Techniques for Harmonic Sinusoidal Coding

by

David Grant Rowe

Bachelor of Engineering in Electronic Engineering (1989)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

School of Physics and Electronic Systems Engineering

Faculty of Information Technology

July 1997

Contents

List of Figures

List of Tables

Glossary of Abbreviations

AR	Auto Regressive
ARMA	Auto Regressive Moving Average
CD	Cepstral Distortion
CDMA	Code Division Multiple Access
CELP	Code Excited Linear Prediction
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
FEC	Forward Error Correction
FIR	Finite Impulse Response
GSM	Global System for Mobile telecommunications
IIR	Infinite Impulse Response
IDCT	Inverse Discrete Cosine Transform
IDFT	Inverse Discrete Fourier Transform
LPC	Linear Predictive Coding
LSP	Line Spectrum Pair
IMBE	Improved Multi-Band Excitation
MBE	Multi-Band Excitation
MSE	Mean Square Error

NLP	Non-Linear Pitch
PCM	Pulse Code Modulation
PSTN	Public Switched Telephone Network
RMS	Root Mean Square
RPE	Regular Pulse Excitation
SD	Spectral Distortion
SEGSNR	Segmental Signal to Noise Ratio
SNR	Signal to Noise Ratio
VSELP	Vector Sum Excited Linear Prediction

Glossary of Symbols

a_m	lower limit of band in DFT domain
$\{a_k\}$	set of p linear prediction coefficients for $k = 1, 2, \ldots, p$
$\mathbf{a}$	vector of L complex sinusoidal amplitudes
$A(z)$	linear predictive analysis filter
$\frac{1}{A(z)}$	linear predictive synthesis filter
$\{A_m\}$	set of L complex sinusoidal amplitudes, for $m = 1,2,\ldots, L$
$\{\hat{A}_m\}$	set of L synthesised complex sinusoidal amplitudes, for $m = 1,2,\ldots, L$
b_m	upper limit of band in DFT domain
$\mathbf{B}$	L by L polynomial weighting vector
$\{B_m\}$	set of L complex sinusoidal magnitudes, for $m = 1,2,\ldots, L$
$\{\hat{B}_m\}$	set of L modelled complex sinusoidal magnitudes, for $m = 1,2,\ldots, L$
c	constant used to normalise sinusoidal analysis window, also used as constant describing trapezoidal synthesis window
c_k	weighted polynomial coefficient
$c(k)$	constant used in DCT calculation
$\mathbf{c}$	vector of L modelled complex sinusoidal amplitudes, or vector of K weighted polynomial coefficients

C	number of transmitted DCT coefficients
$\hat{C}_m$	m^{th} modelled complex sinusoidal amplitude
D	decimation ratio
$e(n)$	error sequence
$\mathbf{e}$	vector of phase modelling error samples
E	mean square error
$E_a^{(L)}$	mean square error of L^{th} adaptive codebook entry
$E_s^{(i)}$	mean square error of i^{th} stochastic codebook entry
$E\left(\hat{\omega}_0^l\right)$	squared error between original and all-voiced synthetic speech
$E_m\left(\hat{\omega}_0^l\right)$	squared error between original and all-voiced synthetic speech for the m^{th} band
$\left\|E(k)\right\|^2$	Error magnitude spectrum
$F0$	fundamental frequency in Hz
F_s	sampling frequency in Hz
g	real linear predictive model gain
G	complex linear predictive model gain for positive frequencies
G_0	Magnitude component of G and $G(\omega)$
$G(\omega)$	complex linear predictive model gain for all frequencies
$G(k)$	frequency-dependant error weighting function
$h(n)$	impulse response

$\hat{h}(n)$	impulse response of synthesis filter
$H(z)$	z-transform of $h(n)$, notch filter transfer function
$H_{\min}(z)$	z-transform of minimum phase system
$H_{ap}(z)$	z-transform of all pass system
i	stochastic codebook entry
$k_{\max}$	DFT bin containing global maxima of $U(k)$
k_v	DFT bin corresponding to the v^{th} local maxima of $U(k)$
K	order of phase residual modelling polynomial
L	adaptive codebook entry (CELP), number of harmonics (sinusoidal coder)
L_T	last voiced harmonic
$L_{\max}$	Maximum number of sinusoids in sinusoidal coder
M	size pitch analysis window
$\mathbf{M}$	L by K matrix of $\mathbf{p}(m)$
n_0	impulse position for phase model
N	frame size
N_{SF}	CELP sub-frame size
N_s	number of stochastic codebook entries
N_a	number of adaptive codebook entries
N_w	analysis window size

N_{wl}	lower limit of analysis window
N_{wu}	upper limit of analysis window
N_{dft}	size of discrete fourier transform
$\mathbf{p}(m)$	vector of powers of m
P	pitch period in samples
$P(z)$	symmetric LSP polynomials
$PSNR$	Prototype Signal to Noise Ratio (dB)
$\mathbf{q}$	vector of K phase residual samples
$Q(z)$	anti-symmetric LSP polynomial
$r(n)$	linear prediction residual
R_m	RMS magnitude of m^{th} band
$\mathbf{R}$	$K+1$ by $K+1$ matrix used in solving for weighted polynomial coefficients
$R(k)$	autocorrelation sequence
$\hat{R}(k)$	autocorrelation sequence of synthesis filter impulse response
$s(n)$	original speech sequence
$\hat{s}(n)$	synthesised speech sequence
$\hat{s}_{zi}(n)$	zero input response of $\frac{1}{A(z)}$
$\hat{s}_{zs}(n)$	zero state response of $\frac{1}{A(z)}$

$\hat{s}_a^{(L)}(n)$	unit-gain, zero state response from the adaptive codebook excitation
$\hat{s}_s^{(i)}(n)$	unit-gain, zero state response from the stochastic codebook excitation
$s_s(n)$	original speech with $\hat{s}_{zi}(n)$ removed
$s_w^l(n)$	n^{th} sample of l^{th} windowed analysis frame
$s^l(n)$	n^{th} sample of l^{th} analysis frame
$\hat{s}_l(n)$	synthesised speech from the l^{th} frame
S	$U(k)$ DFT bin spacing in Hz
$S(z)$	z-transform of $s(n)$
$\hat{S}(z)$	z-transform of $\hat{s}(n)$
$S_w^l(k)$	discrete fourier transform of $s_w^l(n)$
$S^l(k)$	discrete fourier transform of $s^l(n)$
$\hat{S}(k)$	estimate of $S^l(k)$
$\hat{S}_w^l(k,m)$	frequency domain synthesised speech for the m^{th} band
$t(n)$	trapezoidal synthesis window
T	Unvoiced energy threshold (dB), pitch candidate selection threshold
T_0	Experimentally derived constant for pitch candidate selection
T_w	constant defining overlap of trapezoidal synthesis window

$\mathbf{u}$	$K+1$ element vector used in solving for weighted polynomial coefficients
$U(k)$	$U(k)=\lvert Z(k)\rvert^2$, power spectrum of $Z(k)$
$v^l(m)$	voicing measure in m^{th} band
V	is the number of local maxima in $U(k)$
$V(k)$	DCT of $v(n)$
$w(n)$	data window
$w_r(n)$	rectangular data window
$W(k)$	discrete fourier transform of $w(n)$
$x(n)$	excitation sequence, time domain sequence
$x_a^{(L)}(n)$	n^{th} sample of adaptive codebook entry L
$x_s^{(i)}(n)$	n^{th} sample of stochastic codebook entry i
$X(k)$	DFT of $x(n)$
$X(z)$	z-transform of $x(n)$
$y(n)$	time domain sequence
$Y(k)$	DFT of $y(n)$
$Z(k)$	DFT of $x(n)y(n)$, DFT of $x^2(n)$
α	adaptive codebook gain

β	stochastic codebook gain
γ	bandwidth expansion factor
$\delta(n)$	delta function
φ	phase of complex LPC gain G
$\Delta\omega_0^l$	error in fundamental estimate
$\{\theta_m\}$	set of L sinusoidal phases, for $m = 1,2,\ldots,L$
$\{\hat{\theta}_m\}$	set of L modelled sinusoidal phases, for $m = 1,2,\ldots,L$
$\{\phi_m\}$	set of L phase residual samples, for $m = 1,2,\ldots,L$
$\{\hat{\phi}_m\}$	set of L modelled phase residual samples, for $m = 1,2,\ldots,L$
$\{\omega_m\}$	set of L sinusoidal frequencies, for $m = 1,2,\ldots,L$
ω_0	fundamental frequency in normalised radians/sec
ω_0^l	fundamental frequency of l^{th} frame in normalised radians/sec
$\hat{\omega}_0^l$	estimate of ω_0^l
ω_t	two-band voicing model transition frequency
ω_v	angular frequency corresponding to the pitch candidate DFT bin k_v
ω_i	i^{th} LSP frequency

Summary

Harmonic sinusoidal coders represent the speech signal as a sum of sinusoidal oscillators, each oscillator having an independent magnitude and phase. The frequencies of each oscillator are an integer multiple (harmonic) of the fundamental frequency. When the model parameters (amplitudes, phases, and fundamental frequency) are updated every10-30 ms, high quality speech can be produced.

This thesis presents several speech coding techniques for harmonic sinusoidal coders. Introductory chapters describe time and frequency domain speech coding, with special discussion and comparsion of CELP and sinusoidal algorithms. Several major contributions then follow. Demonstrations of the algorithms developed for this thesis are available in the form of speech files available via the internet.

A (fundamental frequency) pitch estimation algorithm based on a square law non-linearity is presented, known as Non-Linear Pitch (NLP). The algorithm has moderate computational complexity, low algorithmic delay (small buffering requirements), and robustness to gross pitch errors (halving and doubling). The algorithm employs a minimum number of experimentally derived constants.

A generic harmonic sinusoidal coder is presented. This coder has been implemented using a range of techniques from existing sinusoidal and Multi-Band Excitation (MBE) coders. In addition, new techniques are presented for the analysis, spectral magnitude modeling and synthesis stages. The unquantised coder produces output speech of very high quality, in some cases almost indistinguishable from the original speech signal.

Three parametric models for the compact representation of the harmonic phases are described. A phase model for voiced speech that consists of a minimum phase LPC synthesis filter cascaded with an all pass filter is proposed, and three candidate systems are then developed that implement this model. These models are evaluated using objective and informal subjective methods, and found to produce speech quality comparable to VSELP in clean speech conditions.

Finally, the thesis contributions are combined to produce a fully quantised coder. The quantised coder is based on the generic sinusoidal coder developed in this thesis, with the magnitudes modeled using LSP quantised LPC parameters. The sinusoidal phases are represented using the analysis by synthesis phase model developed in this thesis. The fully quantised coder produces communications quality speech at 8.3 kbit/s. The performance of the coder is analysed and discussed using objective and subjective techniques. Finally, suggestions for further work are provided.

Declaration

I declare that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text.

Signed: ....Date:...................................

Acknowledgements

I wish to thank my supervisor, Dr. Bill Cowley, for his guidance during the course of my studies. My thanks also go to my parents, for supporting my academic pursuits, and my wife Rosemary for her help and encouragement.

Dedication

This thesis is dedicated to my son, Matthew Nicholas, who never knew life; and to my daughter, Amy Kathleen, who has it in abundance.

1. Introduction

Speech coding deals with the problem of reducing the bit rate required for a speech representation while preserving the quality of speech reconstructed from that representation [1]. One motivation for speech coding is to allow speech communication over power or bandwidth limited channels. Such channels exist in communications systems employing digital radio, for example mobile services where power (due to portability constraints) and in particular spectrum is limited.

Recent interest in speech coding is motivated by the introduction of fully digital telecommunications services designed to carry speech signals. These include satellite systems using mobile terminals such as Inmarsat-M [12], and second generation terrestrial mobile systems such as GSM [13][14]. Apart from these systems, proposed uses for speech coding are future micro cellular mobile systems based on Code Division Multiple Access (CDMA) [15], and multimedia applications where efficient storage of speech information is required.

An important parameter of a speech coding system is the quality of the reconstructed speech. This can be used to describe the system, and the types of applications it might be suitable for. Table 1.1 is a summary of current systems used to code speech. Note that these systems are designed to code speech signals only; other coding systems exist that can code any audio signal [9], however these are beyond the scope of this thesis and will not be considered further.

This thesis deals with communications quality speech coding. Communications quality speech is defined as highly intelligible but with noticeable distortion [1]. However the quality of the reconstructed speech is such that different speakers using the system can be identified, unlike synthetic quality systems where all speakers sound similar due to the large amounts of distortion present. An example of communications quality speech is the existing analogue cellular system, and many non-digital Public Switched Telephone Networks (PSTN).

1.1 Speech Coding Overview

In practice, speech coding is performed by first sampling analogue speech signals at a sampling rate and resolution sufficient to preserve the desired quality. Commonly, 8 kHz sampling rate and 12-16 bit linear PCM is used to sample communications quality speech. Before sampling, the speech is usually passed through an analogue anti-aliasing filter with a pass band of 300-

3300 Hz. The speech is then processed using Digital Signal Processing (DSP) techniques to extract a compact representation in the form of a small number of bits that are then sent over the communications channel. The channel may be imperfect and can introduce errors in some cases. At the decoder the received bits are used to reconstruct the speech signal, which is then converted back to an analogue signal using a digital to analogue converter and reconstruction filter. Figure 1.1 illustrates the process.

Quality	Audio Bandwidth (Hz)	Distortion	Bit Rate (kbit/s)	Typical Application
Commentary	50 - 7000	Imperceptible to Slight	16 - 128	Video Telephone
Toll	300 - 3300	Imperceptible to Slight	8 - 64	Network Telephony
Communications	300 - 3300	Perceptible	2.4 - 16	Mobile Comms.
Synthetic	300 - 3300	Mechanical Sounding	0.8 - 2400	Secure Defence Comms.

Table 1.1: Summary of Speech Coding Systems

Several different techniques are currently used for speech coding [1][61]. Parametric or model based coders assume a speech production model and extract parameters from the speech signal that describe that model, updating the model parameters periodically as the characteristics of the speech signal change. Waveform coders assume no model, but attempt to minimise the error between the original and reconstructed speech waveforms. Many low bit rate communications quality speech coders use a combination of parametric and waveform matching techniques, and are thus known as hybrid coders.

The signal processing techniques used for speech coding may be based on time or frequency domain processing. The speech signal is often buffered into frames of short blocks of samples, typically 20 ms long. This allows the coding techniques to exploit redundancies across the frame to improve coding efficiency, at the expense of introduced delay.

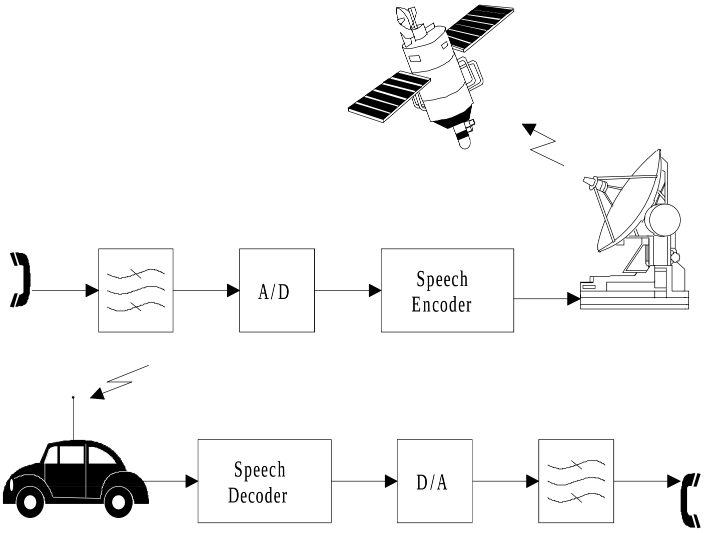


Figure 1.1: Speech Coding Model

1.2 Current Speech Coding Issues

From the author’s experience in the speech coding field, the following major issues in speech coding have been identified. Many authors have considered a subset of the issues below, and it is conceded that the issues are well known in practice, however no equivalent statement has been encountered by the author.

1.2.1 Speech Quality

An important speech coding consideration is speech quality versus bit rate. A superior speech coding algorithm will produce better quality speech at a given bit rate than an inferior algorithm. How to determine the speech quality of a given algorithm is still a matter of some debate [53][61]. The aim of most speech coding research is to lower the bit rate required for a given level of speech quality.

1.2.2 Computational Complexity

Lowering the bit rate while maintaining quality is often achieved at the expense of increased complexity. A complex algorithm requires powerful DSP hardware that is expensive and power hungry. Until the late 1980's, many speech coding algorithms were not implementable in real time due to the lack of sufficiently powerful real time DSP hardware. Powerful DSP chips now exist, however many existing algorithms push their capabilities to the limit [64]. Future algorithms are expected to demand more powerful DSP devices.

DSP hardware consumes significant amounts of power due to the high clock rates required (typically 50 MHz). This is a concern for services requiring speech coding for hand held or portable terminals. The reduction in complexity of speech coding algorithms would enable lower power consumption through the use of less powerful DSP hardware. Less powerful DSP hardware would also reduce cost.

Thus the search for computationally efficient algorithms is an important research activity; to reduce DSP hardware requirements, power consumption, and cost of speech coding hardware.

1.2.3 Robustness to Channel Errors

Speech coding is often used in channels that are power limited and thus subject to bit errors with random or bursty distributions [62]. In these cases, it is necessary that the speech coding algorithm be robust to the error conditions likely to be encountered in the prescribed operating conditions. This may be achieved in several ways. The first is by adding Forward Error Correction (FEC) to the encoded speech to protect against bit errors introduced during transmission. This can be used to improve the robustness of any speech coding algorithm but carries the penalty of extra bandwidth due to the added FEC information that must be transmitted.

The second method to provide robust coded speech transmission is to use algorithms with inherent robustness. Certain speech coding algorithms are more robust to bit errors than others [63], and are thus preferred for use with noisy channels. Other algorithms may perform poorly in noisy channels, or break down completely at even moderate bit error rates. The audible effects of this may be poor speech quality, annoying, or even physically painful noises being emitted from the speech decoder [65]. The investigation and improvement of algorithms robust to bit errors is therefore an important research issue.

1.2.4 Robustness to Acoustic Background Noise

Background noise presents a problem to many speech coders [66]. One of the reasons that speech can be coded at low bit rates is through the use of model based or hybrid techniques that exploit the redundancy of the speech signal. The assumptions these models make for speech coders are not necessarily true for other audio signals such as single sinusoids or background noise. These coders may reproduce speech sounds faithfully but distort or corrupt background noise in an annoying fashion. Another effect is that the signal processing techniques used to extract model parameters may fail when speech corrupted by high levels of background noise is coded. For example, many of the very low rate, synthetic quality vocoders used by the military fail in moving vehicles or helicopters due to the presence of periodic background noise.

1.2.5 Coding Delay

Many speech coders buffer speech into frames. For communications quality speech coders a frame length of 20-30 ms is common. This introduces delay into the system, the end to end delay (total delay of the encoding/decoding process) of the speech coder being a multiple of the frame length due to processing and transmission delays. Total coding delays of 80-120 ms are common. Delay can also be introduced by specific features of the speech coding algorithm. Several coding algorithms exist that examine speech information in future frames before coding information in the current frame, this is implemented in a causal system by introducing delay.

Delay becomes a problem for two reasons. Firstly, speech coders are often interfaced to the PSTN via four to two wire converters or "hybrids". A side effect of using these devices is that a proportion of the output signal from the codec is fed back into the input of the codec. Due to coding delays, this introduces echo. This is extremely disconcerting to the user, who hears one or more echoes of his own voice returned at multiples of 80-120 ms. The second problem with delay is when the coding delay is coupled with long transmission delays such as those encountered with transmission via satellites in geosynchronous orbit (200 ms round trip). In this case a total delay of over 300 ms may be encountered, making actual conversation difficult. Thus minimisation of coding delay is an important research aim.

1.3 Contributions

This section describes the major and minor contributions presented in this thesis. The major contributions presented in this thesis are:

1. A pitch estimation algorithm based on a square law non-linearity is presented in chapter 4, known as Non-Linear Pitch (NLP). The algorithm has moderate computational complexity, low algorithmic delay (small buffering requirements), and robustness to gross pitch errors (halving and doubling). The algorithm employs a minimum number of experimentally derived constants.

2. A generic harmonic sinusoidal coder (chapter 5). This coder has been implemented using a range of techniques from existing sinusoidal and Multi-Band Excitation (MBE) coders. In addition, new techniques are presented for the analysis, spectral magnitude modelling and synthesis stages. The unquantised coder produces output speech of very high quality, in some cases almost indistinguishable from the original speech signal.

3. In chapter 6, several parametric models for the compact representation of the harmonic phases are described. A phase model for voiced speech that consists of a minimum phase LPC synthesis filter cascaded with an all pass filter is proposed, and three candidate systems are then developed that implement this model. These models are evaluated using objective and informal subjective methods, and found to produce speech quality comparable to VSELP in clean speech conditions.

Several minor contributions are also presented:

1. A definition of current low rate speech coding issues (section 1.2).

2. A quantitative description of the procedure used to determine the excitation for Code Excited Linear Prediction (CELP) coders (section 2.5.3).

3. A derivation of expressions to obtain the harmonic sinusoidal model parameters for both sinusoidal and Multi-Band Excitation (MBE) coders (section 3.5).

4. A qualitative description of the problems encountered with existing sinusoidal and Multi-Band Excitation (MBE) analysis procedures due to the breakdown of assumptions of short-term stationarity (section 3.7).

5. A fully quantised 8.3 kbit/s sinusoidal coder that combines the pitch estimation, sinusoidal coding, and phase modelling techniques developed in this thesis.

1.4 Thesis Outline

Chapters 2 and 3 provide the conceptual and mathematical background information necessary to describe the contributions presented in subsequent chapters. Chapter 2 deals with basic speech coding principals such as the nature of speech waveforms and then proceeds to introduce several key speech coding concepts and analysis techniques. It concludes with a presentation of several popular time domain coding algorithms.

Chapter 3 presents background information on low bit rate frequency domain coding, which forms the basis for the major contributions of this thesis. The sinusoidal coder is introduced, and analysis techniques for the sinusoidal model parameters are presented in mathematical form. Several problems with the sinusoidal coder are discussed. Finally a qualitative comparison between CELP and sinusoidal coding is presented.

Chapter 4 presents a new pitch estimation algorithm known as Non Linear Pitch (NLP). This algorithm uses the harmonic distortion introduced by a square law non-linearity to determine the pitch period of speech signals. The algorithm contains three stages, a basic pitch extractor, a post processor, and a pitch refinement stage. The algorithm has been tested using objective and subjective methods and has been found to produce good results when used with the generic sinusoidal coder presented in chapter 5.

Chapter 5 presents a generic sinusoidal coder, which is a development of existing sinusoidal coders. This unquantised coder is computationally simple, and produces speech of very high quality, in some cases almost indistinguishable from the original speech signal. Analysis and synthesis techniques for this coder are presented, as well as a method for modelling the sinusoidal magnitudes using linear predictive techniques.

Parametric methods for modelling the sinusoidal phases of voiced speech are presented in chapter 6. Three separate phase modelling techniques are described and contrasted to each other in terms of subjective and objective results.

The various techniques presented in this thesis are combined in chapter 7 in the form of a fully quantised communications quality speech coder operating at 8.3 kbit/s. This harmonic sinusoidal coder employs one of the phase models presented in chapter 6, and the amplitude modelling discussed in chapter 5. Quantisation of these parameters is discussed and objective and informal subjective results presented.

Chapter 8 summarises the new work presented in this thesis, and provides several areas of interest for further work. Appendix B provides instructions on obtaining speech files via the internet which demonstrate the techniques developed for this thesis.

2. Speech Coding Techniques

The purpose of this chapter is to introduce and discuss basic speech coding principals and several complete time domain coding algorithms. This information, together with the background information on frequency domain coding presented in the next chapter provides the conceptual and mathematical framework for the rest of this thesis.

Section 2.1 introduces the source-filter model, a speech production model used as the basis of many model based and hybrid speech coding algorithms. Section 2.2 discusses several key features of speech signals that are often exploited to reduce the bit rate. In section 2.3 the linear predictive model of speech production is introduced, and the methods for extracting the linear predictive model parameters discussed. Pitch estimation is the topic of section 2.4, the problems with this challenging area of speech research are discussed and a typical algorithm presented. Section 2.5 introduces several complete time domain coding algorithms, including a derivation of the CELP codebook searching procedure. Finally, section 2.6 discusses the Line Spectrum Pair (LSP) representation for quantising and transmission of LPC coefficients.

2.1 Source-Filter Model

Speech is a time varying acoustic pressure wave. For the purposes of analysis and coding, it can be converted to electrical form and sampled. Speech signals are non-stationary; the characteristics of speech evolve over time. As the characteristics vary slowly, speech signals can be approximated as stationary over short periods (in the order of a few tens of milliseconds) [1].

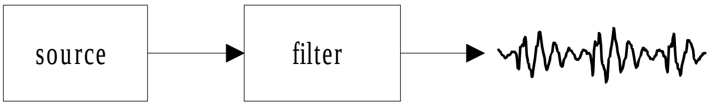


Figure 2.1: Source-Filter Model of Speech Production

A convenient model of speech production is the *source-filter* model (Figure 2.1) [1][2][3]. Speech production is modelled as a filtering operation, where a sound source excites a filter formed by the vocal tract. As the speech signal is non-stationary, the characteristics of the source and filter are time varying.

A simplified view of speech is that it consists of two types of sounds, voiced (vowels) and unvoiced (consonants). Voiced sounds are produced by air from the lungs being interrupted periodically by the vocal folds at the base of the vocal tract. Unvoiced sounds are produced by air from the lungs passing through constrictions in the vocal tract, and are usually of lower energy. The filtering operation performed by the speech organs in the vocal tract enhances some frequencies and attenuates others, depending on the position of the articulators. The articulators (vocal folds (cords), tongue, lips, teeth, velum, jaw) are moved by voluntary muscle control to form different sounds [1]. Physiologically the source and filter are not separate. However for analysis purposes this is a useful assumption.

For the purposes of speech coding, accurate modelling of the vocal tract is important in retaining the intelligibility of coded speech, while effective modelling of the excitation source produces natural sounding speech.

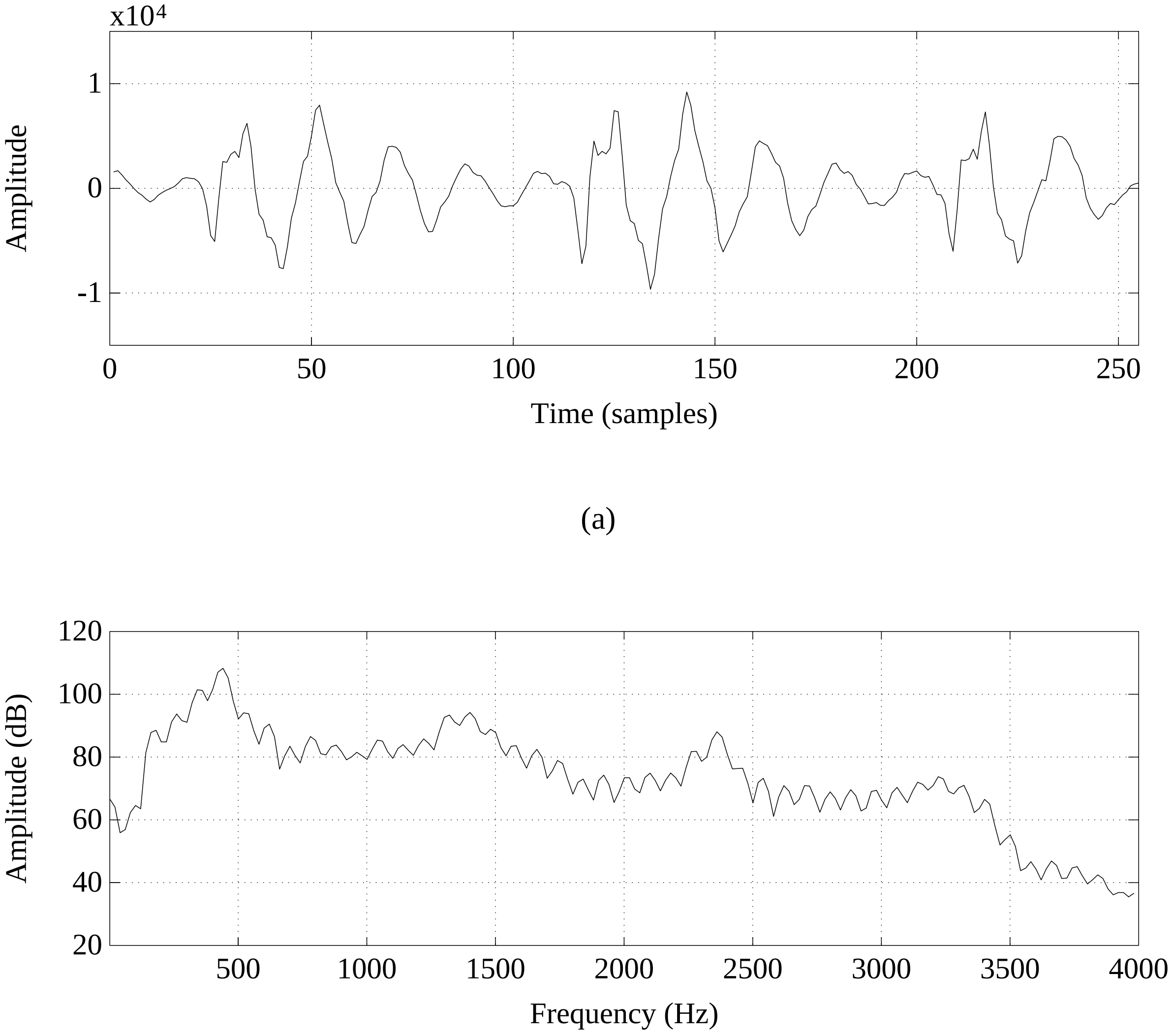
2.2 Characteristics of Speech Signals

Figure 2.2 is a time and frequency domain plot of a segment of *voiced* male speech sampled at 8 kHz. The segment is $N = 256$ samples long, and the speech was bandpass filtered between 300 Hz and 3300 Hz before sampling.

Note the periodicity in the time domain. The waveform in this example has a period of approximately $P = 90$ samples where P is known as the *pitch period* (units samples). The pitch period (or simply pitch) corresponds to the time between successive openings of the vocal folds and is related to the fundamental frequency $F0$ (units Hz) by:

$$F0 = \frac{F_s}{P} \qquad (2.1)$$

where F_s is the sampling frequency in Hz.



(a)

(b)

Figure 2.2: Male Speech Segment: (a) Time Domain, and (b) Frequency Domain

In sampled systems $F0$ is often expressed in normalised radians as ω_0 where:

$$\omega_0 = \frac{2\pi}{P} \tag{2.2}$$

In human speech $F0$ ranges from about 50-500 Hz, 50-160 Hz for males and 100-500 Hz for females and children. As the excitation signal for this example is periodic, the magnitude spectrum is also periodic and consists of harmonics of the fundamental frequency. The amplitude of the harmonic series is modulated by a slowly changing function of frequency. This is the filtering effect of the vocal tract. Note several peaks in the spectrum, at around 400, 1200 and 2300 Hz. These peaks correspond to resonances in the vocal tract known as *formants*. The number, frequency and bandwidth of the formants are time varying and change as we articulate sounds.

2.3 Linear Predictive Coding

The source-filter model was introduced in the previous section as a convenient way to model speech production. To code speech at low bit rates we require a compact and accurate representation of the excitation source and vocal tract filter. A popular method of modelling the vocal tract filtering is Linear Predictive Coding (LPC) [6]. In its most common form, this technique uses a small number of parameters in the form of an all pole filter to model the vocal tract. Parameters for a segment of speech are derived by analysing the original speech signal using the procedure described below.

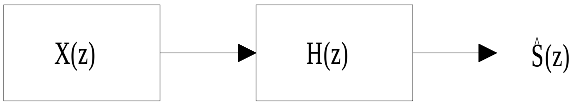


Figure 2.3: Source-Filter Model in *z*-domain

To derive the linear predictive model we consider the source-filter model in the *z*-domain, illustrated in Figure 2.3. An excitation source $X(z)$ drives a vocal tract filter $H(z)$, to generate synthetic speech $\hat{S}(z)$ (the *z*-transform of time domain synthetic speech $\hat{s}(n)$) such that:

$$\hat{S}(z) = X(z)H(z) \tag{2.3}$$

where $H(z)$ is defined as an all-pole filter of the form [1][2]:

$$H(z) = \frac{G}{1-\sum_{k=1}^{p} a_k z^{-k}} = \frac{G}{A(z)} \tag{2.4}$$

where $\{a_k\}$ for $k = 1, 2, \ldots, p$ is a set of p linear prediction coefficients, that characterise the filter's frequency response and G is a scalar gain factor. Note that other definitions of $H(z)$ exist [3][6], that have a positive sign in the denominator of (2.4):

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{2.5}$$

This yields an equivilant model except for the sign of $\{a_k\}$. The LPC model of the form presented in (2.4) is used in this thesis. The number of poles in the all pole filter is equal to the LPC order p.

Our task is to choose $\{a_k\}$ and G such that $\hat{S}(z)$ is as close as possible to the original speech signal $S(z)$. This can be achieved as follows. In the ideal case, we wish $\hat{S}(z)$ to be identical to $S(z)$, thus we substitute $S(z)$ for $\hat{S}(z)$ and re-arrange (2.3) to obtain:

$$X(z) = \frac{S(z)A(z)}{G} \tag{2.6}$$

Taking the inverse *z*-transform:

$$x(n) = \frac{1}{G}\left(s(n) - \sum_{k=1}^{p} a_k s(n-k) \right) \tag{2.7}$$

If $H(z)$is a good approximation of $S(z)$ then the energy in the signal $x(n)$ will be minimised where the total energy is given by:

$$E = \sum_{n=-\infty}^{\infty} x^2(n) = \frac{1}{G^2} \sum_{n=-\infty}^{\infty} \left(s(n) - \sum_{k=1}^{p} a_k s(n-k) \right)^2 \tag{2.8}$$

Thus $\{a_k\}$ can be found by equating (2.8) to zero and finding the partial derivatives with respect to a_i for *i*=1,2,...,*p*. From this procedure we obtain p equations in p unknowns:

$$\sum_{n=-\infty}^{\infty} s(n)s(n-i) - \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} s(n-k)s(n-i) = 0, \quad i = 1,2,\ldots,p \tag{2.9}$$

which can be expressed in terms of autocorrelation values $R(k)$:

$$R(i) - \sum_{k=1}^{p} a_k R(i-k) = 0, \quad i = 1,2,\ldots,p \tag{2.10}$$

where:

$$R(k) = \sum_{n=-\infty}^{\infty} s(n)s(n-k) \tag{2.11}$$

This is known as the *autocorrelation* method of deriving the linear prediction coefficients. Several other techniques exist for obtaining $\{a_k\}$, such as the covariance [1] and Burg [2] methods. The autocorrelation method is popular due to an efficient method of solving (2.10) known as the Levinson-Durbin algorithm [6].

To derive *G*, we need to define the excitation signal $X(z)$ in (2.3). Taking the inverse *z*-transform of (2.3):

$$\hat{s}(n) = Gx(n) + \sum_{k=1}^{p} a_k \hat{s}(n-k) \tag{2.12}$$

where $\hat{s}(n)$ and $x(n)$ are the inverse z-transforms of $\hat{S}(z)$ and $X(z)$. Now consider the case [6] where the excitation signal $x(n)$ is a unit impulse, and $\hat{h}(n)$ is the impulse response of (2.12):

$$\hat{h}(n) = G\delta(0) + \sum_{k=1}^{p} a_k \hat{h}(n-k) \tag{2.13}$$

By multiplying both sides by $\hat{h}(n-i)$ and summing over all n we obtain two expressions relating $\hat{R}(i)$, defined as the autocorrelation of $\hat{h}(n)$:

$$\hat{R}(i) = \sum_{k=1}^{p} a_k \hat{R}(i-k), \quad |i| > 1 \tag{2.14}$$

$$\hat{R}(0) = G^2 + \sum_{k=1}^{p} a_k \hat{R}(i-k) \tag{2.15}$$

It is reasonable to choose *G* to minimise the difference in the energy of $s(n)$ and $\hat{h}(n)$. Thus the total energy in $s(n)$ and $\hat{h}(n)$ must be equal:

$$\hat{R}(0) = R(0) \tag{2.16}$$

Due to (2.16) and the similarity between (2.14) and (2.10), we can conclude [6] that:

$$\hat{R}(i) = R(i), \quad i = 0,1,\ldots,p \tag{2.17}$$

Thus (2.15) can be manipulated to obtain G for the unit impulse case:

$$G = \sqrt{R(0) - \sum_{k=1}^{p} a_k R(i-k)} \tag{2.18}$$

In practice the original speech signal $s(n)$ is windowed so speech segments (typically 16-30ms long) are used to derive $\{a_k\}$. This limits the autocorrelation summation to a window of N speech samples, where N is the frame length. Due to the non-stationary nature of speech, the LPC model must be updated regularly by moving the window forward in time. Figure 2.4 shows an example of LPC modelling applied to the speech segment presented previously in Figure 2.2. The dashed line is the magnitude spectrum of the original speech segment. The solid line represents the magnitude spectrum of the $p = 10$, LPC model $\left|H\left(e^{-j\omega}\right)\right|$.

