Low Bit Rate Speech Coding

Carl Kritzinger

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Engineering Sciences at the University of Stellenbosch

Promoter: Dr. T.R. Niesler

April 2006

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature

Date

Abstract

Despite enormous advances in digital communication, the voice is still the primary tool with which people exchange ideas. However, uncompressed digital speech tends to require prohibitively high data rates (upward of 64kbps), making it impractical for many applications.

Speech coding is the process of reducing the data rate of digital voice to manageable levels. Parametric speech coders or *vocoders* utilise a-priori information about the mechanism by which speech is produced in order to achieve extremely efficient compression of speech signals (as low as 1 kbps).

The greater part of this thesis comprises an investigation into parametric speech coding. This consisted of a review of the mathematical and heuristic tools used in parametric speech coding, as well as the implementation of an accepted standard algorithm for parametric voice coding.

In order to examine avenues of improvement for the existing vocoders, we examined some of the mathematical structure underlying parametric speech coding. Following on from this, we developed a novel approach to parametric speech coding which obtained promising results under both objective and subjective evaluation.

An additional contribution by this thesis was the comparative subjective evaluation of the effect of parametric speech coding on English and Xhosa speech. We investigated the performance of two different encoding algorithms on the two languages.

Opsomming

Ten spyte van enorme vordering in digitale kommunikasie is die stem steeds die primêre manier waarmee mense idees wissel. Ongelukkig benodig digitale spraakseine baie hoë datatempos, wat dit onprakties maak vir menigte doeleindes.

Spraak kodering is die proses waarmee die datatempo van digitale spraakseine verminder word tot bruikbare vlakke. Parametriese spraakkodeerders oftewel *vocoders*, gebruik voorafbekende informasie oor die meganisme waarmee spraak produseer word om besonder doeltreffende kodering van spraak seine te verrig (so laag soos 1kbps).

Die meerderheid van hierdie tesis bevat 'n studie oor parametriese spraak kodering. Die studie bestaan uit 'n oorsig van die wiskundige en heuristieke tegnieke wat in parametriese spraak kodering gebruik word sowel as 'n implementasie van 'n aanvaarde standaard algoritme vir spraak kodering.

Met die oog op moontlike maniere om die bestaande kodeerders te verbeter, het ons die wiskundige struktuur onderliggend aan parametriese spraak kodering ondersoek. Hieruit spruit 'n nuwe algoritme vir parametriese spraak kodering wat onder beide objektiewe en subjektiewe evaluering belowende resultate gelewer het.

'n Verdere bydrae van die tesis is die vergelykende subjektiewe evaluering van die effek van parametriese kodering van Engelse en Xhosa spraak. Ons het die doeltreffendheid van twee verskillende enkoderings algoritmes vir die twee tale bestudeer.

To my father, for his quiet greatness.

Contents

List of Figures

List of Tables

Symbols

H	Entropy
$H(z)$	Transfer Function of a Linear System
R	Information Rate
Φ	Covariance Matrix of a Random Process
$s[n]$	Speech Signal
$s'[n]$	Approximation to the Speech Signal $s[n]$
$e[n]$	Error signal
$w[k]$	Windowing function e.g. Hamming window.
z	complex number
E	Error (Usually a scalar function of a vector space.)

Acronyms

ACF	Auto-Correlation Function
ACR	Absolute Category Rating
AMDF	Amplitude Magnitude Difference Function
ARMA	Auto Regressive Moving Average
ASE	Adaptive Spectral Enhancement
AST	African Speech Technologies
CELP	Codebook Excitation with Linear Prediction
DC	Direct Current
DFT	Discrete Fourier Transform
DRT	Diagnostic Rhyme Test
DSP	Digital Signal Processor
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GSM	Global System for Mobile Communications
HF	High Frequency
IS-MELP	Irregularly Sampled MELP
LAR	Log-Area Ratio
LPC	Linear Predictive Coding
LPC	Linear Predictor Coefficients
LP	Linear Predictor
LSF	Line Spectrum Frequencies
MDF	Magnitude Difference Function
MELP	Mixed Excitation with Linear Prediction
MOS	Mean Opinion Score
MSE	Mean Squared Error
MSVQ	Multi-Stage Vector Quantisation
NATO	North Atlantic Treaty Organisation
PCM	Pulse Code Modulation

PDF	Probability Density Function
PESQ	Perceptual Evaluation of Speech Quality
P	Order of LP system
RCs	Reflection Coefficients
RMS	Root Mean Squared
SD	Spectral Distortion
SNR	Signal to Noise Ratio
STANAG	Standardisation Agreement
STTD	Short Term Temporal Decomposition
TCP/IP	Transmission Control Protocol/Internet Protocol
TIMIT	Texas Instruments and Massachusetts Institute of Technology. Refers to a speech corpus published jointly by these institutions.
VOIP	Voice Over Internet Protocol
VQ	Vector Quantisation

Acknowledgements

- My supervisor, Dr. Thomas Niesler, a huge thanks for his almost infinite patience with my ramblings, for his enormous insight and positive energy and his tranquility.
- Gert-Jan for his relentless enthusiasm, his thesis template and for fearlessly sticking his neck out to get me into the DSP lab in the first place.
- My parents, for their love and support, emotional and financial.
- Trizanne for listening, for being there and for believing in me even when I didn't.
- Stephan, for teaching me about engineering, DSP and pragmatism.
- All the DSP lab rats for being a great bunch of people to be around and for providing an endless supply of laughter, advice and distraction.

Chapter 1

Introduction

Voice recognition systems have become increasingly popular as a means of communication between humans and computers. An excellent example of this is the AST automated reservation system developed at the University of Stellenbosch, which makes hotel reservations over the telephone.

It is a well-known problem that the accuracy of these voice recognition systems is adversely affected by the effects of telephone channels.

Therefore it would be advantageous to be able to use digital voice for the recognition system. This could potentially reduce the amount of training data required by reducing the number of telephone channel conditions which must be catered for. At the same time digital transmission of voice could minimise the transmission channel effects, thus improving the clarity of the input voice and improving the overall recognition accuracy of the system.

This need for digital voice communications suggests the implementation of a voice coder suitable for a Voice Over Internet Protocol (VOIP) system. Recent changes in Telecommunications legislation have made such systems a highly viable proposition [34].

However, most parametric voice coders have been developed within the context of an extremely euro-centric paradigm, with the design and testing of vocoders focusing mainly on the transmission of European languages. The suitability of these voice coders for the transmission of African languages has not been investigated.

The focus of this thesis is the development and testing of voice coding system which caters for the following needs:

- Low rate or multi rate implementation to cater for applications where bandwidth is limited.

- Multi-language compatibility. Most current voice encoding standards are aimed at European languages or American English. The phonemic richness of the African languages pose a potential challenge and the voice coding should be able to handle this.

1.1 History of Vocoders

The fundamental principle underlying vocoders is that low rate phonetic information may be separated from high frequency acoustic information.

This idea is usually attributed to H.W. Dudley. In fact, during the 18th century, Wolfgang von Kempelen built a device which could be operated to produce human-like speech, as described by Dudley himself [22]. Of course, von Kempelen simply envisioned the device as a novelty and a demonstration of his observations of human speech production.

Dudley [21] and later Dunn [23] built speech synthesisers but Dudley seems to have been the first to envision a substantially different application for the speech synthesisers, that of speech transmission. In his 1950 paper, he shows a schematic diagram for a low-bandwidth speech transmission system using a speech synthesiser.

In 1972 Atal and Hanauer proposed voice coding using Linear Prediction Analysis [2], which has become by far the most popular approach to low-rate speech coding. Despite the low bandwidth necessary for speech transmission by this system, it was not widely accepted and in many applications, true low rate vocoders (as opposed to the higher rate 'waveform' coders) did not achieve great popularity. David [24] mentions that

> In spite of their great potential for bandwidth saving in long distance telephony, vocoders have not found widespread acceptance.

Possibly the two greatest factors contributing to this lack of popularity were:

1. Typically, voice transmitted via a vocoder had a distinctly mechanical quality.

2. Vocoders tend to be computationally intensive. Flanagan et. al. [30] noted that computational complexity of vocoders is positively correlated.

In the 1980's vocoders began to achieve recognition. Technology had improved substantially in the field of the low-power digital signal processors (DSP). This means that the computational and memory requirements for a vocoder to run in real-time in a field unit were no longer impossible to meet.

Vocoder usage was further driven by need for secure voice communications. Analog voice is notoriously difficult to encrypt efficiently, whereas digital voice may be easily encrypted to a very high degree of security. Particularly in military applications, the need for security far outweighed the importance of natural sounding voice.

In 1984 this led to the adoption of the first standard for digital voice communication, FS-1015 described a vocoder known as LPC10e.

In subsequent years, vocoder development has been driven by a number of primary applications:

1. Cellular telephones usually have limited bandwidth available for transmission of voice. The massive growth in this market spurred vocoder research during the 1980s and 1990s, leading to the development of several high quality medium rate vocoders, such as the Regular Pulse Excited vocoder specified by the GSM protocols [33]. However, the bandwidth restrictions of cellular network protocols have not necessitated true low-rate vocoders.

2. In long haul communications via high-frequency (HF) radio (HF is the term used to describe the radio spectrum between 3 and 30 MHz), received analog speech is typically extremely poor due to severe transmission channel effects such as interference, noise and signal multi-path. A full treatment of the subject may be found in Betts [7]. However, reliable data transmission is possible even in extremely poor conditions [71]. This means that there is significant scope for digital voice over HF radio links.

3. Voice Over Internet Protocol (VOIP) has been increasing enormously in popularity due to its potential for extremely low-cost long distance telephony [58]. However, the restrictions imposed by TCP/IP protocols means that voice coders must tolerate lost data and long transmission delays as well as varying data throughput, since fluctuations in network load may result in substantial variation in the available bandwidth.

Additionally, vocoders have been used in niche applications such as satellite communications, voice recorders [15] and more esoterically to modulate voices in music [86].

1.2 Objectives

- The main focus of this project will be an investigation of the implementation of a low bit-rate vocoder. By low bit-rate we mean a vocoder which has a bandwidth requirement of at most 2400 bits per second.

- There are currently several published vocoder standards which may be suitable for this application such as LPC, CELP and MELP. The first stage of the thesis will comprise an overview and understanding of the current vocoder standards and the selection of a standard for implementation.

- Once a suitable candidate has been chosen, it will be implemented and evaluated in a high level language such as MATLAB to be used as a reference implementation.

- Once the reference implementation is complete, the reference vocoder is to be thoroughly tested.

- This stage of the project will comprise of an investigation into the shortcomings of the reference implementation and an investigation into potential avenues of improvement.

- Finally, the performance of the reference and improved vocoder designs will be tested to compare their respective performance for various languages. Performance of the vocoders will be measured with subjective listener tests.

1.3 Overview

The structure of this thesis will be as follows:

Chapter 2 In this chapter we will discuss the various approaches which have historically been used to reduce the bandwidth of the speech waveform.

Chapter 3 In this chapter we will discuss the background knowledge essential to the understanding of current voice coding techniques.

Chapter 4 In this chapter we will discuss some current voice coding techniques and standards.

Chapter 5 In this chapter we will present the implementation of a modern vocoder.

Chapter 6 In this chapter we will examine some of the mathematical properties of the operation of a parametric voice coder in an attempt to improve on parametric voice coding.

Chapter 7 In this chapter we will describe the implementation of a variable frame-rate vocoder based on the vocoder described in chapter 4 and chapter 5.

Chapter 8 In this chapter we will present a comparative evaluation of the reference vocoder as well as the improved vocoder.

Chapter 9 In this chapter we will present our conclusions and recommendations for further work.

Chapter 2

An Overview of Voice Coding Techniques

The aim of speech coder is fundamentally that of any data compression system: to represent information as efficiently as possible.

Claude Shannon [76] introduced three fundamental theorems of information. One of these is the source-coding theorem which established a fundamental limit on the rate at which the output of an information source may be transmitted without causing a large error probability.

In the most naive approach, and following the ideas of Shannon, we may regard the speech signal as the random output produced by a source. The source is characterised in two fundamental ways:

- By its alphabet, the set of possible symbols which it can produce.
- By the entropy of the source, which describes how much information it outputs per symbol.

Most of Shannon's work deals with a signal consisting of discrete symbols from a finite alphabet. The speech waveform, however is a continuous time signal. This does not pose an insurmountable difficulty, since speech may be sampled and quantised without significant loss of information, as we will describe in 2.2. The quantised samples may then be regarded as the alphabet of the speech source.

We regard the speech samples as the output of a random source with entropy H. According to Shannon if we then encode the speech so that we transmit information at a rate R, the following will hold:

1. If $R > H$ then it is possible to encode the speech so that the probability of error is arbitrarily small.

2. If $R < H$ then the error probability will be non-zero, regardless of the complexity of our coding algorithm.

Unfortunately, the above *source coding theorem* does not enlighten us as to the actual encoding scheme which we need to use in order to achieve such efficient compression. Voice coders all represent algorithms which attempt to minimise $R - H$ while simultaneously minimising the probability of error.

The way in which this is achieved may be divided into three broad categories:

- Waveform Coders
- Parametric Coders (Vocoders)
- Segmental Coders

It may be useful to suggest a theoretical lower bound on the amount of compression which one can hope to achieve on a speech signal without introducing errors.

The difficult part of this analysis is to determine a meaningful quantity for the entropy of the speech source, H. In the following section, we try to estimate H by using a theoretical construct which we shall call an *Ideal Vocoder*.

2.1 Ideal Voice Coding

We could consider the construction of an ideal voice encoder - one which achieves perfect separation of the information components of a speech signal as well as optimal (in an information-theoretic sense) compression of the information components. To this end, we assume that the information payload of the speech signal can be divided into the following components.

Phoneme information; This would roughly be the textual information which the speech represents.

Prosodic information; This would include such information as : Emotion, inflection, etc.

Speaker information; This would be the components of the waveform needed to characterise the speaker at least as well as could be expected of a good Machine Speaker Recognition system or a human listener.

We assume that each of these information bearing components of the speech waveform is modulated at a different rate. Furthermore we will assume that the three components are statistically independent. The last assumption is perhaps somewhat unrealistic (one would be extremely surprised if the potential prosodic content of the phrases ‘good morning’ and ‘go away now’ were statistically similar in spoken English). However, the assumption makes the analysis much more tractable.

2.1.1 Quantifying the information of the speech signal.

Shannon [76] estimated the entropy of written English as being in the order of approximately 1 bit per character. Making the very rough estimate of using one character per phoneme on average, and using the commonly accepted average phoneme rate [20] of around 20 phonemes per second, we arrive at an estimated entropy of the phoneme information around 20 bits/second.

We assume that the speaker identity remains constant over the duration of a reasonable length of time needed by a person to recognise a specific speaker from an unknown utterance, and that this interval is about 1 second[1]. Furthermore, we assume that the average person is able to distinguish in the region of about 1000 different speakers, thus the recognition of a speaker transmits about 10 bits of information. This means that speaker information is on the order of 10 bits/second.

The prosodic information content is the most difficult to estimate but one would imagine that it is at least that of the phonetic content[2].

Thus, H_{voice}, the information rate of the speech signal, may be calculated from the entropy of the alphabet (H_{alphabet}), prosodic information (H_{prosody}), and speaker information (H_{speaker}). We also use the average phoneme length (T_1) and the amount of time needed to identify the speaker (T_2).

$$H_{\text{voice}} = \frac{H_{\text{alphabet}}}{T_1} + \frac{H_{\text{prosody}}}{T_1} + \frac{H_{\text{speaker}}}{T_2}$$

Where

$$\begin{aligned} H_{alphabet} &= log_2(N_{\text{symbols}}) \\ H_{speaker} &= log_2(N_{\text{speakers}}) \\ H_{prosody} &= log_2(N_{\text{prosody}}) \end{aligned}$$

Now approximate values for the above parameters are given in table 2.1.1:

The information rate of the speech waveform under these conditions can therefore be

[1] One would expect that most people can recognise a known person from about 1 second of speech

[2] Consider the number of different ways in which the single phoneme “Ah” may be phrased

T_1	0.1 sec
T_2	10 sec
N_{alphabet}	26
N_{speakers}	1000
N_{prosody}	26

Table 2.1: *Estimated quantities used to calculate approximate information rate of speech.*

conservatively estimated to be approximately

$$\begin{aligned} H_{\text{voice}} &= \frac{\log_2(26)}{0.1} + \frac{\log_2(26)}{0.1} + \frac{\log_2(1000)}{10} \\ &\approx 2\frac{5}{0.1} + \frac{10}{10} \qquad (2.1) \\ &\approx 100 \text{ bits per second.} \qquad (2.2) \end{aligned}$$

From the above derivation, it would therefore seem unlikely that a vocoder could operate effectively at a data rate substantially less than this.

2.2 Pulse Code Modulation (PCM)

While this is, strictly speaking, a waveform coder as described in 2.3, we will treat it separately, since PCM is fundamental to voice coding and is a component of every digital voice coding scheme. PCM is the name given to the voice coder which performs a simple 2-step encoding of the speech signal.

1. Sampling

2. Quantisation

While the two steps are performed simultaneously by an analog-to-digital converter, it is convenient to think of them as separate stages. In the first stage (sampling), the most important consideration is the Nyquist sampling theorem [62]. This theorem states that we must sample the speech at at least $2F_{max}$ samples/second where F_{max} is the highest frequency present in the analog speech signal. The usual frequency range to which speech may be limited without severe degradation is 4kHz [15], and thus the speech must be sampled at a minimum sampling frequency of 8kHz, in order to prevent significant reduction of the speech quality.

After the sampling has been performed, the individual samples of the waveform cannot be represented with arbitrary precision, and the sampled values must be quantised.

We may regard the quantised signal as the original signal plus an error term.

The theory of scalar quantisation of a random variable is well documented in [67].We call the variable to be quantised x, the quantised approximation to x is x_q and the quantisation error is e. Then

$$x_q = x + e. \tag{2.3}$$

The variance of the error resulting from the quantisation is

$$\begin{aligned} E &= \int_{-\infty}^{\infty} (x - x_q)^2 p(x) \mathrm{d}x && (2.4) \\ &= \int_{-\infty}^{\infty} e^2 p(x) \mathrm{d}x && (2.5) \end{aligned}$$

If we assume that x is distributed on the interval $(-X_{\max}, X_{\max})$, then uniform quantisation of the x using B bits will result in 2^B quantisation intervals of size Δ such that

$$2X_{\max} = 2^B \Delta \tag{2.6}$$

This implies that

$$-\frac{\Delta}{2} < e < \frac{\Delta}{2} \tag{2.7}$$

If, additionally, we assume that x and e are independent (i.e. that $E[xe] = 0$) and that e is uniformly distributed in the interval $(\frac{-\Delta}{2}, \frac{\Delta}{2})$, then equation 2.4 may be written as

$$\begin{aligned} E &= \int_{-\infty}^{\infty} e^2 \mathrm{d}x \int_{-\infty}^{\infty} p(x) \mathrm{d}x && (2.8) \\ &= \int_{-\infty}^{\infty} e^2 \mathrm{d}x && (2.9) \\ &= \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 \mathrm{d}e && (2.10) \\ &= \frac{\Delta^2}{12} && (2.11) \\ &= \sigma_e && (2.12) \end{aligned}$$

Thus, if we apply simple linear quantisation to the samples, we may expect to obtain a SNR of approximately:

$$\begin{aligned} \mathrm{SNR(dB)} &= 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right) && (2.13) \\ &= (20 \log_{10} 2) B + 10 \log_{10} 3 - 20 \log_{10} \left(\frac{X_{\max}}{\sigma_x} \right) && (2.14) \\ &= (20 \log_{10} 2) B + 10 \log_{10} 3 - 20 \log_{10} \left(\frac{4\sigma_x}{\sigma_x} \right) && (2.15) \\ &\approx 6B - 7 && (2.16) \end{aligned}$$

Where B is the number of bits used to quantise each sample.

Thus, for PCM to achieve an acceptable SNR of 40dB and an acceptable bandwidth of 4kHz, we need to use at least 8 bits per sample and a sampling rate of 8kHz for a total of 64kbps.

2.3 Waveform Coders

Waveform Coders attempt to exploit the statistical properties of the waveform in order to achieve a coding gain [67].

Typically, waveform coders attempt to reproduce the speech waveform in such a way that the reconstructed waveform is as close as possible to the original. This is usually quantified by means of the average signal to noise ratio, which is typically measured in decibels (dB). The noise signal ($e[n]$) is defined as the sample-wise difference between the original ($s[n]$) and trans-coded ($s'[n]$) signals.

$$e[n] = s[n] - s'[n] \tag{2.17}$$

Then the signal to noise ratio (SNR) is defined as follows:

$$\mathrm{SNR} = 20\log_1 0\left(\frac{\sum_n (s[n])^2}{\sum_n e[n]}\right) \tag{2.18}$$

SNR is not a robust measure of speech quality but in high bandwidth applications it is generally regarded as sufficiently accurate to provide a means of comparing various encoding schemes [20, 33].

In a completely rigorous sense, any band-limited or sampled (digital) voice transmission system is a type of waveform coder.

More typical examples of waveform coders are PCM such as Differential PCM (DPCM) and Adaptive Differential PCM (ADPCM) as well as Delta Modulation. These are described in detail by Proakis and Salehi [68] as well as Goldberg [33].

The two most significant advantages of waveform coders are their low computational complexity and their ability to compress and represent a wide variety of signals, such as music, speech and side noise. This tends to make waveform coders much more robust against noisy input signals than other vocoders. Waveform coders typically operate effectively on speech in the region between 16 and 256 kbps.

2.4 Parametric Coders

Parametric coders utilise *a priori* information about the physical speech production process in order to achieve far greater coding gains than waveform coders.

Parametric coders are interesting from a psycho-acoustic point of view because, although the coding error between the reconstructed signal and the original signal may be almost as large as the original signal, the original and reconstructed speech signals may be perceptually almost identical. This implies that the SNR is a poor metric to describe the perceptual 'distance' between speech samples.

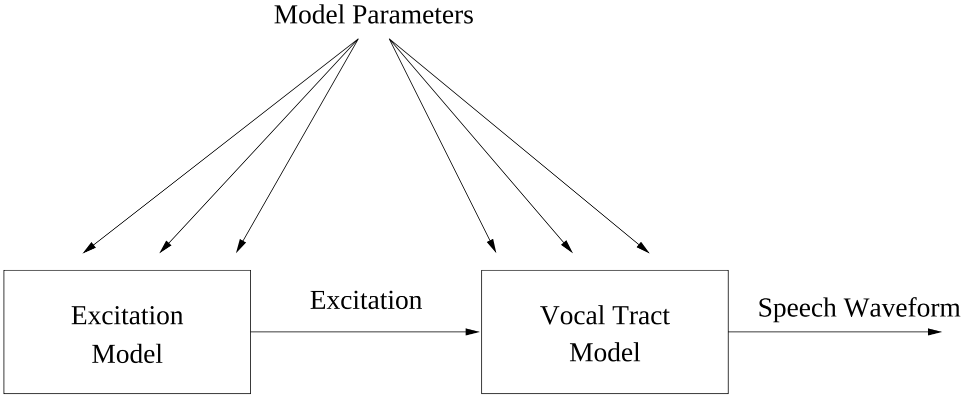


Figure 2.1: *A Typical Model for Parametric Speech Coding*

Parametric voice encoding typically consists of three sub-problems:

1. Estimate the envelope. This corresponds very closely to an estimate of the vocal tract parameters.
2. Estimate the excitation signal. This corresponds closely to an estimate of the nature of the glottal excitation.
3. Quantise each of the former.

2.4.1 Spectrum Descriptions

There are a few popular ways of describing the shape of the spectral envelope:

Homomorphic Vocoders; Homomorphic vocoders use the short term cepstrum to represent envelope information. The idea of using the cepstrum to separate the *low-time* from the *high-time* components of the speech waveform was first proposed by Oppenheim [62].

Formant Vocoders; Formant Vocoders use the positions of the formants to encode envelope information as first investigated by Flanagan [29, 28]. Unfortunately the formants of typical speech waveforms are extremely difficult to track efficiently and are mathematically ill-defined. Various algorithms have been proposed to track formants [54] but these vocoders never became extremely popular .

Linear Predictive Coders; This is the class of parametric vocoder which has found the greatest popularity in the literature and which has also been used in by far the majority of voice coding standards. Linear predictive coders use a digital all-pole filter (linear predictor) to describe the shape of the spectral envelope. Typically the residual (linear predictor error) has about 10dB less power than the original waveform [15].

2.4.2 Excitation Models

Similarly there are several popular ways of describing the excitation signal.

The Buzz-Hiss Model; This model is used explicitly in LPC10e and is the simplest (and also most bit-efficient) of all the models described here. The Buss-Hiss model is the first model which was successfully used in a voice coding standard. In this case the glottal excitation is simply modelled as being a pulse train (voiced) or white noise (unvoiced).

Mixed Buzz-Hiss; This model is a refinement of the above, where a combination of buzz and hiss in various frequency bands is additively combined in order to create the excitation signal. This is one of the most recently developed models and was first proposed by McCree in 1995 [55].

Harmonic excitation; In this model the excitation signal is composed of the sum of a number of sinusoids. This method has been used by Spanias [1] and McAulay [53].

Codebook Excitation; This was first proposed by Schroeder and Atal [5]. The idea is that a large *codebook* of excitation signals is used and the excitation signal in the codebook which most closely matches the glottal excitation is used.

2.5 Segmental Coders

Segmental coders are an attempt to encode the speech waveform at a much lower resolution - with the basic unit of information being regarded as being on the phoneme length scale rather than on a sample length scale. Segmental coders have only recently begun to receive attention, possibly because their high computational cost has meant that only recently has there been any possibility of implementation of a segmental coder in a functional real-time system.

However, segmental coders show great promise in terms of ultra-low bit rate speech coding. Cernocky [11, 12] is showing very promising results using this approach. However, thus far his vocoders have only operated on single-speaker speech corpora which is a situation far removed from a practical voice coding system.

In segmental voice coding, feature vectors are calculated for segments of the speech signal. These feature vectors are compared to the pre-calculated feature vectors for segments of speech speech in a database. The index of the segment in the database which is closest to the original segment is transmitted. To recreate the speech signal, the successive transmitted indices are decoded to speech segments which are then concatenated. This is illustrated diagramatically in figure 2.2. In the most extreme cases the encoder effectively becomes a speech-to-text converter and the decoder a text-to-speech system, as described in [49].

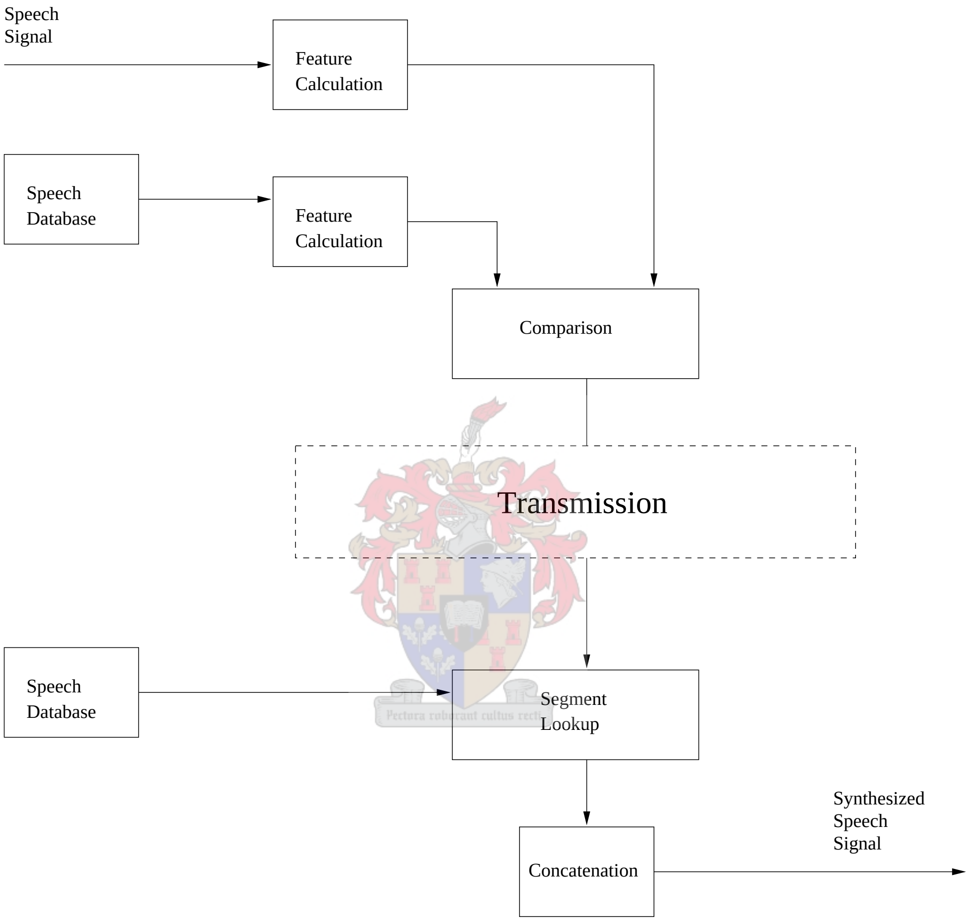


Figure 2.2: *A schematic representation of segmental voice coding.*

Segmental coders are typically very efficient in terms of the compression which is achieved, data rates as low as 200bps are claimed by Cernocky, but the computational cost associated with the search through the database of speech segments means that real time implementations of high quality segmental vocoders is not currently feasible.

Chapter 3

Fundamentals of Speech Processing for Speech Coding

3.1 The Mechanics of Speech Production

On the highest level speech communication is an exchange of ideas between people. To understand speech compression one must understand the mechanism by which these ideas are transferred from one person to another.

Human speech production begins with an idea at the speaker. The idea is translated to a series of words which are in turn converted into a phoneme sequence. Neural impulses transmit the phoneme sequence to the speech organs, which induces a series of muscle actions producing sound. The way in which the speech sounds are produced determines the many of the characteristic properties of the speech waveform. Thus we examine the mechanism by which speech is produced and those aspects of human physiology which play a role in the production of speech.

3.1.1 Physiology

The primary components of the human speech production system are:

Lungs; The lungs produce the airflow and thus the energy required to generate vocal sound.

Trachea; Conveys air from the lungs to the larynx.

Larynx; The main organ of voice production. The larynx provides periodic excitation to the system for sounds that are referred to as *voiced*.

Pharyngeal Cavity, Oral Cavity and Nasal Cavity; These comprise the main organ of modulation of the speech waveform. This will be described in more detail in the next section.

The human speech production system also contains the following finer structures which contribute to finer modulation of the speech. These include:

Soft Palate; The soft palate regulates the flow of air through the nasal cavity in order to alternate between nasalised and non-nasalised sounds.

Tongue, Teeth and Lips; These organs contribute to the general shape of the vocal tract. They are also used to form the class of phonemes referred to as *plosives* (see 3.2.1).

3.2 Modelling Human Speech Production

In order to efficiently compress the speech waveform, it is necessary to understand how it is produced, since this will ultimately determine the properties of the time-domain waveform. Since the human speech production system is extremely complex it is highly desirable to reduce this complexity to a model which is simple enough to allow thorough analysis.

In almost every book on speech processing reference is made to the so called *engineering* or *source-filter* model of speech production. This ubiquitous model is illustrated in figure 3.1.

The fundamental idea of the source-filter model is to reduce speech production to 2 independent components, namely:

1. A source, which produces the signal energy. The excitation energy is almost always generated in such a way that its spectrum is approximately flat. This component corresponds to the function of the larynx or glottis in actual speech production.

2. A modulator component which 'shapes' the spectrum of the excitation. This corresponds in the physical system to the vocal and nasal tract.

The most common model for the vocal tract is, the so-called *lossless multi-tube model.* This means that the vocal tract can be modelled as a series of concatenated open tubes.

The transfer function for a single lossless tube in the complex plane (z) can be shown to be [20]:

$$H(z) = \frac{1}{\cos(\frac{zl}{c})}$$

With l the tube length and c the speed of sound in air (340m/s)

After a substantial but very standard derivation (see [20, 15]) the transfer function of a P section lossless multi-tube system is found to be:

$$H(z) = \frac{z^{-P/2}\prod_{k=1}^{P}(1+\rho_k)}{1-\sum_{k=1}^{P} b_k z^{-k}}$$

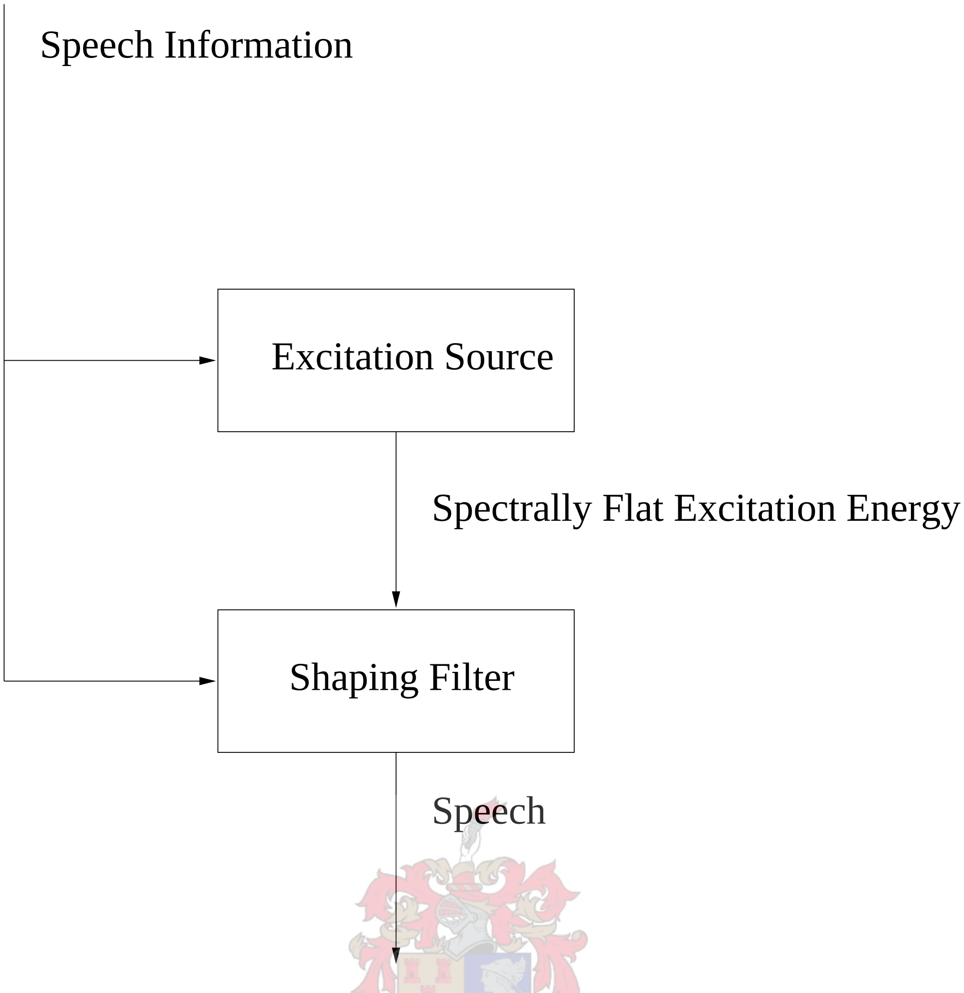


Figure 3.1: *Engineering Model of Speech Production*

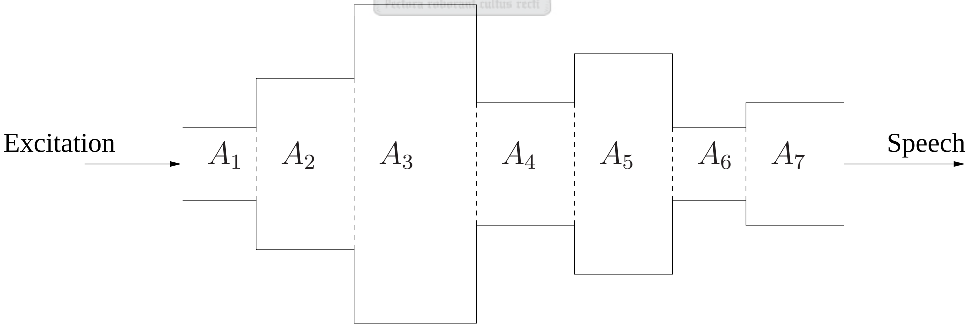


Figure 3.2: *Diagram of the Lossless Tube Model of the Vocal Tract*

Which is usually simplified to

$$H(z) = \frac{H_0}{1 - \sum_{k=1}^{P} b_k z^{-k}} = \frac{H_0}{\prod_{k=1}^{P}(1 - \rho_k z^{-1})}$$

This kind of transfer function is characterised by strong resonances at certain frequencies, with the characteristic resonance frequencies referred to as *formants*.

This model is extremely simple, yet it has become almost ubiquitous. As the following chapters will indicate, this model has formed the basis of an enormous amount of research in the field of speech coding. Yet it does not cater well for certain aspects of speech production, most notably *nasalisation* and *fricatives*. Deller [20] commented on this, referring to attempts to enhance the performance of the model:

> Unfortunately these methods have generally been introduced to address the existing limitations of the present digital filter model for speech coding and have only partially addressed the need for formulating improved human speech production models.

3.2.1 Excitation

While the excitation makes a key contribution toward synthesising ‘natural’ sounding speech, the spectral envelope is usually the dominant feature used by both humans and machines in phoneme classification and speaker recognition. Thus, for the purposes of voice coding it is generally considered sufficient to describe the excitation of a phoneme as being in one of the following classes:

Voiced; Periodic movement of the vocal folds resulting in a stream of quasi-periodic puffs of air.

Unvoiced; Noise-like turbulent airflow through a constriction.

Plosive; Release of pressure built up behind a completely closed portion of vocal tract.

Whisper; Air forced through partially open glottis to excite an otherwise normally articulated utterance.

In [20] a much more exact categorisation of excitation is enumerated. The range of excitation types described here is sufficient for the low-rate, low-quality approach to speech coding taken in this thesis.

3.3 Psycho-Acoustic Phenomena

In order to design efficient voice coders it is of immeasurable utility to understand how humans perceive sounds. This allows us to design voice coders in such a way that the information content of inaudible sounds is not encoded. Furthermore, an understanding of the way in which complex sounds are processed will lead to the development of more accurate methods of characterising how the transmitted speech will be perceived by a listener. This idea is described in more detail in 3.7.

3.3.1 Masking

Frequency masking is the term commonly used for the phenomenon which occurs when certain sounds are rendered inaudible by other sounds, usually closely spaced in frequency and of greater amplitude. The generally used model is that of a triangular masking curve around each frequency in the spectrum. In other words, any tone f_1 with amplitude A_1 which satisfies

$$\|f_0 - f_1\| < k_m(A_0 - A_1) \tag{3.1}$$

may be regarded as inaudible since it is 'masked' by f_0 [15]. In this case, k_m is a perceptual constant which determines how large the 'shadow' of a tone is. This situation is illustrated in figure 3.3.

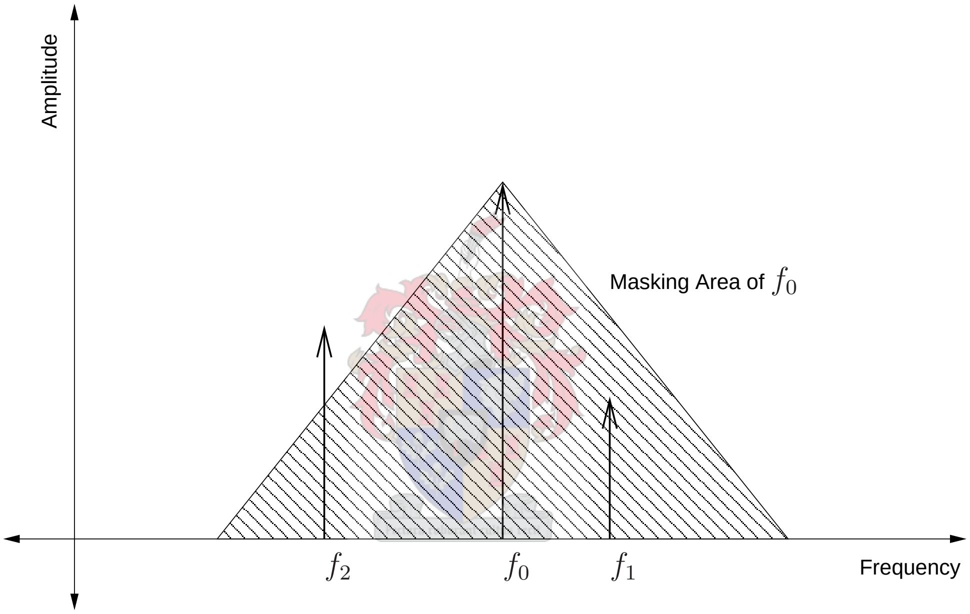


Figure 3.3: *The Masking Phenomenon*

Temporal masking is similar to frequency masking except that the tones are separated in time instead of simply in frequency. The effect of temporal masking is typically from 5ms before the onset of the masking tone until 200ms after the masking tone ends.

