PITCH SYNCHRONOUS WAVEFORM INTERPOLATION FOR VERY LOW BIT RATE SPEECH CODING

Thesis submitted in accordance with the requirements

of the University of Liverpool for the degree of

Doctor of Philosophy

by

Hung Bun Choi

Department of Electrical Engineering and Electronics

The University of Liverpool.

January 1997

Acknowledgements

I would like to express my sincere gratitude to my supervisors Dr. B.M.G. Cheetham and Dr. C. C. Goodyear for their continuous support, advice, encouragement and most importantly their patience throughout the course of this work.

Thanks are particularly expressed to my industrial supervisor Dr. W. T. K. Wong for his expert advice, invaluable discussions, encouragement and precious friendship. I also wish to thank Mr. P. A. Barrett for his invaluable discussions.

Thanks are also expressed to Dr. K. G. Evans for his constant help and many expert consultations.

I would also like to thank my fellow research students and research colleagues Dr. M. K. Y. Lo, Dr. J. K. W. Tang, Dr. X. Q. Sun, Dr. Fabrice Plante, Mr. S. D. Watson, Mr. M. P. Pollard and Mr. D. J. Jones for their invaluable discussions, encouragement and help. I am grateful to my friends in the Department for a great time and an experience that I will always remember.

I would like to thank British Telecom for providing facilities and financial support which have made this research possible, to the Department of Electrical Engineering and Electronics for making facilities available and the British Government Overseas Research Student Award for the financial assistant.

I express my most sincere appreciation to my parents and sisters for their love, support and sacrifice for the opportunity I never deserved. I share my achievement with them, as without their support it would not have been possible.

"Brothers, I do not consider myself yet to have taken hold of it. But one thing I do: Forgetting what is behind and straining toward what is ahead, I press on toward the goal to win the prize for which God has called me heavenward in Christ Jesus."

Philippians 3:13, 14

To my

beloved parents

Contents

Glossary of acronyms

ADPCM	Adaptive Differential Pulse Code Modulation
AMDF	Average Magnitude Difference Function
ANSI	American National Standards Institute
APC	Adaptive Predictive Coding
APCM	Adaptive Pulse Coded Modulation
AT&T	American Telephone and Telegraph
CELP	Code-Excited Linear Prediction
CSA-CELP	Conjugate-Structure Algebraic Code-Excited Linear Prediction
DFT	Discrete Fourier Transform
DM	Delta Modulation
DoD	Department of Defence
DPCM	Differential Pulse Code Modulation
ETSI	European Telecommunications Standards Institute
FFT	Fast Fourier Transform
GPSWI	Generalised Pitch-Synchronous Waveform Interpolation
GSM	Groupe Special Mobile
IMBE	Improved Multiband Excitation Vocoding
IMS	Interframe Multi-stage Split
Inmarsat	International Maritime Satellite Organisation
IS	Inverse Sine Transform
ITU	International Telecommunication Union
JDC	Japanese Digital Cellular Network
LAR	Log Area Ratio
LBG	Linde-Buzo-Gray codebook training algorithm
LBG-CS	Linde-Buzo-Gray Clustering Split
LD-CELP	Low-Delay Code-Excited Linear Prediction
LP	Linear Prediction
LPC	Linear Prediction Coding
LSF	Line Spectral Frequency
MBE	Multiband Excitation Vocoding
ME-LPC	Mixed-Excitation Linear Prediction Coding
MP-LPC	Multi-Pulse Linear Prediction Coding
MQ	Matrix Quantisation
MS	Multi-stage Split
PARCOR	Partial Correlation Coefficients
PCM	Pulse Code Modulation
PSTN	Public Switched Telephone Network
PSWI	Pitch-Synchronous Waveform Interpolation
PWI	Prototype Waveform Interpolation
RP-LPC	Regular-Pulse Linear Prediction Coding
RPE-LTP	Regular-Pulse Excited Long-Term Prediction
SAPD	Semi-Automatic Pitch Detector
SC	Speech Classification
SCA	Spectral Comb Analysis
SES	Slowly Evolving Spectrum

SEW	Slowly Evolving Waveform
SHS	Sub-Harmonic Summation
SNR	Signal-to-Noise Ratio
SQ	Scalar Quantisation
STC	Sinusoidal Transform Coding
RES	Rapidly Evolving Spectrum
REW	Rapidly Evolving Waveform
TFI	Time Frequency Interpolation
TIA	Telecommunications Industry Association
TPD	Two-way Pitch Detector
TPSWI	Two-mode Pitch-Synchronous Waveform Interpolation
VL	Voicing Confidence Level
VLSI	Very Large Scale Integration
VPCM	Vector-quantised Pulse Code Modulation
VQ	Vector Quantisation
VSELP	Vector Sum Excited Linear Prediction
WI	Waveform Interpolation

Abstract

Considerable effort is currently being devoted to the development of new digital signal processing techniques to meet the demands of very low bit-rate speech coding for a wide range of applications. Such applications include digital mobile radio, secure speech transmission, storing and archiving speech, telephone based speech services and real time speech transmission over the internet. The ascendancy of prototype waveform interpolation (PWI) and waveform interpolation (WI) techniques in this field has generated much interest in their use for speech coding at bit-rates around 2.4kb/s. The purpose of this thesis is to assess the performance of current PWI and WI coding techniques and to contribute to their further development.

The concept of PWI coding for voiced speech is to periodically extract from the linear prediction residual segments of length equal to the current pitch-period. Efficient descriptions of these segments are encoded along with the estimated pitch-period and the parameters of an all-pole synthesis filter. At the decoder, reconstituted residual segments at adjacent update-points are interpolated to approximate missing portions of the residual. The resulting signal is then used to excite an all-pole synthesis filter which imposes an approximation to the original spectral envelope and thus produces an output signal which is close to the original voiced speech. In the original work on PWI, unvoiced speech was coded by switching to a form of CELP. In this thesis, a PWI/CELP coder has been studied, implemented as a C-program, and evaluated.

One of the major sources of distortion found in PWI coded speech arises from error in the estimated pitch-period of voiced speech segments. This leads to the extraction of segments of inappropriate length and consequently a reconstructed residual with grossly incorrect periodicity and spectral content. A novel and robust pitch detector has therefore been developed and tested both for clean natural speech and also for speech signals corrupted by various types of background noise.

Various linear prediction (LP) analysis techniques are studied in this thesis to find ways of providing an LP residual signal with the flattest possible frequency spectrum and best possible all-pole synthesis filter. This makes the residual closer to a pseudo-periodic impulse train for voiced speech and thus increases the effectiveness of quantisation schemes for the residual. It was found that Burg's algorithm is generally more accurate than the more commonly used autocorrelation method, especially when pitch-synchronous analysis is used.

An LP analysis and synthesis filter pair which uses line spectral frequencies (LSF) directly as filter coefficients has been developed and compared with conventional ladder filter structures. Using the new filter structure, a smooth evolution of the spectral envelope of the reconstructed speech signal can be preserved by interpolating on a sample-by-sample basis the LSF's as encoded at adjacent update-points. Computational costs associated with the conversion between LSF's and LP ladder filter coefficients are eliminated by this approach.

An all-pass filtering scheme is investigated for deriving the phase spectrum of the residual from magnitude only information. This scheme is based on assumptions about the glottal excitation to the human vocal tract, and by eliminating the transmission of phase information can be used to further reduce the bit-rate required for encoding voiced residuals.

The thesis presents a 2.4kb/s two-mode pitch-synchronous waveform interpolation (TPSWI) coder which uses PWI to encode voiced speech and switches to a pseudo-random sequence based model for unvoiced speech. Transitions between the two coding modes are modelled by an overlap-and-add technique. Informal listening tests suggest that the decoded speech obtained from the TPSWI coder is better than that obtained from the original PWI/CELP coder. In particular, the TPSWI decoded speech was found to have voiced/unvoiced transitions that are smoother and more natural.

More recent development in WI coding aim to generalise the voiced speech model to include unvoiced speech and transitions. This eliminates the need for a two-mode technique. A 2.4kb/s generalised pitch-synchronous waveform interpolation (GPSWI) coder is devised in this thesis. Informal listening tests suggest that the decoded speech obtained from the 2.4kb/s GPSWI coder is substantially better than that obtained from the 2.4kb/s LPC-10e and the 4.1kb/s IMBE coders, and is comparable to that obtained from the 2.4kb/s ME-LPC and the 2.4kb/s WI coder from AT&T. ME-LPC and the AT&T coder were candidates in the recent American DoD 2.4kb/s speech coding standardisation competition and the former was the winner. Fully quantised versions of TPSWI and GPSWI coder have been produced in simulation.

Chapter 1

Introduction

1.1 Digital processing of speech signal

Since the invention of telephony in the mid-nineteenth century, speech has been converted to an electrical signal and conveyed over long distance analogue communication channels such as twisted pairs, coaxial cables and radio. The disadvantage of analogue telephony is the irreversible degradation in speech quality that accumulates over long distance communication links. Repeaters installed at regular intervals amplify not only the speech waveform but also the distortion that has become embedded in it.

The innovation of pulse code modulation (PCM) allowed a digital representation of speech to be transmitted. Repeaters were then able to reconstruct an almost error-free digital signal since distortion in the pulse-like analogue waveform conveying the digital information could be removed by generating new pulses. The use of digital transmission with computer controlled networks leads to flexible and efficient systems. Individual units of a global digital communication system may now be designed independently. These individual units include modules for source coding, signal modulation, error protection and error correction.

A digital communication network also allows the existing land-based and radio spectrum to be utilised more efficiently and intensively. This is important for future expansion in telecommunications, in areas such as long distance telephony, digital cellular communication networks, mobile satellite communications and aeronautical services. Such expansion must be achieved with the restrictions imposed by limitations of transmission capacity in existing world-wide networks and the available radio spectrum. With phenomenal advances in VLSI technology over the past decade, the computational power of digital processors is large and still rapidly increasing while the price of the technology is falling. Hence digitised speech

can be processed efficiently in ways that allow it to be transmitted more efficiently and at less cost over digital communication links.

Speech coders and decoders are designed to convert analogue speech into digital form and vice-versa. The term encoder may also be usefully applied to devices or algorithms which convert speech digitised at one bit-rate to an alternative digital representation which may be at a considerably lower bit-rate. Many speech coding algorithms have been established and some of them have been standardised for implementation in international exchanges as well as at regional levels.

One reason for the growing interest in speech coding algorithms comes from the rapidly expanding area of mobile communications. The channel capacity available for the current generation of mobile telephony will be saturated very soon and one of the possible ways of increasing the channel capacities is to reduce the bit-rate requirement of the existing speech coders, with minimum effect on perceptual quality. Research in speech coding algorithms will also have a role in the establishing of a flexible platform for the future development of a global communication network, with integration of voice, video and data network, to form the so-called information highway.

Many international and regional bodies have been established in the past decades to standardise the design and use of speech coders. At the international level, a number of speech coders have been standardised by the International Telecommunication Union (ITU). These include the universally used 64kb/s PCM coder, G712 [7], the 32kb/s ADPCM coder, G726 [8], and the 16kb/s Low-Delay CELP (LD-CELP) coder, G728 [9]. Recently an 8kb/s "Conjugate-Structure Algebraic" CELP coder (CSA-CELP), G729 [10], has reached the final phase of standardisation and the requirement of a 4kb/s ITU standard is in preparation.

At the regional level, the Pan European Mobile Communication system adopted a 13kb/s regular pulse excited long-term prediction coder (RPE-LTP) in the ETSI/GSM system [11]. Together with channel coding, the complete system requires an overall transmission rate of 22.8kb/s. In order to use the available frequency

bandwidth efficiently, the GSM committee is considering the introduction of a half-rate system.

In the United States, an 8kb/s vector-sum excited linear predictive coder (VSELP) is used in the North American digital cellular system [12]. The ANSI/TIA is currently examining a half-rate coder. In Japan, the Japanese Digital Cellular (JDC) network has adopted a 6.7kb/s VSELP coder, and a half-rate coder for this network has been chosen recently. In the area of satellite communications, Inmarsat has adopted a 4.1kb/s "improved multiband excitation" coder (IMBE) for the Inmarsat-M system. The complete system requires a total bit-rate of 9.6kb/s [13]. The planned Iridium satellite cellular network is still examining the speech coders for their system.

Speech coders utilised in military applications have a less demanding perceptual quality specification than domestic ones. The American department of defence (DoD) has implemented a 2.4kb/s LPC-10e coder [14] in their military communication network. This coder preserves the intelligibility of speech but sacrifices the overall speech naturalness and speaker recognisability. Recently the DoD has completed the assessment of a new candidate for their network. The new coder is known as a mixed-excitation linear predictive coder (ME-LPC) and is able to deliver natural speech quality at 2.4kb/s [67].

Packet switching networks, which use techniques for detecting and rectifying data transmission errors, have been increasingly used to replace conventional circuit switching networks. By packetising information and transmitting packets through computer networks, source coders with different source bit-rates are allowed to run side-by-side on the same network. The variety of source coders used for voice traffic includes those for long distance telephony, digital cellular telephony, satellite communication systems and public switched telephone networks (PSTN). Packet switching networks are also capable of handling non-voice traffic including telex, image data and control signals for PSTN. Speech coding circuitry implemented in packet switching networks must be able to handle both voice as well as non-voice traffic. In addition, they must be able to work in tandem with other speech coders.

These considerations reveal the importance of speech coding in relationship to our daily lives. They show the importance of continuing research and development in speech coding algorithms. A great deal of research is going on all over the world in order to meet new challenges which include further reduction in the system bit-rate, minimisation of the system delay and system complexity and the search for efficient low cost and low power consuming real time implementation of the algorithms.

1.2 Evolution of speech coding algorithms

Speech coders are generally classified into three categories, waveform coders, vocoders and hybrid coders.

Waveform coders

Waveform coders aim to represent the exact shapes of speech waveforms as precisely as possible. The perceptual quality of the synthesised speech obtained from waveform coders is preserved by using a sufficiently high bit-rate to achieve the required accuracy. The earliest algorithm implemented in a waveform coder is known as pulse code modulation (PCM). This assigns code-words to a number of quantisation levels. The use of uniformly spaced quantisation levels would be appropriate for an input signal which may be assumed to have a uniform probability density function (p.d.f.) and where the effect of quantisation is considered independent of sample value. However this is not the case for speech since the p.d.f. of a speech signal is not uniform and higher noise levels can be tolerated for higher sample values. The probability of higher amplitude sample values is generally smaller than that of lower amplitude samples. It has also been suggested [22] that the dynamic range of telephone quality speech from a single talker generally varies by about 20dB to 40dB, and there is a further 20dB to 40dB variation among different talkers in different environments. Hence with uniform quantisation, many quantisation levels must be included to preserve the perceptual quality of speech over this very wide dynamic range. Consequently, a very high bit-rate would be required with uniform quantisation. A solution to this problem is to use non-uniform

quantisation, such as is provided by A-law or μ-law PCM. Such versions of PCM aim to make the signal-to-quantisation-noise-ratio for signals above a certain minimum level independent of signal level. This is achieved by having quantisation steps logarithmically spaced (or approximately so) above a certain minimum level. This is a static quantisation scheme in that the quantisation levels are fixed in relation to specified maxima and minima. Alternatively, a dynamic quantisation scheme may be used where the quantisation levels effectively vary with the characteristics of the signal, e.g. adaptive delta modulation and adaptive PCM (APCM).

The performance of PCM can also be enhanced by reducing the variance and the dynamic range of the input signal. A differential PCM (DPCM) coder quantises the difference between each speech sample and a prediction to it based on previous quantised samples. This exploits the fact that correlation normally exists between adjacent speech samples. The prediction is calculated as the sum of suitably scaled previous quantised speech samples. The scaling factors ideally depend on the characteristic of the speech and should be updated frequently. However, fixed values are used by simpler DPCM coders normally with first order predictors. At the decoder, an identical copy of the predictor used at the encoder is installed to reconstruct the speech sample. To further improve the system performance, the quantisation levels can be made adaptive to the behaviour of the input signal. This is the concept used in adaptive differential PCM (ADPCM) speech coders and the well known standard G726.

Conventional waveform coders are able to produce high quality speech around 64kb/s to 32kb/s. However the speech quality deteriorates rapidly below these bit-rates. The G726 version of ADPCM achieves at 32kbit/s a quality which is equivalent to and in some way superior to 64kb/s A-law PCM.

Vocoders

A vocoder attempts to represent speech by sets of parameters that allow a signal which sounds like the original speech to be regenerated but whose waveform shape does not necessarily resemble the original wave shapes. This is realised by

using a simplified speech production model, in which the speech is assumed to be produced by passing an excitation signal through a vocal system filter. The excitation signal is assumed to be a pseudo-periodic impulse train for voiced speech and white Gaussian noise for unvoiced speech. The vocal system filter models the composite effect of the glottal excitation, vocal tract and lip-radiation. Vocoders are typically designed for bit-rates between 1.2 to 2.4kb/s. An example of a vocoder is the 2.4kb/s LPC-10e coder [14]. Although vocoders are capable of delivering highly intelligible speech at very low bit-rates, the reproduced speech generally has a synthetic, unnatural and sometimes metallic quality.

Hybrid coders

Hybrid coders combine the best qualities of waveform coders and vocoders at the expense of increased complexity. An example of a hybrid coding technique is adaptive predictive coding (APC) [85] which is capable of representing good speech quality at around 16kb/s.

In APC, a predictor computes an estimate of each speech sample from previous quantised samples and the difference between the true value and the predicted value is quantised for each sample. The predictor coefficients are computed by linear prediction analysis. Various speech coding algorithms have been developed from the basic idea of APC. Speech quality comparable with that obtained from 16kb/s APC has been obtained from modified APC algorithms operating at bit-rates around 8kb/s. These algorithms include regular pulse linear predictive coding (RP-LPC), multi-pulse linear predictive coding (MP-LPC) and vector sum excited LPC (VSELP). Another well-known technique based on APC is known as coded excited linear predictive coding (CELP) [86], which is able to achieve good speech quality at a bit-rate as low as about 4.8kb/s. Recent research results have indicated that the performance of CELP coders drops dramatically as the required bit-rate is reduced below 4.8kb/s.

It is widely believed [15] that interpolation is one of the keys to achieving low bit-rate speech coding at bit-rates in the range 4kb/s to 2.4kb/s or even lower. Many interpolation algorithms have been intensively studied over the past decade.

These include sinusoidal transform coding (STC) as proposed by R. J. McAulay and T. F. Quatieri [62], linear predictive coding (LPC) as used by the American DoD [14], prototype waveform interpolation (PWI) as proposed by W. B. Kleijn [70],[74]-[76], time-frequency interpolation (TFI) by Y. Shoham [65],[66] and multiband excitation vocoding (MBE) by D. W. Griffin and J. S. Lim [63].

Another aspect of low bit-rate speech coding is the exploitation of known characteristics of human perception of sound. Current understanding is generally based on models of the human auditory system. By studying the sensitivity of the human ear to various forms of distortion, better sounding can be encoded at low bit-rates by allocating fewer bits/s to perceptually less important information. Experiments have shown that the frequency scale can be divided into a number of critical bands for which the sensitivity of the ear to frequency may be considered approximately the same. A masking threshold can be computed for each band which is dependent on the energy levels in adjacent bands. Any frequency component which has a spectral amplitude less than this threshold will not be heard. Hence these frequency components need not be allocated any precious bits for transmission. Also if sufficient coding bits are allocated so that the quantised noise spectrum is maintained below this masking threshold, quantising noise will not be audible.

The use of vector quantisation (VQ) [51] allows a block of samples or parameters to be quantised collectively as a group instead of quantising them individually as with scalar quantisation. Many aspects of vector quantisation are being studied, such as split VQ, multi-stage VQ and various code-book searching algorithms [53]. The main difficulty of VQ for real-time implementation arises from the computational complexity involved in code-book searching. The computation required for training an efficient code-book can be very great, and this is also a major difficulty. Recently matrix quantisation (MQ) has also been proposed as a variation of vector quantisation. It is believed [3] that MQ may be the key effective technique for speech coding at bit-rate below 1kb/s.

This section has briefly explored the evolution of speech coding algorithms in general terms. A number of these algorithms have been standardised by the ITU.

Among these, the 64kb/s PCM and the 32kb/s ADPCM coders have been classified as waveform coders and the 16kb/s LD-CELP and the 8kb/s CSA-CELP coders are considered as hybrid coders. The performance of these coders would deteriorate rapidly if they were made to operate at lower bits-rate. Great effort is required to reduce the bit-rate requirement of existing algorithms without affecting speech quality. New coding algorithms are also required to cope with future developments in telecommunications. With the rapid advances in the microelectronics technology, digital processors are becoming more and more powerful. This allows more complex speech coding algorithms to be implemented in real time.

1.3 Project objectives

The objective of this project has been to develop a high quality low bit-rate speech coding algorithm using speech interpolation methods. The new coding algorithm aimed to achieve a 2.4kb/s coder which has a speech quality comparable to that of 32kb/s ADPCM coders. The coding algorithm developed was to be based on PWI coding.

PWI coding was originally designed to handle voiced speech by exploiting the fact that it is a pseudo-periodic signal [68]. Much information content of an individual pitch-cycle is repeated. The repeated information can be discarded at the encoding stage and regenerated at the decoder using interpolation. In the earliest PWI coding techniques, a CELP coder was used for unvoiced speech. A PWI/CELP coder is capable of producing good quality speech at bit-rates between 3 and 4kb/s [70][71]. The source of speech degradation in a PWI/CELP coder comes from two areas, a) multiple and sub-multiple pitch errors in the PWI coding and b) the switching between the two coding modes. Recently a general waveform interpolation algorithm (WI) was proposed [73]-[76]. The new coding algorithm further exploits the characteristics of speech signals and removes the need for a CELP coder for unvoiced speech. The WI coder has been successfully implemented and good speech quality has been obtained at 2.4kb/s [76]. A disadvantage of WI coding is its computational complexity.

Two 2.4kb/s speech coders have been designed in this project. The first coder has been named as the two-mode pitch-synchronous waveform interpolation (TPSWI) coder [83]. It is composed of two individual units: a PWI based coder for voiced speech and a noise generator for unvoiced speech. An overlap and add technique is employed to ensure a smooth transition between the two coding modes. The TPSWI coder was modified by removing the need to switch to a different model for unvoiced speech. This leads to a uniform coding algorithm, applicable to all speech types, which is referred to as the generalised pitch-synchronous waveform interpolation (GPSWI) coder [84]. To accomplish the design of both these coders, studies have been made in various aspects of coding including robust pitch-period detection, effective LP analysis methods, LP analysis and synthesis filter structures, quantisation of spectral information, interpolation based speech coding algorithms and the quantisation of speech coder parameters.

1.4 Thesis organisation

This thesis is composed of eight chapters and three appendices. The advantages of digitally representing speech in a communication network and the evolution of various speech coding algorithms have been discussed in this chapter. In addition, a number of important speech coder standards have also been mentioned.

In chapter 2, some of the available pitch-period determination algorithms are described and the design of a two-way pitch detector (TPD) is then presented. The TPD consists of four elements: a pitch pre-processing unit, a speech classifier, a pitch estimator and a pitch post-processing unit. The design objectives of the individual units will be discussed. Finally the TPD is tested both for clean speech and for speech contaminated by various types of additive noise. Experimental results from these tests will be presented.

Chapter 3 is concerned with the linear prediction analysis of speech. Two LP analysis techniques: the autocorrelation method and Burg's method are discussed.

Line spectral frequencies (LSF) are described and investigated. These are generally used as alternatives to conventional LP ladder filter coefficients for representing the frequency response of the all-pole synthesis filter. An LSF analysis and synthesis filter pair is proposed as an alternative to conventional ladder filter structures. The performance of the LSF filters is compared with that of the corresponding ladder filters. The results of LP analysis using the autocorrelation method and Burg's method are also compared. Experiments carried out include evaluations of the performance of the two methods for both pitch-asynchronous and pitch-synchronous analysis. The effect on the accuracy of the LP analysis of varying the length of the analysis window and of positioning the analysis window at different location within a speech frame are investigated. An adaptive window length is proposed for use with the Burg pitch-synchronous LP analysis method.

In chapter 4, the representation of short-term spectral information using LSF coefficients will be illustrated. The chapter begins with an introduction to LSF's as an alternative representation of conventional ladder filter coefficients. The reasons for preferring LSF's to other available candidates is presented. Issues concerning the design of a vector quantiser (VQ) for LSF's are also discussed. These issues include the choice of quantiser structure, the complexity of the available procedures, the assessment criteria, the code-book training and the choice of distortion measure. Finally the design stages of a 24-bit LSF vector quantiser is presented, together with experimental results evaluating the performance of this vector quantiser.

In chapter 5, the principle of PWI coding is introduced. The design of a two-mode pitch-synchronous waveform interpolation (TPSWI) coder is presented. Finally, a phase derivation scheme which aims to model the phase spectrum of speech residuals is explored. The phase derivation scheme is derived from a voiced speech production model which considers the nature of the glottal excitation.

Chapter 6 starts with a description of the principles of WI coding algorithms. The design stages of the generalised pitch-synchronous waveform interpolation (GPSWI) coder are presented. Finally the performance of the TPSWI coder and the GPSWI coder are compared.

In chapter 7, quantisation of the 2.4kb/s TPSWI coder and the GPSWI coder is achieved. Chapter 8 concludes the findings of this project and make suggestions for further improvements to the quality of the speech coders. The thesis has three appendixes. Appendix A illustrates the structure of a CELP coder. In appendix B, the conversion between LSF and LP filter coefficients is discussed. Finally, in appendix C, the mathematical proof of frequency-domain manipulations on the TPSWI coder and the GPSWI coder is presented.