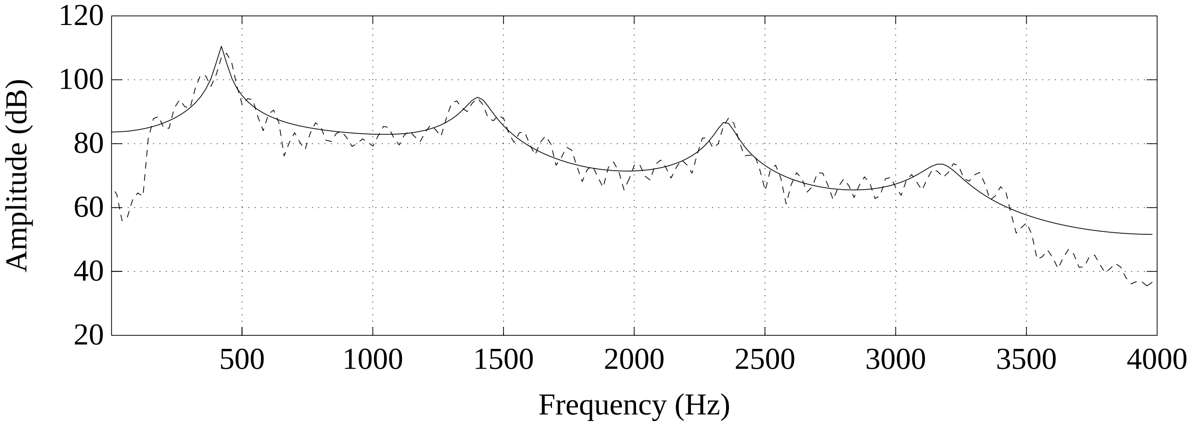


Figure 2.4: LPC Modelling of Male Speech

Linear predictive coding effectively models the vocal tract frequency response. In the example, the LPC magnitude spectrum follows the slowly varying amplitude component, or *spectral envelope*. Note that the LPC envelope falls just below the peaks of the speech spectra in high energy areas (formants), and just above the peaks in low energy areas (anti-formants). The LPC model fits the peaks better than the valleys due to the greater contribution of the peaks to the minimum Mean Square Error (MSE) criterion [1].

As the LPC model order p increases, the LPC model provides a better fit to the speech spectrum. As $p \to \infty$ the fit becomes exact [6] and the ragged excitation information is matched as well as the spectral envelope. However as the model order increases the number of parameters requiring transmission and hence the bit rate increases, thus the model order is a trade off between model accuracy and economy of bit rate. LPC is normally used only to represent the vocal tract filtering. For this purpose enough poles must be present in the model to represent the number of formants, plus a few poles extra to approximate the effects of zeros in the vocal tract. Model orders of 10-16 are common for speech coding.

The all-pole filter $1/A(z)$ is known as the *synthesis* filter. If driven with a suitable excitation signal $x(n)$, it can be used to synthesise artificial or coded speech $\hat{s}(n)$. From (2.7):

$$\hat{s}(n) = x(n) + \sum_{k=1}^{p} a_k \hat{s}(n-k) \tag{2.19}$$

The all-zero filter $A(z)$ is defined as the *analysis* filter. It can be used to remove the vocal tract information from a speech signal, leaving the *residual* $r(n)$ defined as:

$$r(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{2.20}$$

The residual is effectively the "ideal" excitation signal that is driving the LPC model of the vocal tract filter. Accurate modelling of this signal results in natural sounding speech.

2.4 Pitch Estimation

Pitch or $F0$ estimation is one of the most difficult problems in speech analysis . It is also one of the most important, as the ear is more sensitive to changes in $F0$ than any other speech parameter by an order of magnitude [4]. The basic problem of indentifying the fundamental frequency of a harmonic series may appear straight forward, however it is complicated by several factors [4][35]:

1. The common assumption of speech being stationary for short periods (10-30ms) often breaks down. For example in transition regions the speech characteristics (including pitch period) can change rapidly (e.g. Figure 2.5).

2. Effect of vocal tract filtering. Some harmonics will be amplified, other attenuated. In particular, the first formant (F1) gives rise to a large amount of periodic energy that is not related to $F0$.

3. Band limiting and phase distortion. Communications quality speech is band limited to 300-3300 Hz, thus the first few harmonics for male speakers (including the fundamental) are often lost.

4. Wide range (50-500 Hz) of possible $F0$ frequencies.

5. Simultaneous presence of voiced and unvoiced energy (mixed excitation).

6. Presence of ambient environmental noise, e.g. background noise encountered in a moving vehicle.

A pitch estimator is often comprised of 3 functional elements, a *pre-processor*, a *basic* $F0$ *extracter*, and a *post-processor* (Figure 2.6).

The pre-processor may consist of linear and/or non-linear processing. The pre-processor attempts to make the speech signal more suitable for the next stage of processing. For example, it may consist of inverse LPC filtering that removes the vocal tract information from the speech signal, leaving a spectrally "flat" signal.

The basic pitch extracter uses time or frequency domain techniques to provide one or more estimates of the pitch period, P (units samples), or fundamental frequency, $F0$ (units Hz). Many basic extracters use a short term transform applied to a frame of speech samples containing several pitch periods. The transformed data is characterised by maxima or minima that correspond to possible pitch estimates.

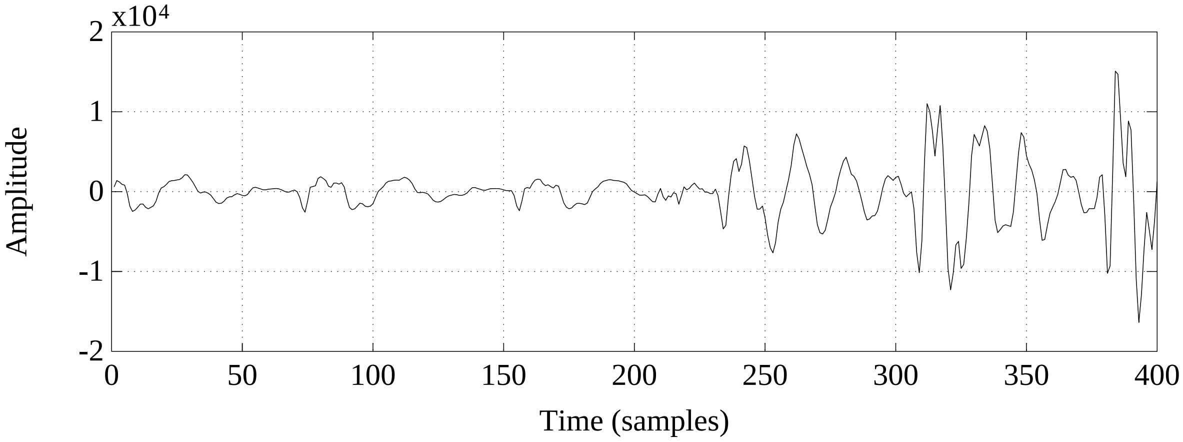


Figure 2.5: Breakdown of Short-Term Stationarity: Speech Signal in Transition Region

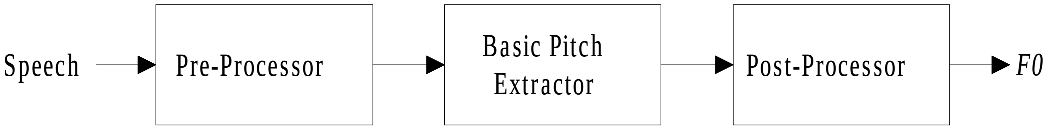


Figure 2.6: Pitch Estimator Block Diagram

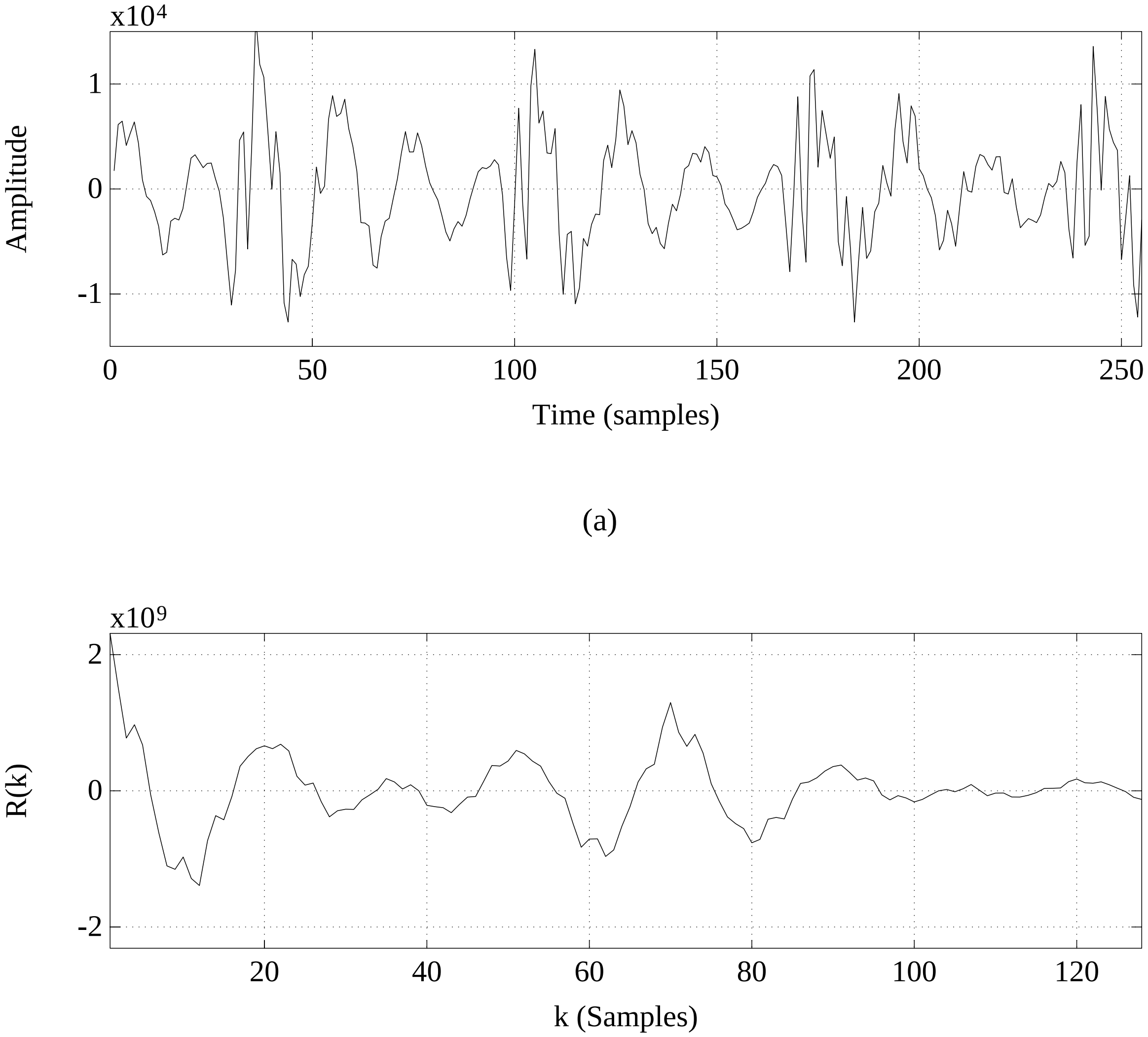
The post processor evaluates the possible pitch estimates, and determines the most likely pitch value. The post processor may "smooth" the pitch contour by comparing the current pitch estimate to past and future estimates and acting to remove any discontinuities caused by temporary failure of the basic extractor.

The windowed autocorrelation function is commonly used as a basic pitch extractor [36]:

$$R(k)=\sum_{n=-\infty}^{\infty}s(n)w(n)s(n-k)w(n-k), k=P_{\min},\ldots,P_{\max} \tag{2.21}$$

where $P_{\min}=F_s/F0_{\max}$ and $P_{\max}=F_s/F0_{\min}$. A range of pitch values from 20-160 samples corresponding to an $F0$ range of 50-400 Hz is common. The speech samples are multiplied by an N sample tapered data window, $w(n)$ to limit the range of the summation. Windows of 20-40ms ($N=160-320$ samples at 8 kHz sampling rate) are common. A Hanning window is a common choice for $w(n)$:

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right), n = 0,1,\ldots, N-1 \tag{2.22}$$



(a)

(b)

Figure 2.7: Male speech segment: (a) Time Domain, and (b) Autocorrelation Function

Figure 2.7 is an example of the autocorrelation function of a segment of male speech. Visual examination of Figure 2.7 (a) indicates a pitch period of about $P = 70$ samples. Note the corresponding peak in Figure 2.7 (b) at $k = 70$ samples.

2.5 Time Domain Coding Algorithms

LPC provides an efficient way of coding the vocal tract filter information. The next stage in the coding process is to determine an efficient method of coding the excitation for this filter.

2.5.1 LPC Vocoder

The synthesiser (decoder) for a simple LPC vocoder is illustrated in Figure 2.8. Speech is synthesised by exciting an LPC synthesis filter with either a periodic (voiced) or white noise (unvoiced) source. The periodic source consists of impulses spaced by the pitch period. Both the periodic and noise sources are scaled by an appropriate gain.

The speech encoder determines the LPC filter coefficients, the pitch, and a single voiced/unvoiced decision for each frame. These parameters are quantised and sent to the decoder. This type of vocoder is capable of sending intelligable speech at bit rates of 2400 bit/s and below.

The main drawback is that the synthesised speech has a mechanical quality, due to the simple excitation model. The LPC vocoder assumes speech to be either voiced or unvoiced. In practice speech often contains both voiced and unvoiced energy, which cannot be adequately modelled by this coder.

The LPC vocoder requires accurate estimation of the excitation model parameters, such as pitch and the voiced/unvoiced decision for each frame. This is a difficult task, which is futher complicated when acoustic background noise is present.

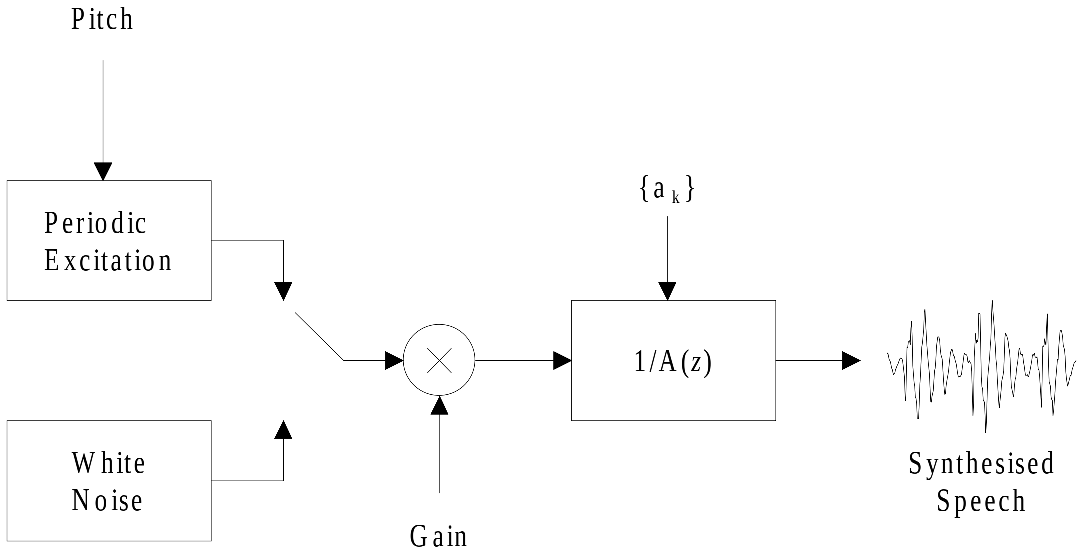


Figure 2.8: LPC Vocoder

2.5.2 Multipulse-Excited LPC

Multipulse coders [26] model the residual, $r(n)$, using a series of pulses. The positions and amplitudes of the pulses are chosen to minimise the error between the original and synthesised speech over the current analysis frame (typically 5ms long). Figure 2.8 illustrates the multipulse analysis loop.

To determine a pulse location and amplitude the excitation generator produces an excitation sequence for each possible pulse location in the analysis frame. These candidate excitations are passed through the synthesis filter, and the MSE between the synthesised and original speech measured. The optimum pulse amplitude is obtained by minimising the MSE at each candidate pulse position. The candidate position and amplitude that minimises the MSE is chosen, and the procedure is repeated for the desired number of pulses.

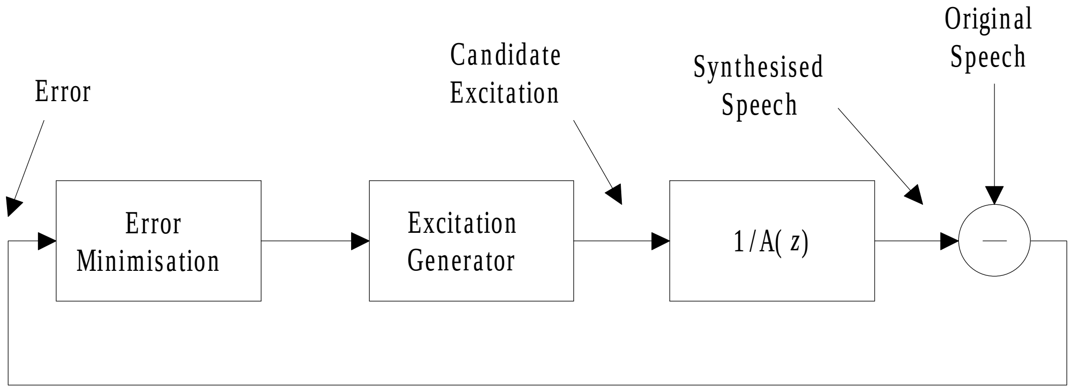


Figure 2.9: Multipulse-Excitation Encoder

This technique is a form of *analysis by synthesis* or *closed-loop* coding, as the canditate excitation signals are synthesised as part of the analysis procedure.

The pulse locations and amplitudes for successive pulses are found iteratively to reduce complexity. After the optimum position and amplitude of pulse *n* has been chosen, the synthesised speech from this pulse is subtracted from the original speech. The result of the subtraction is then used as the original speech for determing pulse $n+1$.

Multipulse-Excitation requires no pitch or voicing detectors, which tends to make it more robust to different speakers and acoustic background noise conditions than the LPC vocoder.

Multipulse coders can produce communications quality speech at bit rates of around 10 kbit/s. Typically around 4-8 pulses per 5ms analysis frame are required for communications quality speech. At bit rates below 10 kbit/s, not enough bits are available for the number of pulses required to produce an adequate excitation signal.

A low complexity development of Multipulse-Excitation is Regular Pulse Excitation (RPE) [10]. This coder only optimises the position of the first pulse in each analysis frame, the rest are regularly spaced. The amplitudes of each pulse are individually chosen to minimise the MSE in a similar fashion to multipulse coders.

2.5.3 Code Excited Linear Prediction

One of the most popular methods for communications quality speech coding is Code Excited Linear Prediction (CELP) [11]. This coder uses the sum of one or more vector *codebooks* to quantise the excitation. An important feature of this coder is that the vector quantisers or codebooks are searched using an *analysis by synthesis* procedure. CELP is capable of coding communications quality speech at bit rates of 4-8kbit/s. To discuss the operation of CELP it is useful to first describe the CELP decoder (Figure 2.10).

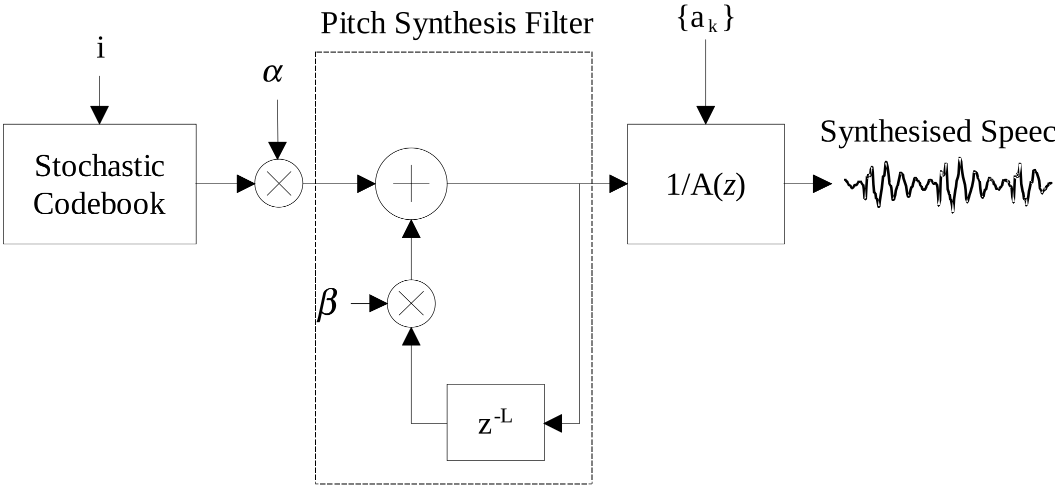


Figure 2.10: CELP Decoder

In Figure 2.10 a *stochastic codebook* excites the cascade of a *pitch synthesis* and LPC synthesis filter. The stochastic codebook consists of N_s vectors, each vector containing N_{SF} samples. Each vector is populated with a fixed sequence of randomly distributed numbers, each vector (codebook entry) has a different sequence. The stochastic codebook contribution is defined by the entry (i) and the gain (α).

The pitch synthesis filter models the long term periodicity present in the excitation signal during voiced speech. This is acheived by modelling the current excitation sample as a weighted version of a previous excitation sample. The pitch synthesis filter contribution is defined by the pitch delay or lag (L), and the pitch gain, (β).

Both the stochastic codebook and pitch synthesis filter parameters are time varying. They are updated every *subframe* of N_{SF} samples, where N_{SF} is usually a submultiple of the LPC frame size, N (e.g. $N = 4N_{SF}$). In between updates the excitation parameters remain fixed.

Thus the LPC synthesis filter models the short term periodicity (vocal tract filtering), the pitch synthesis filter models the long term periodicity (pitch structure), and the stochastic codebook models the random component (remaining modelling errors and unvoiced speech).

The pitch synthesis filter can be viewed as an *adaptive codebook* consisting of N_a (N_{SF} sample) vectors. Thus for each possible value of L, we have a codebook entry of N_{SF} samples. Note that adjacent entries (entry L and $L+1$) are identical except for the first and last samples. This representation of CELP is useful for explaining the methods used to determine the excitation parameters and is illustrated in Figure 2.11

The adaptive codebook contents are time varying, and are updated every subframe from the composite (sum of adaptive and stochastic) excitation. Due to this feature the adaptive codebook tends to build up a good approximation of the ideal excitation waveform over time, especially in continuous segments of voiced speech.

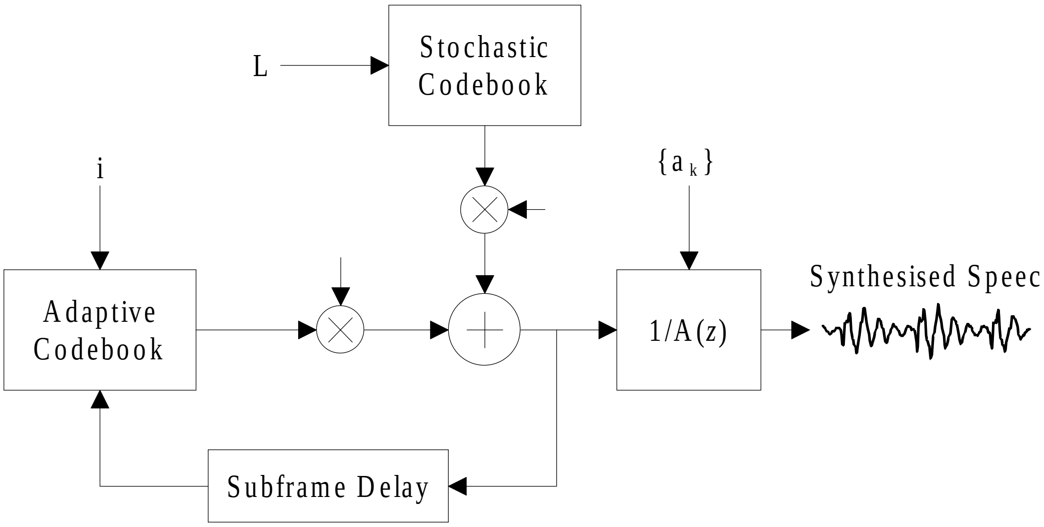


Figure 2.11: CELP Decoder with Adaptive Codebook

The CELP encoder exhaustively searches the adaptive and stochastic codebooks to determine the optimum excitation vectors and gains. The synthesis filter response to each possible excitation vector is determined. This is then compared to the target (original) speech vector in a mean-square error (MSE) sense. The parameters that represent the excitation that minimises the MSE for the current subframe are transmitted to the decoder.

The two codebooks are usually searched sequentially. First, the optimum excitation vector (L) and gain (β) for the adaptive codebook is determined. The stochastic codebook entry (i) and gain (α) are then chosen. A simultaneous estimation of both adaptive and stochastic codebook parameters is possible, however the large increase in complexity does not warrant the slight performance increase obtained.

An analytical formulation of the CELP codebook search procedure developed by the author is now presented. The procedure is well known, and described qualitatively in many sources. However, a literature search by the author has found no equivilent analytical presentation of the entire search procedure.

The composite excitation signal $x(n)$ can be described by:

$$x(n) = \beta x_a^{(L)}(n) + \alpha x_s^{(i)}(n), \quad n = 0,1,2,\ldots,N_{SF} - 1 \tag{2.23}$$

where $x_a^{(L)}(n)$ is the n^{th} sample of adaptive codebook entry *L*, and $x_s^{(i)}(n)$ is the n^{th} sample of stochastic codebook entry *i*. As the adaptive codebook consists of previous excitation samples, (2.23) can be expressed as:

$$x(n) = \beta x(n - L) + \alpha x_s^{(i)}(n), \quad n = 0,1,2,\ldots,N_{SF} - 1 \tag{2.24}$$

The aim of the analysis by synthesis codebook search is to choose the excitation parameters $\{L, \beta, i, \alpha\}$ such that the Mean-Square Error (MSE), *E*, between the original speech and synthesised speech for this subframe is minimised:

$$E = \sum_{n=0}^{N_{SF}-1} e^2(n) = \sum_{n=0}^{N_{SF}-1} \left(s(n) - \hat{s}(n)\right)^2 \tag{2.25}$$

The synthesised speech is obtained from the LPC synthesis filter $1/A(z)$:

$$\hat{s}(n) = x(n) + \sum_{k=1}^{p} a_k \hat{s}(n - k) \tag{2.26}$$

As this is a causal Infinite Impulse Response (IIR) filter, it may be expressed as a convolution of the current and all prior input samples $x(k), \ k = -\infty,\ldots,n-1,n$ with the impulse response of the synthesis filter $h(k)$:

$$\hat{s}(n) = \sum_{k=-\infty}^{n} x(k)h(n-k) \tag{2.27}$$

Equation (2.27) can be expressed in terms of the response to the excitation signal before the current subframe, and from the current subframe:

$$\hat{s}(n) = \sum_{k=-\infty}^{-1} x(k)h(n-k) + \sum_{k=0}^{n} x(k)h(n-k) \tag{2.28}$$

The first term of (2.28) is known as the *zero input* response, $\hat{s}_{zi}(n)$. This is the component of the synthesis filter response from the input samples *before* the start of the current subframe. The zero input reponse can be obtained using the synthesis filter defined in (2.26) with $x(n) = 0$:

$$\hat{s}_{zi}(n) = \sum_{k=1}^{p} a_k \hat{s}_{zi}(n-k), n = 0,1,\ldots,N_{SF} \tag{2.29}$$

The samples $\hat{s}_{zi}(n) = \hat{s}(n)$ for $n = -p, -p+1, \ldots, -1$ are the last p synthesised speech samples of the previous subframe. As (2.26) has an infinite impulse response these samples are non-zero, therefore the zero input response will be non-zero.

The second term of (2.28) is known as the *zero state* response, $\hat{s}_{zs}(n)$, of the synthesis filter. This part of the synthesis filter repsonse is due to input samples from the current subframe. It can be obtained from (2.26) by setting the state variables $\hat{s}_{zs}(n) = 0$ for $n = -p, -p+1, \ldots, -1$, and evaluating:

$$\hat{s}_{zs}(n) = \sum_{k=1}^{p} a_k \hat{s}_{zs}(n-k), n = 0,1,\ldots,N_{SF} \tag{2.30}$$

The *zero state* response can also be obtained by directly computing the second term of (2.28).

The first term of (2.28) is the contribution to the current output sample from input samples before the start of the current subframe. It is *not* a function of the excitation parameters for the *current* subframe $\{L, \beta, i, \alpha\}$. The second term of (2.28) *is* a function of $\{L, \beta, i, \alpha\}$, this is known as the *target* vector. We wish to choose the excitation parameters such that the synthesised speech closely matches the target. To isolate the target, we subtract the first term of (2.28) from the original speech for this subframe.

Re-formulating (2.25):

$$E = \sum_{n=0}^{N_{SF}-1} \left(\left(s(n) - \hat{s}_{zi}(n) \right) - \hat{s}_{zs}(n) \right)^2 = \sum_{n=0}^{N_{SF}-1} \left(s_s(n) - \hat{s}_{zs}(n) \right)^2 \qquad (2.31)$$

The problem is now choosing $\{L, \beta, i, \alpha\}$ to minimise *E*, the MSE between the *target* speech $s_s(n)$ (original speech with zero input synthesis filter response removed), and $\hat{s}_{zs}(n)$. Substituting (2.23) into the 2nd term of (2.28) we obtain an expression for the zero state response of the synthesis filter for this subframe:

$$\hat{s}_{zs}(n) = \beta \sum_{k=0}^{n} x_a^{(L)}(k) h(n-k) + \alpha \sum_{k=0}^{n} x_s^{(i)}(k) h(n-k) \qquad (2.32)$$

$$\hat{s}_{zs}(n) = \beta \hat{s}_a^{(L)}(n) + \alpha \hat{s}_s^{(i)}(n) \qquad (2.33)$$

where $\hat{s}_a^{(L)}(n)$ is the unit-gain, zero state response from the adaptive codebook excitation and $\hat{s}_s^{(i)}(n)$ is the unit-gain, zero state response from the stochastic codebook excitation.

The excitation parameters for each codebook $\{L, \beta, i, \alpha\}$ can be found sequentially, first choosing $\{L, \beta\}$ to minimise:

$$E_a^{(L)} = \sum_{n=0}^{N_{SF}-1} \left(s_s(n) - \beta \hat{s}_a^{(L)}(n) \right)^2 \qquad (2.34)$$

then choosing $\{i, \alpha\}$to minimise:

$$E_s^{(i)} = \sum_{n=0}^{N_{SF}-1} \left(\left(s_s(n) - \beta \hat{s}_a^{(L)}(n) \right) - \alpha \hat{s}_s^{(i)}(n) \right)^2 \qquad (2.35)$$

By finding the derivative of (2.34) with respect to β, and equating to zero, an expression for the optimum gain β and the resulting MSE may be obtained:

$$\beta = \frac{\sum_{n=0}^{N_{SF}-1} s_s(n)\hat{s}_a^{(L)}(n)}{\sum_{n=0}^{N_{SF}-1}\left(\hat{s}_a^{(L)}(n)\right)^2} \tag{2.36}$$

$$E_a^{(L)} = \sum_{n=0}^{N_{SF}-1}\left(s_s(n)\right)^2 - \beta\sum_{n=0}^{N_{SF}-1} s_s(n)\hat{s}_a^{(L)}(n) \tag{2.37}$$

Similar results may be obtained for $\{i,\alpha\}$ from (2.35):

$$\alpha = \frac{\sum_{n=0}^{N_{SF}-1}\left(s_s(n) - \beta\hat{s}_a^{(L)}(n)\right)\hat{s}_s^{(i)}(n)}{\sum_{n=0}^{N_{SF}-1}\left(\hat{s}_s^{(i)}(n)\right)^2} \tag{2.38}$$

$$E_s^{(i)} = \sum_{n=0}^{N_{SF}-1}\left(s_s(n) - \beta\hat{s}_a^{(L)}(n)\right)^2 - \alpha\sum_{n=0}^{N_{SF}-1}\left(s_s(n) - \beta\hat{s}_a^{(L)}(n)\right)\hat{s}_s^{(i)}(n) \tag{2.39}$$

The codebooks are searched by determining the response to each excitation vector, then determining the optimum gain. The resulting MSE for each codebook entry is determined and stored. After all codebook entries have been evaluated, the entry and corresponding gain resulting in the minimum MSE is chosen.

A block diagram of the CELP encoder codebook searching procedure is presented in Figure 2.12. Only one codebook is drawn, as the search procedures are similar for both codebooks. After removing the zero input response from the input speech vector, the codebook is exhaustively searched to find the optimum excitation vector and gain. This is determined by synthesising the response to each vector, then comparing it with the original speech on a MSE basis.

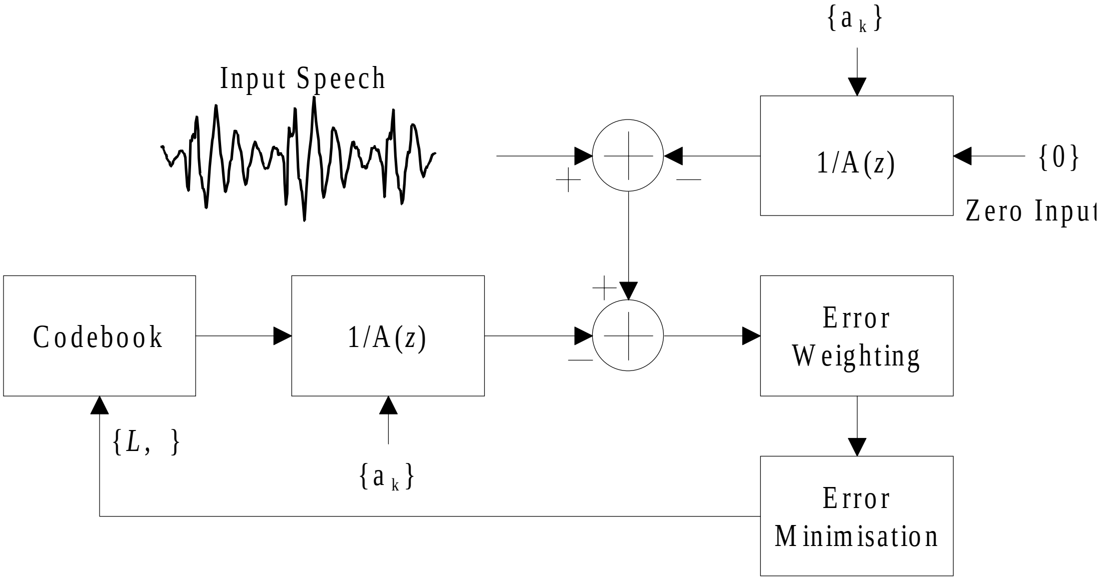


Figure 2.12: CELP Encoder

Before the error minimisation block the error signal is passed through an error weighting filter. This filter ensures that certain parts of the spectrum are given more importance, according to a perceptual criterion. The error weighting filter is of the form:

$$W(z) = \frac{A(z)}{A(z/\gamma)} \tag{2.40}$$

which tends to attenuate the error signal in the formants and enhance the error signal in the anti-formant regions. Figure 2.13 is an example of an error weighting filter response with $\gamma = 0.9$. The corresponding synthesis filter frequency response is the dashed line. The filtered error signal is used to determine the MSE for the current codebook entry. Less emphasis is placed on matching the speech signal in the formant regions than in the anti-formant regions. This can be interpreted perceptually in terms of the masking effect of the human ear. A poorer match of the signal in the formants will result in coding noise in this region. However, this coding noise will be *masked* by the relatively high signal energy in the formant. In the anti-formant region, the signal level is low, thus any error signal will be highly audible to the listener. Thus we weight the error measure to ensure a better match in the anti-formant regions.

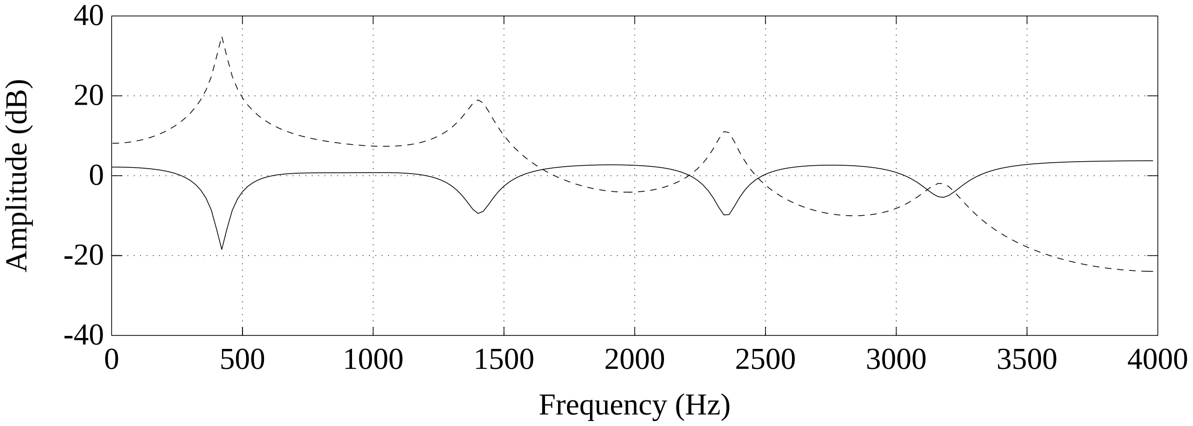


Figure 2.13: Error Weighting Filter Response

In practice the effect of the error weighting filter can be combined with the synthesis filter to simplify implementation. The codebook searching procedure defined above remains the same, except that a weighted synthesis filter impulse response is substituted for $h(n)$.

2.6 Quantisation of LPC parameters using LSPs

Section 2.3 discussed the modeling of the vocal tract filter using LPC techniques, where a set of linear prediction coefficients $\{a_k\}$ for $k = 1, 2, \ldots, p$ describes the vocal tract filter. To be useful in low bit rate speech coding it is necessary to quantise and transmit the LPC coefficients using a small number of bits. Direct quantisation of these LPC coefficients is inappropriate due to their large dynamic range (8-10 bits/coefficient [6]). Also, there is no direct way of ensuring synthesis filter stability, which is perceptually important to the synthesised speech quality. Thus for transmission purposes, especially at low bit rates, other forms such as the *Line Spectral Pair* (LSP) frequencies [57] are used to represent the LPC parameters.

The LSP coefficients represent the LPC model in the frequency domain, and lend themselves to a robust and efficient quantisation of the LPC parameters [58]. The prediction filter is an all-zero (analysis) filter, where the zeroes correspond to the poles of the all-pole (synthesis) filter.