3.3.2 Non-Linearity

Loudness Perception

Sound is the perception of variations in atmospheric pressure. The pressure difference between sounds which are barely perceptible and those which are painful, is on the order of 10^6. Because of this huge variation in the range of values which sounds may assume, sound intensity is usually measured in terms of sound pressure levels on a logarithmic scale, known as the *decibel* scale:

$$L = 20\log_{10}\frac{p}{p_0} \tag{3.2}$$

Where p is the sound pressure and p_0 is the sound pressure for a sound which is at the threshold of audibility[1].

Whereas sound intensity is a physical quantity, loudness is a perceptual quantity not only dependant on the intensity of a sound but also on the frequency of the sound. In order to address this shortcoming, units known as the *phon* and the *sone* are used to characterise loudness of tones.

The *phon* unit is defined as follows. A sound which has a loudness of n phons is perceived to be as loud as a 1kHz tone with an intensity of n dB. However, a doubling of loudness of a sound on the phon scale does not translate to a doubling in the perceived loudness of the sound. Because of the logarithmic nature of the phon scale, the perceived loudness of a sound doubles every 10 phons. The sone was introduced to take this into account. The conversion from loudness in phons (L_p) to loudness in sones (L_s) is as follows:

$$L_s = 2^{\frac{L_p - 40}{10}} \tag{3.3}$$

The result of this relation is that a doubling in the sone value of a sound is equivalent to a doubling of the perceived loudness of the sound.

Pitch Discrimination

The smallest change in pitch which humans can recognise is not a constant quantity, but is dependent on the frequency of the original pitch. In the frequency band which is of interest to us (between 500 and 4000Hz), changes in frequency of around 0.3% are noticeable [52].

The explanation for the pitch discrimination abilities of the ear is usually described as being related to the critical bands. The auditory system works by decomposing sounds into component frequencies. Thus the ear acts as if it is composed of a number of band-pass filters. The bandwidth and centre frequencies of these filters are known as the *critical bands*. The critical bands affect the resolution with which different pitch frequencies may be discriminated.

It is generally accepted [39] that the critical bands are not regularly distributed in frequency. Therefore it is desirable to define a frequency scale along which the critical

[1] $p_0 = 20\mu\text{Pa}$ [52]

bands are regularly distributed. One commonly used frequency scale is the so-called *mel* scale. The frequency in Hz is transformed to frequency in mel using equation 3.4

$$m(f) = 1125 \log_e(1 + \frac{f}{700}) \tag{3.4}$$

3.4 Characteristics of the Speech Waveform

3.4.1 Quasi-Stationarity

A random process (such as the samples of a signal) is referred to as stationary if none of its statistics are affected by a shift in the time domain [77]. Speech signals are not stationary but the concept of *quasi-stationarity* is often used when referring to speech signals [15, 20, 39].

The term *quasi-stationary* is usually applied to a signal in which intervals may be found such that the statistics of the signal (or the short term *features* of the signal) do not change significantly over the intervals.

The speech signal is usually regarded as being quasi-stationary over an interval of 50ms [20, 39]. This interval corresponds approximately to the average phoneme rate. Indeed if one examines the parameters relevant to parametric voice coding, one finds that these parameters are usually very nearly constant over typical phonemes (except in the case of diphthongs where they tend to be interpolated linearly over the duration of the phoneme [84]). Vocoders typically utilise this quasi-stationarity implicitly through their block-processing of the speech signal. Typical processing segment lengths for vocoders vary between 5ms and 30ms as shown in the table below.

Vocoder	**Analysis frame length**
MS-3005 MELP	22.5
FS-1016 CELP	30
FS-1015 LPC10e	22.5
GSM AMR	50

Table 3.1: *Analysis frame length of some common vocoders (from [33]).*

It is impractical to use segments of substantially more than 25ms since this would mean that a large number of segments would substantially overlap phoneme boundaries. This in turn would usually mean that the parameters calculated for the analysis segment would be a mixture of the parameters for the various phonemes in the analysis segment, weighted by the duration for which the phoneme appears in the analysis segment. Clearly this is undesirable since we want to transmit the parameters of the individual phonemes as distinctly as possible and we wish to avoid possible cross-talk between phonemes which will diminish the accuracy of the parameter estimation process.

The optimal solution to this problem would be to align the analysis segments with the phoneme boundaries. This would present quite a challenge since not only are phoneme boundaries typically quite difficult to characterise algorithmically, but phonemes also vary substantially in length (between approximately 40 and 400 ms for vowels [20]).

3.4.2 Energy Bias

The long term spectrum of the speech waveform is not flat but exhibits noticeably more energy in the lower frequency part of the spectrum. This bias is roughly 10 dB/octave in frequencies above 500Hz [70]. The primary reason for this bias is given in [15] as the radiation effect of sound from the lips.

While the predominance of low frequencies may not be *perceptually* noticeable, it tends to affect the analysis of the speech waveform. The most noticeable occurrence of this is in Linear Predictive (LP) Analysis. As we will later demonstrate, LP analysis consists of a residual energy minimisation. Since the speech energy is not equally distributed throughout the spectrum, the linear predictor which we obtain tends to sacrifice accuracy in the high frequency regions in favour of the low frequency regions.

The most common solution to this problem is to filter the raw speech signal before processing with a digital filter designed to remove the energy bias. This is the approach used in CELP, LPC and MELP, as described in Chapter 4.

3.5 Linear Prediction and the All-Pole Model of Speech Production

Linear Predictive analysis was first proposed by Atal and Schroeder in 1968 [4]. Since then it has become exceedingly popular for a wide variety of speech processing applications. Deller, Proakis and Hansen [20] go so far as to say that:

> The technique has been the basis for so many practical and theoretical results that it is difficult to conceive of modern speech technology without it.

There are a number of reasons for this. First and foremost is that linear prediction using an all-pole filter very closely models the physical model of speech production as shown in 3.1. A full treatment of the open-tube model of speech production can be found in [20], demonstrating its equivalence to the all-pole model for speech production.

3.5.1 Derivation of the LP System

A complete treatment of the derivation of linear predictive analysis may be found in [20] and [15].

Kay [46] defines an auto regressive moving average (ARMA) process as one described by the difference equation

$$s[n] = -\sum_{k=1}^{p} a[k]s[n-k] + \sum_{k=0}^{q} b[k]s_0[n-k] \tag{3.5}$$

Given a stationary output signal, $s(n)$, produced by an auto regressive moving average (ARMA) process with transfer $H(z)$ function, driven by an input sequence $s_0(n)$, we denote the spectrum of $s(n)$ by $\Theta(z)$ and that of $s_0(n)$ by $\Theta_0(z)$. Further, we can write the output as a function of the parameters of the process and the input in the time domain as follows:

$$\begin{aligned} \Theta(z) &= \Theta_0(z)H(z) && (3.6) \\ &= \Theta_0(z)\frac{1+\sum_{i=1}^{L} b(i)z^{-i}}{1-\sum_{i=1}^{R} a(i)z^{-i}} && (3.7) \end{aligned}$$

In terms of the source-filter model discussed in 3.2, $s(n)$ is the speech waveform, $s_0(n)$ is the excitation waveform produced by the source and $H(z)$ represents the transfer function of the vocal tract during the formation of the current phoneme. Here the notion of *quasi-stationarity* is once again relevant; we regard the ARMA process that is the vocal tract as having constant parameters over the entire segment of speech under consideration.

Following Chu [15], we ignore the zeros of the transfer function for the following reasons:

1. We can represent the magnitude spectrum of the speech sufficiently well with an all-pole system. Thus we lose only the phase of the speech signal through this generalisation. The human ear is effectively ‘phase-deaf’, thus the phase of the output signal may be regarded as redundant information.

2. The poles of an all pole system can be determined from the output, $s[n]$, by simple linear equations. In the case of LP analysis, this output is all the information we have available, since we have no explicit information about the excitation energy produced by the glottis.

Then as shown in [15] and [20] we can re-write the system as an all-pass system in series with a minimum phase system, in series with a real-valued gain.

$$\Theta(z) = \Theta_0(z)\Theta_{min}(z)\Theta_{ap}(z)$$

thus

$$S(z) = \Theta_0(z)\Theta_{min}(z)\Theta_{ap}(z)E(z)$$

or in the time domain

$$s(n) = \sum_{i=1}^{I} a(i)s(n-i) + \Theta_0 e(n) = \mathbf{a}^T\mathbf{s} + \Theta_0 e(n)$$

where

$$\mathbf{a} = [a(1), a(2), a(3) \ldots a(I)]^T$$

and

$$\mathbf{s}(\mathbf{n}) = [s(n-1), s(n-2), s(n-3), \ldots s(n-I)]^T$$

3.5.2 Representations of the Linear Predictor

There are various representations which are commonly used to represent the linear predictor.

Linear Predictor Coefficients

These are the most obvious of the representations and are simply the elements of the vector **a** as shown above. They are exactly the coefficients (taps) of the direct form 1 realisation of the predictor.

Working directly with the LP coefficients has a number of advantages.

1. The transfer function of the Linear Predictor may be easily manipulated using the LP coefficients. As will be shown in 5.1.6, it may be advantageous to manipulate the 'optimal' linear predictor to produce perceptually better results.

2. The simplest algorithm for linear prediction synthesis is the direct form 1 filter realisation, [62]. This realisation of the linear predictor requires that we use the LP coefficients.

Reflection Coefficients

The reflection coefficients are very strongly suggested by the lossless open-tube acoustic model of speech production (see 3.1). A thorough treatment of the derivation of the reflection coefficients from the physical parameters of the vocal tract can be found in [20]. The reflection coefficients have an obvious advantage over the predictor coefficients in that they are bounded between -1 and 1. This makes them substantially easier to quantise and to deal with on fixed-point architectures. The reflection coefficients can be used to directly compute the output of the LP system by means of a lattice filter realisation. For any set of LPCs of an LP system of order P, we can compute the equivalent set of RCs by means of the following recursion [39].

$$\begin{array}{rcll} k_i & = & a_i^i & , \quad i = P \ldots 1 \\ a_j^{i-1} & = & \frac{a_j^i + a_i^i a_{i-j}^i}{1-k_i^2} & , \quad j = 1 \ldots i \end{array}$$

Similarly, we can use the following recursion to convert the set of reflection coefficients to an equivalent set of predictor coefficients.

$$\begin{array}{rcll} a_i^i & = & k_i & , \quad i = 1 \ldots P \\ a_j^i & = & a_j^{i-1} - k_i a_{j-1}^{i-1} & , \quad j = 1 \ldots i \end{array} \tag{3.8}$$

Log-Area Ratios

The i'th Log-Area Ratio (LAR) is defined as

$$\mathrm{LAR}_i = \ln\left(\frac{1-k_i}{1+k_i}\right)$$

The physical equivalent of the Log-Area Ratio (and also the reason for the name) is the natural logarithm of the ratio of the cross-sectional areas of adjacent sections of the lossless tube model of the vocal tract. The log area ratios are not theoretically bounded in any interval but they are usually distributed very near 0 and usually have a magnitude of less than 2 [15].

Line Spectrum Frequencies

The line spectrum frequencies represent another way of stable representing the LP system so that small changes in the parameters produce small changes in the perceptual character of the system.

The line spectrum frequencies LSF were first proposed by Itakura [40] as a representation of the linear predictor.

There are a number of advantages to using the line spectrum frequencies.

1. Line spectrum frequencies are bounded between 0 and π. This makes them highly suitable for situations where numerical precision is limited, such as environments using fixed point arithmetic.

2. The positions of the LSFs is closely related to the positions of the formants. This makes them ideal for the simple calculation of perceptually motivated distance measures.

3. LSFs may be interpolated to produce interpolated values of the LP spectrum. Additionally, linear interpolation of the LSFs will always result in a stable predictor.

4. Perturbations in a single LSF cause very local perturbations in the LP spectrum (near the frequency of the perturbed LSF).

5. As long as the frequencies are ordered, the LSF representation will result in a stable predictor. This means that by simply re-ordering the frequencies of an unstable predictor we can create a stable one. With the reflection coefficients one may also easily verify that the predictor is stable, however with the LP coefficients one needs to evaluate the transfer function in order to determine stability.

In Appendix B we present a derivation of the line spectrum frequencies.

3.5.3 Optimisation of the Linear Prediction System

In LP analysis of time signals we wish to determine **a** according to some optimality criterion, i.e. by minimising a parameter which is some function of the signal and the predictor coefficients.

One common parameter for optimisation is the energy of the residual signal. We seek **a** so that $s_0(n)$ has minimum variance. We demonstrate in appendix A that there indeed exists a unique solution.

There seems little motivation in the physical system for minimising the energy in the glottal excitation. The primary reason for choosing this parameter to minimise would be that the mathematics then leads to a solution to the system which is easily computed and numerically tractable.

As shown in appendix A, the optimal LP system may be obtained by solving the matrix equation:

$$\mathbf{r} = \Phi \mathbf{a}$$

Where Φ is the covariance matrix of the speech segment and **r** is the biased autocorrelation estimate.

3.5.4 The Levinson-Durbin Algorithm

Using this form Φ assumes a very special structure known as *Toeplitz*. This allows us to use the extremely efficient Levinson-Durbin algorithm to solve the normal equations.

Normally, using Gauss-Jordan reduction, the solution to the equation would require order P^2 operations to solve for the P’th order predictor coefficients [80].

In 1947 Norman Levinson proposed an algorithm for solving the problem $\mathbf{Ax} = \mathbf{b}$ for the special case where **A** is Toeplitz, symmetric and positive definite. In 1960, Durbin found a slightly more efficient algorithm for the special case where **b** has a special relationship to the elements of **A**.

The Levinson-Durbin algorithm uses only order P operations to solve this system. The L-D algorithm iteratively uses the optimal $(P-1)$'th order predictor to determine the optimal P'th order predictor.

In Appendix C we present a derivation of the Levinson-Durbin recursion.

3.5.5 The Le Roux-Gueguen Algorithm

The reflection coefficients (RCs) of a stable linear predictor are bounded by the interval $(-1,1)$ and we can force the reflection coefficients of an unstable predictor to fall within this interval, thus stabilising the predictor. Unfortunately no theoretical bound can be placed on the linear predictor coefficients LPCs. This presents a problem on many fixed point architectures.

To circumvent this problem Le Roux and Gueguen [48] proposed a method to compute the reflection coefficients directly from the autocorrelation values without dealing with the LPCs. Hence, problems related to dynamic range in a fixed point environment are eliminated.

Of course since the algorithm only yields the reflection coefficients, a direct form realisation of the synthesis filter is not possible. However, a lattice realisation of the synthesis filter can re-synthesise the signal directly and stably from the reflection coefficients [20].

A complete treatment and derivation of the Leroux-Gueguen Algorithm can be found in [15].

We present here only a description of the algorithm.

1. Initialisation:

   $$\forall \;\; k \in \{-(P-1), -(P-2), \ldots, (P-1), P\},$$

   Set

   $$\epsilon_{0,k} = r_k$$

   Where r_k is the kth autocorrelation value.

2. For $l = 1, 2, \ldots, P$, Let

   $$\begin{aligned} k_l &= \frac{\epsilon_{l-1,l}}{\epsilon_{l-1,0}} && (3.9) \\ \epsilon_{l,k} &= \epsilon_{l-1,k} - k_l \epsilon_{l-1,l-k} && (3.10) \end{aligned}$$

   Then k_l is the lth reflection coefficient.

3.6 Pitch Tracking and Voicing Detection

The twin concepts of pitch and voicing are so ubiquitous in speech processing that they are often vaguely defined. This is also perhaps because there is no rigorous mathematical definition of either, and further because the two are so closely intertwined that it is virtually impossible to define one without referring to the other.

Pitch

Pitch is defined as the frequency at which the glottis closes during voiced sounds. Pitch is generally not defined for unvoiced sounds. The so called *pitch marks* correspond to glottal closures and are characterised by strong impulses in the excitation signal and also in the speech waveform. Generally, any given speaker will have preferred pitch around which the pitch of their speech will fluctuate. For male speakers, the pitch range is usually between 50-250Hz and for female speakers, the range is usually 120-500Hz [20].

Voicing

Voicing is a slightly more vague concept and is often not calculated explicitly, particular in higher-rate excitation coders such as CELP. However, we usually define voicing as the presence or absence of periodic glottal closures during a phoneme. If few or no glottal closures occur, the sound is referred to as unvoiced, whereas a number of regular glottal closures would characterise a sound as voiced.

3.6.1 Pitch Tracking

For most low-rate voice coding algorithms, accurate determination of the pitch and voicing of the speech waveform is crucial to the quality of the synthesised speech. By necessity, pitch is a short term feature of the speech waveform and short-term analysis techniques are used to determine the pitch of a frame.

Most pitch tracking algorithms may be distinctly divided into two steps, namely:

1. A pre-processing step
2. An instantaneous pitch determination step.
3. A post-processing step.

Pre-Processing

The pre-processing usually involves band-pass filtering to remove the high-frequency and DC components of the signal. Most of the pre-processing algorithms are designed to emphasise the high amplitude pulses which characterise the voiced speech. Common techniques include:

- Centre-clipping the speech. This involves applying the following non-linear operation to the speech signal:

  $$s^*(t) = \begin{cases} s(t), & |s(t)| > T \\ 0, & |s(t)| < T \end{cases} \tag{3.11}$$

  Clearly, the choice of the clipping threshold (T) is crucial. [20] suggest 30% of the maximum value of the input signal as the clipping threshold. Centre clipping has a whitening effect on the speech spectrum which may aid in pitch determination. Additionally, because the pitch marks are generally of high amplitude, centre clipping removes the effect of the vocal tract response while maintaining the excitation pulse train [15, 20, 33, 70].

- Raising the speech waveform samples to a large (odd - to preserve the sign) power.

- Filtering with the inverse of the optimal linear predictor to obtain the predictor residual. This idea was introduced by Markel [51] as part of the SIFT Algorithm for fundamental frequency estimation. This removes the effect of the vocal tract response from the speech waveform, thus resulting in a flat spectrum. As mentioned below, the formants resulting from the vocal tract response may often interfere with the pitch estimation by emphasising portions of the speech spectrum. Furthermore, this predictor residual should closely resemble the glottal excitation.

Instantaneous Pitch Estimation

Typically the instantaneous determination of pitch involves finding the candidate pitch from a set of candidate pitches which maximises the *candidate pitch score*. Usually if the best candidate pitch score is below a threshold the sound is classified as unvoiced.

The following short term features are commonly used to estimate the pitch and voicing of a speech segment.

Short Time Autocorrelation; Because of the regularity of the pitch pulses, the speech signal exhibits strong self-similarity at the pitch period, and thus the autocorrelation of the speech signal exhibits a strong peak at the pitch period. However, since the excitation is approximately an impulse train, the autocorrelation also exhibits peaks at multiples of the pitch period.

Amplitude Magnitude Difference Function; The AMDF or MDF exhibits very similar properties to the auto-correlation except that it is minimised when the autocorrelation is maximised. The MDF of the speech segment $\mathbf{s}$ of length N is defined as:

$$\mathrm{MDF}[\tau] = \sum_{n=\tau}^{N-1} \|s[n] - s[n-\tau]\|$$

Short Time Cepstrum; The harmonics of the pitch period are combined by the short term cepstrum. This typically results in a large peak at the pitch period in the high time portion of the cepstrum.

Harmonic Product Spectrum(HPS)/Harmonic Sum Spectrum(HSS); The HSS are calculated by summing compressed (in frequency) versions of the spectrum (FFT magnitude). The idea is that at the fundamental frequency, the various harmonics of the fundamental will sum together constructively. The net effect is that the energy of the first few harmonics of each frequency will be taken into account when calculating the fundamental. Of course this is reliant on a high sampling rate, since one can of course only sum together at most N harmonics where

$$N = \left\lfloor \frac{f_s}{2P_{max}} \right\rfloor \tag{3.12}$$

with f_s the sampling frequency at which the power spectrum is evaluated and P_{max} the maximum pitch candidate frequency. Of course the HPS is calculated by simply summing the *log* spectrum. A very complete description of the HSS/HPS may be found in [84].

Post-processing

The post-processing step usually considers constraints such as

1. Continuity of the pitch track.
2. Realistic pitch values.

Often in pitch processing, the use of non-linear filters such as median filtering may be useful. This is described in detail in [70]. We also investigated the use of the non-linear $LULU$ filters as described in appendix F.

3.6.2 Pitch Estimation Errors

Pitch and voicing are notoriously difficult short-term features to estimate, at least in part because they are mathematically ill-defined. While pitch and voicing errors do not usually substantially affect the intelligibility of synthesised speech, they often result in extremely irritating artifacts.

The typical problems experienced by pitch tracking algorithms are as follows:

Pitch Harmonic Errors; Because the glottal excitation is often approximately an impulse train, it has a spectrum which is close to an impulse train. Often, the higher harmonics of the signal may be stronger than the fundamental, due to the location of formants or simply due to noise in the estimation. This results in the estimated pitch being an integer multiple or fraction of the actual pitch. These phenomena are referred to as *pitch doubling* or *pitch halving* respectively.

Noise; The presence of background noise in the speech signal may result in pitch estimation errors, particularly if the noise has strong periodic components.

Vocal Fry; While the pitch track is generally smooth, in some speakers it may occasionally suddenly change substantially, particularly at the end of a voiced phoneme. The smoothness constraints imposed by most pitch tracking algorithms may cause problems in these cases [20].

Formant Interference; It is often difficult to separate the first formant (which lies between 150-800Hz [20] - overlapping with the region of allowable pitch frequencies.) from the pitch frequency. As mentioned above, this problem may be alleviated by performing the pitch estimation algorithm on the linear predictor residual instead of on the raw speech waveform.

In order to counteract this dilemma, some speech coders attempt to make the voicing decision less binary, by allowing for a smoother transition between voiced and unvoiced frames. MELP [55] does this by mixing voiced and unvoiced excitation and introducing aperiodic pulses in weakly voiced segments (see 4.3.1). CELP [5] does this using a closed loop analysis with various excitation vectors.

3.7 Speech Quality Assessment

The eventual aim of every voice coding system is to reproduce input sounds with the maximum fidelity allowed by the system (computational complexity, memory constraints and of course the allowed bit rate). In order to assess the relative success with which these objectives have been achieved, one of course requires a metric which is an accurate reflection of the 'distance' between two speech segments. Numerous solutions have been proposed for this problem, none of which has truly been found to be completely satisfactory.

3.7.1 Categorising Speech Quality

Generally in digital communications, voice quality is divided into four categories. These categories are fairly broad but are as follows:

Broadcast Quality; This is approximately the sort of quality one would expect from CD recordings of speech, at data rates upward of 256kbps.

Network or Toll Quality; This is approximately the quality one would expect over a standard 3kHz B/W telephone line, or which one can expect to achieve with data rates of around 64kbps.

Communications Quality; This refers to slightly degraded speech which is nevertheless of sufficiently high quality and suitable for general telecommunications.

Synthetic Quality; This is intelligible speech which has may sound unnatural and impair speaker recognition.

3.7.2 Subjective Metrics

Of course, since speech is inherently a human phenomenon, the only truly accurate judge of speech, and indeed the ultimate authority on the matter, is the human ear.

Diagnostic Rhyme Test

A subjective measure of voice quality is provided by the *Diagnostic Rhyme Test* (DRT) originally credited to Fairbanks [27]. In the DRT, listeners are asked to distinguish between phonetically similar 'rhyming' words, such as 'heat' and 'meat'. Later, an enhanced version of the DRT was presented by House [38]. This required that the listener listens to an utterance and decide which of six candidate (printed) words the utterance represented. This enhanced test is known as the *Modified Rhyme Test* (MRT). According to Goldberg [33] it is seldom used. While these tests provides very accurate assessments of the *intelligibility* of the voice coder, they nevertheless do not take into account some of the features which contribute to making a voice coding system acceptable to the user, such as the naturalness of the synthesised voice.

Diagnostic Acceptability Measure

It is likely with this last idea in mind that Voiers in 1977 [85] proposed the *Diagnostic Acceptability Measure* (DAM) as a measure of the quality of synthesised speech. The DAM requires that listeners evaluate speech on 16 different scales, divided into 3 categories: signal quality, background quality and total quality. Each of these is divided up into descriptive sub-categories such as: *Fluttering, Muffled, Tinny, Rumbling, Buzzing, Hissing, Intelligible, Pleasant*. Further details may be found in [20].

Assessment (Segment Quality)	**Score**
Excellent	5
Good	4
Fair	3
Poor	2
Bad	1

Table 3.2: *ACR Scores*

MOS	**Corresponding Quality**
4-4.5	Toll Quality
3.5-4	Communications Quality
2.5-3.5	Synthetic Quality

Table 3.3: *MOS Scores for Various Speech Qualities*

Mean Opinion Score

The importance of certain features in determining the quality of speech may differ substantially between individual listeners. Thus it may be desirable to have a much broader measure of quality for speech. The mean opinion score or MOS is obtained as follows: A subject is presented with a synthesised speech segment. The subject then assigns a subjective *score* to the synthesised speech based on the perceived quality. Listeners assign a score (or absolute category rating) to each synthesised segment according to table 3.2 :

The average of the scores over all listeners over all segments for a particular voice coder is known as the *Mean Opinion Score* (MOS). Mean opinion scores may vary substantially from test to test and are thus not a good absolute reference of voice coding quality. Nevertheless, they are generally used as the preferred method to evaluate the quality of a voice coding system. Often, the MOS is used to demonstrate the superiority of one coder over another [26] or simply to demonstrate the functionality of a coder [55].

The generally accepted relation of MOS to the quality categories mentioned above is presented in table 3.3, the data for which is excerpted from [79].

3.7.3 Objective Metrics

While the DRT and MOS provides one with an ultimate assessment of the subjective quality of a voice coding system, it is unfortunately a very impractical system to use. The need to train listeners is described by Spanias [79]. Additionally, accurate results require that one use a large number of listeners (at least 12 [79]) and a large evaluation corpus. These factors combine to create a very cumbersome test procedure, one which is highly undesirable when developing a voice coding system, when a large number of design decisions must be made. In the actual processing performed by a voice coder, comparison of speech segments may also be necessary, for example when performing the closed-loop analysis-by-synthesis encoding such as in CELP (described in section 4.2.1). Thus it is desirable to be able to quickly compare two speech samples, preferably using an algorithm which can be implemented on a computer. The following measures are commonly used to compare speech segments.

Signal to Noise Ratio (SNR)

The signal to noise ratio is familiar to anyone who has had any dealing with communications systems. In the case of speech signal we usually define the noise as the difference between the original and synthesised voice. Thus the SNR is expressed as:

$$\mathrm{SNR} = 10 \log_{10} \frac{\sum_{n=0}^{M} s(n)^2}{\sum_{n=0}^{M} (s(n) - \hat{s}(n))^2} \tag{3.13}$$

Segmental SNR

The segmental SNR is a variant on the SNR which is defined as follows.

$$\mathrm{SEGSNR} = \frac{10}{N} \sum_{i=1}^{N} \log_{10} \frac{\sum_{n=M*i}^{M*(i+1)} s(n)^2}{\sum_{n=M*i}^{M*(i+1)} (s(n) - \hat{s}(n))^2} \tag{3.14}$$

In other words it is the average of the short term SNR over many finite length speech segments. The segmental SNR tends to penalise more strongly coders which have varying quality.

Unfortunately neither of the above metrics have much bearing on the perception of the speech and additionally they are very sensitive to features to which the ear is essentially deaf. Consider the following examples:

1. Change the polarity of each sample, i.e. $\hat{s}(n) = -s(n)$. This is a change which is imperceptible to the human ear, but it results extremely poor SNR.

2. Delay the speech by a single sample. At a sampling rate of 8kHz, this will imply a delay of 0.125ms, which would be completely inaudible to the human ear. However, the substantial de-correlation between successive samples (high entropy) of the speech signal will mean that the expected value of $(s(n-1) - s(n))^2$ will be large and consequently the SNR will be poor.

Consequently, both the SNR and SEGSNR will reflect poorly in these cases, despite the fact that the above signals are perceptually identical to the original (reference) signal.

While these may seem to be somewhat pathological examples we must consider the fact that parametric coders tend to transmit information like phase extremely poorly. Additionally, because of the frame-based approach to analysis taken by many parametric coders we must assume that the correlation between the individual samples of the original and synthesised speech will be quite poor.

We therefore wish to derive a more appropriate metric, in other words, one which positively correlates with the perceived distance of the speech samples. Additional constraints are of course that the calculation of the metric should be computationally tractable.

A few metrics which have become popular, and which take more cognisance of perceptual considerations are:

Itakura-Saito Spectral Distortion Pseudo-Metric

As we have mentioned before, the human auditory system is relatively insensitive to phase distortion. Therefore we may base our metric purely on the magnitude spectrum of the waveform. The Itakura-Saito pseudo-metric is formulated according to this ideas:

1. The two segments $a(n)$ and $b(n)$ may be compared by comparing their associated optimal linear predictors (for some relevant predictor order). This is because the linear predictor spectrum of a signal models the magnitude envelope of the signal very well.

2. We may compare the linear predictor spectra by directly comparing the two vectors of optimal LP coefficients for the respective segments ($\hat{a}$ and $\hat{b}$).

3. This comparison proceeds as follows:

   Let $\alpha(m) = [1 - \hat{a}^T]^T$ and $\beta(m) = [1 - \hat{b}^T]^T$

   Let $\tilde{R}_a$ be the auto-correlation matrix for $a(n)$

   Define

$$d(a,b) = \log\left(\frac{\beta^T \tilde{R}_a \beta}{\alpha^T \tilde{R}_a \alpha}\right) \tag{3.15}$$

Thus according to this metric the distance between the two segments is defined as the relative accuracy with which the optimal linear predictor of one predicts the other. Also, from the definition, since $\tilde{R}_a \neq \tilde{R}_b$ therefore $d(a,b) \neq d(b,a)$. Therefore we cannot call this distance a true metric but refer to it as a *pseudo-metric*. A complete description of this metric is found in the seminal paper by Itakura, Saito et. al. [41].

Frequency-Weighted Segmental SNR

The FW-SEGSNR has been formulated in several ways [6, 83], most of which follow approximately the following pattern, as described in [20]:

$$\frac{10}{M}\sum_{j=0}^{M-1}\log_{10}\left[\frac{\sum_{k=1}^{K} w_{j,k}\log_{10}\left[E_{s,k}(m_j)/E_{\epsilon,k}(m_j)\right]}{\sum_{k=1}^{K} w_{j,k}}\right] \tag{3.16}$$

.

We have omitted a factor 10 found within the summation, because it contributes only a constant term to the FW-SEGSNR and is thus not of interest. The energies of the signal ($E_{s,k}$) and noise ($E_{\epsilon,k}$) in each band (k) are weighted according to perceptual considerations and the choice of bands is usually also perceptually motivated.

Weighted Spectral Slope Measure

The weighted spectral slope measure (WSSM) is also known as the *Klatt Measure* [47]. Every 12ms, a bank of 36 filters is used to calculate a smoothed short term spectrum. The filters have bandwidth corresponding to the ear's critical bands. This means that the filters implicitly impart an equal perceptual weight to each critical band. The method uses the above short term spectrum to estimate weighted differences between the spectral slopes in each band. This gives us a metric which is relatively insensitive to differences in formant peak height but very sensitive to differences in formant location. The WSSM was rated very highly in [20] but [36] reports less positive results with this measure and describes the Itakura-Saito measure as being more effective.

PESQ

In February 2001, the International Telecommunications Union published an algorithm known as *Perceptual Evaluation of Speech Quality* or PESQ [43]. The aim of this algorithm was to estimate the perceptual quality of narrow band voice codecs. PESQ was intended to address factors such as packet loss, variable delay, coding distortions and channel errors which are very poorly handled by conventional methods of comparison. PESQ is designed to compare a reference and degraded version of the same waveform. The basic idea of PESQ is to transform the respective waveforms into a perceptual representation similar to the parametric representation used by narrow band vocoders. The difference between these parametric representations is evaluated by a cognitive model in order to estimate the perceptual distance between the two speech samples. This distance is expressed as a quality evaluation of the degraded sample.

3.7.4 Purpose of Objective Metrics

Having defined a metric, we may easily use this metric as a measure of the performance of a voice coding system by simply measuring the distance between the original and synthesised speech. Since, as we mentioned above, the true measure of a vocoders performance is given by the DAM, we can assess our objective measure by comparing the correlation of the objective measure to the DAM.

Objective Quality Measure	$\lvert\hat{\rho}\rvert$
SNR	0.24
SEGSNR	0.77
F-W SEGSNR	0.93
Itakura-Saito	0.59
WSSM	0.74
PESQ	0.935

Table 3.4: *Objective Quality Measure Performance*

In table 3.4 $\hat{\rho}$ is the average correlation coefficient between the results of each objective quality measure discussed in this section, and the MOS, measured over a large number of conditions. The data in the table are extracted from [20], who do not explicity specify the experimental conditions used to obtain the data.

In the case of each of the three SNR measures, we must note that these scores are only calculated for waveform coders. None of the SNR measures is considered suitable for the measurement of the performance of parametric coders.

Chapter 4

Standard Voice Coding Techniques

The following sections are intended to illustrate the state of the art of voice coding techniques. The sections on the various coders are arranged roughly in the chronological order in which the coders were developed. This arrangement was chosen because to a large extent each design follows as a logical refinement to the preceding design.

4.1 FS1015 - LPC10e

LPC10e refers to an algorithm which may originally be attributed to Atal and Hanauer [2]. FS1015 and LPC10e have essentially become synonymous.

4.1.1 Pre-Emphasis of Speech

Speech is pre-emphasised with a first order IIR filter with the following transfer function:

$$H_{\text{pre-filter}}(z) = \left(1 - \frac{15}{16}z^{-1}\right)$$

The purpose of this filter is to improve the numeric stability of the LP analysis. As mentioned in 3.4.2, the speech waveform typically exhibits a high-frequency roll-off. Reducing this roll-off decreases the dynamic range of the power spectrum of the input speech, resulting in better modelling of the features in the high frequency regions of the speech spectrum [15].

The transfer function of this filter is illustrated in figure 4.1.

4.1.2 LP Analysis

The LPC10e standard (FS1015) specifies that a covariance method with synthesis filter stabilisation should be used to determine the LP spectrum of the speech. However, most modern implementations instead use an autocorrelation approach due to its improved

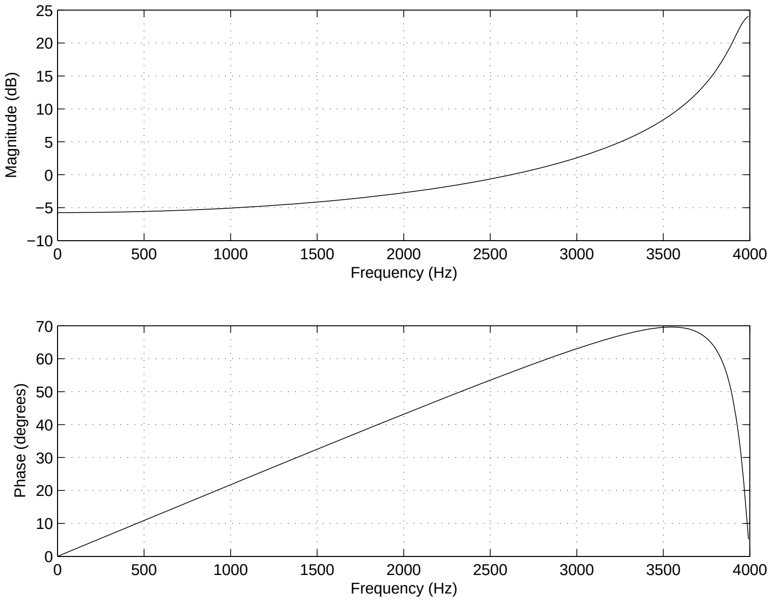


Figure 4.1: *Transfer function of pre-emphasis filter for LPC10e*

numerical stability and computational efficiency and since this does not affect the interoperability of the vocoder at all. FS1015 favours a *pitch synchronous* LP analysis. This means that the position of the LP analysis window is adjusted with respect to the phase of the pitch pulses. This design improves the smoothness of the synthesised speech, since the effect of the glottal excitation spectrum on the LP analysis of the speech is reduced substantially.

4.1.3 Pitch Estimate

LPC10e allows pitch ranged between 50 and 400Hz. The pitch estimate is obtained as follows.

1. Low-pass filter the speech signal.

2. Inverse filter the speech signal with a second-order approximation to the optimal 10th order predictor determined by the LP analysis.

3. Calculate the minimum value of the *Magnitude Difference Function* (MDF). The MDF is not as accurate as the auto-correlation in pitch determination but was chosen mainly for reasons related to computational efficiency [33]. On many architectures, the computational cost of a multiplication was an order of magnitude larger than that of an addition. On these architectures the MDF would be substantially more

efficient to calculate than the auto-correlation. [1]

4.1.4 Voicing Detection

A linear pattern classifier is used to perform the voicing detection task. The pattern vectors are composed of the following parameters:

1. Low-band energy.
2. Max-min ratio of the MDF.
3. Zero-crossing rate.
4. First reflection coefficient.
5. Second reflection coefficient.
6. Pitch prediction gains
7. Pre-emphasised energy ratio

The linear discriminant classifier defined by

$$\sum_{i-1}^{N} a_i p_i + c_j > 0, \quad \text{where} \quad j \in 0, 1, \ldots, M-1 \tag{4.1}$$

Finally, a smoothing algorithm is applied to the voicing decision. This smoothing algorithm is essentially a modified median smoother, which takes into account the voicing strength of each frame [10]. This prevents the occurrence of single voiced segments within unvoiced segments. The appearance of these single voiced frames may cause annoying artifacts in the synthesised speech, typically producing isolated tones.

4.1.5 Quantisation of LP Parameters

The first two reflection coefficients are converted to log-area ratios because statistically only the first two RCs have a significant probability of being close to 1. Chu [15] demonstrates some histograms of the size distributions of the various reflection coefficients.

[1] However, most modern DSPs have highly sophisticated architectures, which can typically perform a multiplication and addition in a single clock cycle and are often optimised for FFT calculation [82]. Furthermore there is a well-known relation that the IFFT of the Power Spectral Density is the auto-correlation [37]. If this theorem is used to calculate the auto-correlation, it is unlikely that the MDF will exhibit any substantial advantage in terms of computational complexity. For this reason most modern implementations of LPC10e use the auto-correlation estimate for pitch estimation.

In LPC10e a very simple scalar quantisation scheme is used, using a different codebook for each LP parameter. Each codebook is optimised for the particular LP parameter it is intended to encode. Two different quantisation schemes are used, depending on the outcome of the voicing decision. These are detailed in table 4.1.

Parameter	Encoded as	Number of Bits Allocated	
		Voiced Segment	Unvoiced Segment
a_1	LAR	5	5
a_2	LAR	5	5
a_3	RC	5	5
a_4	RC	5	5
a_5	RC	4	0
a_6	RC	4	0
a_7	RC	4	0
a_8	RC	4	0
a_9	RC	3	0
a_{10}	RC	2	0
Total		41	20

Table 4.1: *Quantisation of linear predictor parameters in LPC10e*

The choice of parameters to quantise, namely the LARs for the first two parameters and the reflection coefficients thereafter is probably motivated by the results observed by Gray and Markel [35], namely that the LARs are superior for encoding of the first two parameters but thereafter present no substantial advantage over the reflection coefficients.