Chapter 2

Pitch Determination

2.1 Introduction

A pitch detector classifies the nature of speech segments into different categories such as silence (s), voiced (v), unvoiced (uv) and mixed excitation (m) [24], and for voiced segments gives an estimation of the true pitch-period. Hence a pitch detector is able to perform a) speech classification and b) pitch estimation. It is believed [25] that the perceptual quality of speech produced by a coder is strongly dependent on the accuracy of the pitch detector used. This accuracy is particularly important for speech coders which implement interpolation techniques. For example, three major types of distortion, i.e. noisy speech, reverberation and tonal artefacts, have been observed [70] in PWI encoded speech. Experiments have shown that these three types of distortion are closely related to pitch errors [70].

A number of pitch determination algorithms have been proposed in past decades [5], and a comprehensive study of some of the algorithms has been published [25]. Generally, a pitch detector can be classified as time-domain or frequency-domain, depending on the domain from which the features of the input speech signal are extracted. In time-domain pitch detectors, features are taken directly from the speech waveform. The features commonly used are signal energy, zero-crossing rate, autocorrelation function as well as the cross-correlation function [23][29]. Autocorrelation function algorithms have remained the most popular pitch determination algorithms in the time-domain family. In an autocorrelation-based pitch detector, speech classification can be carried out simply using an autocorrelation threshold and the pitch-period is generally estimated by locating the global maximum position in the autocorrelation function [30]. A simple way of computing a type of autocorrelation function is used in the Average Magnitude Difference Function (AMDF) technique [31], and has been successfully implemented in the LPC10e speech coder [14]. Different from the autocorrelation

method, the AMDF algorithm finds the location of the nulls in an AMDF function and the pitch-period is obtained by measuring the differences between adjacent nulls.

The complex cepstrum is recognised as being the most popular technique for pitch determination in the frequency-domain family [33]. This pitch detector computes the FFT of the logarithm of the power spectrum of a section of speech signal, which effectively separates the effects of the vocal tract and the vocal source in a so called quefrency spectrum. Hence, pitch estimation is realised by locating the quefrency of the global maximum in the quefrency spectrum and speech classification can be immediately performed by assessing the magnitude of the peak. The disadvantage of a cepstrum pitch detector is that a number of pitch-cycles must be included in the analysis frame, in order to produce a well-defined ripple structure in the logarithmic power spectrum. This vastly increases the computational complexity of the pitch detector. A cepstrum pitch detector is not able to estimate the frequency of a pure sine tone.

It is claimed that a more reliable pitch determination can be obtained by using a sub-harmonic summation (SHS) approach [34]. In the SHS method, the linear frequency axis of the frequency spectrum is transformed to a logarithmic frequency axis. Through a series of summation and shift operations on the modified frequency spectrum, a predominant peak is obtained at the location of the pitch-frequency. Speech classification in the SHS method can be done by evaluating the correlation function between two pitch-cycles using the estimated pitch-frequency.

Spectral comb analysis (SCA) [35] is also a candidate belonging to the frequency-domain family. The SCA approach reduces the computational complexity of the SHS method, by computing the correlation between the power spectrum of a section of speech signal and a comb-shaped spectrum. The teeth of the comb are separated by the pitch-frequency being analysed. The amplitude of the teeth are made decreasing with increasing frequency in order to avoid the possible multiple-pitch errors. Voicing decision can be made by comparing the energy detected by the comb spectrum and the total energy.

With the impact of developments in the field of artificial neural networks, the implementation of neural models in speech processing has been actively researched in recent years [36]. Barnard *et al* [37] proposed a pitch detector which implements a neural-net speech classifier. The speech classifier was claimed to achieve a 96.5% accuracy in the tests carried out [37]. The use of artificial neural networks provides another possible option in dealing with pitch determination. It is notable that Schroeder [5, page 521] has stated that "We do not have a single pitch determination algorithm which operates reliably and accurately for all applications, all signals, all speakers, all recording conditions and all possible quality degradation ". Many pitch determination algorithms are surveyed in this reference [5].

In previous work at the University of Liverpool [16][17], pitch detectors have been developed for various types of speech coder. Lo [16] developed a cross-correlation based pitch determination algorithm for a single pulse excitation coder. The accuracy of the pitch detector was found more reliable than those implemented using an autocorrelation function approach. Tang [17], adapted this algorithm into a 4-way pitch detector, which is used for a type of PWI coder. The performance of the pitch detector was found to be comparable to the one used in the 4.1kb/s IMBE coder [64].

In this chapter a pitch detector which implements a type of cross-correlation function, based on Lo [16], will be introduced. The basic idea of the cross-correlation method is described in section 2.2. In sections 2.3 and 2.4, the enhancement techniques for pitch determination will be discussed. A full description of a two-way pitch detector (TPD), i.e. a pitch detector which classified the input speech signal into either voiced or unvoiced and a pitch-period is given in case of voiced speech, will be presented in section 2.5. The TPD has been tested for clean speech and various examples of noisy speech and the results of the tests will be presented in section 2.6.

2.2 Pitch determination using a cross-correlation function

Correlation is a measure of similarity between signals. In a cross-correlation pitch determination method, two adjacent and non-overlapping segments of speech with identical time duration, m samples, may be examined. Suppose we denote two such segments with samples denoted by x(n) and y(n), for n=0, 1, 2, ... , m-1. If the speech is voiced and m is the pitch-period, x(n) should be, approximately, a scaled version of y(n). Therefore we can define an error signal e(n) such that,

$$e(n) = x(n) - k\,y(n) \qquad n = 0, 1, \ldots, m-1 \tag{2.1}$$

which should be small for each n when the scaling factor k is correctly chosen. The sum of squared errors for a given value of m and k may be defined by,

$$\begin{aligned} E_m(k) &= \sum_{n=0}^{m-1} e^2(n) \\ &= \sum_{n=0}^{m-1} \big(x(n) - k\,y(n)\big)^2 \end{aligned} \tag{2.2}$$

By setting to zero the derivative of equation 2.2 with respect to k, the value of k can be computed which minimises $E_m(k)$ for a given value of m. Therefore,

$$\frac{\partial E_m(k)}{\partial k} = -2\sum_{n=0}^{m-1} x(n)\,y(n) + 2k\sum_{n=0}^{m-1} y^2(n) = 0 \tag{2.3}$$

and hence the required value of k is,

$$k = \frac{\sum_{n=0}^{m-1} x(n)\,y(n)}{\sum_{n=0}^{m-1} y^2(n)} \tag{2.4}$$

By substituting equation 2.4 into 2.2, the minimum obtainable total square error for a given value of m becomes,

$$\begin{aligned} E_m(k) &= \sum_{n=0}^{m-1} x^2(n) - 2k\sum_{n=0}^{m-1} x(n)\,y(n) + k^2\sum_{n=0}^{m-1} y^2(n) \\ &= \left(1 - \frac{\left(\sum_{n=0}^{m-1} x(n)\,y(n)\right)^2}{\sum_{n=0}^{m-1} x^2(n)\sum_{n=0}^{m-1} y^2(n)}\right)\sum_{n=0}^{m-1} x^2(n) \end{aligned} \tag{2.5}$$

As a result, $E_m(k)$ will be at its minimum for a given value of m when C(m) is maximised, where,

$$C(m) = \frac{\sum_{n=0}^{m-1} x(n)\, y(n)}{\sqrt{\sum_{n=0}^{m-1} x^2(n) \sum_{n=0}^{m-1} y^2(n)}} \tag{2.6}$$

C(m) is known as the normalised cross-correlation function of the two signal segments $\{x(n)\}_{0,m-1}$ and $\{y(n)\}_{0,m-1}$. A high C(m) means that the difference between the two signals is small and hence the two signals are highly correlated.

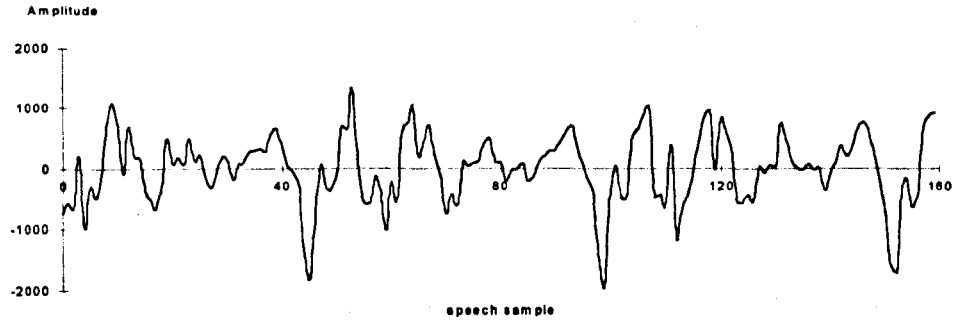
2.2.1 Implementation of a cross-correlation function in pitch detection

In a speech coder, the input speech signal s(n) is normally segmented into fixed duration frames which are typically of length between 10ms and 30ms. For each of these frames, normalised cross-correlation measurements between s(n) and s(n+m) are computed for a range of values of m.

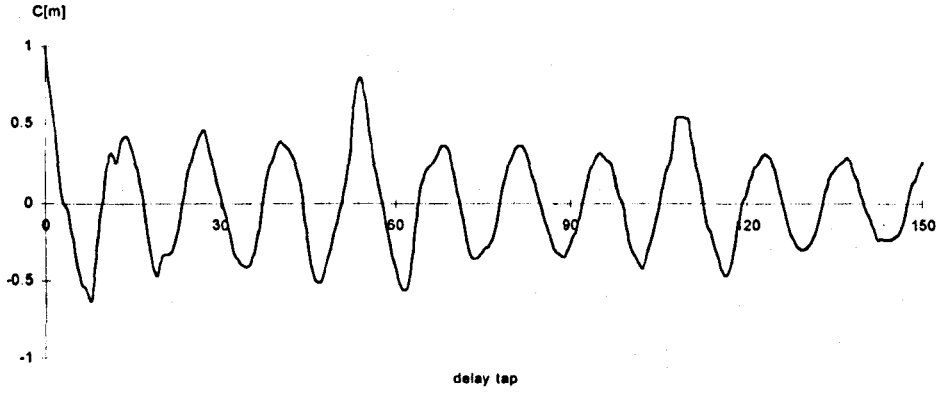
$$C(m) = \frac{\sum_{n=0}^{m-1} s(n)\, s(n+m)}{\sqrt{\sum_{n=0}^{m-1} s^2(n) \sum_{n=0}^{m-1} s^2(n+m)}} \tag{2.7}$$

Note that 2m speech samples must be acquired in order to compute C(m). Depending on the size of the speech frame and the maximum value of m, some terms in the summations may require look-ahead into the next speech frame.

The range of values of m should be the range of possible pitch-periods. It has been suggested [5, page 64] that the lowest possible pitch-frequency of human speech is about 50Hz and the highest possible pitch-frequency is about 500Hz. This corresponds to a pitch-period range of $16 \leq m \leq 160$ samples, for an 8kHz sampling frequency. Examples of sections of voiced and unvoiced speech are shown in figures 2.1 and 2.2 respectively, together with their cross-correlation functions.

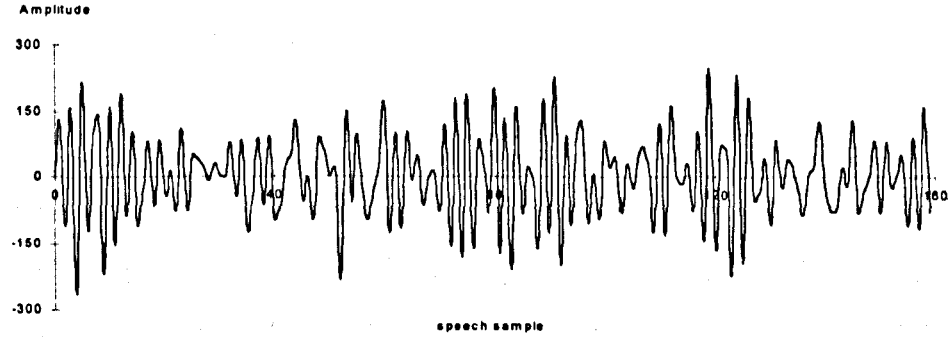


(a)

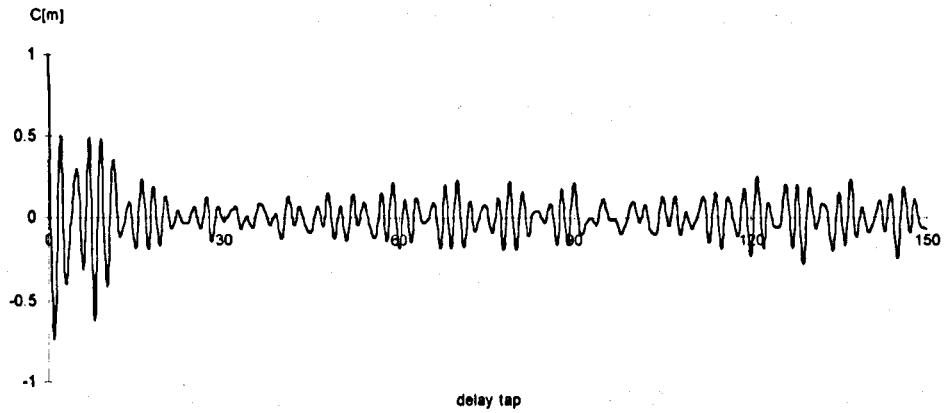


(b)

Figure 2.1 A frame of voiced speech together with the cross-correlation function.
(a) speech signal (b) cross-correlation function



(a)



(b)

Figure 2.2 A frame of unvoiced speech together with the cross-correlation function.
(a) speech signal (b) cross-correlation function

It may be seen in figure 2.1b that the cross-correlation function of the voiced speech segment has a maximum (close to unity) at

$$m = p$$

where p is the true pitch-period of the voiced speech.

The correlation function itself is a periodic signal with a period equal to the pitch-period of the current analysed voiced speech, i.e.

$$C(m) = C(m+p)$$

This is not the case for the unvoiced speech shown in figure 2.2a. The correlation function of the unvoiced speech is more random and the correlation values are generally very small across the entire range of values of m, figure 2.2b. As a result, the cross-correlation function can be used as a simple speech classifier by using a suitable correlation threshold. The correlation thresholds commonly used are 0.25 [30] and 0.3 [29]. Pitch estimation can be realised easily, in principle, by locating the global maximum of the correlation function. The cross-correlation function thus provides a straightforward means of pitch determination with low computational complexity suitable for real-time implementation.

2.2.2 A backward mode cross-correlation function

In contrast to the normal cross-correlation function defined above, a backward mode cross-correlation function takes the last sample of a speech frame as a starting point, and works backward towards the beginning of the frame [16], i.e.

$$C_b(m) = \frac{\sum_{n=N-1}^{N-m-1} s(n)\, s(n-m)}{\sqrt{\sum_{n=N-1}^{N-m-1} s^2(n) \sum_{n=N-1}^{N-m-1} s(n-m)}} \qquad (2.8)$$

where N is the number of speech samples in a frame.

The use of the backward mode cross-correlation function together with the forward mode function can make the detection of voicing transitions more reliable than when only the forward mode function is used [16][17]. Furthermore, it

increases the pitch estimation accuracy during voicing transitions. In figure 2.3a, a voicing onset frame is shown. The corresponding cross-correlation function in forward and backward modes are shown in figures 2.3b and c.

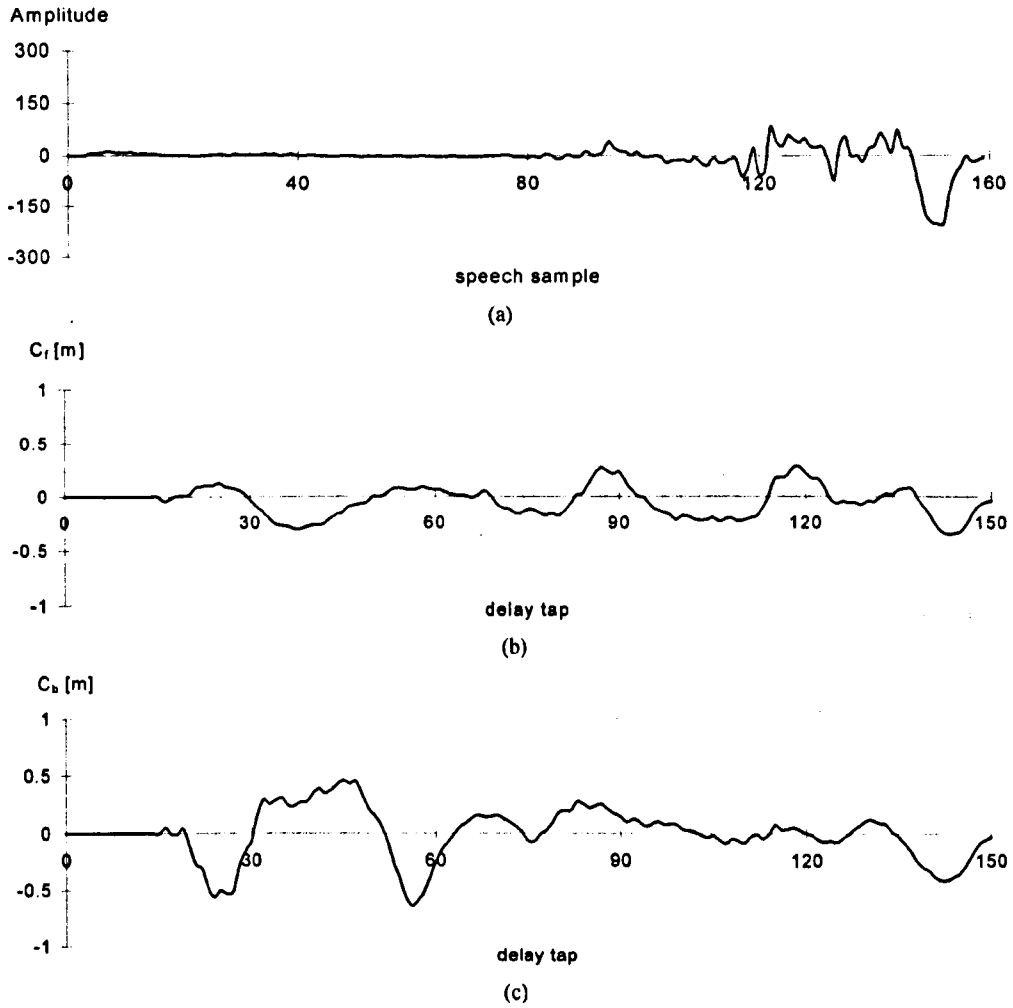


Figure 2.3 An example of voicing onset frame together with the cross-correlation functions in forward and backward mode. (a) speech signal (b) cross-correlation function in forward mode (c) cross-correlation function in backward mode

Owing to the lack of pitch information at the beginning of the speech onset frame, it is not possible to provide an accurate pitch estimation with using only the forward mode cross-correlation function. Using the backward mode method, a peak is found around the position of the true pitch-period. As a result, pitch estimation during the voicing onset can be made more reliable. A pitch detector using only the backward mode method would have the advantage of reducing the system delay, since no future information is required. This is seen in equation 2.8 that when the number of speech samples required is more than the size of a speech frame (i.e. $2m > N$), the extra speech samples are taken from the previous speech frame.

2.3 Pitch enhancement using spectral flattening

When voiced speech is produced, the glottal waveform is not a perfectly periodic pulse train and the formant frequencies may be changing significantly. This leads to considerable variations in the detailed structure of individual pitch-cycles. The resonances of the vocal tract largely determine the shape of these individual pitch-cycles. It has been claimed [27] that the reliability of a pitch detector can be increased by pre-processing the input speech signal using spectral flattening.

The purpose of spectral flattening is to remove the formant structure from voiced speech thus making the harmonics of the fundamental frequency equal in amplitude. The magnitude spectrum of the resulting waveform would then be the magnitude spectrum of a train of pulses separated by the slightly variable pitch-period. In the case of unvoiced speech, no such pseudo harmonic structure would be obtained. A primitive spectral flattening technique is that of centre clipping [27]. In one form of centre clipping technique, a threshold value (A_{th}) is computed from the maximum absolute amplitude of a speech frame. All portions of the speech bounded by $\pm A_{th}$ are removed and the speech samples which have an absolute value larger than A_{th} are retained. Thus the resulting speech, in principle, will contain peaks which are separated by the pitch-period, i.e. the formant structure is effectively removed. A comprehensive study of various centre clipping techniques was published in [30], where the results suggested that correlation peaks due to the formant structure of speech signal can be reducing by all the centre clipping techniques under investigation.

Spectral flattening can also be achieved by using a linear predictive (LP) analysis filter [32]. Passing speech through such a filter produces a "residual" signal r(n), whose magnitude spectrum is relatively flat. For voiced speech, the residual signal will, in principle, be a sequence of pulses with a separation equal to the current pitch-period. For unvoiced speech, the residual signal will be white Gaussian noise.

2.4 Pitch smoothing using median smoothing

A common problem with correlation based pitch detectors is the occurrence of what are known as "multiple-pitch" errors. For any truly periodic signal, the cross-correlation function will have a peak at the delay of one period and also at integer multiples of the period. The peaks will be equal in amplitude. Since speech is not truly periodic, it is anticipated that the peak at one pitch-period delay will be higher than the others. This is not likely to be a reliable distinction however, and in practice the peak at the multiple pitch-period positions may have a higher amplitude than one at the true pitch-period location. This is often found during voicing onsets when the periodicity of voiced speech is building up slowly.

When a multiple pitch-period peak is taken to be the true pitch-period peak, serious pitch-period estimation error occurs. Steps must be taken to eliminate the occurrence of such errors. One of the possible ways to get rid of such errors is known as non-linear smoothing, using a median smoother [28]. Median smoothing has been widely incorporated into pitch detectors, where the current estimated pitch-period is assessed together with a number of pitch-periods obtained from the previous and future speech frames. The median of the group of pitch-periods is chosen to be the output of the pitch detector. Using median smoothing, any sharp change in the estimated pitch-period, which may be caused by a multiple-pitch error, can be smoothed out and thus a smooth pitch-period trajectory can be anticipated.

An example of the pitch contour obtained by a correlation based pitch detector is presented in figure 2.4. It may be seen that two rapid changes in the pitch contour occur. These are caused by double-pitch errors. The double-pitch errors can be eliminated using a 3-point median smoother, in which an individual pitch-period is assessed together with the pitch-periods in the previous and the next frames. The median of the three is chosen to be the true pitch-period. Using the 3-point median smooth, the double-pitch errors indicated in figure 2.4a were successfully removed and a smooth pitch contour was obtained as shown in figure 2.4b.

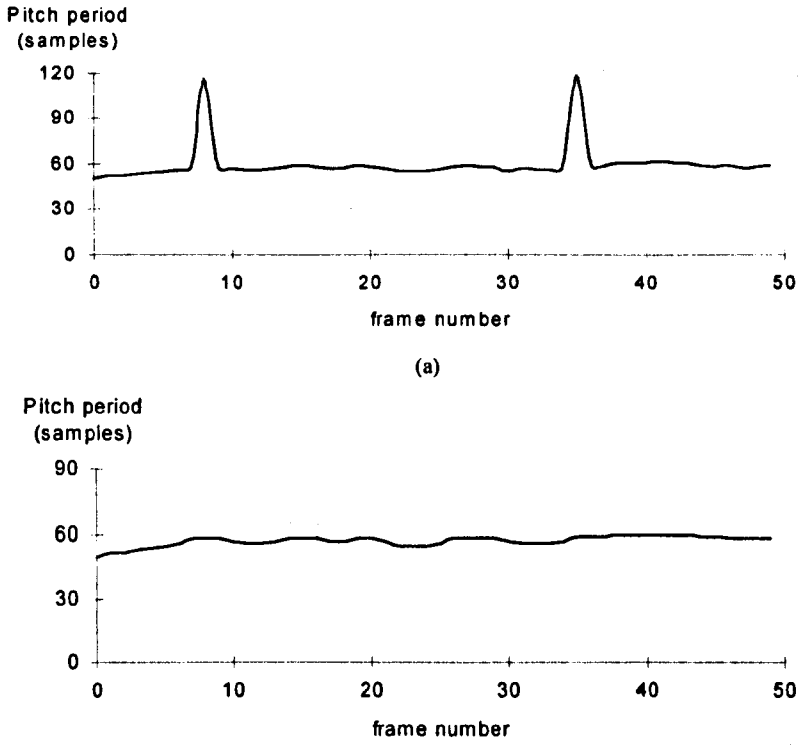


Figure 2.4 A pitch contour before and after application of a 3-point median smoother.
(a) original pitch contour (b) pitch contour after median smoothing

The order of a median smoother must be chosen carefully. The larger the order the more effective will be its ability to smooth the pitch contour. However, as the order increases so does the amount of smoothing that occurs during sharp pitch-period changes, for example at voicing transitions. This also increases the system delay. On the other hand, if the order of a median smoother is too small, the efficiency of error correction will be reduced. In general, the size of median smoothers can be varied from 3 to 9.

2.5 A two-way pitch detector

A two-way pitch detector has been developed in the project. The schematic diagram of the two-way pitch detector (TPD) is shown in figure 2.5 [38].