The LSP frequencies can be derived by decomposing the p^{th} order polynomial $A(z)$, into symmetric and anti-symmetric polynomials $P(z)$ and $Q(z)$:

$$P(z) = A^{+}_{p+1}(z) = A_p(z) + z^{-(p+1)} A_p(z^{-1}) \tag{2.41}$$

$$Q(z) = A^{-}_{p+1}(z) = A_p(z) - z^{-(p+1)} A_p(z^{-1}) \tag{2.42}$$

where:

$$A_p(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \tag{2.43}$$

An important property of the polynomials is that the roots appear in complex conjugate pairs on the unit circle [57]. This is shown below where $P(z)$ and $Q(z)$ are expressed in factored form [57]:

$$P(z) = (1 + z^{-1}) \prod_{i=1}^{p/2} (1 - 2\cos(\omega_{2i-1}) z^{-1} + z^{-2}) \tag{2.44}$$

$$Q(z) = (1 - z^{-1}) \prod_{i=1}^{p/2} (1 - 2\cos(\omega_{2i}) z^{-1} + z^{-2}) \tag{2.45}$$

where ω_{2i-1} and ω_{2i} are the LSP frequencies, found by evaluating the polynomials on the unit circle. Figure 2.14 illustrates possible root locations for even order $P(z)$ and $Q(z)$ polynomials, where the roots at location 0 and π are left out for simplicity.

Other characteristics of the LSP frequencies include being interlaced with each other, where $0 < \omega_1 < \omega_2 <, \ldots, < \omega_p < \pi$. The stability of $A(z)$ is preserved after quantisation of $P(z)$ and $Q(z)$, as long as the zeroes of the LSP polynomials are on the unit circle and are interlaced. The separation of adjacent LSP frequencies is related to the bandwidth of spectral nulls in $A(z)$ (or spectral peaks in the synthesis filter $1/A(z)$). A small separation indicates a narrow bandwidth.

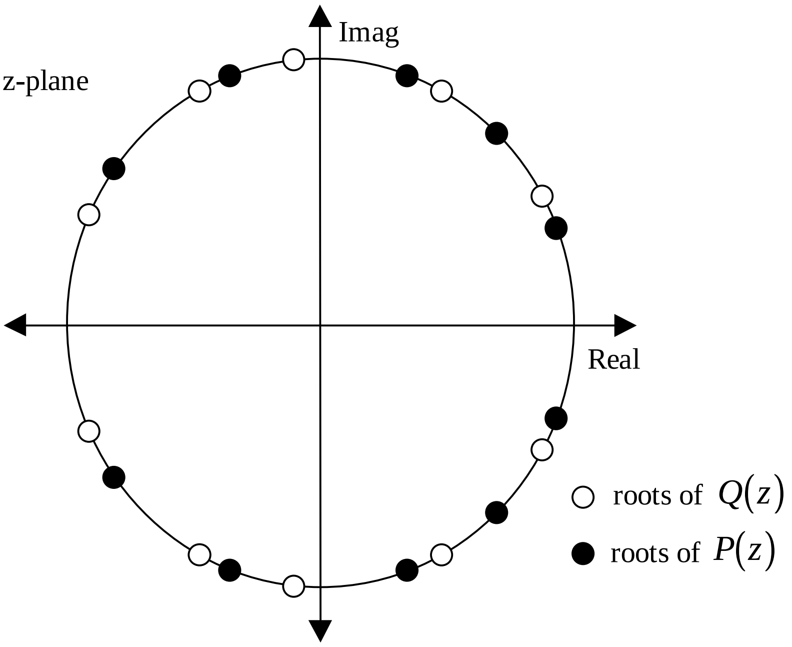


Figure 2.14: Possible root locations for an even order of $P(z)$ and $Q(z)$.

The LPC analysis filter, $A(z)$, may be reconstructed using the $P(z)$ and $Q(z)$ polynomials:

$$A(z) = \frac{P(z) + Q(z)}{2} \tag{2.46}$$

Thus to transmit the LPC coefficients using LSPs, we first transform the LPC model $A(z)$ to $P(z)$ and $Q(z)$ polynomial form. We then solve $P(z)$ and $Q(z)$ for $z = e^{j\omega}$ to obtain p LSP frequencies $\{\omega_i\}$. The LSP frequencies are then quantised and transmitted over the channel. At the receiver the quantised LSPs are then used to reconstruct an approximation of $A(z)$. Chapter 7 describes the use of LSPs to quantise the generic sinusoidal coder described in this thesis.

3. Frequency Domain Coding Techniques

This chapter introduces the concepts involved in frequency domain speech coding, in contrast to the previous chapter which introduced general speech coding concepts and discussed several time domain coding algorithms. This information is used to generate a framework for the thesis contributions presented in later chapters.

Section 3.1 introduces the concept of sinusoidal coding. In section 3.2, sinusoidal coding techniques are extended to harmonic coding. Voicing models for sinusoidal coders are introduced in section 3.3, where mixed excitation is discussed. Section 3.4 presents a mathematical treatment for the estimation of harmonic sinusoidal model parameters, while sections 3.5 and 3.6 discuss existing pitch and voicing estimation algorithms. Several problems with sinusoidal coders are presented in section 3.7, while section 3.8 presents a comparison of time and frequency domain coding algorithms. Methods of measuring speech quality are presented in section 3.9, and section 3.10 concludes the chapter.

3.1 Sinusoidal Coding

The previous chapter discussed coding algorithms that used time domain analysis to model speech in terms of the source-filter model. The object of these coding algorithms was to determine a suitable excitation sequence for the LPC synthesis filter. The concept of analysis by synthesis was introduced, where the encoder synthesises the candidate excitation sequences to determine the best excitation sequence in a weighted minimum MSE sense.

It is also possible to represent speech signals using frequency domain models [7][8]. Consider a segment of voiced speech 10-30ms long, such that the characteristics can be considered stationary. In terms of the source/filter model, this segment can be viewed as a time domain impulse train (excitation) convolved with the impulse response of the vocal tract filter. In the frequency domain this corresponds to a frequency domain impulse train (excitation) multiplied by the frequency response of the vocal tract filter. The spacing of the frequency domain impulse train is the fundamental frequency, $F0$, of the speech.

Sinusoidal coders [18][27] represent speech as the sum of a bank of sinusoidal oscillators:

$$s(n) = \sum_{m=1}^{L} B_m \cos(\omega_m n + \theta_m) \tag{3.1}$$

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

As the speech characteristics are non-stationary, the model parameters must be updated at regular intervals. Parameter update intervals (frames lengths) of 10-30ms are common. A limit, $L_{\max}$, is placed on the number of possible peaks. In frames with more than $L_{\max}$ peaks, only peaks above a certain magnitude threshold are included in the model. Note that L is time varying, as the number of peaks from frame to frame varies.

To synthesise speech using the sinusoidal model, the decoder generates L sine waves of the estimated magnitude, frequency, and phase. However, care must be taken to ensure continuity of the sinusoids at frame boundaries. This is achieved by slight adjustment of the model parameters to ensure smooth evolution of the synthesised speech signal across frame boundaries.

The sinusoidal coder can be described as a parametric coder, as it describes the speech signal using a set of model parameters. Unlike waveform and hybrid coders such as CELP, no attempt is made to reproduce the original speech waveform exactly[1]. Instead, the validity of the model assumptions are relied upon to produce good quality synthesised speech.

The sinusoidal model is capable of representing both voiced and unvoiced speech. Peak picking the short term DFT magnitude spectrum of unvoiced speech will produce model parameters that tend to be randomly distributed. For example, the frequencies of the peaks will not be related as for voiced speech which can be modelled as a harmonic series.

[1] Due to the error weighting filter in the analysis by synthesis loop, CELP actually attempts to match the weighted speech waveform, *not* the original speech waveform.

The authors of [18] report that for frame lengths of less than 10ms, the speech signal reconstructed by the sinusoidal model is perceptually indistinguishable from the original.

3.2 Harmonic Coding

One of the problems with the sinusoidal coder is the time varying number of parameters, which after quantisation leads to a non-uniform bit rate. During voiced speech, the sinusoid frequencies $\{\omega_m\}$ will be multiples of the fundamental. Thus $\{\omega_m\}$ can be efficiently modelled as multiples of the fundamental frequency for the current frame:

$$\omega_m = m\omega_0 \tag{3.2}$$

Only the fundamental $\omega_0 = 2\pi F0/F_s$ is transmitted, $\{\omega_m\}$ are then determined as harmonics of the fundamental. The number of harmonics, L, can also be determined from ω_0:

$$L = \left\lfloor \frac{\pi}{\omega_0} \right\rfloor \tag{3.3}$$

Thus (3.1) becomes the *harmonic* [27][21][28] sinusoidal model:

$$s(n) = \sum_{m=1}^{L} B_m \cos(\omega_0 mn + \theta_m) \tag{3.4}$$

This is a reasonable approximation for voiced speech, however for unvoiced speech problems arise. In some cases modelling unvoiced speech with a harmonic model produces a periodic component in the unvoiced synthesised speech. This is a poor model of unvoiced speech which is usually aperiodic noise. Thus care must be taken when using the harmonic model to represent unvoiced speech. It is shown in later chapters that by suitable selection of the harmonic phases, unvoiced speech can be faithfully represented using the harmonic sinusoidal model.

3.3 Mixed Excitation

The classic source-filter model assumes the excitation signal is either voiced or unvoiced. This leads to coding models that synthesise either voiced *or* unvoiced speech. This model is adequate for representing many speech signals, however in practice there is often a mixture of periodic and aperiodic energy. Thus there exists a need for a more general model that is capable of representing voiced, unvoiced and partially voiced speech.

When producing voiced speech, a partial constriction of the vocal tract can produce turbulence that introduces unvoiced energy into the speech signal. The sounds resulting from the partial constriction are known as *voiced fricatives*. For example consider the voiced dental phoneme δ (e.g. "th" in "these"). This is produced by creating a constriction near the end of the vocal tract (tongue touching teeth).

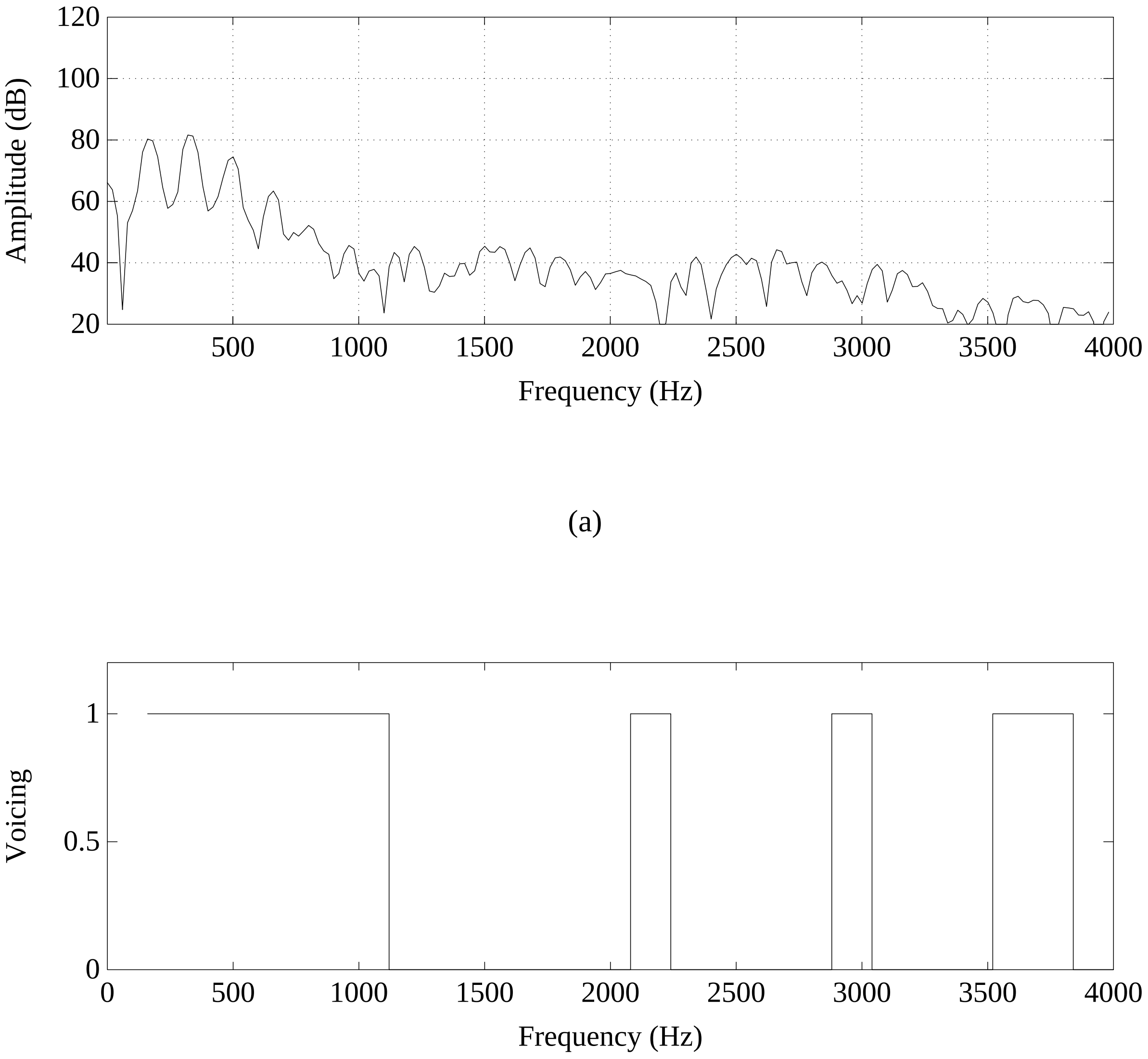
Figure 3.1(a) is the magnitude spectrum of δ from a female speaker. Note the uniformly spaced harmonics below 1000 Hz, indicating periodic or voiced energy. Above 1000 Hz, the distribution of the energy appears more random, indicating the presence of unvoiced energy.

The simple voiced/unvoiced excitation models fail to faithfully reproduce partially voiced sounds. In addition, non-speech inputs such as background noise are often poorly modelled by these coders, leading to unpleasant perceptual effects at the coder output.

To alleviate these problems, the *mixed excitation* voicing model was proposed [31]. Coders employing mixed excitation relax the voiced/unvoiced classification to provide excitation signals containing both periodic and aperiodic energy. Usually, the spectrum is split into several regions or bands. A separate voicing decision is then made for each band and transmitted to the decoder.

A mixed excitation vocoder that has gained recent prominence is the Multi-Band Excitation (MBE) coder [28], which divides the spectrum up into L bands of width $F0$. Thus for voiced speech, one harmonic will be present in each band. A voiced/unvoiced decision is then made for each band, based on the type of energy (periodic or aperiodic) the band contains.

Figure 3.1(b) illustrates the MBE model voicing decisions for the voiced fricative example of Figure 3.1(a). This is an example of applying the MBE model to partially voiced speech. For fully voiced speech, each band will contain periodic energy, therefore the entire spectrum would be declared voiced. Conversely, unvoiced speech consists of aperiodic energy, thus the entire spectrum would be classified as unvoiced. Therefore the MBE model is capable of representing voiced, unvoiced, and partially voiced sounds.



(a)

(b)

Figure 3.1: Voiced Fricative δ, (a) Magnitude Spectrum, and (b) MBE Voicing Decisions

Note the clustering of the voiced harmonics at the low frequency end of the spectrum. This leads to a more economical voicing representation for reasons described below.

Partially voiced sounds are formed by the turbulence created by partial constrictions in the vocal tract during voiced speech. Usually, the constrictions are formed near the end of the vocal tract. For example, the voiced dental phoneme discussed above is formed by the tongue touching the teeth. The resulting unvoiced energy is therefore only filtered by a short length of the vocal tract. However the voiced component of the excitation is always produced at the base of the vocal tract. Sounds that excite a shorter cavity lead to higher frequency energy than those that excite a longer cavity [1]. A musical analog is a pipe organ; lower frequency notes are produced by longer pipes than higher frequency notes.

The high frequency unvoiced energy tends to be of lower energy than the low frequency voiced energy due to the combined effect of lip radiation (+6dB per octave), and the high frequency roll off of the glottal waveform (-12dB per octave). The net effect is an attenuation of -6dB per octave over the entire speech spectrum, thus attenuating the high frequency unvoiced energy with respect to the low frequency voiced energy.

Thus for partially voiced (mixed excitation) speech signals the unvoiced energy tends to be confined to the higher frequencies, while voiced energy is present at the lower frequencies. The above physiological argument leads to the justification of a *two-band* mixed excitation model. A transition frequency, ω_t, is defined. The two-band model uses voiced (periodic) excitation beneath ω_t and unvoiced (aperiodic) excitation above ω_t. Such a model has been proposed and implemented by several authors in time [32][33][34] and frequency domain [21][29] coders.

Using Figure 3.1(a) as an example of partially voiced speech, the transition frequency, ω_t, would be located at 1000 Hz. In cases of fully voiced speech, ω_t would be situated at a high frequency (4000 Hz). Conversely, for fully unvoiced speech, ω_t would be positioned at 0 Hz.

3.4 Harmonic Magnitude and Phase Estimation

This section presents an approach for estimating the harmonic magnitudes and phases that derives results previously presented for sinusoidal [18] and MBE [28] coders. To the authors best knowledge this derivation has not been presented elsewhere, however a different derivation for the sinusoidal case was presented in [18].

For the purposes of speech analysis the time domain speech signal $s(n)$ is divided into overlapping analysis windows (frames) of N_w samples. The centre of each analysis window is separated by N samples. To analyse the l^{th} frame it is convenient to convert the fixed time reference to a sliding time reference centred on the current analysis window:

$$s_w^l(n) = s(lN+n)w(n), n = N_{wl}, \dots, N_{wu} \tag{3.5}$$

where $w(n)$ is a tapered even window of N_w (N_w odd) samples , $N_{wl} = -\left\lfloor \frac{N_w}{2} \right\rfloor$ is the lower limit of the window, and $N_{wu} = \left\lfloor \frac{N_w}{2} \right\rfloor$ is the upper limit of the window. A suitable window function is a shifted Hanning window:

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi(n - N_{wl})}{N_w - 1}\right), n = N_{wl}, \dots, N_{wu} \tag{3.6}$$

To analyse $s_w^l(n)$ in the frequency domain the N_{dft} ($N_{dft} > N_w$, N_{dft} even) point Discrete Fourier Transform (DFT) of $s_w^l(n)$ can be computed:

$$S_w^l(k) = \sum_{n=N_{wl}}^{N_{wu}} s_w^l(n) e^{-j\left(2\pi/N_{dft}\right)kn}, k = 0,1,\dots,\frac{N_{dft}}{2} \tag{3.7}$$

As $s_w^l(n)$ is real, $S_w^l(k)$ for $k = 0,1,\dots,\frac{N_{dft}}{2}$ is sufficient to represent the signal $s_w^l(n)$ in the frequency domain.

From the frequency domain speech signal, $S_w^l(k)$, we wish to obtain estimates of the harmonic model parameters for the l^{th} frame; $\{A_m^l\}$ for $m = 1,2,\dots,L$, and ω_0^l, where $\{A_m^l\}$ are defined as the complex sinusoidal amplitudes:

$$A_m^l = B_m^l \exp\left(j\theta_m^l\right) \tag{3.8}$$

Consider a voiced speech signal centred on the current analysis frame. This can be represented using the harmonic sinusoidal model:

$$s^l(n) = \sum_{m=1}^{L} B_m^l \cos(\omega_0^l m(n - lN) + \theta_m^l) \tag{3.9}$$

Consider the DFT of $s^l(lN+n), n = N_{wl}, \ldots, N_{wu}$; $S^l(k)$, and the DFT of $w(n), n = N_{wl}, \ldots, N_{wu}$; $W(k)$. If (3.9) is substituted into the $s(n)$ term of (3.5), the DFT computed with (3.7) will consist of the convolution of $S^l(k)$ with $W(k)$:

$$S_w^l(k) = S^l(k) * W(k) \tag{3.10}$$

where the $*$ operator denotes (circular) convolution. The sequence $S^l(k)$ is a weighted frequency domain impulse train:

$$S^l(k) = \frac{N_{dft}}{2} \sum_{m=1}^{L} A_m^l \delta\left(k - m\frac{\omega_0^l N_{dft}}{2\pi}\right), \quad k = 0,1,\ldots,\frac{N_{dft}}{2} \tag{3.11}$$

The impulses are weighted by A_m^l and spaced by $\frac{\omega_0^l N_{dft}}{2\pi}$ samples. In (3.11) the DFT index $k - m\frac{\omega_0^l N_{dft}}{2\pi}$ is rounded to the nearest integer, this convention applies to all DFT indexes in this thesis.

As $w(n)$ is an even real sequence, $W(k)$ is even and real. In most cases (except for very low ω_0^l speakers), the spacing between the impulses in (3.11) is large compared to the "width" of $W(k)$. Consider the amplitude of $W(k)$ at a distance of one harmonic spacing either side of the centre of the window, i.e. $|k| = \frac{\omega_0^l N_{dft}}{2\pi}$. The magnitude of $W(k)$ is small compared to the magnitude at the centre of the window:

$$\left|W\left(\left|\frac{\omega_0^l N_{dft}}{2\pi}\right|\right)\right| << |W(0)| \tag{3.12}$$

When $W(k)$ is convolved with $S^l(k)$, the effect of the window is to spread the energy of the frequency domain impulses in $S^l(k)$ over a small region centred on each harmonic. Thus a small amount of energy from the m^{th} harmonic will be spread to the adjacent ($m-1$ and $m+1$) harmonics. However, due to (3.12) the impact of adjacent harmonics is small.

Thus the weight A_m^l of the m^{th} harmonic is not significantly affected by adjacent harmonics, as the magnitude of $W(k)$ falls of quickly either side of the centre of each harmonic. Therefore $S_w^l(k)$ can be approximated as the sum of $L = \lfloor \pi/\omega_0 \rfloor$ orthogonal functions, each function consisting of the convolution of a shifted frequency domain impulse and $W(k)$.

Using the identity $Z(k) = X(k) * Y(k) = \frac{1}{N}\sum_{i=1}^{N} X(i)Y(k-i)$, we can obtain an expression for $S_w^l(k)$:

$$S_w^l(k) = \frac{1}{2}\sum_{m=1}^{L}\sum_{i=a_m}^{b_m} A_m^l \delta\left(i - m\frac{\omega_0^l N_{dft}}{2\pi} \right) W(k-i), \quad k = 0,1,\ldots,\frac{N_{dft}}{2} \tag{3.13}$$

where:

$$a_m = \left\lfloor (m-0.5)\frac{\omega_0^l N_{dft}}{2\pi} + 0.5 \right\rfloor \tag{3.14}$$

and:

$$b_m = \left\lfloor (m+0.5)\frac{\omega_0^l N_{dft}}{2\pi} + 0.5 \right\rfloor \tag{3.15}$$

As adjacent harmonics are assumed to have no effect on the current harmonic, the convolution can be bounded by a_m and b_m to either side of the centre of the current harmonic. For example, with $F0 = 400$ Hz, $F_s = 8{,}000$ Hz, $N_{dft} = 256$, a_m and b_m are tabulated below:

m	1	2	3	4	5	6	7	8	9
a_m	6	19	32	45	58	70	83	96	109
b_m	19	32	45	58	70	83	96	109	122

Table 3.1: Example of $\{a_m\}$ and $\{b_m\}$ calculation

Assuming ω_0^l is known, equation (3.13) can be used to determine estimates of the model amplitudes $\{A_m^l\}$. Harmonic sinusoidal coders [21] sample the magnitude and phase of $S_w^l(k)$ at the centre of the harmonics of ω_0. Substituting $k = m\frac{\omega_0^l N_{dft}}{2\pi}$ into (3.13):

$$\hat{A}_m^l = S_w^l\left(m\frac{\omega_0^l N_{dft}}{2\pi}\right) = \frac{A_m^l W(0)}{2} \tag{3.16}$$

Another approach is used for MBE coders [28]. Consider the m^{th} band (harmonic) in (3.13) bounded by $k = a_m, \ldots, b_m$. A cost function is defined that measures the error between the measured signal $S_w^l(k)$ and the synthesised signal given by:

$$\hat{S}_w^l(k) = \frac{A_m^l}{2} W\left(k - m\frac{\omega_0^l N_{dft}}{2\pi}\right), \quad k = a_m, \ldots, b_m \tag{3.17}$$

The cost function is defined as:

$$E = \sum_{k=a_m}^{b_m} \left| S_w^l(k) - \hat{S}_w^l(k) \right|^2 \tag{3.18}$$

This cost function may be minimised with respect to A_m^l using calculus. For the purposes of calculus A_m^l must be considered to be a function of two independent variables (the real and imaginary part). Therefore there are several ways to differentiate E with respect to A_m^l, depending on the method used to combine the partial derivatives. A more meaningful least squares estimation procedure is described in Appendix A1.2, which leads to the expression:

$$\hat{A}_m^l = 2\frac{\sum_{k=a_m}^{b_m} S_w^l(k) W^*\left(k - m\frac{\omega_0^l N_{dft}}{2\pi}\right)}{\sum_{k=a_m}^{b_m} \left| W\left(k - m\frac{\omega_0^l N_{dft}}{2\pi}\right)\right|^2} \tag{3.19}$$

Figure 3.2 illustrates the estimation of $\left\{B_m^l\right\}$ for a $N_w = 255$ sample frame of voiced speech. The solid line represents the magnitude spectrum of the speech, the stars represent the magnitude samples for each band obtained using (3.19).

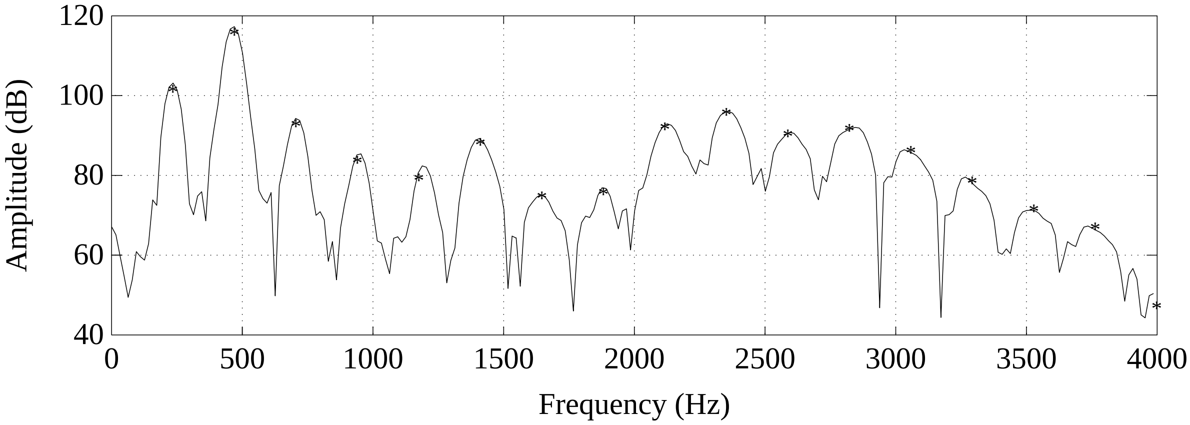


Figure 3.2: Sinusoidal Model Magnitude Estimation using the MBE Method

3.5 Pitch Estimation for Harmonic Coders

The expressions for estimating the harmonic magnitudes and phases in the previous section were derived assuming the fundamental frequency for the current analysis frame ω_0^l is known. In practice this must be estimated from the input speech signal.

A method for determining an estimate of ω_0^l based on a frequency domain analysis by synthesis model is presented in [28]. A cost function is defined:

$$E\left(\hat{\omega}_0^l\right) = \sum_{m=1}^{L} \left|E_m\left(\hat{\omega}_0^l\right)\right|^2 \tag{3.20}$$

$$E_m\left(\hat{\omega}_0^l\right) = \sum_{k=a_m}^{b_m} \left(S_w^l(k) - \hat{S}_w^l(k,m)\right) G(k) \tag{3.21}$$

$$\hat{S}_w^l(k,m) = \hat{A}_m^l W\left(k - m\frac{\hat{\omega}_0^l N_{dft}}{2\pi}\right) \tag{3.22}$$

where $\hat{S}_w^l(k,m)$ is the frequency domain synthesised speech for the m^{th} band, and $G(k)$ is an optional frequency-dependant error-weighting function. The modelled speech, $\hat{S}_w^l(k,m)$, is synthesised assuming that every band is voiced (*all-voiced synthesised speech*). The parameters $\{\hat{A}_m^l\}$ are estimated using (3.19).

To determine the fundamental (3.20) is evaluated for a range of possible $\hat{\omega}_0^l$ values (e.g. 50-400 Hz), and the value of $\hat{\omega}_0^l$ that minimises $E(\hat{\omega}_0^l)$ is chosen as the fundamental frequency estimate for this frame. The procedure can be considered to be analysis by synthesis in the frequency domain. For each possible fundamental, the model parameters are estimated and used to synthesise an all-voiced estimate, $\hat{S}_w^l(k,m)$, of the original frequency domain speech, $S_w^l(k)$. This estimate is then compared to the original in a MSE sense to determine the optimum pitch estimate.

The algorithm is described below:

1. Initialise $\hat{\omega}_0^l$ to start value.
2. Estimate $\{\hat{A}_m^l\}$ using (3.19), given the current value of $\hat{\omega}_0^l$.
3. Determine $\hat{S}_w^l(k,m)$ using $\{\hat{A}_m^l\}$ estimated in step (2).
4. Determine $E(\hat{\omega}_0^l)$, store current $\hat{\omega}_0^l$ if this is the global minima.
5. Increment $\hat{\omega}_0^l$. If new $\hat{\omega}_0^l$ smaller than stop value, go to step (2).

Figure 3.3 is an example of evaluating (3.20) for a range of possible fundamental values. In this case a global minima at the fundamental frequency ($F0 = 233$ Hz) is evident.

The estimation of the harmonic magnitudes and phases depends on the accurate estimation of the fundamental, $\hat{\omega}_0^l$ [28]. For the m^{th} harmonic, any error in the fundamental estimate $\Delta\omega_0^l = \omega_0^l - \hat{\omega}_0^l$ is magnified by a factor of m. Thus errors in $\hat{\omega}_0^l$ will influence the estimation of the high order model parameters more than the low order parameters. Therefore the component of the total error, $E(\hat{\omega}_0^l)$, contributed by the high order harmonics is more sensitive to variations in $\hat{\omega}_0^l$ that the component from the low order harmonics.

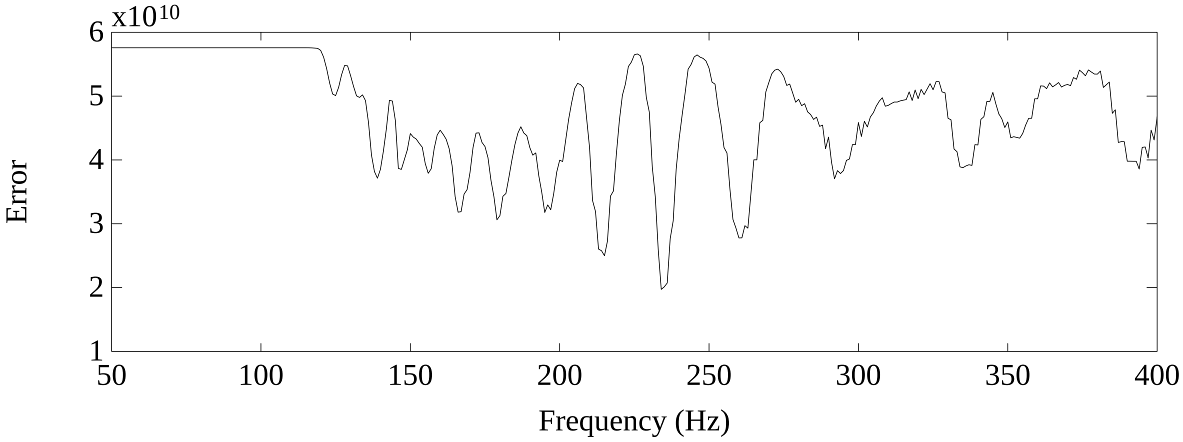


Figure 3.3: Plot of $E(\hat{\omega}_0^l)$ for Frame of Female Speech

Unfortunately, the error from the low order harmonics has a much higher amplitude due to the predominance of low frequency speech energy. Thus the low order harmonics contribution to the total error tends to dominate $E(\hat{\omega}_0^l)$. This effect can be removed by appropriate choice of $G(k)$. The frequency weighting function for Figure 3.3 was chosen to emphasise the error in the high order harmonics:

$$G(k) = \begin{cases} 0, k < N_{dft}/4 \\ 1, k \geq N_{dft}/4 \end{cases} \tag{3.23}$$

In other words, an ideal high pass filter with a cut off at half the Nyquist rate. This prevents the contribution from the high energy, low frequency harmonics contributing to the error term. Instead, the error is determined from the high order harmonics. This produces an error term, $E(\hat{\omega}_0^l)$, that is more sensitive to errors in the estimated fundamental.

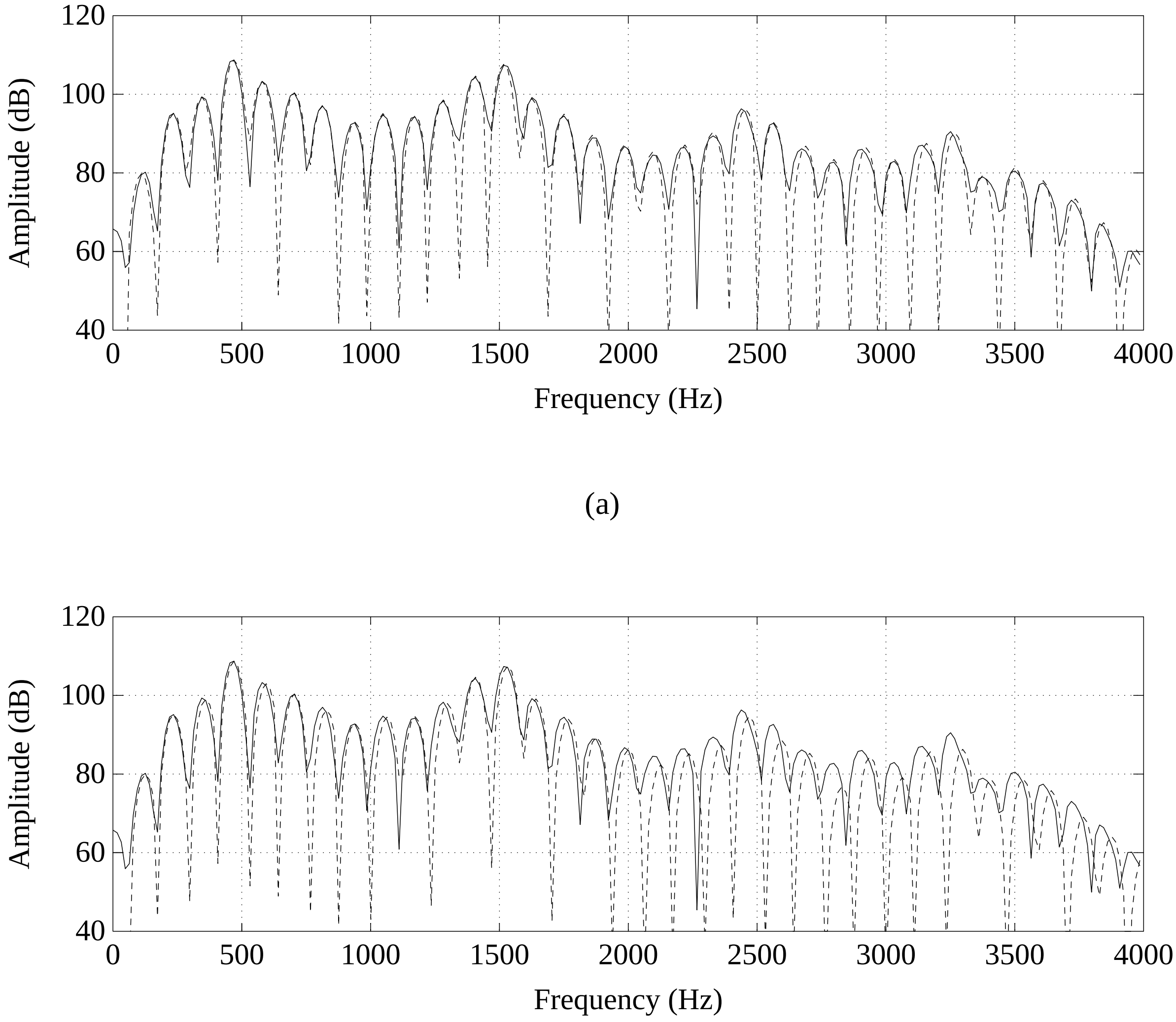
This pitch estimation technique has the ability to resolve $\hat{\omega}_0^l$ to any desired resolution. This is important as if $\hat{\omega}_0^l$ is too coarsely estimated, then substantial errors will be introduced, particularly into the high order model parameters. The authors of [28] suggest sampling

$E(\hat{\omega}_0^l)$ at a resolution of 1 Hz to avoid errors in the pitch estimate affecting the accuracy of the other estimated parameters.

Figure 3.4 presents the magnitude spectrum of a segment of voiced male speech, $S_w^l(k)$ (solid), overlayed with the all-voiced synthetic speech estimate, $\hat{S}_w^l(k,m)$ (dashed). In Figure 3.4(a), the fundamental, $\hat{\omega}_0^l$, was obtained to an accuracy of 1 Hz. This high resolution estimate of $\hat{\omega}_0^l$ was used to obtain $\hat{S}_w^l(k,m)$. Note the excellent match between the original and synthetic speech. Figure 3.4(b) illustrates the error in the synthetic estimate when a 1 Hz error in $\hat{\omega}_0^l$ was introduced. The match between the original and synthetic signals is especially poor at high frequencies.

The computational effort of sampling $E(\hat{\omega}_0^l)$ at 1 Hz intervals of $\hat{\omega}_0^l$ is very high. To minimise computational effort a two stage process is used. Initially, $E(\hat{\omega}_0^l)$ is sampled on a coarse grid to determine the minima of $E(\hat{\omega}_0^l)$ corresponding to the fundamental, $\hat{\omega}_0^l$. Then $E(\hat{\omega}_0^l)$ is sampled to the desired resolution in a small range centred on the initial coarse estimate.

