions and voicing levels. Functions have been added to Snack so that these values can be loaded onto the screen and edited manually with electronic mouse operation. The PS Encoder mod and PS Decoder mod buttons re-encode and decode with these manually edited pitch and voicing files, producing the output speech shown at the bottom of the plot. A playback toolbar is built into the BSE for the playback of input/output speech. The general structure of the BSE operation is shown as Figure 5.3.
5.3. Utilisation of Bit Stream Editor

It is believed that the most likely cause of the artefacts that limit the quality of synthetic speech produced by the PS SB-LPC are [40]:

- Pitch cycle detection - occasional errors in the pitch cycle detection process results in incorrect pitch sizes being analysed producing pops and roughness in the voiced synthetic speech. This requires post-processing to be carried out as described in Section 4.5.1.1. However these post-processing routines do not eliminate all pitch cycle size errors.

- Voicing estimator - the voicing estimator is believed to incorrectly select a voicing level which corresponds to a frequency level higher than should be expected for the speech waveform at certain sections. This results in synthetic speech that does not have the same perceptual quality as the original.

The BSE can be used to determine the if pitch and voicing estimation errors
5.3. Utilisation of Bit Stream Editor

Figure 5.3: General structure of BSE operation
are definitely causing the majority of errors heard in the synthetic speech. The aim therefore is to use the BSE to manually set the pitch and voicing values on the screen of the interface through mouse operation. This allows the maximum speech quality of the speech coder to be determined. The technique of interfacing a speech coder to a GUI analysis tool has not been presented in any relevant literature as far as the author is aware. This technique is an efficient and modern approach to improving speech coder performance. Its operation is summarised in flow chart form as Figure 5.4 and is described in Sections 5.3.1 and 5.3.2.

![Flow chart for removing speech artefacts in BSE](image)

**Figure 5.4**: Process of removing speech artefacts in BSE

### 5.3.1 Pitch Cycle Position Editing

All operations take place on 16 kHz PCM files from the NTT database [57] which have been down sampled to 8 kHz to allow for use in the PS SB-LPC. The PS SB-LPC encoder is executed without any smoothing algorithm in place and the pitch
5.3. Utilisation of Bit Stream Editor

cycle sizes written to a text file. These pitch cycle positions can then be plotted onto the BSE window, adjusted and then the encoder/decoder executed with these new positions and the decoded speech evaluated perceptually on the BSE. The Modified

![Flow chart of pitch cycle position editing](image)

**Figure 5.5: Flow chart of pitch cycle position editing**

Time Envelope (MTE) signal used by the Trapezoidal Search can be viewed on the Editor as a guide to estimate PCW positions as it is time aligned to the encoded speech. Major modifications to Snack, the BSE and the PS SB-LPC were needed to achieve these goals. This operation is summarised in Figure 5.5. Figure 5.6 shows the
Figure 5.6: BSE with pitch error (a) Pitch sizes, (b) Pitch cycle positions, (c) Input Speech and (d) decoded speech with artefact

Figure 5.7: BSE with pitch error (a) Pitch sizes, (b) Pitch cycle positions, (c) Input Speech and (d) decoded speech with artefact removed
5.3. Utilisation of Bit Stream Editor

occasional pitch cycle detection resulting in harmonic damage to the synthetic speech, this is perceptually displeasing and easily noticed by the listener. It can be seen at (a) and (b) that the pitch sizes variation is not smooth and a sharp change occurs which does not correspond to the input speech which is expected to have a regular pitch variation at this point.

![Figure 5.8: Original speech (top), MTE (middle) and original speech residual (bottom)](image)

Using the BSE the pitch position was manually adjusted to correspond to the neighbouring cycles. This is illustrated in Figure 5.7 where the pitch cycle positions have been manually adjusted and the speech artefact removed. By listening to the output speech on the BSE and manually adjusting all the pitch positions which were causing pitch speech artefacts it was possible to remove almost all errors caused by incorrect pitch cycle positions.

It was found on the BSE that at certain sections where there is little excitation and only resonance from the vocal tract the MTE does not produce a signal with a clear pitch structure. This is illustrated in Figure 5.8 at points (a) and (b), when the MTE is processed by the Trapezoidal Search routine irregular pitch sizes are produced to which post-processing smoothing is applied in the PS SB-LPC.
5.3. **Utilisation of Bit Stream Editor**

5.3.2 **Voice Level Editing**

The operation of altering the voicing levels in the PS SB-LPC and then viewing/listening to the effect on the decoded speech is very similar to the operation of pitch position editing. This operation is surmised in Figure 5.9. The Snack toolkit upon which the BSE is based provides support for visualisation of speech in the frequency domain. This is illustrated in Figure 5.10, part (a) shows the voicing levels encoded by the PS SB-LPC, these voicing levels have been found from the input speech shown in (c). Part (d) shows the frequency spectrum of the input speech at point 1 in (c). By modifying the BSE the voicing levels in (a) can be manually adjusted through a simple mouse movement or in screen (d) in the frequency domain any changes to the voicing level made in (d) will be updated in (a). This gives great flexibility as the voicing levels can be set on the BSE both perceptually and in the frequency domain. Snack also allows different window types and Fourier Transform lengths to be used which can alter the views in the speech spectrum plot of Figure 5.10 (d). It was found using the BSE that the current existing voicing algorithm which has been described in Sections 4.5.2.4 and 4.5.2.6 made excellent voiced/unvoiced decisions. However the soft decision voicing levels were generally set too high and at many sections the synthetic speech produced was over voiced and as a result perceptually sounded harsh.

This over voicing is caused by variations in the band-limited peakiness signal. Initially when each signal is band-limited the highest peakiness value from each of the seven cut off frequencies is chosen as the voicing level for that PCW. Figure 5.11 (right) shows three PCW of speech. These PCWs have very similar voicing parameters but the centre PCW is considered to be a voicing level calculation of seven from the voicing determination algorithm currently used by the PS SB-LPC. Figure 5.11 (left) shows the band-limited peakiness values for the three PCWs.

From this figure the first and third PCWs have a maximum peakiness value at a level of two, however although the second PCW varies little over a cut off of two by using the maximum peakiness value the PCW is declared fully voiced. This change in voicing level is not due to the characteristics of the signal but is caused by the voicing calculation itself. Although this variation occurs infrequently to overcome these slight variations
5.3. Utilisation of Bit Stream Editor

Figure 5.9: Flow chart of voice level editing
that can occur in the peakiness signal a small amount of historical bias was originally introduced.

The peakiness value for the band cutoff frequency selected as the previous PCW voicing level is multiplied by a factor of $1 + \epsilon$. A value of $\epsilon$ of 0.07 was found and set experimentally in [40]. If applied to the second PCW in Figure 5.11 (right) it will bias this PCW to have a voicing level of 2. However this solution may remove problems caused by occasional irregular variations in the band limited peakiness but it limits natural variation that can occur in voicing of the speech signal at some sections.

This is illustrated in Figure 5.12 which shows the voicing levels set on the BSE known here as reference voicing and those from the PS SB-LPC through the usual voicing estimation algorithm described in Section 4.5.2.6. For this section of input speech the levels from the PS SB-LPC match closely those of the ideal reference at the start of the speech waveform, however towards the end of the waveform the levels in the PS SB-LPC do not fall to those of the reference. Inspection of the band-limited peakiness
5.3. Utilisation of Bit Stream Editor

Figure 5.11: Band-limited peakiness values for three PCW (left), shown (right) with original encoded voicing level

Figure 5.12: Encoded voicing levels in the PS SB-LPC against reference voicing manually set on the BSE
values determined that this was due to the addition of historical bias, ɛ.

As a result the synthetic speech produced at such sections sounds perceptually over synthetic and does not reflect the natural evolution of the speech waveform. In order to remove the heuristic tuning of the voicing levels which cannot accommodate every input situation we aim to improve upon the voicing level accuracy by using the manually tuned BSE reference voicing levels in a database and utilise vector quantisation techniques to make voicing level decisions. This will be discussed further in the next section.

5.4 Voicing Estimation

5.4.1 Introduction

Standard sinusoidal coders such as MBE and MELP extract parameters at regular intervals; parameter estimation is achieved using the speech waveform falling under an analysis window centered on an analysis point. This procedure assumes the speech to be stationary during the speech segment under analysis. However speech is non-stationary at transitional sections of speech such as onsets, offsets and plosives, which although they only account for a small percentage of speech they are very important perceptually. Therefore it can be considered that a disadvantage of TS coders such as MELP and MBE is that they smooth transitional sections of speech resulting in the loss of fine detail.

PS coders such as the PS SB-LPC operate on a per cycle basis and therefore do not smooth transitional sections as they have a shorter analysis window which should result in superior performance over TS coders at such sections. Figure 5.13 demonstrates these differences when applied to the voicing classification of input speech. The TS coder will extract parameters at points 1 and 2, which may be incorrect in its classification (d) of the voicing content of the speech signal at points 3 and 4. At point 3 the segment is too short compared to the length of the window and at point 4 the TS method does not provide the necessary time accuracy at transitions. The PS classification (c) unlike the TS method should correctly classify the first and third speech segments as voiced.
5.4. Voicing Estimation

5.4.2 Classic Voicing Estimation Methods

After speech is declared as unvoiced or voiced as described in Section 3.6 using a hard voicing decision, voiced speech is further classified to estimate its actual frequency content. To make this soft decision classic voicing methods such as those in the MBE and the SB-LPC typically rely on the comparison of 2 functions:

- A voicing function computed in the speech spectrum with a value per harmonic or frequency band
- A threshold function computed as heuristic function of several speech parameters.

The voicing threshold is necessary as the performance of the voicing function is not sufficient [34]. Several speech parameters are used to give an indication of voicing. The main concern with this method is how to generate a threshold function based on these parameters. The next section describes how this is carried out classical sinusoidal speech coders.
5.4. Voicing Estimation

5.4.2.1 MBE Mixed Voicing

In the MBE coder harmonic voicing is estimated by comparing the error of a synthetic voiced spectrum $\hat{S}(m, w_0)$ with respect to the speech spectrum $S(m)$ and comparing it against a threshold function for each harmonic band. Using $\hat{S}(m, w_0)$ a voicing measure is computed for each band on which the voicing decision is made. These bands do not have to be single harmonic bands, they can cover a number of harmonic bands. The MBE splits the spectrum in groups of three harmonics and performs the voicing decisions on these groups.

$$D_k = \frac{\sum_{m=a_k}^{b_k} |S(m) - \hat{S}(m, w_0)|^2}{\sum_{m=a_k}^{b_k} |S(m)|^2}$$

(5.1)

where $w_0$ is the selected fundamental frequency and $a_k$ and $b_k$ are the lower and upper boundaries of the decision bands. Each band is declared voiced if its voicing measure is above the threshold function, unvoiced otherwise. The threshold is defined as $\Delta_k(w_0)$

$$\Delta_k(w_0) = (\alpha + \beta w_0)(1.0 - \epsilon(k - 1)w_0)M(E_0, E_{av}, E_{min}, E_{max})$$

(5.2)

where $\alpha = 0.35$, $\beta = 0.557$ and $\epsilon = 0.4775$ are the factors that give good subjective quality and

$$M(E_0, E_{av}, E_{min}, E_{max}) = \begin{cases} 
0.5; & E_{av} < 200 \\
\frac{(E_0 + E_{min})(2E_0 + E_{max})}{(E_0 + \mu E_{max})(E_0 + E_{max})}; & E_{av} \geq 200 \text{ and } E_{min} < \mu E_{max} \\
1.0; & \text{otherwise} 
\end{cases}$$

(5.3)

is the adaption factor that controls the decision threshold for voicing decisions. A favourable value for $\mu$ is 0.0075. Parameter $E_0$ is the energy of the current frame and the parameters $E_{av}$, $E_{max}$ and $E_{min}$ correspond to the local average energy, the local maximum energy and the local minimum energy respectively. These three speech parameters are updated every frame according to [46].
5.4.2.2 MELP Mixed Voicing

The voicing decision in MELP is performed using time domain techniques on bandpass filtered versions of the original speech. The original speech is separated into 5-sub bands using 6\textsuperscript{th} order Butterworth filter, with pass-bands of 0-500, 500-1000, 1000-2000, 2000-3000, and 3000-4000 Hz \cite{44}.