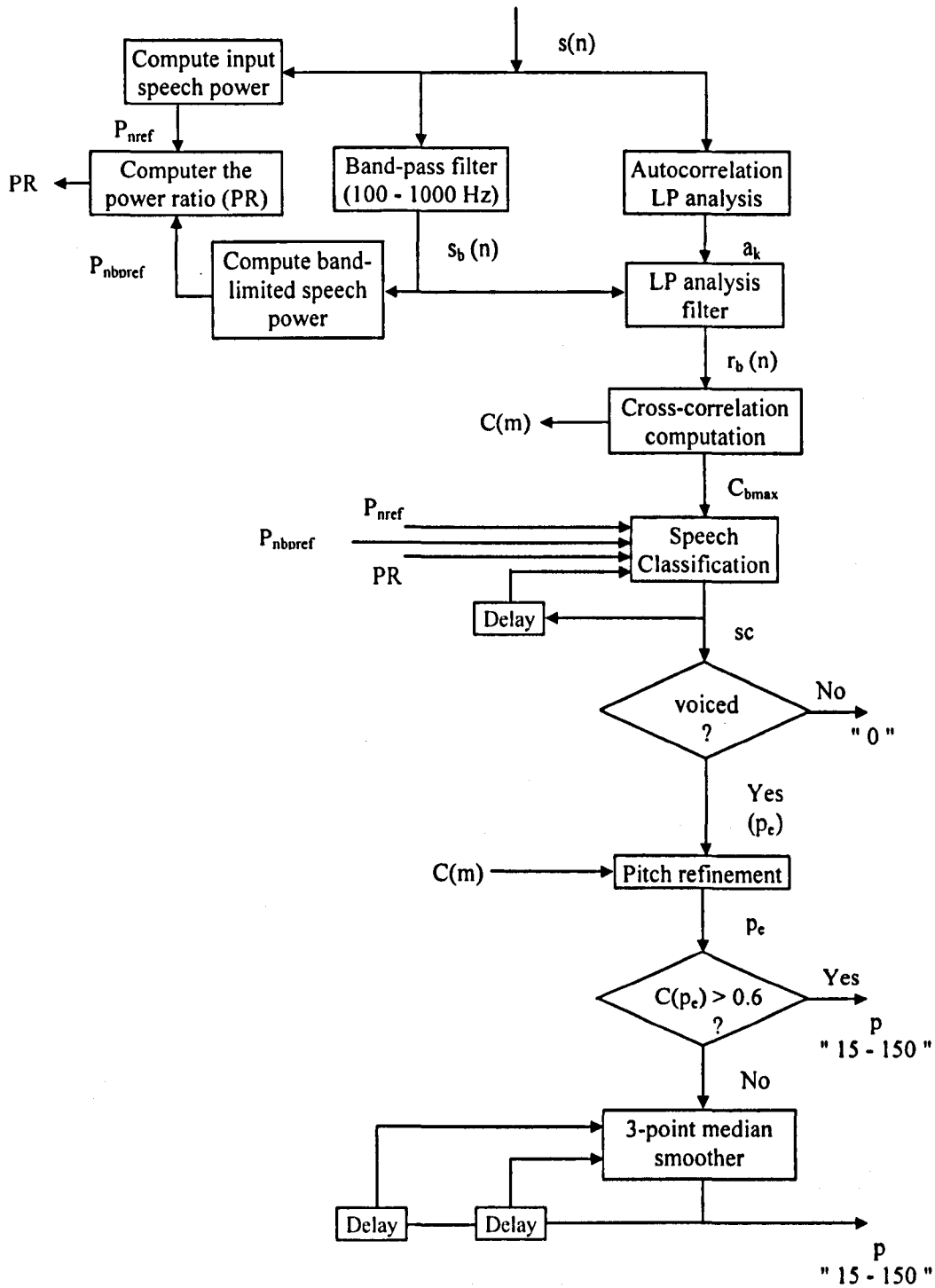


Figure 2.5 Schematic diagram of the two-way pitch detector (TPD).

The pitch detector consists of a pitch pre-processing unit, a speech classifier, a pitch estimator and a pitch post-processing unit. In the TPD, the input speech is first processed by the pitch pre-processing unit, in which the frequency range of the input speech signal is limited by a band-pass filter. Formant structure retained in the band-pass filtered speech signal is then removed by a 10th order FIR filter, with filter coefficients computed by LP analysis. In the speech classifier, the input speech is classified into either voiced or unvoiced speech, using a set of feature data. The features are extracted from the speech frame and from the corresponding cross-correlation function. The cross-correlation function is computed for the band-limited speech residual, taken from the output of the LP analysis filter. The output of the pitch detector is set to zero when unvoiced speech is indicated, otherwise the pitch estimator is called upon. The pitch estimator locates the position of the global maximum in the correlation function and uses it as the estimated pitch-period for the current speech frame. The range of pitch-periods is chosen to be 16 to 150 samples (8kHz sampling frequency). The estimated pitch-period is then processed by the pitch post-processing unit, in which pitch-period refinement is carried out in order to avoid possible multiple-pitch errors. Afterwards, the correlation value associated with the refined pitch-period is examined. If the correlation value is larger than a pre-defined threshold, it is directly used as the output of the pitch detector. Otherwise, a 3-point median smoother is deployed. The 3-point median smoother operates on the current estimated pitch-period and the pitch-periods of the two previous consecutive speech frames, the median of the three being chosen as the output of the pitch detector.

2.5.1 The pitch pre-processing unit

Prior to the speech classification, the input speech is segmented into 160 sample frames (i.e. 20ms frame in a 8kHz sampling frequency). A 10th order autocorrelation LP analysis is performed on the current speech frame to yield a set of filter coefficients. A 200 sample asymmetric window, which includes 40 speech samples from the previous frame, is deployed to extract the speech samples for the LP analysis as illustrated in figure 2.6.

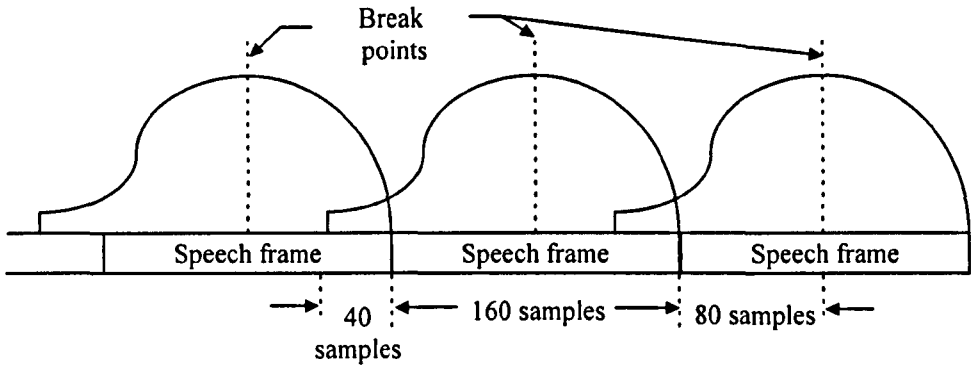


Figure 2.6 The asymmetric window used in the TPD.

The first part of the window consists of a half Hamming window and the second part is a quarter of a cosine function. The break-point between the two parts is located in the middle of the speech frame. The asymmetric window is defined as,

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{2M-1}\right) & 0 \le n \le M-1 \\ \cos\left[\dfrac{2\pi\,(n-M)}{4(N_w-M)-1}\right] & M \le n \le N_w-1 \end{cases} \tag{2.9}$$

where

M is the break-point between the two functions

N_w is the length of the asymmetric window

Through the use of the asymmetric window, no future samples are required in the LP analysis. Details of the asymmetric window will be discussed in chapter 3.

At the same time, the frame of speech signal is frequency limited using a band-pass filter. The band-pass filter is composed of a 2nd-order IIR low-pass section and a 2nd-order IIR high-pass section cascaded. The high-pass filter is included to reject low frequency noise arising, for example, from capacitive pick-up from the 50Hz mains supply. The high-pass filter will also reduce car noise for which most of the energy is concentrated at frequencies below about 100Hz. The cut-off frequency of the high-pass section is set to 100Hz. The transfer function is [10],

$$H_{hf}(z) = \frac{0.9398058 - 1.8795834z^{-1} + 0.93980581z^{-2}}{1 - 1.9330735z^{-1} + 0.93589199z^{-2}} \tag{2.10}$$

The aim of the low-pass filter is to reduce the effects of formant structure which can tend to obscure the true pitch-period. The filter is also beneficial in removing a substantial amount of high frequency noise (e.g. babble noise) which may contaminate the speech signal. The cut-off frequency of the low-pass section is set to 1kHz. The transfer function is,

$$H_{lf}(z) = \frac{0.097631 + 0.195262z^{-1} + 0.097631z^{-2}}{1 - 0.942809z^{-1} + 0.333333z^{-2}} \tag{2.11}$$

After passing through the band-pass filter, the speech signal is processed by an LP analysis filter with the current set of filter coefficients. A band-limited speech residual is thus obtained.

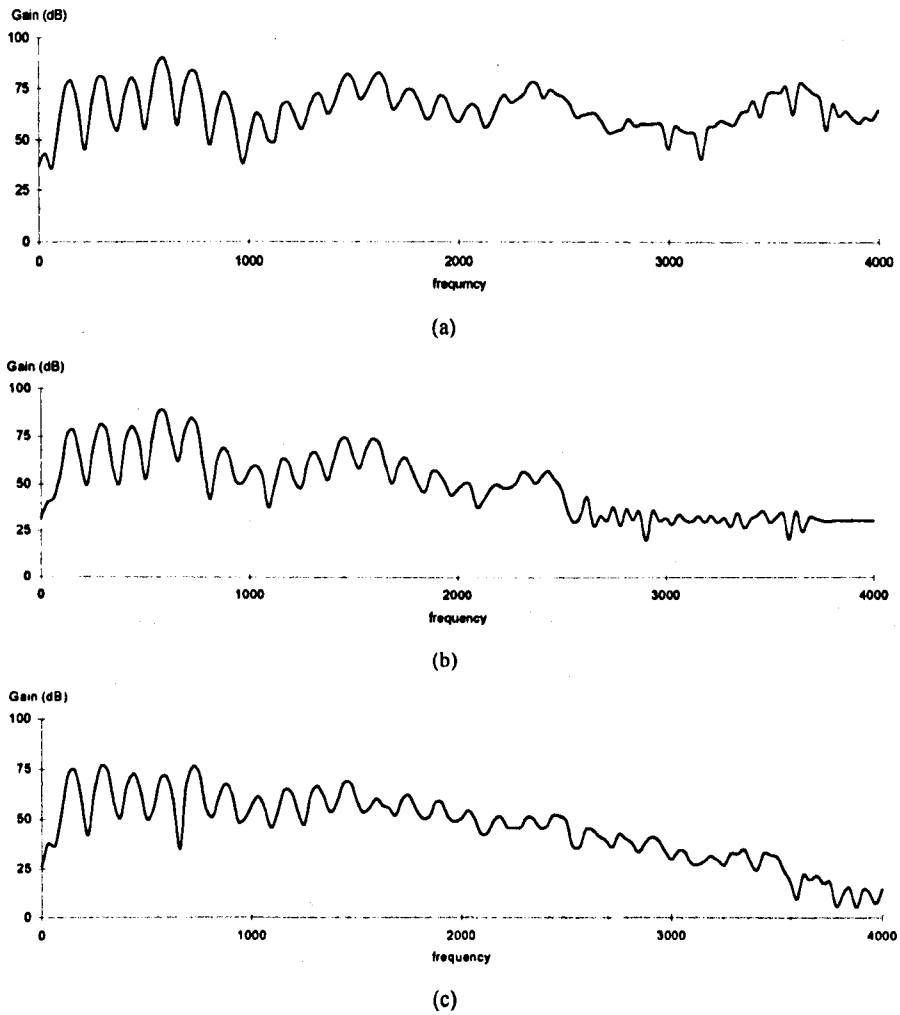


Figure 2.7 Frequency spectra of the signal in various stages of the pitch pre-processing unit. (a) input speech signal (b) band-limited speech signal (c) band-limited speech residual

In figure 2.7, frequency spectra of a section of voiced speech at various stages of the pitch pre-processing unit are presented. A well-defined formant structure is seen in the spectral envelope of the voiced speech, in figure 2.7a. The harmonic peaks due to pitch periodicity are also clear at low frequencies, though they become blurred at high frequencies because of small variations. By measuring the distance between the peaks in the low frequency region, the pitch-frequency may be found to be about 150Hz. A peak is also found at about 50Hz due to the 50Hz main supply. The energy of the high frequency components as well as the 50 Hz component should be attenuated by the band-pass filter. This is seen to have happened in figure 2.7b which is the magnitude spectrum band-limited between 100Hz and 1KHz. In figure 2.7b some formant structure of the original speech signal still remains but this can be effectively removed using a 10th order LP analysis filter. In figure 2.7c, the frequency spectrum of the band-limited LP residual is shown. This demonstrates that the first formant has been eliminated and a relatively flat magnitude spectrum has been obtained at the lower half frequency region. The magnitudes at the upper half frequency band is rolling off with increasing frequency. It will be shown in the next section that the reliability of a pitch detector may be increased by using a speech residual with frequency characteristics as illustrated in figure 2.7c, rather than one with a flat magnitude spectrum across the entire frequency band.

2.5.2 The speech classifier and the pitch estimator

After the pitch pre-processing unit, a rectangular window is used to extract the segments of the band-limited speech residual, $r_b(n)$ and $r_b(n-m)$, used to compute the backward mode cross-correlation function. The rectangular window is positioned within the frame of band-limited speech residual as shown in figure 2.8.

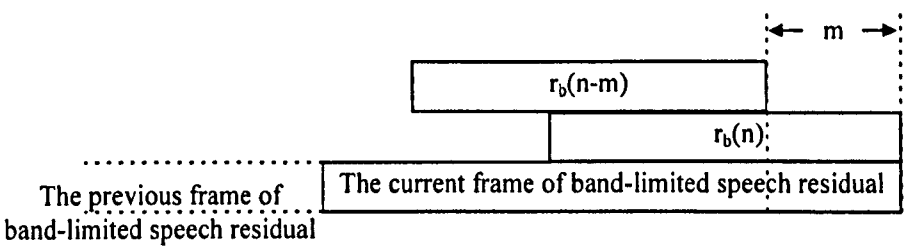


Figure 2.8 Positioning of the rectangular window to extract the segments of band-limited speech residual for computing the backward mode cross-correlation function

The first problem encountered is to choose the size of the analysis window. Errors occur if too few samples are contained in the window since there will be insufficient periodic structure. However, a large analysis window may lead to a pitch-period estimation error in the case of voiced speech with rapidly changing pitch-period. This is very often observed during voicing transitions. A fixed window has been chosen with the window length long enough to accommodate at least one pitch-cycle at the maximum pitch-period, i.e. 150 samples. The window is located at the end of a residual frame.

2.5.2.1 Backward mode cross-correlation function using a band-limited LP filtered residual

The backward mode cross-correlation function is computed because, a) it requires no looking forward to future speech samples, b) it allows detection of a voicing onset. Furthermore, a voicing offset frame could be defined if an unvoiced frame is indicated directly after a voiced frame. The backward mode cross-correlation function is,

$$C_b(m) = \frac{\sum_{n=N-1}^{10} r_b(n) r_b(n-m)}{\sqrt{\sum_{n=N-1}^{10} r_b(n)^2 \sum_{n=N-1}^{10} r_b(n-m)^2}} \qquad (2.12)$$

where r_b is the band-limited speech residual. In this case some terms require looking back to the previous frame.

The range of pitch-periods is set to be from 16 to 150 samples, corresponding to the range of pitch frequencies from 500 to 53.3 Hz, for a 8kHz sampling frequency system. Experiments [39] have shown that this pitch-period range is adequate in practice.

The band-limited speech residual is used in order to enhance the performance of the pitch estimator. In figure 2.9, cross-correlation functions for a frame of voiced speech, for the corresponding speech residual and for the band-limited speech residual are shown. In all the three cases, the values of C(m) for the first 15 delay taps have been set to zero.

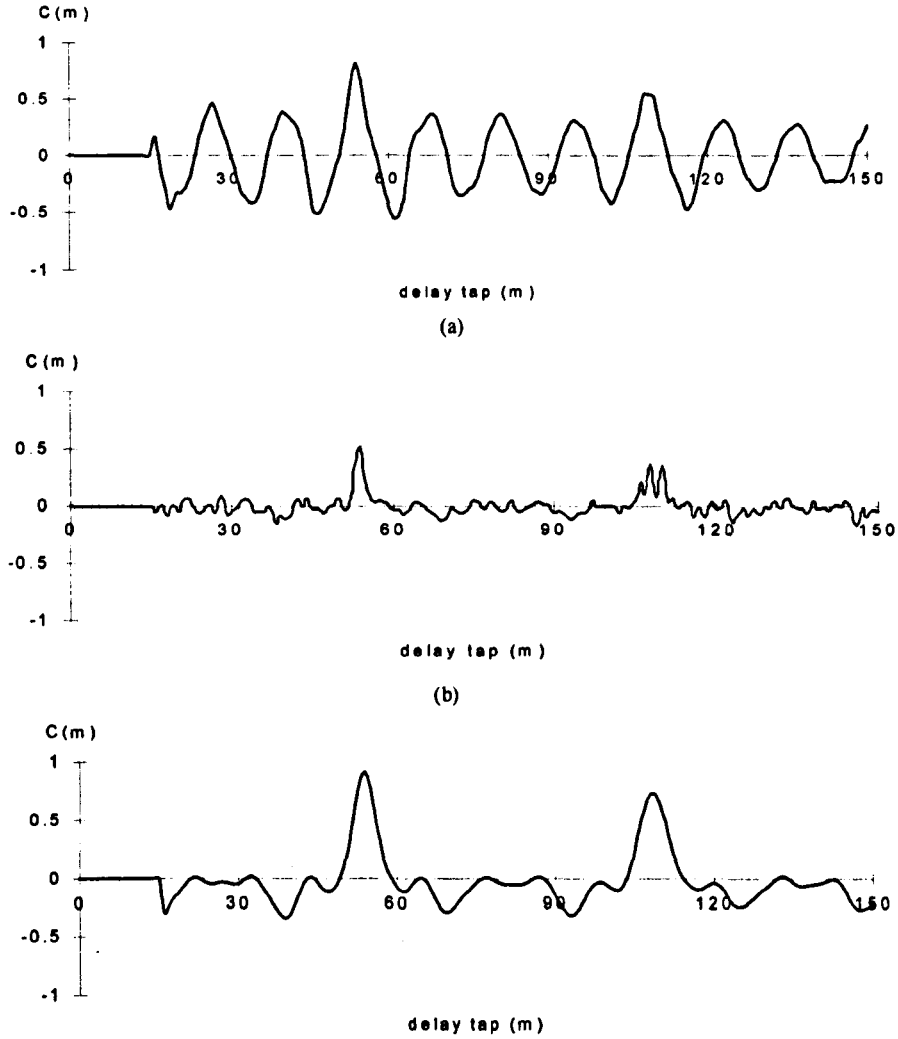


Figure 2.9 Cross-correlation functions computed from (a) a frame of speech signal (b) the corresponding speech residual (c) the corresponding band-limited speech residual

The correlation function in figure 2.9a shows a clear peak structure with two main peaks, one at the true pitch-period about 55 sampling intervals and one at double the true pitch-period, i.e. at about 110 samples. Sidelobes with substantial correlation values are also evident. These sidelobes are caused by the formant structure existing in the speech signal. They could badly affect the performance of the pitch detector. With the use of an LP analysis filter to achieve a degree of spectral flattening, it may be seen in figure 2.9b that the correlation function of the speech residual exhibits a much sharper peak structure and that the sidelobes due to speech formants have been effectively attenuated. The disadvantage of using this function is that it is rather noisy and the correlation peaks are low. In figure 2.9b, three consecutive peaks with almost the same correlation values are found around the double-pitch position. This increases the difficulty of pitch estimation, and may lead to an estimation error. The

correlation function in figure 2.9c show a much clearer peak structure with a higher correlation value at the pitch position. This illustrates the reason why a band-limited speech residual was chosen for computing the correlation function.

2.5.2.2 The feature measures used in the speech classifier

To classify a speech frame, the classifier extracts a number of feature measurements from both the current speech frame and the cross-correlation function. The feature measurements used are,

a) Scaled input speech power (P_{nref}),

$$P_{nref} = \frac{\frac{1}{N}\sum_{n=0}^{N-1} s(n)^2}{A} \tag{2.13}$$

where N is the frame length (160 samples) and A is numerically equal to the maximum speech amplitude allowed in the speech coder, thus $0 \leq P_{nref} \leq A$.

b) Scaled band-limited speech power (P_{nbpref}),

$$P_{nbpref} = \frac{\frac{1}{N}\sum_{n=0}^{N-1} s_b(n)^2}{A} \tag{2.14}$$

where s_b is the band-limited speech signal

c) Power ratio between P_{nref} and P_{nbpref} (PR),

$$PR = \frac{P_{nref}}{P_{nbpref}} \tag{2.15}$$

d) Global maximum of the cross-correlation function (C_{bmax}),

$$C_{b\max} = \max_{m}\{C_b(m)\} \tag{2.16}$$

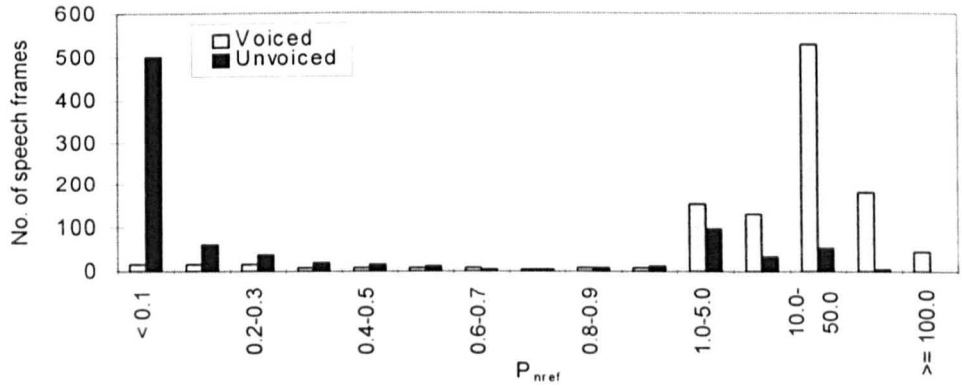
The speech classifier must search through the cross-correlation function to yield this global maximum. Based on the feature data, the speech classifier utilises a statistical approach to make a voiced/unvoiced decision. In figure 2.10, a set of histograms corresponding to the features are shown. The histograms were constructed by

examining a section of the speech file "GSP.DAT" [20] provided by British Telecom. The GSP.DAT file contains 10 minutes of natural speech band-limited from 0 to 3.4kHz and is sampled at 8kHz. Two thousand speech frames from the speech file, all of which were recorded from a radio station, were used to form a training set with voiced/unvoiced decisions marked by hand. Note that the x-axis of the histograms in figures 2.10a to c are not on a linear scale to save space.

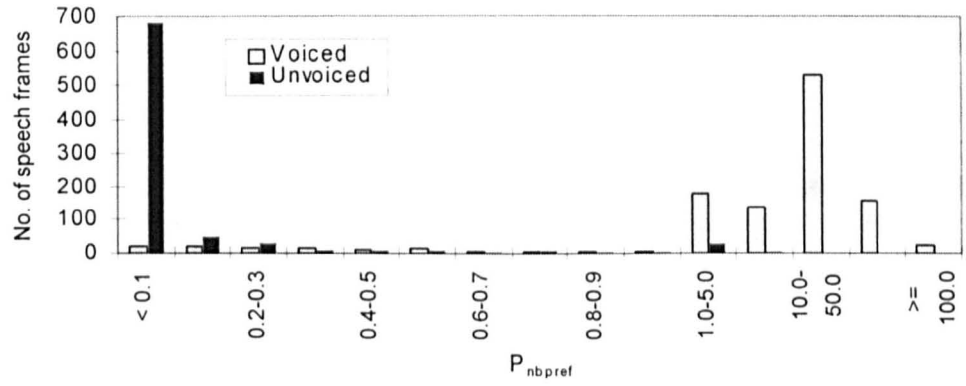
The histogram in figure 2.10a shows that many of the unvoiced frames have P_{nref} less than 1.0. Most of these frames correspond to silence in the training set. It may be observed that there are a considerable number of unvoiced frames which have P_{nref} greater than 1.0. All of these frames correspond to unvoiced speech. The P_{nref} for the voiced frames is generally greater than 1.0. Figure 2.10b is a histogram of the P_{nbpref}. The unvoiced frames are concentrated on the left hand side of the 1.0 line whereas most of the voiced frames lie to the right. The histogram in figure 2.10c shows that the PR measure during voiced frames is generally very low, less than 2.5. It was suggested [4, page 267] that the frequency ranges of the first three formants are,

$$200Hz \leq F_1 \leq 900Hz$$
$$550Hz \leq F_2 \leq 2700Hz$$
$$1100Hz \leq F_3 \leq 2950Hz$$

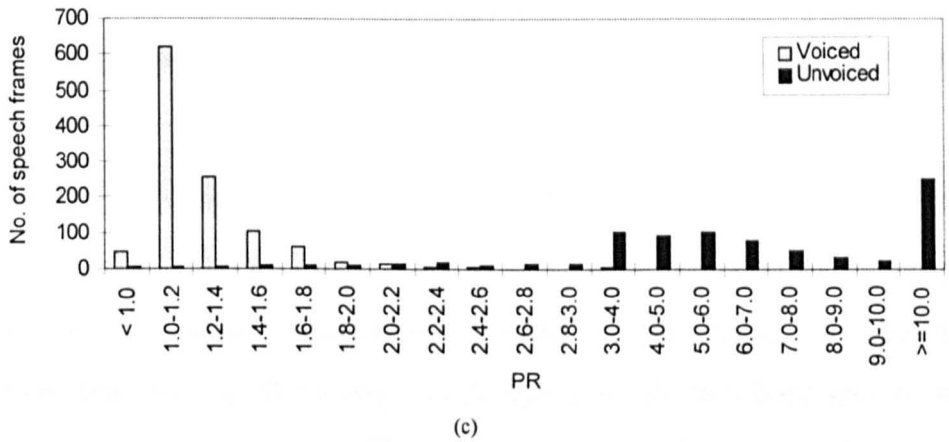
The 100Hz to 1kHz band-limited speech signal at least contains the first formant and possibly the second formant, in some vowels. This suggests that a considerable amount of energy may remain after band-pass filtering and hence a low PR would be expected. Unvoiced speech is considered to be approximately Gaussian with a white frequency spectrum. Therefore it is reasonable to expect that the PR of an unvoiced speech would be greater than 3.0. Note that there were some speech frames which had a PR less than 1.0. This was due to the delay between the input and output signal of the band-pass filter. Many of these frames occurred during voicing offsets. Finally a histogram of the maximum cross-correlation function is shown in figure 2.10d. A Gaussian shaped distribution is observed for the unvoiced frames, whereas an approximately exponential distribution appears for the voiced frames.