The pitch estimates obtained using this technique are still subject to gross errors, as the function $E(\hat{\omega}_0^l)$ typically has several local minima. Thus a pitch tracker is used to post process the information obtained from the initial coarse sampling of $E(\hat{\omega}_0^l)$. The method used in [28] looks at several analysis frames in the past and future to determine a suitable coarse pitch estimate for the current frame. Note that analysis of future frames requires the introduction of delay, significantly increasing the overall coding delay. After pitch tracking, the coarse estimate is refined to the desired accuracy.



(a)

(b)

Figure 3.4: Plot of $S_w^l(k)$ (solid) and $\hat{S}_w^l(k,m)$ (dashed) for: (a) No error in $\hat{\omega}_0^l$, and (b) 1 Hz error in $\hat{\omega}_0^l$

3.6 Voicing Estimation for Mixed Excitation Coders

The analysis by synthesis pitch estimation equations in the previous section assumed that voiced speech was presented as the input signal for the current analysis frame. As discussed previously, the input speech signal may co

+nsist of both voiced and unvoiced energy. For mixed excitation frames, the periodic energy will tend to be restricted to the lower frequencies, and aperiodic energy confined to the higher frequencies.

The authors of [21] have described a two-band voicing model based on a Signal to Noise Ratio (SNR) which measures the fit of the harmonic model for this frame:

$$SNR = \frac{\sum_{n=N_{wl}}^{N_{wu}} \left| s_w^l(n) \right|^2}{\sum_{n=N_{wl}}^{N_{wu}} \left| s_w^l(n) - \hat{s}_w^l\left(n, \hat{\omega}_0^l\right) \right|^2} \tag{3.24}$$

where $\hat{s}_w^l\left(n, \hat{\omega}_0^l\right)$ is the all-voiced time domain speech for this frame synthesised using the estimated model parameters. Equation (3.24) can also be expressed in the frequency domain:

$$SNR = \frac{\sum_{k=1}^{N_{dft}/2} \left| S_w^l(k) \right|^2}{E\left(\hat{\omega}_0^l\right)} \tag{3.25}$$

with $G(k) = 1$ in (3.21). To derive the harmonic model parameters it was assumed that the speech for this analysis frame was fully voiced ((3.9) and (3.22)). Aperiodic (unvoiced) regions of the spectrum do not contain harmonics of the fundamental, thus in these regions the model will break down.

For fully voiced speech the all-voiced estimated model parameters will closely match the original speech, thus the SNR will be large. A partially voiced frame will result in a poorer match to the harmonic model, thus the SNR will decrease with decreasing voicing. A totally unvoiced frame will result in a still lower SNR. Assuming that the voiced energy will be confined to lower frequencies, a rule based approach is used in [21] to relate the SNR to the two-band voicing model transition frequency, ω_t. The transition frequency is quantised with 3-4 bits.

Multi-Band Excitation (MBE) coders determine a local voicing measure for each band:

$$v^l(m) = \frac{E_m\left(\hat{\omega}_0^l\right)}{\sum_{k=a_m}^{b_m} \left|S_w^l(k)\right|^2}, m = 1,2,\ldots,M \tag{3.26}$$

with $G(k) = 1$ in (3.21). This measure represents the normalised model error over the m^{th} band between the original speech and the speech synthesised using the harmonic model. A band containing voiced energy will have a small error term, $E_m\left(\hat{\omega}_0^l\right)$, thus $v^l(m)$ will be close to zero. Unvoiced energy will have a poor fit to the harmonic model, $v^l(m)$ will therefore be close to one.

In [28] the voicing measure in each band, $v^l(m)$, is quantised to one bit by comparison to a fixed threshold. The energy in each band is therefore declared voiced *or* unvoiced. More recently, other MBE coders have been presented that quantise the voicing using one bit for every three bands [30]. Typically, 12 bits per frame are required to quantise the voiced/unvoiced decisions.

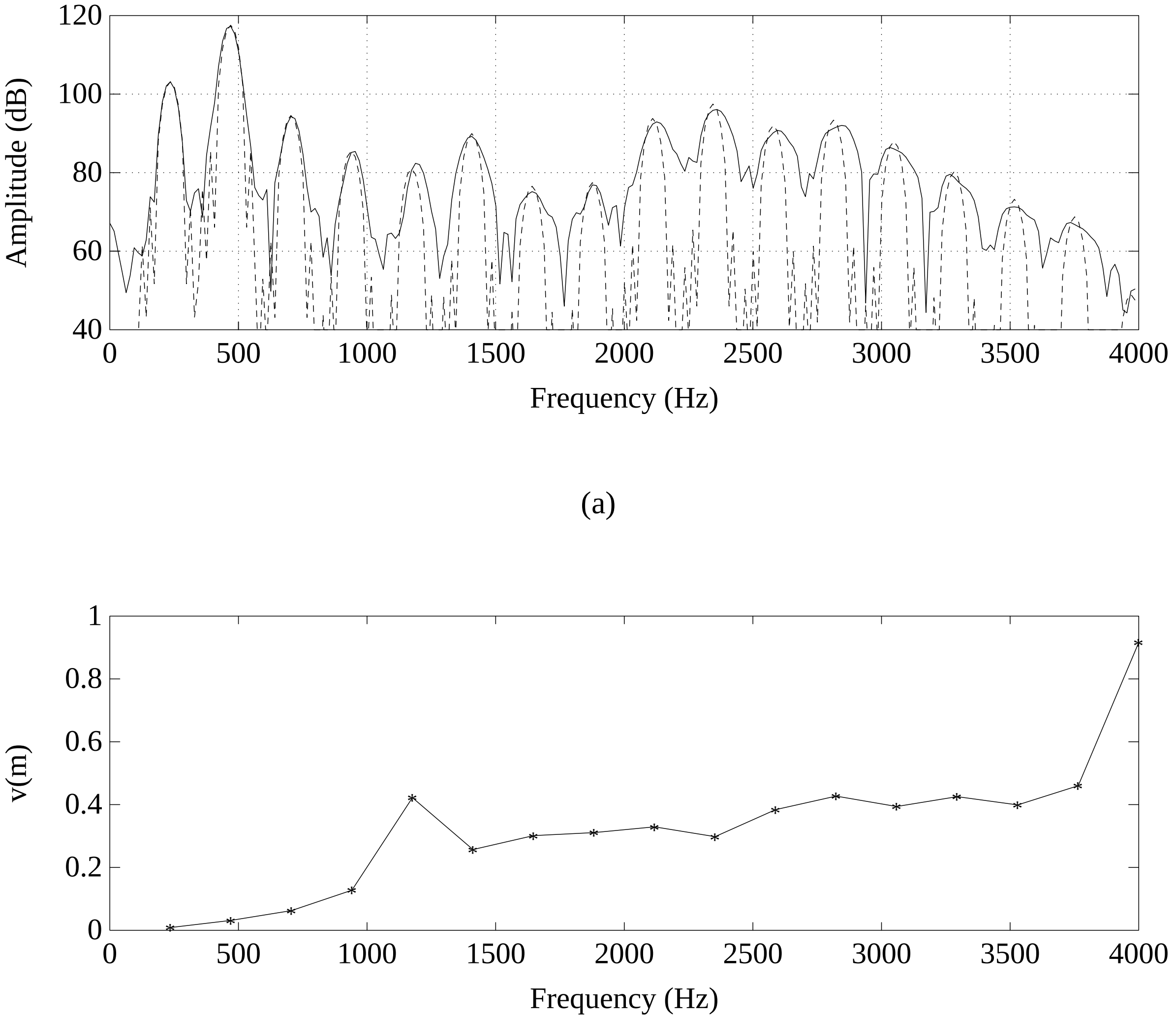
3.7 Problems with Analysis Techniques

Some problems with the parameter estimation algorithms described above have been encountered by the author. A literature survey has indicated that the problems have not been previously documented. The problems are due to the assumption that the speech (and hence sinusoidal model) parameters are stationary over short periods (10-30ms).

Figure 3.2 demonstrates a violation of the short-term stationary assumption. The high order harmonics appear to have energy spread over a wider range than the low order harmonics. This is caused by ω_0 changing as the frame was sampled. The effect is most noticeable for high order harmonics, as any change in the fundamental is multiplied by m for the m^{th} harmonic.

The assumption that ω_0 is stationary over short (less than 30ms) periods therefore appears invalid. This can lead to errors in the estimation of the other model parameters, such as the harmonic magnitudes and phases. Errors can arise as the techniques presented in the previous sections assume a fixed fundamental frequency across the entire analysis frame.

One effect of the non-stationarity of ω_0 will be a bias in the estimated harmonic magnitudes, $\left\{\hat{A}_m^l\right\}$. The expressions used to determine $\left\{\hat{A}_m^l\right\}$ assume the energy in the band is from a fixed frequency sinusoid. The effect of a non-stationary fundamental is to spread the energy over a wider frequency range. Therefore only a portion of the harmonic's energy is within the range expected by the fixed harmonic frequency analysis equations.



(a)

(b)

Figure 3.5: Voicing Estimation Problem, (a) Original and Synthetic Spectrum, (b) Voicing function, $v^l(m)$

Thus the estimated harmonic magnitudes tend to be biased below the actual harmonic magnitudes. The effect is more pronounced for the high order harmonics, as the change in harmonic frequency across the frame is greater.

Another effect of the non-stationarity of ω_0 is a bias of the voicing estimators described in the previous section. Consider the MBE voicing measure for the m^{th} band, $v^l(m)$. This function is evaluated by determining the error, $E_m\left(\hat{\omega}_0^l\right)$ between the all-voiced harmonic model of the synthesised speech, $\hat{S}_w^l(k,m)$, and the original speech, $S_w^l(k)$.

The model of the synthesised speech assumes the speech is stationary across the analysis frame. Thus the synthesised harmonics are modelled as stationary sinusoids. Figure 3.5(a) illustrates the original magnitude spectrum (solid) with the all-voiced synthetic spectrum (dashed). The corresponding voicing measure as a function of frequency is presented in Figure 3.5(b). A voicing measure near zero indicates voiced energy in a band, a voicing measure near 1 indicates unvoiced energy in the band. Due to the poor match between the original and synthetic spectrums at high frequencies, the voicing function is biased towards unvoiced. The speech is, however, clearly voiced. Thus the non-stationarity of w_0 could result in an erroneous classification of the energy in the high frequency bands to unvoiced.

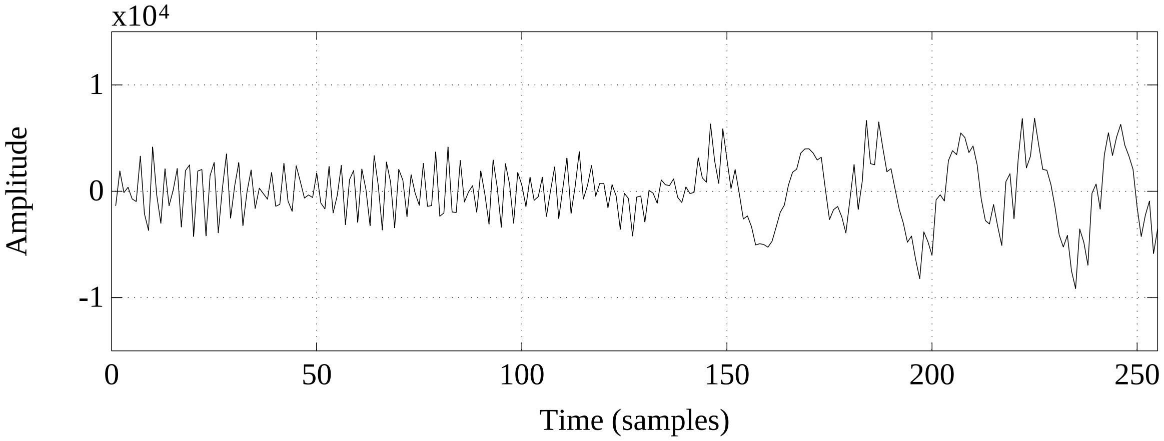


Figure 3.6: Time Domain Plot of Transition Region in "Juice"

The sinusoidal/MBE analysis expressions also have difficulty correctly analysing speech during transition regions, for example onsets of voiced speech. Consider the example presented in

Figure 3.6, a time domain plot of a unvoiced to voiced transition at the start of the word "juice", uttered by a female speaker. Note the abrupt transition from unvoiced to voiced.

Figure 3.7(a) shows the same segment in the frequency domain, $S_w^l(k)$ (solid), and the all-voiced synthetic estimate of this speech, $\hat{S}_w^l(k,m)$ (dashed). The MBE analysis expressions have had some difficulty in attempting to model the speech. An error in $\hat{\omega}_0^l$ is evident, as the positions of the original and synthesised harmonics do not match. Errors are also evident in the harmonic amplitude modelling.

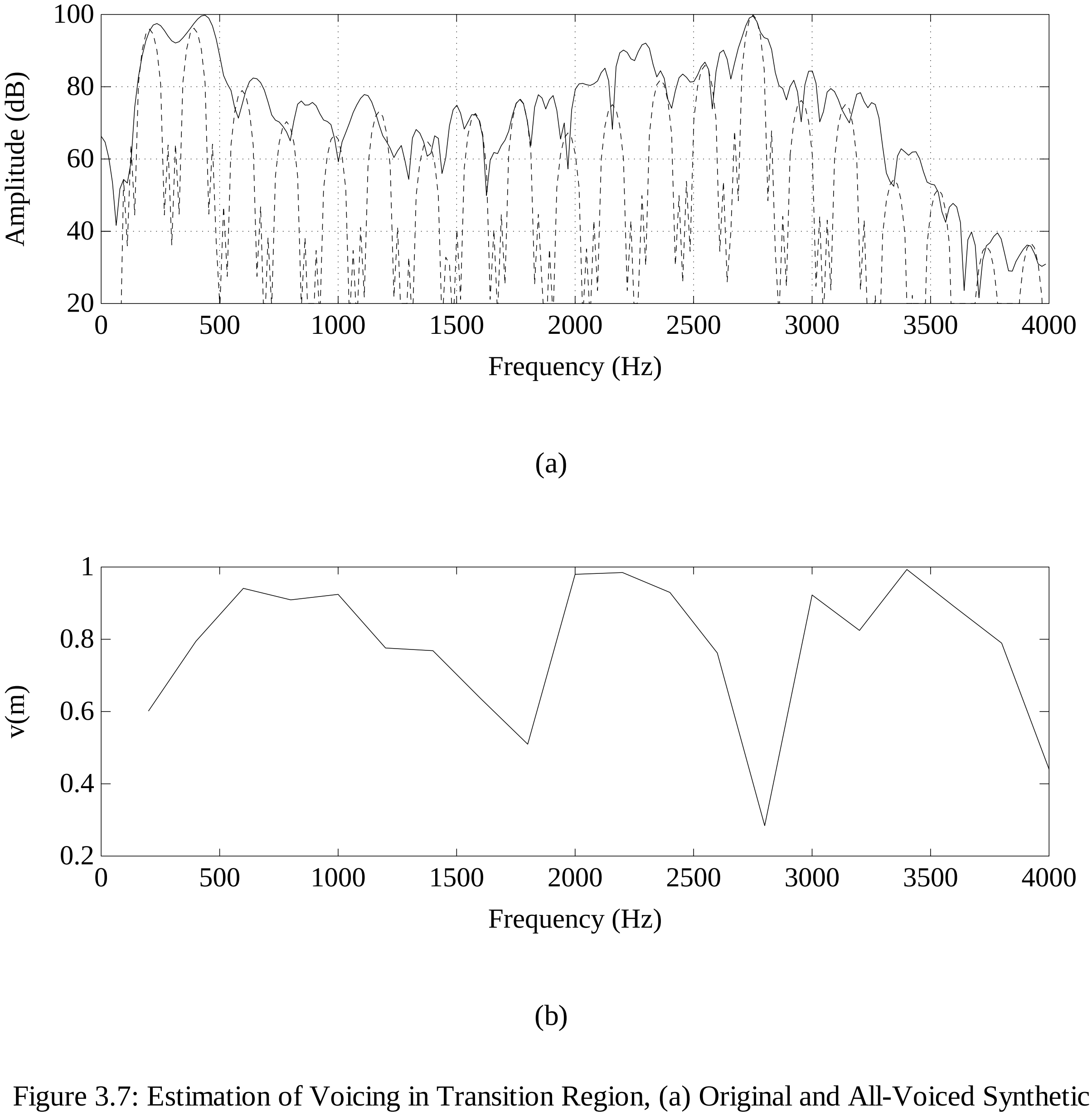


Figure 3.7: Estimation of Voicing in Transition Region, (a) Original and All-Voiced Synthetic Speech Magnitude Spectrum, and (b) Voicing Function

Figure 3.7(b) illustrates the corresponding voicing function, $v^l(m)$. Due to modelling errors, the voicing function estimates the speech as completely unvoiced. This is despite the fact that the amplitude of the voiced section of Figure 3.6 is much larger than the amplitude of the unvoiced section. Further examination of Figure 3.7(b) suggests that the frame should be considered partially voiced due to the harmonic structure of the low frequency region.

Thus it can be argued that the voicing measure defined in [28] and (3.26) fails to adequately deal with transition regions that clearly contain a mixture of voiced and unvoiced energy. An optimal voicing estimation algorithm would declare the low order harmonics voiced, and the high order harmonics unvoiced.

The failure of the voicing measure to deal with transition regions such as illustrated in Figure 3.6 is significant subjectively. In this example a speech frame that should be modelled as partially voiced is modelled as completely unvoiced. The energy of this frame is higher than the previous unvoiced frames, due to the onset of the high energy voiced speech. Thus a short, high energy burst of noise is heard in the synthesised speech, instead of a smooth transition from unvoiced to voiced speech.

The reason for the failure of the analysis expressions is due to the rapid transition. The noise (unvoiced section) is suddenly switched off, and the periodic (voiced) section suddenly starts. These rapid transitions can be viewed as rectangular windowing of the noise and voiced speech sequences in the time domain. Thus the frequency domain harmonics of the voiced speech are convolved with the DFT of the rectangular window, resulting in significant smearing of the harmonic energy in the frequency domain. The "smeared" harmonics no longer resemble the voiced synthetic harmonics, thus a poor match and high $v^l(m)$ term is obtained resulting in an estimate of the energy in that band as unvoiced.

3.8 Comparison of Coding Schemes

The following observations have been made by the author after conducting a literature survey of the time and frequency domain communications quality speech coding algorithms presented in the literature.

CELP is capable of coding communications quality speech at bit rates down to about 5kbit/s. Typically, 75% of the bit rate is allocated to the excitation (codebook parameters). Below 5kbit/s, the quality sharply degrades as not enough bits are available to adequately represent the excitation. Most of the computational complexity is in the codebook searching algorithm, and is proportional to the codebook sizes and update rates of the excitation parameters.

The adaptive codebook contributes significantly to the quality of CELP due it's ability to build up a good model of the excitation. However, the inherent memory of the adaptive codebook also contributes to its poor performance in noisy channels. Bit errors in the excitation information cause the adaptive codebook contents of the encoder to differ from those in the decoder. Due to the recursive nature of the adaptive codebook these errors may remain for some time.

As CELP uses a modified waveform matching criteria to code the input speech, it is capable of coding background noise with reasonable fidelity. Few annoying artefacts are introduced with non-speech inputs. However, waveform matching may be a somewhat wasteful approach to coding the speech, as the human ear is relatively insensitive to short term phase errors. Also, the MSE criteria used in CELP does not reflect the human ear's logarithmic magnitude response.

Parametric coders (sinusoidal, harmonic, MBE) can provide communications quality speech below 5kbit/s. They do not exhibit a knee in performance, but rather degrade gracefully with falling bit rate. Unlike CELP, they require model parameters such as pitch and voicing to be extracted from the speech signal. Reliable estimation of these parameters is still a significant problem. The estimation of these parameters is often performed by analysing several frames, introducing significant delay. As these coders attempt to fit a speech production model to the input signal, they tend to be poor at reproducing non-speech signals such as background noise.

Parametric coders have a longer frame rate than CELP for the excitation information (typically 20ms compared with 5ms for CELP), leading to lower overall bit rates for equivalent speech quality. In addition, there is no inherent memory, as with the adaptive codebook in CELP. For this reason, parametric coders tend to be more robust to channel errors than CELP.

The large, time varying number of parameters in sinusoidal coders can make them difficult to quantise. For example, in harmonic coders, the number of phase and amplitude parameters is dependant on the current fundamental frequency and therefore time varying. This is in contrast to CELP coders which have a fixed number of parameters for each frame. Fortunately there is usually a large amount of correlation present between adjacent parameters in a given frame. For instance adjacent harmonics often have similar magnitudes. This can be exploited to convert the magnitude parameters to a fixed number of parameters per frame for quantisation and transmission.

3.9 Measures of Speech Quality

One difficult aspect of low rate speech coding research is the assessment of speech quality after coding. Speech quality assessment is divided into two general areas; objective and subjective measures.

Subjective measurements are performed by playing coded speech samples to listeners and using their subjective responses to assess the speech quality. For *formal* subjective testing large numbers of listeners and careful experimental design and statistical analysis are necessary to remove experimental bias and obtain useful results [53]. Typically, only large organisations are equipped to perform formal subjective testing.

Informal subjective tests discard some or all of the experimental rigor of the formal tests to expedite the testing procedure. For example, simply listening to coded speech compared to the original may yield significant information about the strengths and weaknesses of the coding process. Such tests are highly subjective, and perceived coding characteristics tend to vary widely with different listeners. However these tests are useful where significant differences are easily determined. Informal subjective tests are very useful during development and tuning of coder design as they can be executed quickly.

Objective testing uses a computer program to compare coded and original speech waveforms using a distortion metric [54]. Ideally, this distortion metric will correspond to our subjective perception of speech and therefore give results similar to formal subjective testing. An objective measure capable of reproducing subjective results for any coded speech signal does not exist yet [56][53], however this field is being actively researched [55]. For this reason, subjective and objective tests often produce different results.

Several objective measures do exist that produce useful results for coders of a given family, for example Segmental Signal to Noise Ratio (SEGSNR) [54] is useful for time domain waveform coders and some hybrid coders such as CELP that incorporate waveform matching mechanisms. Also, measures exist that evaluate elements of speech coding algorithms, such as Cepstral Distortion (CD) [54], and Spectral Distortion (SD) [54], which are often used for measuring the distortion introduced by the quantisation of LPC parameters.

Sinusoidal coders have proven difficult to evaluate using objective measures, mainly because most low rate algorithms discard phase information. Thus waveform based measures such as SEGSNR which are very sensitive to phase have not been useful for most previous sinusoidal coding work. However, the techniques developed for sinusoidal coders in this thesis do attempt to preserve the harmonic phases, therefore a useful objective measure based on SEGSNR in the frequency domain is used to evaluate the phase modeling techniques in Chapter 6. A Spectral Distortion (SD) method is also used to evaluate the LSP quantiser performance in Chapter 7.

3.10 Conclusion

This chapter has presented the background information and analysis that is necessary for the presentation of the following thesis contribution chapters. Most importantly, the concept of harmonic sinusoidal coding was introduced and qualitatively and quantitatively examined.

Chapter 4 presents a pitch estimation algorithm developed by the author for use in harmonic sinusoidal coding algorithms. This algorithm combines the properties of a square law non-linearity with the MBE pitch estimation algorithm presented in section 3.5 to produce a robust pitch estimator.

4. Non-Linear Pitch Estimation

This chapter presents a pitch estimator based on the application of a square law non-linearity to the input speech signal. The algorithm is denoted Non-Linear Pitch (NLP). The algorithm has moderate computational complexity, low algorithmic delay (small buffering requirements), and robustness to gross pitch errors (halving and doubling). The algorithm employs a minimum number of experimentally derived constants.

After a survey of existing pitch estimation algorithms, it appears that no other algorithms in the literature use a non-linearity combined with a secondary pitch estimator for post processing. Pitch estimation concepts were introduced in section 2.4 of this thesis.

The NLP algorithm was developed using three evaluation techniques denoted as *automatic*, *contour,* and *subjective*. These techniques are presented and discussed in section 4.5, and objective results presented for the automatic method, where the NLP algorithm is shown to perform well compared to two other algorithms. *All* pitch estimators will fail under certain circumstances. The failure modes of the NLP algorithm are carefully examined and presented in section 4.6.

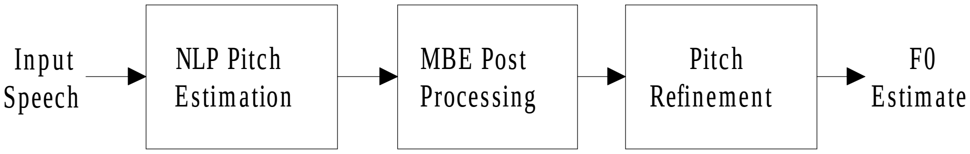


Figure 4.1: Pitch Estimation Block Diagram

The NLP pitch estimation algorithm is a three stage process, illustrated in Figure 4.1. The first stage (section 4.1) , *basic pitch extraction,* determines a set of candidate values from a frame of input speech using a process based on the properties of a square law non-linearity. The next stage (section 4.2), *post processing,* uses a variation of the MBE pitch estimation technique [28]. The final stage (section 4.3), *pitch refinement,* is used to obtain an accurate estimate of the fundamental frequency for the current frame. This stage uses a low complexity pitch refinement algorithm. One possible disadvantage is that the algorithm does not provide an estimate of voicing, ie the algorithm will return a pitch estimate even if the speech is unvoiced.

Pitch estimation is a difficult task for reasons described in section 2.4. One of the main problems is that no comprehensive speech production model exists that describes all of the effects observed in speech signals. For this reason it is difficult to formulate or evaluate a pitch detection algorithm based on a quantitative model of speech production. Therefore the operation of the NLP algorithm is analysed in a largely qualitative manner.

One problem with many pitch estimators is heavy reliance on experimentally derived constants. These are often included in the post processing stage to improve the performance of a basic pitch extractor. Problems arise as these constants are often tailored to correct specific failures of the basic pitch extractor encountered during development. This leads to breakdown of the pitch estimator with other speech utterances outside the development database. It is for this reason that many of the pitch estimation algorithms reported in the literature have performance that is difficult to reproduce, or even assess objectively.

4.1 Basic Pitch Extractor

Pitch estimators based on non-linearities have been proposed by several authors, a summary of these algorithms is presented in [4]. The non-linearity is generally applied directly to the speech signal. Non-linearities are often used to spectrally "flatten" the speech, ie to partially remove the effects of the vocal tract filtering. Another use of non-linearities is to enhance the fundamental through the superposition of difference tones produced by harmonic distortion.

A new basic extraction algorithm is presented, based on a square law non-linearity. The non linearity is used to regenerate the fundamental from band limited speech. By peak picking the Discrete Fourier Transform (DFT), a set of candidate pitch estimates can be determined. These candidates are then passed to the next processing stage for evaluation.

Several other non-linearities were tested experimentally (for example cube law and centre clipping) however the square law provided the best performance. For this reason and it’s ease of implementation on modern DSP chips, the square law non-linearity was chosen.

Given a speech signal represented by the sinusoidal model:

$$s(n) = \sum_{m=1}^{L} B_m \cos(\omega_0 mn + \theta_m) \tag{4.1}$$

consider the application of a square-law non-linearity to (4.1):

$$s^2(n) = \sum_{m=1}^{L} B_m \cos(\omega_0 nm + \theta_m) \sum_{l=1}^{L} B_l \cos(\omega_0 nl + \theta_l) \tag{4.2}$$

$$s^2(n) = \frac{1}{2} \sum_{m=1}^{L} \sum_{l=1}^{L} B_m B_l \left[\cos(\omega_0 n(m+l) + \theta_m + \theta_l) + \cos(\omega_0 n(m-l) + \theta_m - \theta_l) \right] \tag{4.3}$$

The second term inside the square brackets will introduce a large number of components at the fundamental, when $|m-l| = 1$. A large DC term will also be present, when $m = l$. A smaller number of components will be generated at multiples of the fundamental. Note that energy at the fundamental frequency will be generated by the non-linearity even when the fundamental has been removed from $s(n)$, eg. when the speech has been band-pass filtered.

A frequency domain view of the effect of the square law non-linearity can be obtained by considering the identity:

$$DFT\{x(n)y(n)\} = Z(k) = \frac{1}{N} \sum_{l=0}^{N-1} X(l)Y(k-l) \tag{4.4}$$

where $X(k)$ is the N point DFT of $x(n)$ and $Y(k)$ is the N point DFT of $y(n)$. This identity states that a multiplication of two time domain signals corresponds to a (circular) convolution of the Discrete Fourier Transforms of the two signals. Therefore, the multiplication of a time domain signal by itself (squaring), leads to the autocorrelation of the DFT of the signal in the frequency domain:

$$DFT\{x^2(n)\} = Z(k) = \frac{1}{N} \sum_{l=0}^{N-1} X(l)X(k-l) \tag{4.5}$$

The magnitude of this signal $|Z(k)|$ will have peaks at frequencies corresponding to $F0$ and multiples of $F0$, and can therefore be used as a pitch detector.

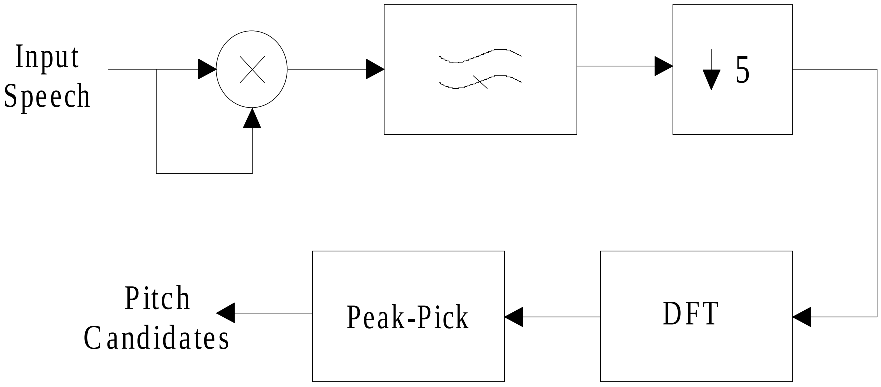


Figure 4.2: Basic Pitch Extraction

Figure 4.2 illustrates the basic pitch extractor. The fundamental frequency is estimated in the range of 50-400 Hz, typical of other pitch estimators. The algorithm is designed to take blocks of $M = 320$ samples at a sample rate of 8 kHz (40 ms window). This block length ensures at least two pitch periods lie within the analysis window at the lowest fundamental frequency.

The analysis window size is a trade off between algorithmic delay and low frequency accuracy. A shorter analysis window will reduce the delay of the algorithm but have poor accuracy when used to estimate the pitch of low $F0$ speech signals. Other algorithms, such as Inmarsat-M IMBE [12], use a shorter block length (280 samples for the basic pitch extractor) to offset the delay caused by forward pitch tracking algorithms. As the NLP algorithm has no forward pitch tracking, increasing the block length of the basic pitch extractor to improve $F0$ performance was deemed a reasonable compromise.

The speech signal is first squared then notch filtered to remove the DC component from the squared time domain signal. This prevents the large amplitude DC term from interfering with the somewhat smaller amplitude term at the fundamental. This is particularly important for male speakers, who may have low frequency fundamentals close to DC. The notch filter is applied in the time domain and has the experimentally derived transfer function:

$$H(z) = \frac{1 - z^{-1}}{1 - 0.95z^{-1}} \tag{4.6}$$

Before transforming the squared signal to the frequency domain, the signal is low pass filtered and decimated by a factor of 5. This operation is performed to limit the bandwidth of the squared signal to the approximate range of the fundamental frequency. All energy in the squared signal above 400 Hz is superfluous and would lower the resolution of the frequency domain peak picking stage. The low pass filter used for decimation is an FIR type with 48 taps and a cut off frequency of 600 Hz. The decimated signal is then windowed and the $N_{dft} = 512$ point DFT power spectrum $U(k) = |Z(k)|^2$ is computed by zero padding the decimated signal with 448 zeros.

The DFT power spectrum of the squared signal generally contains several local maxima. In most cases, the global maxima of $U(k)$ will correspond to $F0$, however occasionally the global maxima corresponds to a spurious peak or multiple of $F0$. Thus it is not appropriate to simply choose the global maxima of $U(k)$ as the fundamental estimate for this frame. Instead, a set of $F0$ candidates are extracted by determining the positions of *all* of the local maxima of $U(k)$. These candidates are then passed to the post processing stage for further analysis.

To limit the number of candidates, only those local maxima above a certain threshold T are considered. The value of the threshold varies dynamically with each frame and is set as a fraction of the global maximum of the frame:

$$T = T_0 U(k_{\max}) \tag{4.7}$$

where T_0 is an experimentally derived constant set to 0.1, and $k_{\max}$ is the DFT bin containing the global maxima of $U(k)$. The frequencies of all those local maxima above a certain threshold are determined and are known collectively as the set of fundamental candidates $\{k_1, \ldots, k_V\}$ where V is the number of local maxima in $U(k)$ larger than T, and k_v is the DFT bin corresponding to the v^{th} local maxima. It may be useful to adopt a fixed maximum V for real time implementation of the algorithm. This will constrain the post processing stage to a known maximum computational load.

The frequency resolution of the candidates is determined by the size of the DFT. A larger DFT will enable a higher accuracy in locating the peak of the magnitude spectrum. The resolution of the NLP algorithm is given by the DFT bin spacing S, which may be computed as:

$$S = \frac{F_s}{N_{dft} D} \tag{4.8}$$

where F_s is the sampling rate, N_{dft} is the DFT size, and D is the decimation ratio. For the current implementation, $\mathrm{S} = 3.125\ \mathrm{Hz}$.

4.2 Post Processing

The post processing algorithm evaluates each of the candidates determined by the basic pitch extractor, and chooses one as the pitch estimate for this frame. The decision is made by evaluating a cost function $E(\omega_v)$ at each of the candidate fundamental frequencies ω_v, where ω_v is related to the candidate DFT bin k_v by $\omega_v = 2\pi k_v / DN_{dft}$. The fundamental candidate that minimises $E(\omega_v)$ is chosen as the fundamental estimate for this frame.

The cost function $E(\omega)$ (3.20) is based on the MBE pitch estimation algorithm [28], discussed in section 3.5 of this thesis. The MBE pitch estimation algorithm samples $E(\omega)$ over the entire pitch range, and chooses the fundamental frequency ω_0 that minimises $E(\omega)$. For MBE coders such as [12], an efficient time domain form of the algorithm is used. The cost function of previous and future frames are compared to the current frame using a pitch tracking algorithm, to determine a coarse pitch estimate. This is then refined by sampling the more complex, but more accurate frequency domain form of the algorithm in the region of the coarse pitch estimate.

In the post processing role for the NLP algorithm, the frequency domain form of the MBE cost function is sampled in small regions around each of the fundamental candidates, leading to a moderate computational complexity. This cost function was described in section 3.5 of this thesis.

To reduce computational complexity, the frequency weighting function $G(k)$ was set to exclude all harmonics above 1000 Hz:

$$G(k) = \begin{cases} 1, & k < N_{dft}/8 \\ 0, & k \geq N_{dft}/8 \end{cases} \tag{4.9}$$

The MBE cost function is sampled between plus/minus 10 Hz of each candidate frequency, in 2.5 Hz steps.

4.3 Pitch Refinement

A simple pitch refinement algorithm has been successfully used to refine the accuracy of the fundamental estimate obtained from the post processing stage. This refinement technique is very computationally simple compared to others such as dense sampling of the MBE cost function [28]. The accuracy has been found to be sufficient for the sinusoidal coder described later in this thesis.

A cost function is defined:

$$E(\omega_0) = \sum_{m=1}^{L} \left| S_w(m\omega_0) \right|^2 \tag{4.10}$$

which simply samples the power spectrum of the windowed speech signal for this frame. The argument of S_w is rounded to the nearest integer. This function is obviously biased towards longer pitch periods (smaller fundamentals) as it converges to the energy in the spectrum. However, over the small range of frequencies the function is sampled the bias is not significant. Most importantly, the function does exhibit local maxima in the vicinity of the fundamental of the frame.

In the pitch refinement role, the function is sampled in two steps. First, the function is sampled in the range of plus or minus 5 samples of pitch period in 1 sample steps. Then the function is sampled in the range of plus of minus one sample of the pitch period, in 0.25 sample steps.

4.4 Example of NLP Algorithm Operation

This section provides a graphical illustration of the NLP algorithm using a frame of female speech as an example. The input time domain speech is shown in Figure 4.3, and the corresponding magnitude spectrum in Figure 4.4. Note that the periodic energy in this signal is largely confined to the low frequency end of the spectrum.

Figure 4.5 illustrates the squared input speech in the time domain. Note the strong periodic component at the pitch period, and the DC offset of the signal. Figure 4.6 shows the squared time domain signal after notch filtering to remove the DC component that would otherwise interfere with the signal components at $F0$.

Figure 4.7 shows the squared speech power spectrum. In this case, there are two candidates, corresponding to the local maxima near 180 Hz and 360 Hz. These are evaluated by sampling the MBE cost function in the vicinity of the local maxima of $U(k)$, illustrated in Figure 4.8. The MBE cost function shows a minima around 180 Hz, which is chosen as the fundamental estimate for this frame.