The normalised correlation is then computed at the pitch value \( P \) for the first band as well as the range \( P-5, P+5 \). The maximum of these correlations is then used as the bandpass voicing strength for the first band and the corresponding lag is saved for use in the computation of the bandpass voicing for the remaining bands. The bandpass voicing strength for the other bands is computed again using the normalised autocorrelation at the lag chosen for the first band on the bandpass filtered signal, and also on the time envelope of that signal. The maximum of these two correlations is then taken as the bandpass voicing strength of the considered band.

These bandpass voicing strengths \( V_{bp_i} \) with \( i \) equal to 1,...,5 are then biased using the peakiness of the signal. If the signal is very peaky \( (P_k > 1.6) \), \( V_{bp_i} \) for \( i = 1, 2, 3 \) are forced to 1. If it is moderately peaky \( (P_k > 1.34) \), \( V_{bp_i} \) is forced to 1.0. Finally the voicing decision for each band is made using \( V_{bp_i} \):

- If \( V_{bp_1} \leq 0.6 \) all bands are declared unvoiced
- If \( V_{bp_1} > 0.6 \), the first band is declared as voiced and each band \( i \) is set to voiced if \( V_{bp_i} > 0.6 \), unvoiced otherwise
- A voicing pattern of the five bands of 10001 is not allowed and is replaced by 10000 where 0 indicates unvoiced and 1 voiced.

The voicing decision itself makes use of two parameters; the normalised autocorrelation and the peakiness of the signal. This voicing decision is shown as Figure 5.14. Although providing good voicing indication, there are cases when both these parameters fail therefore more parameters are needed for reliable voicing determination.
5.4. Voicing Estimation

5.4.2.3 SB-LPC Mixed Voicing

Coders such as the MBE use a voicing decision for each harmonic or group of harmonics (typically 2 or 3), the SB-LPC coder assumes all bands to be voiced from DC to a certain cutoff frequency and unvoiced above this cutoff frequency. This has the advantage of requiring only a small number of bits to represent the cutoff frequency, 3 bits usually being sufficient. This represents a large saving over the MBE approach which requires up to 12 bits.

The cutoff frequency decision is made by considering the voicing likelihood for each individual harmonic. A voiced band should have a spectral shape similar to the spectral shape of the window used, prior to Fourier Transformation. Unvoiced bands will be random in nature. The voicing likelihood of each band is measured as the normalised correlation between the considered harmonic band and the spectral shape of the window positioned on the harmonic location. The voicing likelihood $V(l)$ for the $l^{th}$ harmonic

![Figure 5.14: Voicing classification technique in MELP coder](image-url)
5.4. Voicing Estimation

is given by

\[ V(l) = \frac{\left( \sum_{m=1}^{b_h} S(m)W(\frac{2\pi}{N}m - lw_{0}) \right)^2}{\sum_{a_l} W^2(\frac{2\pi}{N}m - lw_{0}) \sum_{a_l} S^2(m)} \]  

(5.4)

where \( S \) is the Fourier Transform of the speech and \( W \) of the analysis window. The value of \( V(l) \) is between 0.0 and 1.0 respectively corresponding to fully unvoiced and unvoiced cases. This value is then compared to a threshold function \( T(l) \) for each individual harmonic. This threshold calculation is the most important stage during Split Band voicing estimation [34].

The value of \( T(l) \) is determined by taking several factors into account. Firstly the lower harmonics are more likely to be voiced so the threshold value is lower for the lower harmonics. Secondly a harmonic is more likely to be unvoiced if unvoiced in the previous frame, so the threshold is raised for harmonics that were previously unvoiced. Thirdly the harmonics are more likely to be voiced if the hard decision voicing metric indicated a voiced signal, therefore the threshold is lowered.

The voicing threshold is biased by using a range of voicing parameters as described in Section 3.6 such as zero crossing and autocorrelation. Thresholds are set for each of these parameters and if triggered the voicing threshold function is biased towards voiced or unvoiced. The pitch value is also used to bias the voicing threshold function. An example of voicing likelihood and threshold function is given in Figure 5.15.

Using a limited number of speech characteristics for the threshold computation does not lead to good voicing determination. In the MBE energy alone is not a reliable enough voicing indication as there can be high energy unvoiced speech sections. In MELP the peakiness factor is not entirely reliable, single peaks can lead to high peakiness, likewise for autocorrelation; in the case of pitch variations, normalised autocorrelation may be quite low when the speech is voiced.

By increasing the number of parameters and other speech characteristics, the SB-LPC improves upon these two coders when it comes to finding a suitable threshold function. However even in the SB-LPC, this threshold function has to be tuned, through trial and error of several speech parameters which can be difficult and unreliable as every
5.4. Voicing Estimation

![Voicing Likelihood, Threshold Function, Speech Spectrum, and Cutoff Frequency](image)

Figure 5.15: Original Speech spectrum with voicing likelihood, threshold function and cutoff frequency [34]

new filtering/noise condition calls for retraining. Given that the voicing has a large impact on speech quality we aim to apply a systematic approach to overcome these disadvantages.

This approach uses the BSE to generate reference hand marked voicing files for a speech database. This database of speech information can be utilised:

- A classifier is trained using these and a number of selected speech parameters
- Vector Quantisation techniques are used to cluster speech parameters and associate each cluster with a threshold function.
- Generate a threshold function computed as giving best classification in the cluster according to the training database.
- The PS SB-LPC encoder stores a list of clusters and a voicing threshold function for each cluster. It compares speech parameters for current cycle to stored clusters and chooses a threshold function associated with best matching cluster.

This technique should be easy to adapt for use in various filtering/noise conditions. As
5.4. Voicing Estimation

an example if this approach was carried out and implemented in the MELP, the result may be as shown as Figure 5.16, this can be compared to the original MELP approach previously shown in Figure 5.14.

![Voicing Estimation Diagram](image)

**Figure 5.16:** Proposed classification technique if applied to MELP voicing

The next section will describe this method in detail where we employ a Codebook classifier technique; during normal PS SB-LPC operation, if the PCW is declared as voiced the peakiness values at each frequency cutoff level are used by the classifier, which has been trained to give the best voicing decision.

5.4.3 PS SB-LPC Voicing Classifier

Using the BSE, for each PCW several speech parameters are stored along with their corresponding reference voicing decision from the manually tuned results and the corresponding peakiness values at each frequency cutoff - this vector is known here as the voicing vector. These voicing vectors therefore contain the voicing information for each PCW, PCWs which have similar speech parameters usually have a similar evolution of peakiness over the seven cutoff frequencies.

Vector quantisation techniques are used to determine which of the N1 cycles over the
5.4. Voicing Estimation

range of the training database are the closest. Over the length of the training database, 9300 cycles, the N1 closest PCW in terms of the speech parameters are grouped together by finding the minimum distortion between the current test vector and all vectors in the codebook

\[
(k_1, k_2, ..., k_{N1}) = \arg \min \sum_{j=1}^{N1} \left( \sum_{i=0}^{L-1} (X(i) - C_{k,j}(i))^2 \cdot W(i) \right)
\]  

(5.5)

where \( W \) is the training weights applied to the speech parameters, the number of weights used is given by \( L \) here a value of 9, these are defined in Table 5.1. \( X \) is the current test vector and \( C \) are the vectors in the database.

<table>
<thead>
<tr>
<th>Number</th>
<th>Speech Parameters</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Energy to Peak Energy Ratio</td>
<td>1.5</td>
</tr>
<tr>
<td>2</td>
<td>Peakiness</td>
<td>1.6</td>
</tr>
<tr>
<td>3</td>
<td>Correlation</td>
<td>1.4</td>
</tr>
<tr>
<td>4</td>
<td>Zero Crossing Ratio</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>Low Band to Full Band Energy Ratio</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>Pre-Emphasis Energy Ratio</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>Previous Voicing</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>( P_1 )</td>
<td>2.1</td>
</tr>
<tr>
<td>9</td>
<td>( P_{\text{max}} - P_{\text{min}} )</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.1: Weights used in training procedure

In Table 5.1 \( P_1 \) is the peakiness value at the cutoff level of one and \( P_{\text{max}} \) and \( P_{\text{min}} \) are the maximum and minimum peakiness values of the seven cut off levels of each pitch cycle. The remaining weights are those parameters described in Section 3.6. The previous voicing value was normalised by its maximum size of 7 to ensure it had similar values to the other weights. These weighting values were found empirically after extensive trial and error.

Once N1 closest vectors have been found from (5.5) a set of thresholds for the N1 must be found which when input to a algorithm will produce the closest match to the
reference voicing. The first step is to determine a good value for N1. This illustrated in Figure 5.17 which shows the band-limited peakiness and reference voicing (manual voicing) for a PCW.

![Figure 5.17: Bandlimited peakiness, reference voicing and threshold functions](image)

Also shown are some suggested threshold functions. The aim here is to find a threshold function which will produce a voicing level of 5 - the reference level. In this example threshold function one maybe set too low but threshold functions two and three may both be suitable, many more threshold functions will be suitable also over the number of combinations. As there can therefore, be considerable variation between suitable threshold functions for each of the voicing vectors, N1 voicing vectors were used to generate the one threshold function. It was determined experimentally that N1 equal to 25 was a good value to use.

A step size δ is found from \((P_{max} - P_{min})/7\) of the N1 vectors. The value of the band-limited peakiness at each of the seven cutoff frequencies \(P(l)\) is then compared to a threshold function which corresponds to the current threshold at that cutoff frequency. Over every iteration of δ the value of i which produces the greatest value of matching function \(M(i)\) is considered to be the calculated voicing. This matching function \(M(i)\) is
5.4. Voicing Estimation

given by:

\[ M(i) = \sum_{i=1}^{7} (P(l) - T(l))V_iB(l) \]  

For any given voicing level \( i \) individual voicing decisions \( V_i(l) \) will have +1 (i.e. voiced) up to the cut off \( f_c(i) \) and -1 for the higher values (i.e. unvoiced). The matching function is computed at each iteration for every possible voicing step \( i \). For any given \( i \), each voicing level computed correctly i.e. the product \( (P(l) - T(l))V_i(l) \) is positive, will contribute to the total sum \( M(i) \) proportionally to the difference between the band-limited peakiness value at \( P(l) \) and the threshold \( T(l) \). Each incorrect voicing level will decrease the total sum. The weighting \( B(l) \) is usually set to 1.0 when unvoiced \( (T(l) > P(l)) \) and higher for voiced, as it is more important perceptually to get the higher voiced levels correctly set.

The matching function \( M \) from (5.6) returns one of the voicing levels (1 - 7), this calculated voicing known here as \( V_{\text{calc}} \) for each threshold iteration is compared to the reference voicing \( V_{\text{ref}} \) (set in the BSE of Section 5.3.2) for each of the \( N_1 \) vectors. A score \( V_{\text{score}} \) is found over all \( N_1 \) vectors using

\[ V_{\text{score}} = \sum_{i=0}^{N_1-1} (V_{\text{ref}}(i) - V_{\text{calc}}(i))^2 \]  

The lowest value of \( V_{\text{score}} \) corresponds to the best thresholds for these \( N_1 \) vectors as it indicates that the calculated voicing is closest to the reference voicing for each of the \( N_1 \). The parameters associated with these \( N_1 \) vectors and their associated thresholds at each of the seven cut off frequencies are then written to a training database. During normal operation of the PS SB-LPC encoder for each PCW the closest \( N_2 \) vectors in the training database, \( N_2 \) equal to 25, are found using the same weights as in the training procedure. The associated thresholds values are then averaged at each cutoff frequency and input into (5.6) to determine the voicing level for that PCW.

We compared the results from the PS SB-LPC classifier method to those employed by the 2.4 kbps MELP method of Section 5.4.2.2 at a 8 kHz sampling rate. The MELP coder makes a voicing decision for each 200 sample frame of speech at the encoder and then interpolates voicing across the decoded frame. The speech files used were
5.4. Voicing Estimation

Figure 5.18: Voicing levels in the MELP and PS SB-LPC against reference voicing not included in the training database. A comparison of the manually set voicing on the BSE (reference) is shown against the PS SB-LPC classifier and MELP voicing in Figure 5.18.

As shown in the figure at the waveform onset and offset where the pitch cycles are changing more rapidly in the time domain, the superior time resolution of the PS SB-LPC more closely follows the reference voicing levels set in the frequency domain. In addition at the waveform centre where there are no PS issues the PS SB-LPC still gives a more accurate result. It was generally found that the classifier method returned a voicing level within one voicing level of the reference voicing.