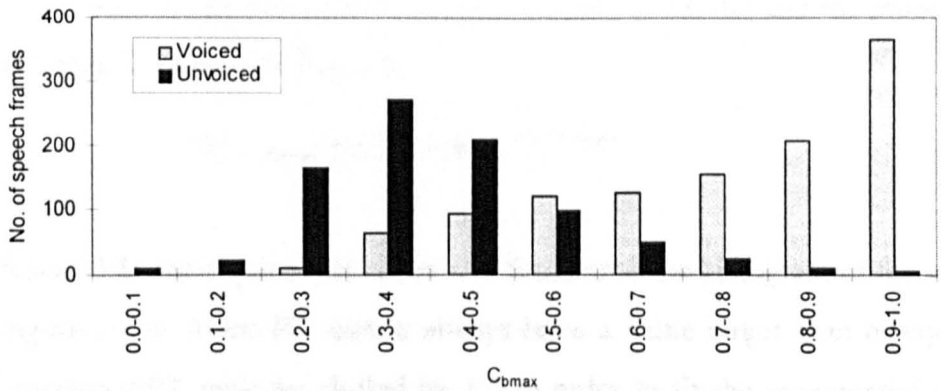
(a)



(b)



(c)



(d)

Figure 2.10 Histograms of the feature data constructed from a 2000 frame training set. (a) scaled speech power (b) scaled band-limited speech power (c) power ratio between the speech and the band-limited speech (d) maximum backward mode cross-correlation function.

2.5.2.3 Computation a voiced confidence level

Results from the histograms suggest that each feature may contain information about the probability of voicing for a speech frame. In figure 2.10a, the histogram for the scaled speech power p_{nref} of the unvoiced frames shows an approximately exponential shape. An exponential distribution function may be fitted to the histogram to produce an estimate of the probability of voiced speech for a range of values of measured P_{nref}, assuming that voiced and unvoiced speech are mutually exclusive. The probability density function of an exponential distribution is given by,

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.17}$$

where $1/\lambda$ is the mean of the samples. Furthermore the cumulative distribution function the exponential distribution is given by,

$$F(x) \begin{cases} 1.0 - e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.18}$$

The λ of the unvoiced P_{nref} was calculated as 10.058 and thus the probability of voiced speech for a measured P_{nref} is then,

$$\Pr\left(P_{nref} \right) = 1.0 - \exp^{-10.058\, P_{nref}} \tag{2.19}$$

Similarly for the scaled band-limited speech power P_{nbpref}, an exponential distribution function was fitted onto the histogram of the unvoiced speech frames, figure 2.11b. The λ of the unvoiced P_{nbpref} is computed as 14.163 and the probability of a voiced speech for a given P_{nbpref} is,

$$\Pr\left(P_{nbpref} \right) = 1.0 - \exp^{-14.163\, P_{nbpref}} \tag{2.20}$$

For the feature PR, the exponential curve was fitted onto the histogram of the voiced frames, figure 2.11c. Since PR should always have a value larger than or equal to 1.0, the measured PR must be shifted by 1.0 in order to fit the exponential curve. The λ of the voiced PR was computed as 1.794 and the probability of a voiced frame for a given PR is then,

$$\Pr(PR) = 1.0 - \left(1.0 - \exp^{-(1.794-1.0)(PR-1.0)}\right)$$
$$= \exp^{-0.794(PR-1.0)} \quad (2.21)$$

The maximum cross-correlation function, which indicates level of periodicity in a speech frame, is used directly as an indication of the probability of voicing for the category.

It has been discussed that a probability of voicing could be obtained from each of the four measured feature data for a speech frame. To include the information provided by all the four measured feature data, we defined a voice confidence level (VL). The voice confidence level is defined as the scaled sum of the probability of voicing due to each feature data measured for the current speech frame and the speech class for the previous speech frame,

$$VL^{(l)} = 0.2 * \left(\Pr\left(P_{nref}^{(l)}\right) + \Pr\left(P_{nbpref}^{(l)}\right) + \Pr\left(PR^{(l)}\right) + C_{b\max}^{(l)} + SC^{(l-1)}\right) \quad (2.22)$$

where $SC^{(l-1)}$ is a classification of the $(l-1)$th speech frame, which is "1" for a voiced frame and "0" for an unvoiced frame

Note that by including the previous speech class in computing the voicing confidence level, an adaptive aspect is built into the confidence measure.

2.5.2.4 Speech classification

Speech classification is carried out by a sequence of logic decisions as shown in figure 2.11. It was mentioned in section 2.5.2.2 that no speech frame should have a PR less than one, the classification of a speech frame is set to be the same as the previous frame if the PR of a speech frame is less than 1.0. Figure 2.10c provides clear guidance about making a voiced/unvoiced decision. It is shown that the probability of a speech frame with a low PR being unvoiced is very small. On the other hand, the PR for a voiced frame can never be very high. A fixed threshold is set for PR at 3.0 meaning that a frame would be declared unvoiced if the PR of the frame is greater than 3.0. For the speech frames which have a PR between 1.0 and

3.0, the voice confidence level is computed and this is compared to a threshold value such that,

$$sc = \begin{cases} 0\ (Unvoiced) & if\ VL < 0.5 \\ 1\ (Voiced) & if\ VL \geq 0.5 \end{cases} \tag{2.23}$$

where sc is the speech class of a speech frame

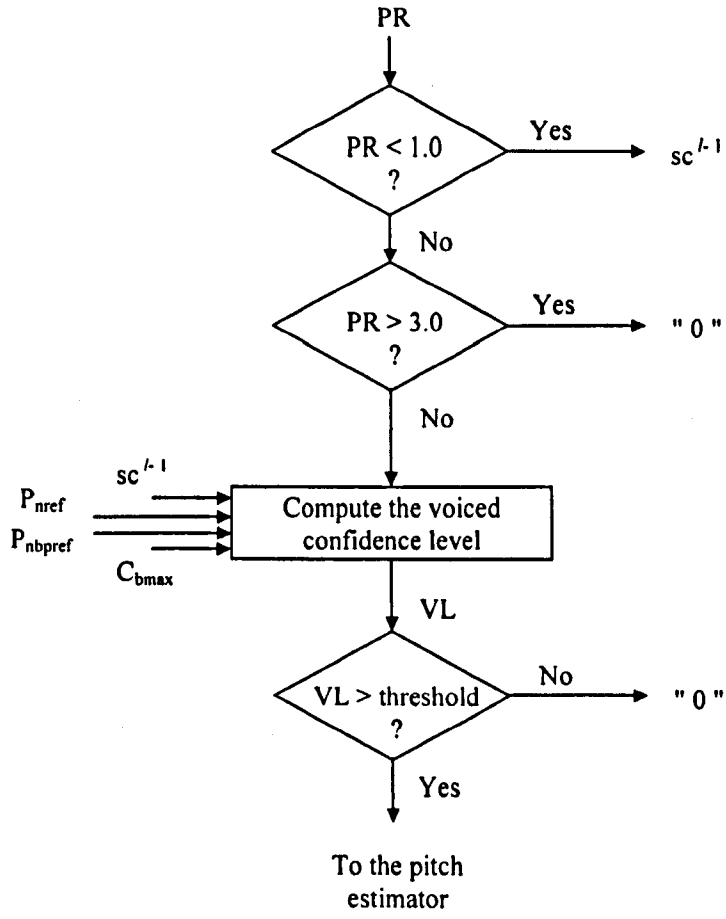


Figure 2.11 Schematic diagram of the speech classifier

2.5.2.5 Pitch estimation

When the speech classifier indicates that a frame of speech is voiced, the pitch estimator uses the delay associated with the maximum cross-correlation function as the estimation of the true pitch-period. The estimated pitch-period would be further verified by the pitch post-processing unit. Otherwise, the pitch-period would be assumed to be zero.

2.5.3 The pitch post-processing unit

The pitch post-processing unit consists of two parts: a pitch-period refinement unit and a 3-point median smoother. The pitch-period refinement unit searches through a number of potential pitch-period candidates and performs a second estimation of the true pitch-period based on a forward mode cross-correlation function. The second part of the pitch post-processing unit is a 3-point median smoother which is only applied when the cross-correlation function associated with the estimated pitch-period is less than 0.6 (refer to figure 2.5). The 3-point median smoother operates on the current estimated pitch-period and the values of pitch-period obtained for the two previous consecutive speech frames. The median of the three would be set as the final pitch-period.

2.5.3.1 The pitch-period refinement unit

In the pitch-period refinement unit, after the global maximum of the cross-correlation function and its corresponding delay have been determined, an adaptive correlation threshold is computed.

$$C_{th}^{(l)} = 0.7 * C_{b\,\max}^{(l)} \tag{2.24}$$

Any local maximum of the correlation function with a delay smaller than the delay associated with the global maximum and has a correlation value larger than the adaptive threshold would be classified as a potential candidate (C_k). The potential candidates, C_1, C_2, ... , C_g, are arranged in an ascending order, i.e. $C_1 < C_2 < ... < C_g$. To determine the true pitch-period, the forward mode cross-correlation function is deployed. It is defined as,

$$C_f(k) = \frac{\sum_{n=-140}^{10} r_b(n) r_b(n+k)}{\sqrt{\sum_{n=-140}^{10} r_b^2(n) \sum_{n=-140}^{10} r_b^2(n+k)}} \tag{2.25}$$

$$k = C_1, C_2, \ldots, C_g$$

The first candidate which has a forward-mode cross-correlation function larger than the adaptive threshold would be classified as the estimated pitch-period. The pitch-period refinement procedure ceases when such a candidate has been found. If none

of the potential candidates fulfils this criterion, the delay of the global maximum would be used as the estimated pitch-period. After refining the pitch-period, the correlation function associated with the estimated pitch-period is examined. If it is greater than a fixed threshold (0.6), the pitch-period would be directly used as the output of the pitch detector. Otherwise, the 3-point median smoother would be called in to apply non-linear smoothing on the estimated pitch-period.

2.5.3.2 The 3-point median smoother

The 3-point median smoother operates on the current estimated pitch-period and the estimated pitch-periods obtained for the two previous consecutive frames, $p_e^{(l)}$, $p^{(l-1)}$, $p^{(l-2)}$. The median of the three estimates is set to be the output. The 3-point median smoother is different from the one discussed in section 2.4, in that its ability in smoothing a pitch contour is better than the latter. In figure 2.12a, an example of pitch contour is shown. Two consecutive double-pitch errors are seen on the pitch contour. Using the 3-point median smoother discussed in section 2.4, the double-pitch errors cannot be rectified as illustrated in figure 2.12a. By smoothing a new pitch-period with the two previously smoothed ones, the two consecutive double-pitch errors have been successively smoothed out, as shown in figure 2.12c.

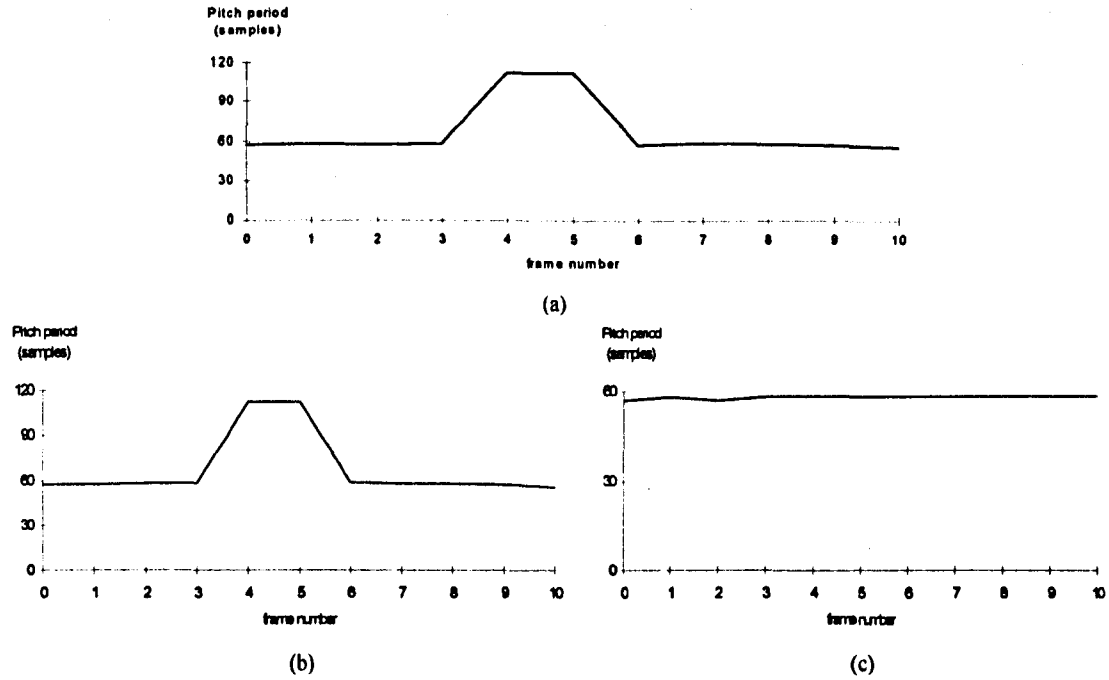


Figure 2.12 An example of pitch contour in which two consecutive double-pitch errors are found. (a) original pitch contour (b) pitch contour after being smoothed by a 3-point median smoother proposed in section 2.4 (c) pitch contour after smoothed by a 3-point median smoother proposed in this project

The disadvantage of such a 3-point median smoother is that it is easily trapped into a single pitch-period. In figure 2.12c, it is seen that the pitch-period stayed at about 58 samples after the double-pitch errors had been smoothed out. To allow the median smoother to jump out from this kind of trapped pitch-period, it is only used when the cross-correlation function of the estimated pitch-period is smaller than a fixed threshold. This is because if the cross-correlation function at the estimated pitch-period is relatively high, a highly periodic signal is under consideration. and the probability of an accurate estimation is high. By directly using the output of the pitch-period refinement unit as the estimated pitch-period in this case, the pitch detector is allowed to jump out from the trap error. In other words, the pitch detector recognises a rapid pitch-period change in the speech signal. Using such a median smoother instead of the one proposed in section 2.4, the delay introduced by the pitch detector can also be reduced since no future pitch-periods are required.

There are two special situations where no previous pitch-periods are available. They can occur at voicing onset frames and at voiced speech frames immediately following voicing onset frames. When a voicing onset frame is detected, the pitch detector would directly supply the pitch-period estimated by the pitch-period refinement unit as its output. In the case of a speech frame just after the voicing onset frame, the output of the pitch-period refinement unit would be used directly if the cross-correlation function of the estimated pitch-period is greater than 0.5. Otherwise, the output is deduced by averaging the current estimated pitch-period and the pitch-period of the onset frame.

2.6 Performance evaluation of the two-way pitch detector

In this section the performance of the TPD will be evaluated. This includes the performance of each individual unit in the TPD. The tests were carried out using a clean speech file as well as files containing speech contaminated by various types and levels of noise. A number of measuring parameters used to assess the performance of the TPD will also be introduced.

2.6.1 Creating the reference pitch-period file and the noisy speech files

The performance of the pitch detector was evaluated using a new speech file, "OPERATOR.DAT" [21]. The speech file contains a 36 second conversation (1800 frames) between a male and a female speaker. The speech signal is band-limited from 0 to 3.4kHz and sampled at 8kHz. The pitch-period of the file was manually marked by the author [39]. The manually adjusted estimates of pitch-period were used as a reference, against which the pitch-periods estimated by the pitch detector could be compared. The reference pitch-period file was created using a modified version of the Semi-automatic pitch detector (SAPD) [26], where only the pitch marker and the autocorrelation pitch detector were used (the cepstrum pitch detector has been left out).

2.6.1.1 Creating the reference pitch-period file

During the pitch marking, the speech file was segmented into 200 sample frames. The speech signal was displayed on the computer screen together with a previous and a future speech frame. A marker was placed on the zero-crossing position just before the peak pulse of each pitch-cycle. The distance between adjacent pitch markers was measured to obtain a reference pitch-period. The pitch-period thus obtained was assigned to each sample of the entire pitch-cycle as the instantaneous pitch-period of the sample.

Simultaneously in the autocorrelation pitch detector, a rectangular window was centred on the speech sample under analysis to extract a frame of speech signal. The normalised autocorrelation function was then computed as follow,

$$A(m) = \frac{\sum_{n=0}^{N_w - m + 1} s_w(n)\, s_w(n+m)}{\sqrt{\sum_{n=0}^{N_w - m + 1} s_w^2(n) \sum_{n=0}^{N_w - m + 1} s_w^2(n+m)}} \qquad (2.26)$$

where

s_w is the windowed speech signal

m is the interested pitch-period range, 16 to 200 samples corresponding to a pitch-frequency of 500 to 40Hz

N_w is length of the rectangular window and is set to 300 samples, i.e. it contains 50 overlapping samples at both ends

The location of the maximum autocorrelation function was found and this is used as the instantaneous pitch-period of the speech sample being analysed. The procedure was carried out for each individual sample over the entire speech file.

The pitch-periods estimated by hand and by the autocorrelation methods were compared to create a reference pitch-period file for the tested speech file. If they were close to each other, the pitch-period estimated by the autocorrelation pitch detector was used. Otherwise, the final decision was made based on the pitch-period estimated manually. A pitch-period of "0" was recorded for each sample during periods of silence whilst "10" was assigned to each sample during unvoiced speech. It was found that the autocorrelation pitch detector was always able to estimate an accurate pitch-period or integer multiples of it, during a stable voiced speech. In these cases, the pitch-periods provided by the pitch marker were useful to verify the estimates from the autocorrelation pitch detector. The autocorrelation pitch detector performed poorly during voicing transitions. In these circumstances, the estimation was biased to the result provided by the pitch marker.

2.6.1.2 Creating the noisy speech files

The pitch detector was tested using clean speech as well as speech with various noise backgrounds. The noise types investigated were white-Gaussian noise (White) [93], car noise (Car) [90], babble noise (Babble) [91] and multi-speaker

(Multi) noise [92]. The noise samples were again provided by British Telecom. In figures 2.13, average frequency spectra for the four noise types are shown.

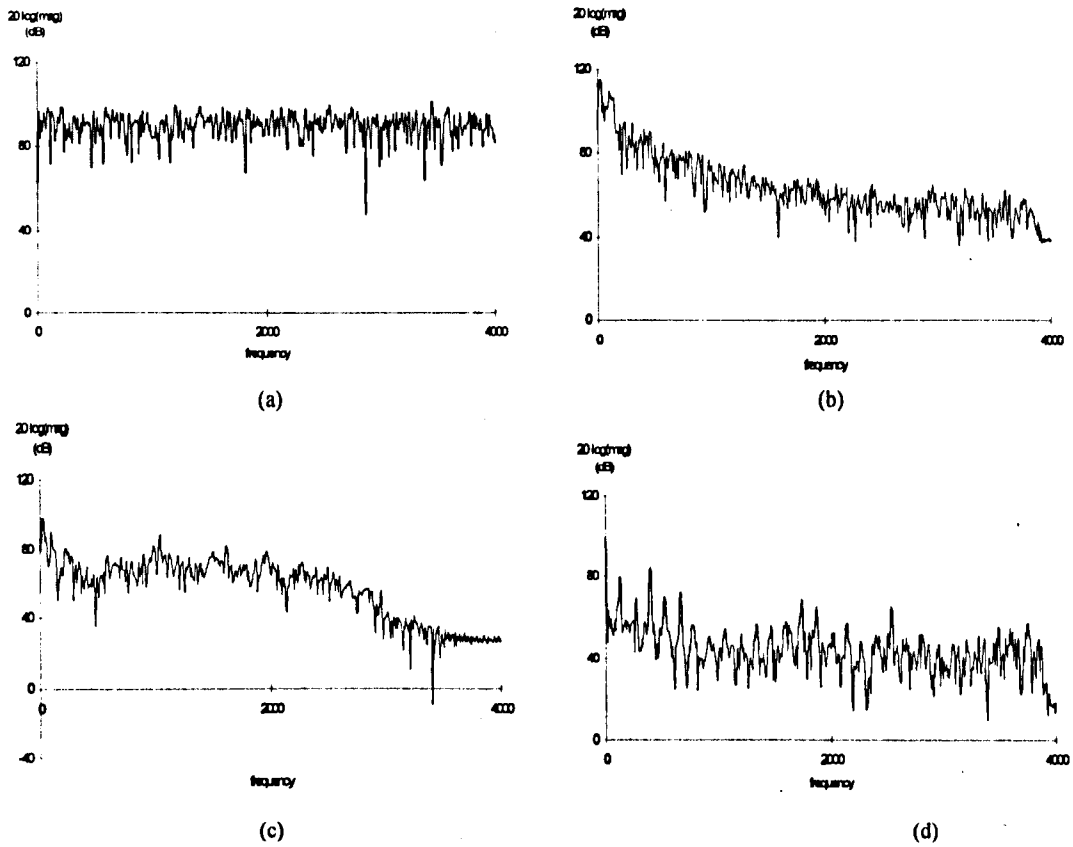


Figure 2.13 Frequency spectra of the noise files used in the test.
(a) white noise (b) car noise (c) babble noise (d) multi-speaker noise

In figure 2.13a, a flat spectrum is seen for the white noise. In figure 2.13b, the energy of car noise is at low frequencies especially below 100Hz. Frequency spectra of the babble noise are different from one frame to another. One of the typical examples is shown in figure 2.13c, in where it is seen that the energy is evenly spread from d.c. to 2kHz. Finally, the frequency spectrum of the multi-speaker noise is fairly flat with a virtual formant structure, which may correspond to the background speakers.

Normalised noise files were created by normalising the rms value of each of the noise files, i.e. for the four noise types, to be exactly the same as the rms value of the entire speech file. Noisy speech files with signal-to-noise ratio (SNR) level of 0dB, 10dB, 20dB and 30dB were then created by adding to the clean speech file a noise file attenuated by 0dB, 10dB, 20dB and 30dB respectively.

2.6.2 Error parameters used in the experiments

A number of error parameters have been suggested in the literature [25], in order to evaluate the performance of a pitch detector for a lengthy example of typical speech. They have been used in here with some modifications:

(i) Classification accuracy

a. Unvoiced classification accuracy Acc_{uv},

$$Acc_{uv} = \frac{M_{uv} - N_{uverr}}{M_{uv}} \times 100\ (\%) \tag{2.27}$$

where

M_{uv} is the total number of unvoiced frames

$N_{uv\ err}$ is the number of unvoiced frames being mis-classified as voiced frames

b. Voiced classification accuracy Acc_v,

$$Acc_v = \frac{M_v - N_{verr}}{M_v} \times 100\ (\%) \tag{2.28}$$

where

M_v is the total number of voiced frames

$N_{v\ err}$ is the number of voiced frames being mis-classified as unvoiced frames

c. Total classification accuracy $Acc_{v/uv}$,

$$Acc_{uv/v} = \frac{\left(M_{uv} + M_v\right) - \left(N_{uverr} + N_{verr}\right)}{\left(M_{uv} + M_v\right)} \times 100\ (\%) \tag{2.29}$$

(ii) Gross error rate and pitch estimation accuracy

If the pitch estimation error p_{err} is defined as

$$P_{err} = p_{ref} - p_{est} \tag{2.30}$$

where p_{ref} is the true (reference) pitch-period and p_{est} is the estimated pitch-period, a gross pitch estimation error is defined as occurring when,

$$\left| P_{err} \right| > 0.01 * P_{ref} \tag{2.31}$$

The pitch estimation accuracy is given by Acc_{pe},

$$Acc_{pe} = \frac{M_v - N_{gerr}}{N_v} \times 100\ (\%) \tag{2.32}$$

where $N_{g\,err}$ is the number of gross errors

(iii) Fine pitch-period error

In contrast to the definition of gross error, a fine pitch-period error is defined as occurring when,

$$\left| P_{err} \right| \leq 0.01 * P_{ref} \tag{2.33}$$

Fine pitch-period errors can be characterised by two parameters:

a. Mean of fine pitch-period errors μ_{err},

This is a measure of bias in the pitch-period measurement during voiced intervals and is defined as,

$$\mu_{err} = \frac{1}{N_v} \sum_{i=1}^{N_v} P_{err_i} \tag{2.34}$$

where N_v is the total number of voiced frames being correctly marked

b. Standard deviation of the fine pitch-period error υ_{err},

This is a measure of the accuracy of the pitch detector during voiced interval and is defined as [25],

$$\upsilon_{err} = \sqrt{\frac{1}{N_v} \sum_{i=1}^{N_v} P_{err_i}^2 - \mu_{err}^2} \tag{2.35}$$

2.6.3 Performance of the speech classifier

While the reference pitch-period file was being semi-manually created, many ambiguities were encountered while classifying the pitch-periods at voicing transitions. The reference pitch-periods were in some cases questionable or rather arbitrary. Two sets of tests were therefore conducted in order to have a fair evaluation. In the first test, all the voicing transition frames were excluded. These transition frames were then classified as voiced frames in the second test. The tested

speech file contains, 849 voiced frames, 817 unvoiced frames and 134 transition frames.

2.6.3.1 Results of the speech classifier using the clean speech file

The results obtained under clean speech condition are listed in table 2.1.

	Acc_{uv} (%)	Acc_{v} (%)
Excluding voicing transitions	95.96	100
Including voicing transitions	95.96	98.88

Table 2.1 Results of the speech classifier using the clean speech file.