4.2 FS1016 - CELP

CELP was first proposed by Atal and Schroeder in their 1985 paper [5]. It uses the same source-filter model as LPC, except that in the case of CELP, the simple buzz-hiss excitation of LPC is replaced by a more sophisticated excitation model.

In CELP, the excitation used in each frame is selected by the encoder from a large predetermined codebook of possible excitation sequences. Hence the acronym of ***C**odebook **E**xcitation with **L**inear **P**rediction*. The typical way in which the excitation codebook entry is chosen is by means of *analysis by synthesis*.

4.2.1 Analysis by Synthesis

In traditional open-loop analysis methods, an analysis of the speech signal is performed and the excitation sequence is chosen based on the result of this analysis. In the CELP encoder, a more sophisticated closed-loop approach is taken. In this approach, every possible excitation sequence is passed through the synthesis filter. The 'best' candidate from the analysis sequences is chosen by comparing the speech synthesised using each candidate sequence to the original speech segment.

4.2.2 Perceptual Weighting

The phenomenon of *masking* has already been discussed in 3.3.1. A simple way of controlling the CELP noise spectrum is by filtering the error signal through a weighting filter before minimisation. The weighting filter can be efficiently implemented by using the system function:

$$w(z) = \frac{a(z)}{a(\frac{z}{\gamma})} = \frac{1 + \sum_{i=1}^{P} a_i z^{-i}}{1 + \sum_{i=1}^{P} a_i \gamma^i z^{-i}}$$

With γ chosen as a constant between 0 and 1. This filter has a transfer function which is shaped roughly like the reciprocal of the transfer function of the LP synthesis filter. In other words it provides emphasis to the noise spectrum in non-formant regions while de-emphasising the noise in formant regions. As $\gamma \to 1, w(z) \to 1$ and thus no modification of the error spectrum is performed. As $\gamma \to 0, w(z) \to \frac{1}{a(z)}$, the formant analysis filter. Thus the constant γ introduces a bandwidth expansion of the error weighting filter. Generally for any CELP system, γ is selected by subjective listening tests. γ is usually chosen between 0.8 and 0.9 .

4.2.3 Post-filtering

The CELP post-filter was introduces to improve the perceptual quality of the synthesised speech. The post-filter is introduced into the synthesiser after the LP filtering of the reconstructed excitation signal has occurred. The CELP post-filter has the following transfer function:

$$h_1(z) = \frac{1 + \sum_{i=1}^{P} a_i \beta^i z^{-i}}{1 + \sum_{i=1}^{P} a_i \alpha^i z^{-i}}$$

The justification for this unusual choice is given in [15]:

Consider a post-filter composed of the bandwidth expanded LP synthesis filter

$$h(z) = \frac{1}{1 + \sum_{i=1}^{P} a_i \alpha^i z^{-i}}$$

With α chosen as a constant between 0 and 1. This post-filter reduces the perceived noise level of the LP synthesis by emphasising the formant regions. However, as the perceived reduction of noise increases, the synthesised speech acquires a 'muffled' quality, since the post-filter generally has a low-pass spectral tilt. Thus in the choice of α, one must compromise between reducing noise and reducing clarity of the synthesised speech.

A more sophisticated post-filter uses the transfer function:

$$h_1(z) = \frac{1 + \sum_{i=1}^{P} a_i \beta^i z^{-i}}{1 + \sum_{i=1}^{P} a_i \alpha^i z^{-i}}$$

This filter has a magnitude response in dB given by:

$$|h_1(z)|_{\text{dB}} = 20 \log \left(\frac{1}{1 + \sum_{i=1}^{P} a_i \alpha^i z^{-i}} \right) - 20 \log \left(\frac{1}{1 + \sum_{i=1}^{P} a_i \beta^i z^{-i}} \right)$$

In other words it is the difference between the frequency responses of two bandwidth expanded LP synthesis filters. However, the addition of zeros to the transfer function does not completely remove spectral tilt and thus a first order IIR filter with transfer function $h_2(z) = (1 - \mu z^{-1})$ is often used in cascade with the above filter. This filter is similar to the pre-emphasis filter and provides a high-pass spectral tilt to compensate for the low-pass effect. A popular choice for μ is 0.5 [15].

4.2.4 Pitch Prediction Filter

The LP excitation signal in CELP is generated by exciting the *pitch prediction filter* with the chosen excitation code-vector. The pitch prediction filter has the following transfer function:

$$h_p(z) = \frac{G_p}{1 - az^{-\hat{P}}}$$

G_p is a scaling factor, $0 < a < 1$ and $\hat{P}$ is an estimate of the number of samples in the pitch period. [20]. The aim of the pitch filter is to provides the long-term correlation in the excitation signal which is perceived as the pitch of the synthesised voice.

4.2.5 FS-1016 Bit Allocation

FS-1016 CELP Uses a different non-uniform scalar quantisation for each LPC. The individual LPCs are quantised as LSFs. Exact details of the quantisation scheme are detailed in [15].

4.3 MELP

The MELP model was originally developed by Alan McCree as a Ph.D. project and was published by McCree and Thomas Barnwell in 1995 [55]. After some refinement, it was submitted as a candidate for the new U.S. federal standard at 2.4kbps. MELP officially

Parameter	Multiplicity	Resolution	Total Bits per Frame
LPC	10	3,4,4,4,4,3,3,3,3,3	34
Pitch Period (adaptive codebook index)	4	8,6,8,6	28
Adaptive Codebook Gain	4	5	20
Stochastic Codebook Index	4	9	36
Stochastic Codebook Gain	4	5	20
Synchronisation	1	1	1
FEC	4	1	4
Future Expansion	1	1	1
Total			144

Table 4.2: *Bit Allocation for Model Parameters in FS1016 CELP*

became a U.S. federal standard in 1997, replacing LPC10e as the standard vocoder to be used in secure and digital voice communication over low bandwidth channels. The draught 2.4kbps MELP standard can be found in [69].

4.3.1 The MELP Speech Production Model

Mixed Excitation

The major shortcoming of LPC10e is the hard decision switching between voiced and unvoiced segments. This simple excitation model fails to accurately represent certain phoneme types, most notably fricatives. This results in a distinctly buzzy quality of the synthesised voice.

The MELP speech production model attempts to soften this hard voicing decision by introducing intermediate levels between between phonemes which are purely voiced and those which are purely unvoiced. These intermediate levels are achieved by dividing the excitation into sub-bands, where each sub-band may be either voiced or unvoiced.

From [55] :

> The most important feature of [the MELP model] is the mixed pulse and noise excitation.

During synthesis, an LPC10e-type excitation is synthesised for each of the 5 voicing bands as is demonstrated in figures 4.2 and 4.3.

The sub-bands are realised by means of FIR filters. The excitation signal for each excitation band is filtered with a band-pass FIR filter to create a band-limited excitation waveform. The pass-bands of the filters are respectively 0-0.5kHz,0.5-1kHz,1-2kHz,2-3kHz and 3-4kHz as shown in figure 4.4.

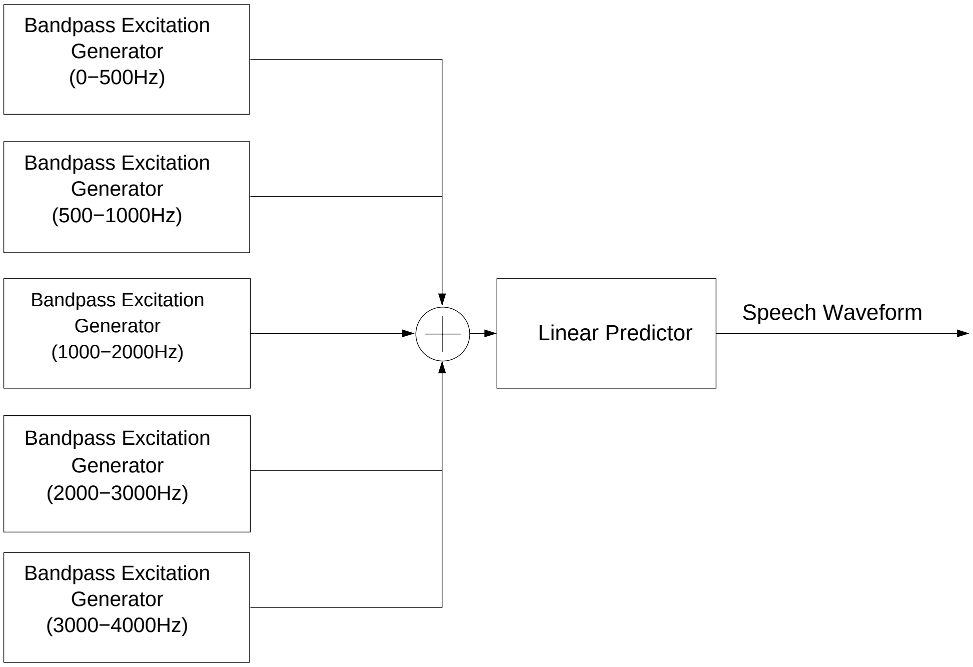

Bandpass Excitation Generator (0−500Hz)
Bandpass Excitation Generator (500−1000Hz)
Bandpass Excitation Generator (1000−2000Hz)
Bandpass Excitation Generator (2000−3000Hz)
Bandpass Excitation Generator (3000−4000Hz)
Linear Predictor
Speech Waveform

Figure 4.2: *The MELP speech production model.*

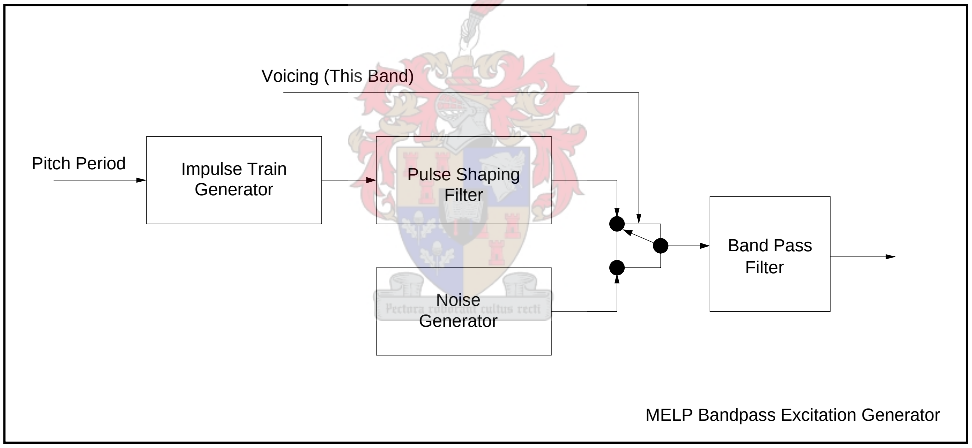


Figure 4.3: *Bandpass Excitation Generation in MELP Synthesis*

In the MELP analysis, the input waveform is filtered by a bank of FIR bandpass filters. These filters are identical to the filters used to band-limit the excitation signals. This produces 5 different band-limited approximations of the input speech signal. A voicing strength is determined in each of these band-limited signals. This voicing strength is regarded as the voicing strength for that frequency band.

These band limited excitation waveforms are added together to produce an excitation signal which is partly voiced and partly unvoiced. In this way, the MELP excitation signal is generated as a combination of bandpass filtered pulses and bandpass filtered

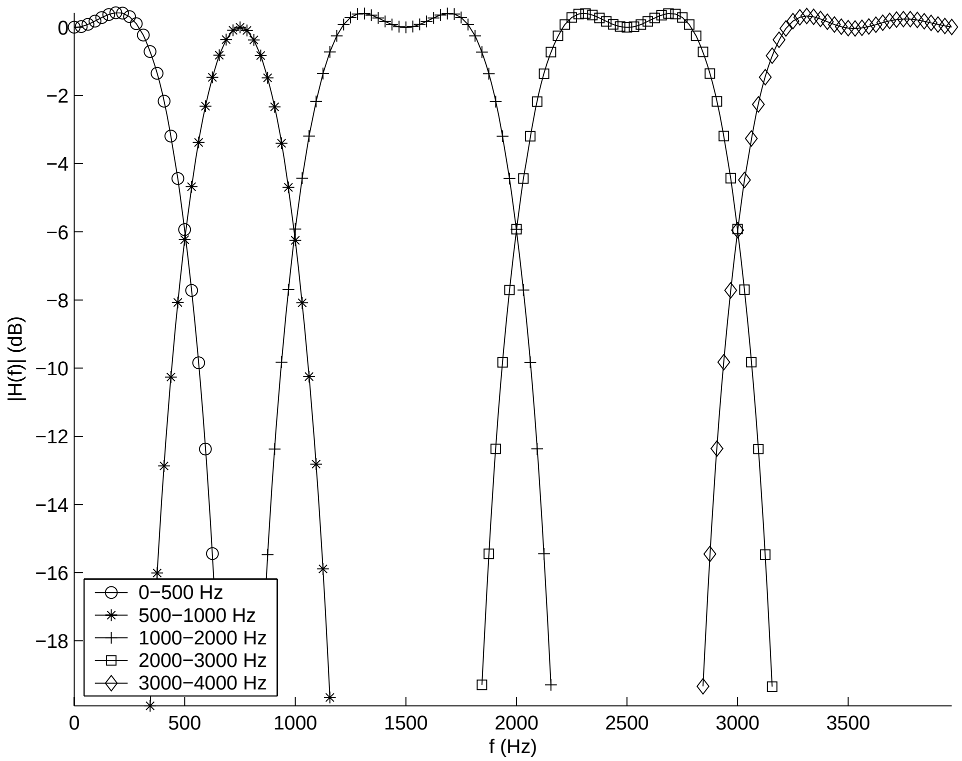


Figure 4.4: *Bandpass Excitation and Analysis Filters used in MELP Synthesis and Analysis. The stop-band part of the transfer function has been omitted for clarity.*

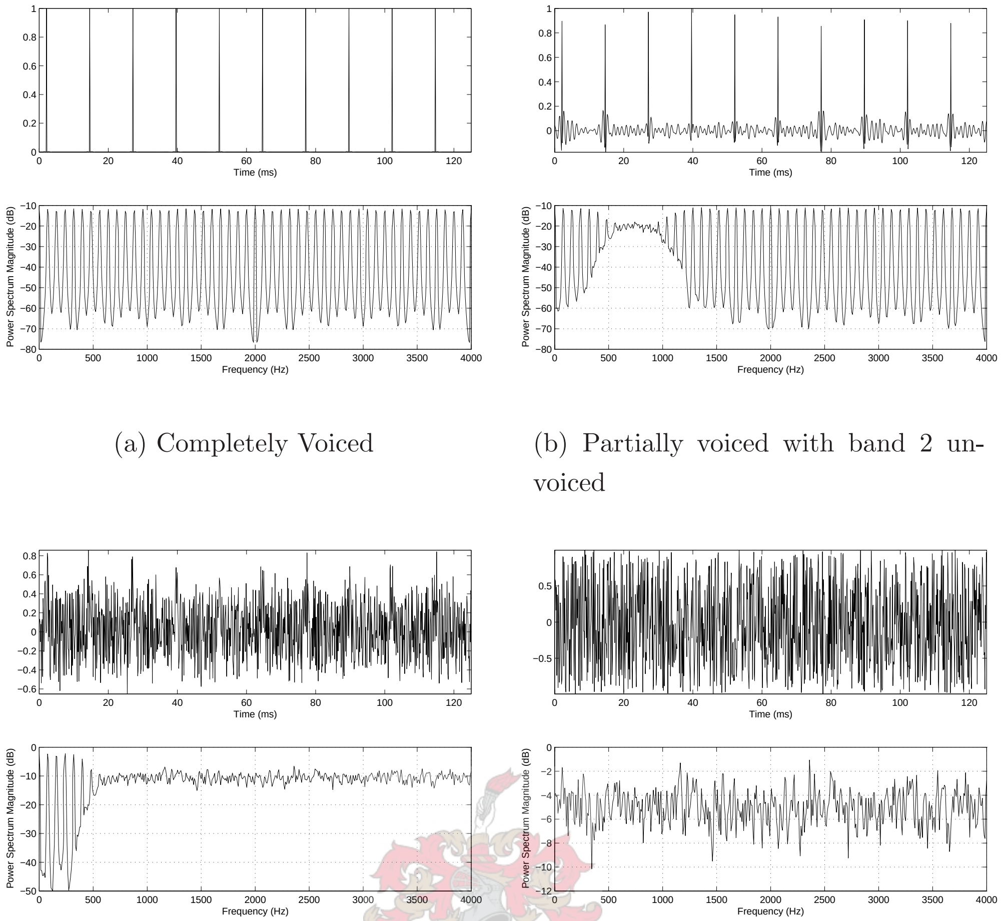
white noise. This substantially reduces the harshness of the voicing decision and removes a great deal of the hissiness and buzziness of LPC10e.

Figure 4.5 demonstrates the time signals and power spectra of some typical MELP excitation sequences. Time domain sequences are shown on the top and power spectra of each sequence is shown underneath. In the successive sequences we see how the excitation varies between a completely voiced (a) and a completely unvoiced (d) mode. In the intermediate some of the bands are voiced and some are unvoiced, as can clearly be seen in the power spectra.

Excitation Pulse Shaping

In order to more accurately model the shape of individual pulses of the voiced excitation vector, the MELP model calculates the strength of the various harmonics of the pitch period. This is done by evaluating the peak values of the predictor error signal PSD near the harmonics of the pitch period.

During synthesis, these magnitudes are used to generate an impulse using the inverse Fourier transform of the measured strengths of the harmonics of the pitch (appropriately padded to the pitch period). This results in a distorted pulse instead of a perfect impulse.



(a) Completely Voiced

(b) Partially voiced with band 2 unvoiced

(c) Partially unvoiced with band 1 voiced

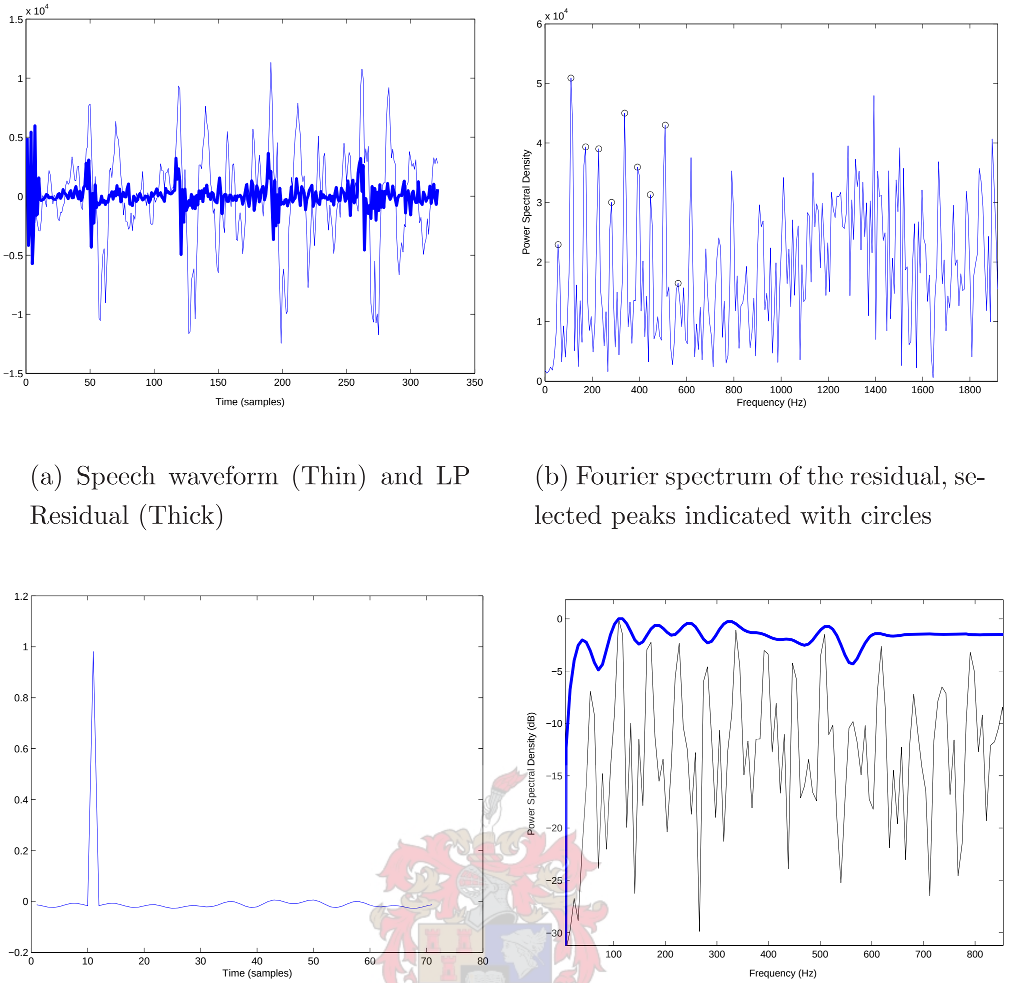
(d) Completely Unvoiced

Figure 4.5: *MELP Excitation Sequences and their Power Spectra*

This distorted impulse is repeated at the pitch period in order to create the excitation pulse train. This distorted impulse train models the impulses caused by glottal opening much more accurately than a simple impulse would, resulting in increased naturalness of the synthesised voice. This is illustrated in figure 4.6.

Aperiodic Pulses

Another problem commonly encountered in LPC10e synthesised speech is the presence of short isolated tones, particularly in speakers with higher pitched voices. These tones seem to be caused by very weakly voiced frames or frames which are misclassified as voiced [55]. The isolated tones can be alleviated by the addition of noise to the excitation signal for the weakly voiced frames but this noise needs to have such large variance in order to properly counter the effect that the noise becomes noticeable. In MELP, the problem is



(a) Speech waveform (Thin) and LP Residual (Thick)

(b) Fourier spectrum of the residual, selected peaks indicated with circles

(c) Reconstructed pulse shape

(d) Impulse spectrum (Thick) shown with Spectrum of LP Residual (Thin)

Figure 4.6: *Impulse Shape Reconstruction using the Fourier Magnitudes*

solved by destroying the periodicity of the voiced excitation by introducing *jitter*. This means that each pitch pulse is perturbed by up to 25% of the pitch period, simulating the erratic glottal movement which is often observed at voicing transitions. However, we cannot do this for strongly voiced frames. Introducing jitter in strongly voiced frames introduces a croaking quality to the synthesised voice which is undesirable. It is therefore crucial to accurately determine whether or not jitter should be introduced to a frame.

McCree [55] mentions that both aperiodic pulses and mixed excitation are needed to remove the buzzy quality of the LPC synthesiser suggesting that the human ear is capable of detecting separately both periodicity and peakiness. From [55] :

> We believe the buzz associated with LPC vocoders comes from higher fre-

> quency waveform peakiness in unvoiced or partially voiced frames while tonal noises are introduced when periodicity is present in speech frames which should be unvoiced.

The peakiness of a zero-mean signal is defined in the publication as:

$$\text{peakiness} = \frac{\sqrt{\sum x_n^2}}{\sum \|x_n\|}. \tag{4.2}$$

This quantity is large when there are a large number of samples which differ from the mean by more than one standard deviation. In other words when there are a large number of outliers in the signal.

McCree's explanation for the introduction of jitter is that the buzz of LPC vocoders is typically caused by the fact that the LP synthesis filter cannot adequately disperse the energy of the pulse train used as excitation for the voiced frames. Adding jitter to the mixed excitation model forcibly reduces this energy.

Pulse Dispersion Filter

The last step in MELP synthesis is a very weak filter referred to as the pulse dispersion filter. McCree and Barnwell [55] justify this filter with the following argument: The filter increases the match between the envelopes of bandpass filtered synthetic speech and natural speech waveforms in regions which do not contain a formant resonance. At these frequencies, the synthetic speech can decay to very low power in periods between pitch pulses. Particularly at lower pitch values, frequencies near the higher formants may decay significantly between excitation pulses. This does not tend to occur in natural speech, possibly because of incomplete glottal closure or secondary excitation pulses resulting from glottal opening.

4.3.2 An Improved MELP at 1.7kbps

In 1998 McCree and DeMartin [56] published an improved MELP vocoder which claimed to produce better speech quality at 1.7kbps. The salient features of this new vocoder are:

Improved Pitch Estimation

A sub-frame based pitch estimation algorithm is used which significantly improves performance in comparison to the pitch tracking used in the Federal Standard. This algorithm minimises the pitch-prediction residual energy over the frame, assuming that the optimal pitch prediction coefficient will be used over every sub-frame lag. This algorithm is substantially more accurate over regions of erratic pitch and speech transitions.

Noise Suppression

An averaged PSD is used to calculate an estimate of the noise power spectrum. The estimate of the noise PSD is used to design a noise suppression filter.

Improved Quantisation

Instead of the 25 bit-per-frame quantisation used in the Federal Standard, a 21bit-per-frame switched predictive quantisation scheme using a theoretically optimised LSF weighting function is used.

Fourier Coefficients

The Fourier coefficients are not transmitted at all in this vocoder, which gives a saving of 8 bits in return for a small loss in performance. However, this loss in performance seems to be compensated for by the gains made in other parameters.

Other Parameters

1. Pitch and voicing are quantised using only six bits instead of seven.
2. Only four possible voicing patterns are allowed.
3. Gain is only transmitted once per frame.
4. The jitter flag is replaced by an equivalent pitch contour perturbation technique.

4.3.3 MELP at 600bps

In 2001 Chamberlain [13] proposed a 600bps vocoder based on the MELP voice model. In this vocoder, the analysis and synthesis are done on 25ms segments. However, four consecutive speech frames are encoded together in order to exploit the substantial inter-frame redundancy which may be observed in the MELP speech parameters. A total of 60 bits are used per 100ms encoding super-frame (4 analysis frames). The encoding structure is as follows

Parameter	Number of Bits Allocated
Voicing	4
Energy	11
Pitch	7
Spectrum	38

Table 4.3: *Bit Allocation in Chamberlain's 600bps MELP vocoder.*

The following obvious changes are made from the standard 2400bps coder described in MIL-STD-3005:

Aperiodic Flag; The aperiodic flag is omitted from this version of MELP. Chamberlain justifies this decision by stating that at this bit-rate, more significant improvements may be obtained by better quantisation of the other speech parameters than by the inclusion of the aperiodic flag.

Band-Pass Voicing Quantisation; Table 4.4 shows the probabilities of occurrence of the various band pass voicing states. From the table it is clear that the band-pass voicing may be quantised to only two bits with very little audible distortion. A further gain is achieved by exploiting the inter-frame redundancy of the band-pass voicing parameters. In this way, Chamberlain manages to compress $4 \times 5 = 20$ bandpass voicing bits into only 4 bits. Chamberlain states that at this level of quantisation some audible differences are heard in the synthesised speech, but that the distortion caused by the band-pass voicing is not offensive.

Voicing States (Lowest Band to Highest Band)	Probability of Occurrence
U U U U U	0.15
V U U U U	0.15
V V V U U	0.11
V V V V V	0.41
Other	0.18

Table 4.4: *MELP band-pass voicing probabilities*

Energy; The energy parameter of the MELP vocoder exhibits considerable inter-frame redundancy. In order to exploit this redundancy, Chamberlain uses vector quantisation of eight energy values (two per frame) for every super-frame. The vector quantisation of the frames is trained using training data scaled by multiple levels, in order to prevent input level sensitivity of the energy quantisation.

Fourier Magnitudes; Chamberlain opts to not transmit any of the Fourier magnitudes. Instead, a single glottal excitation vector is used for each of the two excitation modes. Chamberlain provides no specifics as to the nature of the excitation vector but simply states that it reduces the perceived harshness of the synthesised speech. He further mentions that the loss in quality caused by the reduced rate minimises the perceived effect of the degradation caused by not transmitting the Fourier magnitudes.

Pitch Quantisation; Chamberlain again used four dimensional vector quantisation to encode four successive pitch values using a 128-element codebook trained with the k-means algorithm. In each super-frame, a mean-squared error criterion is then used to select an optimal pitch candidate. He uses the mean squared error of the actual pitch *frequency* in the voiced frames.

Spectrum; A four-stage vector quantisation is used, with the first two stages using ten bits each and the final two stages using nine bits each. Chamberlain emphasises the choice of training corpus. Since the quantiser relies on phoneme transitions more than simply on phonemes, one needs a very representative set of data in order to properly train the quantiser.

4.4 Conclusion

All three of the vocoders mentioned in this chapter use the LP speech model, again confirming the statement made by Deller [20] about the ubiquity of this model. However, the fundamental difference between the three models lies in the sophistication of the excitation model. Certainly, the model employed by the CELP vocoder is the most sophisticated of the three, but the simple MELP model proved to be at least as accurate in modelling the glottal excitation. As [55] mentions, the MELP and CELP vocoders achieve almost equivalent quality, despite the fact that the MELP vocoder is designed to operate at one half the bit-rate of the CELP vocoder. This huge improvement may be due in part to the more sophisticated algorithms used by MELP for pitch tracking, parameter quantisation and adaptive enhancement of the synthesized speech. However, it still presents a very convincing argument in favour of the MELP voice coding model. We will use the MELP model as a basis for the following chapters and we will follow the standard implementation described in [69] since it is the most well documented and commonly used version of MELP.

Chapter 5

Implementation of a MIL-STD-3005 compliant MELP Vocoder

The MELP vocoder was chosen for implementation since it represents an extremely good compromise between bandwidth efficiency and overall voice quality. Additionally it is a very widely accepted vocoder and had been standardised both by the US (MIL-STD-3005) and NATO (STANAG 4591).

The MELP vocoder was implemented in MATLAB. The following chapter describes its implementation.

The MELP Encoder may be divided into 2 components:

Analysis; The purpose of this component is to extract the parameters of the MELP speech production model which would synthesise a waveform as close as possible to the input speech waveform.

Encoding; The purpose of this component is two-fold. Firstly, the parameters are quantised and secondly the parameters are encoded and packed into a bit-stream for transmission.

Similarly, the MELP Decoder consists of the following components:

Decoding; The purpose of this component is to unpack the transmitted bit-stream, correct any detectable transmission errors and re-construct the MELP model parameters.

Synthesis; The purpose of this component is to synthesise the speech waveform from the model parameters.

We note that these components are symmetrical about the transmission. Implementing the MELP vocoder as two separate layers, allows us to verify the components individually, as illustrated in figure 5.1. By synthesising with the un-quantised analysis parameters, we may test the accuracy with which the MELP model can synthesise the target speech. Additionally, by directly investigating the model parameters before and after a simulated transmission we may investigate the quantisation errors directly as well as investigate the effect of bit errors on the decoding of model parameters.

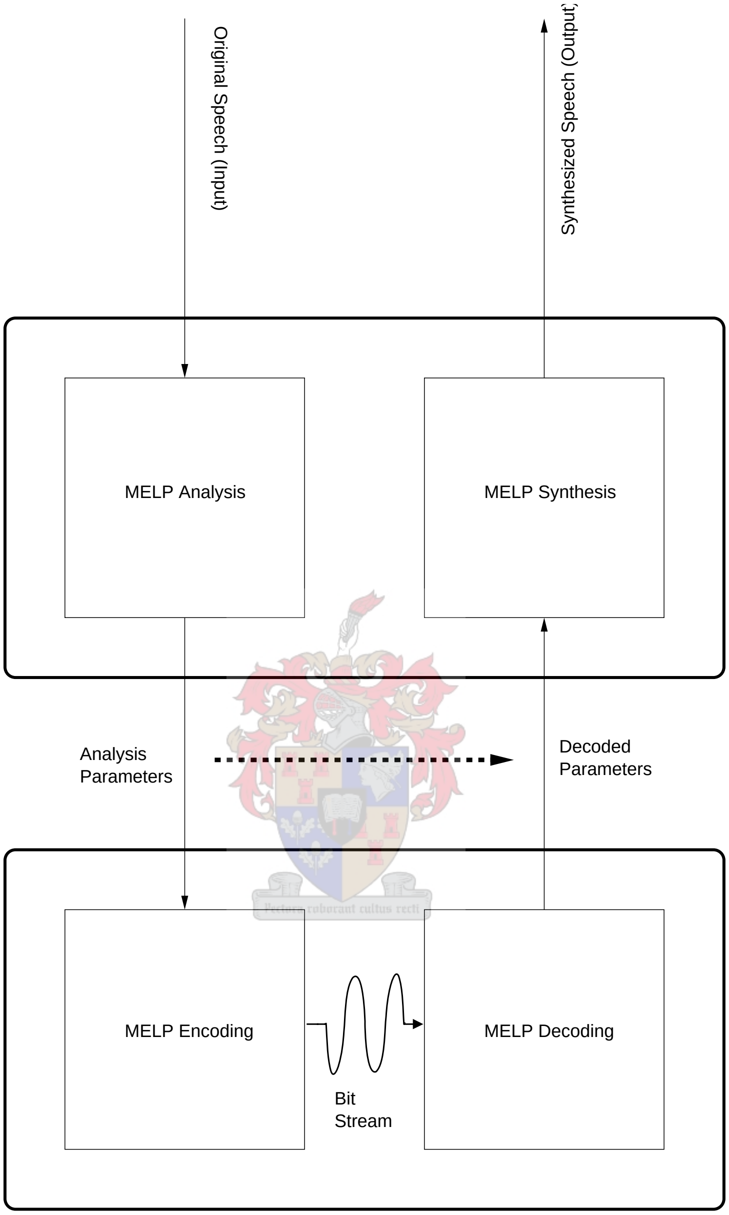


Figure 5.1: *MELP vocoder layers*

5.1 Analysis

A block diagram of the MELP analysis system is presented in figure 5.2.

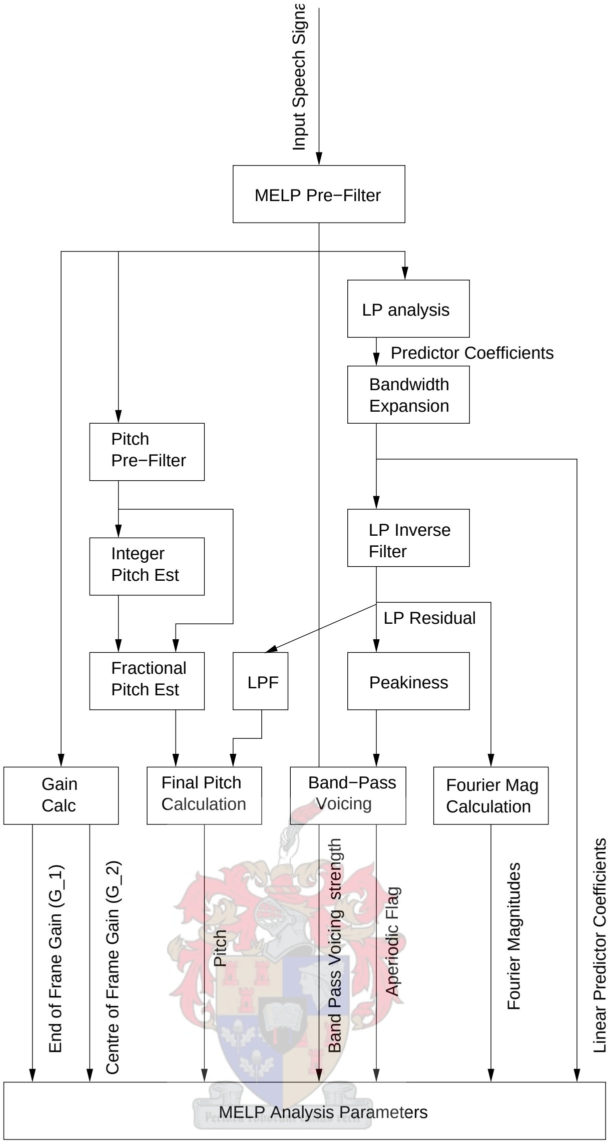


Figure 5.2: *Block diagram of MELP analysis algorithm*

5.1.1 Pre-Processing

The pre-processing of the speech consists solely of a DC removal. The input speech is filtered with a 4th order Chebyshev high-pass filter with a cutoff frequency of 60Hz, as shown in figure 5.3. The purpose of this filter is to remove extreme low-frequency components of the input signal which are inaudible and would interfere with the parameter estimation.

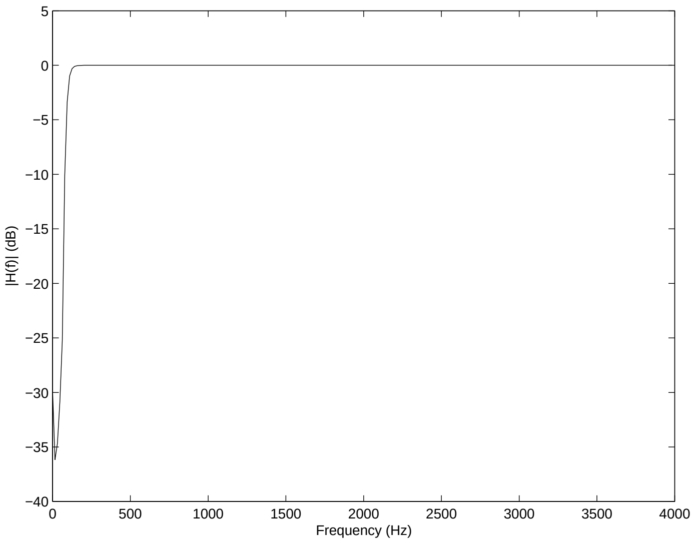


Figure 5.3: *Transfer function of MELP pre-filter.*

5.1.2 Pitch Estimation Pre-Processing

Before the initial pitch estimate is performed, the speech waveform is low-pass filtered with a 6th order Butterworth low-pass filter with a cutoff frequency of 1kHz. This filter removes the higher frequency components which are not necessary for pitch estimation but which may interfere with the pitch estimation algorithm. The transfer function of this filter is illustrated in figure 5.4

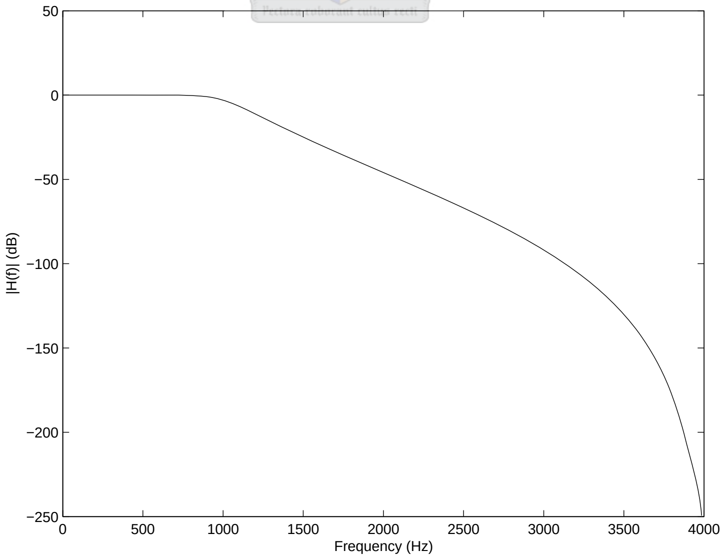


Figure 5.4: *Transfer function of MELP pitch estimation pre-filter.*

5.1.3 Integer Pitch

The integer pitch estimate is calculated by determining the maximum of the normalised autocorrelation of the speech segment. This function is calculated as follows:

$$r(t) = \frac{c_\tau(0,\tau)}{\sqrt{c_\tau(0,0)c_\tau(\tau,\tau)}} \tag{5.1}$$

where

$$c_\tau(m,n) = \sum_{k=-\lfloor\tau/2\rfloor-80}^{-\lfloor\tau/2\rfloor+79} s_{k+m}s_{k+n} \tag{5.2}$$

and τ is the lag in integer number of samples.

In [33], the use of this function over the usual autocorrelation is justified as follows:

The correlation value at $1T_p$, $2T_p$ and $3T_p$ are often very nearly the same for voiced segments of speech, due to the harmonic structure of the excitation waveform. However, rapidly increasing speech power (which is often observed at the onset of voicing) may bias the autocorrelation towards the higher multiples of the pitch period (corresponding to the lower harmonics of the pitch frequency). The normalised autocorrelation compensates for these sudden onsets of speech energy.