A comparison was made using cycle sample lengths of the encoded voiced (fully and mixed) and unvoiced decisions of each cycle (identified by Trapezoidal Search) of the two coders. The results are shown as Table 2.2 with silences excluded. The distinction between silence and non silence was based on an energy cutoff level of five. As can be seen the performances of the MELP and PS SB-LPC are comparable when determining voiced but very different for unvoiced speech cycles.

This primarily reflected the effect at onsets and offsets where a MELP speech frame
5.4. Voicing Estimation

<table>
<thead>
<tr>
<th>Coder</th>
<th>Speaker</th>
<th>Type</th>
<th>V</th>
<th>UV</th>
</tr>
</thead>
<tbody>
<tr>
<td>MELP</td>
<td>F1</td>
<td>V</td>
<td>97.94</td>
<td>48.96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>2.06</td>
<td>51.04</td>
</tr>
<tr>
<td></td>
<td>M1</td>
<td>V</td>
<td>97.56</td>
<td>32.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>2.44</td>
<td>67.98</td>
</tr>
<tr>
<td></td>
<td>F2</td>
<td>V</td>
<td>97.15</td>
<td>40.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>2.85</td>
<td>59.50</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>V</td>
<td>96.70</td>
<td>3.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>2.75</td>
<td>61.57</td>
</tr>
<tr>
<td>PS SB-LPC</td>
<td>F1</td>
<td>V</td>
<td>96.70</td>
<td>3.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>3.30</td>
<td>96.20</td>
</tr>
<tr>
<td></td>
<td>M1</td>
<td>V</td>
<td>96.53</td>
<td>4.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>3.47</td>
<td>95.87</td>
</tr>
<tr>
<td></td>
<td>F2</td>
<td>V</td>
<td>96.98</td>
<td>3.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>3.32</td>
<td>96.13</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>V</td>
<td>96.80</td>
<td>3.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UV</td>
<td>3.20</td>
<td>96.09</td>
</tr>
</tbody>
</table>

Table 5.2: Encoded voicing cycle comparison of MELP and PS SB-LPC against reference voicing. V/UV comparison is made on percentage of cycles according to sample length.

contained only one or a few voiced cycles and a large unvoiced component, consequently the MELP decision was to declare such a frame and therefore its constituent cycles as voiced. It is possible that the TS nature of the MELP coder may have forced the designers to bias the voicing decisions towards voiced as it is usually much better perceptually to declare a unvoiced cycle as voiced rather than to declare a voiced cycle as unvoiced.

A good separation of voiced and unvoiced cycles of speech has been achieved. It may be possible to use such an algorithm in a speech activity detection system for end point detection. Endpoint detection is the detection of speech in non-speech events and background noise, it is important in automatic speech recognition (ASR) and speaker
5.5. Concluding Remarks

In this chapter a novel GUI tool has further developed and utilised to determine the cause and effect of distortion in a PS sinusoidal speech coder. A novel technique for accurately measuring the voicing content of a speech signal has also been introduced. This has been achieved by the measurement of phase-spread information contained within individual pitch cycles. The experimental results show that the methods introduced compare favourably with a standard speech coder that operates time synchronously. These methods can be utilised not only by PS but also TS coders to make more accurate decisions on the voicing content of their speech frames.
Chapter 6

Pitch Cycle Detection

6.1 Introduction

In order to determine the pitch values contained within a frame of input speech, PS coders must segment the input speech into PCWs. Previously it has been shown that the PS SB-LPC uses an algorithm known as the Trapezoidal Search to carry out this task, however this results in pitch errors which subsequently have to be post processed. This Chapter will discuss alternative methods of pitch cycle detection and segmentation with the aim of minimising the number of pitch errors.

6.2 Cycle Size Post Processing

The Trapezoidal Search algorithm correctly segments input speech into PCWs the majority of the time but the effects of pitch size errors and jitter degrade the quality of the synthetic speech produced. These errors typically occur at speech sections where the input speech residual or consequently MTE does not contain a clear pitch structure, such as shown in Figure 6.1. Several pitch size glitches can be seen in the analysed cycle sizes which have to be smoothed by post-processing as discussed in Section 4.5.1.1.

During analysis, PCW following a cycle detection error are not modified. The erroneous cycle sizes are adjusted and analysis size expanded to the adjusted cycle length. Three
rules were developed to identify and rectify changes in pitch cycle errors that are likely to be caused by cycle detection errors; (1) Single cycles that are significantly larger or smaller than cycles either side of them, (2) cycle size errors in regions where the surrounding cycle sizes are constant or smoothly evolving and (3) cycles that are only one sample shorter or longer in length than their neighbouring cycles. These smoothing algorithms remove the great majority of the errors.

It was considered therefore to improve these post processing routines so the remaining artefacts can be removed, however this was not possible for two reasons. Firstly, the smoothing algorithms themselves relied on values found experimentally which gave the best results overall.

As an example consider (6.1) which was first introduced in Section 4.5.1.1 and is presented here again for completeness. This equation fulfils (1) above where $S_i$ is the size

\[ S_i = \frac{S_{i-1} + S_{i+1}}{2} \quad \text{if} \quad \begin{cases} (S_i > \alpha S_{i+1} \text{ and } S_i > \alpha S_{i-1}) \\ \text{or } (S_i > \beta S_{i+1} \text{ and } S_i > \beta S_{i-1}) \end{cases} \]

(6.1)
6.3 Combined Sinusoidal and Waveform Coding

of the cycle to be adjusted. This equation relies on two experimentally found perceptual measures $\alpha = 1.1$ and $\beta = 0.83$. These perceptual measures cannot account for every input condition and so are not reliable over many speech cycles. Secondly, it is not always possible to rely on neighbouring cycle sizes as they too may have been found in error.

Clearly to improve the pitch detection and segmentation processes, new techniques must be found. Some other methods which have been investigated and implemented will now be discussed.

6.3 Combined Sinusoidal and Waveform Coding

Multi-mode coders which combine a CELP based coder for transitional speech sections and a sinusoidal based coder for steady state sections were introduced in Section 2.3.3.1. It was considered therefore to use a CELP based system at speech sections where the PS SB-LPC is currently not of high enough quality and use the PS SB-LPC Trapezoidal Search for other speech sections where there are limited pitch detection errors. Figure 6.2 shows that at transitional speech sections (a) and (b) although in the input

![Figure 6.2: A section of female input speech (top), input speech residual (middle) and Modified Time Envelope (bottom).](image)
6.3. Combined Sinusoidal and Waveform Coding

speech there are clear pitch periods, due to the lack of clear excitation in the residual the MTE lacks a clear pitch structure. It could be considered therefore to switch to a CELP based model at these sections thus improving coder performance.

The PS-SBLPC carries out smoothing of the pitch sizes for all sections of speech, in order to correct pitch detection errors and pitch size jitter. The encoder analyses the speech according to the smoothed sizes and passes the smoothed sizes to the decoder. However the encoder still segments the speech according to the non-smoothed sizes. This is illustrated in Figure 6.3 which shows that after three speech frames the decoder is seven samples away from the encoder. After the first frame the input and synthetic speech are aligned. Because of cycle size jitter in the second frame smoothing is employed and the encoder and decoder become unsynchronised and at the start of the third frame are five samples apart. Further size jitter in the third frame requires more smoothing resulting in further loss of synchronisation.

![Diagram](image)

**Figure 6.3:** Loss of synchronisation between encoder and decoder due to cycle size smoothing

Integration of a CELP based model is a considerable challenge. Sinusoidal coders do not transmit phase information and time alignment between original and synthesised
speech is not present. AbS coders preserve waveform similarity and direct switching between these two types degrades speech quality due to phase misalignment [60] and [12]. Considerable research has been carried out into solving this problem. The loss of synchronisation between encoder and decoder meant that it would be even more complex to integrate a CELP based system into the PS SB-LPC. In addition cycle size errors also occur at steady state speech sections as shown in Figure 6.1. Therefore other solutions were sought to remove the artefacts present in the synthesised speech of the PS SB-LPC, the next section describes these methods.

6.4 Group Delay Method

This method suggested by Yegnanarayana [61] identifies a significant instant of excitation within a frame of speech. The Group Delay based method is a popular method for determination of Glottal Closure Instant (GCI), it determines the instants of glottal closure in voiced speech by using the Group Delay function (GD). Firstly, consider a unit sample sequence with a delay \( \tau \), taking the Fourier transform gives \( e^{-j\omega \tau} \). Hence the Fourier Transform Phase Function is given by \( \phi(\omega) = -\omega \tau \). The negative derivative of the phase function, known as the group delay is \( -\phi'(\omega) = \tau \).

Thus the value of the slope of the phase is constant and is equal to the value of the unit sample delay in the time domain. If we place a unit impulse in an analysis window and the window is moved from left to the right, the value of the GD will linearly increase from left to right with respect to the position of the analysis window. Figure 6.4 shows that the average group delay is zero when the unit impulse is at the centre of a 40 sample window.

6.4.1 Algorithm Specifics

Direct computation of the phase spectrum using the real and imaginary parts of the DFT of the signal results in phase values that are wrapped around the edge \(-\pi\) and \(\pi\). Therefore in order to calculate the phase slope the phase spectrum must be first unwrapped. This can be complex and the accuracy of computation depends on the
6.4. Group Delay Method

Windowing of the signal. Oppenheim and Schafer [12] have shown that the GD function of a signal can be calculated directly from the windowed signal. If we take $F(w)$ and $N(w)$ as the Fourier Transforms of $f(n)$ and $n f(n)$ then the group delay is given by:

$$\phi'(w) = \tau(w) = (F_r(w) N_r(w) + F_i(w) N_i(w)) / (F_r(w)^2 + F_i(w)^2) \quad (6.2)$$

where $F_r(w)$ and $F_i(w)$ are the real and imaginary parts of $F(w)$.

The GD was evaluated on the LP residual as it shows more clearly the points of excitation than the original speech signal. It was found that the output of the GD had a considerable noise element which made the evaluation of zero crossing points difficult. The GD was therefore smoothed with a simple 23rd order linear FIR filter with a passband edge frequency of 400 Hz; these values were found experimentally to produce the best output for both male and female speech.

The GD output signal was then filtered with 15th order linear FIR high pass filter with a passband edge frequency 50 Hz with isolated peaks removed by using a 3-point medium filter. Figure 6.5 shows a GD output for male voiced to voiced transition. Compared
6.4. **Group Delay Method**

Figure 6.5: Male voiced to voiced transition: Input Residual (top), Modified Time Envelope signal used by Trapezoidal Search (centre) and Output of GD (bottom)

to the MTE signal which is used by the Trapezoidal Search at the centre of the figure, the GD signal is considerably smoother and would be less prone to processing errors.

At points in the LP residual where little excitation was present the GD performance degraded. Therefore it was found that at such sections better performance could be found by using the MTE with a value of $\beta$ of 0.8 from (4.2) from Section 4.5.1. The GD method was found to be very sensitive to both window size and shape. Regarding window shape, overall a Blackman window typically gave the best performance as this window did not significantly alter the phase characteristics of the signal but tapered the signal to zero at the ends. Regarding window size, typically due to the longer pitch lengths in male speech a window length of 160 samples gave the best results whilst for female a 80 sample pitch length was optimum.

The GD method was also sensitive to speech sections where the pitch frequency was changing relatively rapidly such as onsets and voiced to voiced transitions. Although a Hann window with its better frequency resolution than the Blackman partially solved this, there was still problems in these areas using GD method. Figure 6.6 demonstrates the uneven variation in pitch sizes found at zero crossing points.
6.5 Closed Loop Pitch Cycle Detection

Closed loop matching methods are essentially used in hybrid coders which utilise an Analysis by Synthesis process (AbS) [62]. In the analysis stage speech is represented by a compact set of parameters which are encoded efficiently. In the synthesis stage these parameters are decoded and used in conjunction with a reconstruction mechanism to form speech. By closed loop analysis the parameters are extracted and encoded by minimising a measure difference between the original and reconstructed speech.

Although this method is of high complexity modern AbS coders such as CELP, which extracts from a codebook the excitation sequence that will produce the closest match, produce high quality speech as they are able to better perform at speech sections where
speech changes rapidly than sinusoidal systems which do not employ any closed loop routine \[12\]. AbS coders such as CELP, by directly coding the original speech signal, implicitly allocate an excessive number of bits to the phase information - more than is perceptually required. At higher rates the allocation of bits to this phase information does not appear to degrade speech quality but at low bit rates this is not the case and it is considered that this is one of the main reasons why CELP based coders performance degrade below this rate \[63\].