Results in table 2.1 show that the speech classifier performed better in classifying voiced speech than unvoiced speech. When transition frames were excluded, a 100% accuracy was obtained in classifying the voiced frames. The accuracy in classifying the unvoiced frames was about 96%. This is because the speech classifier was designed such that it is biased towards a voiced decision in conditions of uncertainty. It was found that a voiced to unvoiced error may affect the speech quality of a speech coder perceptually more severely than vice-versa. With transition frames, the voiced classification accuracy dropped slightly, from 100% to 98.8%. Note that there was no change in the unvoiced case since all the reference transition frames were classified as voiced frames in the test.

2.6.3.2 Results of the speech classifier using the noisy speech files

The speech classifier was tested with speech received under various noise backgrounds. The unvoiced classification accuracy obtained at different SNR levels is shown graphically in figure 2.14 for the four types of noise. The results of the voiced classification accuracy are presented in figures 2.15a and b, corresponding to the cases without and with the transition frames.

Referring to figure 2.14, it is interesting to notice that the ability to accurately classify the unvoiced frames were enhanced by a reduction in the SNR level in the case of white noise and multi-speaker noise. It is shown in figure 2.13a that the energy of the white noise is evenly spread over the frequency spectrum. Therefore a reduce in the SNR level results in an enhancement of the effect of the PR in classifying unvoiced speech. Similar to the white noise, the multi-speaker noise has

a relatively flat frequency spectrum (figure 2.13d). Hence the same effect as for the white noise is seen. For the other two noise types, the unvoiced classification accuracy deteriorated as the SNR level reduced. The speech classifier performed the poorest in the case of the car noise, where the accuracy dropped to about 53% in a 0dB SNR level, whilst a 69% accuracy was still maintained for the babble noise under the same SNR level.

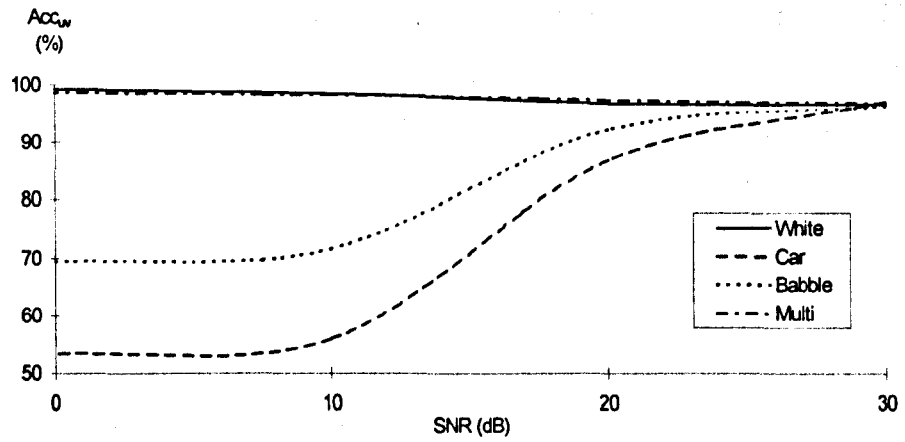


Figure 2.14 Performance of the speech classifier in classifying unvoiced speech using the noisy speech files.

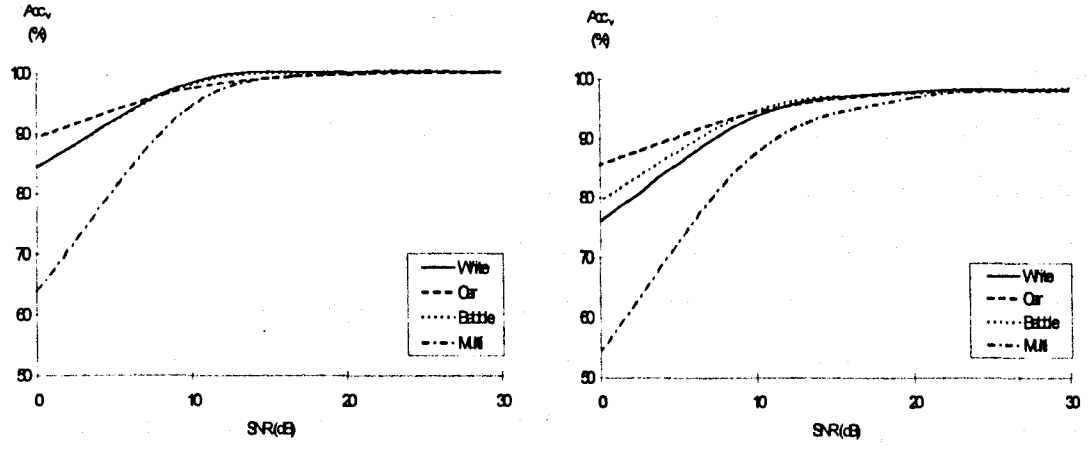


Figure 2.15 Performance of the speech classifier in classifying voiced speech using the noisy speech files. (a) excluding the transition frames (b) including the transition frames

Results in figure 2.15a show that the voiced classification accuracy dropped for all the four noise types, as the SNR level was reduced. Rather difference results were obtained for the unvoiced speech for the white and multi-speaker noise. Since it was mentioned previously that the effect of reducing the SNR level for the white noise and multi-speaker noise is to enhance the effect of the PR in classifying unvoiced speech, the voiced classification accuracy for both these noise types was

much lower than for the other two. Over 80% accuracy was obtained even at a 0dB SNR level, under the white, car and babble noise. In case of the multi-speaker noise, about 64% accuracy was obtained.

Comparing the results in figures 2.15a and b, it was found that the speech classifier performed poorly in voicing transitions. Figure 2.15b indicates that the classification accuracy dropped vastly for all the four noise types. When the transition frames were introduced into the test, an about 5% accuracy drop was observed, in the white, car and babble noise, at a 0dB SNR level. In case of the multi-speaker noise, the classification accuracy dropped by about 10% at a 0dB SNR level.

2.6.4 Performance of the pitch estimator

This section describes how the performance of the backward-mode cross-correlation pitch estimator was assessed, without the speech classifier. The assessment was carried out by first examining the pitch estimation accuracy for clean speech, both with and without the voicing transitions. Afterwards, the pitch estimator were tested using the noisy speech files. The reference pitch-period file was referred to throughout the experiment. From the reference pitch-period file, if a pitch-period is assigned to every sample within the entire speech frame, the speech frame was classified as a voiced frame. The minimum and maximum reference pitch-periods were fetched from the reference file. If the estimated pitch-period fell into this pitch-period range, a correct estimation was declared. Otherwise the estimated pitch-period was examined according to the criteria set in sections 2.6.2 (ii) and (iii). When a pitch-period of "0" or "10" and some pitch-periods existed in the same speech frame simultaneously, the speech frame would be classified as a voicing transition. The pitch-period range during the voiced section would also be found. Once again, this was compared with the estimated pitch-period.

2.6.4.1 Results of the pitch estimator using the clean speech file

The results of the pitch estimator under clean speech are listed in table 2.2.

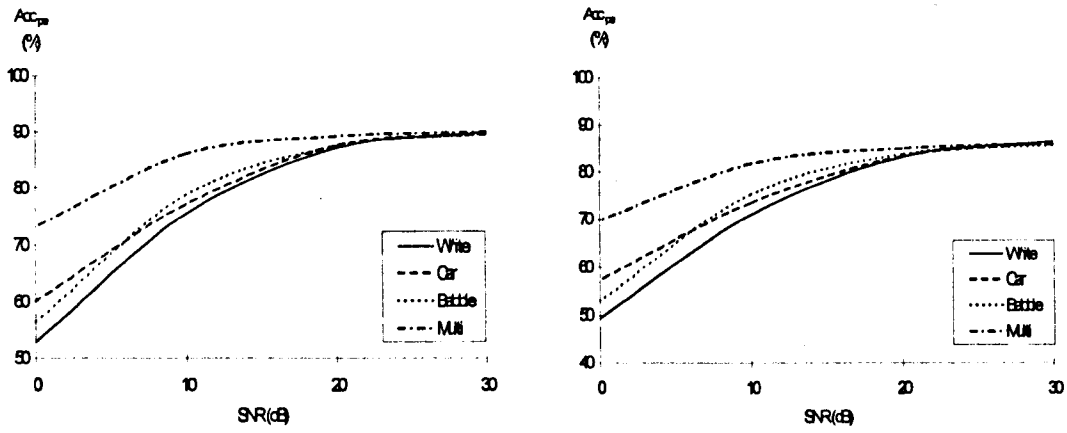
	Acc_{pe} (%)	μ_{err} (samples)	ν_{err} (samples)
Excluding voicing transitions	89.87	0.191	0.974
Including voicing transitions	85.96	0.124	1.083

Table 2.2 Results of the pitch estimator using the clean speech file.

From table 2.2, it can be seen that the gross error rate was about 10% disregarding the voicing transitions. It was found that more than half of the gross errors were multiple-pitch errors and that many of the remaining errors were found during voicing offset frames. This was due to the rapid pitch-period changes that tend to occur during voicing offset. It could be argued that the voicing transitions have already been discarded by excluding the transition frames in the test. However, many of the voicing offset frames may last for a few frames before they actually reach a silence or unvoiced frame. The pitch-periods in these ranges are very difficult to determine and may be arbitrary sometimes. It was also found that the mean pitch-period error was about 0.2 samples, corresponding to about 25μs. The standard deviation was less than one speech sample, i.e. less than 125μs. When transition frames were introduced, 48 more gross errors were introduced, out of the 143 transition frames. This corresponds to about 36% of the entire transition set. The mean pitch-period error was about the same whilst the standard deviation was increased by 0.1 sample. It may be concluded that the pitch estimator performed moderately well for clean speech without any post-processing element. In addition, the pitch estimator is able to estimate the pitch-period during voicing transitions.

2.6.4.2 Results of the pitch estimator using the noisy speech files

In table 2.3, the mean and standard deviation of the fine pitch-period errors under various noise backgrounds are tabulated. Furthermore, the pitch estimation accuracy is shown graphically in figures 2.16a and b.



(a) (b)

Figure 2.16 Performance of the pitch estimator using the noisy speech files.
(a) excluding the transition frames (b) including the transition frames

Noise type	Excluding Transitions		Including Transitions	
	μ_{err} (samples)	ν_{err} (samples)	μ_{err} (samples)	ν_{err} (samples)
wn30dB	0.148	0.990	0.084	1.074
wn20dB	0.112	1.050	0.048	1.111
wn10dB	0.142	1.100	0.105	1.102
wn0dB	0.064	1.188	-0.023	1.310
cn30dB	0.183	1.009	0.121	1.095
cn20dB	0.174	1.010	0.143	1.018
cn10dB	0.165	1.052	0.122	1.052
cn0dB	0.194	1.283	0.119	1.390
bn30dB	0.191	1.013	0.131	1.114
bn20dB	0.212	1.063	0.158	1.110
bn10dB	0.222	1.106	0.187	1.076
bn0dB	0.232	1.289	0.202	1.248
mn30dB	0.170	1.038	0.113	1.116
mn20dB	0.198	0.979	0.143	1.066
mn10dB	0.245	1.085	0.182	1.167
mn0dB	0.250	1.247	0.175	1.272

Table 2.3 Means and standard deviations of the fine pitch-period errors of the pitch estimator using the noisy speech files.

Results in table 2.3 (excluding transitions) and figure 2.16a show that the performance of the pitch estimator deteriorated rapidly when the SNR level dropped below 30dB. At a 30dB SNR level, the performance of the pitch estimator was maintained for all the four noise types, although a slight increase in the standard deviation was observed comparing with the results in table 2.2. The pitch estimator only managed to achieve just over 50% accuracy when the SNR level was reduced to

0dB. The pitch estimation performance degraded even more rapidly for transition frames than for voiced frames when the SNR level was reduced. From table 2.3 (including transitions) and figure 2.16b, it may be seen that the estimated pitch-period for more than 60% of the entire voicing transition set was incorrectly estimated with a 0dB SNR level, for all the four noise types.

2.6.5 Performance of the post-processing unit

To enhance the performance of the pitch estimator, the pitch-period refinement unit was introduced following the pitch estimator. The combined unit was tested with clean speech, the results obtained being presented in table 2.4. A 3-point median smoother was imposed to form a completed pitch post-processing unit. The unit was tested for clean speech as well as speech received in various noise backgrounds. The experimental results are shown in tables 2.5, 2.6 and figures 2.15a and b.

2.6.5.1 Results of the pitch estimator with the pitch-period refinement unit using the clean speech file

	Acc_{pe} (%)	μ_{err} (samples)	ν_{err} (samples)
Excluding voicing transitions	95.52	0.205	1.015
Including voicing transitions	92.37	0.137	1.098

Table 2.4 Results of the pitch estimator with the pitch-period refinement unit using the clean speech file.

Results in table 2.4 in comparison to those in table 2.2 suggest that the number of gross errors have been reduced by about 55% disregarding transition frames and by about 45% when transition frames are included. Most of the remaining gross errors were found in voicing onset frames, where, generally, less than a complete pitch-cycle was contained in the analysis speech frame. These errors occur because the pitch-period refinement unit implements a forward-mode cross-correlation function. It was also shown that the pitch-period refinement unit did not enhance the performance of the pitch estimator at voicing transitions. Only 15 gross errors occurring in the transition set were corrected by the pitch-period refinement

UNIVERSITY

unit. Finally, both the mean and standard deviation of the fine pitch-period error were increased slightly, as compared to table 2.2.

2.6.5.2 Results of the pitch estimator with the pitch post-processing unit using the clean speech file

	Acc_{pe} (%)	μ_{err} (samples)	ν_{err} (samples)
Excluding voicing transitions	93.99	0.292	1.185
Including voicing transitions	91.15	0.293	1.160

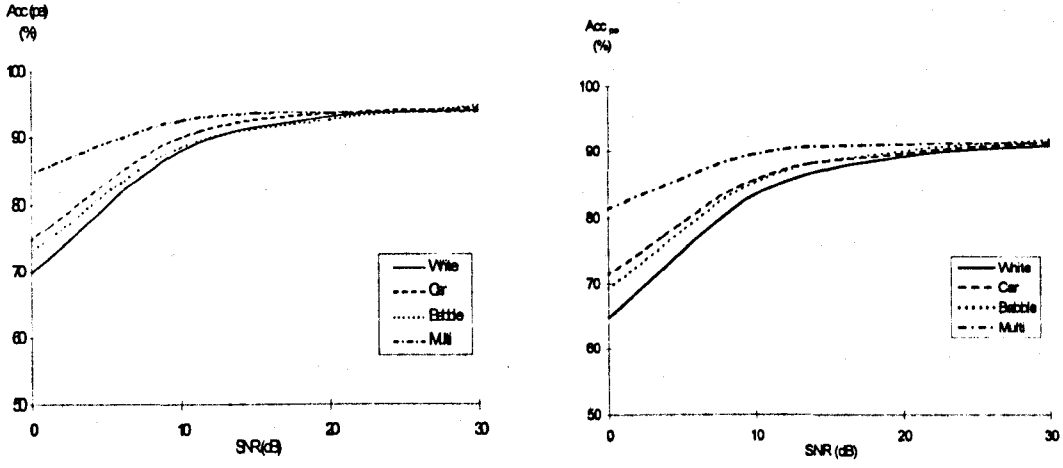
Table 2.5 Results of the pitch estimator with the pitch post-processing unit using the clean speech file.

Comparing the results in table 2.4 with the results in table 2.5 indicate that the 3-point median smoother seemed to generate more errors in the pitch detector instead of improving its performance. Thirteen extra gross errors were introduced when excluding the transition frames. With the transition frames, 12 extra gross errors were made. An increase in the mean and standard deviation of fine pitch-period error was also observed. The increase was due to the estimated pitch-periods in voicing transitions. During voicing onset frames, a pitch estimation error would result if there were not enough pitch structure. This would propagate to the future estimation through the use of the 3-point median smoother until a strong enough voiced speech arrives. During voicing offset frames, a rapid pitch-period change may happen while the signal is generally very weak. The median smoother tends to smooth out the pitch-period change in such regions.

In order to further evaluate the effect of the median smoother, a PWI coder was deployed [81]. Two pitch detectors were used in the PWI coder. The first contained no median smoother and the second incorporated the full pitch post-processing unit. The same speech file [21] was processed. The results suggested that the speech generated by the PWI coder with the median smoother was substantially better than the one without it. Many "wobble" sounds were heard in the output speech from the first version. The output speech obtained from the second version of PWI coder contained no such "wobbles". It appears that, although the 3-point median smoother introduced more pitch detection errors than it removed, it still

managed to improve the subjective quality of the output speech by ensuring a smoother pitch trajectory overall.

2.6.5.3 Results of the pitch estimator with the pitch post-processing unit using the noisy speech files



(a) (b)

Figure 2.17 Performance of the pitch estimator with the pitch post-processing unit using the noisy speech files. (a) excluding the transition frames (b) including the transition frames

Noise type	Excluding Transitions		Including Transitions	
	μ_{err} (samples)	ν_{err} (samples)	μ_{err} (samples)	ν_{err} (samples)
wn30dB	0.259	1.237	0.206	1.204
wn20dB	0.296	1.420	0.216	1.430
wn10dB	0.385	1.690	0.311	1.676
wn0dB	0.630	2.030	0.538	2.031
cn30dB	0.296	1.206	0.245	1.177
cn20dB	0.340	1.314	0.295	1.274
cn10dB	0.356	1.682	0.298	1.643
cn0dB	0.454	1.934	0.397	1.861
bn30dB	0.300	1.241	0.254	1.198
bn20dB	0.369	1.429	0.301	1.393
bn10dB	0.404	1.597	0.322	1.578
bn0dB	0.543	1.948	0.445	1.940
mn30dB	0.287	1.252	0.235	1.207
mn20dB	0.322	1.298	0.267	1.261
mn10dB	0.422	1.417	0.336	1.414
mn0dB	0.434	1.613	0.365	1.571

Table 2.6 Means and standard deviations of the fine pitch-period errors of the pitch estimator with the pitch post-processing unit using the noisy speech files.

Results in figure 2.17 suggest that the robustness of the pitch estimator for noisy speech was greatly increased by the pitch post-processing unit. Comparing the results in figures 2.17 and 2.16, more than 40% of the gross errors were corrected by the pitch post-processing unit and about 90% pitch estimation accuracy was maintained when the SNR level was reduced to 20dB for all the four noise types. The pitch estimation accuracy dropped rapidly as the SNR level was reduced below 20dB. Only about 35% of the gross errors in figure 2.16 were corrected for a 0dB SNR level, in all the four noise types. This was because the first pitch-period estimation provided by the pitch estimator was already wrong (for example a sub-multiple pitch error). The pitch-period refinement unit was not capable of rectifying these errors and the estimation error propagated to the following speech frames.

Comparing figures 2.17a and b, the pitch post-processing unit seemed to have very little effect on the transition frames. The pitch estimation accuracy dropped by about 5% for the transition frames. Results in figure 2.17 also suggest that the pitch estimator was more successful with the multi-speaker noise than with the other types of noise. The performance for the car noise and babble noise were very close. The white noise seemed to have the worst effect on the pitch estimator.

2.6.6 Performance of the two-way pitch detector

To assess the performance of the TPD, the clean speech file [21] and the speech received under the four noise types [90]-[93] were used again. The results for clean speech conditions are reported as,

$$Acc_{v/uv} = 97.56\%$$
$$Acc_{pe} = 90.53\%$$
$$\mu_{err} = 0.332 \quad \text{(sample)}$$
$$\nu_{err} = 1.406 \quad \text{(sample)}$$

2.6.6.1 Speech classification accuracy of the TPD using the noisy speech files

The speech classification accuracy of the TPD for the speech files synthesised with the four noise backgrounds are presented in figures 2.18.

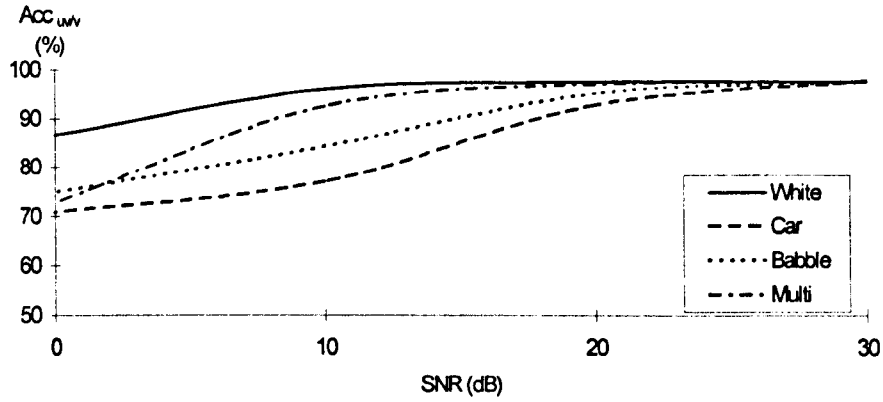


Figure 2.18 Speech classification accuracy of the TPD using the noisy speech files.

Considering the speech classification, the results in figure 2.18 suggest that the TPD performed well in all noisy backgrounds when the SNR level was above 20dB. In all the cases, over 90% accuracy was still maintained. However its performance dropped considerably when the SNR was reduced below 20dB. The pitch detector performed better for the white noise than for the other three types of noise. This was due to the enhancement in the effect of the PR in classifying unvoiced speech. The multi-speaker noise which also has a fairly flat frequency spectrum, was found to be the least damaging of the remaining three noise types for the pitch detector. The car noise had the worst effect on the pitch detector. This is very much due to the detector's poor ability to detect unvoiced speech in such conditions (figure 2.15b). In case of babble noise, the pitch detector performed moderately.

2.6.6.2 Pitch estimation accuracy of the TPD using the noisy speech files

The pitch estimation accuracy of the TPD for the speech files synthesised with the four noise backgrounds are presented in figure 2.19. The mean and the standard deviation measures of the fine pitch-period error are tabulated in table 2.7.

In case of the pitch estimation accuracy, the TPD performed fairly consistently for all the four types of noise, when the SNR level was above 20dB. When the SNR level was reduced below 20dB, the performance of the TPD deteriorated, in the cases of white, car and babble noise. However, the TPD seemed to continue to perform well for the multi-speaker noise. Less than a 5% reduction in pitch estimation accuracy was found, when the SNR level was reduced from 30dB to 0dB. This is because the speech classifier performed poorly for the multi-speaker

noise. Only the voiced speech which exhibited a highly periodic structure could be correctly recognised as a voiced frame and thus its pitch-period could be correctly estimated. Finally, results in table 2.8 suggest that both the mean and standard deviation of the fine pitch-period errors increase when the SNR level was reduced. An exceptional case is seen once again in the multi-speaker noise, in which the mean and standard deviation of the fine pitch-period errors decreased when the SNR level was reduced.

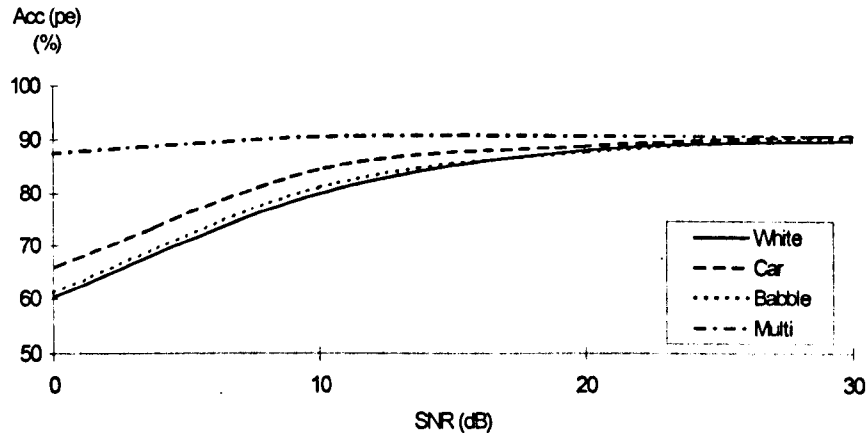


Figure 2.19 Pitch estimation accuracy of the TPD using the noisy speech files.

	SNR Level							
	30 (dB)		20 (dB)		10 (dB)		0 (dB)	
	μ_{err}	ν_{err}	μ_{err}	ν_{err}	μ_{err}	ν_{err}	μ_{err}	ν_{err}
White	0.340	1.338	0.368	1.486	0.455	1.555	0.535	1.598
Car	0.339	1.417	0.432	1.475	0.429	1.573	0.375	1.641
Babble	0.336	1.432	0.301	1.443	0.480	1.756	0.639	1.883
Multi-speaker	0.338	1.412	0.319	1.366	0.333	1.263	0.173	0.813

Table 2.7 Means and standard deviations (samples) of fine pitch-period errors of the TPD using the noisy speech files.

2.7 Conclusions

A two-way pitch detector (TPD) has been proposed and evaluated. The TPD consists of a pitch pre-processing unit, a speech classifier, a pitch estimator and a pitch post-processing unit. In the pitch pre-processing unit, input speech is band-limited to the range from 100Hz to 1kHz, using two 2nd-order IIR filters. The band-limited speech signal is further processed by a 10th order LP analysis filter to yield a band-limited speech residual signal. The band-limited speech residual is used to compute a backward mode cross-correlation function. The speech classifier uses four features of the current speech frame and the speech classification of the previous speech frame to compute the voice confidence level of a speech frame. The features are the scaled rms speech power, the scaled band-limited rms speech power, the ratio between the two powers and the maximum correlation function in the cross-correlation function. Speech classification is carried out based on the power ratio and the voice confidence level. If unvoiced speech is indicated, the pitch estimator sets the pitch-period to zero. In the case of a voiced frame, the delay associated with the global maximum in the cross-correlation function is used by the pitch estimator as the estimated pitch-period. The estimated pitch-period is further processed by the pitch-period refinement unit in the post-processing unit, in order to eliminate multiple-pitch errors which may occur. The result from the pitch-period refinement unit is further verified conditionally, by a 3-point median smoother. The TPD was tested for clean speech and for speech received with four types of noise. Experimental results suggest that the pitch detector works well in clean speech and for all the noisy speech examples when the SNR level was greater than 20dB. The performance of the pitch detector deteriorated when the SNR level dropped below 20dB.