5.1.4 Fractional Pitch Estimate

The true pitch of the original continuous speech waveform is not an integer value as suggested by the discrete time autocorrelation, but a real number. For this reason inaccuracies in the pitch analysis may be caused by sampling effects. These effects may be minimised by performing the pitch tracking on the speech waveform at a much higher sampling rate. However, this would result in substantially more computational load and it is much more efficient to perform the interpolation on the autocorrelation function. The following interpolation formula is used:

First the fractional component of the pitch is calculated:

$$\Delta = \frac{c_T(0,T+1)c_T(T,T)+c_T(0,T)c_T(T,T+1)}{c_T(0,T+1)[c_T(T,T)-c_T(T,T+1)]+c_T(0,T)[c_T(T+1,T+1)-c_T(T,T+1)]} \tag{5.3}$$

Then Δ is used to calculate the interpolated value of the normalised autocorrelation:

$$r(T_p+\Delta) = \frac{(1-\Delta)c_T(0,T)+\Delta c_T(0,T+1)}{\sqrt{c_T(0,0)[(1-\Delta)^2c_T(T,T)+2\Delta(1-\Delta)c_T(T,T+1)+\Delta^2c_T(T+1,T+1)]}} \tag{5.4}$$

The interpolated maximum value of the auto-correlation for this lowest band is used as the overall voicing strength of the frame.

5.1.5 Band-Pass Voicing Analysis

For the band-pass voicing analysis, as described in section 4.3.1, the input signal is first divided into 5 sub-bands. In each of these sub-bands, the voicing strength (auto-correlation maximum) and fractional pitch (position of the auto-correlation maximum) is estimated using equations 5.3 and 5.4 respectively. The calculation of 5 (potentially) different pitch values may seem redundant but one must bear in mind that the sampling and filtering may result in the peaks of the autocorrelation being slightly different in the various bands. The different pitch values allow the analysis to take each maximum value into account. The pitch and voicing strength in the lowest band has already been calculated. The higher band fractional pitch values are calculated around the integer pitch value of the lowest band.

5.1.6 Linear Predictor Analysis

A 10th order LP analysis is performed on the speech frame. This is done in the standard way.

1. Window the frame with a Hamming window.

2. Calculate the auto-correlation

3. Using Levinson-Durbin recursion calculate the LP coefficients.

Additionally, the LP coefficients are bandwidth-expanded by multiplying each LP coefficient, a_i by 0.994^i for $i = 1, 2 \ldots, 10$. The reason for this is described by [15]. The optimal linear predictor returned by L-D recursion will of course be stable. However, it may have poles which lie close to the unit circle. Furthermore, these poles may shift slightly due to numerical effects. This may result in a filter which is unstable or very nearly unstable. Such a filter will produce audible chirps and squeaks in the synthesised signal. By moving the poles of the filter further away from the unit circle, the occurrence of these chirps is minimised. Furthermore, bandwidth expansion has the effect of shortening the duration of the impulse response of the filter which results in shorter transient effects when the excitation signal changes, such as at the onset of voicing.

5.1.7 LP Residual Calculation

The LP residual is calculated by filtering the frame with the inverse of the LP filter. The inverse filter is implemented in the usual way, by inversion of the transfer function of the LP all-pole filter.

5.1.8 Peakiness

The notion of peakiness has previously been used in 4.3.1. High peakiness of the predictor error signal is of course a strong indication of a voiced sound. The peakiness of the residual signal is calculated using equation 4.2. High peakiness values in the LP error signal cause the band-pass voicing strengths to be forced to high values, since this implies that the excitation was contained in a number of high amplitude samples and this type of excitation is characteristic of voiced speech.

5.1.9 Jitter (Aperiodic) Flag

An essential part of the MELP model is that of partially aperiodic pulses during voiced excitation. The encoder signals that the excitation is aperiodic by means of the jitter flag. This flag is set when the maximum interpolated autocorrelation near the pitch period (or voicing strength) as calculated in equation 5.4 is less than 0.5.

5.1.10 Final Pitch Calculation

The purpose of the final pitch calculation is to use the information from both the predictor residual signal and the input signal to determine a final pitch value for the frame. First the LP residual signal is low-pass filtered. The fractional pitch estimate is performed on this low-pass filtered residual signal and on the original speech signal, producing two pitch and voicing strength candidates for the frame. The candidate which has the highest voicing strength is used for the remainder of the algorithm, and will be referred to henceforth as the pitch signal.

Finally, a pitch doubling check is performed using the final pitch value. The pitch doubling check proceeds by evaluating the normalised auto-correlation of the pitch signal at fractions of the pitch value. If a candidate pitch (which is a fraction of the current pitch) is sufficiently 'good' (i.e. the relevant segment has a sufficiently high auto-correlation at that pitch value), then it replaces the current pitch as the final pitch for the frame.

5.1.11 Gain

The gain of the segment is calculated over two sub-frames, one centred on the centre of the analysis frame and referred t as G_1, the other centred over the end of the analysis frame and referred t as G_2. The length of these sub-frames is set as the lowest multiple of the pitch period greater than 120.

5.1.12 Fourier Magnitudes

The Fourier magnitudes are evaluated using the FFT magnitude of the LP residual, zero-padded to 512 samples, as described in section 4.3.1. The value of each Fourier magnitude is the maximum value of the FFT within one half of a pitch period from each harmonic of the pitch. If the pitch period is small and the frequency therefore high, the pitch will not necessarily have 10 harmonics within the Nyquist bandwidth. In this case the harmonics of the pitch which are greater than the Nyquist frequency are set to 1. The vector of Fourier magnitudes is normalised to have a Euclidean magnitude (RMS value) of 1.

5.1.13 Average Pitch Calculation

In order to ‘coast’ over regions of indeterminate pitch, the encoder tracks an ‘average’ pitch value. This is in fact a *median* value of the previous three strong pitch candidates. A strong pitch candidate is characterised by both a large energy in the frame and a strong auto-correlation. In this case the oldest of the three values in the history is replaced with the current frame pitch. If the current pitch value does not meet the criteria, all three candidates in the pitch history decay to a default value with a gain of 0.05 (this represents an approximate time constant of approximately 400ms if only indeterminate pitch values are received).

5.2 Encoding

5.2.1 Band-Pass Voicing Quantisation

The voicing of the segment is quantised according to the following rules:

1. If the lowest band voicing strength is less than 0.6, the frame is regarded as unvoiced and all higher band voicing strengths are set to 0.

2. If the lowest band voicing strength is greater than 0.6, the frame is regarded as voiced and all higher band voicing strengths are quantised to 0 only if their voicing strength is less than 0.6.

3. If only the highest band voicing and the lowest band are voiced then the highest band is forced to be unvoiced.

5.2.2 Linear Predictor Quantisation

The linear predictor is quantised as line spectrum frequencies. The LSFs are forced to be in ascending order and have a minimum separation of 50 Hz. The resulting LSF vector is quantised using a four stage MSVQ. The codebook consists of a first stage of 128 entries and three subsequent stages of 64 entries. The quantisation metric used is a simple weighted Euclidean distance between the quantised and un-quantised LSF vectors.

$$d(f,\hat{f}) = \sum_{i=0}^{9} w_i (f_i - \hat{f}_i)^2 \tag{5.5}$$

The weighting vector of the metric is calculated as follows: Let $P(f) = \frac{1}{A(2\pi \frac{f}{f_S})}$, i.e. let $P(f)$ be the inverse prediction filter transfer function evaluated at f Hz.

Then

$$w_i = \begin{cases} P(f_i)^{0.3} & , \quad i = 0 \ldots 7 \\ 0.64P(f_i)^{0.3} & , \quad i = 8 \\ 0.16P(f_i)^{0.3} & , \quad i = 9 \end{cases} \tag{5.6}$$

Since the codebook used for quantisation is essential to the interoperability of the system, it is presented in the standard.

5.2.3 Gain Quantisation

The first gain value (referred to in section 5.1.11 as G_2) is quantised using a 32 level uniform quantiser between 10 and 77 dB. The second gain value (G_1) is quantised using an adaptive quantisation algorithm. The algorithm compares the G_2 of the previous and current frame. If the two gain levels differ by less than 5dB and the measured value of G_1 differs from the mean of the two G_2 values by less than 3dB, then a special index is sent to indicate that G_1 is approximately the mean of the 2 G_2 values. Otherwise, G_1 is quantised using a seven level uniform quantiser.

5.2.4 Pitch Quantisation

The logarithm of the final pitch value is quantised using a 99 level uniform quantiser. The output index of the quantiser is encoded using a lookup table. The purpose of the lookup table is to protect the overall voicing of the segment. The lookup table is structured in such a way that single transmission errors in the overall voicing may be corrected and double errors may be detected.

5.2.5 Quantisation of Fourier Magnitudes

The Fourier Magnitudes are quantised using a standard single stage vector quantiser, with a weighted Euclidean distance metric using fixed weights to emphasise low frequencies. The values of the weights are given by:

$$w_i = \left[\frac{117}{25 + 75\left(1 + 1.4\left(\frac{f_i}{1000}\right)^2\right)^{0.69}} \right], i = 1, 2, \ldots, 10 \tag{5.7}$$

In the above, $f_i = \frac{8000i}{60}$, or the i'th harmonic of a default pitch period of 60 samples (133 Hz).

The following figure shows the weights used at the various harmonics of a default pitch of 60 samples.

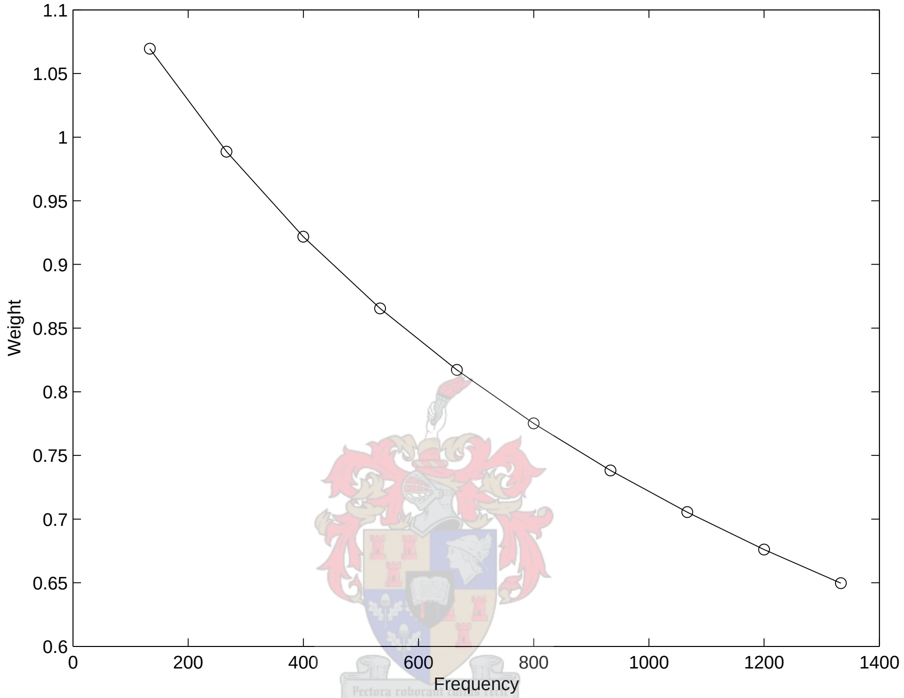


Figure 5.5: *Weighting vector of MELP fourier series quantisation metric, at approximate frequency values of pitch harmonics.*

5.2.6 Redundancy Coding

In unvoiced mode the following bits are not used:

1. Fourier magnitudes
2. Bandpass voicing strengths
3. Aperiodic flag

This makes available thirteen bits for error correction. The available bits are used for error correction in the following way:

1. The four most significant bits of the LSF MSVQ first stage index are protected with a Hamming (8,4) code.

2. The three least significant bits of the LSF MSVQ first stage index are protected with a Hamming (7,4) code. The fourth data bit of this codeword is always set to 0.

3. The four most significant bits of the $G2$ codeword are protected with a Hamming (7,4) code.

4. The least significant bit of the $G2$ codeword combined with the G_1 codeword are protected with a Hamming (7,4) code.

5.2.7 Transmission Order

The Standard (MIL-STD-3005) specifies a transmission order for the bits. Before transmission, the bits are re-arranged in the order of transmission.

5.3 Decoder

5.3.1 Error Correction

The (7,4) Hamming code will correct a single bit error, while the (8,4) Hamming code will additionally detect the presence of two bit errors. Additionally, the pitch encoding allows the detection of one or two bit errors. If uncorrectable bit errors are detected, a frame erasure is signalled. In this case the MELP synthesiser is simply given the parameters of the previous frame again.

5.3.2 LP and Fourier Magnitude Reconstruction

The linear predictor and fourier magnitudes are reconstructed using the received codebook indices and the respective codebooks using the standard MSVQ decoding algorithm described in appendix E

5.4 Synthesis

5.4.1 Pitch Synchronous Synthesis

In the MELP synthesiser, the synthesised speech is generated pitch-synchronously. This means that the excitation energy is generated one pitch period at a time. Furthermore, parameters such as the LP parameters, gain and Fourier magnitudes are updated once per pitch period.

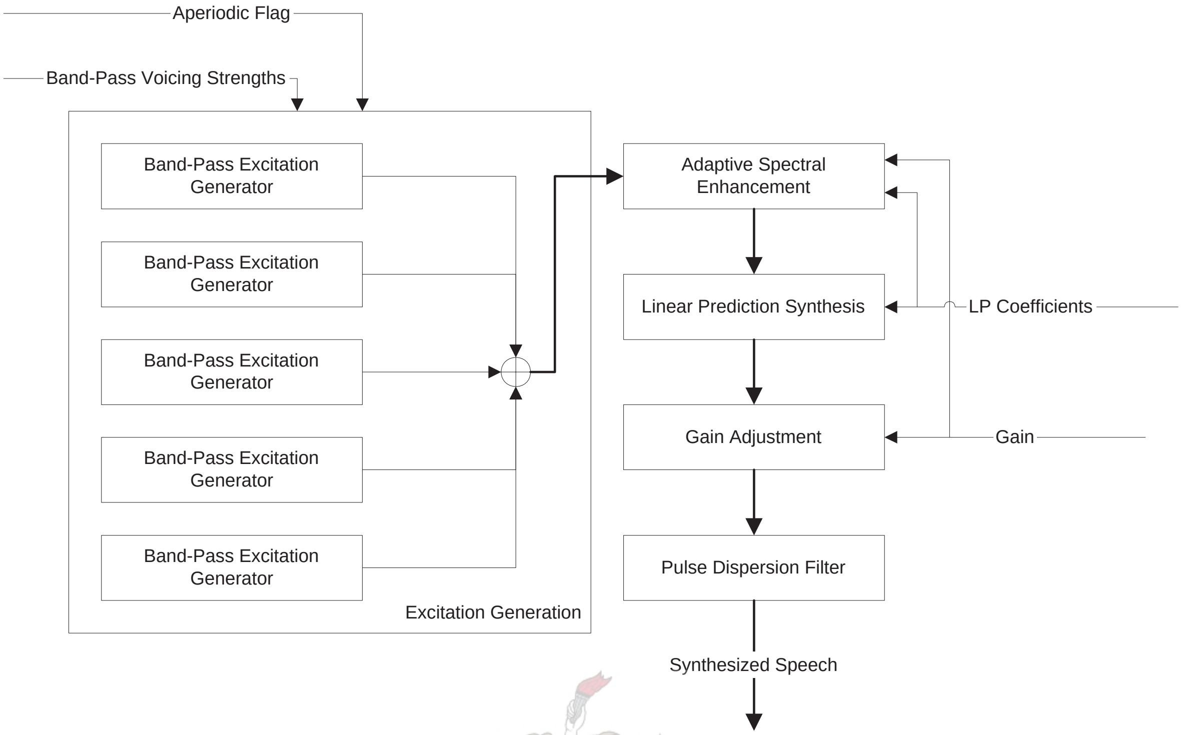

Aperiodic Flag
Band-Pass Voicing Strengths
Band-Pass Excitation Generator
Band-Pass Excitation Generator
Band-Pass Excitation Generator
Band-Pass Excitation Generator
Band-Pass Excitation Generator
Excitation Generation
Adaptive Spectral Enhancement
Linear Prediction Synthesis
Gain Adjustment
Pulse Dispersion Filter
LP Coefficients
Gain
Synthesized Speech

Figure 5.6: *MELP synthesis signal flow*

Flow Control

Since the synthesiser produces samples in block of length equivalent to approximately the pitch period, we will not always be assured that exactly 180 samples (or 22.5ms of analog output) will be produced per set of synthesis parameters. This means that the MELP synthesiser will continue to synthesise speech samples from the current received parameters until the level of the buffer of synthesised samples exceeds 180. At this point the oldest 180 samples of synthesised speech will be copied to output and the remaining samples will we kept in the buffer for use in the following frame.

5.4.2 Parameter Interpolation

The values of all parameters used by the MELP model to synthesise speech are linearly interpolated between the centres of successive frames.

$$X = \alpha X_{\text{next frame}} + (1 - \alpha) X_{\text{current frame}} \tag{5.8}$$

The position of the start of the current pitch period is used as the interpolation factor for the parameters.

$$\alpha = t_0/180 \tag{5.9}$$

The exception to this rule occurs at a sudden change in signal power, such as at the onset of voice activity or the onset of voiced speech. In this case, the trajectory of the MELP model parameter vector may change quite rapidly. Because the gain is measured more than once per analysis frame, the gain trajectory is a much more accurate interpolation factor in this case and the interpolation factor is calculated from the interpolated gain (G_{int}) and the current G_2 and sucessive G_{2p} gain values as:

$$\alpha = \frac{G_{int} - G_{2p}}{G_2 - G_{2p}} \tag{5.10}$$

5.4.3 Pitch Period

For voiced frames, the pitch period is used as the length of the excitation sequence. For unvoiced frames, a default length of 20 samples is used.

Jitter

In weakly voiced frames, indicated by the jitter flag, we add a uniformly distributed random integer to the pitch period. This random integer has a maximum absolute value of 25% of the pitch period and may be positive or negative. Thus the modified length of the synthesis segment is given by:

$$\hat{P} = P(1 + 0.25x), \quad x \in [-1, 1] \tag{5.11}$$

5.4.4 Impulse Generation

We generate the pulse shapes for voice excitation by performing an inverse DFT of one pitch period in length on the received Fourier coefficients.

5.4.5 Mixed Excitation Generation

The mixed excitation is generated using the sum of the outputs of the bandpass excitation generation filters. Each of these filters is driven with either a shaped impulse or with a segment of white noise, dependent on the detected voicing level in that band. Because of the linearity property of the filters, we may implement this simply by summing the filter taps corresponding to the voiced bands and summing the filter taps corresponding to the

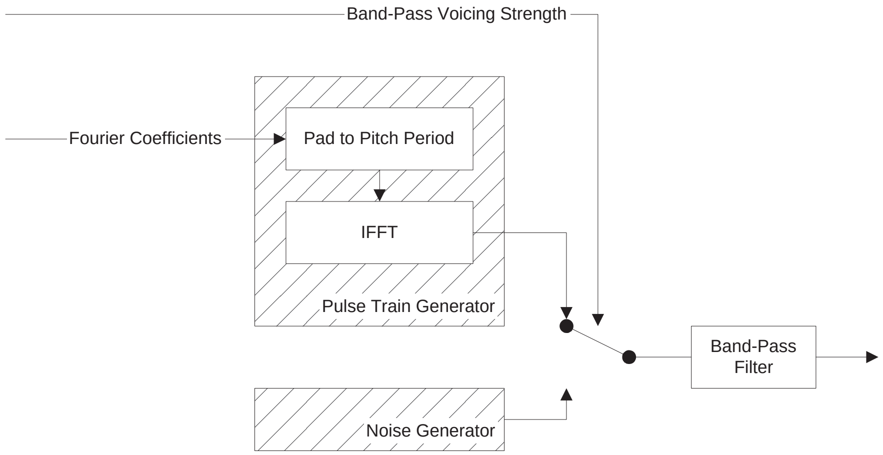


Figure 5.7: *Mixed Excitation Generation*

unvoiced bands and using these to filter the impulse and noise excitations respectively. This means that we need only sum two signals instead of five.

5.4.6 Adaptive Spectral Enhancement

The mixed excitation signal is filtered with the *Adaptive Spectral Enhancement* filter prior to LP synthesis. This filter is generated by bandwidth expansion of the linear prediction filter. The bandwidth expansion is performed according to the *signal probability*, which may be interpreted as the probability that the current synthesis frame represents a frame containing voice as opposed to simply background noise. The *signal probability* is estimated using the formula:

$$\rho = \frac{G_{int} - G_n - 12}{18} \tag{5.12}$$

Where G_{int} is the interpolated gain for the current pitch period and G_n an estimated gain for the background noise.

Additionally, the first reflection coefficient, k_1 is calculated from the decoded line spectral frequencies.

The signal probability and first reflection coefficient are used to calculate the bandwidth expansion factors, α and β as well as a *tilt coefficient*, μ. This is done as follows:

$$\alpha = 0.5\rho \qquad \beta = 0.8\rho \qquad \mu = max(0.5k_1, 0) \tag{5.13}$$

The transfer function $A(z^{-1})$ of the LP synthesis filter is calculated from the LSFs. This produces the transfer function of the ASE filter:

$$H_{ASE}(z) = \frac{A(\alpha z^{-1})}{A(\beta z^{-1})}(1 + \mu z^{-1}) \tag{5.14}$$

The purpose of the ASE filter is to emphasise the power of the synthesised speech near the formants, similar to the CELP postfilter described in section 4.2.3.

5.4.7 Linear Prediction Synthesis

The linear prediction filter uses a direct form realisation. The filter state is preserved between pitch frames and the coefficients of the filter in each pitch frame correspond to the interpolated LSFs for that frame.

5.4.8 Gain Adjustment

The excitation signal is generated at an arbitrary level and we therefore need to scale the synthesised speech so that the power of the synthesised frame corresponds to the power of the original speech frame. This is done by multiplying the pitch frame by a scaling factor. The scaling factor is calculated as follows:

$$S_{gain} = \frac{10^{\frac{G_{int}}{20}}}{\sqrt{\frac{1}{T}\sum_{n=1}^{T} \hat{s}_n^2}} \tag{5.15}$$

Naturally, one does not want to produce sudden changes in the signal amplitude and therefore the above gain value is linearly interpolated between the previous and current values over the first ten samples of the pitch period.

5.5 Results

In McCree and Barnwell [55], the MELP speech production model is presented and several of the key algorithms are described. In a subsequent paper by McCree and De Martin [56], several improvements to the model and the analysis and synthesis algorithms are proposed. Heuristic justification is given and subjective test results are presented which demonstrate the success of the overall system in each case. However, neither of the above-mentioned publications present detailed investigation into the effects of the various *individual* components of the vocoder.

5.5.1 MELP Analysis Results

Pitch Tracking

Plante et. al. [57] have made available a speech corpus, (referred to as the Keele corpus since it was produced at Keele University) designed to test the reliability of pitch tracking algorithms. This corpus consists of ten speech samples from different speakers (five male and five female speakers), each of which is approximately 60 seconds in length. Each speech sample is accompanied by a laryngogram signal, which may be used to calculate the pitch contour. Additionally, each sample is accompanied by a pitch contour sampled at 100Hz. This pitch contour was automatically generated from the laryngogram signal using the autocorrelation method but was further verified manually. The pitch contour also differentiates between voiced, unvoiced and indeterminate speech segments.

In our experiment we have separated the frames which are regarded as unvoiced as well as those which are regarded as indeterminate. The remaining frames of each speech segment are compared to the results obtained from the MELP pitch tracker. De Cheveigne [19] uses the gross pitch error rate as a measure of the accuracy of a pitch tracker. A *gross* pitch error is defined as a frame with measured pitch differing by more than 20% from the true pitch. In this paper various results from the above-mentioned corpus are also presented, which are useful for comparative purposes. De Cheveigne reports a gross pitch error rate of 2.8% using an optimised version of the normalised ACF - the same feature as used in MELP.

We used the Keele corpus to investigate the occurrence of pitch errors in the MELP pitch tracker. Results were obtained over the entire Keele speech corpus, for all voiced frames. $g(e_p)$ is the cumulative probability density function of the relative pitch error, where the relative pitch error p_e is defined in terms of the true pitch p_{true} and the estimated pitch p_{est} as:

$$p_e = \frac{|p_{true} - p_{est}|}{p_{true}} \tag{5.16}$$

.

Figure 5.8 indicate that the standard pitch tracking algorithm presented in MELP achieves a gross pitch error rate of 6.5% for the Keele corpus. This suggests that the MELP pitch tracker performs quite poorly compared to the results presented by De Cheveigne. Since a very similar feature is used for the results documented in De Cheveign's results, we conclude that the post-processing applied to pitch track by the MELP pitch tracker may be improved substantially.

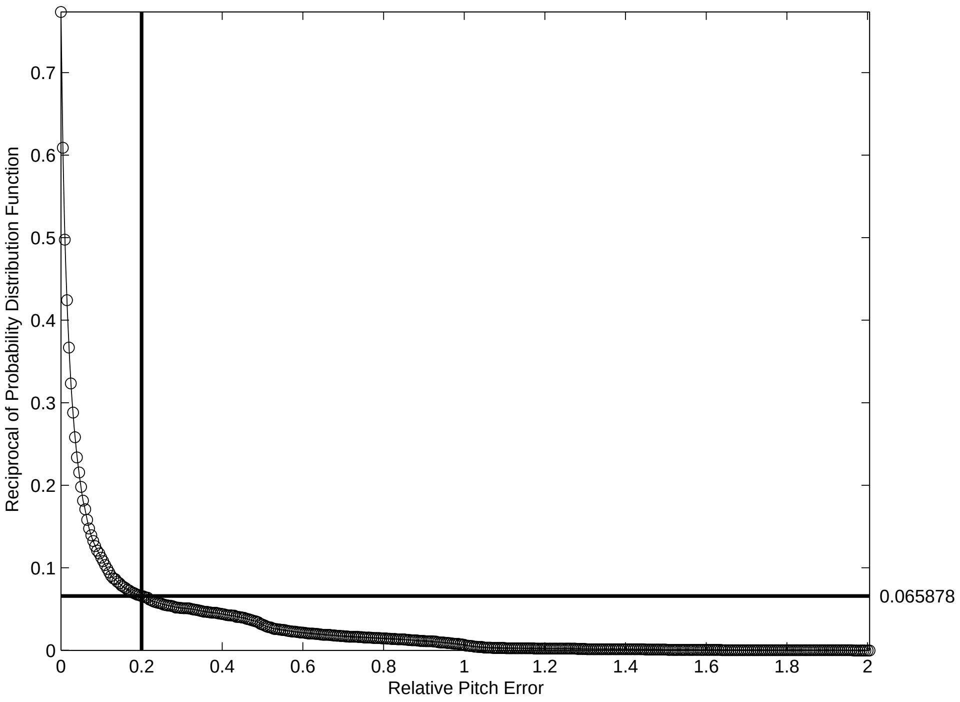


Figure 5.8: *Pitch Tracking Results From MELP Analysis.*

Voicing Analysis

The effect of the band-pass voicing analysis may be investigated by forcing all the higher band voicing strengths to conform to the voicing of the lowest band. This reduces the MELP speech production model to a simpler model, very close to the original LPC speech production model. It is well established in the literature [15, 33, 55] that the LPC model typically produces synthesised speech with a distinctly buzzy quality. We can reduce the buzziness somewhat by reducing the voicing decision threshold, but this would result in the synthesised voice taking on a hissy, whispering character.

We investigated the accuracy of the overall voicing decision using speech samples in which the voicing has been accurately determined. For this purpose we used the same speech corpus as above. The overall frame voicing as determined by the MELP algorithm is compared to the voicing as marked up in the corpus.

	Keele : Voiced	**Keele : Unvoiced**
MELP : Voiced	0.4854	0.0178
MELP : Unvoiced	0.2381	0.2586

Table 5.1: *Accuracy of MELP voicing decision*

In table 5.1, we demonstrate the frequency of occurrence of each of the four possible outcomes of the MELP estimated voicing and the true frame voicing as determined by the laryngiograph[1]. The MELP voicing decision exhibits a substantial bias toward an unvoiced decision. This appears to be a perceptually motivated design choice, since incorrectly classification of frames as voiced will result in very audible 'musical' tones in the synthesised voice, while the incorrect classification of frames as unvoiced will result in 'hisses' which are much less disturbing to listeners.

5.5.2 Quantisation Effects

Quantisation of LP Coefficients

Figure 5.9 illustrates the effect of quantisation on the frequency response of the linear prediction synthesis filter.

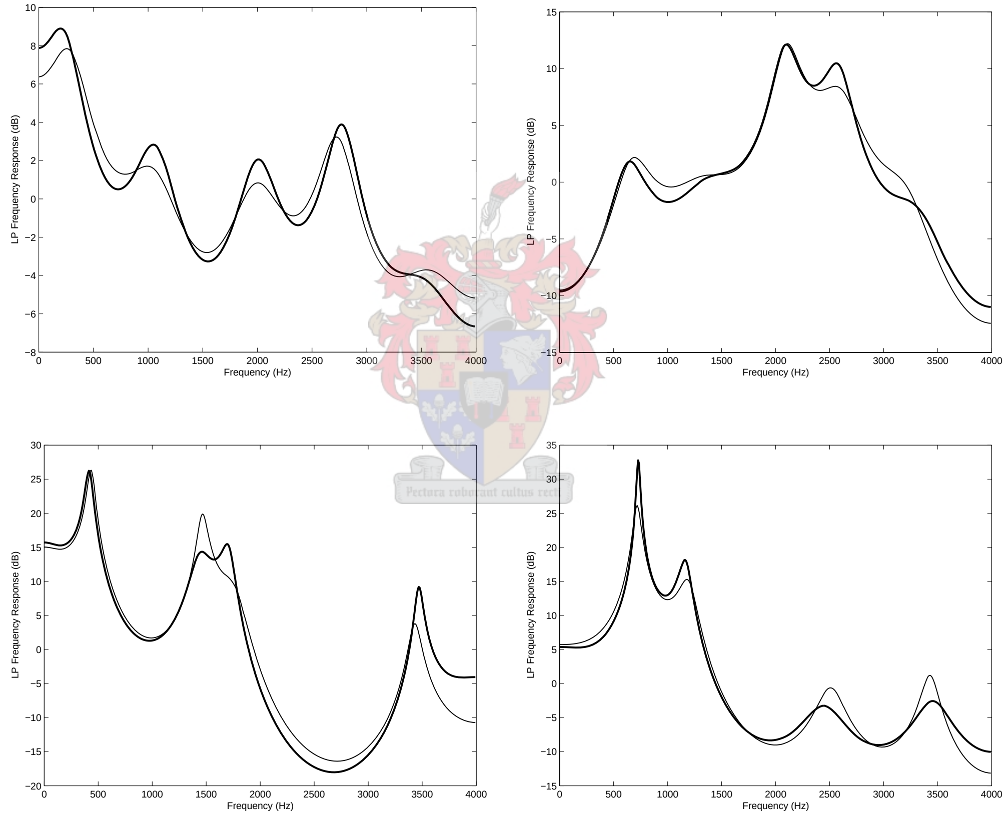


Figure 5.9: *Effect of quantisation on the LP. Figures show the frequency response of the MELP LP before (thin line) and after (thick line) quantisation for a few representative frames.*

[1] This implies that the columns and rows of the table will not sum to 1 but the four entries of the table sum to 1

We can examine the objective effect of LP quantisation using some of the objective speech metrics mentioned in 3.7. Paliwal and Atal [63], introduced the idea of *transparent quantisation* which has been discussed in appendix E. Transparent quantisation is defined in terms of the spectral distortion imposed by the quantisation. The spectral distortion is defined in terms of the frequency response of the original linear predictor ($A(f)$) and the transfer function of the quantised linear predictor ($A'(f)$). The spectral distortion (SD) is calculated as:

$$\mathrm{SD} = \frac{\int_{-\infty}^{\infty}(A(f) - A'(f))^2 \mathrm{d}f}{\int_{-\infty}^{\infty} A(f)^2 \mathrm{d}f} \tag{5.17}$$

. In a discrete time system, the integrals are usually approximated by sums.

Paliwal and Atal suggested 24 bits as being the minimum information necessary per frame in order to achieve transparent quantisation of the linear predictor power spectrum. Since MELP uses 25 bits per frame, we expect that transparent encoding of the LP should be possible. In order to test this, the per-frame Spectral Distortion was measured for a large speech corpus, taken from the TIMIT speech corpus. The estimated PDF of the per-frame SD produced by the MELP VQ is presented in figure 5.10.

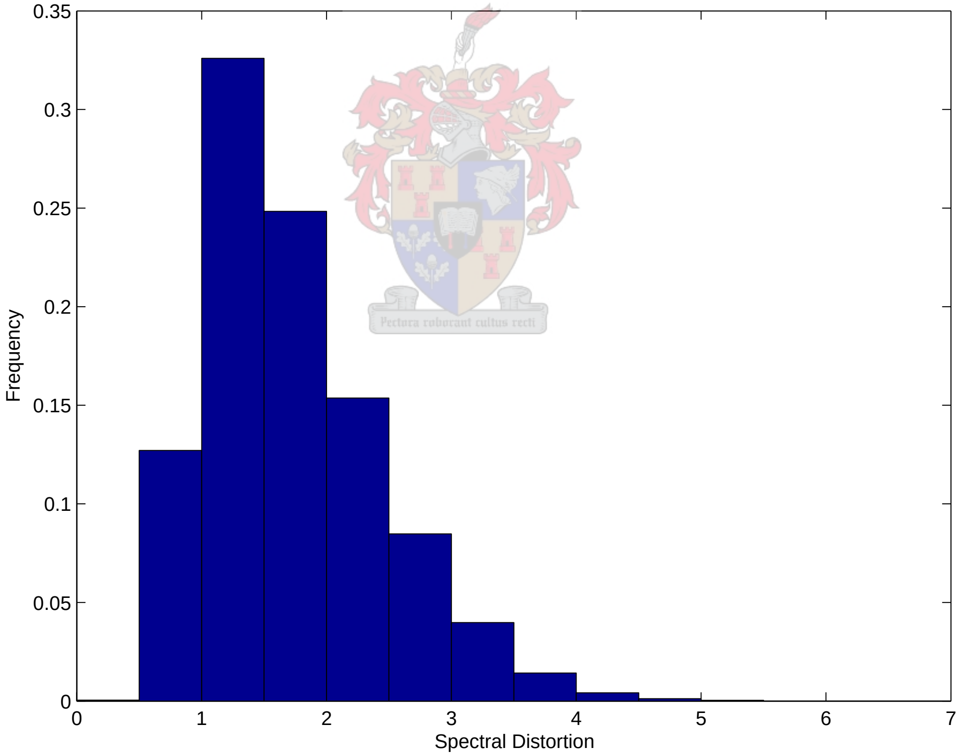


Figure 5.10: *Histogram of spectral distortion measured over 75000 frames of MELP encoded speech.*

The results here are comparable to the results reported by Paliwal and Atal using their Split VQ and MSVQ schemes (see appendix E), but the quantisation scheme in MELP definitely appears to introduce more distortion. There are many factors which might contribute to this, one possibility is the choice of speech corpus used for the evaluation. Paliwal and Atal used about 160s of speech recorded from radio stations for their tests, while we used 1600s from the TIMIT corpus.

However, investigating the SD histograms for various individual speakers presented the surprising result that the distribution of spectral distortion varied only slightly for the individual speakers, as illustrated in figure 5.11.

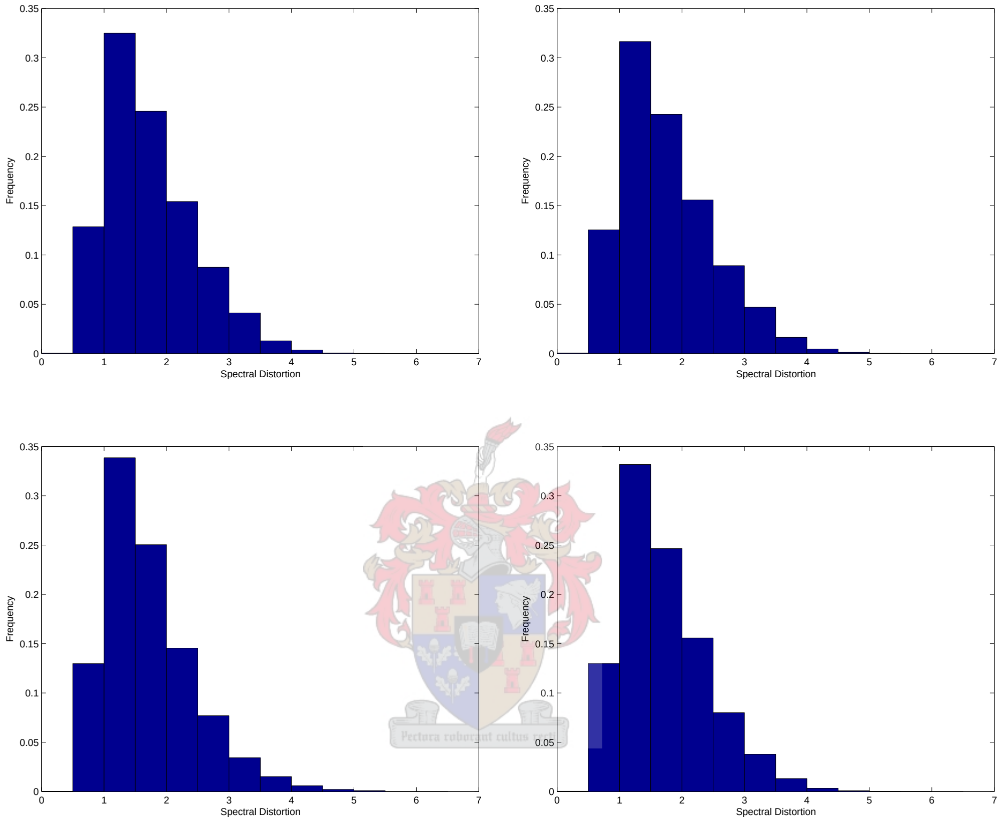


Figure 5.11: *Spectral distortion histogram plots for various speakers. In each sub-figure, the histogram of SD occurrence for a single speaker is shown.*

Another explanation would be that the two stage MSVQ or the two stage split VQ used by Paliwal and Atal represents a more efficient quantisation than the four stage MSVQ used by MELP. The decision to use a four stage MSVQ was most probably motivated by computational considerations. This idea has been discussed in appendix E.

5.6 Conclusion

The floating point ‘C’ reference implementation of the MELP algorithm is available at [61]. This implementation was used as a qualitative verification of the implementation.

This verification was done in three ways.

1. Comparison: Transcoded speech produced by the two implementations appeared to the author's ear to be identical.
2. The model parameters produced by analysis of speech signals by the two implementations were identical.
3. The two vocoders were found to be interoperable.

In the implementation of the MELP vocoder it was found that the overwhelming majority of the development effort was spent on the implementation of the finer details of the algorithm. While the analysis-synthesis model is conceptually simple, the MELP vocoder relies on a number of seemingly heuristic algorithms, which are justified neither in McCree and Barnwell's original paper nor in the MELP standard. Examples would be the pitch doubling check algorithm and the adaptive spectral enhancement filter. Additionally, the frame-based nature of the MELP analysis and synthesis engines mean that continuity constraints must be carefully enforced. Thus we hope that our MELP MATLAB model will represent a useful tool for future work.

In the following chapters we will consider some of the limitations of the MELP vocoder and investigate potential avenues of improvement.

Chapter 6

The Temporal Decomposition Approach to Voice Coding

In chapter 2, we discussed the *segmental* and *parametric* approaches to voice coding. Segmental vocoders achieve extremely low bit rates but have the disadvantage of being speaker dependent and have extremely high computational and memory costs [12]. Parametric coders achieve less impressive compression but are computationally more tractable, have substantially lower memory requirements and do not suffer from speaker dependence as severely.

Not a great deal of work has been published to bridge the substantial gap between these two approaches.

In this chapter we will examine some of the mathematical properties of the operation of a parametric voice coder in an attempt to improve on parametric voice coding by using some of the ideas typically associated with segmental voice coding.

6.1 History of Temporal Decomposition

This idea of temporal decomposition voice coding seems to have originated with Atal [3], who noted that the parameters of the speech waveform varied at different rates at different times and proposed the idea of *temporal decomposition.* Interpolating functions were used to reduce the bandwidth of reflection coefficients. No quantitative results were presented in the paper.