Sinusoidal coders do not extract and transmit this phase information as it is generally considered that the human ear is insensitive to it. A sinusoidal system which was better able to code transitional speech sections more accurately should produce synthetic speech approaching the quality of hybrid coders. If the PS SB-LPC was able to operate in a closed loop manner, although there may be an increase in complexity, the synthetic speech should be of higher quality as pitch errors introduced in an open loop system should be greatly reduced and the need for post processing eliminated.

However for a sinusoidal closed loop method to operate successfully the input and

![Figure 6.7: Comparison of (a) input speech and (b) output speech of the SB-LPC](image)

output speech produced must have similar phase properties for the two waveforms to be successfully matched. The dissimilarity of the input and synthetic speech in the SB-
6.5. Closed Loop Pitch Cycle Detection

LPC is illustrated in Figure 6.7. Clearly these two waveforms cannot be compared in a closed loop system as they contain different phase information. Therefore the phase information must be removed from the original speech or a synthetic phase must be utilised by the PS SB-LPC for matching purposes but not transmitted to the decoder.

A speech coder utilising phase removal was used by [64]. This method was later utilised in a MELP/CELP hybrid coder [13]. Known as Zero Phase Equalisation it was used in order to remove the phase component in the CELP in order that when switching between the coder types artefacts are not produced. As CELP is a waveform coder it preserves waveform shape and time synchrony of the input speech, the MELP as a sinusoidal coder does not. This method utilises Matched filtering techniques.

6.5.1 Phase Removal in the PS SB-LPC

In the 4 kbps hybrid CELP/MELP [13] coder Zero Phase Equalisation is applied to the LP residual. The speech domain target signal is generated from the equalised LP residual and the estimated LP parameters. In a frame for which the CELP is chosen, equalised speech is used as the AbS target. The filter coefficients are found once per 20 ms frame and interpolated for each 2.5 ms sub-frame.

In the PS SB-LPC we do not interpolate values across the frame, so the filter coefficients must be found for each cycle, as significant excitation in the LP residual is not always present the Zero Phase Equalisation is applied to the input speech waveform. The equalisation is performed with time domain FIR filtering utilising matched filter techniques. The input is converted into a speech pulse train through the FIR filter whose coefficients are the values of the input speech cycles. The coefficients are generated by extending the pitch waveforms with zeros so that the middle of the waveform corresponds to the middle filter coefficient. The number of added zeros is such that the length of the equalisation filter is equal to length of the input speech frame of 160 samples.

From the PDA employed in the PS SB-LPC the average pitch size per 160 samples of the input speech frame is found. From sample 10 to 150 of each speech frame every sample of the frame is placed at the centre of a Blackman window of the average pitch
6.5. **Closed Loop Pitch Cycle Detection**

length. Samples 10 to 150 were used to ensure the cycles chosen were only from that particular frame. A Blackman window was eventually chosen over other window types as it had a quicker roll off at its centre which would emphasise the important feature of the PCW at the centre of the window whilst minimising the signal at the window edges.

Once the window with the highest energy is selected, a DFT the length of the windowed signal is performed. The spectral amplitudes of the filter coefficients are normalised, i.e. the gain of the filter is set to one, to ensure an approximate all pass filter characteristic. The phases are then unwrapped and the IDFT taken. This produces the coefficients of an FIR approximate all pass filter that when used to filter the corresponding frame of speech will produce an zero phase signal.

To prevent glitches at frame boundaries in the phase removed signal the filter coefficients are adjusted by interpolation to smooth frame boundary effects. Before interpolation was carried out it was necessary to align the current with the previous cycle. This was achieved by cross correlating the current with previous interpolated cycle and finding the delay $d$ which results in the highest correlation value resulting from (6.3). The current cycle is then adjusted by the value of $d$.

$$r(d) = \frac{\sum_{i=0}^{L-1} [(x(i) - \bar{x})(y(i - d) - \bar{y})]}{\left[\sum_{i=0}^{L-1} (x(i) - \bar{x})^2 \sum_{i=0}^{L-1} (y(i - d) - \bar{y})^2\right]^{1/2}} \quad (6.3)$$

where $x$ is the last interpolated cycle, $y$ is the current not yet interpolated cycle.

The process of normalising the filter coefficients caused considerable variation in the extracted cycles from cycle to cycle. This is illustrated in Figure 6.8. When these normalised filter coefficients are used to zero-phase equalise the input speech at sections where the input speech changes rapidly glitches appear in the phase removed signal. Low pass filtering of the normalised coefficients does smooth their variation from cycle to cycle but it also affects their amplitudes which causes the phase removed speech to vary considerably in amplitude compared to the original speech.

To smooth the normalised cycles their frequency content is band-limited. After taking the DFT instead of setting all the spectral amplitudes to one, the root mean square of
6.5. Closed Loop Pitch Cycle Detection

Figure 6.8: (a) Input speech, (b) Extracted interpolated cycles and (c) the cycles in (b) Normalised (scaled here for clarity)

the cycle is taken over the length of the window. A limit is then set over which the spectral amplitudes are set to one and under which the spectral amplitudes are set to zero.

It was found experimentally that this limit should be set at a value of 0.6, this smooths the normalised cycles without substantially affecting their amplitudes by limiting their frequency content. When this is carried out the variation in the normalised cycles from cycle to cycle is reduced and glitches in the ZPE signal reduced to an acceptable level. This is shown in Figure 6.9. Although we are removing significant frequency content from the resulting ZPE signal, this does not significantly affect its pitch structure which remains essentially the same as the original speech signal.

Figure 6.10 shows a section of male input speech that has a large amount of phase information and below is shown its approximately zero phase equivalent. Although
6.5. Closed Loop Pitch Cycle Detection

Figure 6.9: Cycles after normalisation (top) Cycles after frequency bandlimiting the normalised cycles (bottom)

Figure 6.10: Male Input Speech (top) and approximately Zero Phase Equalised signal (bottom)

there is considerable phase information in the input speech the great majority of this information has been removed leaving an approximately zero phase signal with clear peaks present.
When the ZPE was tested perceptually it was very similar to the original speech however due to the frequency band limiting effect of the filter coefficients the ZPE does not have the same perceptual variation. Also occasional artefacts can be heard which are caused by the rapid cutting off in frequency content from cycle to cycle and also by the process of phase removal itself.

In Figure 6.11 at 125 samples in the ZPE we can see that there is not a subtle variation from the cycles to the left of this point as there in the original speech. The amplitude of the pulse at 125 samples is distinct from its surrounding cycles, this is not seen in the input speech. However there is a clear pitch structure present which is the primary objective of the phase removal process.

By utilising matched filtering techniques an approximately zero phase speech signal has been produced. This speech signal sounds similar to the original except for a few artefacts caused by the phase removal techniques employed. These artefacts were reduced to an acceptable level by band limiting in the frequency domain the filter coefficients employed and interpolating them across frame boundaries. The next section will describe the steps taken to use this signal as a target in a waveform matching process.
6.5.2 Waveform Matching to Phase Removed Signal

A approximately zero phase speech signal has been produced which preserves time synchrony with the original speech signal by accurately replicating its pitch track. This allows the possibility of utilising AbS waveform matching techniques to match the decoded speech of the PS SB-LPC with the ZPE signal while requiring no increase in bit rate.

The aim is to carry out a Root Mean Square Error (RMSE) evaluation for every PCW contained within a frame of speech. By altering independently the pitch length of each PCW and comparing the subsequent synthetic speech to the ZPE signal the optimum pitch track can be found. The comparison that produces the lowest value in error terms will give the best pitch values for that frame of input speech. Figure 6.12 demonstrates how a comparison of the ZPE signal and the synthetic speech produced by the PS SB-LPC can be carried out.

![Figure 6.12: Procedure for waveform matching of the ZPE signal and synthetic speech from PS SB-LPC.](image)

The procedure to carry out the matching process is:

1. Carry out zero phase equalisation over two frames of input speech.
2. From PDA interpolate N PCWs over length L greater than 160 samples.
3. Execute encoder and decoder over length L to generate synthetic speech.
4. Over length L compute RMSE:

\[
RMSE = \left( \frac{1}{L} \sum_{i=0}^{L-1} (X(i) - Y(i))^2 \right)^{1/2}
\]  

(6.4)

where \(X(i)\) and \(Y(i)\) are the ZPE and synthetic speech signals respectively
5. If \(RMSE < RMSE_{\text{previous}}\) then save PCW positions and error value.
6. Alter by one sample length, one of the N PCW over new length L.
7. Repeat steps 3 - 6 until all possible combinations tried.
8. PCW positions associated with lowest RMSE are used for that length L of input speech.
9. Finally execute encoder and decoder with these saved PCW positions over final length L to generate synthetic speech.

6.5.2.1 Algorithm Specifics

The pitch range in the PS SB-LPC varies from 15 to 150 samples. To allow for the PCW in a frame to vary independently for all values between these limits would produce a search which is computationally very high. For example, consider a frame of female speech consisting of seven PCWs. Consider now that we vary their lengths between only 15 to 50 samples. The number of possible combinations for this frame (complexity \(\lambda\)) is then given by the number of PCWs raised to the power of the range of samples, in this case this gives \(35^7\) or \(64.34 \times 10^9\) possible combinations.

The major determining factor on the complexity of the search is therefore the number of PCWs per frame. This has implications on the quality of the synthetic speech produced for male and female speakers as the longer pitch sample length for male speakers produces less PCWs per frame. As the number of PCWs per frame cannot be reduced clearly some limitation has to be placed on the the possible pitch sample search range of individual PCWs. During encoding and decoding some functions will be called many times; these functions must be optimised for speed in order to ensure
an optimum search is a practical possibility both in development and implementation terms. To reduce complexity a limit is placed on the pitch search range when waveform

![Figure 6.13: Limiting search range of pitch values, showing the initial start positions of PCWs and their pitch sample search range](image)

matching. The search range is based on \([\text{minsize}/2]\) where \(\text{minsize}\) and \(|x|\) are the smallest PCW length found over the frame of interpolated PCW sizes and \(\text{floor}\) function respectively. A simple algorithm is then used to ensure all samples between the PCW positions are covered.

Figure 6.13 shows the initial starting positions of PCWs in a frame of speech given by interpolating the pitch values from the PDA. \(T_0\) is the starting point of the first estimated PCW and cannot be moved as it was set as end point in previous frame. In this example the minimum size is 39 samples between \(T_1\) and \(T_2\) so the range is based on \([39/2]=19\) samples either side of the initial start positions. This overall procedure was not found to be a limiting factor on synthetic speech quality because of the good pitch estimates that are available from the pitch values found by interpolating the PDA decision.

During matching all cycles are declared as fully voiced, this is done because to generate the unvoiced portion of a cycle a random number generator is used. When cycles are not declared as fully voiced no valid comparison can be made because of the non-repeatable element of not declaring a cycle as fully voiced. Also as the voicing level estimation cannot guarantee always the correct voicing result this could introduce errors into the matching routine.
The PCW positions are then varied independently and for every single sample change in a PCW length the matching process is performed. No matching is required at frames of speech which can be clearly marked as silence or unvoiced. If all the cycles contained in the frame had an energy level lower than 5.0 or were marked as unvoiced using the elements of the hard decision voicing no matching is carried out for that frame and the interpolated PCW sizes are used.

6.5.2.2 Waveform Matching at Speech Onsets

At most sections of unvoiced or silence speech no waveform matching is carried out but at some of these sections it is. Here there is no clear ZPE signal to be waveform matched and therefore after these sections, at onsets the synthetic speech can be several samples away from the ZPE signal. A simple onset detector was therefore designed to reset the ZPE signal, this was based on normalised correlation and energy of the interpolated pitch cycles. In this section the PCW referred to are the interpolated values from the PDA.

The normalised correlation is calculated between the current PCW and the two neighbouring PCWs, to the left and right. The previous PCW (left) compared to the current is defined as corr\text{cycle} and the next PCW (right) compared with the current as corr\text{cycle}. At onsets the value of corr\text{cycle} will be higher than corr\text{cycle} as this cycle will be highly correlated with the next and not the previous PCW. The energy of the interpolated PCWs and frames can also be utilised.