Chapter 3

Liner Prediction: Analysis and Filtering

3.1 Introduction

Conventional waveform coders (such as PCM, DM, ADPCM) require a high bit-rate to transmit the information needed to maintain the perceptual quality of the decoded speech. To reduce the bit-rate while preserving perceptual quality, a parametric approach is required. Linear prediction provides a powerful tool for estimating the parameters which are assumed to control the human speech production mechanism. In linear prediction, a human speech production model is assumed which is separated into two components: a) an all-pole vocal tract transfer function H(z) which models the composite effects of the glottal excitation (for voiced speech), the vocal tract and lip-radiation, b) a speech excitation signal u(n). In voiced speech, the excitation signal u(n) is assumed to be a pseudo-periodic impulse train with each impulse separated by the current value of the pitch-period. In unvoiced speech, the excitation signal is assumed to be white noise which has a Gaussian distribution. The excitation signal conveys information which is strongly related to speech naturalness. A schematic diagram of the model is shown in figure 3.1 [40]. A general account of linear prediction can be found in the literature [1][2][6][40][41].

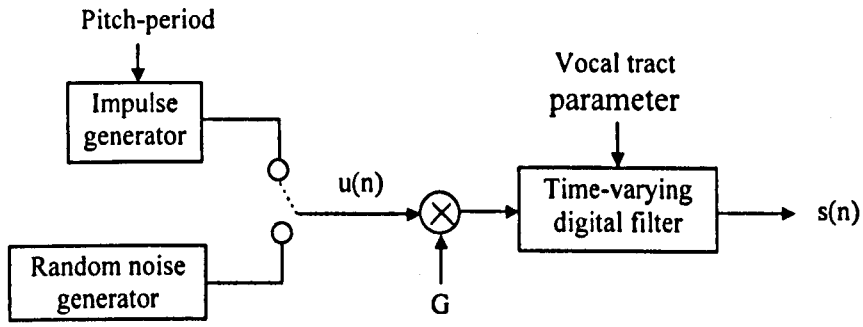


Figure 3.1 Schematic diagram of an all-pole model for speech production.

The assumed vocal tract transfer function is defined by the all-pole expression,

$$H(z) = \frac{G}{1 + \sum_{i=1}^{P} a_i z^{-i}} \tag{3.1}$$

where

G is the gain parameter,

P is the order of the transfer function,

a_i are the coefficients of the denominator polynomial,

The all-zero transfer function A(z) is defined as,

$$A(z) = 1 + \sum_{i=1}^{P} a_i z^{-i} \tag{3.2}$$

Therefore,

$$H(z) = \frac{G}{A(z)} \tag{3.3}$$

Equation 3.1 can be represented in the time-domain by a difference equation,

$$s(n) = G u(n) - \sum_{i=1}^{P} a_i s(n-i) \tag{3.4}$$

$$-\infty < n < \infty$$

Equation 3.4 suggests that a speech sample s(n) can be predicted to some extent by a weighted sum of P previous samples of s(n). The weighting coefficients a_i correspond to the coefficients of the denominator of the all-pole transfer function H(z). The optimum coefficients for a given segment of N speech samples $\{s(n)\}_{n=0, N-1}$ are determined by minimising the sum of the squared differences between the samples s(n) of the segment and the prediction to s(n): $\hat{s}(n) = -\sum_{i=1}^{P} a_i s(n-i)$ over a suitable range of values of n. Let e(n) be the prediction error defined by,

$$e(n) = s(n) - \hat{s}(n) \tag{3.5}$$

$$-\infty \le n < \infty$$

To determine the optimum coefficients a_i, for i = 1, 2, ..., P, for a given speech frame, the total squared error E is minimised over a range of M, where,

$$E = \sum_{n=0}^{M-1} e^2(n)$$
$$= \sum_{n=0}^{M-1} \left(s(n) + \sum_{i=1}^{P} a_i\, s(n-i) \right)^2 \tag{3.6}$$

The value of M will be discussed later.

This is done by partially differentiating the total squared error with respect to each filter coefficient and setting the result in each case to zero, i.e. setting $\frac{\partial E}{\partial a_i}=0$ for i = 1, 2, ..., P. A set of P simultaneous equations is thus obtained for the a_i coefficients,

$$-\sum_{n=0}^{M-1} s(n)s(n-i) = \sum_{l=1}^{P} a_l \sum_{n=0}^{M-1} s(n-i)s(n-l) \tag{3.7}$$

For a Pth order all-pole model, these equations may be expressed as follows,

$$-\phi(i,0) = \sum_{l=1}^{P} a_l \phi(i,l) \tag{3.8}$$
$$i = 1, 2, \ldots, P$$

with,

$$\phi(i,l) = \sum_{n=0}^{M-1} s(n-i)s(n-l) \tag{3.9}$$

Equation 3.8 may be expressed in matrix form as,

$$\Phi \mathbf{a} = \Psi \tag{3.10a}$$

with

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \cdots & \phi(1,P-1) & \phi(1,P) \\ \phi(2,1) & \phi(2,2) & \cdots & \phi(2,P-1) & \phi(2,P) \\ \phi(3,1) & \phi(3,2) & \cdots & \phi(3,P-1) & \phi(3,P) \\ \vdots & \vdots & & \vdots & \vdots \\ \phi(P,1) & \phi(P,2) & \cdots & \phi(P,P-1) & \phi(P,P) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{bmatrix} = - \begin{bmatrix} \phi(1,0) \\ \phi(2,0) \\ \phi(3,0) \\ \vdots \\ \phi(P,0) \end{bmatrix} \tag{3.10b}$$

The optimum filter coefficients can be obtained by solving the set of P linear equations.

There are two commonly used LP analysis methods, namely the autocorrelation method and the covariance method. The two methods differ in the way they define the range of summation M in equation 3.6 for a given speech frame of N samples [2].

For the covariance method, M is made equal to N and samples of s(n) for $n<0$ are taken to be the appropriate samples of the previous frame. For the autocorrelation method, M is made equal to N+P and samples of s(n) are taken to be zero for $n<0$ and $n>N-1$. The two methods will produce slightly different all-pole models [42][43] for the given speech frame.

In the autocorrelation method, a tapered window is generally used to gradually diminish the speech samples as the frame boundaries are approached. The set of linear equations can be solved efficiently using Durbin's algorithm [2]. The autocorrelation method predicts an all-pole model which is, in principle, stability guaranteed. The estimated spectrum includes the effect of the analysis window.

Windowing is not necessary in the covariance method. Instead, speech samples in the previous speech frame are included during the analysis. The covariance method has the potential for more accurate performance than the autocorrelation method [43]. However the stability of the resulting all-pole model is not guaranteed. The all-pole model tends to become less stable as the number of speech samples in the analysis is reduced [43].

It was suggested [44] that the autocorrelation method can be used to produce approximately the same result as the covariance method by extracting a single speech cycle from the voiced speech and circularly repeating the periodic cycle. Using this approach, the advantages of the two methods can be obtained.

In the autocorrelation method and the covariance method of LP analysis, the a_i coefficients are estimated via two steps [2]: a) computation of the correlation matrix and b) solving the set of P linear equations. A class of LP analysis techniques which combine the two procedures is also available. These techniques are known as lattice LP analysis techniques [2]. The lattice methods have the advantage over the autocorrelation method that they require no tapered window to modify the speech segment being analysed [45]. Unlike the covariance method, the stability of the all-pole transfer function computed by the lattice methods can be easily preserved [45]. The disadvantage in using lattice methods over the autocorrelation method and the covariance method is the increase in the computational complexity [45]. Burg's [45] method is a particularly effective lattice method which is investigated later in the chapter.

In linear prediction, after the set of a_i coefficients has been computed for a given frame of speech, the corresponding frame of speech excitation can be obtained by passing the frame of speech through an LP analysis filter, which has the transfer function defined by the all-zero model A(z). This is effectively to flatten the frequency spectrum of the frame of speech signal in the frequency-domain. Conversely, the short-term frequency spectrum can be re-imposed onto the frame of speech excitation signal using an LP synthesis filter, which has a transfer function defined by the all-pole model H(z). The simplest form of LP analysis and synthesis filters uses ladder structures, shown in figures 3.2a and b respectively, where the a_i coefficients are directly used as the ladder filter coefficients.

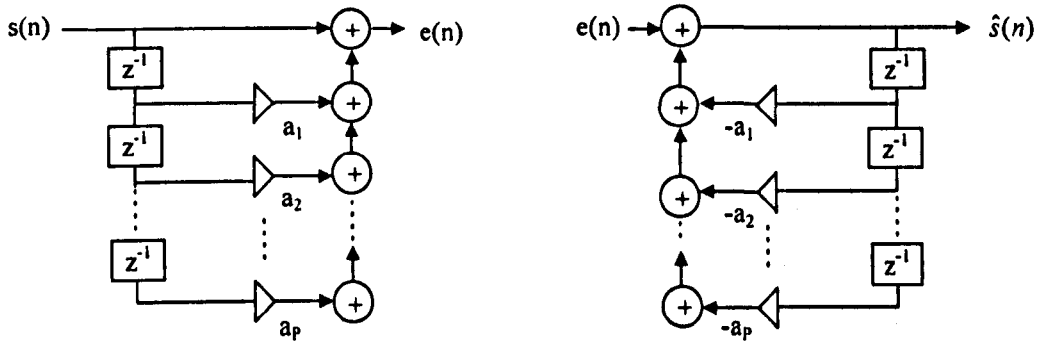


Figure 3.2 The schematic diagram of an LP analysis/synthesis filter pair.

Owing to the poor performance of a_i coefficients under quantisation [55] and interpolation [49], they are usually transformed to other representations for quantisation and interpolation. Alternative representations of a_i coefficients include PARCOR coefficients, log-area ratios and line spectral frequencies [55][56]. It is reasonable to believe that the computational efficiency of the LP analysis and synthesis filter may be increased by using filter structures appropriate to the chosen parametric representation, thus avoiding the need to convert between the new representation and the a_i coefficients.

Line spectral frequencies (LSF's) have been widely used in the past decade as an alternative representation of a_i coefficients, due to their good performance in quantisation [56] and interpolation [49]. The LSF's can be directly used as parameters of analysis and synthesis filters [47][50], and thus the computational cost of converting between the LSF's and the a_i coefficients can be eliminated.

LP analysis techniques have been briefly discussed above. Various important decisions have to be made when LP analysis is implemented in practice. These include the choice of a suitable LP analysis method and the associated window function, the location of the window in the speech frame being analysed [42] and the structure of the LP analysis and synthesis filters. These aspects will be discussed later in this chapter. In sections 3.2 and 3.3, LP analysis using the autocorrelation method and Burg's method (which is a form of lattice LP analysis) will be introduced respectively. In section 3.4, an LSF analysis and synthesis filter pair which is used as an alternative to LP ladder analysis and synthesis filters will be presented. The two filter structures will be compared using both objective and subjective tests. In section 3.5, the accuracy of the autocorrelation method and Burg's method in LP analysis will be compared using a synthetic vowel. Both pitch-asynchronous and pitch-synchronous LP analysis will be conducted. The two methods i.e. autocorrelation method and Burg's method will also be compared when the size of the analysis window is smaller than a complete pitch-cycle. In section 3.6, the two methods will be examined using a speech file of clean natural speech [21].

3.2 LP analysis using the autocorrelation method

In the autocorrelation method of LP analysis, each speech frame is defined over a finite range, from n = 0 to N-1 say, the speech samples outside this range being assumed to be zero. Thus if the summation range of the error signal is defined to be n = -∞ to ∞, the summation will have non-zero values only for values of n between 0 to N+P-1. Therefore,

$$E = \sum_{n=0}^{N+P-1} \left(s_w(n) + \sum_{i=1}^{P} a_i s_w(n-i) \right)^2 \tag{3.11}$$

where s_w is the frame of speech signal multiplied by a window function.

Under the assumption that the speech samples outside the range n = 0 to N-1 are zero, equation 3.11 can be expressed as,

$$\phi(i,l) = \sum_{n=0}^{N+P-1} s_w(n-i)\, s_w(n-l) = \sum_{n=0}^{N-1-(i-l)} s_w(n)\, s_w(n+i-l) \tag{3.12}$$

where $\phi(i,l)$ in this case is an autocorrelation function for a delay (i-l), i.e.

$$\phi(i,l) = R(i-l) \tag{3.13}$$

As the Autocorrelation function R(i) is an even function of R, the set of P linear equations may be re-expressed as,

$$-R(i) = \sum_{l=1}^{P} a_l R\left(\left|i-l\right|\right) \qquad i = 1, 2, \ldots, P \tag{3.14}$$

with,

$$R(i) = \sum_{n=0}^{N-1-i} s_w(n)\, s_w(n+i) \tag{3.15}$$

In matrix form, equation 3.14 is expressed as,

$$\mathbf{Ra} = \Re \tag{3.16a}$$

with

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(P-2) & R(P-1) \\ R(1) & R(0) & \cdots & R(P-3) & R(P-2) \\ R(2) & R(1) & \cdots & R(P-4) & R(P-3) \\ \vdots & \vdots & & \vdots & \vdots \\ R(P-1) & R(P-2) & \cdots & R(1) & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{bmatrix} = - \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(P) \end{bmatrix} \tag{3.16b}$$

The matrix **R** is a Toeplitz matrix. The P linear equations can be solved by Durbin's recursive algorithm [1] which is described below.

First, the autocorrelation coefficients R(i) are computed for i = 1, 2, ..., P. Then the first filter coefficient a_1 is computed as:

$$k_1 = -\frac{R(1)}{E(0)} \tag{3.17a}$$

$$a_1 = k_1 \tag{3.17b}$$

where $E(0) = R(0)$. Index i is now set to 2 and a set of coefficients $a_j^{(i)}$ are computed for j = 1, 2, ..., i-1, using equations 3.17c to f.

$$k_i = -\frac{R(i) + a_1^{(i-1)} R(i-1) + \cdots + a_{i-1}^{(i-1)} R(1)}{E(i-1)} \tag{3.17c}$$

$$a_i^{(i)} = k_i \tag{3.17d}$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i \, a_{i-j}^{(i-1)} \qquad j = 1, 2, \ldots, i-1 \tag{3.17e}$$

$$E(i) = (1 - k_i^2) \, E(i-1) \tag{3.17f}$$

This procedure is repeated for i = 3, 4, ..., P. The ith filter coefficient a_i is taken as,

$$a_i = a_i^{(i)} \tag{3.17g}$$

The vocal tract transfer function can be modelled by an all-pole digital filter shown in figure 3.2b.

3.3 LP analysis using Burg's method

During the derivation of the a_i coefficients, for i = 1, 2, ..., P, using Durbin's recursive algorithm, a set of intermediate parameters, k_i for i = 1, 2, ... , P, are obtained. It is known [45] that stability of the all-pole transfer function H(z) obtained from Durbin's algorithm is guaranteed if,

$$-1 \le k_i \le 1 \qquad i = 1, 2, \cdots, P \tag{3.18}$$

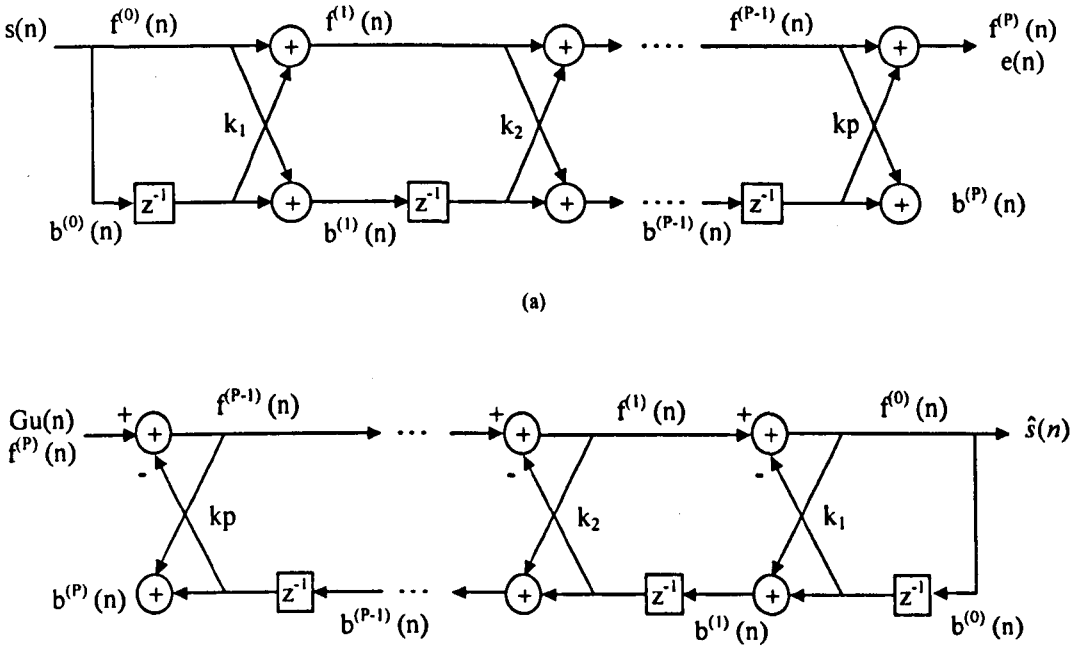


Figure 3.3 The schematic diagram of a lattice filter pair.
(a) the analysis filter (b) the synthesis filter

The all-pole transfer function H(z) obtained from Durbin's algorithm can be implemented in the form of a recursive lattice filter, as shown in figure 3.3b, rather than a ladder structure, as shown in figure 3.2b. If the lattice coefficients k_i are made identical to the k_i coefficients obtained from Durbin's algorithm, the transfer function of the lattice filter will be identical to that of the ladder filter in figure 3.2 with the corresponding a_i coefficients obtained from Durbin algorithm. For any set of stable a_i coefficients, there is a corresponding set of stable k_i coefficients and vice versa. Hence a lattice filter can be devised with the same transfer function as a given ladder filter and vice versa. Given a recursive lattice filter as shown in figure 3.3b

realising H(z), a corresponding non-recursive (all-zero) lattice filter, as shown in figure 3.3a with the same k_i coefficients, realises A(z). The coefficients k_i of a lattice filter are often referred to as partial correlation (PARCOR) coefficients. The non-recursive lattice filter realising A(z) generates a series of prediction errors f(n) and b(n) defined by,

$$f^{(i)}(n) = f^{(i-1)}(n) + k_i\, b^{(i-1)}\left(n-1\right) \tag{3.19a}$$

$$b^{(i)}(n) = k_i\, f^{(i-1)}(n) + b^{(i-1)}\left(n-1\right) \tag{3.19b}$$

$$i = 1, 2, \ldots, P$$

where $f^{(i)}(n)$ is known as the forward prediction error and $b^{(i)}(n)$ is known as the backward prediction error at lattice stage i. When speech samples s(n) are applied as input to the non-recursive lattice filter, the two prediction errors $f^0(n)$ and $b^0(n)$ at the input to the first lattice stage satisfy,

$$f^{(0)}(n) = b^{(0)}(n) = s(n) \tag{3.20}$$

The output of the non-recursive lattice will be the LP residual e(n) which will be identical to what would be obtained from the corresponding non-recursive ladder filter. The Pth order non-recursive lattice output is the forward prediction error at the Pth stage, i.e.

$$e(n) = f^{(P)}(n) \tag{3.21}$$

The recursive lattice filter realising H(z) given PARCOR coefficients k_i for i = 1, 2, ..., P, computes the forward and backward errors at end stage from the input e(n) as follow,

$$f^{(i-1)}(n) = f^{(i)}(n) - k_i\, b^{(i-1)}\left(n-1\right) \tag{3.22a}$$

$$b^{(i)}(n) = k_i\, f^{(i-1)}(n) + b^{(i-1)}\left(n-1\right) \tag{3.22b}$$

$$i = 1, 2, \ldots, P$$

The synthesised speech signal $\hat{s}(n)$ is taken from the forward prediction error at the lattice output, i.e. lattice stage zero.

A way of computing k_i coefficients, for an LP lattice filter, which is different from the autocorrelation method, is achieved by computing the PARCOR coefficients individually at each lattice filter stage. The computation is done by considering the forward and/or backward errors at the particular stage, windowed appropriately for that stage. There are several variations of this approach. In one of the variations, the squared forward prediction error, $E^{(i)} = \sum_{n=0}^{N-1} f^{(i)}(n)^2$, at each stage i is minimised. This is carried out by setting $\frac{\partial E^{(i)}}{\partial k_i} = 0$ for each k_i with i = 1, 2, ..., P. Hence,

$$k_i = \frac{-\sum_{n=0}^{N-1} f^{(i-1)}(n)\, b^{(i-1)}(n-1)}{\sum_{n=0}^{N-1} b^{(i-1)}(n-1)^2} \qquad (3.23)$$

$$i = 1, 2, \cdots, P$$

The set of PARCOR coefficients generated by this equation will be different from those obtained for the autocorrelation method because a) the frame of speech samples is not modified by a tapered window b) the range of error signal is defined only from n = 0 to N-1.

Alternatively, a set of PARCOR coefficients may be obtained by minimising the squared backward prediction error, $E^{(i)} = \sum_{n=0}^{N-1} b^{(i)}(n)^2$, by setting $\frac{\partial E^{(i)}}{\partial k_i} = 0$ for each stage i with i = 1, 2, ..., P. In this case, we obtain,

$$k_i = \frac{-\sum_{n=0}^{N-1} f^{(i-1)}(n)\, b^{(i-1)}(n-1)}{\sum_{n=0}^{N-1} f^{(i-1)}(n)^2} \qquad (3.24)$$

$$i = 1, 2, \cdots, P$$

The k_i coefficients derived using equations 3.23 and 3.24 are not stability guaranteed [45]. In order to provide a stable set of filter coefficients, Burg's method may be used. In Burg's LP analysis, the sum of the squares of the forward and backward prediction errors at each stage is minimised [45]. Thus, the error at the ith stage is defined as,

$$E^{(i)} = \sum_{n=0}^{N-1} \left(f^{(i)}(n)^2 + b^{(i)}(n)^2 \right) \tag{3.25}$$

•

By differentiating the total error $E^{(i)}$ with respect to each k_i coefficient at the ith stage for i = 1, 2, ..., P, the set of PARCOR coefficients can be determined using equation 3.26,

$$k_i = \frac{-2 \sum_{n=0}^{N-1} f^{(i-1)}(n)\, b^{(i-1)}(n-1)}{\sum_{n=0}^{N-1} f^{(i-1)}(n)^2 + \sum_{n=0}^{N-1} b^{(i-1)}(n-1)^2} \qquad i = 1, 2, \ldots, P \tag{3.26}$$

The k_i coefficients derived from equation 3.26 is always bounded by ±1. This can be shown using the inequality that $\sum_{n=0}^{N-1} \left(f(n) + b(n) \right)^2 \geq 0$ and thus,

$$\sum_{n=0}^{N-1} f^2(n) + \sum_{n=0}^{N-1} b^2(n) \geq -2 \sum_{n=0}^{N-1} f(n) b(n).$$

As a result, the all-pole transfer function H(z) derived from Burg's method can be stability guaranteed.

To obtain the set of a_i coefficients corresponding to a given set of k_i coefficients, equations 3.17d, e and g are repeated to compute each a_i from k_i for i = 1, 2, ..., P.

3.4 Implementation of the vocal tract filter

LP analysis allows speech segments to be modelled in terms of a vocal tract transfer function and an excitation signal. The vocal tract transfer function characterises the short-term frequency spectral envelope of the speech, and can be implemented by an all-pole digital filter. The required excitation signal may be obtained by passing the input speech segment through an LP analysis filter realising A(z). The output is the LP residual which, in principle, is equal to the required excitation signal. If the LP analysis has been well carried out the residual should be spectrally flat.

A simple LP analysis filter in ladder form is shown in figure 3.2a. This is a FIR filter. The corresponding IIR ladder synthesis filter is shown in figure 3.2b. The ladder filter coefficients are taken as the a_i coefficients. Alternative implementations of the LP analysis and synthesis filters can be realised as the lattice structures, shown in figure 3.3a and b respectively. The filter coefficients in this case are the PARCOR coefficients. Lattice filters have been widely used in speech coders because of the quantisation characteristics of PARCOR coefficients. In addition, the stability of the synthesis filter can be easily preserved by ensuring that all the PARCOR's are bounded by ±1.

3.4.1 LP filtering using the LSF filters

Line spectral frequencies (LSF's) have been widely used as an alternative representation of a_i coefficients [56]. They are more directly related to the short-term speech spectrum [48] than a_i or k_i coefficients. LSF's may be determined from the a_i coefficients of a given inverse filter transfer function A(z), by decomposing A(z) into two all-zero transfer functions P(z) and Q(z) where,

$$P(z) = A(z) + z^{-(p+1)} A\left(z^{-1}\right) \tag{3.27a}$$

$$Q(z) = A(z) - z^{-(p+1)} A\left(z^{-1}\right) \tag{3.27b}$$

where P(z) and Q(z) are P+1 order polynomials and we labelled the coefficients for each polynomial P(z) and Q(z) as p_i and q_i respectively.