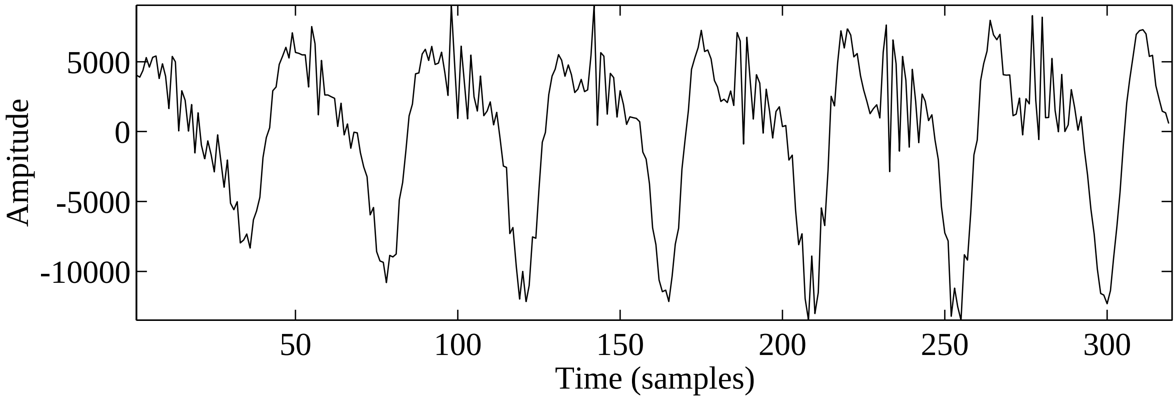


Figure 4.3: Frame of Female Input Speech

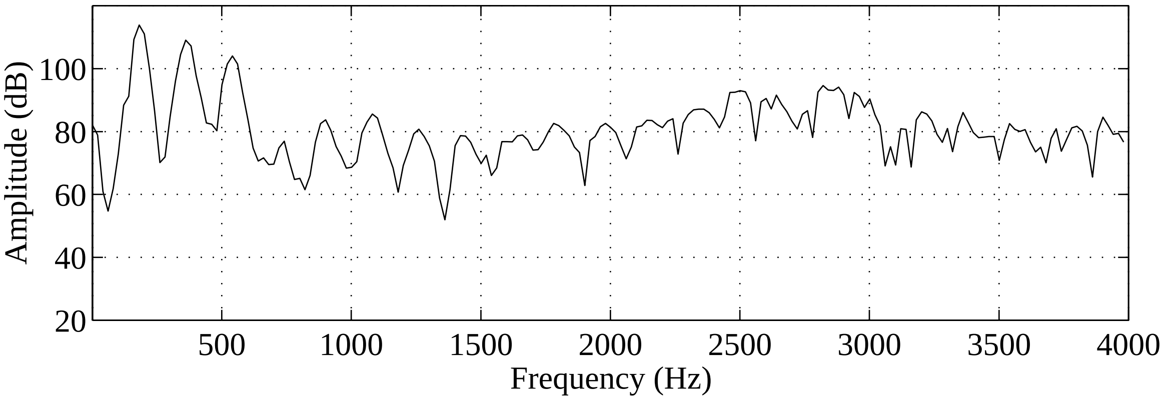


Figure 4.4: Input Speech Magnitude Spectrum

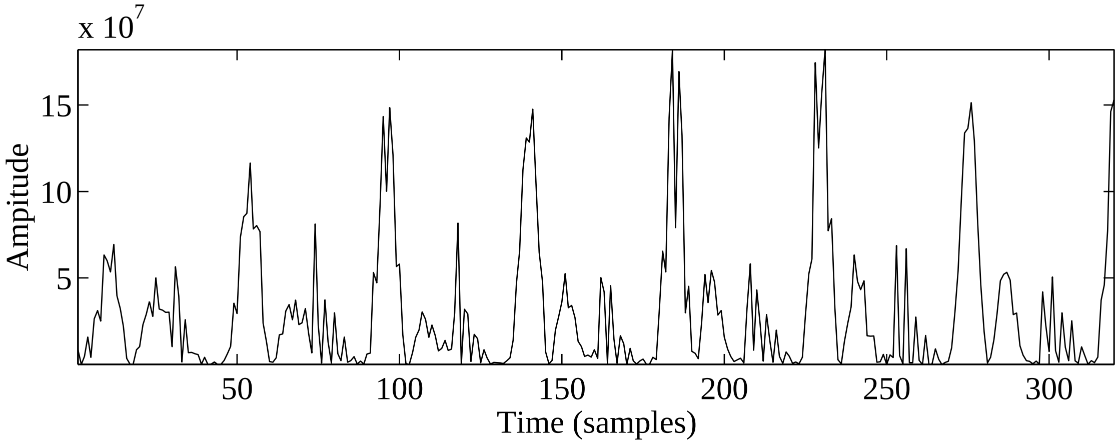


Figure 4.5: Squared Time Domain Speech

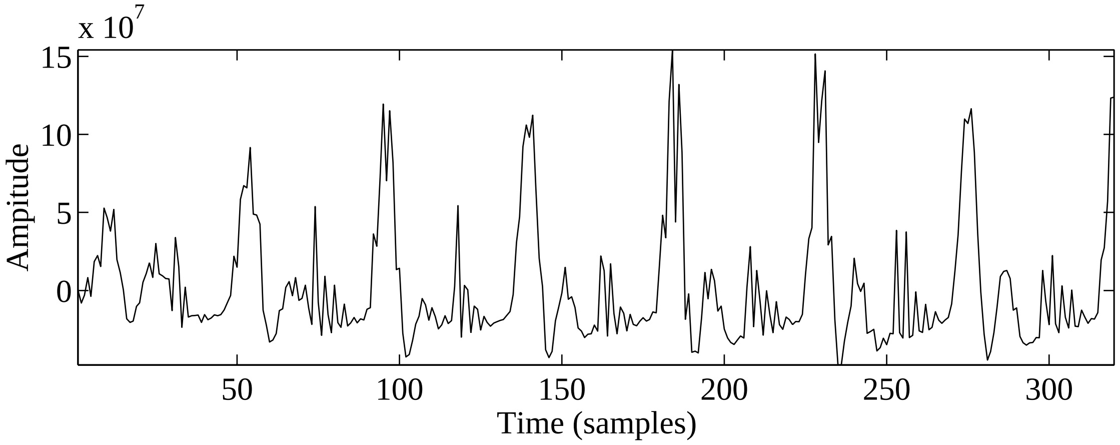


Figure 4.6: Notch Filtered Squared Time Domain Speech

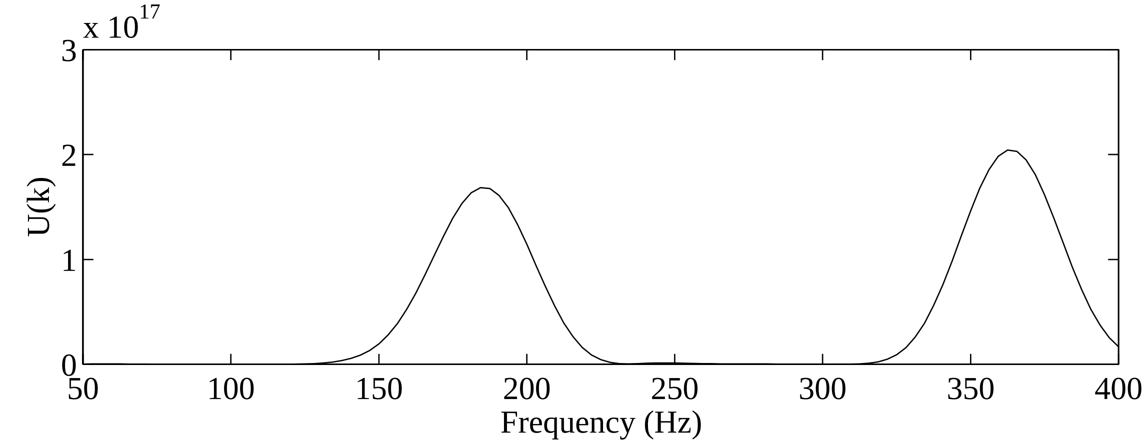


Figure 4.7: Squared Speech Power Spectrum $U(k)$

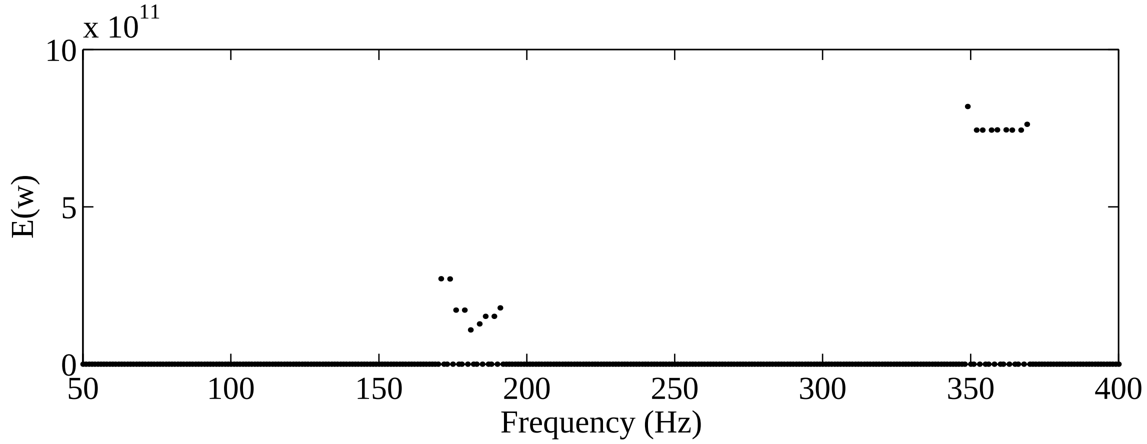


Figure 4.8: Samples of MBE Cost Function at Local Maxima of $U(k)$

4.5 NLP Algorithm Evaluation

Testing pitch estimators is an important part of pitch estimator development, often ignored by many authors in the field. It is relatively easy to develop a pitch estimator that will work reliably on a certain utterance, but it is very difficult to develop a pitch estimator that will work reliably across a range of speakers and conditions. Therefore extensive testing over as wide a range of speakers and conditions as possible is desirable to fully test a pitch estimation algorithm. This however, presents problems in the complexity and range of tests to be considered.

This section presents tests results obtained with the NLP algorithm. The tests are divided into 3 types, automatic, subjective, and listening. The test methodology for each method is presented below, and objective results are presented for the automatic method. Before proceeding it is useful to discuss the types of pitch errors and the effects of these errors on speech coders, in particular sinusoidal speech coders.

Failures in pitch estimators can be grouped into two types [37]:

- *Gross errors*, where the pitch error is greater that 1 ms. These errors often occur when the pitch estimator looses the pitch track completely, for example when the pitch estimator chooses a pitch estimate corresponding to the first formant, or to a multiple or submultiple of the actual pitch.

- *Fine errors,* where the pitch error is less than 1 ms.

Of the two error types, gross errors are by far the most serious for sinusoidal coders, and represents one of the most challenging parts of sinusoidal coder development. Gross pitch errors produce perceptually disturbing distortions in the coded speech that seriously degrade the coded speech quality. Once a reliable coarse pitch estimate is obtained (free of gross pitch errors), reliable algorithms (for example [28]) exist for increasing the accuracy of the estimate and therefore reducing the fine error. For this reason the tests applied are confined to the evaluation of gross error performance only.

Due to the lack of comprehensive speech production models, synthetic speech signals with artificially generated pitch contours are not suitable for pitch estimator evaluation, except perhaps for evaluating pitch estimator fine error performance. Thus databases of real speech signals must be used to evaluate pitch estimator performance. Several criteria can be used to evaluate pitch estimator performance (objective or subjective), these are considered in the tests described below.

4.5.1 Automated Pitch Estimator Evaluation (*automatic*)

This type of testing compares a candidate pitch estimator to a hand estimated reference pitch database. The criteria of gross errors can be applied and the number of errors counted for each candidate algorithm. The candidate pitch estimator with the highest “score” (smallest number of gross errors) when compared with the reference database is judged to be superior.

In this case, manually estimated pitch values for a 2400 frame database of 4 speakers were derived using a semi-automatic pitch detector system similar to [38]. This system presents the operator with several "views" of the current frame of speech. The views are the time domain speech, the DFT magnitude spectrum, and the autocorrelation function. Based on the operator's judgement, a decision is made on voicing. If the frame is voiced the operator then chooses the peak of the autocorrelation function that correctly represents the periodicity of the frame.

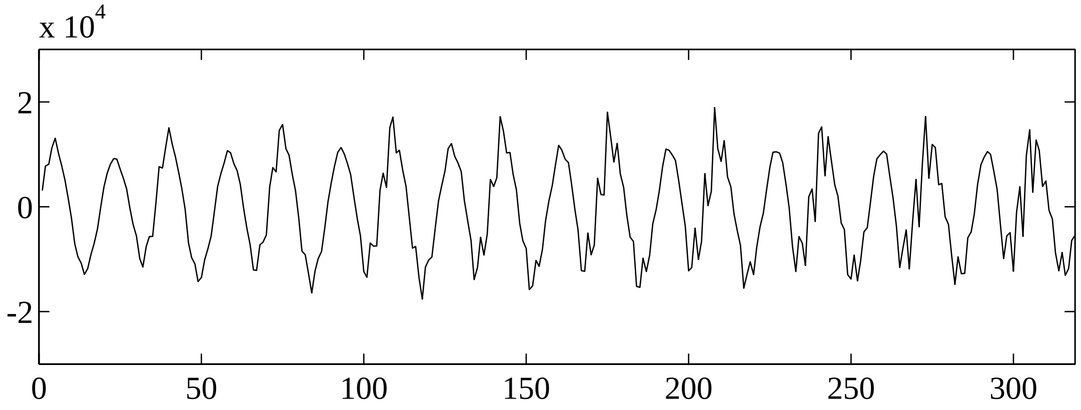


Figure 4.9: Original Speech Window

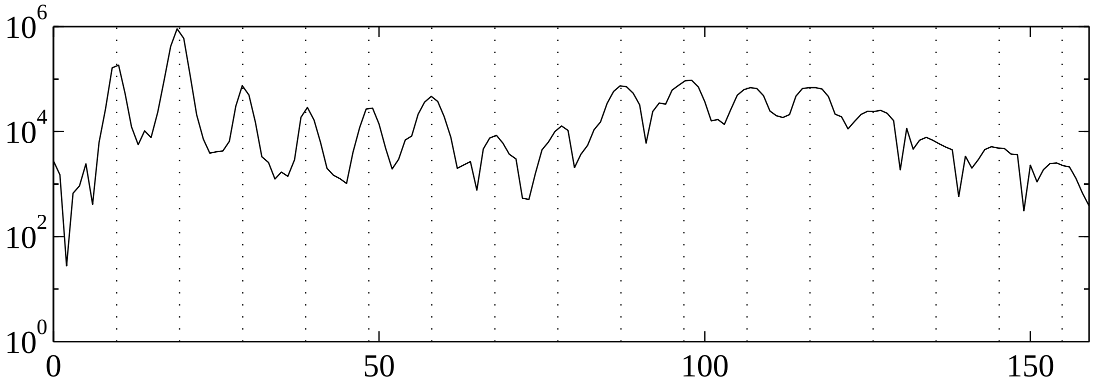


Figure 4.10: Magnitude Spectrum Window with Harmonic Marker Lines (dotted)

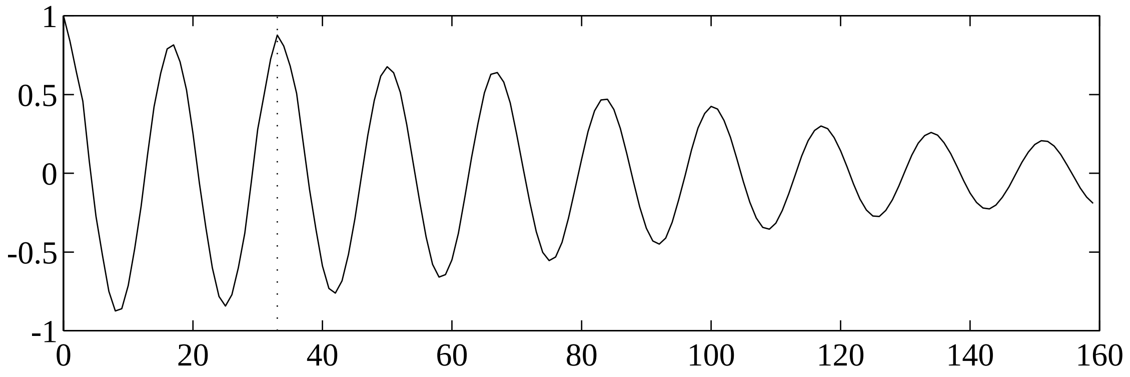


Figure 4.11: Autocorrelation Window with Pitch Marker Line (dotted)

Figures 4.9 to 4.11 provide an example of the semi-automatic pitch detector operation. Figure 4.9 is the first window presented to the operator, this window displays the time domain speech. From this window the operator can judge if speech is present, and if it is voiced or unvoiced. The next window (Figure 4.10) presents the magnitude spectrum (solid line) of the speech. Superimposed on this window are dashed harmonic marker lines. These lines are drawn at multiples of the current $F0$ estimate. If the current $F0$ estimate is correct, these markers will be aligned with the magnitude spectrum harmonic peaks. This system aids the operator in choosing the correct $F0$ estimate for this frame. The final view of the signal is in the autocorrelation domain, Figure 4.11. For each frame, the software automatically chooses the peak of the autocorrelation function as an initial estimate, placing the dotted marker line on the chosen peak. This initial estimate may be changed by the operator until the correct peak (in the operators opinion) is chosen.

The reference pitch database was compared to the NLP algorithm and two control pitch estimation algorithms. The first algorithm is the pitch estimator from the Inmarsat-M IMBE vocoder [12], which is unique in the literature as being a comprehensively defined specification used in a practical speech coder. Most other algorithms in the literature are only partly defined, for example experimentally derived constants are not presented. This makes exact implementation of these algorithms very difficult. For this reason the Inmarsat-M IMBE algorithm makes an ideal control for testing the NLP algorithm.

The second control algorithm is an autocorrelation type pitch estimator commonly used in the literature [39]. The combined results for all 4 speakers (2400 frames total) are presented in

Table 4.1. In addition Figure 4.12 compares the pitch contour of the NLP algorithm and the reference database for a section of male speech.

Algorithm	Hits	Misses	% Correct
Autocorrelation	488	115	80.9
Inmarsat-M	531	72	88.1
NLP	553	50	91.7

Table 4.1: Results for Automatic Pitch Estimator Evaluation

A "hit" is recorded when the pitch estimator under test matches the reference database, a "miss" occurs when a gross error (pitch deviation greater that 1 ms) occurs. No comparison is performed on unvoiced frames. These were marked during the generation of the reference database, and are ignored during pitch estimator evaluation. Over this test database, the NLP algorithm has the lowest incidence of gross errors.

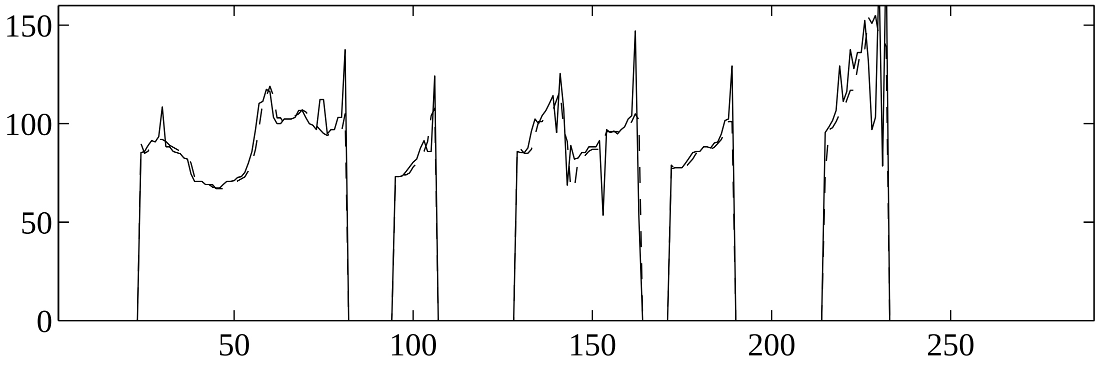


Figure 4.12: Pitch Contour for NLP (solid) and Reference Database (dashed)

There are several problems with this method of pitch estimator evaluation:

1. Typically each 10 ms frame requires about 30 seconds of analysis for an experienced operator. Thus each second of speech requires about 50 minutes of analysis time. For the large speech databases needed for reliable pitch estimator evaluation, the amount of analysis time quickly becomes unreasonable.

2. No two operators will give the same results. They must make subjective judgements such as if any periodic energy is present in the frame, if so which of the possible candidates presented by the automatic parts of the system represents the likely pitch. This problem can be alleviated by operator training and averaging the hand estimates over several operators, however this multiplies the analysis time by the number of operators.

3. A pitch estimator with a higher "score" when compared to the reference database is supposedly superior to another algorithm, however this may not be the case. Gross pitch errors have different subjective impact depending on where they occur. For example, a gross error in a high energy voiced section will be subjectively very disturbing whereas a gross error in a low energy partially voiced region may be inaudible. Thus simply comparing candidate pitch estimators against a reference on the basis of a match score is not sufficient to determine the ranking of the candidate algorithms performance. For example, during the development of the NLP algorithm it was found that some variants of the NLP algorithm with lower match scores actually sounded superior to those with a higher match score.

4.5.2 Semi-Automatic Pitch Contour Evaluation (*contour*)

This pitch estimator evaluation scheme uses an interactive computer program to simultaneously view several frames of the input speech, and the resulting pitch contour generated from those frames. The operator visually compares the two windows, noting the shape of the pitch contour and the speech. Gross errors are easily identified as discontinuities in the pitch contour during voiced speech.

This method allows evaluation of speech at a faster rate than the automatic method discussed above, as several frames are examined at once. Also, the importance of any pitch errors can be evaluated by the operator by examining the input speech where the error occurred. However, this method is still somewhat slow, and the entire test database must be manually re-examined every time a modification is made to the pitch estimator.

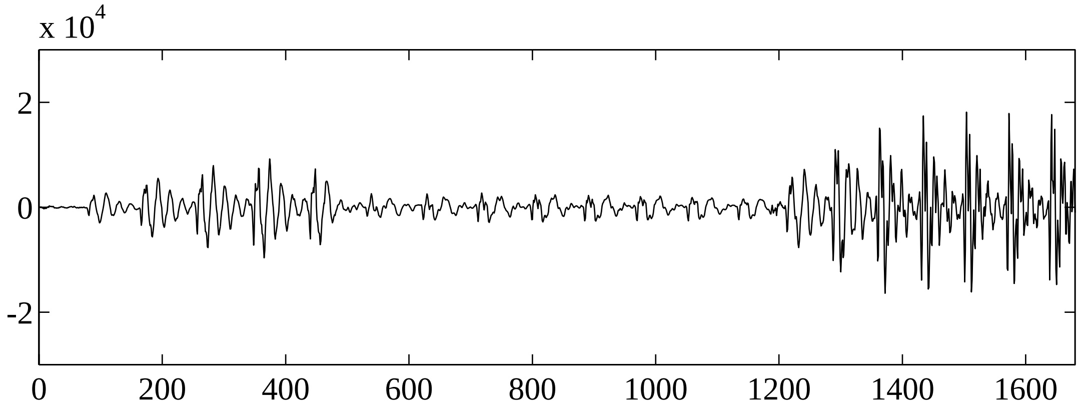


Figure 4.13: Input Speech Window for Contour Pitch Estimator Evaluation Method

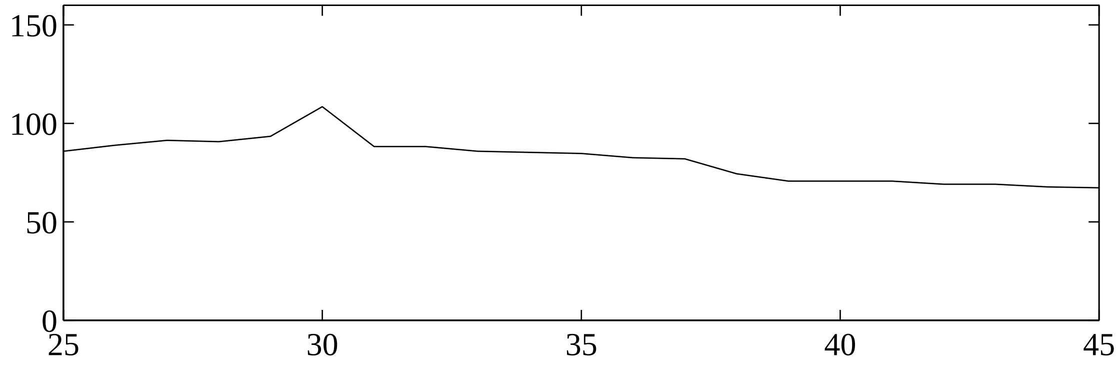


Figure 4.14: Pitch Contour Window for Contour Pitch Estimator Evaluation Method

The test database detailed in Appendix C was processed using the NLP algorithm and the resulting pitch contours examined with the contour method. During development of the NLP algorithm, this provided a way of rapidly determining the position and importance of gross errors. For example errors in low energy, partially voiced regions were found to have little impact on the subjective speech quality and could safely be ignored. Gross errors in high energy, strongly voiced regions, however were very important and therefore caused analysis of and modification of the algorithm to prevent their occurrence. Thus the contour method proved to be more effective in testing the algorithm performance than the automatic method, due to it’s ability to distinguish the subjectively important gross errors.

Using the final version of the NLP algorithm presented in this chapter, no gross errors in strongly voiced regions were encountered over the test database in Appendix C.

4.5.3 Subjective Testing Using Sinusoidal Coder (*subjective*)

The final testing method is based on listening tests where the pitch estimator is used to encode and decode speech using the sinusoidal coder algorithm [21]. This method is relatively fast, as listener evaluation occurs at the rate the processed speech is played back. Perceptually important gross pitch errors show up as impulsive sounds such as “cracks” of “beeps” in the processed speech. These errors can then be traced to specific coder frames using the contour method described above.

The test database in Appendix C was processed using the NLP algorithm and the generic sinusoidal coder described in Chapter 5. No gross perceptual errors were evident over the entire database.

4.6 Failure Modes for Non-Linear Pitch Estimation

During development the NLP algorithm was tested widely on a variety of speech sources using the subjective testing method described above. Under certain conditions, the algorithm would break down, producing subjectively annoying sounds in the synthesised speech. Using the contour testing scheme the problem frames were isolated for further analysis. This section examines the reasons for these failures of the NLP algorithm, and proposes a pitch tracking method to alleviate the problems encountered.

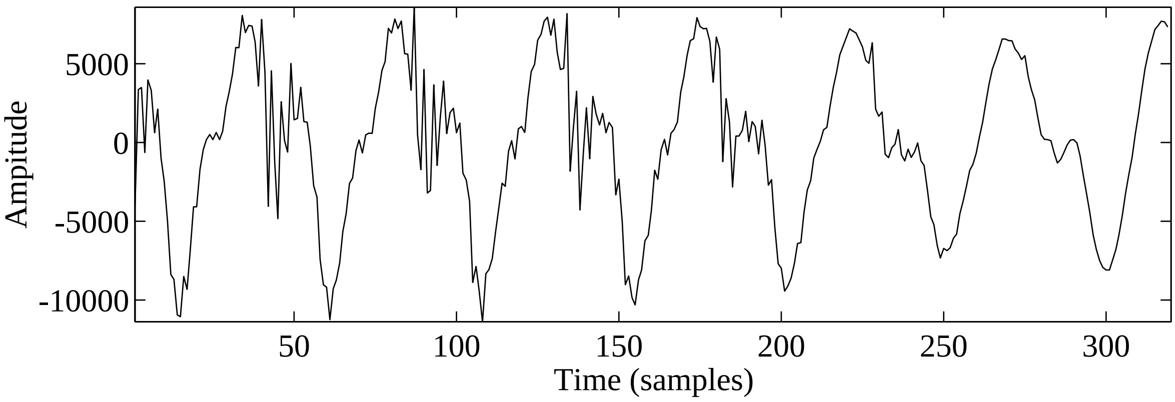


Figure 4.15: Frame of Input Speech

Figure 4.15 is an example of an input frame of speech that produces a failure in the NLP algorithm, Figure 4.16 is the corresponding squared speech power spectrum $U(k)$. The failure occurs because the local maxima in $U(k)$ at the fundamental (around 170 Hz) is suppressed compared to the strong global maxima at the 2nd harmonic (around 340 Hz). The local maxima at the fundamental fails to exceed the threshold given by (4.7) and is therefore not considered a candidate for the post processing stage.

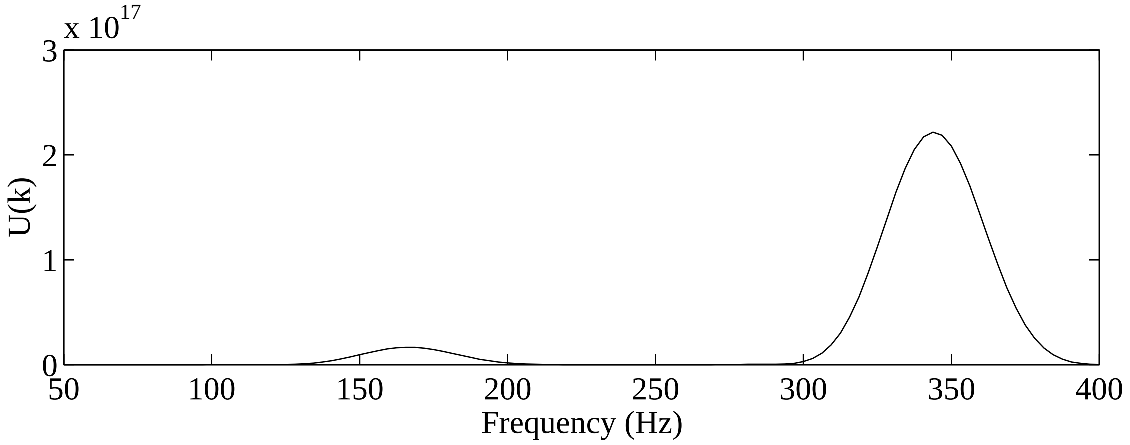


Figure 4.16: Squared Speech Power Spectrum $U(k)$

The fundamental is sometimes suppressed in this manner due to phase effects. Unlike many other pitch estimation algorithms (such as those based on autocorrelation functions) the phase of the speech spectrum affects the NLP algorithm. Phase effects can cause the product terms of the summation of (4.5) to add vectorially in a manner that partially cancels the maxima, or at least attenuates it compared to other local maxima of $U(k)=|Z(k)|^2$.

Consider a speech signal periodic in ω_0. A local maxima should occur at $Z(k_0)$, where $k_0 = \omega_0 D N_{dft} / 2\pi$. However, this local maxima may be attenuated if the complex terms of the summation of (4.5) add in anti phase. For example, consider the following expanded version of (4.5) with $X(0)X(k_0)=1$, $X(1)X(k_0-1)=e^{j\frac{2\pi}{3}}$, and $X(2)X(k_0-2)=e^{-j\frac{2\pi}{3}}$:

$$Z(k_0)=\frac{1}{N}\left[X(0)X(k_0)+X(1)X(k_0-1)+X(2)X(k_0-2)+\ldots\right] \tag{4.11}$$

In this case the first three terms would sum to zero. If this cancellation were to continue through all the terms of the summation, then $U(k)$ may be very small instead of the desired global maxima.

One approach often used to improve the pitch estimation process is pitch tracking. For example the MBE pitch estimation algorithm [28] evaluates the basic pitch extractor results for several frames in the past and future using a dynamic programming approach. This approach is useful for removing the effects of problems in the basic pitch extractor that are isolated to single frames. However the attenuation of the fundamental due to phase effects is usually not isolated to single frames, and may extend across many frames. Thus pitch tracking approaches that consider the basic extractor output from several adjacent frames may not be suitable for correcting the problem described above.

As a first order solution a form of backwards pitch tracking has been added to the algorithm with encouraging results. This algorithm adds to the list of candidates $\{k_1,\ldots,k_V\}$ the DFT bin corresponding to the fundamental estimate for the previous frame. Thus the previous frame's $F0$ is always tested by the post processing algorithm, even if no local maxima of sufficient magnitude is present in the current frame. This technique encourages continuity in the pitch track, but can make the pitch estimator slow to respond to transients, for example the start of new pitch tracks.

Algorithm	Hits	Misses	% Correct
NLP	553	50	91.7
NLP with tracker	531	72	88.1

Table 4.2: Evaluation of Backwards Pitch Tracker using Automatic Evaluation Method

The NLP algorithm with backwards pitch tracker was tested using the automatic method described earlier in this chapter. The results presented in Table 4.2 indicate that the pitch tracker actually slightly increases the number of gross errors. However subjective listening tests conducted on utterances prone to the phase problems indicate that the backwards pitch tracking does reduce the number of perceptually important errors, improving the overall subjective quality.

5. Generic Sinusoidal Coder

This chapter describes the generic sinusoidal coder developed for this thesis which has evolved from studies of existing sinusoidal and MBE coders. Features unique to this coder include techniques used in the analysis, spectral amplitude modelling, and synthesis stages.

The unquantised coder produces output speech of very high quality, in some cases almost indistinguishable from the original speech signal. This quality is achieved using a simple and computationally efficient algorithm, where both voiced and unvoiced speech is represented using the harmonic sinusoidal model.

Section 5.1 presents the sinusoidal analysis algorithm which extracts the sinusoidal model parameters from the input speech. Section 5.2 discusses computationally efficient synthesis algorithms based on overlap/add DFT structures. Section 5.3 presents the spectral magnitude modelling algorithm which uses a fixed number of Linear Prediction Coefficients (LPCs) to model the sinusoidal magnitudes. Speech files are available via the internet which demonstrate the algorithms developed in this chapter (Appendix B).

5.1 Analysis

Figure 5.1 illustrates the sinusoidal encoder algorithm. Input speech is windowed using a 220 sample tapered window, $w(n)$, before being transformed to the frequency domain using a 256 point DFT (computed efficiently using the FFT algorithm). The windowing and DFT of the input signal are performed using the procedure described in equations (3.5) to (3.7) in section 3.4 of this thesis.

Phases and amplitudes are estimated for each harmonic using the $F0$ estimate obtained from the Non Linear Pitch (NLP) algorithm described in Chapter 4. The frame rate is 10 ms, which was chosen to ensure the coder captures rapid changes in the time domain waveform. These are typically poorly reproduced in frequency domain coders due to the 20-30 ms frame rates usually employed.

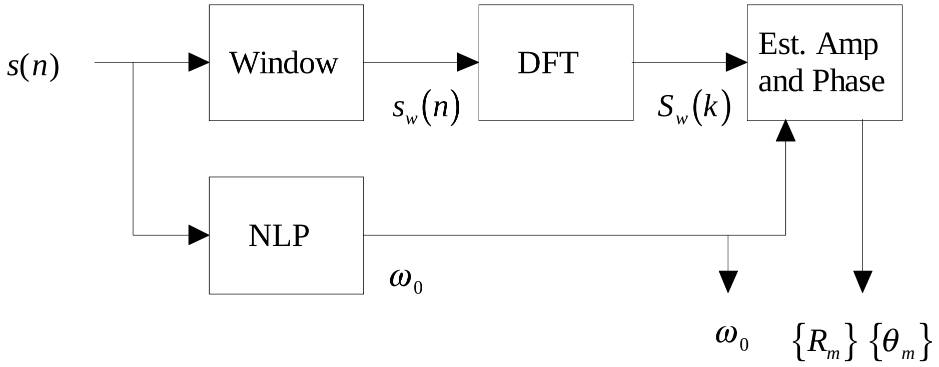


Figure 5.1: Block Diagram of Sinusoidal Encoder

Note that voicing is not explicitly represented, instead it is carried by the phase information. This removes the need for a voicing estimator. As with other parameter estimators (such as pitch), voicing estimators are never perfect, and are often a source of artefacts in speech coding algorithms, in particular in high acoustic background noise environments.

As discussed in section 3.3 of this thesis, most other sinusoidal and MBE vocoders employ a voicing estimator and explicitly transmit a voicing measure to the decoder. For example, the MBE vocoder divides the spectrum up into bands, and uses a single bit to represent the voiced or unvoiced nature of each band. Sinusoidal vocoders generally employ a two band model, where the lower frequencies are voiced and the upper frequencies are unvoiced. A transition frequency determines the changeover between unvoiced and voiced sections of the spectrum, and is usually defined with 3-4 bits.

The potential disadvantage of using the harmonic phases to convey voicing is an increase in bit rate. This problem is addressed in chapter 6 where several schemes designed to model the harmonic phases are presented and investigated. The phase modeling techniques presented in Chapter 6 apply a voicing measure to the coder but this is not fundamental to the operation of the unquantised coder, unlike other algorithms such as MBE [28]. For example, other phase modeling/quantisation algorithms could be employed (eg vector quantisation [60]) that do not require voicing measures or decisions.

To the best knowledge of the author, the concept of using the phase to convey voicing information in a *harmonic* sinusoidal coder is unique. Other authors [18] have used the generalised sinusoidal model (eg equation (3.1)) to convey voicing using the combination of sinusoidal frequencies and phases, however most sinusoidal and MBE vocoders employ voicing estimators as a fundamental part of operation, even in the unquantised state.

Thus the model employed is the harmonic sinusoidal model:

$$s(n)=\sum_{m=1}^{L}B_m\cos(\omega_0 mn+\theta_m) \tag{5.1}$$

where the parameters $\{B_m\}$ and $\{\theta_m\}$ represent the magnitudes and phases of the sinusoids, ω_0 is the normalised fundamental frequency in radians, and L is the number of harmonics given by $L=\lfloor \pi/\omega_0 \rfloor$.

In section 3.4 two methods for estimating the harmonic magnitudes and phases from the DFT of a frame of speech samples were presented. These methods are used in sinusoidal [21] and MBE [28] coders, two members of the harmonic sinusoidal coder family. A different method for estimating and representing harmonic magnitudes is presented here, that appears to be unique.

In this chapter the notation developed in section 3.4 is used, however the superscript denoting the frame is excluded for simplicity. Thus the N_{dft} point DFT of the l^{th} frame of input speech is $S_w(k)$, and the estimate of the fundamental frequency for the current frame is ω_0. The RMS magnitude, R_m for the m^{th} harmonic is defined as:

$$R_m=\left(\sum_{k=a_m}^{b_m-1}\left|S_w(k)\right|^2\right)^{\frac{1}{2}} \tag{5.2}$$

where a_m and b_m (given by (3.14) and (3.15) and repeated here for convenience):

$$a_m=\left\lfloor (m-0.5)\frac{\omega_0 N_{dft}}{2\pi}+0.5\right\rfloor \tag{5.3}$$

$$b_m=\left\lfloor (m+0.5)\frac{\omega_0 N_{dft}}{2\pi}+0.5\right\rfloor \tag{5.4}$$

represent the limits of the current harmonic. This method estimates the average magnitude of the current harmonic. It differs from other magnitude estimators such as those used in sinusoidal [21] and MBE [28] coders in several respects:

1. It will work equally well for voiced or unvoiced energy as it considers the entire energy in the current band and does not constrain it to be sinusoidal (which introduces a bias in favour of voiced bands).