After testing each cycle, the speech containing of N PCWs up to 320 samples then contains an onset if:

\[
\begin{align}
\text{corr}_{\text{compare}} & = 1 \text{ and } \text{ratio}_{\text{corr}} > \alpha \text{ and } \text{corr}_{\text{cycle}} > \beta \text{ and } \\
(E_{f_j}/E_{f_{j-1}}) > \theta \text{ and } E_{c_i} > E_{c_{i-1}} \text{ and } E_{c_{i+1}} \geq E_{c_i}
\end{align}
\]

where

\[
\text{corr}_{\text{compare}} = \begin{cases} 
1 & \text{if } \text{corr}_{\text{cycle}} > \text{corr}_{\text{cycle}} \\
0 & \text{otherwise}
\end{cases}
\]

(6.5)
where \( \text{ratio}_{\text{corr}} = \frac{\text{corr}_{\text{cycle}}}{\text{corr}_{\text{cycle}}}, \) \( E_{ci} \) is the energy of the \( i^{th} \) cycle, \( E_{fj} \) is energy of current frame. The values of \( \alpha, \beta \) and \( \theta \) were set experimentally to be equal to 1.05, 0.75 and 15 respectively. A example of the sections of input speech declared as onsets in shown in Figure 6.14. When an onset is detected the synthetic speech is produced for

![Figure 6.14: Sections of input speech declared as onsets. The frames of speech that contain onsets are shown as non-zero lines.](image)

the interpolated pitch lengths and compared to the ZPE signal and the error calculated as in (6.4). The synthetic speech is then effectively slid to the left and then to the right of the ZPE by reducing and then increasing all the PCW starting points one sample at a time for every PCW until \([\text{minsize}/2]\) is reached on both sides.

If the best match found is a negative index i.e. when the synthetic speech is slid to the left of the ZPE signal, then that number of samples is removed between \( T_0 \) and \( T_1 \) and if the index is positive samples are added. This ensures that when the waveform matching process is carried out the starting positions of the PCWs are not too far away from the ZPE signal. Figure 6.15 shows the resultant synthetic speech with and without onset adjustment.
6.5. Closed Loop Pitch Cycle Detection

6.5.2.3 Irregular Pitch Size Variation

At sections of speech where the pitch value can be irregular over the frame, the decoded speech was not always successfully matched to the ZPE. This is because to reduce complexity we limit our pitch search range according to the minimum size of the initial interpolated PCW positions found per frame, see Section 6.5.2.1. When the pitch varies greatly within a frame we do not always have enough range of pitch values to test all PCW size combinations. In order to detect where such sections might occur the variation in PCW sizes $V$ is measured by:

$$V = \left( \frac{1}{N} \sum_{i=0}^{N-1} \left( T_i - T_{i-1} - T_{\text{avg}} \right)^2 \right)^{1/2}$$

(6.7)

where $N$ is the total number of interpolated PCW sizes of the input speech over length $L$, $T_i$ is the $i^{th}$ PCW position and $T_{\text{avg}}$ is the average size of the PCW over length $L$. It was generally found that at a small number of frames where $V$ was greater than a value of 15 remedial action was required.

If speech over length $L$ is identified to contain an irregular pitch then the following
6.5. **Closed Loop Pitch Cycle Detection**

The algorithm is executed which is based on a Pitch Grid Array search [53]. Over length L the ZPE is searched and all the positive peaks are identified. The positions and amplitudes of the peaks are stored as \( p(n) \) and \( a(n) \). The number of peaks is small as we operate on the ZPE signal. Firstly the smallest peaks are removed. This is done by selecting each peak \( p(n_i) \) in turn and discarding it if there is another peak \( p(n_j) \) which satisfies the following conditions:

\[
a(n_i) < \delta a(n_j) \text{ and } |p(n_i) - p(n_j)| < \alpha_f P_{\text{min}} \text{ and } p(n_i) < p(n_j)
\]

or

\[
a(n_i) < \delta a(n_j) \text{ and } |p(n_i) - p(n_j)| < \alpha_b P_{\text{min}} \text{ and } p(n_i) > p(n_j)
\]

where \( P_{\text{min}} \) is the minimum estimated pitch value over L. Experimentation showed that the values of 0.65, 0.6 and 0.65 for \( \delta \), \( \alpha_f \) and \( \alpha_b \) respectively removed most of the significant peaks that exist, leaving behind a sequence containing a few peaks per frame. The amplitude of the maximum peak is then set as \( \beta \). The value of \( \beta \) is then successively reduced by 5 percent, at each reduction the revealed peaks are set as the PCW positions. The matching process is run and the MSE stored. When \( \beta \) reaches 40 percent of its initial value the process is stopped and the lowest error combination taken with associated PCW positions.

6.5.2.4 **Waveform Matching at Steady State Speech**

At some sections of input speech there is very little variation in the pitch values. This steady speech does not require a large variation in pitch values and hence the complexity of the waveform matching routine can be reduced. Typical speech selected as being declared as steady is shown in Figure 6.16. This decision is based on the normalised correlation introduced as (3.39). Each cycle within a frame is compared with its neighbours and if all cycles have a normalised correlation greater than \( \alpha \) it is declared as steady state speech, \( \alpha \) was set experimentally at the value of 0.98. If the frame of speech is declared as steady in nature then the range of values in the search routine is limited to value of five samples. This applied to only a small percentage of frames.
Despite taking all the steps described above, the complexity of the Full Search is such that the coder in its current configuration is not usable in real time operation due to the length of the encoding/decoding times. Routines developed over several months in the SB-LPC and original PS SB-LPC were not intended to be used in such an implementation. The next section describes the steps taken to make the waveform matching routine a practical solution.

6.5.3 Reducing Complexity of Full Search

In order to ascertain the sections of the speech coder which are most computationally intensive, a Linux operating system based profiling tool known as gprof [65] was used. This produces an execution profile of programs which have been executed in the 'C' programming language. Profiling was used to determine where the coder spent its time and which functions called which other functions while it was executing. This information can show which sections of the coder are slower than expected, and might be candidates for rewriting to make it execute more quickly. It also indicates which functions are being called more or less often than expected.

The time taken by the Full Search routine is shown as column A in Figure 6.17. The
6.5. Closed Loop Pitch Cycle Detection

results are shown for one hundred frames (approximately two seconds) of continuous female input speech sampled at 8 kHz from the NTT database [57]. The speech samples were deliberately chosen to be the most computationally complex for the coder in that the speech file used frequently contained at least seven cycles per frame.

It can be seen from column A that the coder running the Full Search takes almost 11 hours to produce synthetic speech from these frames of female speech. The most computationally expensive sections of the Full Search will now be described.

![Figure 6.17: Reducing complexity of Full Search. Time Taken by the Full Search with reduced complexity. Column A is most complex and column F is least](image)

- At its decoder the PS SB-LPC uses a DFT based speech synthesis to produce an excitation pulse which is used to excite the vocal tract filter defined by the LPC values. As the decoder is called for every change in PCW length this means that a DFT based routine is applied many times.

- The SB LPC along with many other time synchronous vocoders extract LPC parameters from fixed length portions of speech using the autocorrelation method.
6.5. Closed Loop Pitch Cycle Detection

The PS SB-LPC can also use covariance method to extract the LP filter coefficients for each PCW. However the covariance method implemented is considerably more complex than the autocorrelation method and therefore more expensive in computation time.

- During analysis the PS SB-LPC uses a DFT to determine the spectral amplitudes for each PCW. This is necessary to represent the spectral information that is not accurately modeled by the LP filter. If the spectral amplitude information is not transmitted the synthetic speech quality is degraded. Setting the spectral amplitudes to unity at the encoder does not have a marked effect on the pitch structure of the synthetic speech produced but it does degrade the synthetic speech quality.

- During synthesis in order to ensure smooth evolution of the speech signal, energy scaling is applied to the speech excitation. The excitation is passed through the LPC filter per pitch cycle and the energy of the reconstructed speech per cycle is computed. The excitation is then scaled until the the energy of the reconstructed pitch cycle matches that of the transmitted pitch cycle energy. This process is carried out several times per pitch cycle but is important as the accuracy of the matching process degrades if not carried out to a high degree of accuracy.

In order to reduce the coder complexity significantly the following changes were made to the coder, these progressive changes are shown as columns B to F in Figure 6.17. Changes made in coder configuration B were transferred to configuration C and so on.

A Full Search routine. Normal PS SB-LPC operation

B An excitation pulse for each PCW was not generated at the decoder during the matching process. Rather a fully voiced pulse was written to file and used to generate the excitation pulse.

C To determine the LP filter coefficients for each PCW the autocorrelation method was used instead of the covariance method.

D The spectral amplitudes for each PCW were not found during the matching process but were set to unity at analysis.
6.5. Closed Loop Pitch Cycle Detection

During analysis the LP coefficients were found by interpolating over the frame rather than being found for each PCW.

During synthesis the LP coefficients are used directly rather than being found from converting the Line Spectral Frequencies.

Once matching was complete, normal PS SB-LPC operation was used to determine the parameters for each PCW. The most complex routine remaining is the energy scaling of the excitation during synthesis. This accounts for over 20% of the remaining complexity. It was not possible to determine a method to replace this without affecting the quality of the matching routine. The following section describes a Reduced Search matching routine aimed at reducing coder complexity further without degrading the synthetic speech produced.

6.5.4 Reduced Search Solutions

Even with the measures taken to reduce the complexity of the Full Search routine the time taken to fully test all possible combinations was prohibitive. This was most noticeable when the PCW lengths were small and hence a high number were present in a determined frame of speech. This affected in particular female speech where the pitch lengths tend to be smaller than for male speakers. Take for example typical male speech which from the PDA is estimated to contain four pitch cycles each of length sixty samples. This requires the testing of $30^4$ or $81 \times 10^4$ combinations.

Although we have already limited the search range either side of the PCWs contained within each frame according to the interpolated minimum size PCW, consider a female speaker which contains six PCW each of length thirty five samples. This requires the basic testing of 17 samples either side of each PCW. In complexity terms there are $17^6$ or $24.14 \times 10^6$ combinations to be tested. In order to reduce complexity further a Reduced Search routine was developed. This Reduced Search routine is shown as Figure 6.18.

In the figure (a) shows the ZPE which is to be matched, (b) shows the current positions of the synthetic speech for the frame which the estimated starting positions for each
Figure 6.18: Reduced Search routine. (a) ZPE signal, (b) Synthetic speech with initial estimate, (c) starting position for search of $T_1$ (d) final position for search of $T_1$ and (e) After $T_1$ has been set, new start position for remaining cycles.
6.5. Closed Loop Pitch Cycle Detection

cycle $T$ from the PDA. In (c) all PCW positions except for $T_0$ are all reduced by $T$ - range until $T_{\text{min}}$ is reached where range is $\lceil \text{minsize}/2 \rceil$. Then all PCW positions are increased by one sample until $T_{\text{max}}$ is reached shown in (d), over all changes the RMSE is measured over $T_0$ to $T_{\text{current}+1}$ using (6.4). The lowest RMSE gives the best position of $T_{\text{current}}$ which in this case is $T_1$, which is set in (e) After $T_1$ has been set as shown in (e) this routine is then repeated for the remaining PCW positions in the frame but the $T_{\text{previous}}$ are not moved.

It was found that there was considerable jitter in the cycles sizes when this Reduced Search routine was implemented. When attempting to match cycles of the ZPE signal which are dominated by a solitary peak such as shown below in Figure 6.20 the jitter in cycles was minimal and had little impact on perceptual speech quality. However when the energy is spread over the cycle and not dominated by a single peak as shown below in Figure 6.19 it makes it more difficult for the Reduced Search routine to make to decision upon the best pitch sizes based on the current error measure . This jitter in size from cycle to cycle caused artefacts in the synthesised speech which affected its perceptual quality.

This was mostly solved by changing $(X(i) - Y(i))^2$ of (6.4) to $(X(i) - Y(i))^4$, where $X$ and $Y$ are the the ZPE and Synthetic speech signals respectively. This placed greater emphasis on the highest peak of the PCWs, thus reducing the amount of size jitter from cycle to cycle although in some cases it does have an impact on the perceptual speech quality and can be heard as a roughness in the synthetic speech. However when the PCW lengths are quantised the majority of this remaining size jitter is removed as small deviations from cycle to cycle are removed by the quantisation process.

Closed Loop Matching Results

The Full and Reduced Search routines have been integrated into the PS SB-LPC. Waveform matching is switched between these two methods by selecting the complexity setting $\lambda$ given in Section 6.5.2.1 before matching is carried out for each frame. If complexity is above $\lambda$ then the Reduced Search routine is used for that frame. Figures 6.19 to 6.22 demonstrate some results for the matching routines at non-steady speech sections. As can be seen the ZPE signal and synthetic speech are well matched for
different types of input speech and only a small amount of cycle size jitter is present.