Ten years later Cheng and O'Shaughnessy [14] developed this idea further and presented the idea of *Short-Term Temporal Decomposition* (STTD). The stated goal of this algorithm is to match a vector trajectory in certain senses of optimality by means of sets of functions and weight vectors. The functions are called *event functions* and are sequential (in time) and overlapping. In this paper, a distortion measure similar to the *Itakura-Saito* measure described in section 3.7 was used, but no subjective test results were recorded.

Prandoni and Vetterli [65] describe an approach whereby the speech segmentation is optimised in the time domain in order to minimise a cost function based on the modelisation cost. This means that the original speech signal is examined on a sample by sample basis in order to determine the optimal segmentation on which to perform LP analysis and parameter estimation. Their approach was successful, in that they achieved very good rate-distortion trade-offs.

George [31] presented a block-coded variable frame rate strategy. This approach used dynamic programming to select optimal *break points* or frames for transmission. Few details of the algorithm are described in the paper and no quantitative results are documented.

We wish to derive a further mathematical theory of parametric voice representation, in order to better understand this approach.

6.2 A Mathematical Framework for Parametric Voice Coding

In order to attempt to derive a mathematical description of the function that a vocoder performs, we return to our original model of speech production, first described in 3.1.

Parametric voice coders are based on the idea that the speech signal can be represented by the variation of the parameters of the speech production model. This implies that there is a relation between the speech signal (which is a scalar function of time) and the parameters of the speech production model, which are a vector function of time.

6.2.1 Representation of the Speech Signal

We will denote the speech signal as function of time by s and the parameter vector as function of time by $\mathbf{p}$.

These two representations of the speech signal are linked by the speech production model. The model synthesises s from $\mathbf{p}$. The synthesiser also uses its internal state as an input into the speech sample which is produced, and the state of the synthesiser is modified by the input parameters as well as the sample which is produced. We therefore need to consider the entire speech waveform and the entire set of all sampled parameter values, so as to disregard the internal state of the synthesiser.

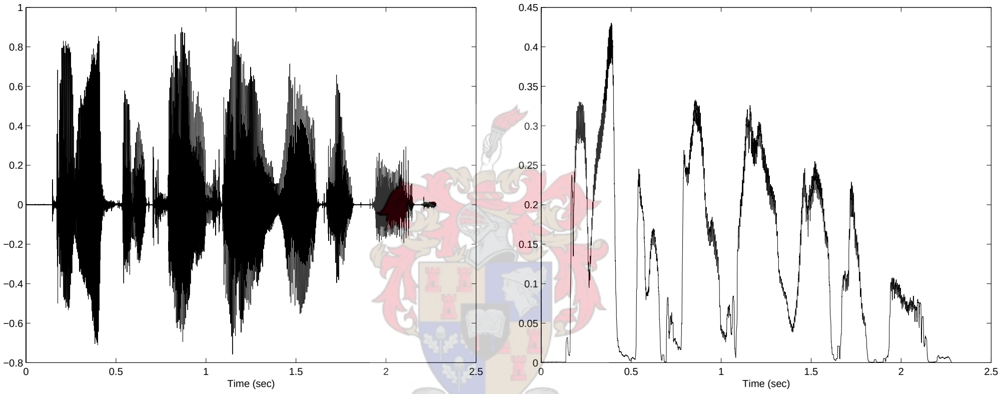
We will use the notation that $\mathbf{p}(t)$ is the parameter vector as a function at time t, and denote $\mathbf{p} = \mathbf{p}(t)\forall t$. Additionally, let $s(t)$ denote the synthesised speech signal at time t with $s = s(t)\forall t$. We may therefore define two sets, we will call these $\{\mathbf{p}\}$ and $\{s\}$ with $\{s\}$ the set of all possible speech waveforms and $\{\mathbf{p}\}$ is the set of all possible parameter vector functions.

Thus the parameter vector describes a trajectory through its *parameter* space at the same time as the speech waveform describes a trajectory in the time domain.

6.2.2 The Parameter Vector Trajectory

We will assume that if we should look at the trajectory of $\mathbf{p}$, we will see that at certain times $\mathbf{p}$ moves quite slowly and at other times moves very quickly. Additionally, we would expect that the trajectory is at times approximately linear [84], but also exhibits regions of high curvature. This idea is also expressed by Atal in [3].

In order to illustrate these ideas, we demonstrate the behaviour of a simple parameter vector transform of the speech waveform, where the parameter vector consists of only the instantaneous power of the waveform. We will regard this as our one dimensional feature vector for the sake of simplicity and in order to aid ease of visualisation of the ideas described.



(a) Speech waveform ($s(t)$) (b) Feature Vector ($\mathbf{p}(t)$)

Figure 6.1: *Time and Parameter domain representations of the speech signal*

As one can see in figure 6.1, the feature vector is characterised by long periods of fairly slow, linear behaviour, with occasional sharp, discontinuous changes. This is in agreement with Atal's observation that [3]:

"Uniform sampling of speech parameters is not efficient."

6.3 Parameter Space and Encoding

Vocoders encode voice by describing the above parameter trajectory efficiently. Typically a vocoder utilises a frame-based approach, with regularly spaced frames. This amounts to sampling the trajectory $\mathbf{p}$ at regular intervals. As described in section 3.4.1, the analysis intervals are chosen to be sufficiently small (and thus the rate at which **p** is sampled is chosen sufficiently high) in order to satisfy a sampling criterion analogous to the *Nyquist criterion* which is described in [77].

6.3.1 Irregular Sampling of the Parameter Vector

As mentioned above, the trajectory of the parameter vector is not necessarily a stationary, band-limited process. This would imply that the rate described above is occasionally lower and often higher than is necessary in order to represent accurately the trajectory of **p** with acceptable accuracy. Additionally, we expect that there are certain points in the trajectory which have a higher information content than others. In other words, given two sets (of the same cardinality) of points sampled from the trajectory, one set of points may allow a more accurate reconstruction of the trajectory than the other.

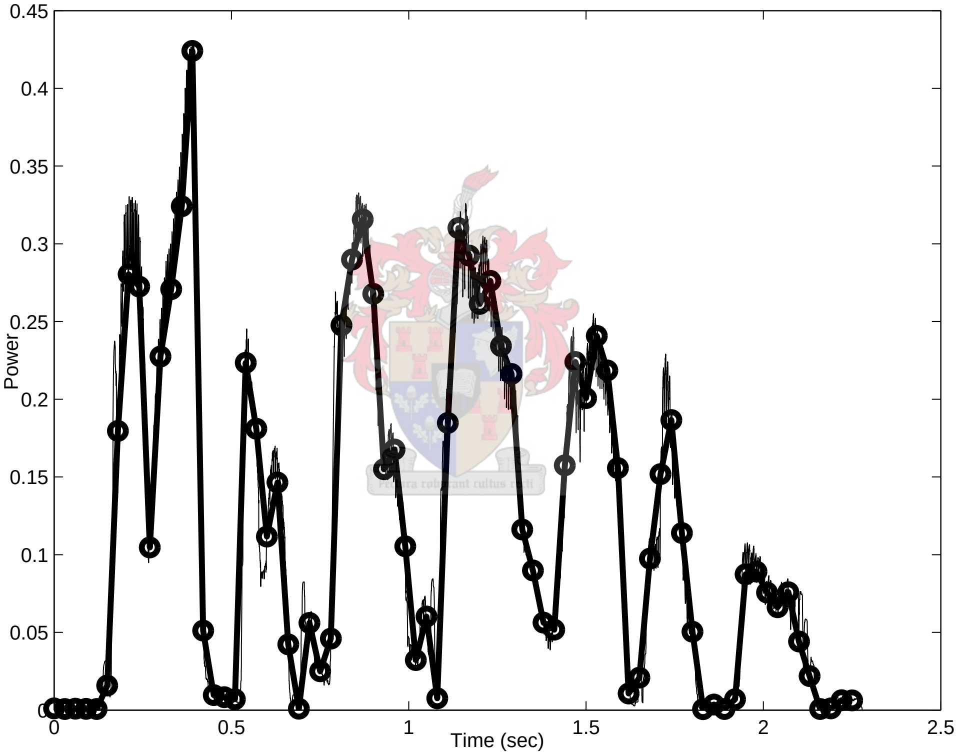


Figure 6.2: *Feature vector trajectory sampled at a typical vocoder sampling rate. The thin line indicates the feature trajectory. The markers indicate the point at which the feature trajectory was sampled and the thick line indicates the estimated value of the trajectory created by linear interpolation between sampling points.*

Most low rate speech coders calculate a feature vector approximately every 30ms. This corresponds to sampling **p** every 30ms. The effect of this is demonstrated in figure 6.2. In figures 6.2, 6.3 and 6.4, the thin line indicates the feature trajectory. The markers indicate the point at which the feature trajectory was effectively sampled and the thick line indicates the estimated value of the trajectory created by linear interpolation between sampling points. One can see that in this case the feature vector trajectory appears to be over-sampled in certain regions.

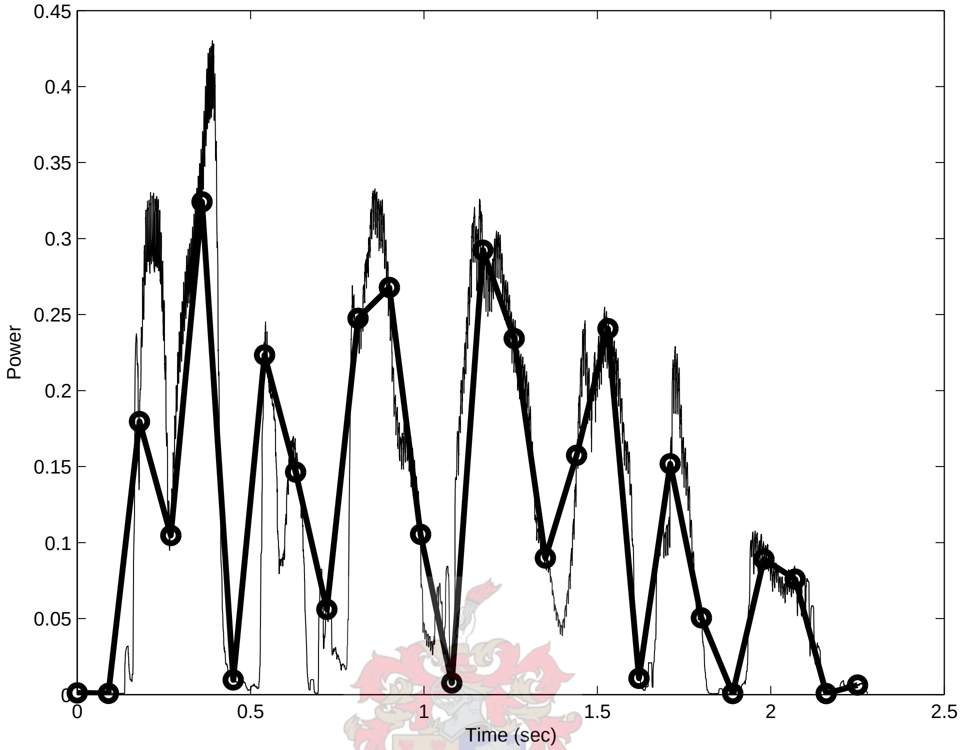


Figure 6.3: *Feature vector trajectory with 'under sampling'*

In figure 6.3, the sampling interval has been increased to 90ms, thus reducing the sampling rate by a factor of 3. Now much of the high-frequency behaviour of the feature vector seems to be lost. Clearly, simply increasing the sampling interval for regular sampling is not an acceptable solution.

In figure 6.4, an average sampling interval of 90ms has been used. However, the sampling points have been manually aligned with the most relevant points of the feature trajectory. Now much of the high-frequency behaviour of the feature vector has been preserved without increasing the number of samples.

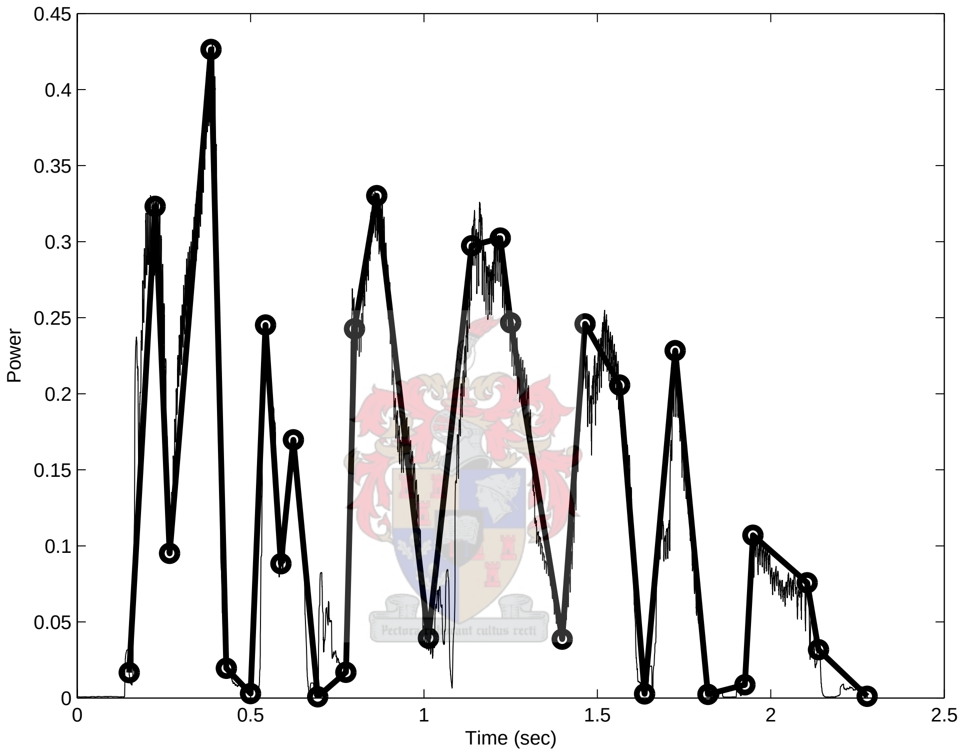


Figure 6.4: *Feature vector trajectory with irregular sampling*

6.4 Conclusion

In this chapter we have shown that irregular sampling of the speech parameter vector may lead to better encoding of the speech signal. In the following chapter we will describe the implementation of a vocoder which makes use of the ideas presented here.

Chapter 7

Implementation of an Irregular Frame Rate Vocoder

In the section 6.3, we illustrated how we may possibly represent the speech signal accurately with fewer sampling points using irregular sampling of the parameter trajectory.

In this chapter we will apply these ideas to the MELP speech production model described in chapter 4 and chapter 5, in order to develop a variable frame-rate vocoder.

The development of such a vocoder requires the following:

1. An algorithm to determine an accurate representation of the feature vector trajectory, by sampling $\mathbf{p}(t)$ at a high sampling rate.
2. A reconstruction algorithm, which can approximate $\mathbf{p}(t)$ from a set of feature trajectory samples,
$$\{\mathbf{p}[\tau_1], \mathbf{p}[\tau_2], \ldots, \mathbf{p}[\tau_N]\}.$$
We will refer to this approximation as $\tilde{\mathbf{p}}(t)$.
3. A corresponding decomposition algorithm to determine an optimal set of sampling points $\{\tau_1^*, \tau_2^*, \ldots, \tau_N^*\}$ so that the reconstruction will be as close as possible to the original for a given frame rate. In contrast to the analysis-by-synthesis approach taken in [3] and [14], we will attempt to determine the sampling points directly from analysis of the feature trajectory. We will refer to the above optimal set of points as the *key frames* for the speech segment.

This is illustrated in figure 7.1.

The way in which this has been implemented is as follows:

1. We adapted the analysis engine of the standard MELP vocoder to determine an over-sampled representation of the parameter trajectory.
2. We used simple linear interpolation to calculate $\tilde{\mathbf{p}}$ from $\{\mathbf{p}[\tau_1], \mathbf{p}[\tau_2], \ldots, \mathbf{p}[\tau_N]\}$.

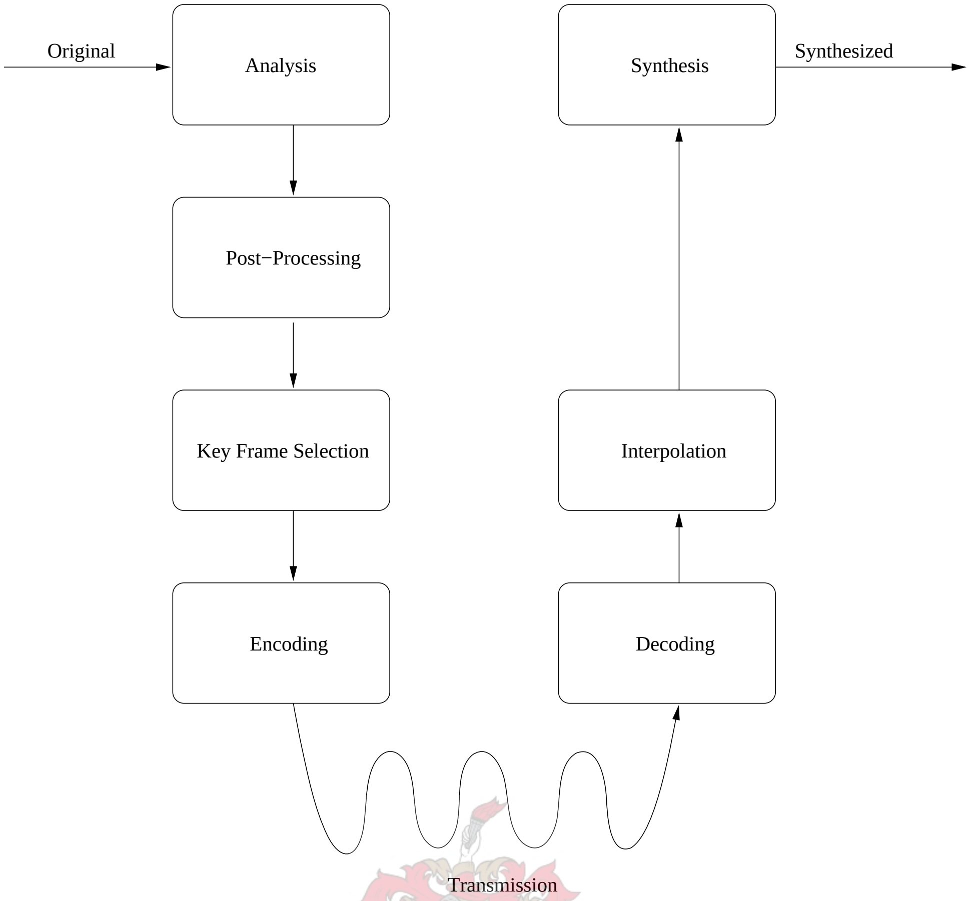


Figure 7.1: *IS-MELP Block Diagram*

3. We used the L_∞ norm to calculate a metric for any point chosen for transmission based on the preceding points.

$$\begin{aligned} E[k_n] &= \max_{j=k_{n-1}\ldots k_n} d\left(\mathbf{p}[j], \tilde{\mathbf{p}}[j]\right) \\ &= \max_{j=k_{n-1}\ldots k_n} d\left(\mathbf{p}[j], \alpha_j \mathbf{p}[k_n] + (1-\alpha_j)\mathbf{p}[k_{n-1}]\right) \\ & \quad \text{With } k_{n-1} < k_n,\ k_n \in \mathbf{Z},\ n \in \mathbf{N} \end{aligned} \tag{7.1}$$

Where $d(\mathbf{p}_1, \mathbf{p}_2)$ denotes an appropriate distance function between parameter vectors, as will be described in more detail later. α_j is the linear interpolation factor as used in the calculation of $\tilde{\mathbf{p}}[t]$ and is calculated as:

$$\alpha_j = \frac{j - k_{n-1}}{k_n - k_{n-1}}$$

Hence $E[k_n]$ represents the maximum distance between $\mathbf{p}[t]$ and $\tilde{\mathbf{p}}[t]$ between the points k_{n-1} and k_n.

4. We modified the MELP synthesis engine so that the synthesis frame length is also received and can vary.

We will call this approach to voice coding using the MELP voice production model Irregularly Sampled MELP or IS-MELP.

7.1 Analysis

In the IS-MELP analysis step, the input speech waveform is analysed using the standard MELP analysis engine as described in chapter 5. However, the IS-MELP analysis window is advanced by only 2.25 ms (or 18 samples) at a time instead of the 22.5ms (180 samples) by which the standard MELP analysis window is advanced. This results in a tenfold oversampling of the parameter trajectory.

The primary purpose of this over-sampling is that the oversampling allows for more accurate identification of the significant points in the speech parameter trajectory.

We determine the feature trajectory in our algorithm by performing MELP analysis on overlapping frames of the speech waveform. The standard MELP analysis is performed on analysis frame of 22.5ms, which is advanced by 22.5ms for every analysis. In our algorithm we attain a high-resolution view of the trajectory by advancing the analysis frame by only 2.25ms. This of course leads to substantial redundancy in the feature vector trajectory, analogous to the redundancy produced by over-sampling a band-limited signal. In order to utilise this redundancy to obtain a more accurate estimation of the trajectory, we will perform a filtering step on the feature trajectory.

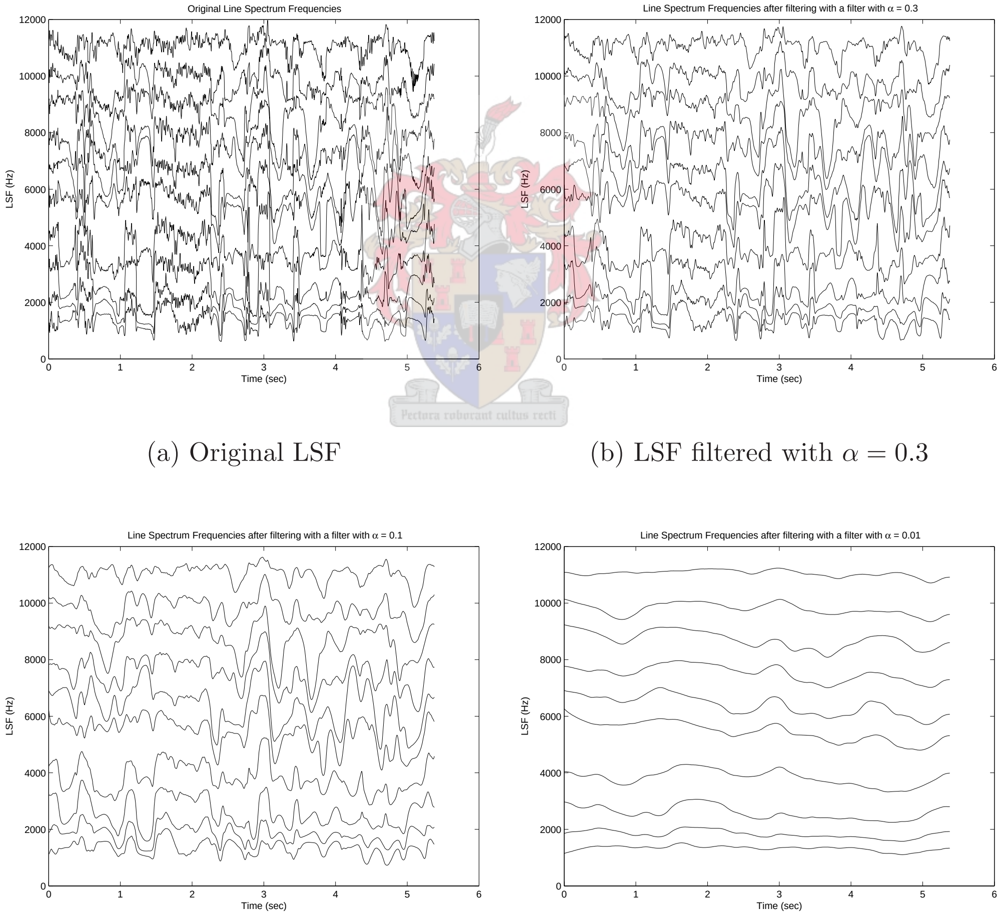
7.2 Filtering of the Speech Feature Trajectory

7.2.1 Pitch and Voicing

The pitch and voicing are filtered using the LULU filters described in appendix F. The aim of this filtering is to remove excursions in the pitch and voicing contours which have a duration less than the minimum duration which would have been allowed by the standard MELP vocoder. Omitting this post processing step result in occasional oscillations of the voicing and pitch, particularly in certain weakly voiced regions of speech. (This oscillation suggests that there is room for improvement in the standard algorithm used by MELP to make these decisions.) After the post-processing step, the occurrence of short voicing regions in the speech is minimised. These regions are undesirable because they affect the key frame selection algorithm, due to the large variation between successive frames and may result in frames being unnecessarily identified as key frames, thus increasing the number of frames which are transmitted and hence increasing the overall bit rate of the vocoder.

7.2.2 Linear Predictor

Post-processing is also applied to the linear predictor as represented by the LSFs. In this case the post processing takes the form of a low-pass filter which is applied to each LSF individually. As was described in 3.5.2, we are still assured of a stable linear predictor system after the filter is applied. The bandwidth of this IIR filter is of crucial importance. A filter with too wide a passband will not have the desired suppression of unwanted artifacts of the LP estimation process. An excessively narrow passband will remove too much of the rapid variation of the LP system which occurs at voicing transitions. Perceptually, this results in the speech assuming a 'slurred' characteristic, as the characteristics of successive phonemes are combined.



(a) Original LSF

(b) LSF filtered with $\alpha = 0.3$

(c) LSF filtered with $\alpha = 0.1$

(d) LSF filtered with $\alpha = 0.01$

Figure 7.2: *Effect of post processing filter on LSF.*

In figure 7.2, we illustrate the effect of the filter with the transfer function

$$H(z) = \frac{\alpha z^{-1}}{1 + (1 - \alpha) z^{-1}} \tag{7.2}$$

on the LSFs for various values of α. In the original LSFs, one notices that the LSFs exhibit a large amount of high-frequency behaviour. We postulated that this high frequency behaviour is not perceptually significant, and may be removed using a low-pass filter, with a suitable narrow bandwidth, such as the filter shown in figure 7.2(c) However, choosing α too small (and correspondingly, narrowing the bandwidth of the filter too much), causes a loss of resolution in the LSF trajectories and correspondingly, a loss in intelligibility of the synthesised speech. Ideally, one would determine the optimal filter cutoff by means of perceptual tests, but this was not feasible within the scope of this project.

7.3 Reconstruction of the Parameter Vector Trajectory

We use linear interpolation between the key vectors to determine the feature vector trajectory. This is computationally tractable and compatible with the standard MELP synthesis algorithm. This presents a substantial advantage of our method, namely that the standard MELP decoder may be used almost without modification.

Before we describe the algorithms used to determine 'good' choices of frames for transmission, we will first discuss the simplest possible method of frame selection.

7.4 Regular MELP as a special case of IS-MELP

Since the IS-MELP encoder is free to select frames at any sampling points, it may thus also select frames at regular sampling points.

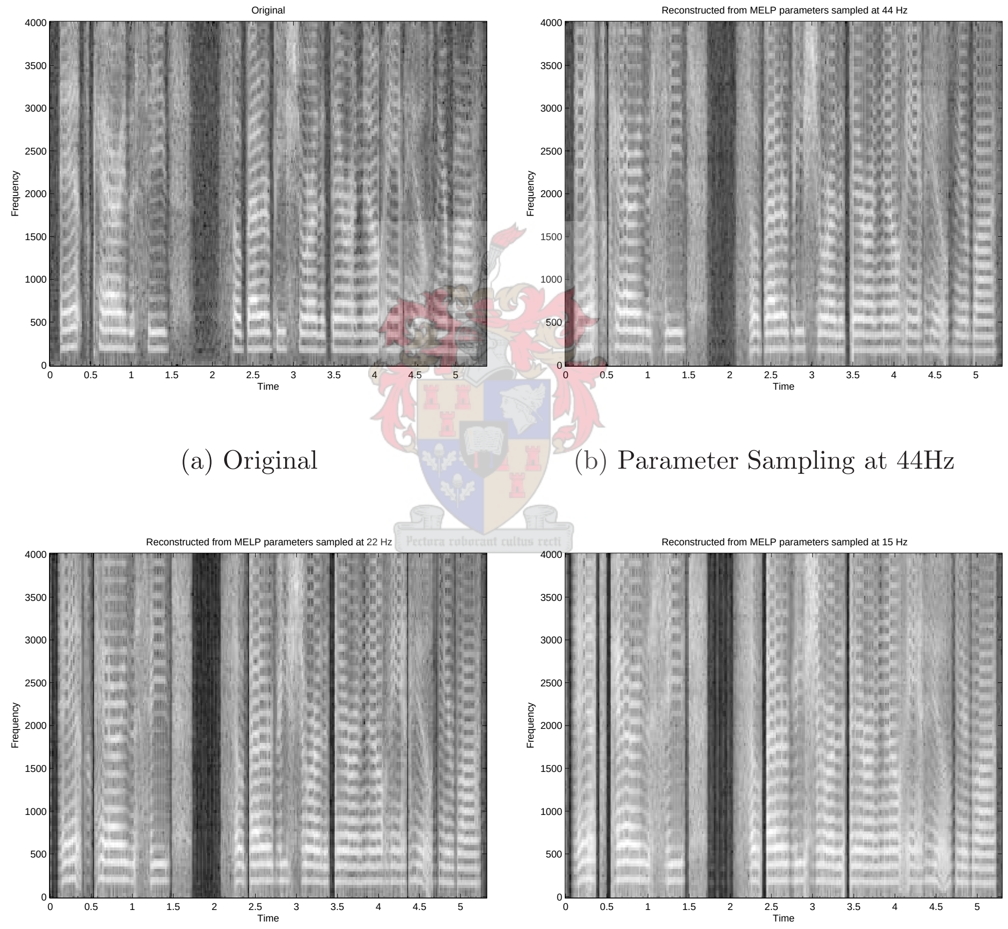
The standard MELP vocoder is thus in fact a special case of the IS-MELP vocoder. In this case the sampling points are simply all chosen to be 22.5ms apart.

However, since the IS-MELP vocoder allows the encoder to choose arbitrary sampling points, an interesting experiment is to investigate the special cases of the IS-MELP where the speech parameter trajectory is regularly sampled at different frame rates. Of course these cases may be divided into two categories:

1. Sampling the MELP parameter trajectory at intervals of less than 22.5ms - referred to hence as over-sampling.
2. Sampling the MELP parameter trajectory at intervals of more than 22.5ms - referred to hence as under-sampling.

In the former case, over-sampling the trajectory by a factor 2 (in other words using sampling intervals of 11.25ms) appears to produce a small but noticeable improvement in speech quality. Over-sampling by a factor of 10 produces speech which appears by to the author's ear to be almost indistinguishable from the original samples. This suggests that the MELP model accurately models the true speech production model and consequently that the majority of the artifacts of the synthesised speech are created by the effects of sampling the parameter trajectory.

In the latter case, under-sampling the parameter trajectory by a factor 2 produces audible degradation of the synthesised speech. Under-sampling by a factor 3 produces substantially more distortion and under-sampling by a factor 4 produces speech which is usually no longer intelligible to the author.



(a) Original

(b) Parameter Sampling at 44Hz

(c) Parameter Sampling at 22Hz

(d) Parameter Sampling at 15Hz

Figure 7.3: *Effect of sampling rate of the MELP parameters.*

Figure 7.3 illustrates the effect of the sampling rate of the MELP parameter vectors on the speech signal. This is best done by means of the spectrograms of the signals. In the case of the parameters sampled at a high rate (44Hz), one can see in the spectrograms how the features of the speech are preserved. In the low sampling rate versions (such as the 15Hz example), the main features of the speech can still be seen to be preserved (such as the main shape of the pitch and the silent regions), but much of the resolution is lost.

From the results shown in figure 7.3 as well as those in figure 6.2 and 6.3 we can therefore conclude that a naive under-sampling of the MELP parameter trajectory will produce a commensurate reduction in bit-rate but will also result in unacceptable losses in the quality of the synthesised speech. This is confirmed by the objective results obtained by the PESQ algorithm in figure 7.4, which illustrate the diminishing returns which are obtained as the frame rate of the regular MELP vocoder is increased. Above a frame rate of roughly 100 frames per second, one does not see any singificant improvement in the quality of the synthesised speech as measured by the PESQ metric. In the region below 50 frames per second, the quality degrades very rapidly with decreasing frame rate. This suggests that the frame rate of 44 frames per second used by the standard MELP vocoder represents a sensible choice.

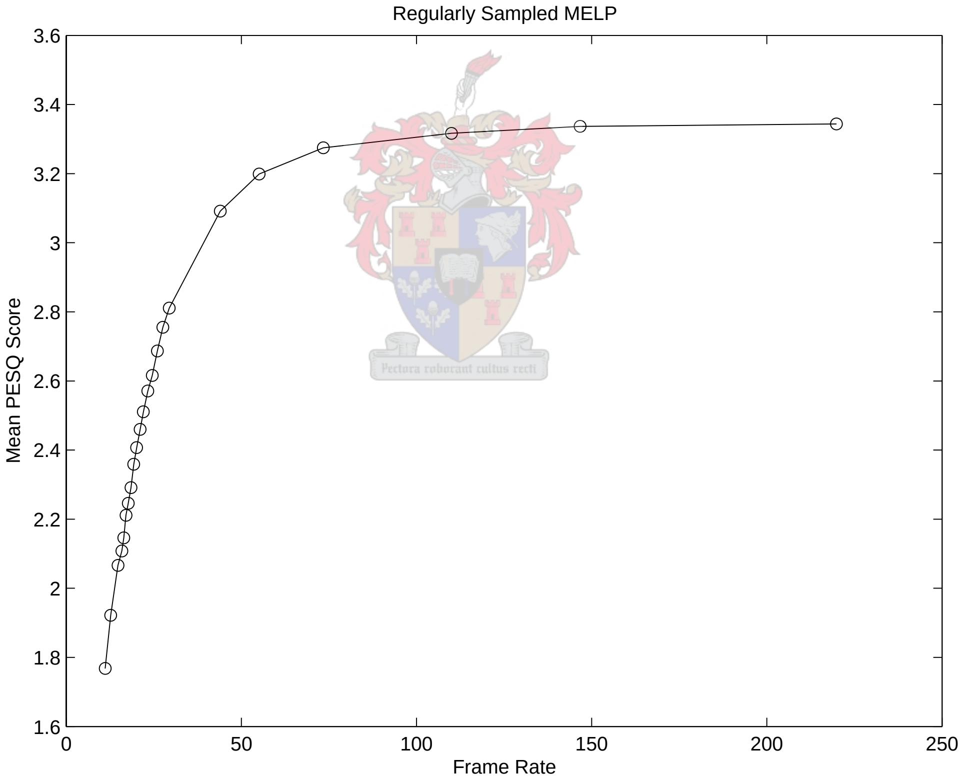


Figure 7.4: *Variation of Bit Rate and Quality by Modification of Regular MELP Sampling Rate*

7.5 Key Frame Determination

We chose for computational reasons to determine the placement of a single frame at a time as opposed to performing simultaneous optimisation of all frame placements. We will therefore optimise k_n based on the previously determined $k_0, k_1, \ldots, k_{n-2}, k_{n-1}$. From an application point of view this approach makes sense since it minimises the coding delay.

We are thus confronted with the problem of selecting the placement of a single key frame at a time. This selection must reflect a compromise the two conflicting requirements of

1. Minimising the bit rate by choosing the key vectors as far apart as possible.
2. Maximising the quality of the synthesised speech by choosing the key frames as close together as possible.

We achieve this by evaluating a cost function which measures the perceptual cost of choosing a given frame as the next key frame. The next frame is chosen as far as possible from the previous frame without exceeding a chosen cost threshold. Clearly the choice of this cost function is fundamental to the success of the algorithm.

Our approach will be to calculate the metric directly from the MELP model parameters. This approach has the advantage of being computationally efficient in comparison with the analysis-by-synthesis alternatives.

7.5.1 Key Frame Determination from Curvature of the Trajectory

In [9], the interpolation error of a one-dimensional function $f(x) \in C^2[x_0, x_1]$ with linear interpolation is given as

$$e_1(x) = \frac{f''(x)\xi(x)}{6}(x - x_0)(x - x_1) \tag{7.3}$$

With $\xi(x) \in [x_0, x_1]$. The interpolation error at any point is therefore linearly proportional to the second derivative of the function at that point. Thus for piecewise linear interpolation it makes sense to choose the interpolation points to be those points which have the largest second derivative. This approach makes sense since these points are those points where the function exhibits maximum curvature and is thus least suited to linear interpolation.

Our first algorithm therefore simply examined the second partial derivatives of the MELP parameter vector trajectory. If any of these second partial derivatives (in other words, the second derivative of any given MELP model parameter) exceeded a threshold, the frame was regarded as significant and transmitted.

Since taking the Euclidean distance between LSFs does not constitute a sensible metric with which to evaluate the magnitude of the derivative of the MELP parameter vector, we used the perceptually based distance measure employed by McCree and Barnwell [55, 69] and mentioned in equation 5.5 in the MELP Vector Quantisation, to estimate a more perceptually applicable form of the derivative.

In figure 7.5 and figure 7.6, we demonstrate the various partial derivatives of the MELP parameter trajectory.

The algorithm was not successful. At low bit rates, synthesised speech was severely distorted and sounded unacceptably unnatural. We believe that the estimation of the MELP model parameters is an inherently noisy process, and calculating the derivative tends to emphasise the noise, leading to highly inaccurate estimation of frame importance.

A second reason for the failure of the algorithm is that the regions of maximum curvature of the feature vector trajectory tend to be clustered together. This means that often many successive frames are chosen for transmission. In order to maintain an acceptable bit-rate, the threshold for transmission must therefore be set to a large value. This large threshold results in some important features of the feature vector trajectory being ignored.

7.5.2 Key Frame Selection by Direct Estimation of Reconstruction Error

A more robust approach would be to calculate directly the reconstruction error produced by the chosen set of key-frames, i.e. estimate the distance between $\mathbf{p}[t]$ and $\tilde{\mathbf{p}}[t]$.

We will attempt to devise a distance metric for any two vectors in this parameter space P which is strongly correlated to the perceptual distance of the associated speech segments in the time domain.

In other words there exists:

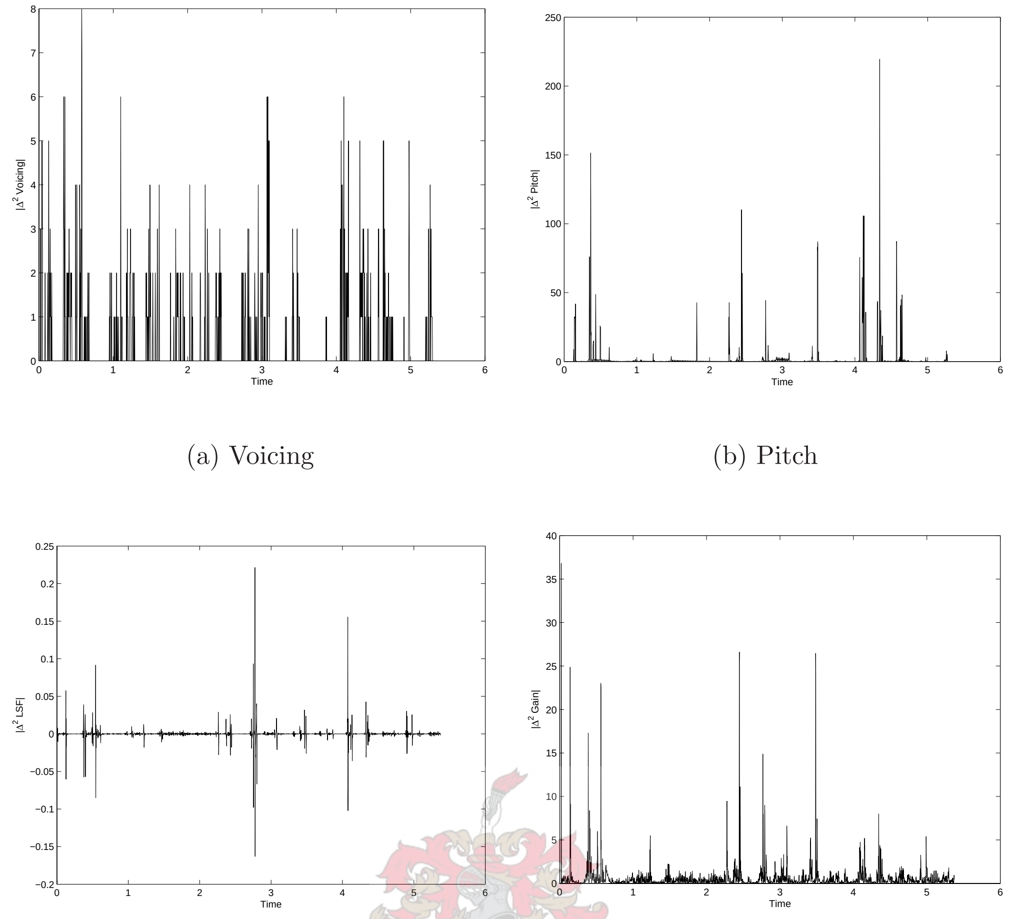
$$d(P \times P) \to \Re \tag{7.4}$$

such that given $\mathbf{p}_1, \mathbf{p}_2 \in P$ and $s_1 = S(\mathbf{p}_1)$, $s_2 = S(\mathbf{p}_2)$

1. If $d(\mathbf{p}_1, \mathbf{p}_2)$ is large, then s_1 and s_2 sound different.
2. If $d(\mathbf{p}_1, \mathbf{p}_2)$ is small, then s_1 and s_2 sound similar.