2. It is relatively insensitive to small errors in the fundamental frequency estimate. The estimator in (5.2) merely requires that most of the energy in the band be contained in the interval between a_m and b_m.

The harmonic phases are estimated by sampling the DFT at the harmonic centres, using a method identical to that used by sinusoidal coders [21]:

$$\hat{\theta}_m = \arg\left[S_w\left(m\frac{\omega_0 N_{dft}}{2\pi} \right)\right] \tag{5.5}$$

5.2 Synthesis

The generic sinusoidal coder algorithm described in this chapter uses sinusoids to synthesise both voiced and unvoiced speech energy. Section 5.2.1 describes a procedure for determining the magnitudes of the sinusoids used to synthesise the speech signal $\{B_m\}$ given the RMS magnitudes $\{R_m\}$. An algorithm for performing the actual synthesis using an overlap add procedure to interpolate the synthesised signal from adjacent frames is described in Section 5.2.2.

5.2.1 Recovering Sinusoidal Magnitudes

To reconstruct the speech signal we need to estimate the harmonic magnitudes $\{B_m\}$ from the RMS magnitudes, $\{R_m\}$. Note that although the original signal in the region of the harmonic may have been voiced (ie a sinusoid), or noise, we use a sinusoid to synthesise the signal. Therefore we need to determine a suitable amplitude for the sinusoid, given the RMS magnitude of the energy in the band. This can be achieved by matching the energy of the synthesis sinusoid to the energy in the band obtained from R_m.

The energy of the sinusoid used to synthesise this signal can be expressed as:

$$E_m = \sum_{k=a_m}^{b_m - 1} \left| A_m W\left(k - m\frac{\omega_0 N_{dft}}{2\pi} \right) \right|^2 \tag{5.6}$$

where A_m is defined as the complex sinusoidal amplitude for the m^{th} band:

$$A_m = B_m \exp\left(j\theta_m \right) \tag{5.7}$$

and $W(k)$ is the DFT of the analysis window, $w(n)$. We assume that most of the energy in $W\left(k - m\frac{\omega_0 N_{dft}}{2\pi} \right)$ lies is the interval bounded by $k = a_m, \ldots, b_m - 1$ and that $W(k)$ is real. Therefore (5.6) can be approximated as:

$$E_m = \sum_{k=0}^{N_{dft}-1} \left| A_m \right|^2 \left| W(k) \right|^2 \tag{5.8}$$

$$E_m = \hat{B}_m^2 \sum_{k=0}^{N_{dft}-1} \left| W(k) \right|^2 \tag{5.9}$$

Note that B_m has been replaced by an estimate $\hat{B}_m$ due to the approximation of the limits of $W(k)$. Applying Parseval's theorem in the DFT domain:

$$\frac{1}{N} \sum_{k=0}^{N_{dft}-1} \left| W(k) \right|^2 = \sum_{n=0}^{N_{dft}-1} w^2(n) \tag{5.10}$$

we obtain:

$$E_m = N\hat{B}_m^2 \sum_{n=0}^{N_{dft}-1} w^2(n) \tag{5.11}$$

The energy in the band can be determined from the input signal as:

$$E_m = \sum_{k=a_m}^{b_m-1} \left|S_w(k)\right|^2 = R_m^2 \tag{5.12}$$

To match the measured energy in the band to the synthesised energy in the band we equate (5.11) and (5.12) and solve for $\hat{B}_m$:

$$\hat{B}_m = \frac{R_m}{\sqrt{N \sum_{n=0}^{N_{dft}-1} w^2(n)}} \tag{5.13}$$

Thus given the RMS magnitude for each band, we can determine an estimate of the sinusoidal magnitude required for synthesis. As the window $w(n)$ is constant, (5.13) reduces to:

$$\hat{B}_m = R_m N \tag{5.14}$$

if we normalise the window such that:

$$\sum_{n=0}^{N_{dft}-1} w^2(n) = \frac{1}{N} \tag{5.15}$$

The factor of N in (5.14) is useful for the DFT based synthesis algorithm, described in the next section.

5.2.2 Overlap Add Synthesis

Synthesis is achieved by constructing an estimate of the original speech spectrum using the sinusoidal model parameters for the current frame. This information is then transformed to the time domain using an Inverse DFT (IDFT). To produce a continuous time domain waveform the IDFTs from adjacent frames are smoothly interpolated using a weighted overlap add [18] procedure.

The estimate of the original speech spectrum is constructed using the sinusoidal model parameters:

$$\hat{S}(k) = N\sum_{m=1}^{L} \hat{A}_m \delta\left(k - m\frac{\omega_0 N_{dft}}{2\pi}\right), \quad k = 0,1,\ldots,\frac{N_{dft}}{2} \tag{5.16}$$

This signal represents the DFT of the synthesised speech signal, and consists of impulses spaced by ω_0 weighted by the complex harmonic amplitudes $\hat{A}_m$ where:

$$\hat{A}_m = \hat{B}_m \exp\left(j\hat{\theta}_m\right) \tag{5.17}$$

As we wish to synthesise a real time domain signal, $\hat{S}(k)$ is defined to be conjugate symmetric in the periodic DFT domain:

$$\hat{S}\left(N_{dft} - k\right) = \hat{S}^*(k), \quad k = 1,2,\ldots,\frac{N_{dft}}{2} - 1 \tag{5.18}$$

where $\hat{S}^*(k)$ is the complex conjugate of $\hat{S}(k)$. This signal is converted to the time domain using the IDFT:

$$\hat{s}_l(n) = \frac{1}{N}\sum_{k=0}^{N_{dft}-1} \hat{S}(k) e^{j\left(2\pi/N_{dft}\right)kn} \tag{5.19}$$

We introduce the notation $\hat{s}_l(n)$ to denote the synthesised speech for the l^{th} frame. To reconstruct a continuous synthesised speech waveform, we need to smoothly connect adjacent synthesised frames of speech. This is performed by windowing each frame, then shifting and superimposing adjacent frames using an overlap add algorithm. A triangular window is employed for this algorithm and is defined by:

$$t(n) = \begin{cases} n/N & 0 \le n < N \\ 1 - [n - N]/N & N \le n < 2N \\ 0 & otherwise \end{cases} \tag{5.20}$$

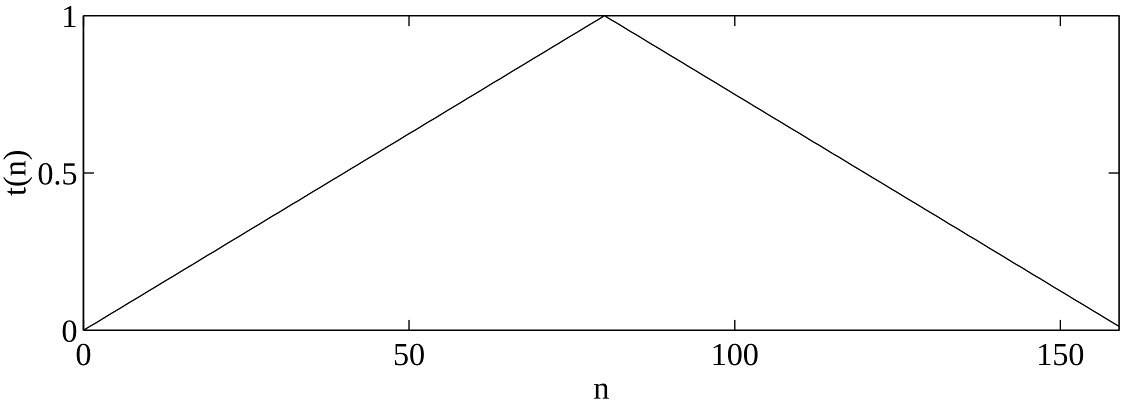


Figure 5.2: Triangular Synthesis Window

The frame size, $N = 80$, is the same as the encoder. Figure 5.2 illustrates the synthesis window employed for the decoder. The shape and overlap of the synthesis window is not important, as long as sections separated by the frame size (frame to frame shift) sum to 1:

$$t(n) + t(n-N) = 1 \tag{5.21}$$

The continuous synthesised speech signal $\hat{s}(n)$ for the l^{th} frame is obtained using:

$$\hat{s}(n+lN) = \hat{s}(n+(l-1)N) + \hat{s}_l\left(N_{dft} - N + 1 + n\right)t(n), \quad n = 0,1,\ldots,N-2 \tag{5.22}$$

$$\hat{s}(n+lN) = \hat{s}_l(n-N-1)t(n), \quad n = N-1, N, \ldots, 2N-1 \tag{5.23}$$

Note that the output for the current frame is $\hat{s}(n+lN)$, $n = 0,1,\ldots,N-1$, however storage must be provided for the future output samples $n = N, N+1, \ldots, 2N-1$ so that they can be added to samples from the next frame.

5.3 Parametric Magnitude Modelling

This section deals with the efficient modelling of the spectral magnitude parameters $\{B_m\}$ with a moderate order Linear Predictive model. The spectral magnitudes consume the greatest portion of the bit rate, thus compact modelling of the spectral magnitudes is an important research issue in sinusoidal coding.

Direct quantisation of spectral magnitudes is possible [28], but requires a large number of bits. Also, the number of spectral magnitudes is related to the fundamental frequency and therefore changes on a frame by frame basis. Thus direct quantisation would require a time varying bit allocation scheme.

Adjacent spectral magnitudes are highly correlated and tend to describe the vocal tract filtering action, thus the Linear Predictive Coding (LPC) model is a good choice for modelling spectral magnitudes in harmonic or sinusoidal coders [29][25][46][48][47]. After the LPC model is derived, the parameters may be easily transformed to Line Spectral Pair (LSP) frequencies for efficient quantisation and transmission.

Obtaining the LPC model for a given frame of speech may be achieved using time or frequency domain techniques. Frequency domain approaches [29][25][46] typically use the harmonic spectral magnitudes (ie some parameter set equivalent to $\{B_m\}$) to directly obtain the autocorrelation coefficients $R(k)$ which are then used to compute the LPC coefficients $\{a_k\}$ and gain term G. Time domain approaches such as [48] use ordinary time domain LPC analysis applied to the input speech signal. In either case, the spectral magnitudes are usually recovered at the decoder by sampling the LPC spectrum at the harmonic frequencies.

As the frequency domain approaches use the spectral magnitudes directly, some potential advantage exists compared to the time domain approaches which consider the entire signal including the unwanted effect of pitch on the magnitude structure of the LPC spectrum. However, the frequency domain schemes exhibit problems in arriving at an accurate LPC model when the number of harmonics is not large compared to the number of poles, for example in female speech [50]. Several techniques have been proposed to combat this problem, for example by using interpolation to produce a large number of spectral magnitudes [18] [29], or by modifying the LPC modelling cost function and using iterative techniques to minimise the resulting non-linear cost function [50][49].

The approach used here is to obtain the LPC model using time domain analysis, and to recover the spectral magnitudes at the decoder using the RMS average of the LPC spectrum in the band containing the harmonic instead of sampling the LPC envelope at the centre of the harmonic. The time domain analysis produces a set of p LPC coefficients $\{a_k\}$ and a LPC gain parameter G. The procedure used for determining these parameters is given in section 2.3 of this thesis, and is performed once per 10 ms frame.

Thus the RMS spectral magnitudes can be recovered by averaging the energy of the LPC spectrum over the region of each harmonic:

$$\hat{R}_m = \left(\sum_{k=a_m}^{b_m - 1} \left| H(k) \right|^2 \right)^{\frac{1}{2}} \tag{5.24}$$

where $H(k)$ is the N_{dft} point DFT of the LPC model for this frame (given by $H(z)$ in section 2.3).

The effect of the LPC model on the coder quality has been determined using objective and subjective tests. Objective results are determined by measuring the Signal to Noise Ratio (SNR) of the LPC modelling procedure defined as:

$$SNR_{dB} = 10 \log_{10} \left[\frac{\sum_{m=1}^{L} R_m^2}{\sum_{m=1}^{L} \left(R_m - \hat{R}_m \right)^2} \right] \tag{5.25}$$

The average SNR was determined for a database of 2400 frames containing 2 male and 2 female speakers for a range of LPC orders. The results are plotted (solid line) in Figure 5.3. For comparison, the average SNR obtained by sampling the LPC spectrum at the harmonic centres is also presented (dotted line). The sampling is achieved using the following expression:

$$\hat{R}_m = \left(\left| H\left(m \frac{\omega_0 N_{dft}}{2\pi} \right) \right|^2 \left(b_m - a_m \right) \right)^{\frac{1}{2}} \tag{5.26}$$

where the term $\left(b_m - a_m \right)$ given by (5.3) and (5.4) is a scale factor that is necessary for direct comparison between the RMS average and sampled cases.

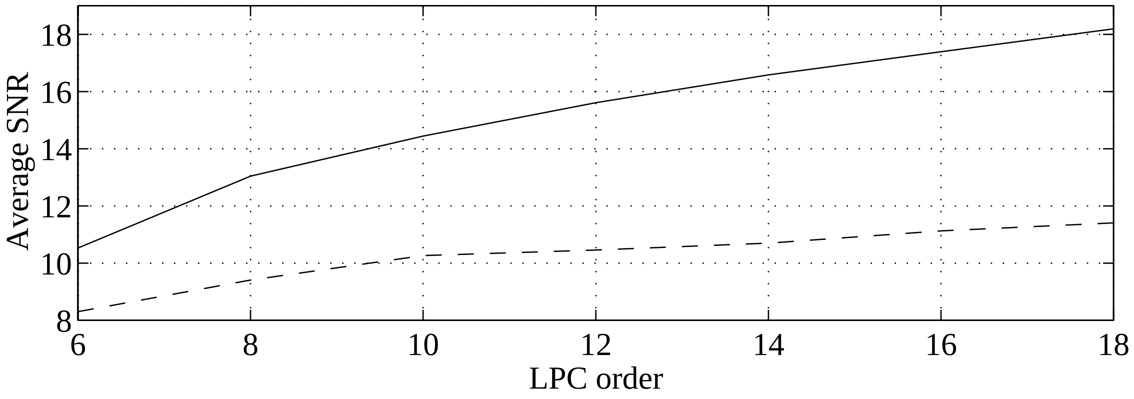


Figure 5.3: Average SNR against p for RMS (solid) and Sampled (dashed) $\left\{\hat{R}_m\right\}$

Before discussing the subjective testing it is useful to compare the baseline coder used in this work to those employed in the literature. The baseline generic coder speech quality is very high, nearly transparent in many cases. Therefore this coder will be extremely sensitive to any distortions introduced by modelling or quantisation. Note that this is in contrast to most of the other harmonic coders that employ LPC modelling, many of these exhibit communications quality speech *before* spectral modelling is employed and are therefore less sensitive to additional distortion introduced by spectral modelling. Thus the requirements for transparent spectral modelling in the case of this coder are probably higher than many of the other coders in the literature.

Informal subjective tests were conducted by processing several test databases and performing listening tests with several listeners through high quality headphones. A model order of $p = 12$ was used for the subjective tests, as this performed well in the objective tests. The LPC modelling stage introduces some noticeable distortion into the output speech, however the distortions are relatively minor and the overall speech quality remains high. A small amount of bass reverberation was heard for very low pitch male speakers, this may be more apparent through loudspeakers, however it is not a concern for band pass applications such as speech through telephone networks.

The subjective tests support the objective tests in the choice of recovery methods for the spectral magnitudes. The RMS average scheme introduces less distortion for female speakers than the sampled scheme. For males speakers the two schemes both perform well.

The objective and subjective results above show that the RMS average method is superior to the direct sampling method, in particular for female speakers. With high pitched female speech, there are fewer harmonics. Consequently, the LPC envelope sometimes attempts to model individual harmonics, instead of providing a smooth spectral envelope that traces through the harmonic peaks. Thus for female speech, the LPC spectrum sometimes models pitch structure as well as spectral envelope information. It is suggested that the RMS average method smooths out any pitch structure in the LPC envelope that produces distortion with the direct sampling method.

From these results it was determined that a $p = 12$ LPC model provides good results. The small trade off in speech quality was judged to be acceptable given the advantage in quantisation that the LPC modelling provides.

6. Parametric Phase Modelling

This chapter describes several parametric models for the compact representation of the harmonic phases, $\{\theta_m\}$. These models are applied to the baseline harmonic sinusoidal coder presented in chapter 5 and evaluated using objective and informal subjective methods. Speech files are available via the internet which demonstrate the algorithms developed in this chapter (Appendix B). Compared to the near transparent baseline coder that uses original phases, the speech quality is degraded to near toll quality (eg VSELP) for clean speech.

The human ear is often described as being relatively insensitive to the short term phase of speech. This argument has been used in several speech coding algorithms to discard the measured phase information entirely. It is then reconstructed at the decoder using techniques that synthesise the phase function, for example using rule [12], or model based approaches [22].

In practice, phase information is important for high quality synthesised speech. While discarding the phase information does not significantly reduce the intelligibility of speech, it can introduce some unpleasant artefacts. In particular, low pitched speakers such as males suffer from reverberation when the harmonic phase information is removed. Thus, for high quality harmonic coding, transmission of phases in some form is necessary.

As with the harmonic magnitudes, the number of harmonic phases, L, varies from frame to frame. If the phases are directly quantised, this can lead to a prohibatively high bit rate. Thus several previous authors have implemented systems that only quantise the low order harmonic phases, substituting uniformly distributed random phases for the high order harmonics [30][17].

Several authors have developed schemes that model amplitude and phase simultaneously, for example using ARMA techniques [43], or combinations of ARMA and vector quantisation [42][44]. In general, the ARMA schemes described assume minimum phases systems. Most of the sinusoidal and MBE coder work has discarded the phase information, synthesising it at the receiver using rule [12], or model based approaches [22].

The work presented here is based on similar assumptions to [41], that speech is not, in general minimum phase. However unlike [41], the magnitude and phase information are treated separately, resulting in three phase modelling schemes based on a general cascade minimum phase/all pass system, excited by an impulse at time n_0. Of these schemes, two are shown to produce high quality speech using a small number of model parameters, with moderate to low computational complexity. The number of model parameters is fixed, despite the varying number of harmonics from frame to frame. For clean input speech, the synthesised speech quality exceeds that of VSELP.

Section 6.1 describes the development of a phase model for voiced speech that consists of a minimum phase LPC synthesis filter cascaded with an all pass filter. It is shown that the all pass filter has a strong linear phase component corresponding to the time shift of the original signal, and a phase residual component. Three candidate systems are then developed that implement this model. The first, described in section 6.2, uses a CELP type analysis by synthesis loop to determine the model parameters. The second system discussed in section 6.3 quantises the phase residual using truncated Discrete Cosine Transform (DCT) coefficients. The third uses a least squares polynomial fit to the phase residual, that is weighted using the harmonic magnitudes (section 6.4).

To evaluate the three phase modelling schemes an objective measure is developed in section 6.5, while techniques used to treat unvoiced speech are presented in section 6.6. Finally, results of informal listening tests are presented in section 6.7.

6.1. Minimum Phase/All Pass Filter Model

One aim of this research is to produce a model capable of accurately representing the harmonic magnitudes $\{B_m\}$, and phases, $\{\theta_m\}$. For convenience, these parameters will be combined into the complex amplitudes $\{A_m\}$, where:

$$A_m = B_m e^{j\theta_m} \tag{6.1}$$

As discussed previously, the all pole LPC model is an efficient way to represent the spectral magnitude information using a small number of parameters. However, normal LPC analysis constrains the LPC synthesis filter to be minimum phase. There is considerable difference of opinion in the literature as to whether minimum phase systems are suitable for modelling speech. In [41] it is argued that speech signals, in general, are not minimum phase, and that moderate order (eg 14th) mixed phase systems are a more suitable choice. In this case, non-causal AR models were fitted to harmonic model magnitudes $\{B_m\}$, and phases, $\{\theta_m\}$ simultaneously. Good results were reported using a 14th order mixed phase model for voiced speech.

It should be noted at this point that several previous authors have used high order minimum-phase systems successfully to model both the harmonic phases and magnitudes. For example, [22] used a high order (24-28 coefficient) cepstral model, and [24] used a high order (18-32 coefficient) LPC model. The cepstral coefficients were found to fit the magnitude spectrum of the speech, and the phase determined using the assumption that for minimum phase signals, the phase and magnitude are related. The authors reported that lowering the model order introduced unpleasant artefacts into the synthesised speech, such as reverberation and muffling.

It is suggested that high order minimum phase systems can approximate the phase spectrum of a non-minimum phase system. However, in the previous chapter we have established that a 12th order LPC model is sufficient to represent the magnitude spectrum of the speech, and raising the order significantly would seem wasteful.

Therefore we require a modelling scheme that can represent non-minimum phase signals with a spectral magnitude component defined by a moderate order all pole LPC filter. Any system, $H(z)$, can be modelled as the cascade of a minimum phase filter and an all pass filter [5]:

$$H(z) = H_{\min}(z)H_{ap}(z) \tag{6.2}$$

Thus our task is to determine a suitable all pass filter, such that when cascaded with a minimum phase LPC synthesis filter $H_{\min}(z) = 1/A(z)$, produces a "close" match to the original harmonic model parameters. For this analysis the original model parameters A_m and ω_0 are assumed to be known. The modelled parameters may be obtained by sampling $H(z)$ at the harmonic frequencies:

$$\hat{A}_m = H\left(e^{jm\omega_0}\right) \tag{6.3}$$

where $\hat{A}_m = \hat{B}_m e^{j\hat{\theta}_m}$. A suitable cost function is therefore:

$$E = \sum_m \left| A_m - \hat{A}_m \right|^2 \tag{6.4}$$

As discussed in the previous chapter, the subjective distortion introduced by LPC modelling of the spectral magnitudes is small. It is therefore reasonable to assume $B_m = \hat{B}_m$, in which case (6.4) reduces to:

$$E = \sum_m B_m^2 \left| e^{j\theta_m} - e^{j\hat{\theta}_m} \right|^2 \tag{6.5}$$

It is interesting to examine the harmonic model parameters in the time and frequency domain. Figures 6.2 and 6.3 illustrate the harmonic model parameters for the input speech signal in Figure 6.1. Figure 6.4 is the inverse Fourier transform of the harmonic model parameters, denoted as a "prototype" after [40]. This signal is periodic in P samples, several cycles are plotted in Figure 6.4. Note the strong likeness between the original speech and the time domain prototype.

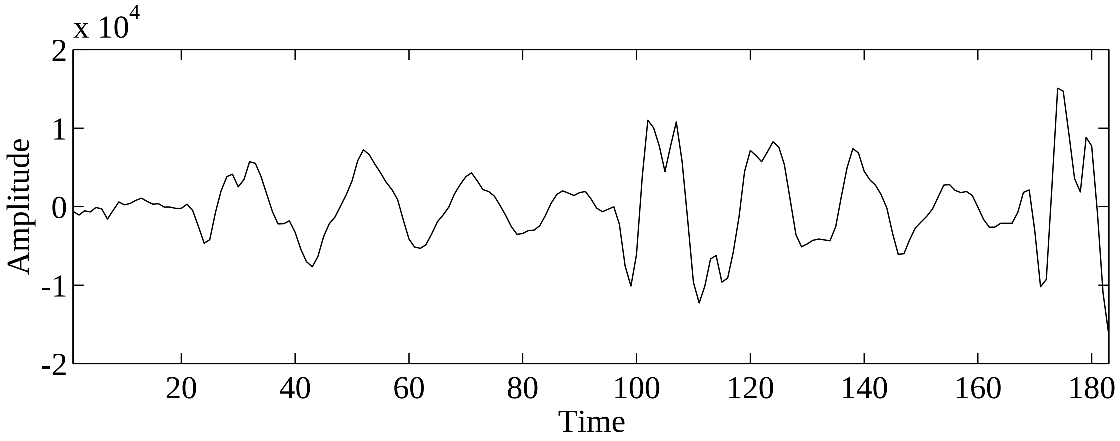


Figure 6.1: Original Speech

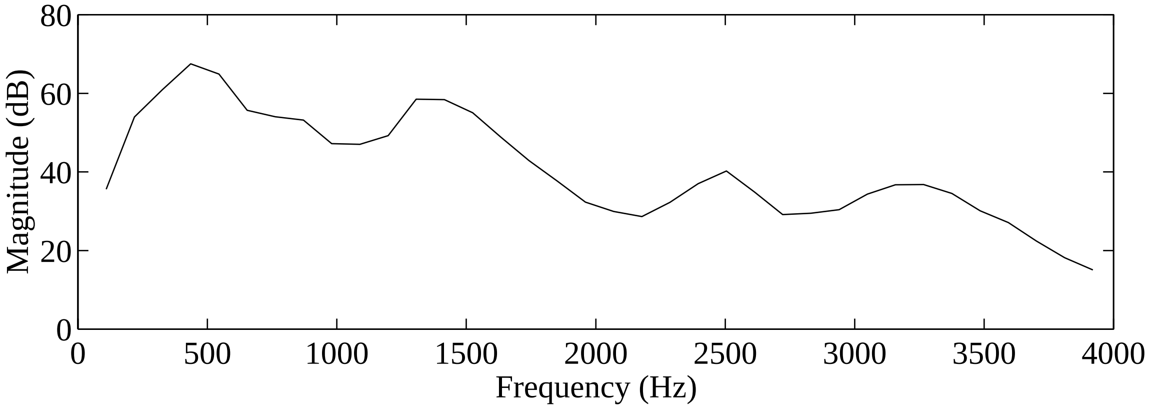


Figure 6.2: Harmonic Magnitude Samples $\{B_m\}$

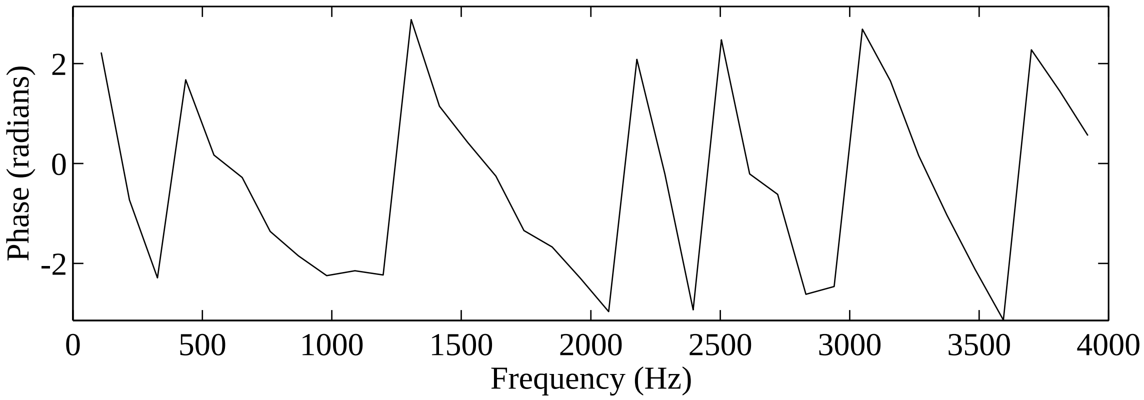


Figure 6.3: Harmonic Phase Samples $\{\theta_m\}$

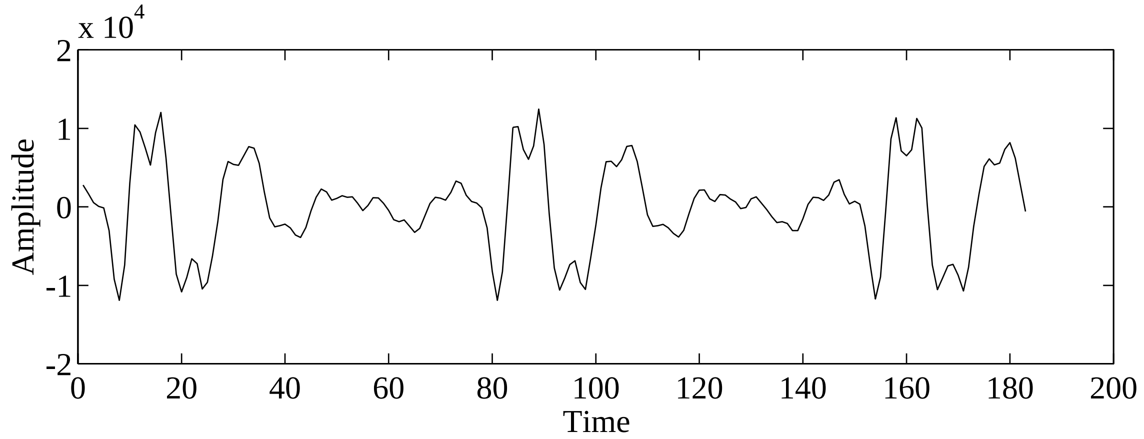


Figure 6.4: Time Domain Prototype

Note the strong linear phase component in Figure 6.3. A linear phase shift in the frequency domain corresponds to a time shift in the time domain. We can thus model the prototype as the time shifted impulse response , $h(n-n_0)$, of a non-minimum phase filter $H(z)$, where n_0 is the impulse position. This approach is similar to [41]. A related approach is presented in [19] by considering the vocal tract filter $H(z)$ to be excited by a steady series of impulses during constant voiced speech. A pitch pulse occurs at time n_0 when all the sine waves in the harmonic model add coherently.

6.2. Analysis by Synthesis Phase Model

This section describes a first order approximation to the all pass filter $H_{ap}(z)$ comprised of a linear phase component defined by the onset time n_0 and a phase term specified by a *complex* gain $G(\omega)$. These parameters can be found using an analysis by synthesis technique, similar to the codebook search used in CELP.

The modelled complex harmonic amplitudes are defined as:

$$\hat{A}_m = \frac{G(m\omega_0)e^{-jn_0 m\omega_0}}{A\left(e^{jm\omega_0}\right)} \tag{6.6}$$

This has a phase component:

$$\hat{\theta}_m = -\arg\left[A\left(e^{jm\omega_0}\right)\right] + \arg\left[G(m\omega_0)\right] - n_0 m\omega_0 \tag{6.7}$$

Both $\hat{A}_m$ and $\hat{\theta}_m$ are defined over the range $m = 1,\ldots,L$. Note the distinction between the LPC analysis filter $A(z)$, and the complex sinusoidal model parameters $\{A_m\}$.

To produce a real synthesised time domain signal, $G(\omega)$ is considered to be conjugate symmetrical about the frequency axis:

$$G(\omega) = \begin{cases} G_0 e^{j\varphi} & \omega > 0 \\ G_0 e^{-j\varphi} & \omega < 0 \end{cases} \tag{6.8}$$

where G_0 is a real positive constant and φ is a real constant. Thus the phase of $G(\omega)$ is constant for all positive and negative frequencies, but changes sign at $\omega = 0$.

For analysis purposes it is only necessary to consider the positive frequencies, ie $m = 1, \ldots, L$. In this case we represent the complex gain as a constant, $G = G_0 e^{j\varphi}$. Equation (6.6) therefore models the all pass component of the phase spectrum as having a constant component φ, and a linear component described by n_0. A cost function can be defined in terms of these parameters by substituting (6.6) into (6.4) with $G(m\omega_0) = G$:

$$E(n_0, G) = \sum_{m=1}^{L} \left| A_m - \frac{G e^{-jn_0 m\omega_0}}{A\left(e^{j\omega_0 m}\right)} \right|^2 \tag{6.9}$$

The model parameters can be estimated using an exhaustive analysis by synthesis procedure, similar to that used for CELP codebook searching. For a range of impulse positions, $n_0 = 0, 1, \ldots, P-1$, the optimal complex gain G is determined. The error can then be determined for each n_0 by evaluating (6.9). After all n_0 positions have been evaluated, the value that minimises (6.9) is chosen, along with the corresponding complex gain G.

The range of n_0 positions can be limited to the pitch period P as the time and frequency domain forms of $\hat{A}_m$ are periodic in $P = \lfloor 2\pi/\omega_0 + 0.5 \rfloor$. An n_0 resolution of 1 sample was found to be adequate by experiment.

The complex gain G for a given n_0 can be obtained using a least squares fit (Appendix A.18) as:

$$G = \frac{\mathbf{a}^H \hat{\mathbf{c}}}{\hat{\mathbf{c}}^H \hat{\mathbf{c}}} \tag{6.10}$$

where:

$$\mathbf{a}^T = \left[A_1 \ldots A_L\right] \tag{6.11}$$

$$\hat{\mathbf{c}}^T = \left[\hat{C}_1 \ldots \hat{C}_L\right] \tag{6.12}$$

$$\hat{C}_m = \frac{e^{-jn_0 m\omega_0}}{A\left(e^{j\omega_0 m}\right)} \tag{6.13}$$

Figures 6.5 to 6.8 illustrate the operation of the analysis by synthesis phase model for a single frame of male speech. A 12th order LPC model was used for $H_{\min}(z)$. Original magnitudes are used, so all errors are due to the phase modelling. Figure 6.5 is a plot of the original and modelled phase spectrums. Note the close match at the low frequency end of the spectrum, but somewhat poorer match at higher frequencies. This effect is also evident in Figure 6.8, a plot of the original and error (modelling noise) magnitude spectrums. The error magnitude spectrum is defined as:

$$|E(m)|^2 = |A_m - \hat{A}_m|^2 \tag{6.14}$$

for $m = 1,\ldots,L$. The low frequency, high energy regions of the spectrum have a smaller error than the high frequency, low energy regions. These effects are due to the error energy minimising properties of the cost function, equation (6.9).

Figures 6.6 and 6.7 show the original and modelled prototypes, note the close match in these time domain waveforms, due to the error energy minimising properties of the cost function. The low energy, high frequency error energy is less visible in the time domain plots than the error magnitude spectrum in Figure 6.8.

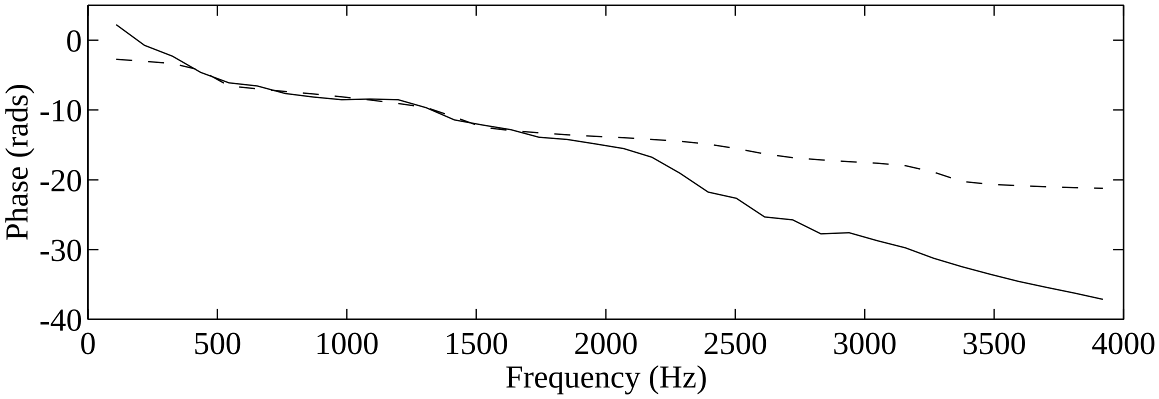


Figure 6.5: Original (solid) and Modelled (dashed) Phase Spectrum

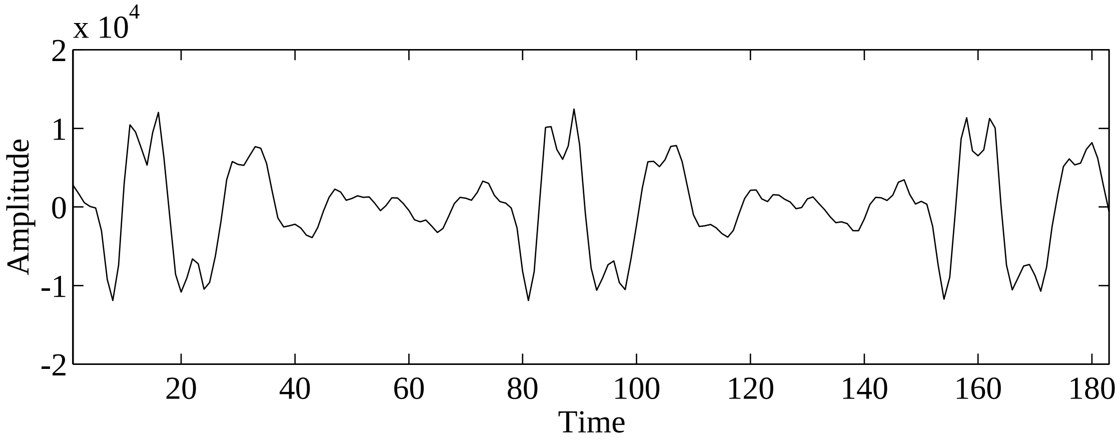


Figure 6.6: Original Prototype

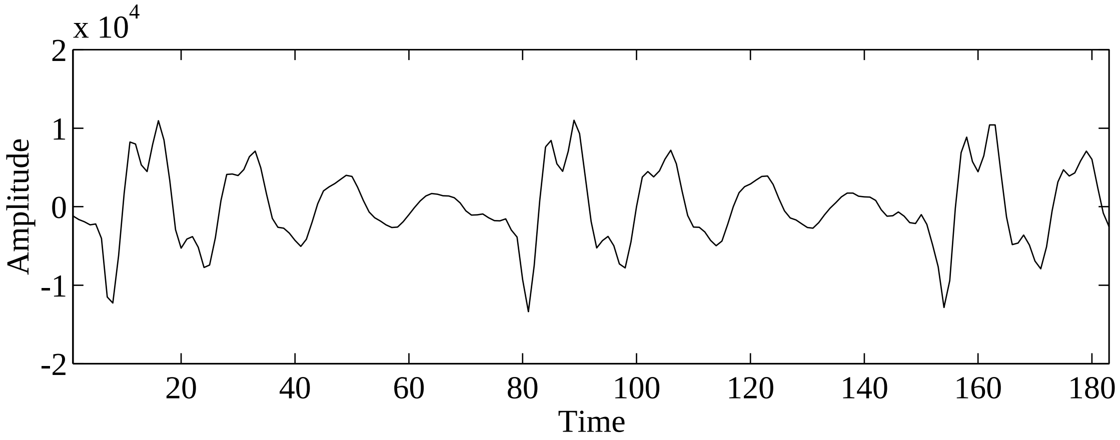


Figure 6.7: Modelled Prototype

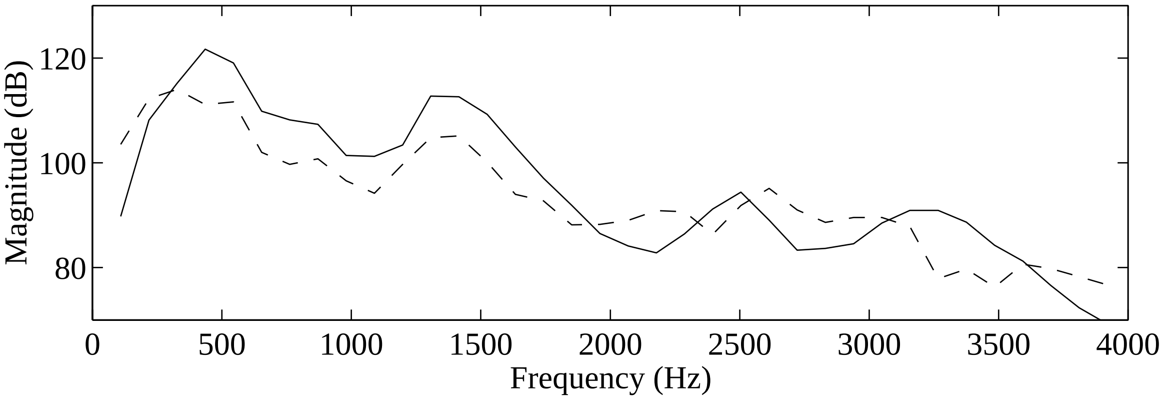


Figure 6.8: Original (solid) and Error (dashed) Magnitude Spectrum

6.3. Truncated DCT Phase Model

This section presents another approach to the phase modelling problem. In this case the synthesised speech is modelled as a cascade of a minimum phase LPC synthesis filter and an all pass filter. The system is excited by a unit impulse at time n_0, and the modelled complex amplitudes are recovered by sampling the modelled spectrum at the harmonic frequencies. The all pass filter is defined by its phase spectrum, a function of frequency.