Figure 6.19: Waveform matching results with at onset and considerable energy spread per cycle

Figure 6.20: Waveform matching results at voiced to voiced transition with resonant section at centre
6.5. Closed Loop Pitch Cycle Detection

Figure 6.21: Waveform matching results at short speech segment with onset and resonant speech at end of section

Figure 6.22: Waveform matching results at voiced to voiced transition with considerable energy spread per cycle
6.6. Remaining Issues With Closed Loop Pitch Cycle Detection

6.6.1 Phase Removal

The method of phase removal relies on obtaining an ZPE signal which accurately replicates the pitch structure in the input speech. This method utilises a cycle per frame of the input speech which is considered to have highest energy. This however can be problematic at speech sections where the structure of the original speech is in transition from cycle to cycle. This is illustrated in Figure 6.23 which shows a male speech offset. From Figure 6.23, in the ZPE signal from point (a) onwards the signal does not accurately follow the original speech shown. This is due to cycle selection and processing in the phase removal process which cannot account for all possibilities in the original speech signal. As a consequence the synthetic speech shown does not match the original speech and some speech distortion can be heard.

![Figure 6.23: Phase Removal Errors. Original Speech (top), ZPE signal (middle) and Synthetic Speech (bottom)](image-url)
6.6. Remaining Issues With Closed Loop Pitch Cycle Detection

6.6.2 Waveform Matching

Although good results have been obtained there some existing issues in the matching process which are mainly, but not solely limited to the Reduced Search matching routine. Figure 6.22 demonstrates some of the problems that currently exist. This figure shows a transition from a resonant to non resonant speech section. At sample time 100 the Reduced Search routine has produced a error in the PCW size. At resonant sections like this there can be a large change in PCW size but the resultant synthesised waveform shape changes very little due to its low energy. As we have limited the range in pitch search to reduce complexity the next and subsequent PCW cannot move to the best position because they are too many samples away. These problems are caused by the fact that the Reduced Search routine does not currently operate on a frame basis.

![Graph showing problems with matching routine in PS SB-LPC](image)

Figure 6.24: Problems with matching routine in PS SB-LPC

However there is a possible solution to problems encountered in the matching process. Also shown is a calculated final error found for the frame which is calculated by

\[
FrameError = (V_{diff} \times RMSE)^{1/2}
\]  

(6.9)
where \( \text{RMSE} \) is given by (6.4) and \( V_{\text{diff}} = (V_{\text{past}} - V_{\text{present}})^2 \). \( V \) is simply the variation of the PCW sizes given by equation (6.7). \( V_{\text{past}} \) and \( V_{\text{present}} \) are the variation in PCW sizes of the PDA estimate and matched sizes respectively. This biases the frame error towards those frames where there is a higher variation in PCW sizes than that of the PDA estimate. This is required because as can be seen in Figure 6.22 there is some inevitable underlying difference between the ZPE and synthetic speech signals. This final error could be fed back into the coding routine to remove matching errors.

![Figure 6.25: Comparison of coder output. Synthetic speech produced by Closed loop method (top) and Trapezoidal Search (middle). Also shown is the MTE signal input to Trapezoidal Search (bottom).](image)

A comparison of the speech coded by the Trapezoidal Search and the Closed loop methods described in this chapter is shown in Figure 6.25 also shown is the Modified Time Envelope (MTE) which is the signal used by the Trapezoidal Search for pitch detection.

The MTE signal (bottom) shows a strong pitch structure up to sample time 250 shown at point (b), after this point its pitch structure breaks down. As a consequence the pitch values in the synthetic speech produced from point (b) onwards must be heavily smoothed by post processing. At point (a) the smoothing has failed and as a result
6.7 Concluding Remarks

an artefact has occurred. The synthetic speech produced by the Closed loop method (top) does not contain an artefact as it does not rely on the MTE but is coded from the original speech which unlike the MTE will have a clearly defined pitch structure.

6.7 Concluding Remarks

This chapter has described the techniques used and evaluated to improve the pitch cycle detection of speech in a PS sinusoidal coder. Open and closed loop methods were described and implemented. A closed loop search routine was described and implemented. This method though of high computational complexity was able to remove most of the problems associated with the open loop method. Although some problems remain with this closed loop method, techniques to improve this method were described and partially implemented.
Chapter 7

Spectral Amplitude Quantisation

7.1 Introduction

The majority of the spectral characteristics of the speech in the PS SB-LPC are modeled by the 10th order LPC filter. The remaining characteristics are provided by the spectral amplitudes of the LP residual signal. These characteristics are required as the LPC filter does not model successfully all the spectral components of the speech signal. The spectral modeling of the LPC filter is limited by three factors; filter order, all pole modeling and speech stationarity assumptions. Because of these assumptions aimed at reducing bit rate, accurate analysis and quantisation of spectral amplitude information is required.

7.2 Background

The main difficulty when quantising spectral amplitude information is that the number of amplitudes to be quantised is dependent upon the pitch length $P$ of the pitch cycle. The number of amplitudes $N$ present is given by

$$N = \frac{P}{f_s} \times f_c = 0.4625 \times P$$

(7.1)

where $f_s$ is the sampling frequency of 8000 Hz and $f_c$ is the cut off frequency typically 3700 Hz. In the PS SB-LPC the pitch cycle length is assumed to vary between 15
to 150 samples applying (7.1) results in a vector length of between 7 to 70. Suitable quantisation schemes must be designed to take into the account the long length of such a vector and the variation in its length as a function of the pitch. Typically VQ techniques is the preferred method for low bit rate speech coders due to its enhanced performance. Normal VQ routines are applied to fixed vector lengths so alternative methods have been produced in order to solve this problem.

The 2.4 kbps MELP [44] coder uses the fact that low frequency components are more important than high frequencies and therefore the corresponding perceptually important components should be quantised more accurately than the rest. This coder utilises VQ to quantise the first 10 spectral amplitudes with the remaining amplitudes set to unity. This method results in lower distortion when the pitch length is shorter such as for females and children but for longer pitch lengths such as in male speech the number of harmonics can be high and as a result only a small number of the spectral amplitudes are transmitted with accuracy. For example a pitch length of 100 samples produces a vector length of over 46 entries, as only accurate information on the first ten values of this vector are kept only a small amount of information up to 800 Hz is utilised which can lead to rather severe quality degradation.

In [66] a Mel-Scale Transformation was used to translate the variable length amplitude vectors to a fixed length. This method divides the spectrum into frequency bands, the amplitudes contained in the bands are averaged and quantised as a single value. The frequency bands are selected by a Mel-Scale measure which takes into account the variation in sensitivity of the human ear with frequency. In theory, the bands are of equal perceptual performance. As none of the spectral amplitudes are discarded this results in less synthetic speech quality degradation especially in male speech where the number of spectral amplitudes to be quantised is high. The variable dimension spectral vector \( x \) of length \( L \) is converted into a fixed dimension vector \( z \) of length \( M \) as

\[
z(m) = \frac{1}{u_m - l_m + 1} \sum_{k=l_m}^{u_m} x^2(k)
\]  

(7.2)

where \( x(k) \) and \( z(m) \) denote the \( k^{th} \) and the \( m^{th} \) elements of the vector \( x \) and \( z \) respectively and \( l_m \) and \( u_m \) denote the lower and upper harmonic bounds of the \( m^{th} \)
7.3. Quantisation of Spectral Amplitudes

spectral band \([L_{\frac{M}{2}} - 0.5]\) and \([L_{\frac{M+1}{2}} - 1.5]\) (with \(u_{M-1} = L - 1\)) respectively. The fixed length vector \(z\) can be quantised through VQ and if it contains enough bands the warping technique should be transparent. However to obtain good speech quality over \(M\), typically 20, or more bands for male speech are needed, requiring many bits to quantise this number of values. Although capable of providing good speech quality at higher bit rates, for low bit rates this method is not efficient.

Optimal vector quantisation of variable-dimension vectors in principle is feasible by using a set of fixed dimension VQ codebooks. However such a multi-codebook approach would be excessive in storage and computational complexity. Variable Dimension Vector Quantisation (VDVQ) \([67]-[68]\) attempts to solve this problem by transforming the codevectors of the quantisers codebook instead of the input vectors. VDVQ uses a single fixed dimension universal codebook covering the entire range of input vector dimensions. This technique aims to reduce the quantisation distortion as the input vectors are not subjected to any transformation which in general create losses.

However this method requires extensive and elaborate training processes to produce the universal codebook with a very large number of training vectors especially with a codebook of high dimension. This method was tried by \([34]\) but gave poor results.

The SB-LPC \([34]\) used a peak picking algorithm to transform the vector to a fixed length. This method selects spectral amplitudes according to their perceptual importance. It was found to produce good quality speech. The PS SB-LPC \([40]\) attempted to utilise this method to quantise the spectral amplitudes, although good quality synthetic speech could not be produced when this method was used in conjunction with the routines outlined in Section 4.5.3.4. The following section describes the actions taken to accurately quantise the spectral amplitudes based on the peak picking method.

7.3 Quantisation of Spectral Amplitudes

The current method of quantising the spectral amplitudes in the PS SB-LPC was summarised in Section 4.5.3.4 is to pick and quantise fourteen amplitudes from the first and last cycle in the frame. The fourteen amplitudes per cycle of the remaining
cycles are effectively selected by a Joint Quantisation Interpolation (JQI) across the frame. This method was originally designed for effectively quantising the LSFs. This is possible because the LSF spectrum shape generally varies in a deterministic pattern for each cycle in the speech frame. The spectral amplitudes however do not represent a smooth shape, they describe the speech signal after the vocal tract information has been de-convolved and are almost noise like in shape. This can be clearly seen in Figure 7.1 which shows a log amplitude plot of the LP spectrum and corresponding spectral amplitudes for several cycles of male voiced speech.

Currently the edge vector containing twenty eight amplitudes in total from the first and last cycle in the frame is quantised with 24 bits and the shape vector with 6 bits. If there are two cycles in a frame then the edge vector is not used and the values are quantised directly. Male speech with its longer pitch values frequently contains only two cycles in a frame, this typically means that for a considerable segments of male speech twenty percent of the bits allocated for spectral amplitude quantisation are not used.
7.3. Quantisation of Spectral Amplitudes

An experiment was initiated to see the effect of raising the bit rate on the JQI method. The edge vectors were re-quantised at 27 and 30 bits, at four stages of 8, 8, 8, 3 and 8, 8, 8, 6 bits respectively. The effect of this increase in bit rate is shown as Figure 7.2.

Figure 7.2: Average MSE per cycle of JQI spectral amplitudes for male and female speech

The current peak picking algorithm see Section 4.5.3.4 can only pick a maximum of fourteen amplitudes. The first two amplitudes and no more than three peak amplitudes and one amplitude either side of the peaks; fourteen in total. For testing purposes the maximum number of peaks is set at five peaks therefore seventeen amplitudes can be (but rarely are) picked for each cycle. For experimental purposes this vector can be considered as the target as a 10th order LP filter is considered to contain two formants per peak. The error is measured against this maximum of seventeen amplitudes only as the other amplitudes such as those corresponding to formant valleys do not contain perceptually important information. The average MSE per cycle is measured as

$$MSE = \frac{1}{N} \sum_{i=0}^{N-1} (x(i) - \hat{x}(i))^2$$  \hspace{1cm} (7.3)
where \( x \) and \( \hat{x} \) are the target unquantised amplitude vector and quantised picked amplitude vector per cycle respectively.

In Figure 7.2 for both male and female speech the average MSE falls with increasing bit rate of the edge vectors, however the fall is greater for male than female speech. It is believed that this due to the fact that female speech uses the shape vector more frequently than male due its greater number of cycles per frame. This increase in bit rate of the edge vectors has a greater effect on male speech as its amplitudes are quantised directly without the influence of the shape vector. As male speech with its greater pitch length, frequently has only two cycles per frame which do not require the use of a shape vector during quantisation.

The number of spectral amplitudes for any given frame is demonstrated in Figure 7.3 which shows Frame A with two cycles and a Frame B with six cycles. It is clear that despite the variation in the number of cycles between the two frames that the number of spectral amplitudes for a given frame size is similar despite the number of cycles per frame. This is a direct consequence of (7.1).

![Figure 7.3: Spectral Amplitude allocation for two frames containing differing numbers of cycles](image)

Instead of allocating six bits to a shape vector which interpolates across a noise like signal, it may be more efficient to allocate the six bits to a scheme which allocates these
7.3. Quantisation of Spectral Amplitudes

bits on a block based scheme. The method to attempt this is known here as Block Amps
Quantisation (BAQ), the next section will describe the steps taken to implement this
idea.