The zeros of P(z) and Q(z) are the required LSF's. They have the properties that H(z), which is equal to 1/A(z), is stable if and only if,

a) All the roots of P(z) and Q(z) lie on the unit circle,

b) The roots of P(z) and Q(z) are interlaced, i.e. if φ_i and θ_i are the roots for P(z) and Q(z) respectively for i = 1, 2, ..., P/2, then,

$$0 \le \varphi_1 < \theta_1 < \cdots < \varphi_{P/2} < \theta_{P/2} \le \pi$$

Given a set of P LSF coefficients, the transfer function A(z) may be derived from P(z) and Q(z) as follows,

$$A(z) = \frac{1}{2}\left[\, P(z) + Q(z) \,\right] \tag{3.28}$$

3.4.1.1 The LSF synthesis filter

Since the LSF's lie on the unit circle, P(z) and Q(z) can be factored into second order linear phase polynomials [46]. To simplify the analysis only even order LP analysis is considered, i.e. P is always even. In this case, we can express P(z) and Q(z) as,

$$P(z) = \left(1 + z^{-1}\right) \prod_{i=1}^{P/2} \left(1 - 2\cos\varphi_i\, z^{-1} + z^{-2}\right) \tag{3.29a}$$

$$Q(z) = \left(1 - z^{-1}\right) \prod_{i=1}^{P/2} \left(1 - 2\cos\theta_i\, z^{-1} + z^{-2}\right) \tag{3.29b}$$

The vocal tract transfer function is,

$$\begin{aligned} H(z) &= \frac{1}{A(z)} \\ &= \frac{1}{\frac{1}{2}\left[\, P(z) + Q(z) \,\right]} \\ &= \frac{1}{1 + \frac{1}{2}\left[\left(\, P(z) - 1 \,\right) + \left(\, Q(z) - 1 \,\right)\right]} \\ &= \frac{1}{1 + W(z)} \end{aligned} \tag{3.30}$$

where

$$W(z) = \tfrac{1}{2}\left[\left(P(z) - 1\right) + \left(Q(z) - 1\right)\right] \tag{3.31}$$

Equation 3.30 can be implemented by a recursive signal flow graph as shown in figure 3.4,

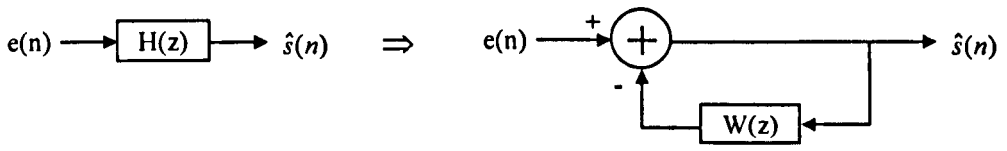


Figure 3.4 The signal flow diagram of an LSF synthesis filter.

Equation 3.31 suggests that W(z) is composed of two modules, $P(z) - 1$ and $Q(z) - 1$. The two modules can be expressed as [46],

$$P(z) - 1 = z^{-1}\left[\begin{array}{l}\left(c_1 + z^{-1}\right) + \sum_{i=1}^{P/2-1}\left(c_{i+1} + z^{-1}\right)\prod_{j=1}^{i}\left(1 + c_j z^{-1} + z^{-2}\right) \\ -\prod_{i=1}^{P/2}\left(1 + c_i z^{-1} + z^{-2}\right)\end{array}\right] \tag{3.32a}$$

$$Q(z) - 1 = z^{-1}\left[\begin{array}{l}\left(d_1 + z^{-1}\right) + \sum_{i=1}^{P/2-1}\left(d_{i+1} + z^{-1}\right)\prod_{j=1}^{i}\left(1 + d_j z^{-1} + z^{-2}\right) \\ +\prod_{i=1}^{P/2}\left(1 + d_i z^{-1} + z^{-2}\right)\end{array}\right] \tag{3.32b}$$

where $c_i = -2\cos\varphi_i$ and $d_i = -2\cos\theta_i$

From equations 3.32a and b, each module can be implemented by cascading a number of second order sections. W(z) can thus be realised by summing the output from each module and dividing by two. A filter structure may thus be derived whose multiplier values are LSF coefficients and whose transfer function is H(z). Such a filter may be referred to as an "LSF synthesis" filter. As an example of a 10th order LSF synthesis filter is shown in figure 3.5.

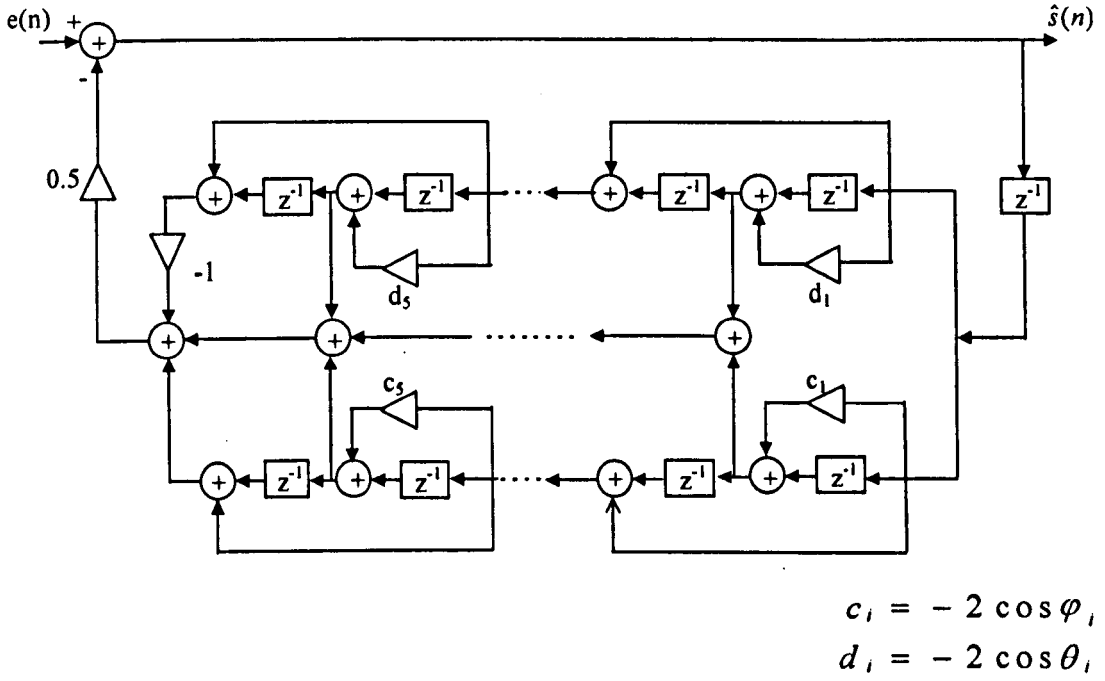


Figure 3.5 A 10th order LSF synthesis filter.

3.4.1.2 The LSF analysis filter

The simplest form of LSF analysis filter may be implemented directly from equation 3.28. Equation 3.28 suggests that an LSF analysis filter can be realised using two separate ladder filters which model the all-zero polynomials P(z) and Q(z) respectively as illustrated in figure 3.6a.

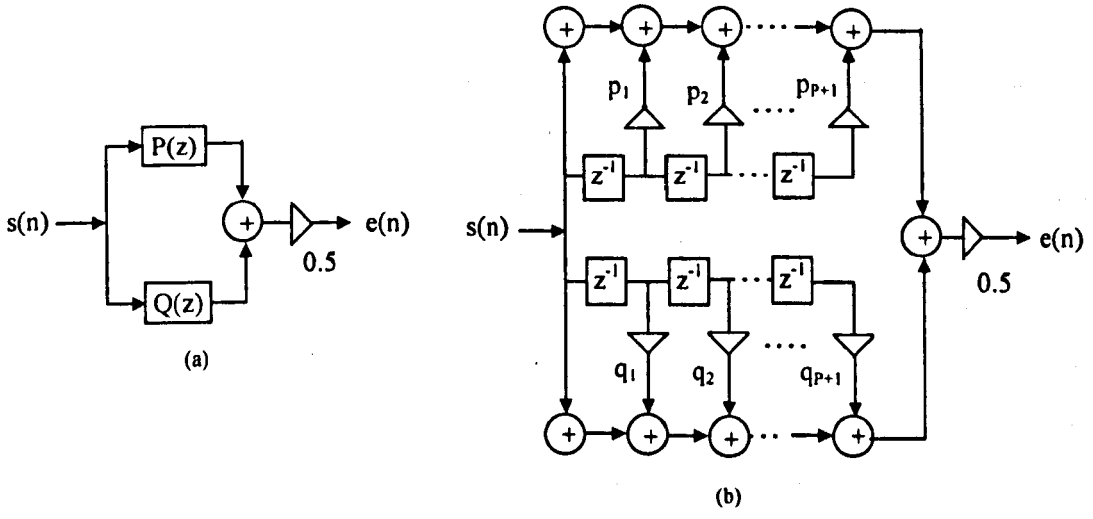


Figure 3.6 A simple LSF analysis filter.
(a) The structure of the filter (b) The signal flow diagram of the filter

The filter residual signal is taken as the sum of the outputs from the two ladder filters and dividing by two. The signal flow graph of such a filter structure is shown in figure 3.6b. The computation of the filter coefficients p_i and q_i from the a_i coefficients are presented in Appendix B.

The behaviour of an LSF analysis filter and its corresponding LSF synthesis filter were investigated using the speech file "OPERATOR.DAT" [21]. A 200 sample asymmetric window (which will be discussed section 3.4.2) was used to extract a 200 sample segment consisting of 160 samples from the current frame, the first 40 speech samples in the window being taken from the previous speech frame. A 10th order LP analysis was performed on the windowed speech segment using the autocorrelation method to yield a set of a_i coefficients. The set of a_i was then converted to LSF's (the conversion from a_i coefficients to LSF's is presented in Appendix B. After the set of LSF's was available, the frame of 160 speech samples was processed by the LSF analysis filter to obtain an LP residual. The LP residual was then used as the input to the LSF synthesis filter which would be used to reconstruct the frame of speech. Theoretically, the output from the LSF synthesis filter should be identical to the input to the LSF analysis filter. This was indeed the case in practice when the filters were in a time-invariant state, i.e. when the LSF's were constant over time. Problems arise, however, when the LSF's are made to vary over time, for example by updating the LSF's at 20ms intervals.

In figure 3.7a, a segment of speech used to test the LSF analysis and synthesis filters is presented. LP analysis was performed on the 160 sample speech segment (with 40 samples being taken from the previous frame) to yield the a_i coefficients. The a_i coefficients were converted to LSF's. The LSF's were used to update the LSF analysis and synthesis filters at exactly the same point in time. In figure 3.7b, the output from the LSF synthesis filter shows that a transient effect occurred at the beginning of the frame, i.e. just after the point when the LSF's were updated. At the beginning of the frame, the LSF synthesis filter output is rather different in shape from the LSF analysis filter input. The shape then starts to become more similar towards the middle of the frame and is very similar towards the end. A certain period was therefore required at the beginning of each frame for a transient

effect to die away and the expected waveshape to emerge. Experiments indicated that the perceptual quality of the output speech can be seriously affected by this transient effect at the beginning of each frame.

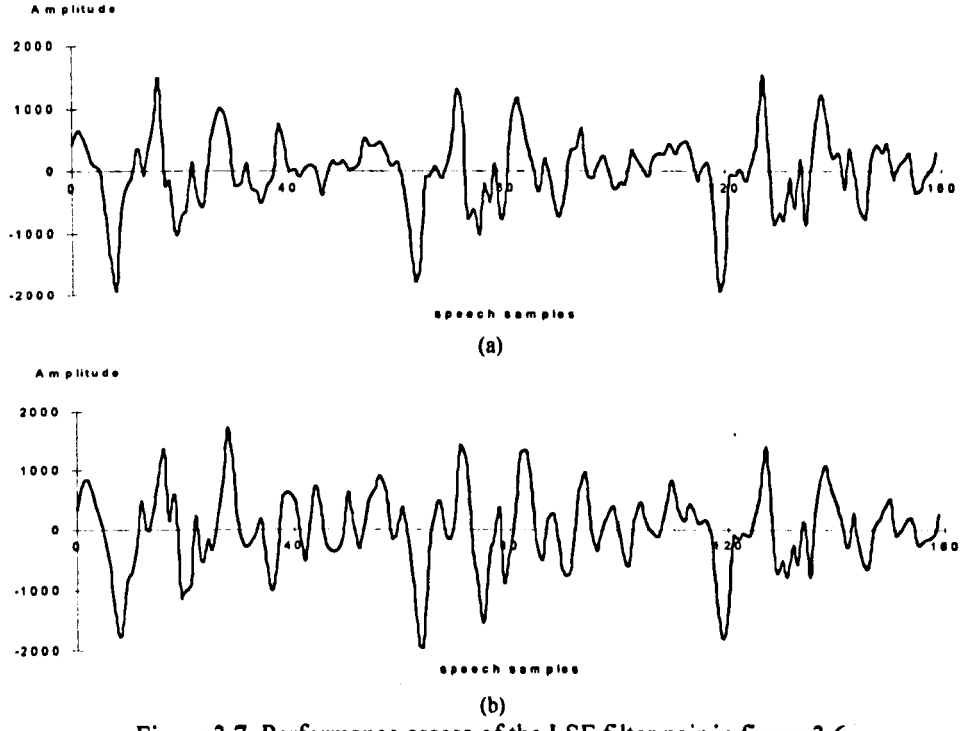
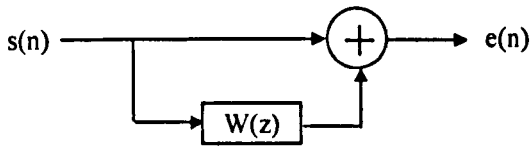


Figure 3.7 Performance assess of the LSF filter pair in figure 3.6.
(a) the original speech signal (b) the reconstructed speech signal by the LSF filter pair

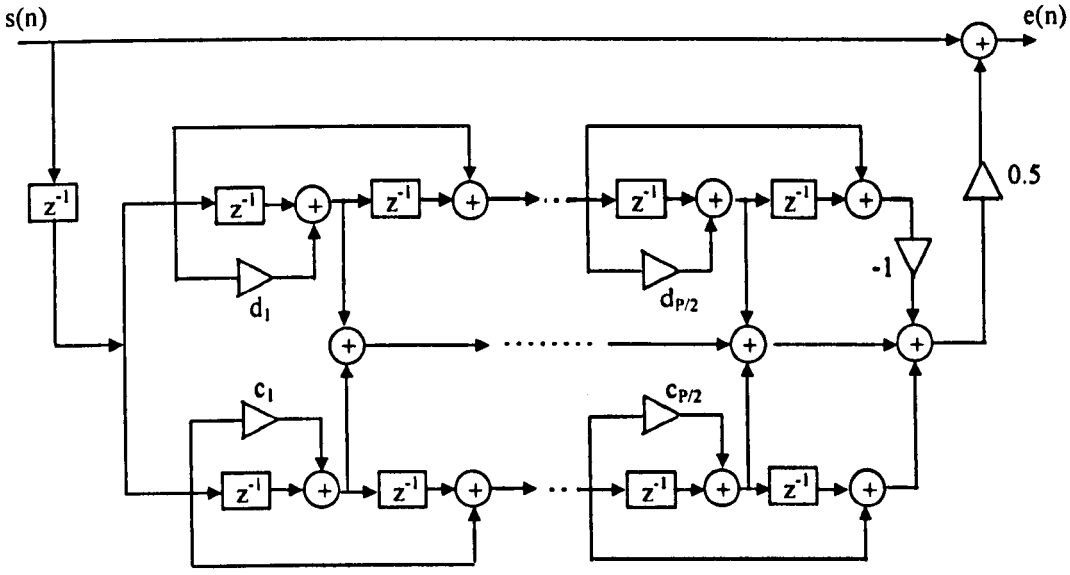
The cause of the transient effect was found to be due to differences in the memory contents in the analysis and synthesis filters, when the LSF's are being updated. The differences arise from the way the analysis filter is realised (figure 3.6) and the way the synthesis filter is realised (figure 3.5). Although under time-invariant conditions (fixed coefficients), these filters are exact inverses of each other, they are not exact inverses of each other when the LSF coefficients are allowed to vary. This suggested that a more complicated structure for the LSF analysis filter should be used which is based on the structure of the synthesis filter. The objective is to make the memory content in both filters remain identical throughout regardless of how the coefficients are changed (as long as they are changed identically for both filters). Using this argument, equations 3.30 and 3.31 are recalled and the LSF analysis filter transfer function is re-expressed as,

$$A(z) = \frac{1}{H(z)}$$
$$= 1 + W(z)$$
$$= 1 + \tfrac{1}{2}\left[\left(P(z) - 1 \right) + \left(Q(z) - 1 \right) \right] \qquad (3.33)$$

Equation 3.33 is repeated diagrammatically in figure 3.8a. The signal flow graph of the new LSF analysis filter is shown in figure 3.8b.



(a)



(b)

Figure 3.8 A new LSF analysis filter.
(a) The structure of the filter (b) The signal flow diagram of the filter

Experimental results showed that the transient effect is completely eliminated by using such an LSF analysis filter and a perfect reconstruction of the original speech signal is always achieved even when the speech is highly non-stationary.

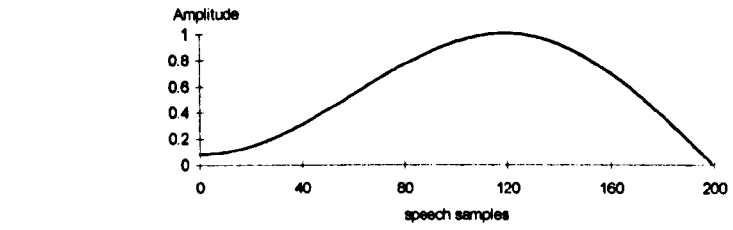
3.4.2 Advantages of using the LSF filters

In this section, the advantages of using the LSF analysis and synthesis filters illustrated above rather than conventional LP ladder-type filters are explored. The advantages are particularly important when the a_i coefficients are interpolated between updates at the decoder and the encoder especially on a sample-by-sample basis. The merits of sample-by-sample interpolation are explored.

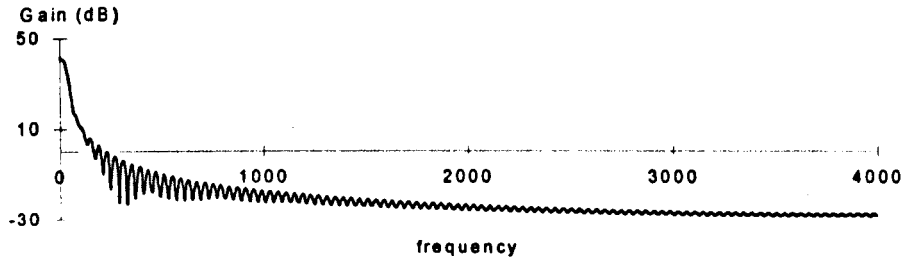
It was suggested [50] that maximum smoothness of the reconstructed speech can be achieved by interpolating the a_i coefficients of the all-pole transfer function on a sample-by-sample basis between one update point to the next. However, instability may occur when a_i coefficients are interpolated in this way. It is easy to demonstrate that interpolating a_i coefficients on a sample-by-sample basis between two sets of stable coefficients can produce intermediate sets which correspond to unstable all-pole synthesis filters. Although an unstable filter coefficient set can be detected and suitable adjustment made, this incurs a considerable computational cost. In addition, transient effects due to the interpolation of the a_i coefficients at voicing transitions may seriously degrade the perceptual quality of the reconstructed speech. Synthesis filter instability can be eliminated by interpolating LSF's instead of a_i coefficients and a smoother evolution of the vocal tract function can be anticipated [50].

In the experiment, the input speech signal was segmented into 20ms frames (160 samples per frame at 8 kHz sampling frequency). Autocorrelation method LP analysis was performed on consecutive windowed speech segments each containing 200 samples, which include 40 overlap samples from the previous speech frame. An asymmetric window composed of half of a Hamming window and a quarter of a cosine function was used, as shown in figure 3.9a. The frequency response of such an asymmetric window is shown in figure 3.9b.

$$w_a(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{2M-1}\right) & 0 \le n \le M-1 \\ \cos\left(\dfrac{2\pi(n-M)}{4(N_w - M - 1)}\right) & M \le n \le N_w - 1 \end{cases} \tag{3.34}$$



(a)

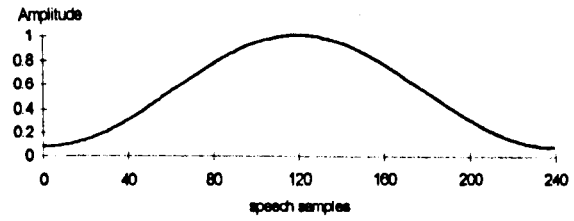


(b)

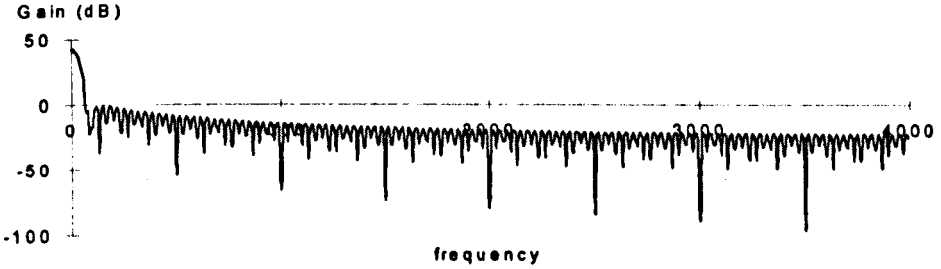
Figure 3.9 An asymmetric window used in autocorrelation LP analysis.
(a) the window shape (b) frequency response of the window

The asymmetric window was chosen because it has a similar frequency response to a 240 samples Hamming window with a smaller system delay [18]. In figures 3.10a and b, the 240 samples Hamming window and its frequency response are shown. In the Hamming window, 40 overlap samples are needed from both the previous and next frames. This means a 5ms coder delay to look ahead the future samples. Through the application of the asymmetric window, the delay can be avoided.

$$w_h(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N_w - 1}\right) \tag{3.35}$$



(a)



(b)

Figure 3.10 A Hamming window used in autocorrelation LP analysis.
(a) the window shape (b) frequency response of the window

The autocorrelation method LP analysis was applied to the windowed speech to yield the a_i coefficients. A 10th order LP analysis was used and 10 filter coefficients were therefore required. In the case of the ladder filters, the a_i coefficients were directly applied to the ladder analysis filter. Otherwise, the a_i coefficients were converted to LSF's using an iterative process (Appendix B) and the LSF analysis filter in figure 3.8 was used. In both filter structures the filter coefficients were linearly interpolated on a sample-by-sample basis. It is known that unstable filter coefficient sets may be found when the a_i coefficients are interpolated. A large proportion of the unstable filter coefficients were found during voicing transitions. Although the unstable filter coefficients would not affect the analysis filter, transient effects occur when the filter coefficients are used in the synthesis filter. Filter instability can be eliminated by calculating the PARCOR coefficients corresponding to the current set of a_i coefficients and ensuring that these lie between +1 and -1. In the experiment, when unstable filter coefficients were detected, the previous stable set of a_i coefficients were used to replace them. In case of the LSF analysis filter, the filter stability is automatically preserved while interpolating the LSF's. This is because if the LSF's are interlaced at the beginning and the end of a speech frame, they will remain interlaced throughout.

3.4.2.1 Objective measures

Two objective measures were used to compare the effect of sample-by-sample interpolation on the two types of filter coefficients. These were a long-term prediction gain (G_{lp}) and the percentage of statistical outliers (SO) for the segmental prediction gain. The long-term prediction gain, comparing the energy of the filter residual with that of the input speech signal, is defined as,

$$G_{lp} = 10\log\frac{\sum_{n=0}^{L-1} s^2(n)}{\sum_{n=0}^{L-1} r^2(n)} \quad (dB) \tag{3.36}$$

where L is the total number of speech samples in the tested speech file

A high G_{lp} means that the analysis filter is likely to reflect the effect of the vocal tract more accurately so that the filter residual may be closer to a true excitation. The

percentage of statistical outliers is useful as a measure of the consistency of the analysis filter. The measurement is made by calculating, on a frame by frame basis, the average segmental prediction gain $\left(\overline{G}_{sp}\right)$ which is defined as,

$$\overline{G}_{sp} = \frac{1}{M}\sum_{m=0}^{M-1} 10\log\frac{\sum_{n=0}^{N-1} s_m^2(n)}{\sum_{n=0}^{N-1} r_m^2(n)} \ (dB) \tag{3.37}$$

where

G_{sp} is the segmental prediction gain

$s_m(n)$ is the nth sample of frame m

N is the number of speech samples in a speech frame

M is total number of speech frame in the tested speech file

A threshold value (G_{th}) is defined,

$$G_{th} = \overline{G}_{sp} - 3 \ (dB) \tag{3.38}$$

and any speech frame which has a segmental prediction gain less than the threshold is classified as an outlier.

i) Objective assessment of the LSF analysis filter

The speech file "OPERATOR.DAT" was used [21] and the objective measurement are summarised in table 3.1.

	Ladder	LSF
G_{lp} (dB)	10.456	10.439
SO (%)	42.11	42.11

Table 3.1 Objective measurements of the two methods

Experimental results indicated that the two methods performed closely, with respect to both the G_{lp} and the SO measure. Thus, it can be concluded that interpolating LSF coefficients produces results which are comparable to those obtained by interpolating a_i coefficients in the respect of minimising the energy of an LP residual, i.e. in reflecting the vocal tract effect.

ii) Objective assessment of the LSF synthesis filter

To assess the performance of the LSF synthesis filter, a CELP coder was used. The CELP coder is presented in Appendix A. In the experiment, the a_i coefficients for each speech sub-frame (40 samples) were computed by linearly interpolating the LSF's across the adjacent frames. These were converted back to the a_i coefficients and used in the synthesis filter. Otherwise, the LSF's were interpolated on a sample-by-sample basis and applied to the LSF synthesis filter directly. The performance of the CELP coder was evaluated using a signal-to-noise ratio (SNR) measure,

$$SNR = \frac{1}{M}\sum_{m=0}^{M-1} 10\log \frac{\sum_{n=0}^{N-1} s_m^2(n)}{\sum_{n=0}^{N-1}\left(s_m(n) - \hat{s}_m(n)\right)^2} \tag{3.39}$$

The speech file "OPERATOR.DAT" [21] was tested and the SNR measures obtained are tabulated in table 3.2.