The modelled complex harmonic amplitudes are defined as:

$$\hat{A}_m = \frac{g e^{-jm\omega_0 n_0} H_{ap}\left(e^{jm\omega_0}\right)}{A\left(e^{jm\omega_0}\right)} \tag{6.15}$$

where g is a real gain factor and $\left|H_{ap}\left(e^{jm\omega_0}\right)\right| = 1$. The LPC filter coefficients and gain g can be found using standard time domain LPC analysis. For this phase modelling technique, we assume that the LPC model represents the harmonic magnitudes adequately, therefore a cost function of the form (6.5) can be used.

This phase model is defined by the parameters n_0 and $H_{ap}(z)$. Note that compared to the previous section, the gain factor g only describes the magnitude of the system, not the magnitude and constant phase term. In this case, the constant phase component is absorbed into the all pass filter $H_{ap}(z)$.

The phase model parameters can be sub-optimally found using a sequential process. First the dominant linear phase component described by n_0 is removed, and the remaining *phase residual* component modelled using $H_{ap}(z)$.

To determine n_0, a cost function can be defined that represents the modelled phase as the LPC synthesis filter excited by the shifted impulse (ie we have removed the all pass filter for this stage of the minimisation):

$$E(n_0) = \sum_{m=1}^{L} B_m{}^2 \left| e^{j\theta_m} - \frac{e^{-jm\omega_0 n_0}\left|A\left(e^{jm\omega_0}\right)\right|}{A\left(e^{jm\omega_0}\right)} \right|^2 \tag{6.16}$$

Note the normalising term $|A(z)|$, this is included as we are interested in only the phase of $1/A(z)$. The impulse position n_0 is determined by sampling (6.16) in the range $n_0 = 0,1,\ldots,P-1$ to determine the minima. The phase residual ϕ_m can then be determined as:

$$\phi_m = \arg\left[\frac{A_m A\left(e^{jm\omega_0}\right)}{e^{-jm\omega_0 n_0}}\right] \tag{6.17}$$

for $m = 1,\ldots,L$. As the dominant linear component has been removed, the dynamic range of $\{\phi_m\}$ is generally much smaller than that of the harmonic phases $\{\theta_m\}$. More importantly, for voiced speech, adjacent values in the set $\{\phi_m\}$ are usually highly correlated. This suggests that higher coding efficiencies may be obtained by coding $\{\phi_m\}$ than $\{\theta_m\}$. The phase residual spectrum $\{\phi_m\}$ is plotted (solid) in Figure 6.9. Compare this to the original phase spectrum from the same frame in Figure 6.5.

The phase residual spectrum can be considered to be the ideal phase spectrum of $H_{ap}(z)$. That is, if we can devise a method to model the phase residual perfectly, then we can reconstruct the original phases.

The high correlation of the phase residual suggests that a decorrelation procedure might be used to compactly model this signal. The Discrete Cosine Transform (DCT) was therefore employed, due to it's well known near optimum decorrelation properties [7]. The following form of the DCT was used:

$$V(k) = \sum_{m=0}^{L-1} \phi_{m+1} c(k) \cos\left(\frac{(2m+1)\pi k}{2L}\right) \quad k = 0,1,\ldots L-1 \tag{6.18}$$

$V(k)$ represents the DCT of ϕ_m and $c(k)$ is defined as:

$$c(k) = \begin{cases} 1 & k = 0 \\ \sqrt{2} & k = 1,2,\ldots L-1 \end{cases} \tag{6.19}$$

The Inverse DCT (IDCT) is defined as:

$$\hat{\phi}_{m+1} = \frac{1}{L}\sum_{k=0}^{L-1} V(k)c(k)\cos\left(\frac{(2m+1)\pi k}{2L}\right) m = 0,1,\ldots,L-1 \tag{6.20}$$

The DCT coefficients $V(k)$ of the phase residual in Figure 6.9 are plotted in Figure 6.10. Note the dominance of the low order coefficients at the left hand side of Figure 6.10, indicating a highly correlated signal. A reasonable assumption is that a subset of the DCT coefficients:

$$\hat{V}(k) = \begin{cases} V(k) & k \le C \\ 0 & k > C \end{cases} \tag{6.21}$$

where $C \le L$ may be sufficient to represent the phase residual. The modelled phase residual obtained from the truncated DCT coefficients $\hat{V}(k)$ (with $C = 5$) is plotted (dashed) in Figure 6.9. The modelled phase residual is a smoothed version of the original phase residual, due to the truncation of the DCT coefficients. The general shape of the original phase residual is preserved.

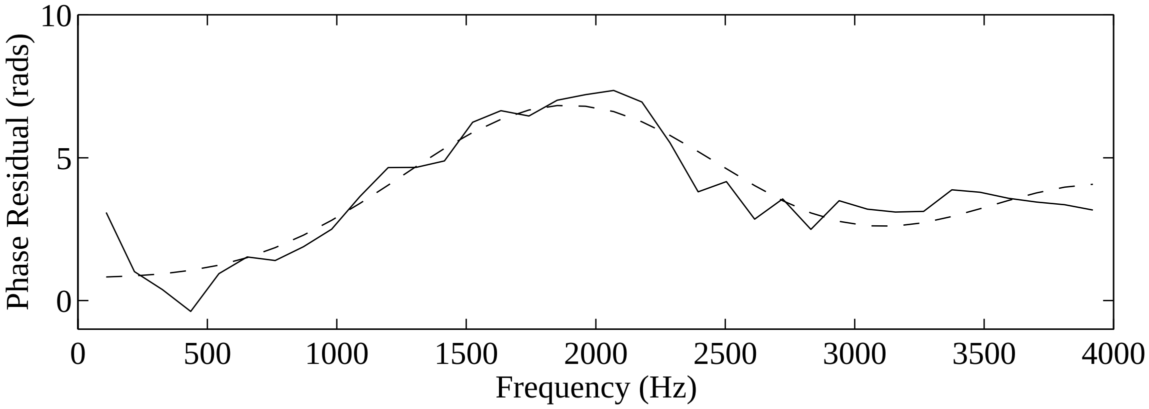


Figure 6.9: Original (solid) and Truncated DCT modelled (dashed) Phase Residual Spectrums

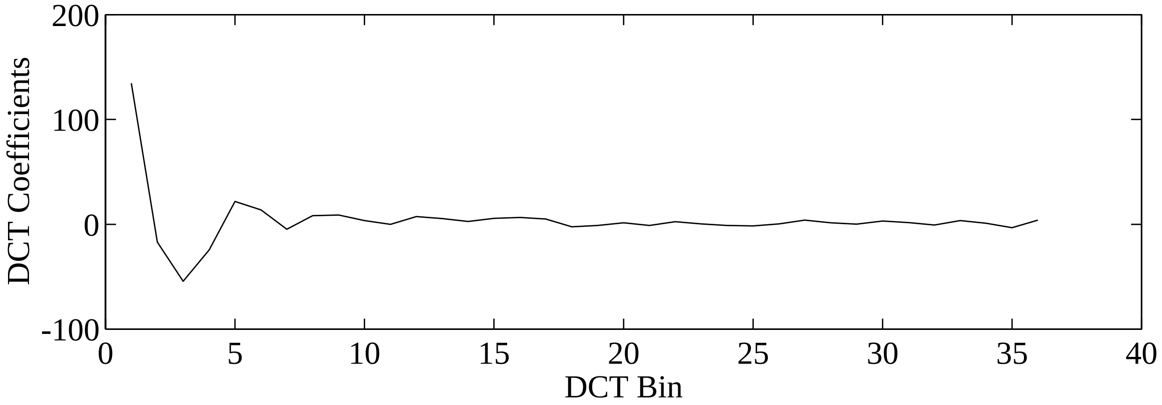


Figure 6.10: DCT Transform of Phase Residual

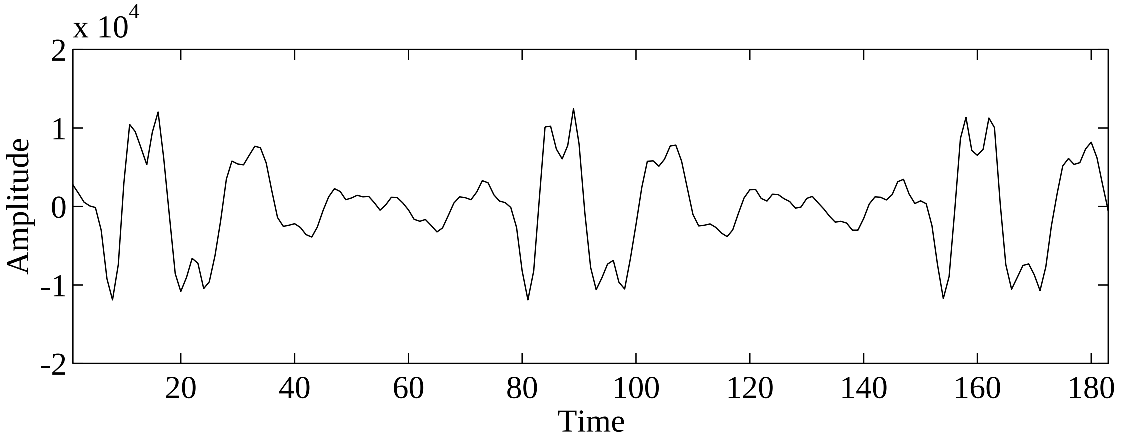


Figure 6.11: Original Prototype

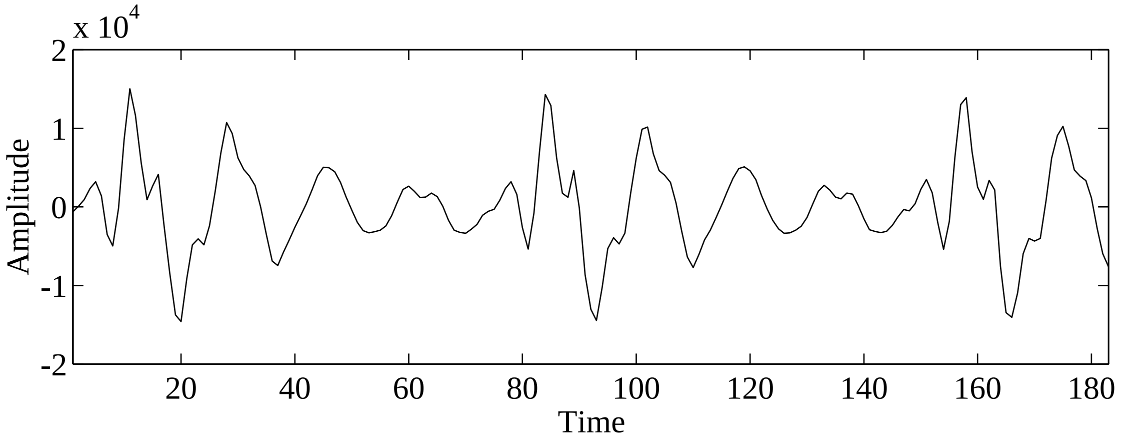


Figure 6.12: Modelled Prototype

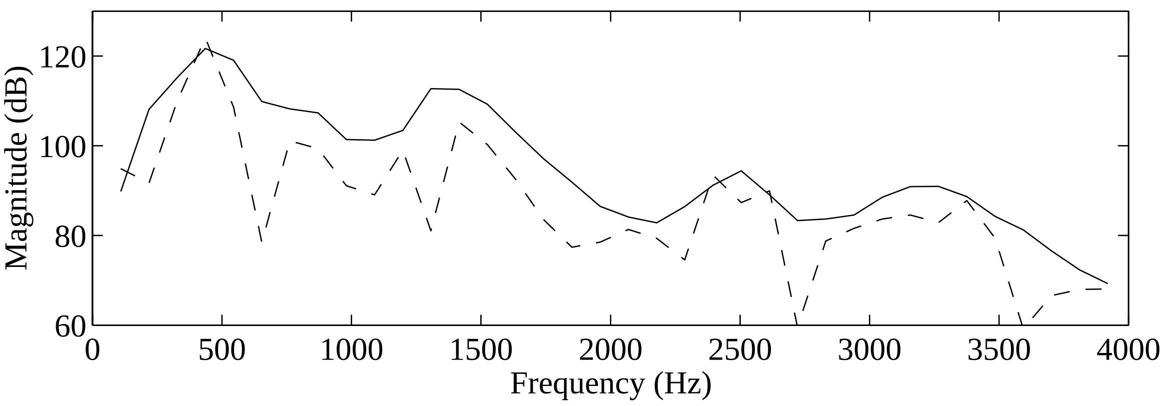


Figure 6.13: Original (solid) and Error (dashed) Magnitude Spectrums

The effectiveness of this phase modelling scheme can be gauged by examining the modelled prototype in Figure 6.12 and the error magnitude spectrum in Figure 6.13. Both figures indicate that this scheme performs poorly. Truncating the DCT coefficients does not take into account the relative importance of different parts of the spectrum. For example, high energy regions of the spectrum contribute a larger error energy for a given phase error than low energy regions.

6.4. Weighted Polynomial Phase Model

The previous section highlighted the need for a phase modelling scheme that takes into account the relative error contribution of different parts of the spectrum. The *analysis by synthesis* phase modelling technique (first scheme presented) is such a method, however this scheme is limited to a first order approximation of the phase spectrum (linear plus constant phase term).

This section introduces a third phase modelling scheme that fits a *weighted* K^{th} order polynomial to the phase residual spectrum. The polynomial fit is weighted to provide a better fit in perceptually important high energy areas of the spectrum than in low energy areas. In other respects, this scheme is similar to the truncated DCT technique, an impulse at time n_0 excites a cascade of a minimum phase LPC synthesis filter and an all pass filter. However in this case, the all pass filter phase spectrum (phase residual spectrum) is described by a K^{th} order polynomial.

As in the previous section, a two stage analysis procedure is employed. First, the dominant linear phase term described by n_0 is removed to obtain the phase residual samples $\{\phi_m\}$. This procedure is described in the previous section. Next, the K^{th} order polynomial is fitted to the the phase residual samples $\{\phi_m\}$.

To derive the polynomial fitting procedure a cost function in the form of (6.5) is used.

$$E = \sum_{m=1}^{L} B_m^2 \left| e^{j\theta_m} - e^{j\hat{\theta}_m} \right|^2 \tag{6.22}$$

Expressing in terms of the linear phase and phase residual components:

$$E\left(n_0, \hat{\phi}_1, \ldots, \hat{\phi}_L\right) = \sum_{m=1}^{L} B_m^2 \left| e^{-jn_0 m\omega_0} \left(e^{j\phi_m} - e^{j\hat{\phi}_m} \right) \right|^2 \tag{6.23}$$

which can be reduced to:

$$E\left(\hat{\phi}_1, \ldots, \hat{\phi}_L\right) = \sum_{m=1}^{L} B_m^2 \left| e^{j\phi_m} - e^{j\hat{\phi}_m} \right|^2 \tag{6.24}$$

as the linear phase component has unit magnitude. Expanding (6.24) for positive and negative sides of the spectrum (and assuming the DC term $B_0 = 0$):

$$2E\left(\hat{\phi}_1, \ldots, \hat{\phi}_L\right) = \sum_{m=-1}^{-L} B_m^2 \left| e^{j\phi_m} - e^{j\hat{\phi}_m} \right|^2 + \sum_{m=1}^{L} B_m^2 \left| e^{j\phi_m} - e^{j\hat{\phi}_m} \right|^2 \tag{6.25}$$

Due to the conjugate symmetry of the phases it can be shown that:

$$E\left(\hat{\phi}_1, \ldots, \hat{\phi}_L\right) = 2\sum_{m=1}^{L} B_m^2 \left(1 - \cos\left(\phi_m - \hat{\phi}_m\right)\right) \tag{6.26}$$

A similar result is presented in [42]. For small angles, a first order Taylor series approximation for $\cos(x)$ is:

$$\cos(x) = 1 - \frac{x^2}{2} \tag{6.27}$$

This approximation is coarse for large angles, however it allows a tractable solution to the least squares problem, and as will be shown, produces good results. Substituting this approximation into (6.26) leads to an elegant least squares expression:

$$E = \sum_{m=1}^{L} B_m^2 \left(\phi_m - \hat{\phi}_m \right)^2 \tag{6.28}$$

Thus to minimise E, the error between the original and synthesised prototypes, we must minimise the squared phase error weighted by the energy of each harmonic. Consider the K^{th} order polynomial approximation to ϕ_m:

$$\hat{\phi}_m = c_0 + c_1 m + c_2 m^2 + \ldots + c_K m^K \tag{6.29}$$

The error at the m^{th} harmonic can be expressed using vectors as:

$$e_m = B_m \left(\phi_m - \mathbf{p}(m)^T \mathbf{c} \right) \tag{6.30}$$

where:

$$\mathbf{c}^T = \left[c_0 \ldots c_K \right] \tag{6.31}$$

$$\mathbf{p}(m)^T = \left[1 \; m \; m^2 \ldots \; m^K \right] \tag{6.32}$$

It is possible to express the error at all harmonics:

$$\mathbf{e} = \mathbf{Bq} - \mathbf{BMc} \tag{6.33}$$

where:

$$\mathbf{B} = \begin{bmatrix} B_1 & & & & \\ & B_2 & & 0 & \\ & & . & & \\ & 0 & & . & \\ & & & & B_L \end{bmatrix} \tag{6.34}$$

$$\mathbf{q}^T = \left[\phi_1 \; \phi_2 \ldots \phi_L \right] \tag{6.35}$$

$$\mathbf{e}^T = [e_1 \ldots e_L] \tag{6.36}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{p}(1)^T \\ \mathbf{p}(2)^T \\ \vdots \\ \vdots \\ \mathbf{p}(L)^T \end{bmatrix} \tag{6.37}$$

Comparing (6.33) to (6.28) we can see that minimising E is equivalent to minimising $\|\mathbf{e}\|$. Using the results in Appendix A, we can see that minimising $\|\mathbf{e}\|$ is the "best-fit" solution to the over determined set of equations (6.33). From equation A1.8, this solution is:

$$\mathbf{c} = \left(\mathbf{M}^T\mathbf{B}^T\mathbf{B}\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{B}^T\mathbf{B}\mathbf{q} \tag{6.38}$$

This expression can be reduced to the problem of solving the matrix equation:

$$\mathbf{R}\mathbf{c} = \mathbf{u} \tag{6.39}$$

where $\mathbf{R}$ is a $K+1$ by $K+1$ matrix:

$$\mathbf{R} = \mathbf{M}^T\mathbf{B}^T\mathbf{B}\mathbf{M} \tag{6.40}$$

$$R_{ij} = \sum_{i=1}^{L} B_i^2 i^{k+j} \tag{6.41}$$

and $\mathbf{u}$ is a $K+1$ element column vector:

$$\mathbf{u} = \mathbf{M}^T\mathbf{B}^T\mathbf{B} \tag{6.42}$$

$$u_j = \sum_{i=1}^{L} B_i^2 \phi_i i^j \tag{6.43}$$

Figures 6.14 to 6.17 illustrate the operation of the weighted polynomial phase modelling technique. The model order is $K = 4$. Note the close match between the modelled and original phase residual in Figure 6.14 in the high energy areas. Also, the original and modelled prototypes in Figures 6.15 and 6.16 are very similar. Figure 6.17 shows that the error energy is suppressed in the high energy formant regions.

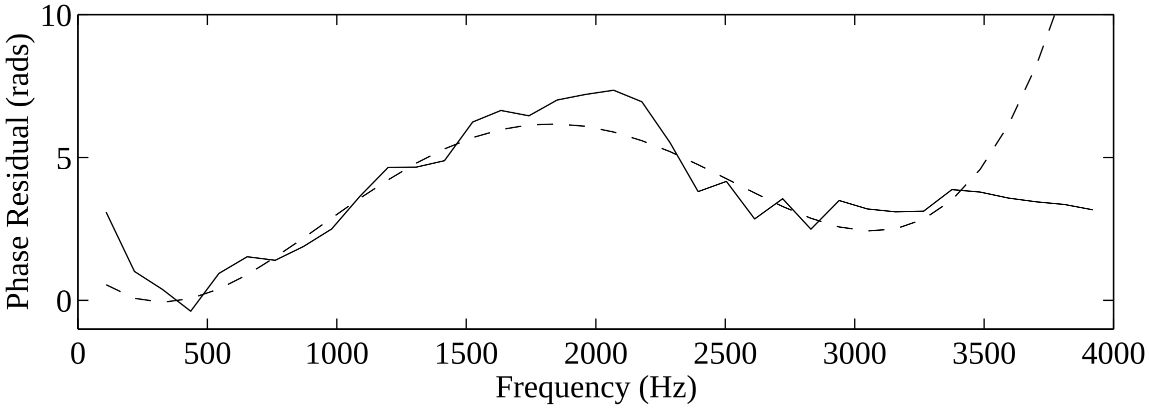


Figure 6.14: Original (solid) and Modelled (dashed) Phase Residual

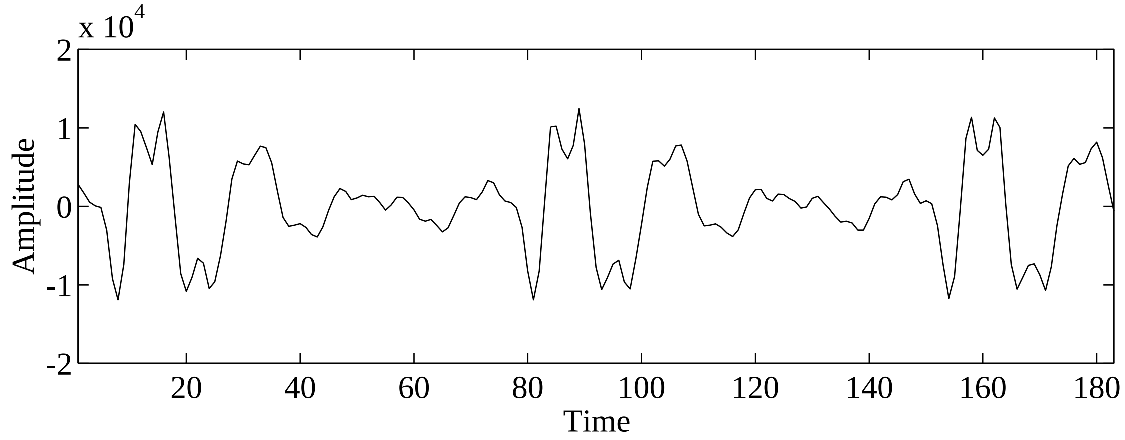


Figure 6.15: Original Prototype

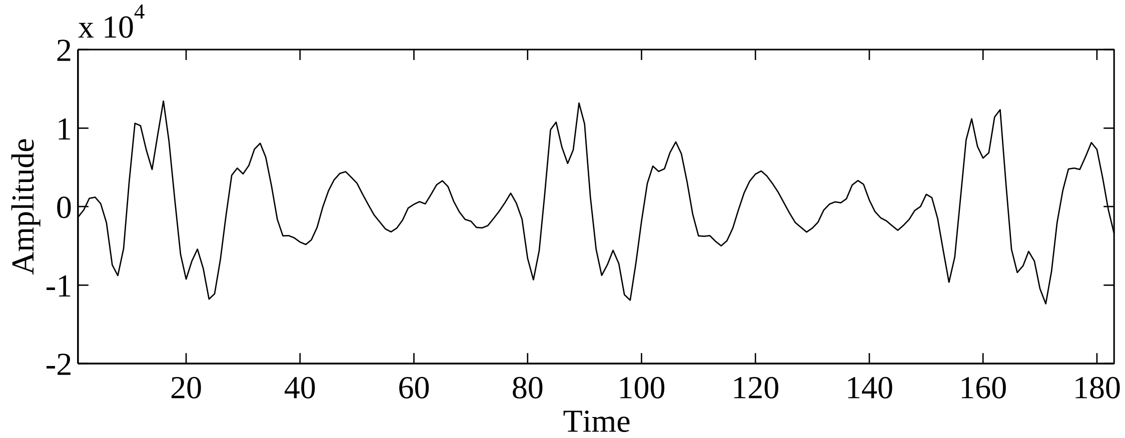


Figure 6.16: Modelled Prototype

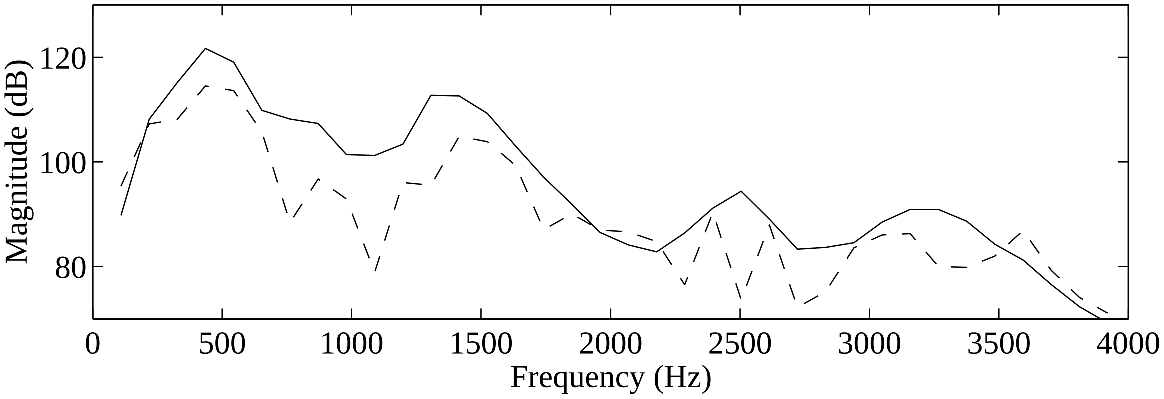


Figure 6.17: Original (solid) and Error (dashed) Magnitude Spectrums

6.5. Objective Results

To quantify the relative performance of the 3 phase modelling schemes, a Signal to Noise Ratio (SNR) measure was devised. This measure determines the signal to noise (error energy) ratio for each frame:

$$SNR = 10\log_{10}\left(\frac{\sum_{m=1}^{L} B_m^2}{2\sum_{m=1}^{L} B_m^2\left(1-\cos\left(\theta_m - \hat{\theta}_m\right)\right)}\right) \qquad (6.44)$$

which is measured in dB. It is important to note here that this measure is only used to evaluate the error contribution of modelling the phases, the magnitudes $\{B_m\}$ are assumed to be modelled perfectly. Average SNR values were determined for a test database of 2 male and 2 female speakers, a total of 2400 frames of clean (no additive noise) speech. The results are tabulated in Table 6.1.

These results support the examples given in the previous sections of this chapter for the 3 schemes. The analysis by synthesis and weighted polynomial schemes appear to perform quite well, whereas the truncated DCT scheme does not. The weighted polynomial scheme shows steady improvement as the polynomial order increases.

Model	Order	SNR (dB)
Analysis by Synthesis	n/a	6.53
Truncated DCT	$C = 5$	2.77
Weighted Polynomial	$K = 2$	5.86
Weighted Polynomial	$K = 3$	7.43
Weighted Polynomial	$K = 4$	8.54

Table 6.1: Average SNR Results for Phase Modelling Schemes

The average SNR values are fairly low as the results are averaged across the entire test database which includes many silence and unvoiced frames. In these cases the SNR is usually very low which drags the average SNR down. In strongly voiced frames, the SNR can be quite high, typically between 10 and 20 dB.

6.6. Unvoiced Speech

Initial listening tests during development indicated that the phase modelling schemes performed quite well for voiced speech but introduced harsh periodic sounds during unvoiced speech, which were perceptually very annoying. This is because for unvoiced speech the phases are generally random. All of the phase modelling schemes developed assume a measure of correlation between adjacent phases and force this correlation onto the modelled phases, thus introducing periodic artefacts into synthesised unvoiced speech.

To improve the quality of unvoiced speech, a simple two band voicing model was used. This model randomises the phases above a transition harmonic, and uses the phase model beneath the transition harmonic. The transition harmonic is determined using a rule based approach:

$$L_T = \begin{cases} L(SNR)/T\, SNR \le T \\ L\ SNR > T \end{cases} \tag{6.45}$$

Where T is the threshold at where unvoiced energy starts to be introduced. As the SNR decreases, more and more unvoiced energy is introduced, until the whole spectrum is unvoiced. The threshold T was chosen by experiment to be 5 dB.

As discussed in section 5.1 the addition of a voicing model is necessary only for the phase models considered in this chapter. The baseline unquantised generic coder does not implicitly require a voicing model.

6.7. Informal Listening Tests

With the voicing model incorporated, the quality of the three schemes could be evaluated using informal listening tests. Four test utterances were used, two male and two female. Of these, one male and one female were clean speech, the other male and female were corrupted by a moderate amount of periodic background noise to simulate a mobile acoustic environment. The 3 models were compared to the baseline coder and a VSELP simulation. High quality headphones were used for replay.

Informal tests using several listeners agreed that the high order weighted polynomial generally performed better, although the analysis by synthesis scheme was often very close. The truncated DCT scheme performed poorly, with very rough speech emerging from some voiced sections.

The general conclusion was that for clean speech the analysis by synthesis and weighted polynomial phase models were superior to VSELP, however for speech corrupted by background noise the quality was about equal to VSELP. All of the phase modelling schemes were inferior (in varying degrees) to the baseline coder, which for clean speech was almost transparent.

The main artefacts appear to be related to the two band voicing model. For example, some unvoiced sections still have audible periodic artefacts, and "rough" background noises can occasionally be heard in voiced sections. Both of these problems indicate occasional incorrect voicing decisions.

7. Quantised Sinusoidal Coder

This chapter describes the development of a quantised sinusoidal coder based on the techniques developed throughout this thesis. The main purpose of this coder is to tie together the techniques presented in this thesis. It should be stressed that the coder is not optimised and therefore not illustrative of the final quality possible from sinusoidal coding techniques.

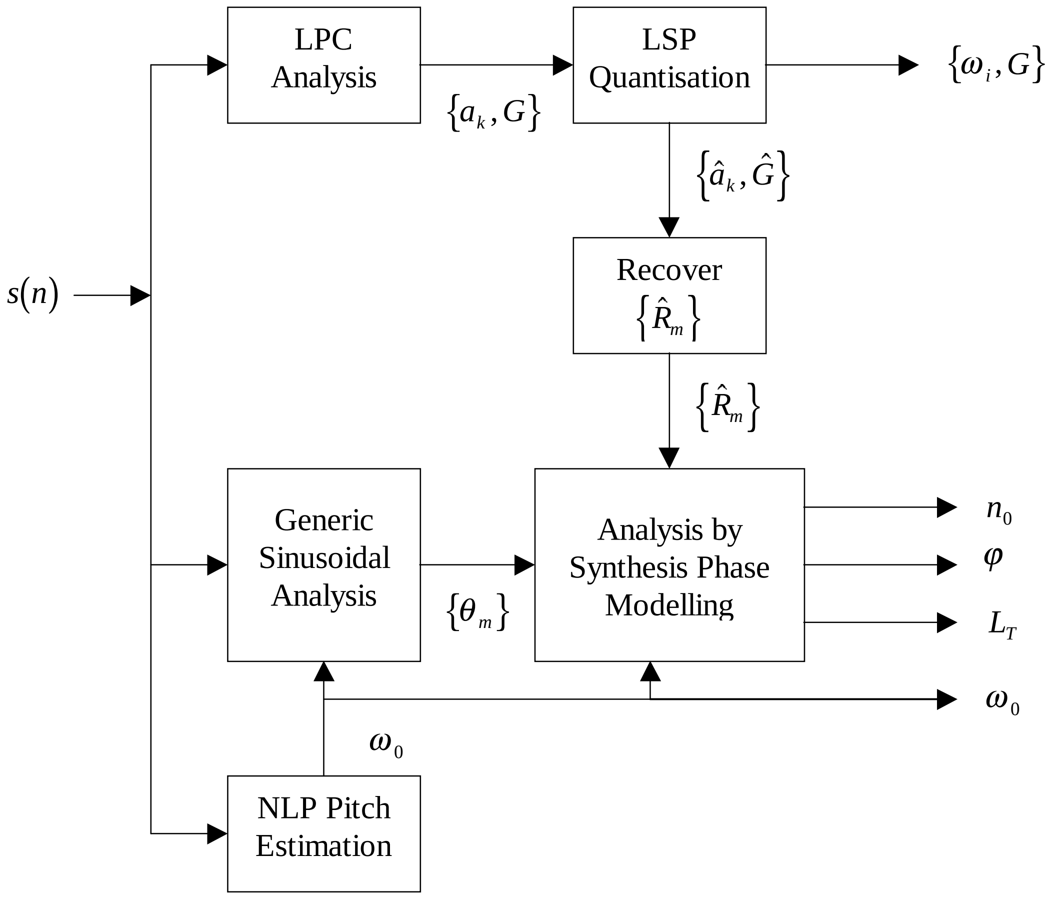


Figure 7.1: Block Diagram of Quantised Sinusoidal Encoder

Section 7.1 describes the operation of the quantised coder, and presents the techniques used to quantise the various parameters, section 7.2 describes the training procedure and results for the LSP quantisers. Section 7.3 evaluates the effects of the various stages of quantisation on the performance of the coder using objective measures and informal subjective testing. Speech files are available via the internet which demonstrate the various stages of quantisation (Appendix B).

7.1 Quantised Coder Operation

Figure 7.1 illustrates the quantised sinusoidal coder operation. Input speech $s(n)$ is used to determine the fundamental frequency for the current 10 ms frame ω_0 using the Non-Linear Pitch (NLP) algorithm described in Chapter 4. The speech is then modeled using the sinusoidal analysis algorithms developed in Chapter 5, however in this case only the harmonic phases $\{\theta_m\}$ are required, as the spectral magnitudes are determined using the LPC spectral modeling method also described in Chapter 5.

Parameter	Symbol	Bits/Frame	Bits/s
Line Spectrum Pairs	$\{\omega_i\}$	52	5200
LPC Gain in dB	G_{dB}	5	500
Fundamental Frequency	ω_0	8	800
Phase model impulse position	n_0	8	800
Phase model phase of complex gain	ψ	5	500
Phase model voicing transition frequency	L_T	5	500
Total		83	8300

Table 7.1: Bit allocation of Quantised Coder

Note that the LPC analysis produces both spectral shape information in the form of LPC coefficients $\{a_k\}$, and a scalar gain parameter expressed in dB G_{dB}. Quantisation of the LPCs is achieved by converting the LPC coefficients to Line Spectrum Pairs (LSPs), and scalar quantising each LSP. The LSP quantiser design is described later in this chapter. The LPC gain is uniformly quantised in the log domain. The recovered RMS spectral magnitudes $\{\hat{R}_m\}$ are then used with the phases from the sinusoidal analysis $\{\theta_m\}$ as the "target" for the phase modeling process.

The Analysis by Synthesis phase modeling scheme was chosen for the quantised coder due to the ease of quantisation compared to the Weighted Polynomial scheme. The parameters determined by the phase model n_0, ψ (the phase of the complex gain term), and the voicing transition parameter L_T are all scalar quantised. Note that the magnitude of the complex gain term does not need to be transmitted, as this information is conveyed by the LPC gain parameter discussed above.

The quantised parameters sent to the decoder and the bit allocations are summarised in Table 7.1.

The parameters G_{dB}, ω_0, ψ and L_T are linearly quantised using a similar method. We wish to represent a real number x with a b-bit integer code c over the range $x_{\min}$ to $x_{\max}$.

$$c = \left\lfloor \frac{x - x_{\min}}{x_{\max} - x_{\min}} + 0.5 \right\rfloor 2^b \tag{7.1}$$

The quantised value $\hat{x}$ can then be recovered from the code:

$$\hat{x} = \frac{c}{2^b}\left(x_{\max} - x_{\min}\right) + x_{\min} \tag{7.2}$$

Table 7.2 lists the parameters for each of the four linear quantisers used in the quantised sinusoidal coder.