We may reduce the problem of determining the distance between two MELP parameter trajectories into the following sub-problems.

1. Given two parameter vectors, devise a metric which quantifies how differently the speech segments they will produce will sound, along the lines of the idea illustrated in equation 7.4.



(a) Voicing

(b) Pitch

(c) LSF

(d) Gain

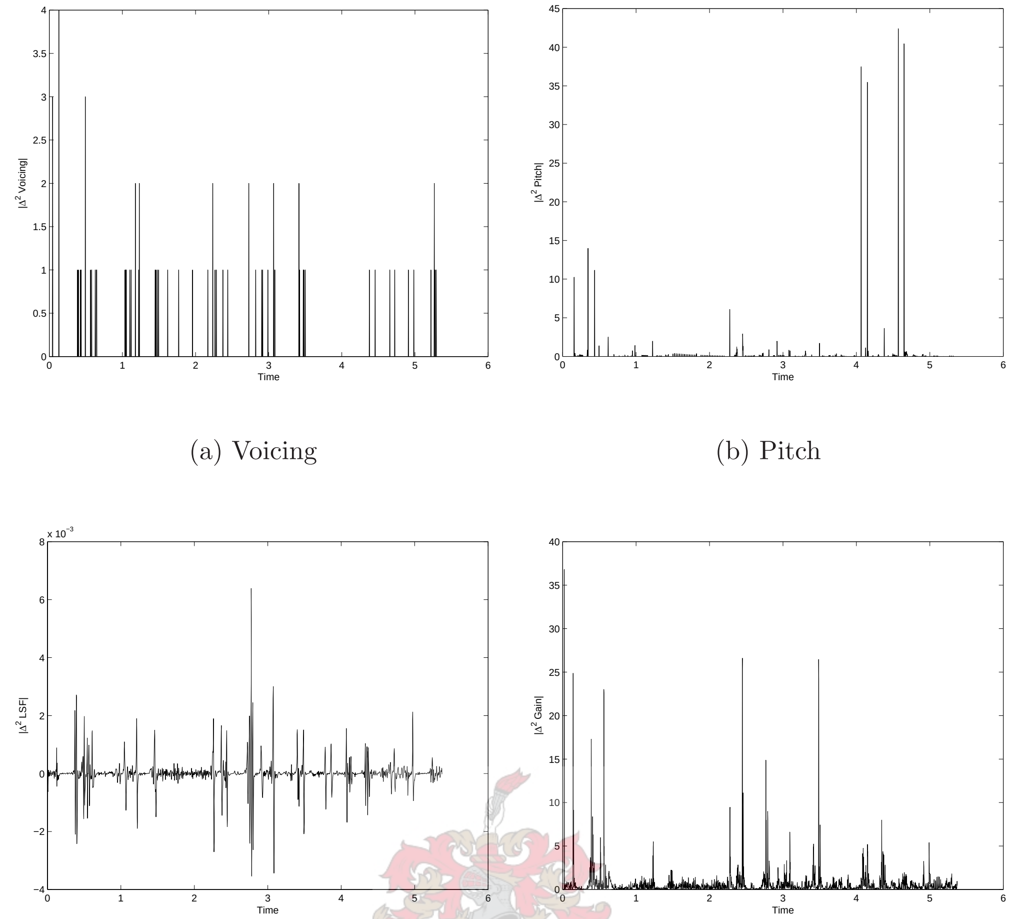
Figure 7.5: *Second partial derivatives of MELP model parameter vector trajectory. No post-processing applied to trajectory.*

2. Given two parameter vector trajectories, devise a metric which quantifies how different the sounds produced by the trajectories will be. The norm described in equation 7.1 was used for this.

As discussed in 3.3, the most suitable measure to discriminate between two speech signals is their *perceptual distance*. As we mentioned, the only true measure of the perceptual distance is the human ear, but some proposed approximations exist. These approximations typically operate in either the time or the frequency domain.

We use knowledge of how the MELP decoder would treat the received data (linear interpolation between successive key frames) in order to choose frames to transmit.

In order to do this we considered the trajectory that would be created by the decoder from interpolation of the candidate set of key frames. We compare the distance between this interpolated trajectory and the original parameter trajectory in order to determine an



(a) Voicing

(b) Pitch

(c) LSF

(d) Gain

Figure 7.6: *Second Partial Derivatives of MELP Model Parameter Vector Trajectory after application of post-processing.*

optimal or near-optimal choice of key frames. This algorithm is described in Algorithm 1. In the algorithm, we use the notation used earlier in the chapter, namely that $\{\mathbf{p}[n], n = 1 \ldots N\}$ is the highly oversampled approximation to the parameter vector function of time, consisting of N samples and $\{\tau_0^*\tau_1^*, \tau_2^*, \ldots, \tau_{M-1}^*\}$ are the set M of optimal frames to transmit. Of course we desire that $M << N$.

As described above, we will simply optimise the placement of a single frame at a time although the optimisation of multiple frames should be similar albeit computationally more demanding. Our simple algorithm examines the distance between the candidate synthesised trajectory (which will be created by the placement of the next key-frame in a particular position) and the original parameter trajectory.

We transmit the furthest frame which does not cause the interpolated trajectory to violate the chosen distortion criterion. The crucial element of the algorithm is the

```
Algorithm 1: MELP Frame Selection based on Interpolation Error
Input: $\mathbf{P}\{\mathbf{p}[n], n = 1 \ldots N\}$
Output: $\{\tau_0^* \tau_1^*, \tau_2^*, \ldots, \tau_{M-1}^*\}$
InterpolationErrorBasedFrameSelection($\mathbf{P}$)
    prev ← 0
    next ← 1
    $\tau_0 \leftarrow 0$
    $k \leftarrow 1$
    while next ≤ N
        if SignificantInterpolationError(($\mathbf{p}[prev] \ldots \mathbf{p}[\text{next}]$))
            $\tau_k \leftarrow$ prev
            $k \leftarrow k + 1$
            prev ← next
            next ← prev+1
        else
            next ← next+1
```

SignificantInterpolationError(($\mathbf{p}[prev] \ldots \mathbf{p}[\text{next}]$))

function, which evaluates the expected interpolation error which will be created by linear interpolation between the previous frame and the candidate frame. We will regard the interpolation error as significant if the original and reconstructed parameter vector trajectories are significantly different. The trajectories are regarded as being significantly different if they are significantly different at any point, thus reducing the problem further to whether the model parameters at any point in the trajectories are significantly different, as in equation 7.1.

Two sets of MELP model parameters are regarded as being significantly different when the distance between them along any dimension exceeds a threshold, i.e. if any of the model parameters are significantly different. This is qualitatively similar to the standard L_∞ norm which is commonly used in approximation theory. More detail about the L_p norms may be found in [64].

It was beneficial to define a different threshold and distance function for each MELP model parameter (for convenience we regard the Linear Predictor as a single model parameter and we regard the set of band-pass voicing strengths as a single model parameter). This was done because a significant change in any of the parameters will result in a perceptually significant difference in the speech waveform.

The metrics used for the individual MELP model parameters are detailed below:

Linear Predictor; The perceptually motivated distance measure used by Barnwell and McCree in [55] and defined in equation 5.5 is used.

Voicing; A weighted Euclidean distance metric was used:

$$d(v_1, v_2) = \sqrt{\sum_{k=0}^{4} w[k](v_1[k] - v_2[k])} \tag{7.5}$$

A number of different weight vectors were tried. The weighting which appeared to work best was to weight only the lowest band voicing (overall voicing) more heavily. It was also beneficial to set the lowest band weight to a very high value, so that a change in the overall voicing would result in frames being classified as ‘very’ different. Thus $w_1 = w_2 = w_3 = w_4 = 1$ and $w_0 = \infty$.

Pitch; We used the maximum of the pitch difference relative to each of the two pitch values. See section 3.3 .

$$d(p_1, p_2) = \max \frac{|p_1 - p_2|}{p_1}, \frac{|p_1 - p_2|}{p_2} \tag{7.6}$$

Gain; We used the absolute value of the logarithm of the ratio of the respective gain values:

$$d(G_1, G_2) = \left| \log \left(\frac{G_1}{G_2} \right) \right| \tag{7.7}$$

Using these metrics, we can compare the interpolated and original parameter trajectories directly in the parameter domain, without having to re-synthesise the speech.

This approach appeared to be quite successful. In comparison to regularly sampled MELP, IS-MELP achieved improved quality at similar bit-rates. Since the bit-rate produced by this algorithm is variable, statistics for the bit-rate were collected over a substantial speech corpus in order to obtain an accurate estimate of the average bit rate produced by a given set of parameters.

Subjective test results for the achieved quality is detailed in chapter 8.

We used the PESQ algorithm described in to estimate speech quality during development.

In figure 7.7 we can examine the points which were selected by the IS-MELP frame selection algorithm, using non-optimised thresholds and compare these with the tracks of the line spectrum frequencies and with the spectrogram of the results.

7.6 Threshold Optimisation

It is interesting to consider the effect of the thresholds for the various MELP model parameters used by the cost function. We can investigate how the average frame rate and the average speech quality of the IS-MELP vocoder varies as a function of a particular threshold by keeping the other thresholds constant and measuring the frame rate and

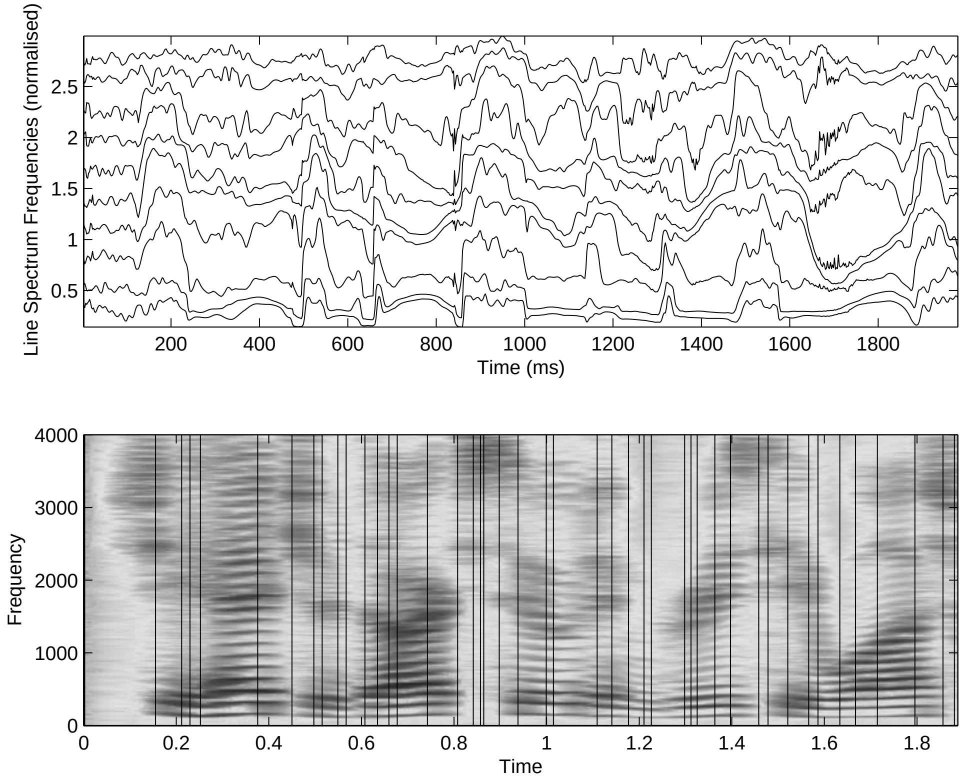


Figure 7.7: *IS-MELP Sampling. The positions of the sampling points are indicated on the spectrogram by the vertical lines.*

quality of the speech as a function of the threshold. In accordance with the results in [18] we would expect to see that the quality of the synthesised speech will increase as a logarithmic function of the bit rate. We present results obtained by variation of the thresholds mentioned in algorithm 1. In each of the following graphs, we plot a parametric curve which illustrates the quality (measured using the PESQ algorithm introduced in section 3.7.3) of the vocoder in comparison to the average bit-rate produced by the vocoder.

Figures 7.8, 7.9, 7.10 and 7.11 show the variation of bit-rate and quality produced by variation of the thresholds for the voicing, gain, LP system and pitch respectively. In each experiment, three of the parameters were kept constant and the fourth was varied.

In each of the figures, it can be seen that there are regions in which the IS-MELP vocoder outperforms the regularly sampled MELP vocoder and regions where the IS-MELP vocoder performs less well than the regular MELP vocoder. This implies that the variation of a single threshold will produce a local maximum in the improvement of the IS-MELP performance over the performance of regular MELP.

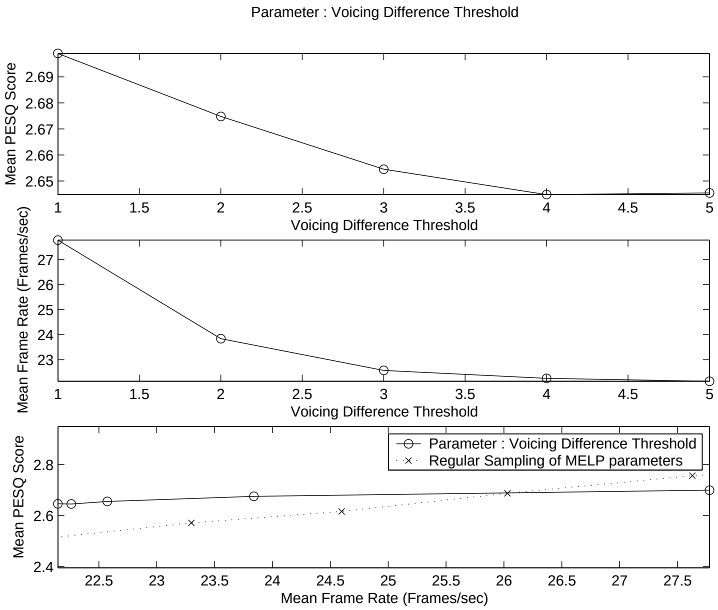


Figure 7.8: *Variation of frame rate and quality by modification of the voicing threshold. The top and middle plots indicate the quality and frame rate of the vocoder respectively as functions of the threshold used. The lower graph plots the quality against the frame rate for voicing, gain, LP and pitch thresholds respectively. For comparative purposes, the quality of regularly sampled MELP at various frame rates is also shown.*

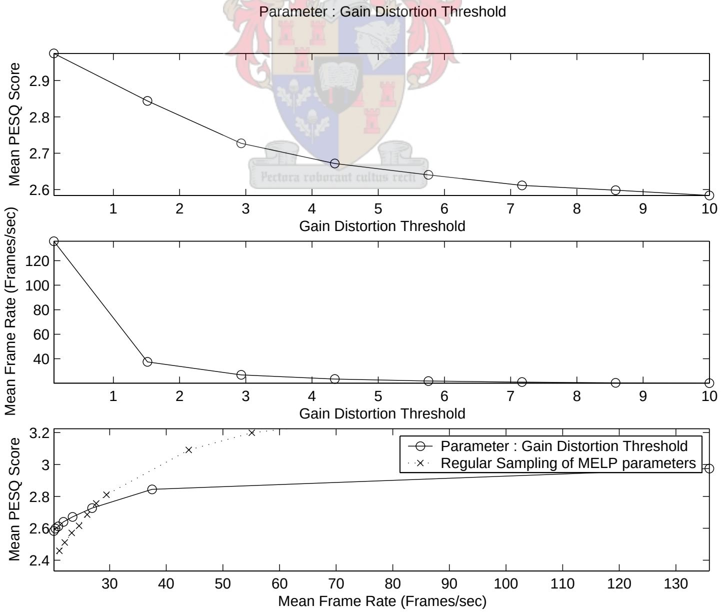


Figure 7.9: *Variation of Bit Rate and Quality by Modification of the Gain Threshold*

7.7 One-Dimensional Optimisation of Thresholds

Due to the high dimensionality of the threshold space and the complexity of calculating a reasonable estimate of the quality and frame rate of the IS-MELP algorithm, an ex-

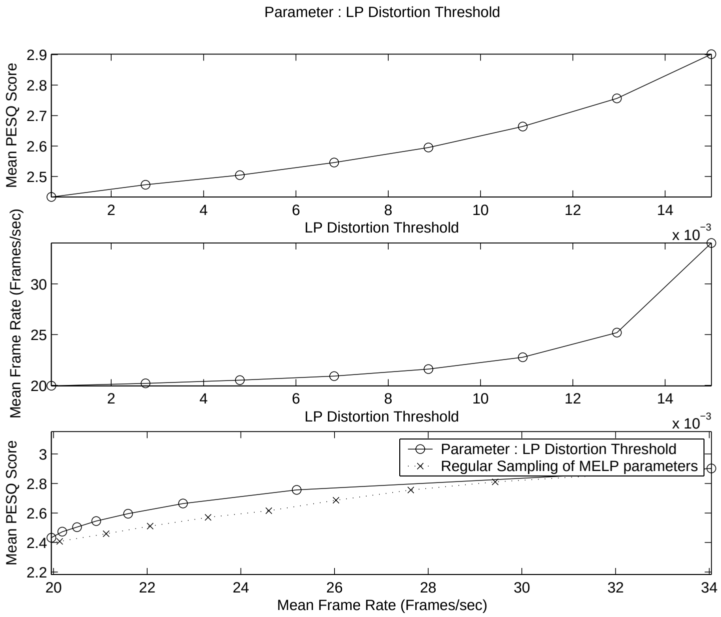


Figure 7.10: *Variation of Bit Rate and Quality by Modification of the LSF Threshold*

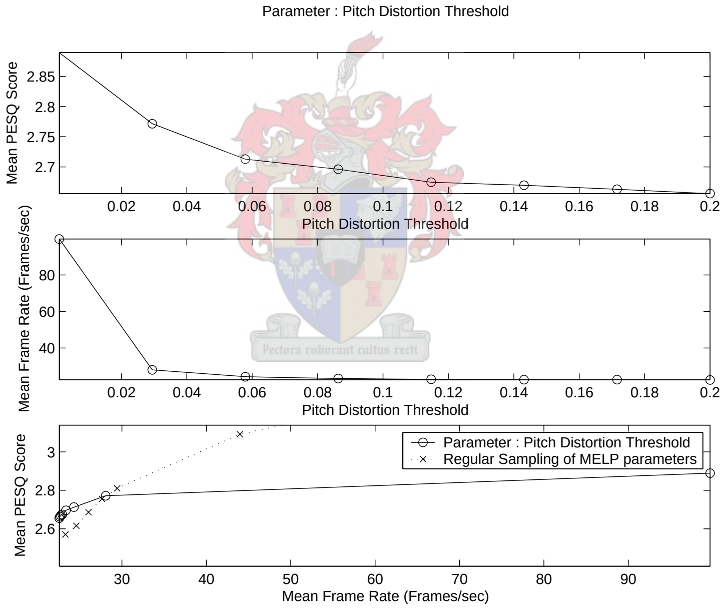


Figure 7.11: *Variation of Bit Rate and Quality by Modification of the Pitch Threshold*

haustive optimisation of the threshold was impractical, and a series of one-dimensional optimisations were used instead.

The approach taken was to perform optimisation of a single thresholds to determine the value of that threshold which maximises the improvement of IS-MELP over regularly sampled MELP. By iterating this optimisation over each of the thresholds in turn,we found improved thresholds for the IS-MELP algorithm

7.8 Effect of Post Processing on Rate and Quality

Using the same technique used to investigate the thresholds in sections 7.6 and 7.7, we can investigate the influence of the post-processing on the quality and bit rate of the voice coding system.

7.8.1 Low-Pass filtering of the LSF trajectories

When filtering the LSF and thus decreasing the rate at which the LP system evolves, the results are illustrated in figure 7.12. As the post processing by the low-pass filter causes a decrease in the bandwidth of the LSFs, we see a corresponding decrease in the bit-rate and in the quality of the transcoded speech.

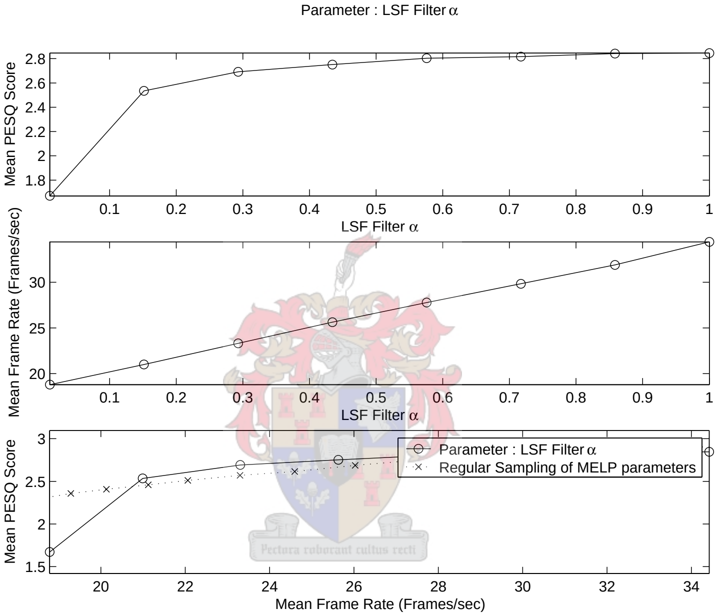


Figure 7.12: *Variation of Bit Rate and Quality by Modification of the LSF Post processing*

In figure 7.12, we note that the bit-rate against quality curve exhibits a sharp change in gradient around $\alpha = 0.2$. This suggests that if the bandwidth of the LSF filter is decreased to less than 15Hz, there is a significant loss of information in the LSF trajectories. This is fairly consistent with the average phoneme rate we discussed in section 2.1.

7.8.2 Filtering of the Pitch and Voicing

When filtering the voicing and pitch, qualitatively different results are observed from those we observed by filtering of the LP system. As the order of the LULU filter is increased and thus the smoothness of the voicing is increased, we observe that the quality measured

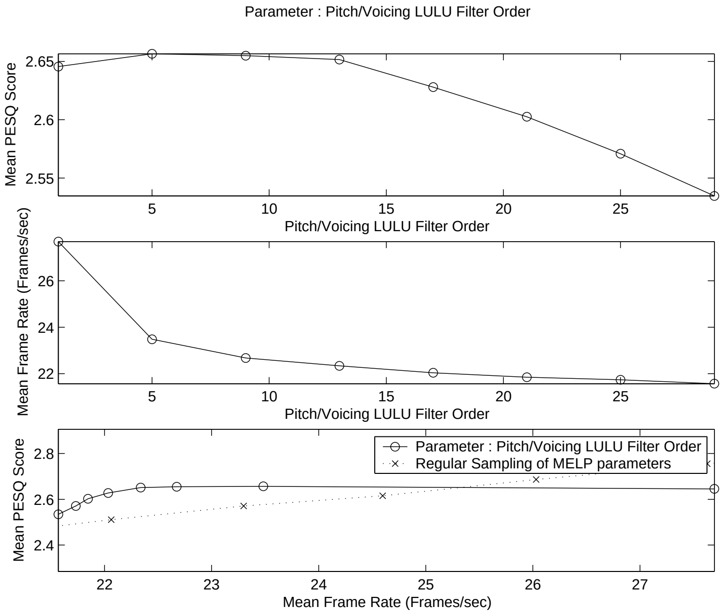


Figure 7.13: *Variation of Bit Rate and Quality by Modification of the Voicing Post Processing*

by PESQ first increases slightly and then decreases with increasing smoothness, as seen in figure 7.13. The marginal increase in perceived quality of the speech is most probably explained by the fact that the post processing removes errors introduced in the pitch and voicing analysis. However, increasing the post processing causes a loss of phonetic content in the synthesised speech and a corresponding loss of quality.

7.9 Conclusions

The bit-rate versus quality curves illustrated in the previous sections indicate that it is possible to achieve continuous variation of the bit rate and quality of the voice coding system by varying the allowable distortion. Furthermore, this decision may be continuously adjusted at the transmitter without introducing the necessity of transmitting additional information to maintain synchronisation with the receiver.

The most significant disadvantage of the IS-MELP vocoder is the difficulty of relating the distortion thresholds to a fixed bit-rate. Since there is no simple mathematical function which determines the bit-rate from a set of thresholds, the bit rate produced by a threshold set must be evaluated empirically. However, in an application environment, this problem could be circumvented in one of two ways:

1. By adaptively altering the thresholds in order to produce the desired bit-rate.
2. By storing optimised threshold sets for various bit-rates and loading an appropriate

threshold set for the desired bit-rate.

Figures 7.8, 7.9, 7.10 and 7.11 clearly demonstrate that in certain regions, particularly in the case of lower frame rates, IS-MELP achieves better PESQ scores than regularly sampled MELP.

Filtering the pitch and voicing with a non-linear filtering scheme appears to significantly improve the performance of the IS-MELP vocoder. Filtering of the LP system produces a less significant but still noticeable improvement in the performance of the IS-MELP vocoder.

In the following chapter, we will verify the performance of the IS-MELP vocoder by means of subjective tests. We will use the optimised threshold generated as described in section 7.6.

Chapter 8

Evaluation of Vocoders

As has been discussed in previous chapters (3.7), there are two standard approaches to the evaluation of the quality of a vocoder, namely algorithmic quality measures and human listening tests.

By comparing these approaches, we aim in this chapter to:

1. Compare the performance of the MELP vocoder for a European and an African language, namely English and Xhosa.
2. Compare the relative performance of the reference and IS-MELP vocoders.
3. Determine whether the IS-MELP promises any improvement in voice quality for African languages.
4. To investigate the correlation between the objective and subjective metrics for vocoder performance.

8.1 Speech Corpus

Fundamental to the effectiveness of the subjective test is the corpus of speech samples used for the test. The speech corpus must be representative of the range of inputs which the vocoder is to encode. This means that it needs to represent a large range of different speakers and phonemes.

The speech corpus used for our test was selected from the AST speech corpus [59]. This speech corpus contains a large set of speech samples for a number of South African languages and accents. Only the English and Xhosa samples from the database were used for the subjective tests. [1]

[1] From the English corpus the phoneme rich sentence common to all speakers, “Helen’s stint as league manager provided useful opportunities but the elementary practical tasks of going to meetings and reading a work sheet bored her.” was used for the test.

From the Xhosa corpus the phoneme rich sentence common to all speakers, “Intetho kamongameli

In the AST speech corpus, each speaker reads two phonetically rich sentences. One of these sentences is individual to the speaker, the other is common to all speakers. We used the latter for our tests.

8.1.1 Recording Artifacts

Since the AST speech corpus was recorded over the telephone, many of the samples are of poor quality. Samples are degraded by recording and transmission artifacts, (such as hissiness, buzziness, static and clipping) as well as side noises (other speakers, background traffic noise and other background noise). Additionally, many of the speakers stutter or mispronounce words. We selected only sentences from speakers whose recordings were read fluently, appeared to be largely free of transmission and recording artifacts and which contained as little side-noise as possible.

Five male and five female speakers were selected for each language.

8.1.2 Utterances

Four speech samples were selected from the phonetically rich sentence spoken by each speaker, thus bringing the total number of samples per language to 40. For every language, the same samples were selected for every speaker. Samples were chosen to be short but grammatically meaningful utterances.

English Utterances

In the English, the four samples used were:

1. Helen's stint as league manager
2. provided useful opportunities
3. but the elementary practical tasks
4. going to meetings and reading a work sheet bored her

Xhosa Utterances

In the Xhosa, the four samples (*and their translations*) were:

idale unxunguphalo kuba ayichaphazelanga le miba ilandelayo izicwangciso ezijongene nenkqubo yemfundo yezikolo needyunivesithi; iimfazwe; iziqhushumbisi; ukusetyenziswa kwezixhobo zokulwa ngengqiqo ukwakhiwa koohola beendlela neebhrorho; ukunqongophala kwamisebenzi; uxanduva lwabantu abamhokamhokana nesifo ugawulayo iintsana ezipenapeneka ezibhedlele inkcitho eburhulumenteni ukunqena kwanokunyoluka kwamagosa karhulumente amajelo onxibelelwano uphuhliso lweenkonzo kwanokutshutshiswa kwabantwana." was used for the test.

1. intetho kamongameli idale unxunguphalo
   (*the president's speech caused tension*)
2. kuba ayichaphazelanga le miba ilandelayo
   (*because it did not include the following issues*)
3. izicwangciso ezijongene nenkqubo yemfundo yezikolo needyunivesithi
   (*plans to look at the education programmes of schools and universities*)
4. ukusetyenziswa kwezixhobo zokulwa ngengqiqo
   (*the justifiable use of weapons*)

8.2 Objective Tests

In our objective tests we used the PESQ algorithm which has already been mentioned briefly in section 3.7.3.

8.2.1 PESQ

The PESQ algorithm refers to the objective metric described in [43]. This metric is recommended by the Telecommunication Standardisation Sector of the International Telecommunication Union as an objective method for the end-to-end speech quality assessment of narrow band speech codecs. We have provided full details of the algorithm in Appendix G. The PESQ metric provides an estimate of the mean opinion score (MOS) which would be assigned to a speech file. Thus a higher PESQ score corresponds to better speech quality.

8.2.2 Rate-Distortion Curves

We investigated the rate-distortion curves for both the IS-MELP and regular sampling MELP vocoders. This was done by performing transcoding on the speech corpus using the vocoders at various rates and measuring the quality of the synthesised speech with the PESQ algorithm. The results are summarised in figure 8.1.

In figure 8.1 we display the PESQ score attained at each bit rate for both the IS-MELP and regularly sampled MELP algorithms. The figure clearly displays how the IS-MELP achieves better quality at low frame rates, whereas the regular MELP algorithm achieves better quality at higher frame rates. The two algorithms appear to achieve equivalent performance at approximately 50 frames per second, which is very close to the frame rate used in the standardised MELP encoding described in chapter 5.

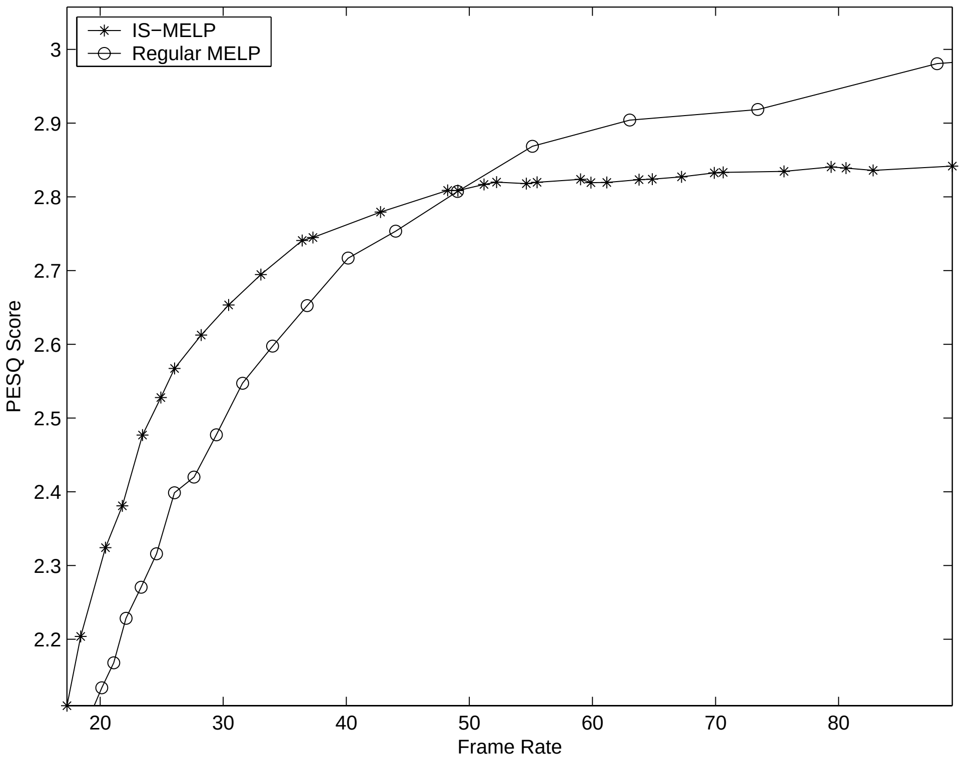


Figure 8.1: *Overall (Combined English and Xhosa) Rate-Distortion Curve for Regular and IS-MELP*

8.2.3 Language Bias in Rate-Distortion Curves

We investigated the separate English and Xhosa rate-distortion curves for both the IS-MELP and regular sampling MELP vocoders. The results are summarised in figures 8.2 and 8.3.

Figure 8.2 compares the distortion measured for English and Xhosa after transcoding with regularly sampled MELP at various frame rates. We see that there is a substantial difference between the performance of the vocoder for English and Xhosa. In particular, according to the PESQ metric, the transcoding quality of the voice is better for English than for Xhosa at almost all frame rates.

Figure 8.3 compares the distortion measured for English and Xhosa after transcoding with IS-MELP at various frame rates. We see that there is very little difference between the performance of the IS-MELP vocoder for English and Xhosa.

8.2.4 Discussion of Objective Test Results

It was interesting to note that the PESQ rate-distortion curves for the English and Xhosa speech were substantially different in the case of regular MELP. However, in the case of

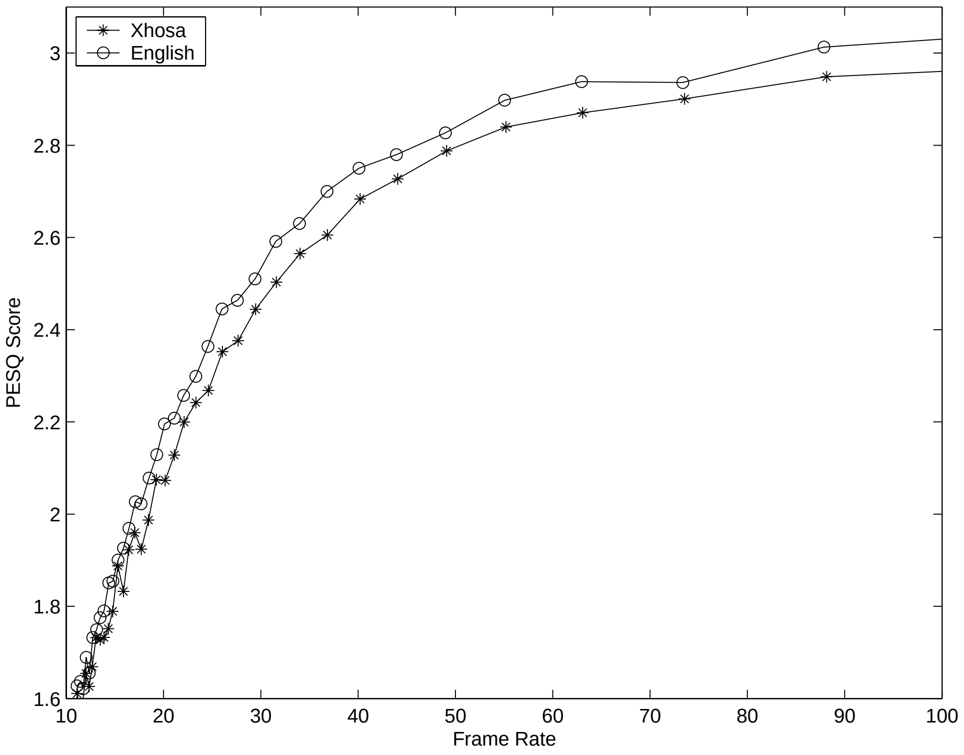


Figure 8.2: *Language Dependence of MELP Rate-Distortion Trade-off using regular sampling*

IS-MELP, the PESQ rate-distortion curves for the English and Xhosa speech were almost exactly equal.

At low bit rates the IS-MELP vocoder produces substantially better speech quality than the regular MELP vocoder according to the PESQ metric. At higher bit-rates the IS-MELP vocoder is less efficient than the regular MELP vocoder.

It was also found that the IS-MELP vocoder exhibited less differentiation in the way in which it handled English and Xhosa. We suspect that this difference is due to the fact that the multi-rate capability of IS-MELP is better able to handle certain phoneme classes. As shown in appendix D.1, there are different distributions of phoneme occurrences in the two languages used in the test. See table D.1.

We hypothesise that the difference is due to a greater variance in the phonetic modulation rate in Xhosa.

8.3 Subjective Tests

As we discussed in section 3.7.2, the ultimate authority on the quality of transcoded speech is a human listener and it is therefore necessary to perform subjective tests in

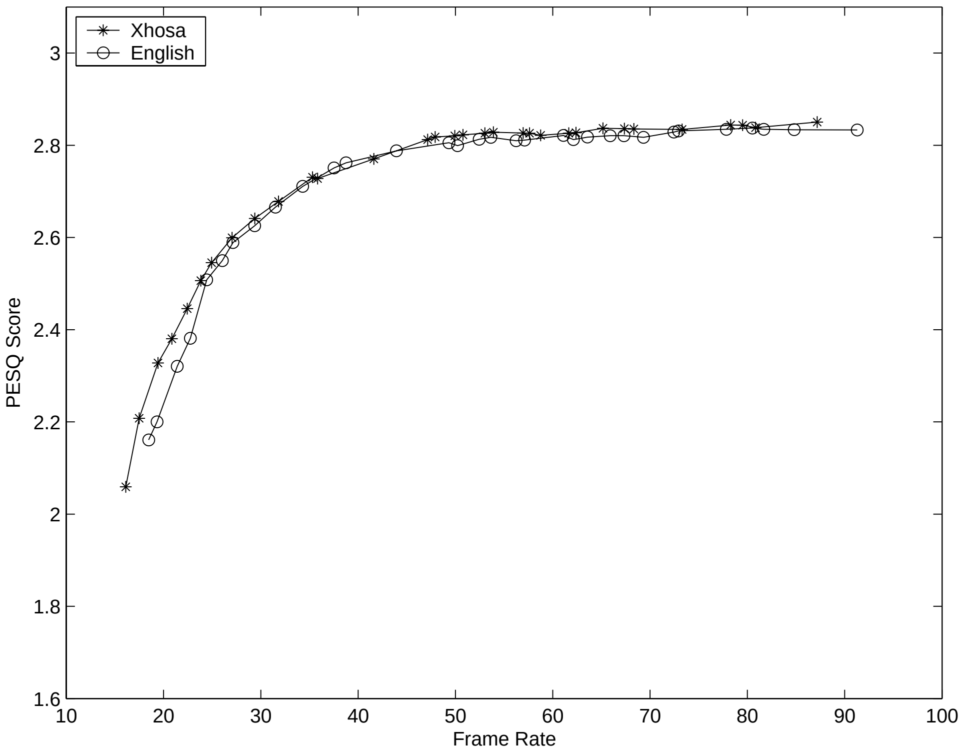


Figure 8.3: *Language Dependence MELP Rate-Distortion Trade-off with irregular sampling using the IS-MELP algorithm*

order to verify the results predicted using the objective PESQ metric.

The ITU-T has published a recommendation describing methods and procedures for conducting subjective evaluations of the quality of transmitted speech [42]. We have followed these recommendations as closely as possible.

8.3.1 Test Conditions

Each of the selected speech samples was transcoded in four ways:

1. Using the standard MELP vocoder at 60 frames per second.
2. Using the IS-MELP vocoder at 60 frames per second.
3. Using the regularly sampled MELP vocoder at 22 frames per second.
4. Using the IS-MELP vocoder at 22 frames per second.