7.3.1 Algorithm Specifics

An amplitude peak picking scheme was first implemented in [34] for the quantisation
of spectral amplitudes in the SB-LPC. This method selected a number of spectral
amplitudes according to perceptual importance, the other amplitudes are deemed to
be of little importance perceptually and are set to one. The selected amplitudes are
chosen as:

1. The first two spectral amplitudes since LP modeling can be poor in this area and
lower frequencies are more important perceptually

2. Spectral amplitudes under a peak in the LPC filter frequency response ensuring
formants are well represented.

3. Spectral amplitudes either side of the peaks in the spectrum due to errors in LSF
quantisation which can make the formant positions shift by a few samples

4. The remaining spectral amplitudes corresponding to valleys in the spectrum are
considered to be unimportant perceptually and are set to one.

This method of selecting the amplitudes forms the basis of the peak picking in the BAQ
method. It is considered for a 10th LPC filter there are only five peaks available for a
cycle of speech as each pair of LPC coefficients describes one formant. The maximum
number of peaks Peak_max which can be picked for each cycle is set at five. The first
two amplitudes are always selected and five peaks plus the two amplitudes either side
of these five peaks which gives a maximum total of seventeen amplitudes picked.

The number of selected amplitudes allocated per frame is set as Ampsfr. For example,
if Ampsfr is set to thirty and there are three cycles in the frame, then the number of
amplitudes per cycle Ampscycle is 10. If the number of peaks found in each cycle is 3 then
this results in eleven amplitudes being picked - the first two plus the three peaks with
one either side - per cycle and the total number of amplitudes for the frame is thirty three. As in this example $Amps_{fr}$ is set to thirty therefore three amplitudes must be de-selected. The amplitudes to be removed are selected from the following list which shows the degree of importance. For example we would start at the bottom of the list and move up until the correct number has been removed. If the number of amplitudes picked is less than $Amps_{fr}$ then more amplitudes are selected according to the list.

1. First two spectral amplitudes per cycle.
2. The peaks in the cycle.
3. Amplitudes either side of the peaks.
4. Further amplitudes either side of peaks.

If the variation in cycle sizes per frame was found to be greater then six according to (6.7) then the number of amplitudes per cycle are found from a simple ratio of the number of harmonics per cycle compared to the total number of harmonics for the frame, the cycle with the largest number of harmonics is given the largest value of $Amps_{cyc}$.

![Figure 7.4: Average MSE per cycle with variation in the number of unquantised amplitudes selected per frame](image)

Figure 7.4: Average MSE per cycle with variation in the number of unquantised amplitudes selected per frame
Before quantisation can take place it is important to determine what is the optimum number of $Amps_{fr}$ that can be quantised without causing significant distortion. The value of $Amps_{fr}$ was varied in the range of 10 to 50 in the speech coder and the average MSE per cycle measured using unquantised values of $\hat{z}$ in (7.3), the results are shown as Figure 7.4.

As expected as the number of amplitudes per frame increases the MSE falls in value. The steepness of the curve begins to decrease in the region of 25 amplitudes onwards. The synthetic speech produced was evaluated perceptually for the various values of $Amps_{fr}$, it was found that only a small amount of distortion was present when a value of 30 amplitudes per frame was selected, when a value of 50 amplitudes was selected there was little or no discernible distortion present in the unquantised BAQ method.

### 7.3.2 MSVQ Experiments

Various bit rate configurations were trained using MSVQ routines, they are shown in Table 7.1. They were implemented in the coder using the M-best tree search of Section 3.7.2.2 where M was set to a value of 8 and the average MSE per cycle was found using (7.3) for various values of selected amplitudes per frame. The MSE results are shown as Figure 7.5. Figure 7.5 shows that initially 26 amplitudes per frame gives the lowest average MSE per cycle. But as the bit rates for 30 and 34 amplitudes per

<table>
<thead>
<tr>
<th>Number of Bits</th>
<th>Bit Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>6,4,4,4,4</td>
</tr>
<tr>
<td>24</td>
<td>6,6,4,4,4</td>
</tr>
<tr>
<td>26</td>
<td>6,6,6,4,4</td>
</tr>
<tr>
<td>28</td>
<td>6,6,6,6,4</td>
</tr>
<tr>
<td>30</td>
<td>6,6,6,6,6</td>
</tr>
<tr>
<td>32</td>
<td>8,6,6,6,6</td>
</tr>
<tr>
<td>34</td>
<td>8,8,6,6,6</td>
</tr>
<tr>
<td>36</td>
<td>8,8,8,6,6</td>
</tr>
</tbody>
</table>

Table 7.1: MSVQ bit allocation for Figure 7.5
7.3. Quantisation of Spectral Amplitudes

frame respectively reach 1 bit per amplitude they give a superior performance.

![Figure 7.5: Variation in average MSE per cycle at various bit rates during testing](image1)

![Figure 7.6: MSE variation in quantiser training for BAQ methods in Table 7.2](image2)

It was found previously that at a bit rate of 1 bit per amplitude was optimum when
quantising spectral amplitudes in the SB-LPC [34]. When these values were evaluated perceptually little difference could be found between selecting 30 and 34 amplitudes per frame. Therefore 30 spectral amplitudes per frame at a resultant bit rate of 1 bit per amplitude was selected for the BAQ method.

To compare the relative performance of the quantisation configurations a spectral amplitude database of 50,000 sets was used. For the purpose of comparison this size of speech database is effective and should provide reliable results. For a given bit rate the MSVQ can differ in stages. The structure of the quantiser affects complexity and memory storage and affects performance. A lower performance usually results when more structure is imposed on the codebooks but at the benefit of reduced complexity and storage. MSVQ quantisers have been trained all using 30 bits for various stages from 4 to 6. The training results are plotted in Figure 7.6 for the BAQ configurations shown in Table 7.2. It can be seen from Figure 7.6 that as more structure is imposed on the codebooks the error rises during the training process.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of stages</th>
<th>Bit allocation</th>
<th>Complexity</th>
<th>Memory</th>
<th>MSE testing</th>
<th>MSE testing interleaved</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAQ 4</td>
<td>8, 8, 8, 6</td>
<td>145920</td>
<td>24960</td>
<td>0.0983</td>
<td>0.0914</td>
<td></td>
</tr>
<tr>
<td>BAQ 5</td>
<td>6,6,6,6,6</td>
<td>63360</td>
<td>9600</td>
<td>0.1031</td>
<td>0.0952</td>
<td></td>
</tr>
<tr>
<td>BAQ 6</td>
<td>5,5,5,5,5,5</td>
<td>39360</td>
<td>5760</td>
<td>0.1068</td>
<td>0.0998</td>
<td></td>
</tr>
<tr>
<td>JQI 3+1</td>
<td>8, 8, 8 + 6</td>
<td>130560</td>
<td>23040</td>
<td>0.1054</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.2: Comparison of MSVQ structures when quantising spectral amplitude information at a 30-bit bit rate.

The memory and complexity requirements in Table 7.2 are found from (3.47) and (3.50) respectively. The number of stages in (3.50) during training was set at M equal to 8. A comparison can be made between the methods shown here and the JQI method of quantising the amplitudes, these are shown in Table 7.2. The MSE results are found using (7.3). For comparison the MSE results are also shown for the JQI method at 24 bits plus 6 bit shape vector. The BAQ method at four and five stages gives lower error values than the JQI method. The BAQ method at six stages gives a similar error value to JQI but at a much lower memory and complexity requirement.
7.3.3 Interleaving of Spectral Amplitudes

During peak picking several selected amplitudes per cycle are selected for quantisation. These selected amplitudes from each cycle are placed into an array and vector quantised. Such arrays are demonstrated in parts (a) and (b) of Figure 7.7 which shows the selected amplitudes for frames of two and three cycles respectively.

<table>
<thead>
<tr>
<th>Frame 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle A</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>A1</td>
</tr>
<tr>
<td>Selected Amplitudes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frame 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle A</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>A1</td>
</tr>
<tr>
<td>Selected Amplitudes</td>
</tr>
</tbody>
</table>

Figure 7.7: Example array of selected amplitudes for quantisation for (a) frame of two and (b) three cycles.

<table>
<thead>
<tr>
<th>Interleaved Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected Amplitudes</td>
</tr>
<tr>
<td>(a)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interleaved Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dequantised Interleaved</td>
</tr>
<tr>
<td>Values</td>
</tr>
<tr>
<td>(b)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dequantised Selected Amplitudes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c)</td>
</tr>
</tbody>
</table>

Figure 7.8: Interleaving of selected amplitudes. (a) The array is interleaved before quantisation and (c) the dequantised values.
These arrays are fairly uncorrelated to each other as amplitudes of expected similar
values are placed at different points in the arrays to be quantised. For example in
the peak picking algorithm the first two amplitudes are always selected. The values
therefore of A1, A2, B1, B2, C1 and C2 in frame 1 are at different points to A1, A2, B1
and B2 of frame 2. As the number of cycles per frame can vary from 1 to 12 and 50,000
amplitude vectors are used for quantisation training this factor may have a considerable
effect on quantisation performance.

It would be better to group similar amplitudes together before quantisation is carried
out. This procedure is illustrated in parts (a) and (b) of Figure 7.8 where values
which are likely to be similar are grouped together before quantisation by interleaving.
This would produce arrays of spectral amplitudes which are more highly correlated for
frames of varying cycle numbers. The interleaved MSE results from testing are shown
in Table 7.2 there is a clear improvement in all cases when the spectral amplitudes are
interleaved before quantisation.

Interleaving substantially improves the MSE error results. After interleaving all num-
ber of stages of the BAQ method give lower error values than the JQI method. When
examined perceptually the interleaved BAQ method with the number of stages at four
and five gave a similar level of performance. Given that a five stage interleaved BAQ
method gives a considerable complexity and memory saving against a four stage im-
plementation, the five stage BAQ method with interleaving was chosen as the method
to be used in the PS SB-LPC for quantising the spectral amplitudes.

7.4 Bit Allocation of Coder

This chapter has detailed the successful design and implementation of spectral am-
plitude quantisation routine. This routine can be integrated along with the other
quantised parameters detailed in the Section 4.5.3. A suggested bit rate allocation for
this coder is presented in Table 7.3.
7.5. Concluding Remarks

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Bits per 20ms frame</th>
<th>kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC</td>
<td>36</td>
<td>1.8</td>
</tr>
<tr>
<td>Spectral Amplitudes</td>
<td>30</td>
<td>1.5</td>
</tr>
<tr>
<td>PCW pitch length and voicing</td>
<td>16</td>
<td>0.8</td>
</tr>
<tr>
<td>Energy</td>
<td>14</td>
<td>0.7</td>
</tr>
<tr>
<td>Total</td>
<td>96</td>
<td>4.8</td>
</tr>
</tbody>
</table>

Table 7.3: Example bit allocation for 4.8 kbps PS SB-LPC

7.5 Concluding Remarks

This chapter has detailed the steps taken to carry out quantisation of pitch synchronous spectral amplitude information in a sinusoidal coder. Previous research in the area was presented and compared to current research. A new quantisation method was presented, known as Block Amplitude Quantisation (BAQ). When this method was used to quantise the spectral amplitude information in the PS SB-LPC significant benefits over previous methods were found. By carrying out interleaving on sets of the spectral amplitude information before quantisation, the sets became more highly correlated which resulted in more efficient quantisation.
Chapter 8

Coder Quality Testing

This thesis has presented advanced signal processing techniques for pitch synchronous speech coders. The techniques described have been applied to the PS SB-LPC speech coder. This chapter describes the evaluation of these new coding techniques against CELP based and other sinusoidal speech coders.

8.1 Listening Tests

8.1.1 Informal Listening Tests

Informal listening tests were used extensively during the development of the new version of the PS SB-LPC speech coder. They were used to evaluate developments in the signal processing techniques and also to compare with the previous version of the PS SB-LPC. From the informal tests it was concluded that there were extensive improvements over the previous PS coder. Artefacts that were present in the pitch and voicing algorithms have been substantially reduced albeit at the cost of increased complexity. Improvements to the quantisation techniques have been made which means the quantised speech coder can be fully evaluated by formal listening tests.
8.1. Listening Tests

8.1.2 Formal Listening Tests

Formal listening tests are a way of evaluating the speech quality provided by speech coders. A standard which defines a set of tests and conditions is available in [69]. These tests are typically carried out in an acoustically treated test room with large numbers of test subjects and speech files. Tests here have been carried out which have followed these methods as closely as possible given the resources available. For example, the tests were carried out in a quiet office environment which may be expected to have a limited impact on the final results.