Parameter	Symbol	b	$x_{\min}$	$x_{\max}$
LPC Gain in dB	G_{dB}	5	-20.0	60.0
Fundamental Frequency	ω_0	8	50 Hz	400 Hz
Phase model phase of complex gain	ψ	5	$-\pi$	π
Phase model voicing transition frequency	L_T	5	1	L

Table 7.2: Linear Quantiser Parameters

7.2 LSP Quantiser Training

The scalar quantisers were trained using a 22 minute database extracted from the TIMIT CD-ROM. Vectors of 12 LSPs were determined at 10 ms intervals, scalar quantisers were then developed for each quantiser using the Lloyd-Max [52] training algorithm. The bit allocation was adjusted experimentally to minimise the average Spectral Distortion (SD) over a 24 second test database of two male and two female speakers. The test database material was from outside the training database. The SD for a given frame is defined as [59]:

$$SD_{dB} = \sqrt{\frac{1}{N_{dft}} \sum_{k=0}^{N_{dft}-1} \left(20\log_{10}\left|A(k)\right| - 20\log_{10}\left|\hat{A}(k)\right| \right)^2} \qquad (7.3)$$

where $A(k)$ and $\hat{A}(k)$ are defined as the N_{dft} point DFTs of the original and quantised LPC coefficients. It is widely accepted [51] that for transparent quantisation of LPC parameters:

- The average SD must be less than 1 dB.
- Less than 2% of frames must not have an SD greater than 2dB.
- No frames must have an SD greater than 4dB.

Using the 24 second test database, quantisers of several bit allocations were tested. It was found that a quantiser with a 4,5,5,5,5,4,4,4,4,4,4,4 bit allocation for LSPs 1 to 12 gave results close to those required over the 24 second test database. The average SD results for several test databases are presented in Table 7.3. The first row is the 24 second database used for experimentally deriving the bit allocation.

Note that the LSP quantiser performs poorly for speech data outside of the training database, despite the relatively high bit rate of 52 bits/frame. This is perhaps due to different recording conditions or spectral characteristics in the test databases, and highlights the need for training material from several different sources/recording conditions.

Description	Frames	Average SD_{dB}	>2dB(%)	> 4dB(%)
24 second, 2 male, 2 female, Australian accent	2400	1.05	3.58	0.06
60 second, mixed male and female, English accent	6000	1.35	19.14	1.83
120 second, mixed male and female, BBC radio broadcast	12000	2.31	49.45	13.76
20 seconds from within training database (TIMIT CD-ROM)	2000	0.82	1.65	0.00

Table 7.3: LSP Quantiser Test Results

7.3 Quantised Coder Testing

This section describes the objective and subjective testing of the quantised sinusoidal coder (see also section 3.9). To evaluate the objective effect of the various quantisation stages, a SNR measure is proposed that is similar to the SNR measures used in sections 5.3 and 6.5. The SNR measure in section 5.3 was derived to test the spectral magnitudes and hence considered magnitudes only. The SNR measure in section 6.5 was derived to test the phase modeling and hence considered phase distortion only (weighted by the magnitude of each harmonic). The SNR measure below considers both magnitude and phase, and is compatible with both of the earlier measures:

$$SNR_{dB} = 10\log_{10}\left[\frac{\sum_{m=1}^{L}|A_m|^2}{\sum_{m=1}^{L}\left|\left(A_m - \hat{A}_m\right)\right|^2}\right] \tag{7.4}$$

where $\{A_m\}$ are the complex harmonic sinusoidal amplitudes for the current frame such that $A_m = B_m e^{j\theta_m}$. Note that this expression reduces to the earlier SNR expressions presented in sections 5.3 and 6.5 if the phases or magnitudes are considered to be unchanged by the modeling/quantisation under test.

	Condition					Average SNR (dB)
Test	LPC	LSP	Phase	Phase Q	$G+\omega_0$ Q	
A	X					15.73
B	X	X				12.03
C			X			6.12
D	X		X			5.50
E	X	X	X			4.07
F	X	X	X	X		3.76
G	X	X	X	X	X	2.40

Table 7.4: Sinusoidal Coder Quantisation Objective Test Matrix

Option	Description
LPC	12th order LPC modeling of spectral amplitudes
LSP	LSP quantisation of LPC model
Phase	Analysis by Synthesis phase modeling
Phase Q	Quantisation of A by S phase model parameters n_0, ψ, and L_T
$G+\omega_0$ Q	Quantisation of LPC Gain term and fundamental frequency

Table 7.5: Definition of Quantisation and Modeling Options

For a database of 2400 frames comprising two male and two female speakers a range of quantisation and modeling conditions were tested using the objective measure, and tabulated in matrix form in Table 7.4. The presence of an ‘X’ indicates which quantisation options are switched on for each test. The various quantisation options are explained in Table 7.5.

A steady decrease in average SNR is observed as the various quantisation options are enabled, with a large drop evident when the Analysis by Synthesis phase modeling is switched on. The reasons for this large drop are discussed in section 6.5.

Subjective testing was performed by informally evaluating the speech synthesised under the conditions listed in Table 7.4 over the 24 second (2400 frame) database of 2 male and two female speakers. High quality stereo headphones were used for the tests. Results indicate that tests A and B produced synthesised speech of relatively high quality, close to the baseline coder. The LSP quantiser (test B) produced no additional subjective distortion compared to LPC modeling alone (test A), indicating that for this test database at least, LSP quantiser performance is acceptable as predicted by the SD tests in the previous section. In this case the SNR measurements that suggest a significant drop in quality between test A and test B do not correlate with the subjective results.

Phase modeling alone produced synthesised speech roughly equivalent in subjective quality to VSELP. However, when LPC modeling and LSP quantisation was applied to the phase modeled coder (test E), the speech quality dropped significantly. The drop in subjective quality is much larger than when the LPC and LSP options were applied to the coder before phase modeling. This is perhaps due to some dependence of the phase model on the integrity of the LPC parameters, which are used to obtain the minimum phase component of the phase model. Another possibility is that the quantised LPC parameters upset the rather delicate balance of the voicing model constants, which were experimentally derived with LPC modeling/quantisation switched off. Section 8.2 discusses possible ways to improve the quality of the coder. The final two quantisation steps (tests F and G) produced no further decrease in subjective quality.

8. Conclusions and Further Work

In the preceding chapters, several original techniques suitable for sinusoidal speech coding were presented. Section 8.1 summarises the work presented in this thesis, and highlights and compares the original contributions to the current state of the art. Finally, several areas for further work are presented in section 8.2.

8.1 Significance of New Work

Time and frequency domain speech coding techniques were introduced in Chapters 2 and 3. These chapters presented background information, including discussions of several time and frequency domain coding techniques, and the mathematical framework necessary for presenting the original contributions in later chapters. Several minor contributions were presented in these chapters, including:

- In Section 2.5.3 an analytical discussion of the CELP analysis by synthesis codebook search, including a mathematical treatment of filter memory effects. Although the practical use of this information is widely known, no equivalent treatment has been found in the literature.

- In Section 3.4 a derivation of the MBE (3.19) and harmonic sinusoidal (3.16) analysis equations. Results similar to those obtained from this analysis have been previously published, but equivalent derivations have not been found in the literature, although alternative derivations have been presented for the sinusoidal analysis equations. This work derives the MBE and sinusoidal equations using the same framework, thus illustrating the similarity of the two models.

- In section 3.7 qualitative arguments demonstrated that due to assumptions of short term stationarity over 10-30 ms intervals, problems can occur in the parameter estimation techniques used for harmonic sinusoidal and MBE coders. For example a changing fundamental ω_0 across the frame can "smear" the high order harmonics across the frequency axis, causing spectral magnitude and voicing estimation errors for MBE coders. Transition frames (those containing onsets or voiced/unvoiced changes) can also cause partial or complete breakdown of the parameter estimation algorithms. These observations lead to the selection of a short frame update rate for the generic sinusoidal coder presented in Chapter 5. No equivilent treatment of this subject is known to the author at this time.

Chapter 4 presented the NLP pitch estimation algorithm. This algorithm has two original features:

- The use of a square law non-linearity as a basic pitch extractor.
- Post processing of the resulting pitch candidates using a second pitch estimation algorithm, in this case the MBE pitch estimation algorithm.

The algorithm was carefully designed and tested using three different methods, (objective, contour and subjective) over a range of speech sources with good results. Another original contribution was the analysis of failure modes and suggested improvements including a simple tracking algorithm.

While other pitch estimators use the same basic pitch extractor/post processing structure of the NLP algorithm, it has several unique features. A survey of existing pitch estimation algorithms suggests that no other algorithms in the literature use a non-linearity combined with a second pitch estimator for post processing. The non-linearity is used specifically to enhance the fundamental through the superposition of difference tones produced by harmonic distortion. Although this effect is widely known [4] it’s use is uncommon in current, state of the art pitch estimation algorithms.

The testing of the NLP algorithm involved the use of a high quality unquantised sinusoidal coder algorithm. Often, pitch estimation algorithms are associated with communications quality [30] or synthetic quality (eg LPC) vocoders. In these coders, pitch detector errors are often masked by other artifacts in the coding process, thus subjective tests of pitch detector performance based on speech quality may have a lower “resolution” than if the same pitch detector was tested with a high quality coder. The high performance of the unquantised sinusoidal coder presented in this thesis demonstrates the reliability and robustness of the NLP pitch estimation algorithm.

Chapter 5 presented the high quality unquantised sinusoidal coder. This coder is related to existing sinusoidal and MBE coders, however the techniques used in the analysis, synthesis, and spectral modelling stages differ from those commonly used in the literature as follows:

- The use of a 10 ms frame rate to provide good transient response to non-stationary speech segments. This frame rate has been used in some of the earlier, non harmonic sinusoidal coders [17], however all recent harmonic sinusoidal and MBE coders employ frame lengths of 20 ms or longer.

- The use of an RMS spectral magnitude measure $\{R_m\}$ that is relatively insensitive to errors in fundamental estimation and the voiced or unvoiced nature of the energy in the current band compared to the traditional MBE and sinusoidal estimation techniques.

- The use of harmonic phases to convey voicing information, which removes the need for a voicing estimator in the unquantised generic sinusoidal coder. This feature has been used in earlier, non-harmonic sinusoidal coders [17], but not in harmonic sinusoidal coders.

- An RMS average method of spectral magnitude recovery from LPC models that provided an efficient means of modelling the spectral magnitudes with a moderate order LPC model while maintaining high speech quality. This method is considerably lower in algorithmic complexity compared to other schemes (summarised in section 5.3) found in the literature.

Informal listening tests through high quality headphones indicated that the speech quality obtained from the generic coder was very high, almost transparent in many cases. The LPC modelling introduced a small amount of distortion, however the overall quality remained high.

The phase modeling techniques presented in Chapter 6 apply a voicing measure to the coder but this is not fundamental to the operation of the unquantised coder, unlike other algorithms such as MBE [28]. For example, other phase modeling/quantisation algorithms could be employed (eg vector quantisation [60]) that do not require voicing measures or decisions.

Chapter 6 presented three original phase models, of which two were shown to be capable of adequately representing phase by the ability to reproduce good quality speech. Phase is usually ignored in sinusoidal and MBE coders, often discarded at the analysis stage and then reconstructed at the decoder using heuristic techniques [28]. Very few authors have attempted to introduce parametric phase models, although several have presented combined phase/magnitude parameteric models [41][42][43][67].

The basic assumption was that the phase spectrum could be modelled as the cascade of a minimum phase filter and an all pass filter, excited by an impulse at time n_0. In all cases a LPC filter provides the minimum phase component, while the various phase models attempt to model the phase residual, or all pass component. The three phase models where:

- The *Analysis by Synthesis* model, where an impulse with a constant *complex* gain excites an LPC synthesis filter at time n_0. In this case the pulse position provides a linear phase term, and the angle of the complex gain term provides a constant phase term. Thus the all pass filter can be considered a 2nd order polynomial (straight line) approximation to the phase residual spectrum. The parameters of the phase model are determined using an analysis by synthesis loop, similar to CELP or multipulse algorithms. This method produced good quality synthesised speech.

- The second phase model used *Truncated DCT* Coefficients to represent a smoothed version of the all pass filter phase spectrum. The motivation for this was that for voiced speech, the phase spectrum is smooth after removal of the dominant linear component caused by the excitation instant, n_0, thus the DCT was used to decorrelate the phase residual spectrum and transmit only the low order DCT coefficients. However, this produced significant distortions in the synthesised speech. This was traced to the inability of the method to take into account the magnitude of the various spectral components when fitting the phase model, unlike the analysis by synthesis phase model.

- This provided the motivation for the *Weighted Polynomial* phase model, which used a K^{th} order polynomial fit to the phase residual spectrum. By using an approximation, it was shown that a tractable least squares solution could be found to fit the polynomial while taking into account the relative importance of each phase based on the magnitude of the spectrum at each harmonic frequency. This method also provided good quality speech.

All of the phase models assumed a certain correlation in the phases of adjacent harmonics, which is true for voiced speech. However, to deal with unvoiced speech, a simple voicing model was introduced, so that the three phase models could be evaluated for real speech signals. Overall, the weighted polynomial scheme performed best, however in many cases the analysis by synthesis scheme came very close.

Finally, chapter 7 combined the techniques presented in this thesis to produce a fully quantised 8.3 kbit/s coder. This coder used a 12th order LPC model quantised with LSPs, and the analysis by synthesis phase model. The effects of the various quantisation and modelling stages were analysed using objective and informal subjective methods. Although the amplitude and phase modelling/quantisation stages work well when separated, some degradation in speech quality was observed when they were combined. The reasons for this degradation are at this stage unclear, however they provide plenty of scope for further work (see below).

A discrepancy between subjective and objective results was noted, the objective results suggesting poorer performance than perceived using informal subjective tests. As discussed in section 3.9 of this thesis such differences in objective and subjective measure performance are common in speech coding research due to the inability of simple objective distance measures to exactly model human perception of speech signals.

8.2 Further Work

One of the main features of the NLP algorithm is the two stage basic extraction/post processing concept, where a basic pitch extractor provides a set of candidates which are tested by sampling another pitch estimator. It would be useful to implement this technique using pitch estimator combinations other than NLP/MBE and compare results.

In vocoder algorithms such as sinusoidal/MBE pitch estimation is usually performed “open loop”, for example the pitch is estimated, then passed to the next analysis stage as an input. This differs from CELP-type analysis by synthesis coders that typically extract the pitch using a closed loop search. The closed loop search effectively determines the effect of a set of pitch candidates on the next analysis stage, then chooses the pitch accordingly. It is suggested that a vocoder pitch estimation algorithm could also be implemented in an analysis by synthesis fashion, perhaps by evaluating a set of basic pitch extractor candidates using the vocoder analysis and modelling stages, and comparing the resultant speech to the original.

A significant problem with the phase models presented in this thesis in the inability to deal with unvoiced harmonics, all of the models assume some degree of correlation in adjacent phase residuals which force a voiced structure onto the modelled phases. This was overcome for the purposes of listening tests by introducing a crude two band voicing model, however a phase scheme that can deal with voiced and unvoiced speech without the need for an external voicing estimator would be a more robust approach.

The LSPs consume a large portion of the quantised coder bit rate, this could be reduced by exploiting time domain correlation in adjacent 10 ms frames, and using vector quantisation of LSPs. Given many references cite less than 25 bits/frame for 10th order LSP quantisers [51] it should be possible to achieve less than 30 bits/frame for a 12th order quantiser. The poor performance of the quantiser outside of the training database highlights the need for careful quantiser design using a training database obtained from more than one recording condition.

The objective and subjective testing of the quantised coder suggests the analysis by synthesis phase model is sensitive to quantisation and modelling of the LPC coefficients. This may be due to distortions introduced into the quantised LPCs that affect the ability of the phase model to match the original phases, or problems with the voicing estimator and should be investigated further.

9. References

[1] Douglas O'Shaughnessey, "Speech Communication - Human and Machine," *Addison-Wesley Publishing Company*, 1987.

[2] L.R. Rabiner, R.W. Schafer, "Digital Processing of Speech Signals," *Prentice-Hall*, 1978.

[3] J.D. Markel and A.H. Grey, Jr., "Linear Prediction of Speech," 1976.

[4] Wolfgang Hess, "Pitch Determination of Speech Signals," *Springer-Verlag*, 1983.

[5] Alan V. Oppenheim, Ronald W. Schafer, "Digital Signal Processing," *Prentice-Hall*, Chapter 7 - "Discrete Hilbert Transforms", 1975.

[6] John Makhoul, "Linear Prediction: A Tutorial Review," *Proc. of the IEEE*, Vol. 63, No. 4, April 1975.

[7] Jose M. Tribolet and Ronald E. Crochiere, "Frequency Domain Coding of Speech ," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-27, No. 5, pp. 512-530, October 1979.

[8] John Makhoul, "Spectral Linear Prediction: Properties and Applications," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23, No. 3, pp. 283-296, June 1975.

[9] N.S. Jayant, J.D. Johnson, Y. Shoham, "Coding of Wideband Speech," *Eurospeech 91*, Genoa, Italy, pp. 373-379, 1991.

[10] P. Kroon and E.F. Deprettere, "A Class of Analysis-by-Synthesis Predictve Coders for High Quality Speech Coding at Rates Between 4.8 and 16 kbit/s," *IEEE Journal on Selected Areas in Communications*, 6, pp. 334-363, February 1988.

[11] M.R. Schroeder and B.S. Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates," *Proc. ICASSP-85*, pp. 937-940.

[12] INMARSAT Council Meeting 1990, "INMARSAT-M Voice Coding Algorithm," *published as SDM\NMOD1\APPENDIX\ISSUE 3.0*, August 1991.

[13] European Telcommunications Standards Institute Technical Commitee, "Recommendation 06.10: GSM Full-Rate Speech Transcoding," Version 3.2.0, Jan. 1990.

[14] EIA/TIA, "IS-54 Dual Mode Subscriber Equipment - Network Equipment Compatibility Specification," 1989.

[15] Andrew DeJaco, William Gardner, Paul Jacobs, Chong Lee, "QCELP: The North American CDMA Digital Cellular Variable Rate Speech Coding Standard," *Proc. IEEE Workshop on Speech Coding for Telecommunications*, pp. 5-6, October 1993.

[16] R.J. McAulay and T.F. Quatieri, "Magnitude-Only Reconstruction using a Sinusoidal Speech Model," *ICASSP-84*, pp. 27.6.1-27.6.4, 1984.

[17] Robert J. McAulay and Thomas F. Quatieri, "Mid-Rate Coding Based on a Sinusoidal Representation of Speech," *ICASSP-85*, pp 945-948, 1985.

[18] Robert J. McAulay, Thomas F. Quatieri, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," *IEEE Trans. on ASSP*, Vol. ASSP-34, No. 4, pp. 744-754, August 1986.

[19] R.J. McAulay and T.F. Quatieri, "Phase Modelling and it's Application to Sinusoidal Transform Coding," *ICASSP-86*, pp. 1713-1715, 1986.

[20] Thomas F. Quatieri and Robert J. McAulay, "Phase Coherence in Speech Reconstruction for Enhancement and Coding Applications," *ICASSP-89*, pp. 207-210, 1989.

[21] Robert J. McAulay and Thomas F. Quatieri, "Pitch Estimation and Voicing Detection Based on a Sinusoidal Model," *ICASSP-90*, pp. 249-252, 1990.

[22] R. J. McAulay and T.F. Quatieri, "Sine-Wave Phase Coding at Low Data Rates," *Proc. ICASSP-91*, pp. 577-580, 1991.

[23] R.J. McAulay and T.F. Quatieri, "The Application of Subband Coding to Improve Quality and Robustness of the Sinusoidal Transform Coder," *ICASSP-93*, Vol. 2, pp. 439-442, 1993.

[24] R. J. McAulay, T. Champion, T. F. Quatieri, "Sinewave Amplitude Coding Using Line Spectral Frequencies," *IEEE Workshop on Speech Coding for Telecommunications*, pp. 53-54, 1993.

[25] R. J. McAulay, T. Champion and T. F. Quatieri, "Sinewave Amplitude Coding Using Line Spectrum Frequencies," *Proceedings of IEEE Workshop on Speech Coding for Telecommunications*, pp. 53-54, October 1993.

[26] B. Atal, J. Remde, "A New Model of LPC excitation for producing natural sounding speech at low bit rates," *Proc. ICASSP*, pp. 614-617, 1982.

[27] Luis B. Almeida, Jose M. Tribolet, "Nonstationary Spectral Modeling of Voiced Speech," *IEEE Trans. on ASSP*, ASSP-31, pp. 664-678, June 1983.

[28] D.W. Griffin and J.S. Lim, "Multiband Excitation Vocoder," *IEEE Trans. on ASSP*, Vol. ASSP-36, No. 8, pp. 1223-1235, August 1988.

[29] M.S. Brandstein, "A 1.5 Kbps Multi-Band Excitation Speech Coder," *Massachusetts Institute of Technology Masters Thesis*, May 1990.

[30] John C. Hardwick and Jae S. Lim, "A 4.8 kbps Multi-Band Excitation Speech Coder," *ICASSP-88*, pp. 374-377, 1988.

[31] Osamo Fujimura, "An Approximation to Voice Aperiodicity," *IEEE Transactions on Audio and Electroacoustics*, AU-16, No. 1, pp. 68-72, March 1968.

[32] J. Makhoul, V. Viswanathan, R. Schwartz & A. Huggins, "A Mixed-Source Model for Speech Compression and Synthesis," *ICASSP-78*, pp. 163-166, 1978.

[33] Soon Young Kwon and Aaron J. Goldberg, "An Enhanced LPC Vocoder with No Voiced/Unvoiced Switch," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32, No. 4, pp. 851-858, August 1984.

[34] Alan V. McDree and Thomas P. Barnwell III, "A Mixed Excitation LPC Vocoder with Frequency-Dependant Voicing Strength ," *Speech and Audio Coding for Wireless and Network Applications*, pp. 259-254, 1993.

[35] James D. Wise, James R. Caprio, Thomas W. Parks, "Maximum Likelihood Pitch Estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24, No. 5, pp. 418-423, October 1976.

[36] Lawrence R. Rabiner, "On the Use of Autocorrelation Analysis for Pitch Detection," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-25, No. 1, pp. 24-33, February 1977.

[37] Lawrence R,. Rabiner, Michael J. Cheng, Aaron E. Rosenberg, Carol A. McGonegal, "A Comparitive Performance Study of Several Pitch Detection Algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24, No. 5, pp. 399-418, October 1976.

[38] Carol A. McGonegal, Lawrence R. Rabiner, Aaron E. Rosenberg, "A Semiautomatic Pitch Detector (SAPD)," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23, pp. 570-574, December 1975.

[39] J. J. Dubnowski, R. W. Schafer and L. R. Rabiner, "Real-Time Digital Hardware Pitch Detector," *IEEE Trans. on Acoust., Speech and Signal Processing*, 24, pp. 2-9, Feb. 1976.

[40] W. B. Kleijn, "Encoding Speech Using Prototype Waveforms," *IEEE Trans. Speech and Audio Processing*, 1, pp. 386-399, 1993.

[41] William R. Gardener, Bhaskar D. Rao, "Mixed-Phase AR Models for Voiced Speech and Perceptual Cost Functions," *ICASSP 94*, 1, pp. 205-208, 1994.

[42] David L. Thomson, "Parametric Models of the Magnitude/Phase Spectrum for Harmonic Speech Coding," *ICASSP 1988*, pp. 378-381, 1988.

[43] I. M. Trancoso, L. B. Almeida, and J. M. Tribolet, "Pole-Zero Multipulse Speech Representation Using Harmonic Modelling in the Frequency Domain," *Proceedings of ICASSP-85*, pp. 260-263, 1985.

[44] Haiyun Yang, Soo-Ngee Koh, Pratab Sivaprakasapillai, "Pitch Synchronous Multi-Band (PSMB) Speech Coding," *Proceedings of ICASSP-95*, 1, pp. 516-519, 1995.

[45] Gilbert Strang, "Linear Algebra and its Applications," *Academic Press*, 1980.

[46] D. Rowe, W. Cowley, and A. Perkis, "A Multi-band Excitation Linear Predictive Speech Coder," *Proceedings of Eurospeech 91*, pp. 239-242, 24-26 September 1991.

[47] G. Nagaratnam, D. Rowe, "Spectral Magnitude Modelling for Sinusoidal Coding," *IEEE Workshop on Speech Coding for Telecommunications*, pp. 81-82, 1995.

[48] B. G. Evans, A. M. Kondoz, S. Yelder, M. R. Suddle, W. Ma, "A High Quality 2.4 kb/s Multi-Band LPC Vocoder and its Real Time Implementation," *Proceedings ISSSE*, 1992.

[49] Clifford I. Parris, Danny Wong and Francois Chambon, "A Robust 2.4kb/s LP-MBE With Iterative LP Modelling," *Proceedings of Eurospeech 95*, pp. 677-680, September 1995.

[50] Amro El-Jaroudi, John Makhoul, "Discrete All-Pole Modelling for Voiced Speech," *Procedings ICASSP-87*, pp. 320-323, 1987.

[51] K.K. Paliwal and B.S. Atal, "Vector Quantisation of LPC parameters in the Presence of Channel Errors," *Speech and Audio Coding for Wireless and Network Applications*, pp. 191-201, .

[52] N.S. Jayant and P. Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video," *Prentice-Hall*, 1984.

[53] S. Dimolitsas, "Subjective Assessment Methods for the Measurement of Digital Speech Coder Quality," *Speech and Audio Coding for Wireless and Network Applications*, pp. 43-53.

[54] S. Dimolitsas, "Objective Speech Distortion Measures and their Relevance to Speech Quality Assessment," *Proc IEEE, Vol. 136, No. 5*, pp. 317-324, October 1989.

[55] S. Wang, A. Sekey, A. Gersho, "An Objective Measure for Predicting Subjective Quality of Speech Coders," *IEEE Journal on Selected Areas in Communications*, Vol. 10, No. 5, June 1992.

[56] N. Kitawaki, H. Nagabuchi, "Quaity Assessment of Speech Coding and Speech Synthesis Systems," *IEEE Communications Magazine*, October 1988, pp. 36-44.

[57] F. Itakura, "Line Spectrum Representation of Linear Predictive Coefficients of Speech Signals," *J. Acoust. Soc. Amer.*, vol. 57, S 35(A), 1975.

[58] F. K. Soong, B-H. Juang, "Line Spectrum Pair (LSP) and Speech Data Compression," *Proc. ICASSP'84*, pp. 1.10.1-1.10.4, 1984.

[59] R. Q. Schuyler, P.T. Barnwell, M.A. Clements, "Objective Measures for Speech Quality," *Prentice Hall New Jersey*, 1988.

[60] Y. Jiang, V. Cuperman, "Encoding Prototype Waveforms Using a Phase Codebook," *IEEE Workshop on Speech Coding for Telecommunications*, pp. 21-22, 1995.

[61] A.S. Spanias, "Speech Coding: A Tutorial Review," Proceedings of the *IEEE*, Vol. 82, No. 10, pp 1541-1582, October 1994.

[62] D. Rowe, A. Perkis, W.G. Cowley, J.A. Asenstorfer, "Error Masking in a Real Time Voice Codec for Mobile Satellite Communications," Proceedings of Speech Science and Technology, pp 498-503, Melbourne, Australia, November 1990.

[63] A. Perkis, D. Rowe, "Improvements in the IMBE coder Utilizing Burst Error Masking Techniques," *IEEE Workshop on Speech Coding for Telecommunications,* pp. 78-79, 1991.

[64] Juin-Hwey Chen, "Toll-Quality 16 kb/s CELP Speech Coding with Very Low Complexity," *Procedings of ICASSP-95,* pp. 9-12, 1995.

[65] A. Perkis, B. Ribbum, "Application of Stochastic Coding Schemes in Staellite Communication," *Advances in Speech Coding*, Kluwer Academic Publishers, pp. 277-286, 1991.

[66] T. Wigren et. al., "Improvements of Background Sound Coding in Linear Predictive Speech Coders," *Procedings of ICASSP-95,* pp. 25-28, 1995.

[67] S. Ahmadi et. al., "A New Sinusoidal Phase Modeling Algorithm”, *Procedings of ICASSP-97, Vol 3,* pp. 1675, 1997.

Appendix A

A1.1 Solving Overdetermined Systems using Orthogonal Projection

In many parameter estimation problems we wish to minimise some cost function with respect to a set of parameters. After manipulation, this often reduces to a set of m equations and n unknowns, such that $m > n$. In these cases no exact solution is possible, however a "best fit" solution in terms of minimum squared error may still be obtained. This appendix presents a generalised derivation of least squares minimisation from a linear algebra point of view [45]. Although similar results may be obtained using calculus, linear algebra provides a more elegant extension to overdetermined systems containing complex numbers.

Consider an overdetermined system of m equations and n unknowns, such that $m > n$:

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \vdots \qquad\qquad\qquad\qquad\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \end{array} \tag{A1.1}$$

Expressing this system in matrix form:

$$\mathbf{Ax} = \mathbf{b} \tag{A1.2}$$

where:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{A1.3}$$

$$\mathbf{x} = \left[x_1 \; x_2 \; \ldots \; x_n\right]^T \tag{A1.4}$$

$$\mathbf{b} = \left[b_1 \; b_2 \; \cdots \; b_m\right]^T \tag{A1.5}$$

The above equations describe a system where there are more equations than unknowns. In most cases no exact solution is possible, instead we must determine the solution vector $\mathbf{Ax}$ to be as "close" as possible in a mean square error sense to the target vector $\mathbf{b}$.

The optimum solution will lie in the subspace spanned by the columns of $\mathbf{A}$. This subspace can be viewed as all the possible linear combinations of the columns of $\mathbf{A}$. The optimum solution, $\mathbf{x}$, will be the point on the column space of $\mathbf{A}$ that is closest to $\mathbf{b}$. This point occurs where the error vector $\mathbf{e} = \mathbf{Ax} - \mathbf{b}$ is orthogonal to the column space of $\mathbf{A}$, and is known as the projection of $\mathbf{b}$ onto the subspace. This situation is illustrated in Figure A1.1.

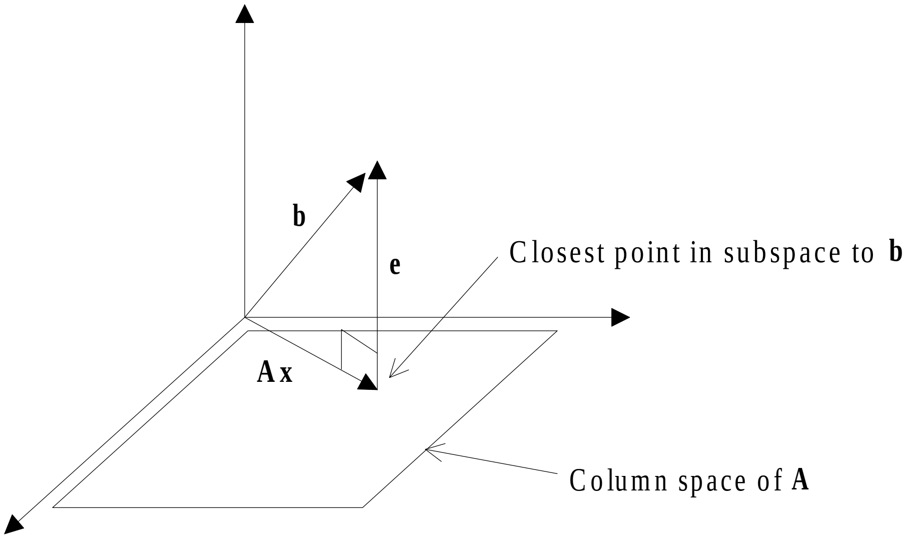


Figure A1.1: Orthogonal Projection of $\mathbf{b}$ onto Column Space of $\mathbf{A}$

Thus for any overdetermined system $\mathbf{Ax} = \mathbf{b}$ the best solution $\mathbf{b}$, in a least squares sense is when $\mathbf{e}$ is orthogonal to $\mathbf{Ax}$:

$$\mathbf{e}^T \mathbf{Ax} = 0 \tag{A1.6}$$

$$(\mathbf{b} - \mathbf{Ax})^T \mathbf{Ax} = 0 \tag{A1.7}$$

which gives a solution vector:

$$\mathbf{x} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{b} \tag{A1.8}$$

which in effect chooses $\mathbf{x}$ to minimise $\|\mathbf{e}\|^2$.

A1.2 Extension of Orthogonal Projection to Complex Numbers

In real linear algebra we define the length of a vector:

$$\|x\|^2 = x^T x \tag{A1.9}$$

There is a close geometric relationship between the length of a vector and it's inner product. To maintain this relationship for complex numbers, the first vector in the inner product is usually conjugated [45]:

$$\|x\|^2 = x^H x \tag{A1.10}$$

where x is now complex and:

$$x^H = \overline{x}^T \tag{A1.11}$$

The same rules can be applied to the inner product of two different complex vectors:

$$\langle a, b \rangle = a^H b \tag{A1.12}$$

This definition of the length of complex vectors enables least squares problems involving complex numbers to be solved using orthogonal projection in a more meaningful way than techniques involving calculus.

Consider the least squares fit applied to lines in K dimensional complex space, defined by the cost function:

$$E = \sum_{k=1}^{K} \left| S(k) - GW(k) \right|^2 \tag{A1.13}$$

Expressed in vector form:

$$E = \|\mathbf{s} - G\mathbf{w}\|^2 \tag{A1.14}$$

where $\mathbf{s}$ and $\mathbf{w}$ are complex K element column vectors:

$$\mathbf{s}^T = \left[S(1)\; S(2) \ldots S(K) \right] \tag{A1.15}$$

$$\mathbf{w}^T = \left[W(1)\,W(2)\ldots W(L)\right] \tag{A1.16}$$

and G is a complex constant. The problem is illustrated graphically for the two dimensional real case in Figure A1.2

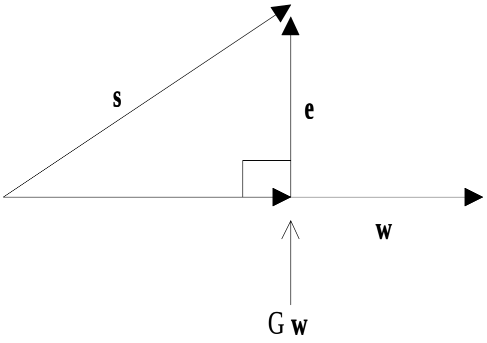


Figure A1.2: Orthogonal Projection of $\mathbf{s}$ onto $\mathbf{w}$

Note that this case is similar to that presented in section A1.1, except that in this case $\|e\|^2 = \|\mathbf{s} - G\mathbf{w}\|^2$ is minimised by adjusting a single complex coefficient G. In this case the column space of the system $G\mathbf{w}$ is the single column or K dimensional line $\mathbf{w}$.

The optimum solution occurs when $\mathbf{e}^T$ is orthogonal to $G\mathbf{w}$:

$$\mathbf{e}^T G\mathbf{w} = 0 \tag{A1.17}$$

$$G = \frac{\mathbf{s}^H \mathbf{w}}{\mathbf{w}^H \mathbf{w}} \tag{A1.18}$$

Appendix B

Demonstration Speech Files

Speech files that demonstrate the various algorithms presented in this thesis are available via the internet. The page:

http://www.itr.unisa.edu.au/scrc/speech/index.html

contains instructions on how to obtain and listen to the speech files. The files available are summarised in the file *demo.txt*, which is listed below.

```
DEMO FILES FOR THESIS
=====================
David Rowe, 17/3/97

The files listed below demonstrate the techniques developed for the
thesis.  They are organised into a suggested listening order.  The source
file hts.wav contains 8 sentences, (4 speakers, 2 sentences each).

It is recommended that the files be broken up into smaller files for the
purpose of comparative listening.  For example, when comparing two files,
just listen to one 3-second interval, such as the first 3 seconds of hts.wav
and htsuq.wav, rather than the entire 24 second file.


Chapter 5 - Generic Sinusoidal Coder
---------

hts.wav         original
htsuq.wav       baseline unquantised generic coder as developed in Ch5.
htsa.wav        spectral magnitude modeling using RMS average method
htsa1.wav       spectral magnitude modeling using sampling method.

Chapter 6 - Parametric Phase Models
---------

hts.wav         original
htsuq.wav       baseline unquantised generic coder as developed in Ch5.
htsp1.wav       A by S phase model
htsp2.wav       Truncated (5 coefficients) DCT phase model
htsp3.wav       K=4 Weighted Polynomial phase model
htsv.wav        IS54 VSELP provided for comparison

mmt1.wav        Original truck noise corrupted speech
mmt1.wav        Baseline coder
mmt1p1.wav      A by S phase model - note background noise cf mmt1.wav
mmt1p3.wav      Polynomial phase model - note background noise cf mmt1.wav
mmt1v.wav       IS54 VSELP for comparison

Chapter 7 - Quantisation
---------

hts.wav         original
htsuq.wav       baseline unquantised generic coder as developed in Ch5.
htsx.wav        where x = a,b,...,g see table 7.3.  Coder in various stages
                of quantisation (Ch7)
htsv.wav        IS54 VSELP provided for comparison
```

Appendix C

Informal Subjective Testing Methodology

Throughout the research described in this thesis, informal subjective tests (defined and discussed in section 3.9) have been used to evaluate the various algorithms. The methodology used was to carefully listen to the decoded speech from the algorithm under test using files from a relatively small test database. Although small, the database includes a range of different speakers, listed in Table C.1.

File	Description	Length (seconds)
hts.spc	2 Australian male, 2 Australian female, 2 sentences each	24
spkdat1.spc	25 British male and female speakers, 1 sentence each	52
r1.spc	BBC1 radio news, 3 speakers, swift speaking	120
Total		196

Table C.1: Informal Subjective Testing Database

Evaluations of the algorithm under test were made on a comparative basis, for example the original (uncoded) speech was compared to the unquantised sinusoidal coder speech in Chapter 5. Comparisons were made on short (approximately 3 second) segments at a time, for example the first sentence of hts.spc would be compared using the algorithm under test and the reference condition for that test. The subjective evaluations were made by the author using a "sound blaster" PC based replay system with both headphones and hi-fi speakers.

Usually, results were evident by listening to just a few utterances, in most cases the results were similar across the entire test database. When the subjective quality of the two utterances was not clearly different, a result of no difference was noted.