Referring to the results in figure 8.1, we note that at the higher rate, we would expect that regular MELP would perform better and at the lower rate we would expect that IS-MELP would perform better.

The transcoding involved only the parameterisation and frame selection before resynthesis. It was felt that quantisation effects would interfere with the effect that was being studied (that of regular and irregular sampling of the speech feature vector trajectory).

The synthesised speech was saved in the form of a standard PCM wave (*.wav*) file for use in the subjective tests.

8.3.2 Subjective Test Overview

We used the Praat software suite [8] to perform the tests. The tests were scripted in order to provide as little variation as possible in the test conditions experienced by different subjects.

The speech corpus for each language consisted of 40 samples as described in section 8.1. Each of the samples was transcoded in 4 different ways as described in section 8.3.1. Thus a total of 160 samples per language were created. Of the 40 samples per condition we randomly selected 20, to give a total of 80 samples for the entire test. This represented the largest number of total samples we could use in the test without making the duration of the test excessively long.

In accordance with the recommendation of Spanias [79], listeners are first familiarised with the listening conditions and range of voice quality they will encounter in the test. This is accomplished by presenting listeners with 20 samples which are not assigned a score but are simply used for calibration purposes. During this phase of the tests, the listeners are also given the opportunity to set the volume of the samples to an acceptable level and correct any problems in the test setup.

Each subject was then presented with a further 80 samples in random order. After each sample, the subject was required to assign a quality to the sample. The range of quality options provided to the listeners was identical to those described in table 3.3. This is also in accordance with the recommendations made by [42] and those used by [18].

We used 20 samples per condition, which is a relatively large number by the standards of other published subjective tests, for example those mentioned in Daumer [18] used a total of seven different samples per condition and Cuperman [26] used sixteen samples per condition.

We were able to recruit 31 English listeners and 18 Xhosa listeners for the test. This compares well to Daumer, who used 42 listeners and Cuperman who used 12.

The scoring of the samples was done according to the Absolute Category Rating. This is also commonly referred to as the Mean Opinion Score (MOS) described in section 3.7. The details of the test are based on the work of Daumer [18] and Noll [44].

	IS-MELP	Regular MELP
22 frames/sec	2.14	2.52
60 frames/sec	2.90	3.83

Table 8.1: *Final Mean Opinion Scores for Subjective Tests*

	English	Xhosa
IS-MELP 22 fps	1.95	2.51
Regular MELP 22 fps	2.27	3.04
IS-MELP 60 fps	2.60	3.51
Regular MELP 60 fps	3.70	4.10

Table 8.2: *Language dependence of IS-MELP and regular MELP, showing mean opinion scores are indicated for each condition and language.*

8.3.3 Results of Subjective Tests

Table 8.1 shows the final results of the subjective testing. In contrast to the objective test results obtained in the previous section, the regular MELP out-performed the IS-MELP at both the frame rates tested.

As we showed in section 8.2, the IS-MELP vocoder exhibited substantially different performance in Xhosa when compared to the regularly samples MELP vocoder using the PESQ objective metric. Table 8.2 compares the subjective performance of the two vocoders for Xhosa and English. Contrary to our expectations, we obtained substantially better results for the Xhosa utterances across all encoding conditions.

8.3.4 Discussion of Subjective Test Results

Overall results of the subjective tests were disappointing. The improvement at low frame rates predicted by the PESQ objective metric was not achieved. The IS-MELP algorithm introduced artifacts which were not regarded as significant by the PESQ metric, yet which appeared to be disturbing to listeners.

However, at the lower frame rate, the IS-MELP vocoder exhibited substantially less performance difference between languages than the regularly sampled MELP. At 22 fps, the difference in MOS between English and Xhosa was 0.56 for IS-MELP and 0.77 for regularly sampled MELP.

The degradation exhibited by lowering the frame rate was lower for IS-MELP than for regular MELP. In this case, the performance degradation between IS-MELP at 22 and 60 fps was a MOS difference 0.76 while for regular MELP, the performance degradation between 60 and 22 fps was a MOS difference of 1.31. Thus the amount of degradation incurred by lowering the frame rate was halved by using the IS-MELP vocoder.

8.4 Discussion of Disparity between Subjective and Objective Tests

Clearly there is a quantitative difference between the results obtained in the subjective and objective tests. Thus the IS-MELP algorithm must introduce distortion which, though not considered significant by the PESQ algorithm, nevertheless is perceptually disturbing.

To better understand this we compare two of the samples transcoded by IS-MELP at 22 fps which were used in the perceptual test. The samples chosen were those which received the worst mean opinion scores. We will compare the spectrograms of the original and transcoded versions of the two files.

Figure 8.4 shows the spectrogram of the sample which obtained the worst MOS in the perceptual tests. When compared to figure 8.5, we can clearly see the manifestation of formants in regions where there are none in the original sample. These formants can be seen between approximately 0.5 and 0.75 sec and again between 1 and 1.5 sec. The appearance of these spurious formants appears to be due to errors made by the frame selection algorithm.

The spurious formants seem to appear in regions of silence, or near silence, where they interpolate between the true formants in adjacent phonemes. This interpolation lends an unpleasant slurring sound to the speech. However, they do not appear to affect the PESQ score of the sample much, as the transcoded sample obtains a PESQ score of 2.24 which is relatively high.

Figure 8.6 shows the spectrogram of another sample which obtained a very poor MOS in the perceptual tests. When compared to the spectrogram of the original in figure 8.7, we notice the presence of very strong pitch marks in many areas of the synthesised speech. These strong pitch marks lend the synthesised speech a harsh musical quality with strong tonal noises. In this case, the speech segment received a low PESQ score (1.67), which correlates well with the MOS for the segment.

It is, however, worth noting that whereas the two synthesised segments received almost identical mean opinion scores (1.16 and 1.19 respectively), their PESQ scores were substantially different. This casts some doubt on the accuracy of the PESQ algorithm and perhaps explains some of the disparity between the subjective and objective results.

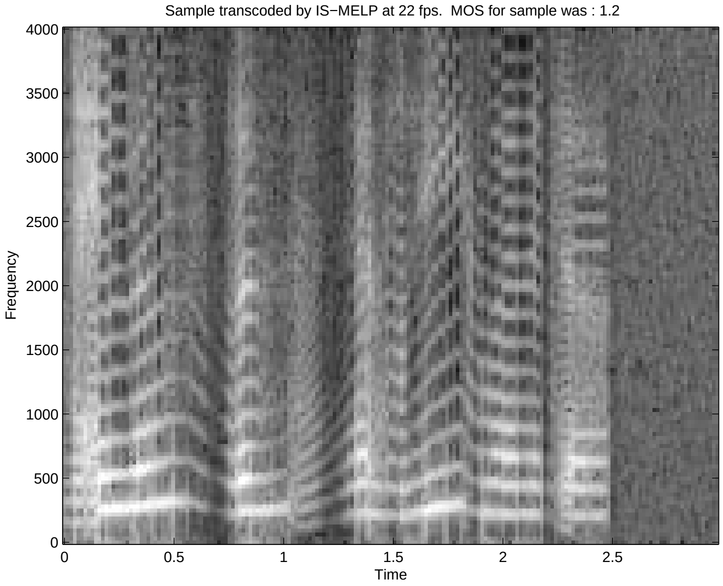


Figure 8.4: *Spectrogram of sample which obtained poor MOS rating but a good PESQ score. Sample was transcoded with IS-MELP algorithm at 22fps.*

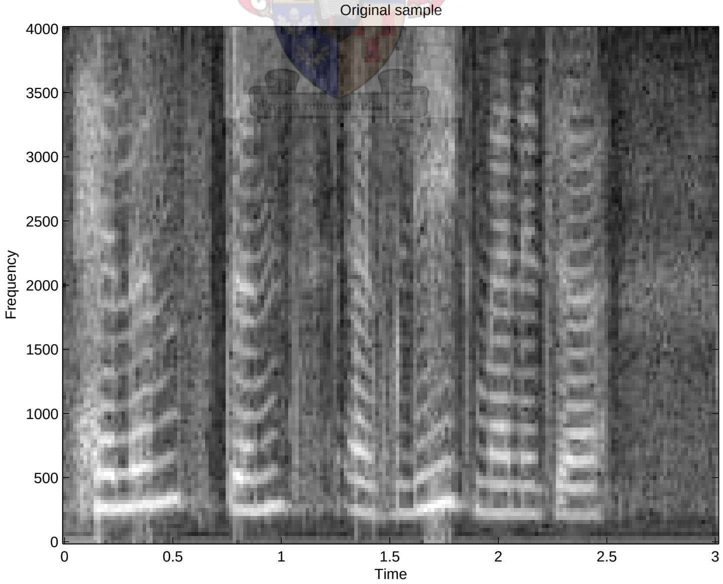


Figure 8.5: *Spectrogram of original speech segment used to generate sample in figure 8.4.*

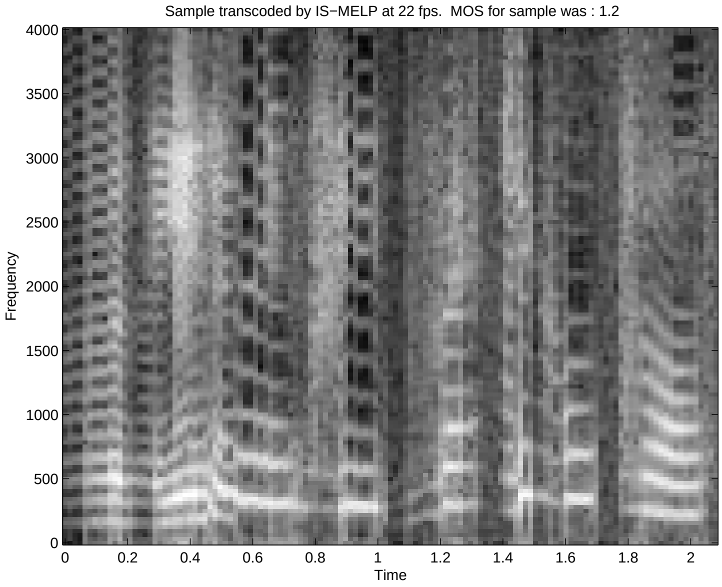


Figure 8.6: *Spectrogram of sample which obtained poor MOS rating and a poor PESQ score. Sample was transcoded with IS-MELP algorithm at 22fps.*

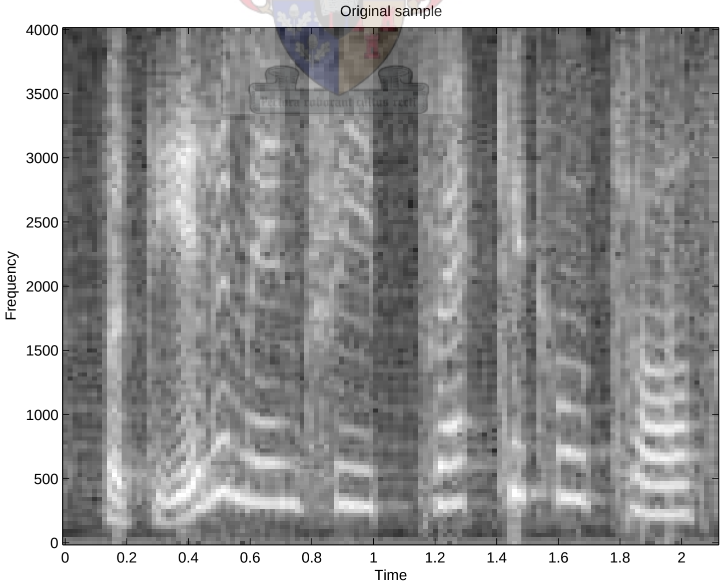


Figure 8.7: *Spectrogram of original speech segment used to generate sample in figure 8.6.*

8.5 Conclusion

While the IS-MELP algorithm has produced results comparable to those of the regular MELP algorithm, and in some cases demonstrated superior performance, the performance, particularly at low frames rates, was found to be unsatisfactory. This was most apparent from the subjective tests. We feel that substantial improvement of the IS-MELP algorithm may still be achieved. Methods of improving the performance of the IS-MELP algorihm are described in section 9.2.

Chapter 9

Summary and Conclusion

9.1 Summary of Results

We undertook a study of the current state of low bit-rate voice coding technology. This encompassed a study of the underlying principles and practices used in low bit-rate vocoders as well as a review of the most important standard algorithms used in low bit-rate voice coding.

The MELP algorithm was selected for more thorough study and implementation. We implemented the standardised MELP algorithm in MATLAB. The MELP algorithm is a complex and intricate software module and represents a useful baseline system for any further work to be undertaken on low-rate speech coding. The implementation of the MELP algorithm in itself represents a substantial piece of work. Having successfully implemented a functional 2400bps MELP prototype we quantitively investigated the performance of the MELP voice coding algorithm.

We investigated the mathematical properties of the voice coding system and attempted to analyse the effect of sampling on the modelisation of the speech signal which is performed by the MELP vocoder. This was done by using the idea of time and parameter domain representations of the speech waveform. This led to the concept of the parameter vector trajectory.

This latter concept was used to develop a novel approach to voice coding, based on the speech production model of the MELP vocoder. The fundamental difference of this encoding model is that model parameters are transmitted irregularly, so that more frames are transmitted in regions where the speech waveform is evolving quickly and fewer frames are transmitted in regions where the speech waveform is evolving slowly. An algorithm to calculate the position of these frames was designed and implemented.

The final portion of the work entailed the evaluation of the standard MELP and novel MELP-based vocoder in a multi-lingual environment. It was hoped that the enhancements made to the improved MELP vocoder would improve the suitability of the vocoder for encoding the short transient phonemes (clicks) characteristic of certain African languages. Both subjective and objective evaluations were performed in order to compare vocoders. Using the PESQ metric for objective evaluation of the vocoders indicated that regular MELP performed substantially more poorly in encoding of Xhosa than in English at all frame rates. Furthermore, the IS-MELP vocoder did not exhibit different performance for English and Xhosa.

The PESQ metric also indicated that at low bit rates, the IS-MELP vocoder exhibited improved performance over the regular MELP vocoder in both English and Xhosa.

Subjective tests did not confirm the improved performance of the IS-MELP algorithm at low frame rates. However as frame rates decreased, the IS-MELP algorithm resulted in more graceful degradation of the transcoded speech than the regular MELP algorithm.

9.2 Recommendations for Future Work

9.2.1 Choice of Metric

We have used a simple perceptually motivated metric to estimate the perceptual distance between the speech which would be synthesised by two different parameter sets. While our metric appeared to be effective and we performed optimisation on it, there is no indication that our metric is an optimal one. Furthermore we did not investigate the underlying form of the metric and it may be possible to obtain improvement by using a metric of a completely different form.

9.2.2 Metric Optimisation

We used iterative one-dimensional optimisation in order to investigate improved threshold sets for the above metric. While some improvement was obtained, this optimisation technique is simplistic and almost certainly leads to non-optimal sets of thresholds. More sophisticated optimisation techniqes are well known. The use of such optimisation algorithms, while more computationally demanding, may well lead to improved results.

9.2.3 Irregular Sampling

The results of the objective tests performed using the PESQ algorithm indicate strongly that there is a potential for substantial improvement in the performance of parametric vocoders to be gained by the use of irregular sampling. Optimisation using the PESQ metric appears to not be a sufficient condition in order to guarantee improvement in subjective testing and other objective error metrics may be investigated to determine whether improvements in the subjective tests can be found.

9.2.4 Parameter Estimation

Currently, the quality of the parameter vector trajectory is not as good as might be desired, due to the inaccuracies involved in the estimation of the MELP model parameters. This inaccuracy interferes with the functioning of the key-frame determination algorithm and ultimately leads to degradation of the quality of the IS-MELP synthesised speech.

We have applied some heuristic post-processing algorithms, which improve the quality of the parameter vector trajectory to some degree but this definitely represents a sub-optimal solution to the problem.

The estimation of the parameter vector trajectory could be improved by using more sophisticated analysis techniques to evaluate parameters such as pitch [19], voicing and linear predictor coefficients [25].

9.2.5 Key Frame Determination

The selection of the key frames to transmit could also be improved. We have used a simple local selection algorithm, in order to satisfy constraints on the computational complexity of the algorithm. However, the algorithm results in non-optimal key frame selection. This would definitely affect the performance of the IS-MELP algorithm.

The key-frame selection algorithm is the most crucial aspect of the IS-MELP vocoder. Thus the most significant improvements may be achieved by improvement of the key-frame selection. As we mentioned briefly before, the key-frame selection algorithm is an optimisation problem and may be treated as such. There are a number of optimisation algorithms which have been successfully used in similar problems at the cost of a significant increase in computational complexity and coding delay. For example, an optimal set of key-frames could be generated using a dynamic programming algorithm [60] such as the one used by Schwardt [74] for pitch tracking.

9.2.6 Statistical Evaluation of Key Frame Transitions

We have not investigated the statistics of the transitions between various key-frames. It is possible that the statistics of the key frames distributions and transitions may lead to clearer understanding of the parameter vector trajectory which in turn will lead to more efficient encoding algorithms.

9.2.7 Shortcomings of the PESQ Metric

As has been demonstrated by the discrepancy between the results obtained in our subjective and objective comparison of IS-MELP and regular MELP, the MOS predicted by the PESQ metric does not always correlate well with the true MOS score obtained by a vocoder in subjective listener tests. We have attempted to analyse this discrepancy particularly with regard to the results obtained by the IS-MELP algorithm, but a full treatment of this topic is beyond the scope of this thesis. However, in the interest of developing more accurate objective metrics (the value of which have already been discussed), a thorough investigation of these results would be valuable.

9.3 Overall Conclusion

The speech waveform can be approximated to a large degree of accuracy by an equivalent representation as the variation of a set of parameters of a speech synthesis model. This representation may be used to achieve very efficient compression of the speech waveform while still maintaining intelligibility and a large degree of naturalness.

Parametric speech coding involves a two step process whereby the parameters are first estimated and then encoded. The typical method of encoding involves regularly sampling of the parameters by analysis of frames of the speech signal.

Usually, these frames are matched in length to the characteristic time-scale over which the parameters of the synthesis model vary. This characteristic time-scale is not a universal constant, but varies with time and speaker.

One may achieve more efficient encoding of the speech signal by exploiting this variation, as shown by the results of the objective tests. However, this encoding may produce artifacts which negatively affected the perceptual quality of the synthesised speech.

A naive approach to the selection of sampling points of the synthesis model parameters does not produce the desired perceptual results. However, we have proposed more sophisticated algorithms which may produce perceptual results which are more in accordance with the objective results.

Bibliography

[1] AHMADI, S. and SPANIAS, A. S., “Low-bit-rate speech coding based on an improved sinusoidal model.” *Speech Communication*, 2001.

[2] ATAL, B. S. and HANAUER, S. L., “Speech Analysis and Synthesis by Linear Prediction of the Speech Wave.” *Journal of the Accoustic Society of America*, 1972.

[3] ATAL, B., “Efficient Coding of LPC Parameters by Temporal Decomposition.” *IEEE ICASSP*, 1985.

[4] ATAL, B. and SCHROEDER, M., “Predictive Coding of Speech Signals.” *Report of the 6th International Conference on Accoustics*, 1968.

[5] ATAL, B. and SCHROEDER, M., “Code-excited Linear Prediction (CELP) : High quality speech at very low bit rates..” *IEEE ICASSP*, 1985.

[6] BARNWELL, T. P. I. and BUSH, A. M., “Statistical Correlation Between Objective and Subjective Measures for Speech Quality.” *IEEE ICASSP*, 1978.

[7] BETTS, J. A., *High Frequency Communications*. London: The English Universities Press Ltd, 1967.

[8] BOERSMA, P. and WEENINK, D., *Praat: doing phonetics by computer*. http://www.praat.org.

[9] BURDEN, R. L. and FAIRES, J. D., *Numerical Analysis*. Third edition. Pacific Grove: Brooks/Cole, 1997.

[10] CAMPBELL, J. P. and TREMAIN, T. E., “Voiced/Unvoiced Classification of Speech with Application to the U.S. Government LPC10e Algorithm.” *IEEE ICASSP*, 1986.

[11] CERNOCKY, J., BAUOIN, G., and CHOLLET, G., “Segmental Vocoder - Going Beyond the Phonetic Approach.” *IEEE ICASSP*, 1998.

[12] CERNOCKY, J., BAUOIN, G., and CHOLLET, G., “Towards a Very Low Bit Rate Segmental Speech Coder.” *IEEE ICASSP*, 1998.

[13] CHAMBERLAIN, M., “A 600 bps MELP vocoder for use on HF channels.” *IEEE Military Communications Conference*, October 2001, Vol. 1.

[14] CHENG, Y.-M. and O'SHAUGHNESSY, D., "On 450-600b/s Natural Sounding Speech Coding." *IEEE Trans. Speech Audio Processing*, April 1993.

[15] CHU, W. C., *Speech Coding Algorithms*. Hoboken: Wiley, 2003.

[16] CHU, W. C., "Window Optimisation in Linear Prediction Analysis." *IEEE Trans. Accoustics, Speech and Signal Processing*, November 2003.

[17] COLLURA, J. S. and TREMIAN, T. E., "Vector Quantizer Design for the Coding of LSF Parameters." *U.S. Department of Defence*.

[18] DAUMER, W. R., "Subjective Evaluation of Several Efficient Speech Coders." *IEEE Transactions on Communications*, April 1982, Vol. 30.

[19] DE CHEVEIGNE, A. and KAWAHARA, H., "YIN, a fundamental frequency estimator for speech and music." *Journal of the Accoustic Society of America*, April 2002.

[20] DELLER, J. R., PROAKIS, J. G., and HANSEN, J. H., *Discrete-Time Processing of Speech Signals*. New York: Macmillan, 1993.

[21] DUDLEY, H., "The Vocoder." *Bell Labs*, December 1939.

[22] DUDLEY, H. and TARNOCZY, T. H., "The Speaking Machine of Wolfgang von Kempelen." *Journal of the Accoustic Society of America*, 1950.

[23] DUNN, H., "The calculation of vowel resonances and an electrical vocal tract.." *J. Acoust. Soc. Amer.*, November 1950.

[24] E.E. DAVID JR, B. L., M.R. SHROEDER and PRESTIGIACOMO, A., "Voice-Excited Vocoders for Practical Speech Bandwidth Reduction." *IEEE Trans. Information Theory*, Sept 1962.

[25] EL-JAROUDI, A. and MAKHOUL, J., "Discrete All-Pole Modeling." *IEEE Transactions on Signal Processing*, 1991.

[26] ETEMOGLU, C. O. and CUPERMAN, V., "Matching Pursuits Sinusoidal Speech Coding." *IEEE Trans. Speech Audio Processing*, 2003.

[27] FAIRBANKS, G., "Test of Phonemic Variation: The Rhyme Test." *Journal of the Accoustic Society of America*, 1958.

[28] FLANAGAN, J. L., "Bandwidth and channel capacity necessary to transmit the formant information of speech." *Journal of the Accoustic Society of America*, July 1956.

[29] FLANAGAN, J. L. and HOUSE, A. S., "Development and testing of a formant-coding speech compression system." *Journal of the Accoustic Society of America*, November 1956.

[30] FLANAGAN, J. L., SCHROEDER, M. R., ATAL, B. S., CROCHIERE, R. E., JAYANT, N. S., and TRIBOLET, J. M., "Speech Coding." *IEEE Trans. Comm.*, 1979, Vol. COM-27, No. 4, No. 4, pp. 710–737.

[31] GEORGE, E. B., MCCREE, A. V., and VISWANATHAN, V. R., "Variable Frame Rate Parameter Encoding Via Adaptive Frame Selection using Dynamic Programming." *IEEE ICASSP*, May 1996.

[32] GERSHO, A. and GRAY, R. M., *Vector Quantisation and Signal Compression*. Kluwer Academic Publishers, 1992.

[33] GOLDBERG, R. G. and RIEK, L., *A practical handbook of speech coders*. Florida: CRC Press, 2000.

[34] GOVERNMENT, S. A., *POLICY ANNOUNCEMENT BY THE MINISTER OF COMMUNICATIONS, DR IVY MATSEPE-CASABURRI*. http://www.info.gov.za/speeches/2004/04090310151004.htm, September 2004.

[35] GRAY, A. H. and MARKEL, J. D., "Quantisation and Bit Allocation in Speech Processing." *IEEE Trans. Accoustics, Speech and Signal Processing*, December 1976.

[36] HANSEN, J. H. and ARSLAN, L. M., "Robust Feature-Estimation and Objective Quality Assessment for Noisy Speech Recognition Using the Credit Card Corpus." *IEEE Trans. Accoustics, Speech and Signal Processing*, May 1995.

[37] HAYKIN, S., *Digital Communications*. New York: John Wiley & Sons, 1988.

[38] HOUSE, A., WILLIAMS, C., HECKER, M., and KRYTER, K., "Articulation testing methods: consonantal differentiation with a closed-response set.." *Journal of the Accoustic Society of America*, 1965.

[39] HUANG, X., ACERO, A., and HON, H.-W., *Spoken Language Processing*. New Jersey: Prentice Hall, 2001.

[40] ITAKURA, F., "Line Spectrum Representaion of Linear Predictive Coefficients of Speech Signals." *Journal of the Accoustic Society of America*, 1975.

[41] ITAKURA, F., SAITO, S., KOIKE, T., SAWABE, H., and NISHIKAWA, A. R., "An Audio Response Unit Based on Partial Autocorrelation." *IEEE Trans. on Communications*, August 1972.

[42] ITU-T, T. S. S. O. I., "Methods for Subjective Evaluation of Transmission Quality." August 1996.

[43] ITU-T, T. S. S. O. I., "Perceptual Evaluation of Speech Quality (PESQ): An Objective method for end-to-end seech quality assessment of narrow-band telephone networks and speech codecs." February 2001.

[44] JAYANT, N. S. and NOLL, P., *Digital Coding of Waveforms*. First edition. New Jersey: Prentice-Hall, 1984.

[45] JOZSEF VASS, Y. Z. and ZUANG, X., "Adaptive Forward-Backward Quantizer for Low Bit Rate, High Quality Speech Coding." *IEEE Trans. Speech Audio Processing.*

[46] KAY, S. M., *Modern Spectral Estimation.* New Jersey: Prentice Hall, 1988.

[47] KLATT, D., "Prediction of perceived phonetic distance from critical-band spectra: A first step." *IEEE ICASSP*, May 1982.

[48] LE ROUX, J. and GUEGUEN, C., "A Fixed-Point Computation of Partial Correlation Coefficients." *IEEE Trans. Accoustics, Speech and Signal Processing*, June 1977.

[49] LEE, K. S. and COX, R. V., "A Very Low Bit Rate Speech Coder Based on a Recognition/Synthesis Paradigm." *IEEE Transactions on Speech Audio Processing*, 2001.

[50] LOPEZ-SOLER, J. and FARVARDIN, N., "A Combined Quantization-Interpolation Scheme For Very Low Bit Rate Coding Of Speech LSP Parameters." *IEEE ICASSP*, 1993, Vol. 2, pp. 21–24.

[51] MARKEL, J. D., "The SIFT Algorithm for Fundamental Frequency Estimation." *IEEE Trans. on Audio and Electroaccoustics*, December 1972.

[52] MATTHAEI, P. E., "Automatic Speech Transcription." Master's thesis, University of Stellenbosch, April 2004.

[53] MCAULAY, R. and T.F.QUATIERI, "Low-bit-rate speech coding based on an improved sinusoidal model." *Speech Coding and Synthesis*, 1995.

[54] MCCANDLESS, S. S., "An Algorithm for Automatic Formant Extraction using Linear Prediction Spectra." *IEEE Trans. Accoustics, Speech and Signal Processing*, 1974.

[55] MCCREE, A. and III, T. P. B., "A Mixed Excitation LPC Vocoder Model for Low Bit Rate Speech Coding." *IEEE Transactions on Speech and Audio Processing*, July 1995.

[56] MCCREE, A. and MARTIN, J. C. D., "A 1.7 kB/s MELP Coder with improved Aalysis and Quantisation." *IEEE ICASSP*, 1998.

[57] MEYER, G. F., PLANTE, F., and AINSWORTH, W. A., "A pitch extraction reference database." *Proceedings of EUROSPEECH*, October 1995.

[58] MINOLI, D. and MINOLI, E., *Delivering Voice over IP Networks.* New York: Wiley, 1998.

[59] NIESLER, T., LOUW, P., and ROUX, J., "Phonetic analysis of Afrikaans, English, Xhosa and Zulu using South African speech databases." *Southern African Linguistics and Applied Language Studies*, 2005.

[60] NORMAN, J., *Elementary Dynamic Programming*. London: Edward Arnold, 1975.

[61] OF DEFENSE DIGITAL VOICE PROCESSOR CONSORTIUM, T. U. D., *MELP at 2.4Kbps*. http://maya.arcon.com/ddvpc/melp.htm, April 2002.

[62] OPPENHEIM, A. V. and SCHAFER, R., *Digital Signal Processing*. New Jersey: Prentice Hall, 1975.

[63] PALIWAL, K. K. and ATAL, B. S., "Efficient vector quantization of LPC parameters at 24 bits/frame." *IEEE Trans. Speech Audio Processing*, January 1993.

[64] POWELL, M. J. D., *Approximation Theory and Methods*. Cambridge: Cambridge University Press, 1981.

[65] PRANDONI, P. and VETTERLI, M., "R/D Optimal Linear Prediction." *IEEE Trans. Accoustics, Speech and Signal Processing*, November 2000.

[66] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., and FLANNERY, B. P., *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1997.

[67] PROAKIS, J. G. (Ed.), *Digital Communications*. Fourth edition. New York: McGraw-Hill, 2000.

[68] PROAKIS, J. G. and SALEHI, M. (Eds), *Communications Systems Engineering*. New Jersey: Prentice Hall, 2002.

[69] PUBLICATION, F. I. P. S., "Analog to Digital Conversion of Voice by 2,400 Bit/second Mixed Excitation Linear Prediction (MELP)." June 1997.

[70] RABINER, L. and SCHAFER, R., *Digital Processing of Speech Signals*. New Jersey: Prentice Hall, 1978.

[71] RAPID MOBILE, Pretoria. *Datasheet for RM6: HF Modem and ALE Controller Unit*, 2005.

[72] ROHWER, C. H., "Variation Reduction and LULU Smoothing." *Quaestiones Mathematicae*, 2002, Vol. 25, No. 2, No. 2, pp. 163–176.

[73] ROHWER, C. H. and WILD, M., "Natural Alternatives for One Dimensional Median Smoothing." *Quaestiones Mathematicae*, 2002, Vol. 25, No. 2, No. 2, pp. 135–162.

[74] SCHWARDT, L., "Voice Conversion: An Investigation." Master's thesis, University of Stellenbosch, December 1997.

[75] SCHWARTZ, M. (Ed.), *Information, Transmission, Modulation and Noise*. Second edition. New York: McGraw-Hill, 1970.

[76] SHANNON, C. E., "A Mathematical Theory of Communication." *Bell System Technical Journal*, 1948, Vol. 27, pp. 379–423, 623–656.

[77] SKLAR, B., *Digital Communications—Fundamentals and Applications*. Englewood Cliffs: Prentice Hall, 1988.

[78] SOONG, F. K. and JUANG, B.-H., “Optimal Quantisation of LSP Parameters.” *IEEE Trans. Speech Audio Processing*, January 1993.

[79] SPANIAS, A., “Speech Coding : A Tutorial Review.” *Proceedings of the IEEE*, October 1994.

[80] STRANG, G., *Introduction to Linear Algebra*. Massachusetts: Wellesley-Cambridge Press, 1993.

[81] SUBRAMANIAM, A. D. and RAO, B. D., “PDF Optimised Parametric Vector Quantisation of Speech Line Spectral Frequencies.” *IEEE Trans. Speech Audio Processing*, March 2003.

[82] TEXAS INSTRUMENTS, Texas. *C64xxDSP Family Manual*, 2005.

[83] TRIBOLET, J., NOLL, P., MCDERMOTT, B., and CROCHIERE, R., “A study of complexity and quality of speech waveform coders.” *IEEE International Conference on Audio, Speech and Signal Processing*, April 1978, Vol. 3.

[84] VISAGIE, A. S., “Speech Generation in a Spoken Dialogue System.” Master’s thesis, University of Stellenbosch, December 2004.

[85] VOIERS, W., “Diagnostic Acceptability Measure for speech Communications Systems.” *IEEE ICASSP*, 1977.

[86] WIKIPEDIA, *The Wikipedia*. http://en.wikipedia.org/wiki/Voder, May 2005.

Appendix A

Optimisation of the Linear Predictor

We therefore consider the mean squared error signal of the predictor $\mathbf{a}$ over a speech segment of N samples, $s[0] \dots s[N-1]$.

$$
\begin{aligned}
E &= \sum_{i=0}^{N-1} e^2[i] && \text{(A.1)} \\
&= \sum_{i=0}^{N-1} (s[i] - s^*[i]) && \text{(A.2)} \\
&= \sum_{i=0}^{N-1} \left(s[i] - \sum_{j=1}^{P} a(j) s(i-j) \right)^2 && \text{(A.3)}
\end{aligned}
$$

Thus the mean squared error signal over a speech segment is a quadratic function of the predictor coefficients and therefore has a unique global minimum, since the function $E(\mathbf{a})$ can be shown to be strictly convex.
We can determine the global minimum by finding the point in the vector space of $\mathbf{a}$ where all the partial derivatives of E are equal to zero. Call this point $\mathbf{a}^*$ then

$$
\left. \frac{\partial E}{\partial a_k} \right|_{a^*} = 0 \; ; \; \forall \{ k \in 1 \dots P \} : \qquad \text{(A.4)}
$$

But

$$
\begin{aligned}
\frac{\partial E}{\partial a_k} &= \frac{\partial}{\partial a_i} \sum_{i=0}^{N-1} \left(s[i] - \sum_{j=1}^{P} a(j) s(i-j) \right)^2 \\
&= \sum_{i=0}^{N-1} \left[2 \left(s[i] - \sum_{j=1}^{P} a(j) s[i-j] \right) (-s[i-k]) \right] \\
&= \sum_{i=0}^{N-1} \left[2 \left(\left(\sum_{j=1}^{P} a_j - s[i-k] s(i-j) \right) - s[i-k] s[i] \right) \right]
\end{aligned}
$$

thus

$$\forall\{k \in 1 \ldots P\} : \sum_{i=0}^{N-1} s[j-k]s(j) = \sum_{i=1}^{P} s[j-k]a(i) \sum_{i=0}^{N-1} s(n-i)$$

if we use the notation that

$$\phi_{m,n} = \sum_{i=0}^{N-1} s[i-n]s[i-m]$$

then the above reduces to:

$$\forall\{k \in 1 \ldots P\} : \phi_{k,0} = \sum_{i=1}^{P} \phi_{i,j}a(j)$$

or in matrix form

$$\begin{bmatrix} \phi_{1,0} \\ \phi_{2,0} \\ \phi_{3,0} \\ \vdots \\ \phi_{p,0} \end{bmatrix} = \Phi \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix}$$

with

$$\Phi = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,p} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,p} \\ \phi_{3,1} & \phi_{2,2} & \cdots & \phi_{3,p} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p,1} & \phi_{p,2} & \cdots & \phi_{p,p} \end{bmatrix}$$

We are able to deduce certain properties of the predictor from analysis of the covariance matrix [46].

- If the covariance matrix is positive definite (all eigenvalues greater than 0), the predictor will be stable - all the poles of the predictor will lie within the unit circle in the z-plane.
- If the covariance matrix is positive semidefinite, this implies that the analysis segment consists of the sum of p perfect sinusoids and the predictor will be marginally stable - all the poles of the predictor will lie on the unit circle in the z-plane.
- If the covariance matrix is not positive semidefinite, this implies that the predictor will be unstable - some of the poles of the predictor will lie outside the unit circle in the z-plane.

This proves to be problematic. An unstable predictor does not only not agree with the physical interpretation of the system, but also will cause problems in the realisation of the predictor.

Additionally, we are now left with the uncomfortable situation that we require samples outside the analysis segment in order to calculate the Φ matrix. Furthermore, the inversion of the Φ matrix is numerically unstable and the nature of the Φ matrix does not guarantee that we will obtain a stable filter design for our LP synthesis filter. There is a common solution to this dilemma. By windowing the (assumed infinite) speech signal with a window $w[k]$ which is uniformly zero outside the analysis interval, we obtain the following result:

$$
\begin{aligned}
\phi_{m,n} &= \sum_{i=0}^{N-1} s[i-n]w[i-n]s[i-m]w[i-m] \qquad (A.5)\\
&= \sum_{i=0}^{N-1-k} s[i]w[i]s[i+k]w[i+k] \qquad (A.6)\\
&= r_{m-n} \qquad (A.7)
\end{aligned}
$$

noting that

$$\phi_{m,n} = \phi_{n,m}$$

and that r_{m-n} is the biased autocorrelation estimate for the (assumed stationary) time signal using the window $w[k]$. We can re-write Φ as

$$
\Phi = \begin{bmatrix}
r_0 & r_1 & \dots & r_p - 1 \\
r_1 & r_0 & \dots & r_p - 2 \\
r_2 & r_1 & \dots & r_p - 3 \\
\vdots & \vdots & \ddots & \vdots \\
r_{p-1} & r_{p-2} & \dots & r_0
\end{bmatrix}
$$

and the normal equations reduce to the special form :

$$\mathbf{r} = \Phi \mathbf{a}$$

where

$$
\mathbf{r} = \begin{bmatrix}
r_1 \\
r_2 \\
r_3 \\
\vdots \\
r_p
\end{bmatrix}
$$

This convenient approximation to the full Yule-Walker Equations, is known as the *modified* or *extended* Yule Walker Equations [46] and has become extremely common in speech processing applications, since it additionally allows for a very convenient numerical solution.

One obvious question to ask would concern the nature of the window $w[k]$ which we so glibly refer to in equation A.7. As previously stated, the window must have finite support. A common choice is the Hamming Window [55] [70] but Chu [16] considers windows more specifically adapted to the task.

Appendix B

Derivation of the Line Spectrum Frequencies

We first derive the LSFs of the linear system with transfer polynomial given by:

$$a(z) = \sum_{i=0}^{P} a_i z^i$$

first define:

$$\begin{aligned} p(z) &= a(z) + z^{-(P+1)} a(z^{-1}) \quad (B.1) \\ q(z) &= a(z) - z^{-(P+1)} a(z^{-1}) \quad (B.2) \end{aligned}$$

Now consider a second order predictor:

$$a(z) = 1 - a_1 z^{-1} - a_2 z^{-2} = 1 - 2\rho_0 \cos(2\pi f_0) z^{-1} - \rho_2^2 z^{-2}$$

then

$$\begin{aligned} p(z) &= 1 - (a_1 + a_2) z^{-1} + (a_1 + a_2) z^{-2} - z^{-3} \quad (B.3) \\ q(z) &= 1 - (a_1 - a_2) z^{-1} - (a_1 - a_2) z^{-2} - z^{-3} \quad (B.4) \end{aligned}$$

Then $z = -1$ is a root of $p(z)$ and $z = 1$ is a root of $q(z)$. We can therefore factorise out $(1 + z^{-1})$ and $(1 - z^{-1})$ from $p(z)$ and $q(z)$ respectively. This Results in

$$\begin{aligned} p(z) &= (1 + z^{-1})(1 - 2\beta_1 z^{-1} + z^{-2}) \quad (B.5) \\ q(z) &= (1 - z^{-1})(1 - 2\beta_2 z^{-1} + z^{-2}) \quad (B.6) \end{aligned}$$

It can be shown that β_1 and β_2 are both on the interval $(-1, 1)$ for any value of f_0 and ρ_0. thus the roots of $p(z)$ and $q(z)$ are complex and given by

$$\beta_1 \pm j\sqrt{1 - \beta_1^2}$$

and

$$\beta_2 \pm j\sqrt{1-\beta_2^2}$$

respectively. Because the roots lie on the unit circle, they can be uniquely represented by their angles. These angles are known as the *line spectral frequencies* of $a(z)$ and are given by:

$$\cos(2\pi f_1) = \rho_0 cos(2\pi f_0) + \frac{1-\rho_0^2}{2} \tag{B.7}$$

$$\cos(2\pi f_2) = \rho_0 cos(2\pi f_0) - \frac{1-\rho_0^2}{2} \tag{B.8}$$

Now it can be shown that $f_1 < f_0 < f_2$ and that the three frequencies become close as the pole of the second order system moves close to the unit circle. The following also hold for more general cases:

1. The roots of $p(z)$ and $q(z)$ lie on the unit circle
2. that ±1 are roots
3. once sorted by complex angle, the roots of $p(z)$ and $q(z)$ alternate on the unit circle.