MOS Tests

To formally evaluate the new version of the PS SB-LPC, MOS tests were carried out as described in Section 2.2.1, speech processed by the PS SB-LPC and various coders were evaluated. Several coders were included in the test, these coders were:

1. G.729 at 8 kbps [11]. This is widely recognised as a toll quality CELP based speech coder used in IP networks.
2. AMR coder at rates of 12.2, 7.4 and 4.75 kbps [4]. This a multi rate CELP based speech coder employed in GSM and 3G networks.
3. G723.1 at 5.3 kbps [23]. A CELP based speech coder frequently used in IP networks.
4. MELP at 2.4 kbps [44]. A low bit rate sinusoidal based speech coder primarily used in military applications.
5. PS SB-LPC v1 UQ (Unquantised) [40]. This is the starting coder for this project, tested in its unquantised configuration.
6. PS SB-LPC v2 Q (Quantised). The coder produced in this project, quantised at a 4.8 kbps bit rate. For male speech an average of 4 percent of the total frames were synthesised using the Reduced Search method of Section 6.5.4, for female speech this figure was 7 percent. The remaining frames were synthesised using the Full Search method of Section 6.5.2 with coder configuration D of Section 6.5.3.
7. PS SB-LPC v2 UQ. This is the same as immediately above but unquantised.
8.1. Listening Tests

The tests were prepared as follows; Four files of speech were used from the NTT database [57], each file contained two sentences of speech. Two male and two female speakers were used, the files were played in random order to eighteen listeners. The average age of the listeners was approximately twenty five years of age. Each of the listeners were asked to give a score to the samples according to the following MOS scale:

- 5 - The speech quality is excellent.
- 4 - The speech quality is good.
- 3 - The speech quality is fair.
- 2 - The speech quality is poor.
- 1 - The speech quality is bad.

The MOS test results are shown in Table 8.1 and in a bar chart form in Figure 8.1. The test results demonstrate that the signal processing techniques described in this thesis have significantly improved the synthetic speech quality produced by the PS SB-LPC speech coder. PS SB-LPC v1 UQ had a quality similar to that of MELP 2.4 kbps another low rate sinusoidal speech coder however PS SB-LPC v2 UQ had a comparable speech quality to the CELP based coder AMR 7.4 kbps, indeed its UQ speech quality was rated near to that of the toll quality speech coder G729.

<table>
<thead>
<tr>
<th>Coder/Sample</th>
<th>AMR 12.2</th>
<th>AMR 7.4</th>
<th>AMR 4.75</th>
<th>AMR 8.0</th>
<th>G729 2.4</th>
<th>G723.1 5.3</th>
<th>PS v1 UQ</th>
<th>PS v2 UQ</th>
<th>PS v2 4.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>male</td>
<td>4.30</td>
<td>3.72</td>
<td>3.30</td>
<td>3.90</td>
<td>3.13</td>
<td>3.20</td>
<td>3.11</td>
<td>3.76</td>
<td>3.48</td>
</tr>
<tr>
<td>female</td>
<td>4.14</td>
<td>3.68</td>
<td>3.10</td>
<td>3.91</td>
<td>3.02</td>
<td>3.30</td>
<td>3.18</td>
<td>3.65</td>
<td>3.40</td>
</tr>
<tr>
<td>average</td>
<td>4.22</td>
<td>3.70</td>
<td>3.20</td>
<td>3.91</td>
<td>3.08</td>
<td>3.25</td>
<td>3.15</td>
<td>3.71</td>
<td>3.44</td>
</tr>
<tr>
<td>95 % C/I</td>
<td>0.15</td>
<td>0.14</td>
<td>0.16</td>
<td>0.11</td>
<td>0.19</td>
<td>0.17</td>
<td>0.15</td>
<td>0.13</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 8.1: MOS scores for tested coders, all bit rates shown are in kbps. The 95 % Confidence Interval is also shown. PS corresponds to the PS SB-LPC coder.

When quantised the synthetic speech quality produced by the new PS SB-LPC v2 Q at 4.8 kbps was still rated more highly than the PS SB-LPC v1 UQ and higher than that
of MELP at 2.4 kbps. Though quantised the speech quality was not rated as highly as AMR 7.4 kbps - a higher rate speech coder - but was still rated better than a higher bit rate coder, namely G723.1 at 5.3 kbps and considerably higher than AMR at 4.75 kbps.

![Figure 8.1: Average MOS scores for tested coders shown in Table 8.1](image)

8.2 Test Results Summary

Quality assessment of the new PS SB-LPC model shows that significant improvements have been made over the original PS SB-LPC version. The majority of artefacts that cause quality degradation have been identified and removed; when evaluated in its unquantised configuration the final speech quality was greatly improved. Improvements in the quantisation process allowed the PS SB-LPC to be fully evaluated at a quantisation rate of 4.8 kbps. When tested against some widely used speech coders the quantised speech quality was rated more highly than various CELP based speech coders operating at similar and higher bit rates.
8.3 Concluding Remarks

This chapter has presented an overview of the test procedures for evaluating the synthetic speech quality of speech coders. These test procedures were used to evaluate the synthetic speech quality of a PS sinusoidal speech coder utilising the new parameter extraction and quantisation routines developed in Chapters 5, 6 and 7. Formal listening tests showed that when compared to various standardised speech coders this coder can provide for high quality synthetic speech.
Chapter 9

Conclusions

9.1 Aims

The subject of this thesis has been the development of a high quality PS sinusoidal speech coder. This work has focused on the following areas:

- To determine the cause and possible solutions to the problems in a PS sinusoidal speech coder namely the PS SB-LPC which was used as a basis for the work in this thesis. By investigating the pitch and voicing algorithms employed by this coder the errors which put a upper limit on the speech quality were quantified in a thorough and systematic manner.

- The development of new parameter extraction schemes to remove the upper limits on the quality of the speech produced by the PS SB-LPC, focusing primarily on improving and developing new pitch and voicing algorithms.

- To enable the PS SB-LPC to operate in a practical manner, the extracted parameters require quantisation. These quantisation techniques determine the final bit rate and speech quality. Quantisation techniques were introduced to enable parameters extracted pitch synchronously to be quantised with minimum distortion.
9.2 Concluding Overview

Chapter 5 presented an investigation into the factors limiting the synthetic speech quality produced by the PS SB-LPC. It was determined that the artifacts in the coder were produced by errors in the pitch and voicing algorithms. To remove the voicing errors present, a GUI tool known as the Bit Stream Editor was applied to the PS SB-LPC and developed to form the basis of an advanced voicing classification technique. This new technique gave good results when compared to a traditional method employed by a standard TS sinusoidal speech coder.

Chapter 6 focused on improvements to the pitch information extraction algorithms. Existing open loop methods were introduced and applied to the PS SB-LPC. In order for a closed loop solution to be implemented a method of producing an approximately zero phase speech signal from original speech known as Zero Phase Equalisation (ZPE) was utilised. Full and Reduced Search closed loop waveform matching techniques were successfully developed and applied to this phase removed signal and a good replication of the pitch information in the original speech signal was produced.

In Chapter 7 the quantisation of spectral amplitude information was investigated. Sets of spectral amplitude information were quantised using a new block based scheme known as Block Amplitude Quantisation (BAQ), interleaving techniques were employed to further improve quantisation efficiency. When compared to a Joint Quantisation Interpolation (JQI) method in a PS sinusoidal speech coder this method produced superior results.

Chapter 8 presented good formal test results when the new configuration PS SB-LPC was compared with other sinusoidal and CELP based coding paradigms. The PS SB-LPC was evaluated unquantised and quantised at 4.8 kbps.

9.3 Future Work

The pitch detection algorithm is one of the limiting factors in the quality of the unquantised synthetic speech produced by the SB-LPC. Many of the functions present
in the coder were developed for open loop operation and subsequently not suitable for a computationally complex implementation that is required for closed loop matching operation. A investigation into reducing the complexity of this matching process should increase the final speech quality.

Currently the method of producing the approximately ZPE signal relies on finding the highest energy cycle per frame of input speech. This can cause problems when subsequent cycles are of similar energy but dissimilar phase spread. Methods to select the best cycle per frame based on phase content of the cycles may improve the phase removal process as more suitable cycles may be selected.

The limiting factor in the quantised synthetic speech quality is the quantisation of the spectral envelope i.e. LSFs and spectral amplitudes. By operating on a PS fashion more information has too be quantised therefore JQI and block based schemes were designed and implemented to enable quantisation of the spectral envelope at a fixed rate. Further investigation into these techniques should allow for the quantised synthetic speech quality to approach that of the unquantised configuration.

Sinusoidal coders may be a good alternative to CELP based coders in VoIP systems due to their higher resilience to packet losses. The coder presented here has increased the synthetic speech quality of a sinusoidal based coder and may find application in such a scheme.
Appendix A

List of Publications

Appendix B

List of Abbreviations

AaS  Analysis and Synthesis
AbS  Analysis by Synthesis
ACELP Alebraic Code Excited Linear Prediction
ADPCM Adaptive Differential Pulse Code Modulation
AM   Autocorrelation Method
AMDF Adaptive Magnitude Difference Function
AMR  Adaptive Multi Rate
BAQ  Block Amps Quantisation
BSE  Bit Stream Editor
CELP Codebook Excited Linear Prediction
CM   Covariance Method
DFT  Discrete Fourier Transform
DTMF Dual Tone Multi-Frequency
ETSI European Telecommunications Standards Institute
FFT  Fast Fourier Transform
FIR  Finite Impulse Response
FR-GSM Full Rate GSM
GSM  Global System for Mobile Communication
HR-GSM Half Rate GSM
I-MBE Improved Multi Band Excitation
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITU</td>
<td>International Telecommunications Union</td>
</tr>
<tr>
<td>INMARSAT</td>
<td>International Maritime Satellite</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>JQI</td>
<td>Joint Quantiser Interpolator</td>
</tr>
<tr>
<td>LBG</td>
<td>Linde Buzo Gray algorithm</td>
</tr>
<tr>
<td>LD-CELP</td>
<td>Low Delay Codebook Excited Linear Prediction</td>
</tr>
<tr>
<td>LP</td>
<td>Linear Prediction</td>
</tr>
<tr>
<td>LPC</td>
<td>Linear Predictive Coding</td>
</tr>
<tr>
<td>LSF</td>
<td>Line Spectral Frequencies</td>
</tr>
<tr>
<td>LSP</td>
<td>Line Spectral Pairs</td>
</tr>
<tr>
<td>LTP</td>
<td>Long Term Prediction</td>
</tr>
<tr>
<td>MBE</td>
<td>Multi Band Excitation</td>
</tr>
<tr>
<td>MELP</td>
<td>Mixed Excitation Linear Prediction</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean Opinion Score</td>
</tr>
<tr>
<td>MTE</td>
<td>Modified Time Envelope</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MSVQ</td>
<td>Multi Stage Vector Quantisation</td>
</tr>
<tr>
<td>PCM</td>
<td>Pulse Code Modulation</td>
</tr>
<tr>
<td>PCW</td>
<td>Pitch Cycle Waveform</td>
</tr>
<tr>
<td>PDA</td>
<td>Pitch Detection Algorithm</td>
</tr>
<tr>
<td>PS</td>
<td>Pitch Synchronous</td>
</tr>
<tr>
<td>PS SB-LPC</td>
<td>Pitch Synchronous Split Band Linear Predictive Coder</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Service Telephone Network</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Square Error</td>
</tr>
<tr>
<td>SB-LPC</td>
<td>Split Band Linear Predictive Coder</td>
</tr>
<tr>
<td>SSMM</td>
<td>Sinusoidal Speech Model Matching</td>
</tr>
<tr>
<td>SVQ</td>
<td>Split Vector Quantisation</td>
</tr>
<tr>
<td>TS</td>
<td>Time Synchronous</td>
</tr>
<tr>
<td>VQ</td>
<td>Vector Quantisation</td>
</tr>
<tr>
<td>V</td>
<td>Voiced</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>UV</td>
<td>Unvoiced</td>
</tr>
<tr>
<td>ZPE</td>
<td>Zero Phase Equalised</td>
</tr>
</tbody>
</table>
References


References


References


References


References


References


References