Thus the P predictor coefficients can always be transformed into P line spectral frequencies.

To compute the LSFs for higher order systems we replace $z = \cos(\omega)$ and compute the roots of $p(\omega)$ and $q(\omega)$ by any root finding method. The *bisection* method for finding these roots is popular [66]. To compute the predictor coefficients form the LSFs we can factor $p(z)$ and $q(z)$ as a product of second order filters, and then

$$a(z) = \frac{p(z)+q(z)}{2}$$

Unfortunately since the LSFs correspond to the poles of the transfer function of the Linear Predictor, calculation of the LSFs involves the factorisation of the LPs characteristic polynomial. This may introduce problems since polynomial factorisation is well known to be a computationally intensive and numerically unstable problem [9]. Fortunately, [66] provides a convenient solution for this problem using the $m \times m$ so-called *Companion Matrix*

$$A = \begin{bmatrix} -\frac{a_{m-1}}{a_m} & -\frac{a_{m-2}}{a_m} & \dots & -\frac{a_1}{a_m} & -\frac{a_0}{a_m} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}$$

The eigenvalues of this matrix correspond to the zeros of the polynomial given by:

$$\sum_{i=0}^{m} a_i z^i$$

Of course a number of well-documented algorithms exist for extraction of eigenvalues in for example [66]. Many of these are numerically well-conditioned problems and are computationally tractable.

Appendix C

Derivation of the Levinson-Durbin Algorithm

The Levinson-Durbin algorithm uses the following properties of the correlation matrix:

1. The correlation matrix of a given size contains as sub-blocks all the lower order correlation matrices.
2. The correlation matrix is invariant under the interchange of its columns and rows.

To do this we consider an augmented normal equation in the form:

$$\begin{bmatrix} r_0 & r_1 & \dots & r_P \\ r_1 & r_0 & \dots & r_P - 1 \\ r_2 & r_1 & \dots & r_P - 2 \\ \vdots & \vdots & \ddots & \vdots \\ r_{P-1} & r_{P-2} & \dots & r_1 \\ r_P & r_{P-1} & \dots & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} J \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

First we calculate the zeroth order predictor:

$$R[0] = J_0$$

Expanding this to a trivial predictor of order 1 $(a_1 = 0)$, we find that :

$$\begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} J \\ \Delta_0 \end{bmatrix}$$

Which is equivalent to:

$$\begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \Delta_0 \\ J \end{bmatrix}$$

implying that

$$\Delta_0 = r_1$$

We are now ready to solve for the optimal predictor of order 1, which satisfies the equation:

$$\begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1^1 \end{bmatrix} = \begin{bmatrix} J_1 \\ 0 \end{bmatrix}$$

Where the superscript of a denotes the predictor order. J_1 represents the minimum mean-squared error (MSE) attainable with a predictor of order 1. Let

$$\begin{bmatrix} 1 \\ a_1^1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - k_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then, multiplying by the correlation matrix produces:

$$\begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1^1 \end{bmatrix} = \begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} - k_1 \begin{bmatrix} r_0 & r_1 \\ r_1 & r_0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Which reduces to:

$$\begin{bmatrix} J_1 \\ 0 \end{bmatrix} = \begin{bmatrix} J_0 \\ \Delta_0 \end{bmatrix} - k_1 \begin{bmatrix} \Delta_0 \\ J_0 \end{bmatrix}$$

Implying that

$$k_1 = \frac{J_0}{\Delta_0} = \frac{r_1}{J_0}$$

Thus the optimal predictor of order 1 is represented by:

$$-k_1 z^{-1}$$

And the MSE for the predictor is given by:

$$J_1 = J_0(1 - k_1^2)$$

To obtain an optimal predictor of order 2 by solving

$$\begin{bmatrix} r_0 & r_1 & r_2 \\ r_1 & r_0 & r_1 \\ r_2 & r_1 & r_0 \end{bmatrix} \begin{bmatrix} 1 \\ a_1^2 \\ a_2^2 \end{bmatrix} = \begin{bmatrix} J_2 \\ 0 \\ 0 \end{bmatrix}$$

We consider a solution of the form

$$\begin{bmatrix} 1 \\ a_1^2 \\ a_2^2 \end{bmatrix} = \begin{bmatrix} 1 \\ a_1^1 \\ 0 \end{bmatrix} - k_2 \begin{bmatrix} 0 \\ a_1^1 \\ 0 \end{bmatrix}$$

Once again we multiply by the correlation matrix and obtain

$$a_2^2 = -k_2 \qquad \text{(C.1)}$$

$$a_1^2 = a_1^1 - k_2 a_1^1 \qquad \text{(C.2)}$$

Where

$$k_2 = \frac{1}{J_1}\left(r_2 + a_1^1 r_1\right)$$

and

$$J_2 = J_1(1 - k_2^2)$$

C.0.1 Summary

The L-D algorithm calculates the lth reflection coefficient by:

$$k_l = \frac{1}{J_l - 1}\left(r_l + \sum_{i=1}^{l-1} a_i^{l-1} r_{l-i}\right)$$

and the l'th predictor MSE by

$$J_l = J_{l-1}(1 - k_l^2)$$

Using the initial condition that

$$J_0 = r_0$$

and

$$k_0 = 1$$

and additionally using the conversion in equation 3.8 to calculate the predictor coefficients at every iteration.

Appendix D

African and European Phonetics

Although the MELP speech production model is suited in a very general and language independent way to the human speech production mechanism, substantial linguistic bias may still be introduced. This was demonstrated by the evaluation of the MELP vocoder in chapter 8, in which we found differences in the quality of the trans-coded Xhosa and English speech both during subjetcive and objetcive evaluation.
As has been shown by Niesler [59], there is a substantial difference between the phonetic content of African and European languages. We investigated two primary areas in which the two language groups may differ, with particular reference to the two languages which were used in evaluation we performed in chapter 8, namely English and Xhosa.

D.1 Phoneme Frequency

The frequency of occurrence of various types of phonemes in Xhosa is different to English. Table D.1 enumerates the most important differences. Values in table D.1 indicate occurrence of the phoneme category as a proportion of the total phone occurrence. Results were taken from approximately 150000 phonemes in each language,

Phoneme Class	Occurrence in English	Occurrence in Xhosa
approximants	0.11	0.08
clicks	0.00	0.02
diphthongs	0.08	0.03
fricatives	0.18	0.12
trills	0.00	0.01
vowels	0.32	0.43

Table D.1: *Some phoneme classes which have significantly different frequency of occurrence in Xhosa and English.*

extracted from sentences in the AST speech database. Since phoneme discrimination is often done primarily on the basis of the spectral envelope, it makes sense that the different phoneme distributions would necessitate different vector quantisation codebooks for the LP system.

D.2 Tempo and Rhythm

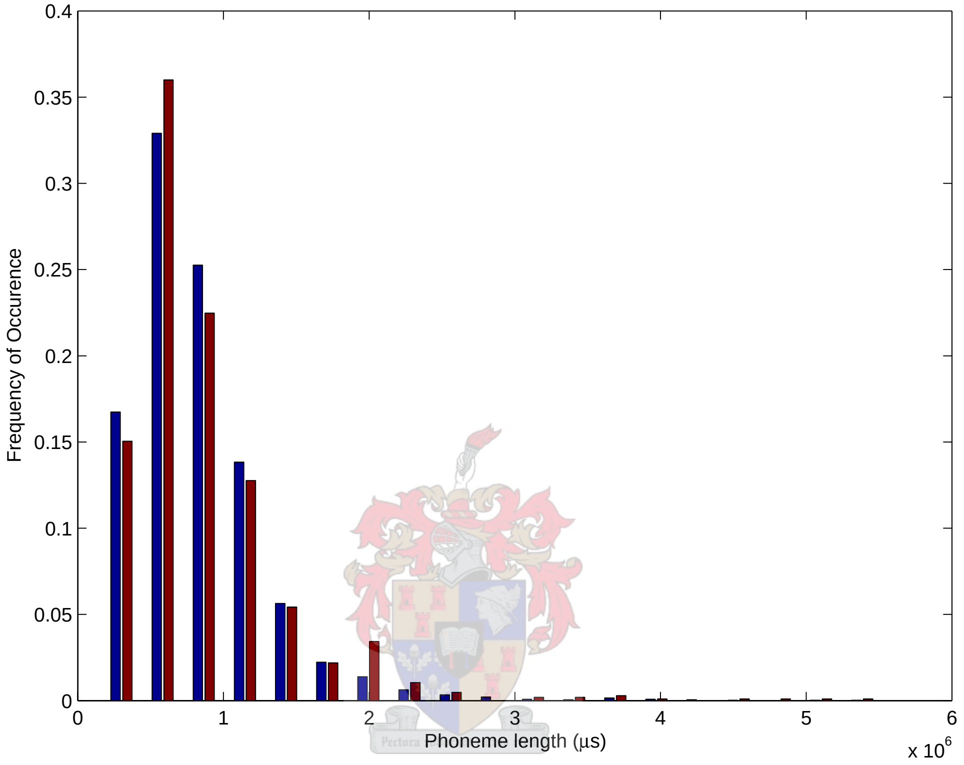


Figure D.1: *Distribution of phoneme lengths in Xhosa and English. Phonemes were automatically segmented. Statistics were collected over the sentences used in testing as described in section 8.1. Approximately 4000 Xhosa phonemes and 1000 English phonemes were used.*

It was hypothesised that there may be more variance in the rate at which Xhosa speakers speak. If this was the case then we would see significant differences in the histograms of phoneme lengths for the two languages. This would mean that a vocoder with an adaptive rate would be advantageous since it would more efficiently encode longer phonemes and would be less likely to distort short phonemes. However, analysis of the speech corpus using automatic segmentation and analysis of phoneme lengths revealed that the phoneme length distributions for English and Xhosa were extremely similar as shown in figure D.1.

Appendix E

Vector Quantisation

When the components of a random vector are statistically dependant, it is substantially more efficient to quantise the entire vector than to quantise the components separately [67].

Vector Quantisation can be viewed as a form of pattern recognition where the input pattern is matched to one of a set of standard or template patterns.

E.1 Definition of a Vector Quantiser

A vector quantiser Q of dimension k and size N is a mapping from a k-dimensional Euclidean space onto a finite set C of N elements, usually represented by a range of integers i.e. :

$$Q : R^n \rightarrow C$$

We commonly refer to the set C as the *codebook*.

The *resolution* of the quantiser is

$$r = \frac{\log_2 N}{k}$$

This gives an indication of the average number of bits per component used to represent the input vector.

E.2 Voronoi Quantisers

Voronoi or *Nearest-Neighbour* quantisers have the feature that the partition of the input space is completely determined by the codebook and distortion measure. The scheme consists only of a metric $d : R^n \times R^n \rightarrow R$ and a codebook C. To quantise a given vector x using this scheme, we simply find the entry $c_{opt} \in C$ which minimises $d(c_n, x)$. This type of vector quantiser is so common [32] that for the remainder of this text we will use

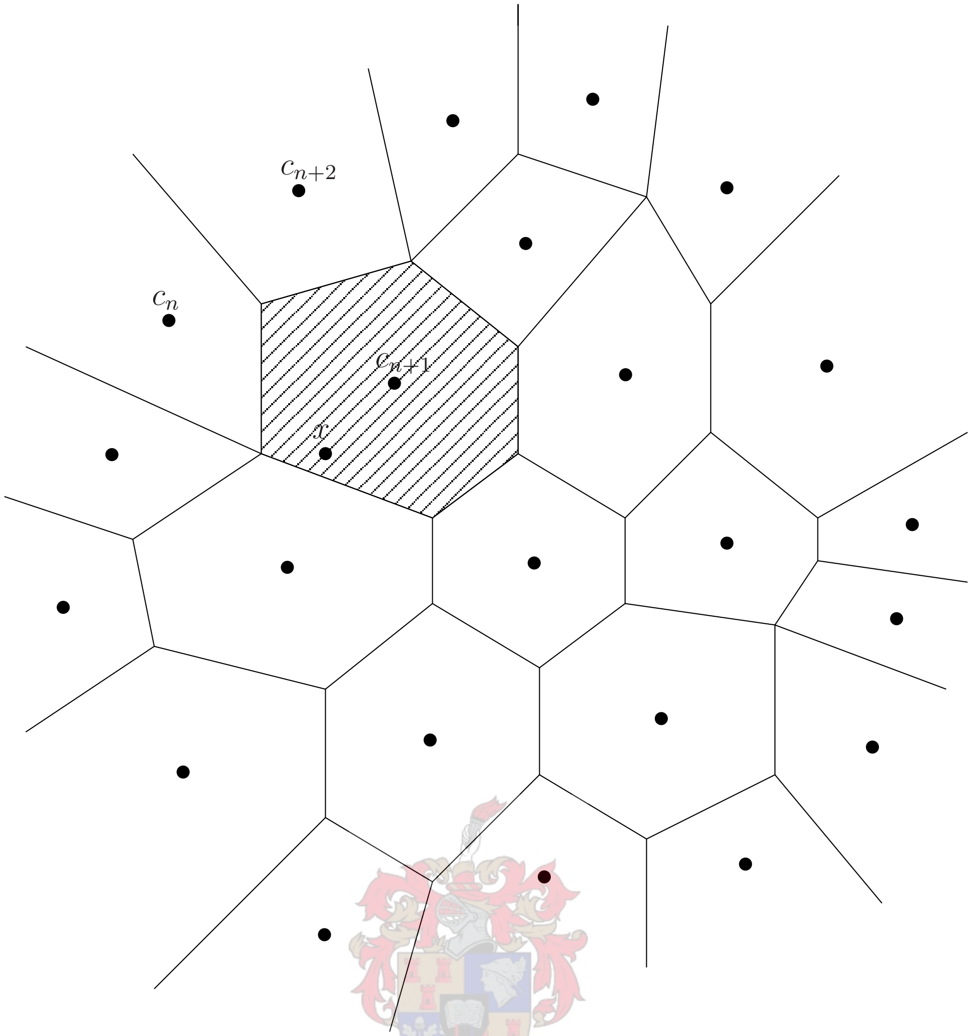


Figure E.1: *2-D Example of a vector quantiser. In the above example the points indicated by c_n, c_{n+1} and $c_{[}n+2]$ represent the various codebook entries. The point labelled x is a vector to be encoded. The region associated with each codebook entry is indicated. x lies in the region associated with c_{n+1} and as such will be encoded as $n+q1$ and decoded as c_{n+1}*

the words *vector quantiser* as synonymous with *nearest neighbour vector quantiser*. The primary advantage of this kind of quantiser is that no explicit description of the geometry of the quantiser cells is necessary.

E.3 Expected Quantisation Distortion

From [32]:
The Average distortion of a quantiser with partition cells R_i and reproduction vectors

$y_i, i = 1 \ldots N$ the average Euclidean distance distortion can be expressed as:

$$D = \sum_{i=1}^{N} \int_{R_i} f_X(x)|x - y_i|^2 dx \tag{E.1}$$

E.4 Constrained Vector Quantisation

The general algorithm for unconstrained VQ amounts to a nearest-neighbour search in a vector space of high dimensionality, which is well known to be a computationally intractable problem . There are several schemes which improve on the computational complexity of the problem, at the cost of less efficient quantisation (higher average quantisation distortion for a given number of bits). A host of different techniques are described in [32]. Here we describe only the three which are most commonly used in speech coding applications.

E.4.1 Multi-Stage Vector Quantisation

In Multi-Stage Vector Quantisation or MSVQ, the target vector is approximated as the sum of entries from several different codebooks.

$$x \approx \sum_{k=1}^{M} \mathbf{c}_k \tag{E.2}$$

Where $\mathbf{c}_k$ represents an entry from the k'th codebook. The approach typically followed is to use the $k+1$th codebook to to encode the residual of the kth stage.

E.4.2 Split Vector Quantisation or SVQ

This is a simple but effective technique where the vector x (with N components) to be quantised is partitioned by component into two or more sub-vectors x^n, where each sub-vector has M^n components. We may think of this equivalently as separate quantising of various *projections* of the vector on sub-spaces of the entire vector space.

$$\text{For} \quad i \in \{1 \ldots M^n\} \tag{E.3}$$

$$x_i^n = x_k \quad ; \quad k \in \{1 \ldots N\} \tag{E.4}$$

Each sub-vector is then quantised separately, using unconstrained VQ or MSVQ.

E.4.3 Transform VQ

In this technique, the vector to be encoded is transformed using a linear transformation to a different (usually orthogonal) basis. The idea of the transformation is to compact the information of the vector into a subset of the vector components. One of the above

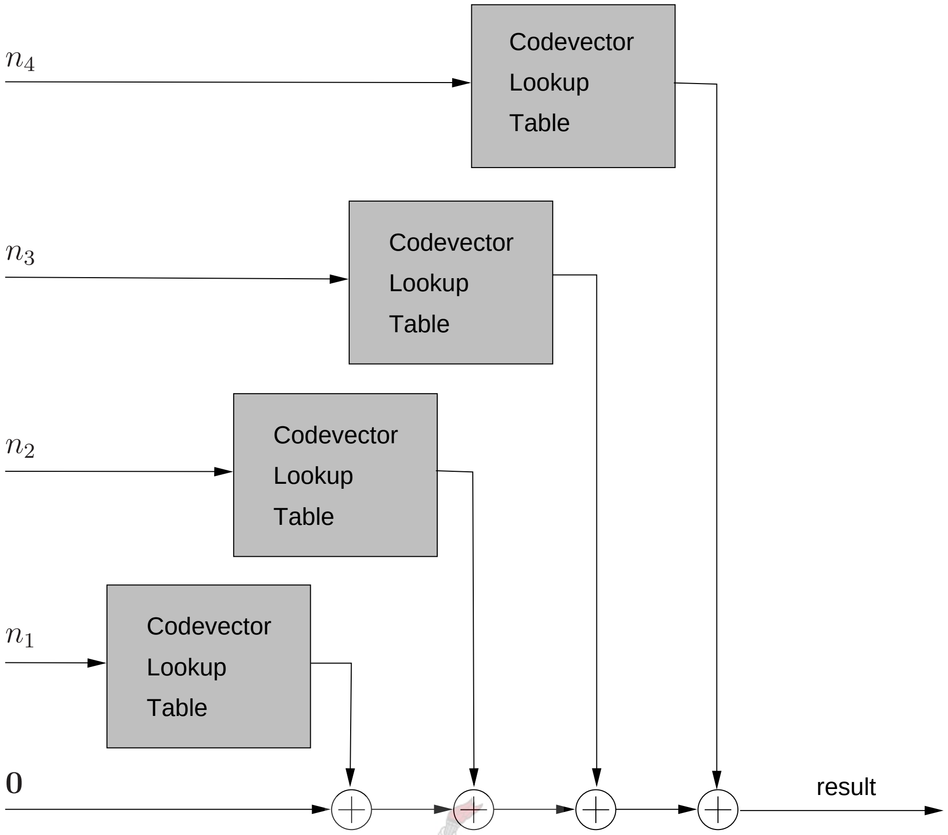


Figure E.2: *Multi-Stage Vector Quantiser*

vector quantisation techniques is then applied to the vector in the transformed space, as illustrated in figure E.3.

E.5 Vector Quantisation in Speech Coding

Vector quantisation techniques have been used in both waveform and parametric coders. Gersho [32] discusses some use of vector quantisation to utilise the substantial inter-sample redundancy exhibited by the speech waveform. However, efficient compression (in relation to the Shannon bound) is only achieved when the encoding is performed on many samples at a time. Since the impracticality of directly encoding high-dimensional vector spaces has already been discussed, we will not dwell on this idea. A much more useful application of VQ in speech coding is found in the design of parametric vocoders.

While we can think of a parametric vocoder as a form of vector quantisation (since the vector of possible input segments is assigned to a single integer index - if we regard e entire output packet of the parametric coder as an index), vector quantisation is most commonly used in parametric speech coding for the purpose of encoding the spectral

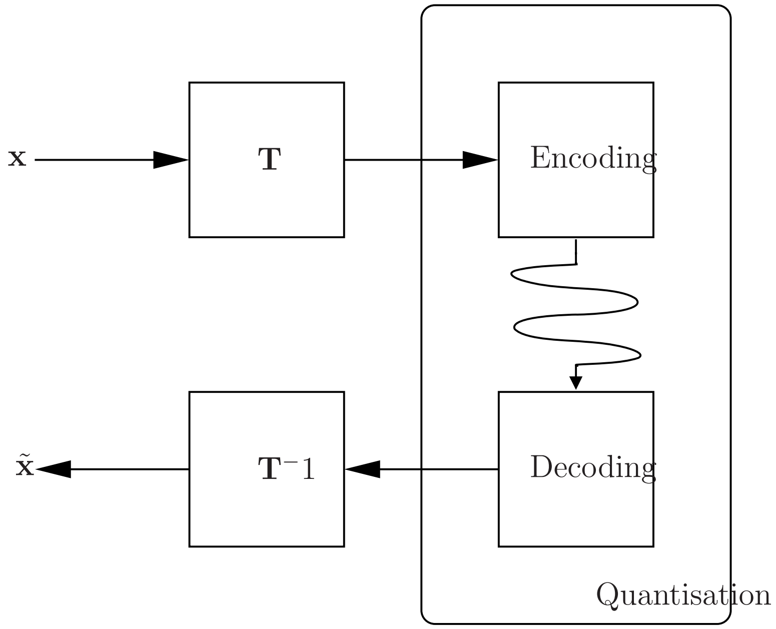


Figure E.3: *Transform Vector Quantisation*

envelope information. In MELP VQ is also used to encode the shape of the glottal pulses (see 4.3.1), but this is not common and was discarded in later refinements of the MELP model. [13, 56]. We will therefore focus our attention in the remainder of this section on the vector quantisation of the spectral envelope.

As mentioned in 3.2.1, the spectral envelope is the dominant factor used in speaker recognition and phoneme classification by both human listeners and automatic speech recognition systems. Thus it is not surprising that a large proportion of the information transmitted by most low rate vocoders is used to describe the spectral envelope. A typical case is MELP where 25 of 54 bits are used to transmit the LSFs.

The most popular methods in recent publications have been Split VQ and MSVQ. Other approaches are in [63] [78] [45].

There is a tradeoff between the computational complexity, distortion and bandwidth efficiency of a VQ. In vector quantisers which achieve equivalent amounts of distortion, one finds that the bandwidth efficiency is positively correlated with both computational complexity of the quantisation and the required memory used to store the codebooks [17].

Of course all of the above methods have only examined the redundancy exhibited between the individual components of a single linear predictor.However, Vass mentions that the

> . . . speech waveform is often slowly time varying and non-stationary. The statistics between the current block and some temporally close blocks may

often be similar leading to close sets of predictor coefficients . . . [45].

E.6 Quantiser Design Metrics

In the quantisation schemes described above, we have mainly mentioned the Euclidean distance as a distance measure. This measure is used, as mentioned above, in two ways:

- To determine a optimal set of codebook vectors.
- To determine the optimal code-vector to transmit for a given input vector.

This measure is both familiar and computationally tractable, but unfortunately in the quantisation of a linear predictor, it performs poorly. This is because the Euclidean distance between the two vectors representing linear predictors is a very poor representation of the perceptual 'distance' between their acoustic properties (or, to be precise, the acoustic properties of the lossless multi-tube system they represent).

This problem is to some degree alleviated by the choice of LP representation (for example, the Euclidean distance of two LSF vectors is a much more accurate measure than the Euclidean distance of two sets of Direct Form I coefficients). However, the non-linearities inherent in human hearing and perception mean that the Euclidean distance if far from optimal. It is necessary to use a more representative measure in the design and implementation of the VQ.

The following properties are of course desirable:

1. The metric should be more sensitive to spectral distortion measure rather than distance.
2. Secondly the distortion should be weighed according to the *subjective* sensitivity of the ear.
3. Thirdly, the energy of the frame should possibly be taken into account. If this is not done the design of the codebook may be influenced by low energy frames which have little or no influence on the perceptual qualities of the quantiser .
4. The voicing of the frame should be taken into account. [81, Rao] does this and seems to get good results but it does not seem to be common.

A metric which conforms to the above requirements may be substantially less computationally tractable, resulting in increased complexity in designing a codebook as well as increased complexity to encode an input vector. Additionally, such a metric may not be simple to analyse mathematically.

A number of approaches have been used in order to improve the perceptual accuracy of such metrics. MELP [55] uses a weighted Euclidean distance measure, discussed in 5. CELP v[5] uses a closed-loop analysis with a perceptual weighting filter as described in

4.2. Additionally, measures such as the *Itakura Saito* measure described in 3.7 may be useful.

E.7 Transparent Quantisation

In the vector quantisation of speech frames one would like to achieve what Paliwal and Atal in [63], call *Transparent Quantisation*. By this they mean quantisation which introduces no further audible distortion to the encoded speech (other than the distortion already introduced by the parameterisation performed by the vocoder). in [63], they mention the following criteria as being necessary for transparent quantisation.

- Average Spectral Distortion of less than 1dB [1].
- No out-lier frames with Spectral Distortion greater than 4dB.
- Less than 2% of frames have spectral distortion of greater than 2dB.

E.8 Literature

Collura and Tremain [17] give indications as to the number of bits per frame required to provide this level of quantisation using various vector quantisation schemes.
Subramaniam and Rao [81] recently proposed a promising new approach to LSF quantisation. The LSF vector is modelled as a random variable which has a PDF described by a Gaussian Mixture Model. The Expectation Minimisation Algorithm is used to generate the GMM from a large speech corpus. To quantise a single vector, we obtain a candidate from each cluster in the GMM which is the approximation of the vector within that cluster, using scalar quantisation on the (de-correlated) individual components of the vector. The 'best' of the candidates from all the clusters is chosen as the quantisation for the vector. Using this approach, almost transparent VQ is achieved at less than 20bits per frame.
In [50] a completely different approach is taken. Because the quasi-stationarity of the speech signal can occur for intervals of much greater than the commonly accepted 20-30ms, one can achieve substantial coding gain by exploiting the occurrence of longer stationary intervals in the speech waveforms. In this paper, only frames in which significant spectral change occurs are transmitted. In the paper, good results were obtained at bit rates as low as 350bits per second.

[1] Spectral distortion was defined in section 5.5.2

Appendix F

LULU Filters

Certain measurements in speech coding are subject to impulsive noise. A typical example of this would be the instantaneous pitch estimate which often contains pitch doubling and pitch tripling errors. Linear filtering is not a good solution to this problem, since a linear filter will disperse such an impulsive error across neighbouring samples. The amount that the impulse is dispersed increases as the amplitude of the impulse is decreased [75].

A solution to this de-noising problem is found in the use of non-linear filters. The most common type of non-linear filter is the median filter, which replaces every point in a sequence with the median of its neighbouring points. Unfortunately, median filters are not *idempotent* - successive application of a median filter to a sequence may not result in a stable solution[1]. Thus median filter are particularly unsuited for sequences which have closely spaced impulse errors. Furthermore, median filters require that the neighbours of a point in a sequence be sorted, a procedure which becomes computationally intractable for higher filter orders.

A refinement of median filters designed to address these problems is proposed by Rohwer using so-called LULU filters. The two types of LULU filters are typically referred to as L_nU_n and U_nL_n.

For a bi-infinite sequence x, we define $\bigvee$ and $\bigwedge$ so that:

$$(\bigvee x)_i = \max\{x_{i+1}, x_i\} \tag{F.1}$$

$$(\bigwedge x)_i = \min\{x_{i-1}, x_i\} \tag{F.2}$$

[1] A typical example of this behaviour results from the repeated application of a median filter to the sequence $\{\ldots, -1, +1, -1, +1, -1, +1, -1, +1, -1, \ldots\}$. Each application of a median filter simply reverses the polarity of the sequence

Then

$$U_n = \bigwedge{}^n \bigvee{}^n \tag{F.3}$$
$$L_n = \bigvee{}^n \bigwedge{}^n \tag{F.4}$$

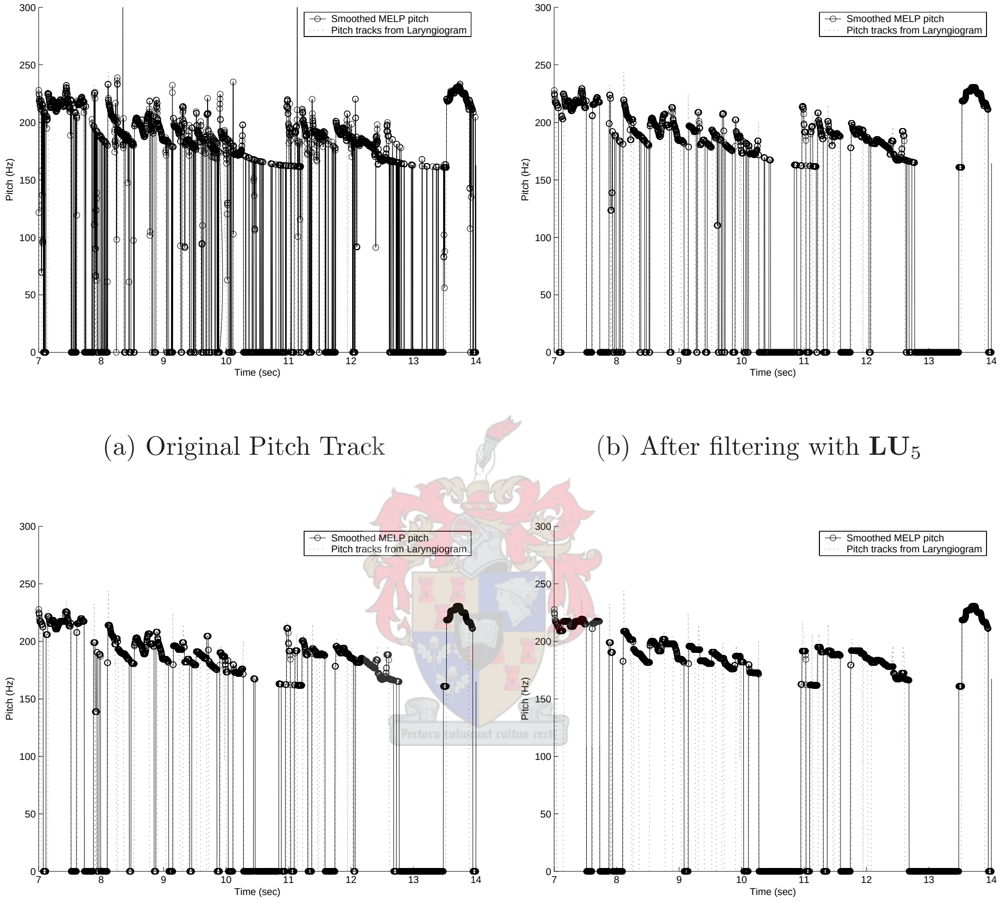
Rohwer describes several desirable properties of the L_n and U_n operators in [72] and [73]. The basic principle which he emphasises is that of *variation reduction*, implying that the L_n and U_n operators reduce the spread of sample values in the sequence. Rohwer illustrates in [72] how the L_n operator removes upward *block-pulses* or consecutive outliers of less than length n and U_n removes downward *block-pulses* of less than length n. Thus the concatenation of L_nU_n or U_nL_n will remove all *block-pulses* of less than length n. However, one must bear in mind that the non-linearity of the operators implies that L_nU_nx and U_nL_nx are not necessarily the same. In the context of a speech processing application we should therefore examine both the L_nU_n and U_nL_n operators and determine which produces better results.

Additionally, Rohwer recommends that one should filter using the concatenated filter constructed as $L_nU_nL_{n-1}U_{n-1}\ldots L_3U_3L_2U_2L_1U_1$ in order to achieve best results. We will refer to these as the LU and UL decomposition filters respectively, since they correspond to the output of various stages of the *LULU decomposition* referred to by Rohwer and denote these concatenated operators with $\mathbf{LU}_n$ and $\mathbf{UL}_n$.

In figure F.1, we illustrate the effect of LULU filtering on the pitch track as estimated by the MELP pitch tracking algorithm. As can be seen, even filtering with a low filter order reduces the number of pitch errors. Increasing the filter order removes the occurrence of isolated frames which are incorrectly classified as voiced or unvoiced. However, increasing the filter order also removes some resolution in the pitch, especially in the sharp pitch peaks exhibited in the correct pitch track. The pitch track is compared with the true pitch track, which was recovered from laryngogram data. [57].

In figure F.2, we illustrate the statistical effect of the L_nU_n and U_nL_n filters of both the simple and concatenated forms on pitch and voicing errors in the MELP analysis. Clearly the filter which exhibits the best performance for the pitch post-processing is the UL decomposition filter. In the optimal order, it reduces the incidence of gross pitch errors by almost 25% relative to the un-filtered pitch track.

In the case of the voicing decision, the optimal choice of filter is less obvious, since each of the filters represents a trade-off between the number of frames which are incorrectly classified as voiced and the number of frames which are incorrectly classified as unvoiced. In this case the optimisation can only be done using perceptual considerations.



(a) Original Pitch Track

(b) After filtering with $\mathbf{LU}_5$

(c) After filtering with $\mathbf{LU}_{10}$

(d) After filtering with $\mathbf{LU}_{20}$

Figure F.1: *LULU Filtering on a typical pitch track.*

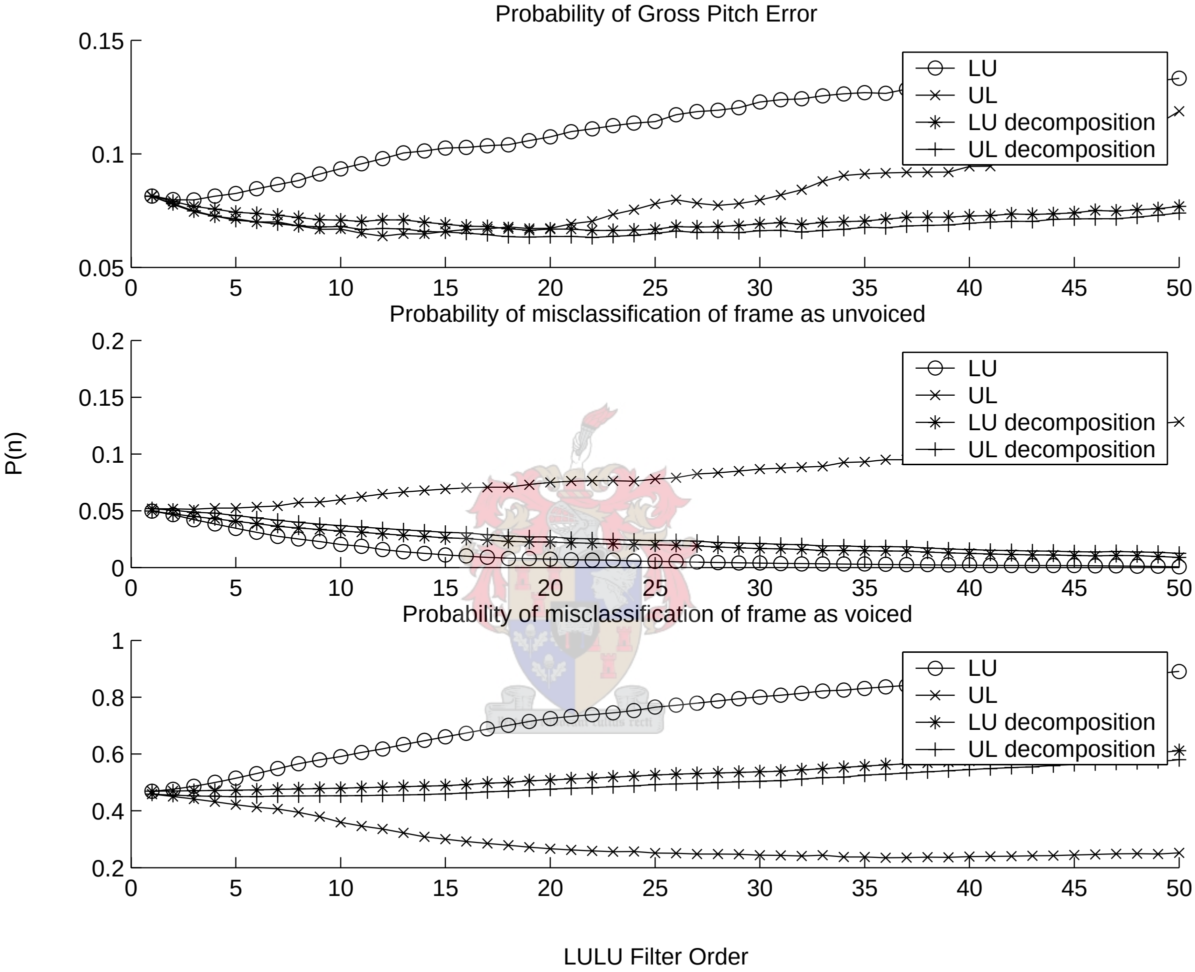


Figure F.2: *LULU Filtering of pitch and voicing.*

Appendix G

PESQ

The PESQ algorithm was used substantially in the development phase of our IS-MELP vocoder. We did not implement the PESQ algorithm, but used the ITU reference implementation. We present a short description of the algorithm.

G.1 Purpose

The PESQ algorithm compares an original and degraded signal and outputs a prediction of the perceived quality that would be given to the degraded signal by subjects in a subjective listening test.

G.2 Limitations

PESQ is not capable of correctly handling certain types of distortion (such as side-tones and delay). Therefore PESQ is not intended to replace subjective listening tests as the ultimate authority on the quality of a speech codec. However PESQ provides a very fast evaluation of the quality of a speech sample. Therefore it presents a substantial advantage over listener tests, since it allows us to make design decisions based on comprehensive analysis of large speech corpora.

G.3 Gain adjustment

The gain of the transcoding system is not known and therefore the level of the original and degraded speech are aligned. This is done by adjusting the frequency weighted energy of the original and degraded speech samples to a reference value. The aim of this step is to compensate for overall spectral balance differences which may be introduced by the coding scheme and which should not affect the score.

G.4 Time Alignment of the Signal

Since a voice coding system may introduce an unknown (and possibly varying) delay in the speech signal, the PESQ algorithm performs a dynamic time alignment between the original and distorted signal. This time alignment is performed by first performing an estimation of the delay of the entire signal and then performing alignment on silence-delimited sections of the signal, which are referred to as *utterances*.
Utterances are split and the sections of the utterances are re-aligned using a dynamic time warping algorithm in order to compensate for delay changes during the utterances. Subsections of utterances are aligned to maximise the correlation of the short-term energy of the original and degraded signals.
Subsequent to application of the perceptual model, severely distorted sections are re-aligned, using a further correlation.

G.5 Perceptual Model

PESQ analysis is performed using frame based processing. Frames are 32ms in length and are overlapped by 16ms. The PESQ algorithms calculates the difference between the original and degraded signal by comparing the power spectra of corresponding frames of the two signals. The comparison of the power spectra is based on a number of perceptual considerations, including the following:

1. The power spectrum is binned into Bark bands to reflect the fact that the human ear has better frequency resolution at lower frequencies.
2. The power spectrum is transformed into a *loudness* value to account for non-linearities in perception of sounds. This transformation is performed using Zwicker's law. This transformed spectrum is referred to as the *loudness* spectrum.
3. The difference between the loudness spectra (as defined in the previous steps) of the original and distorted signals is referred to as the *disturbance density*.
4. Since introduced spectral components are much more perceptually disturbing than removed spectral components, a second (*asymmetric*) disturbance density is calculated to emphasise introduced spectral components in the degraded signal. This error signal is intended to reflect spectral information which is present in the synthesised speech but not in the original.

G.6 Integration of Disturbance

The disturbance density and asymmetrical disturbance density are integrated in time using different L_p norms, with additional weighting given to low-energy frames (such as portions of silence in the speech).

G.7 Final MOS Estimate

The final MOS estimate as predicted by the PESQ algorithm is a linear combination of the average disturbance and and average asymmetrical disturbance values.