	40 samples/block	sample-by-sample interpolation
SNR (dB)	11.121	11.513

Table 3.2 The performance of the CELP coders

The results showed that an about 0.4 dB SNR improvement is obtained in the CELP coder using the sample-by-sample LSF synthesis filter.

3.4.2.2 Subjective assessment of both filtering methods

Informal listening tests were carried out to assess the performance of both filter structures. This was done by incorporating the analysis and synthesis filters, using the two filter structures, in a PWI coder [81] and a CELP coder [88]. Twenty people were invited to choose their preferred output speech. The results of the subjective evaluation are shown in figures 3.11 and 3.12. Results from the PWI coder (figure 3.11) showed that 11 out of the 20 people preferred the speech generated by the LSF filters while 2 expressed no preference. Similar results are obtained from the CELP coder, in which the perceptual quality of the reconstructed speech has been improved by using the sample-by-sample LSF synthesis filter. The

sample-by-sample adaptation may be achieved with a little additional computation complexity.

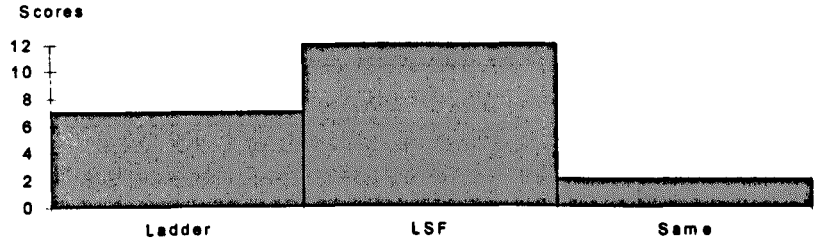


Figure 3.11 Informal listening test in comparing the performance of ladder filters and LSF filters using a PWI coder.

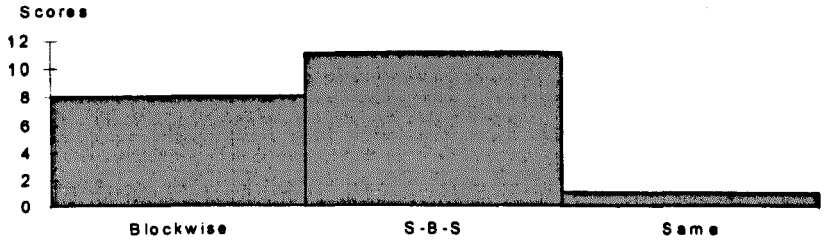


Figure 3.12 Informal listening test in assessing the sample-by-sample LSF synthesis filter using a CELP coder, in which a blockwise interpolated ladder filter is used.

3.4.3 Interpolation of LSF's in various block sizes

In all the above experiments, a sample-by-sample interpolation scheme (i.e. an interpolation block size of 1 sample) was always used for interpolating the LSF coefficients. Different interpolating block sizes in a speech frame were tested for the LSF analysis filter to assess the effect in increasing the size of an interpolation block on the long-term prediction gain G_{lp}. The variation of the block sizes was from 1 speech sample to 160 samples, i.e. from sample-by-sample interpolation of LSF's to no interpolation. The results are presented in figure 3.13.

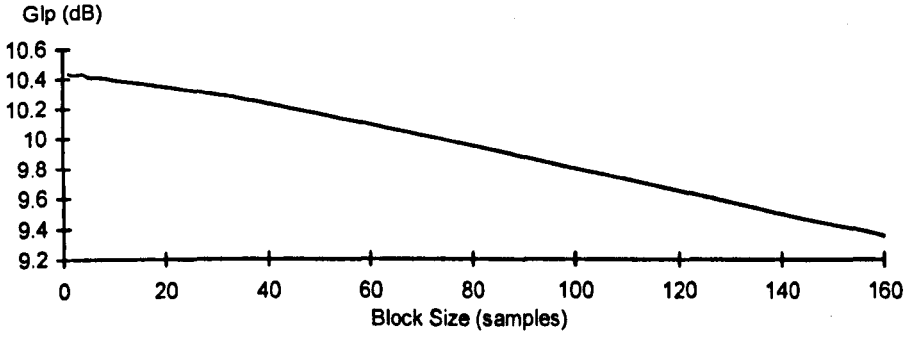


Figure 3.13 G_{lp} performance of the LSF analysis filter under different interpolation block sizes.

It can be seen that the performance of the LSF analysis filter deteriorates when the size of the interpolating block is increased. The maximum performance is achieved

by interpolating the LSF's on a sample by sample basis. The difference in the G_{lp} between the two extremes is about 1.2 dB. It is also observed that the filter performance remained reasonably consistent for block sizes up to about 10 speech samples. The G_{lp} reduced constantly when the interpolating block size is increased towards 160 samples, i.e. no interpolation.

3.4.4 Conclusions of section 3.4

LP filtering using ladder filter structures and different structures called LSF filters have been introduced. Two LSF analysis filter structures have been examined. Experimental results suggested that both the LSF analysis filters work perfectly in steady state. Transient effects occur in the first filter structure (figure 3.6) when the LSF's are adapted. This badly deteriorates the perceptual quality of the synthetic speech, especially during voicing transitions in which a substantial change in the LSF's may occur. Transient effects can be completely eliminated using the LSF filter structure in figure 3.8.

The performance of the LSF analysis and synthesis filters has been compared with conventional ladder structures. Objective measurements showed that both filter structures perform comparably. Informal listening tests have been carried out using a PWI coder and a CELP coder. The results suggested that the decoded speech obtained by using the LSF filters are marginally preferable to those obtained using conventional ladder filters. Owing to the computational simplicity of the LSF filters and the advantage of preserving filter stability during interpolation, it is concluded that the LSF filters can be used as an alternative to conventional ladder filters.

Different interpolating block sizes for the LSF analysis filter have also been investigated. The results suggested that the performance of the LSF analysis filter deteriorates when the number of speech samples in an interpolating block is more than 10 samples. Finally, a sample by sample interpolation of LSF's is recommended in order to maximise the performance of the LSF filters and hence to ensure a smooth evolution of the vocal tract transfer function.

3.5 Comparison of the autocorrelation and Burg's LP analysis method for synthetic speech

In this section the autocorrelation method and Burg's LP analysis method are compared. Tests which have been carried out include pitch-asynchronous and pitch-synchronous LP analysis. The experiments were to apply different LP analysis techniques to segments of synthetic speech. Segments were produced each containing one second of a synthetic vowel sound */a:/* sampled at 8kHz. The synthetic vowel was generated by a 10th order LP synthesis filter excited by a train of impulses. Different pitch-periods were used. The filter coefficients were,

$a_1 = -2.821$ $a_2 = 3.099$ $a_3 = -1.696$ $a_4 = 1.085$ $a_5 = -1.359$

$a_6 = 0.702$ $a_7 = 0.314$ $a_8 = -0.342$ $a_9 = -0.029$ $a_{10} = 0.076$

In figure 3.14, the gain response of the LP synthesis filter is presented. An example of a 160 sample section of the synthetic vowel generated by an impulse train excitation, with pitch-period 40 samples, is shown in figure 3.15.

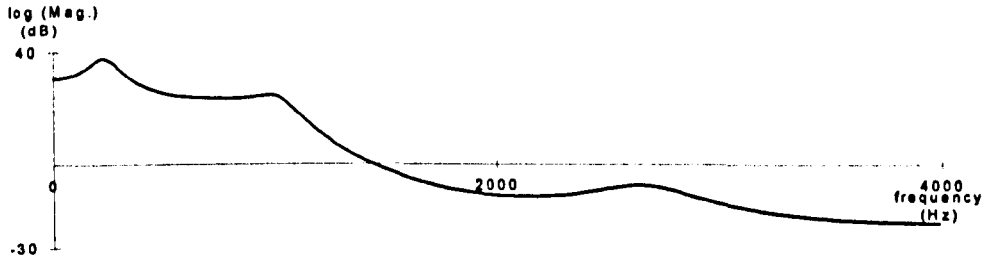


Figure 3.14 Spectral envelope of a tested vowel sound */a:/* across the 4kHz bandwidth.

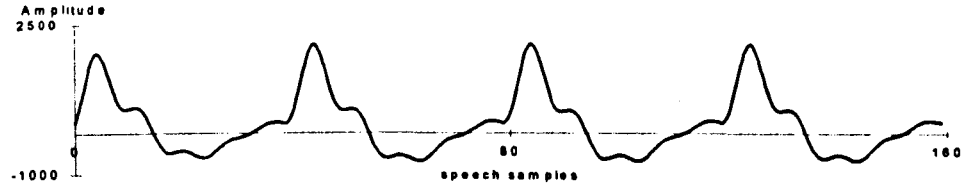


Figure 3.15 A section of synthetic vowel sound */a:/* with pitch-period 40 samples.

A Hamming window $w_h(n)$ was used in the autocorrelation method [43],

$$w_h(n)\begin{cases} 0.54-0.46\cos\left(\dfrac{2\pi n}{N_w-1}\right) & 0 \le n \le N_w-1 \\ 0 & otherwise \end{cases} \tag{3.40}$$

In Burg's method, a rectangular window was used,

$$w_b(n)\begin{cases} 1 & 0 \le n \le N_w-1 \\ 0 & otherwise \end{cases} \tag{3.41}$$

where N_w is the length of the analysis window. The LP analysis was performed on the windowed speech signal, $s_w(n)$. A 10th order LP analysis was used in the experiment. The estimated spectral envelope for a given frame m was compared to the original using a spectral distortion measure, which is defined as [57],

$$D_m^2 = \frac{1}{F_s}\int_0^{F_s}\left[10\log_{10}\left(P_m(f)\right) - 10\log_{10}\left(\hat{P}_m(f)\right)\right]^2 df \ \left(dB^2\right) \qquad (3.42)$$

where

F_s is the sampling frequency (8 kHz)

P_m and $\hat{P}_m$ are the true and estimated LP power envelopes of the mth speech frame respectively,

$$P_m(f) = \frac{1}{\left|A_m\left(\exp\left(j2\pi f / F_s\right)\right)\right|^2} \qquad (3.43a)$$

$$\hat{P}_m(f) = \frac{1}{\left|\hat{A}_m\left(\exp\left(j2\pi f / F_s\right)\right)\right|^2} \qquad (3.43b)$$

The closer D_m is to zero, the better matched will be the estimated spectral envelope to the original for frame m.

3.5.1 Pitch-asynchronous LP analysis

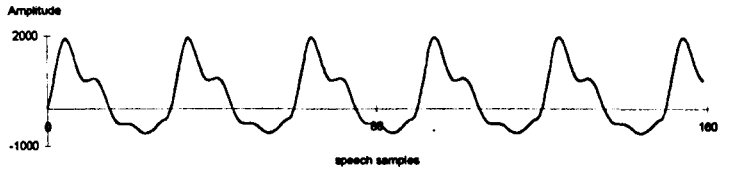
In this section, the performance of the autocorrelation method and Burg pitch-asynchronous LP analysis method are compared. During the experiment, the analysis window was set to a fixed length of 160 samples, i.e. N_w = 160 samples in equations 3.40 and 3.41, to extract the speech samples being analysed. A 10th order LP analysis was performed on each windowed speech segment to yield the a set of a_i coefficients. In case of Burg's method, the k_i coefficient at each stage was computed by equation 3.26 and the set of k_i coefficients was then converted to the a_i coefficients. Note that the term $b^{(i)}\left(n-1\right)$ for n=0 at each stage was set to zero in computing the k_i coefficients at the stage. Once the set of P a_i coefficients was available, they were zero padded to a fixed length of 512 samples and FFT analysis was performed to compute the power spectrum of the estimated all-pole transfer function. The estimated power spectrum was then compared with the reference

spectrum shown in figure 3.14 using the spectral distortion measure. Note that only FFT bin numbers 0 to 256 were included when computing the spectral distortion measure. The analysis was then repeated with the analysis window shifted by one sample on the synthetic speech. A third analysis was carried out with a further one sample shift of the analysis window, and this procedure was continued until the total shift became equal to a complete pitch-cycle.

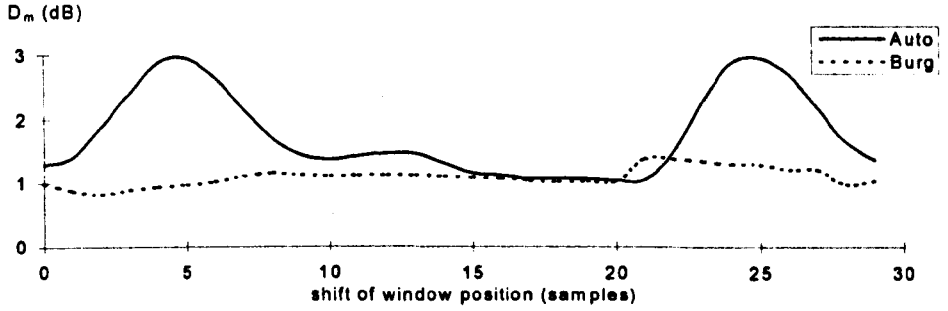
In figures 3.16, the measurements of spectral distortion obtained from both methods using three different pitch-periods are shown. The pitch-periods under test were 30, 80 and 120 sampling intervals. Figures 3.16a, c, and e show the 160 sample frames synthetic speech with pitch-periods of 30, 80 and 120 samples respectively, at zero shift on the analysis window. The results of the spectral distortion measurements for each of these three pitch-periods are shown in figures 3.16b, d and f respectively.

The experimental results suggested that the autocorrelation method performs poorly when either end of the Hamming window coincides with a speech sample with large amplitude. The minimum spectral distortion was found when both ends of the analysis window contain only relatively small speech amplitudes. Similarly, the accuracy of Burg's method could be preserved by avoiding situations where a pitch pulse appears at either end of the rectangular window. An example of such a situation is given in figure 3.16f, where a sudden increase in the spectral distortion is seen at a window shift about 80 samples. Referring to the speech segment in figure 3.16e, such an increase in the spectral distortion corresponds to the situation when the end of the analysis windows coincides with the pitch pulse of the pitch-cycle.

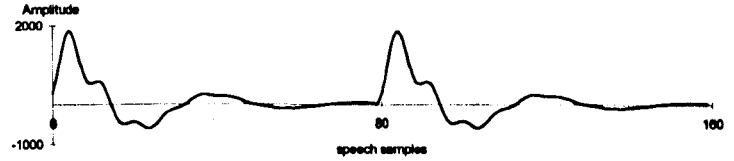
It is also noticed that the performance of both methods deteriorated as the pitch-period of the tested vowel was reduced. This may be due to the lack of information between adjacent pitch peaks on the speech signal. When the pitch-period is small, there is not enough time to allow the complete effect of one excitation impulse to show on the speech samples before the arrival of the next excitation impulse.



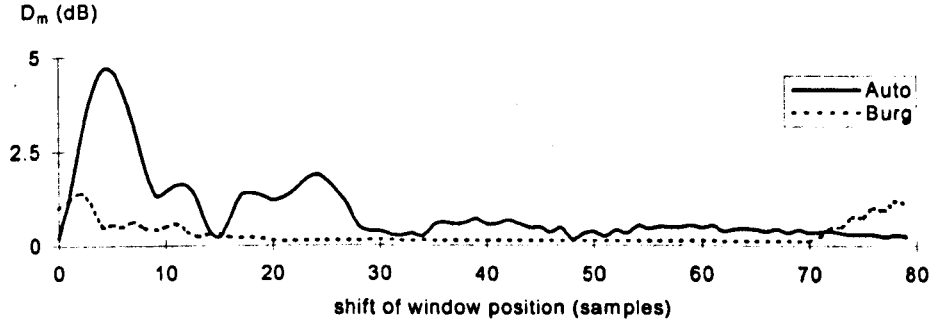
(a)



(b)

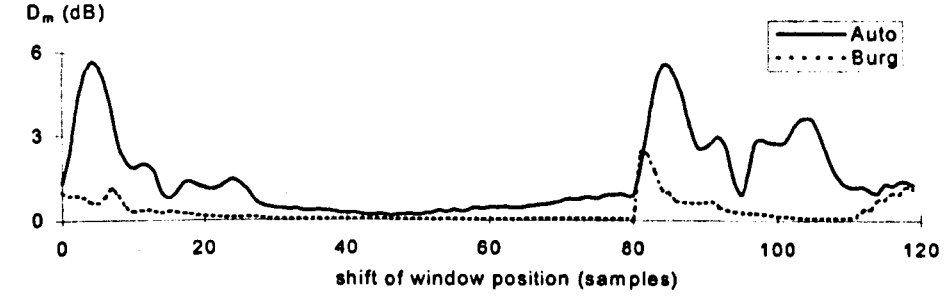


(c)



(d)

(e)



(f)

Figure 3.16 Spectral distortion measure D_m for the autocorrelation and Burg pitch-asynchronous LP analysis methods. (a), (c) and (e) are the speech waveform of the synthetic vowel /a/ under analysed, with pitch-periods 30, 80 and 120 samples respectively (b), (d) and (f) are the spectral distortion measurement for the three cases.

In figure 3.17, the minimum and maximum spectral distortion values over all possible window positions is plotted for a range of pitch-periods from 20 samples to 150 samples in steps of 10 samples. It is observed that when the pitch-period of the tested vowel is small, the variation of the spectral distortion is small. This suggests that both the autocorrelation method and Burg's method perform inadequately with small pitch-periods and that they may be less sensitive to the position of the analysis window. When the pitch-period of the tested vowel was increased, the variation of the spectral distortion increased. In this case window positioning may be crucial. It is also noticed that Burg's method has a more consistent performance than the autocorrelation method. The maximum spectral distortion plot is more constant in Burg's method than for the autocorrelation method. Burg's method always has a lower minimum spectral distortion than the autocorrelation method.

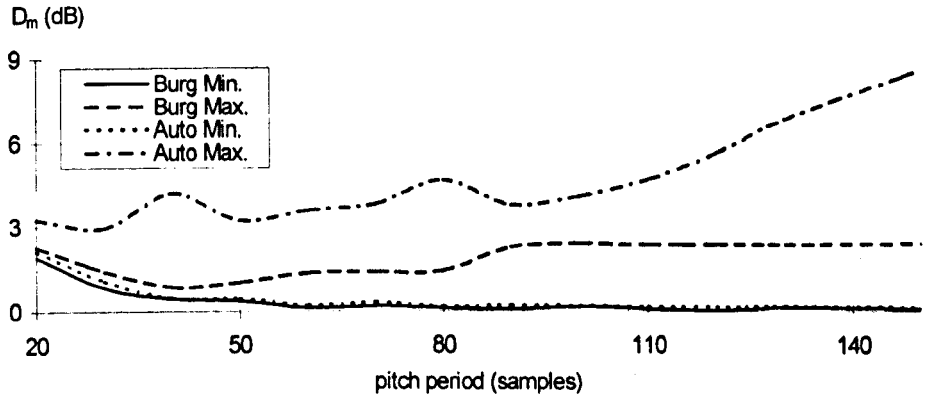
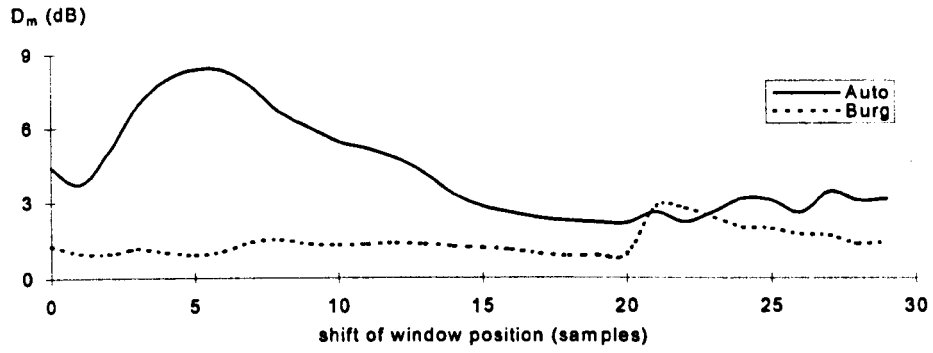


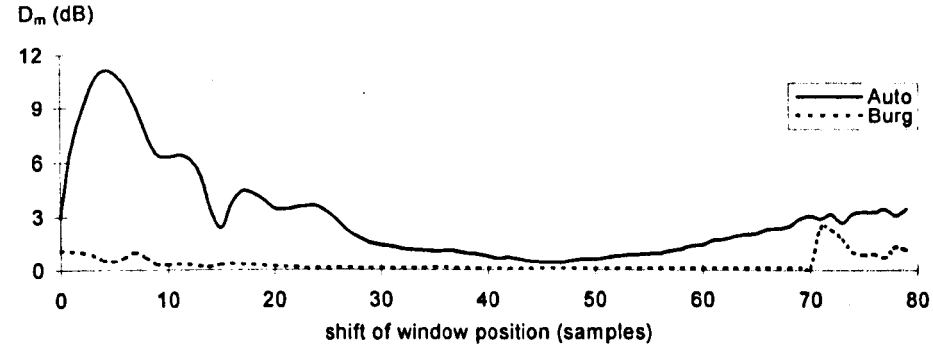
Figure 3.17 Maximum and minimum spectral distortion measurements using pitch-asynchronous LP analysis for various pitch-periods.

3.5.2 Pitch-synchronous LP analysis

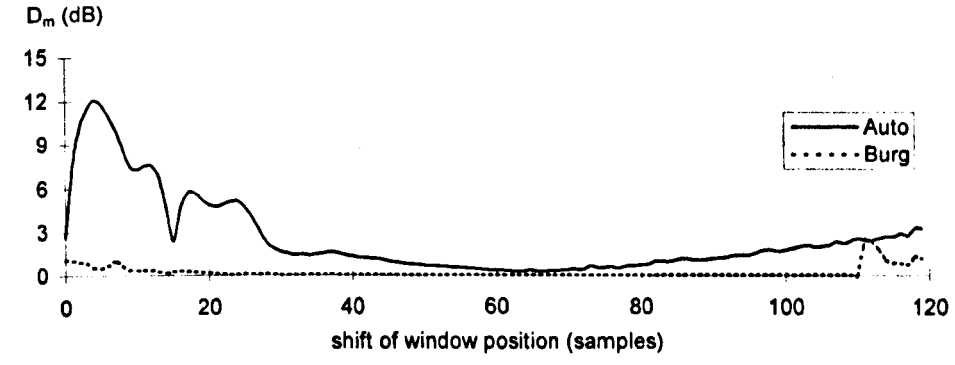
In this experiment a single pitch-cycle was used in LP analysis. The pitch-cycle was extracted using the same window functions, i.e. a Hamming window for the autocorrelation method [43] and a rectangular window for Burg's method, with $N_w = p$. Once again the analysis window was shifted a sample each time in order to examine the sensitivity of window positioning for the pitch-synchronous LP analyses. In figures 3.18a to c, the spectral distortion measurement for the three pitch-periods, 30, 80 and 120 samples respectively, are presented. Furthermore, the maximum and minimum spectral distortion plotted against various pitch-periods from 20 to 150 samples (in steps of 10 samples) are shown in figure 3.19.



(a)



(b)



(c)

Figure 3.18 Spectral distortion measure D_m for the autocorrelation and Burg pitch-synchronous LP analysis methods. (a), (b) and (c) are the results of the vowel /a/ with pitch-periods 30, 80 and 120 samples respectively.

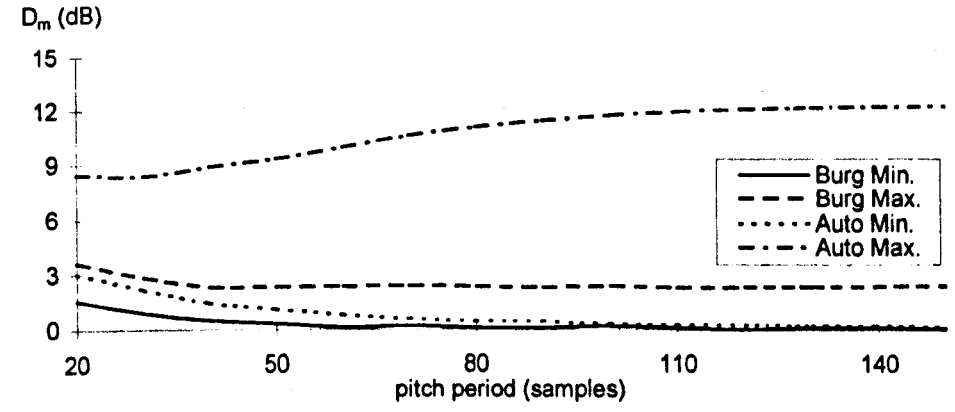


Figure 3.19 Maximum and minimum spectral distortion measurements using pitch-synchronous LP analysis for various pitch-periods.

The results in figures 3.18a to c suggest that the autocorrelation method pitch-synchronous analysis performs badly in all the three cases. Although the minimum spectral distortion may be reduced for a large pitch-period through careful positioning of the analysis window, i.e. shifting the analysis window until both ends of the window contain the lowest possible speech amplitudes, the minimum spectral distortion is not as small as is obtained for the pitch-asynchronous approach. Moreover, the analysis is very sensitive to the position of the analysis window. It can be seen in figure 3.19 that the variation in the spectral distortion is very large when the pitch-period is large.

In the case of Burg's method, an increase in the spectral distortion is found at about 20, 70 and 110 samples window shift in figure 3.18a, b and c respectively. These correspond to the situations when the pitch pulse of the pitch-cycle occurs at either end of the analysis window. In contrast to the autocorrelation method, Burg's method was found to be less sensitive to window positioning in the case of pitch-synchronous analysis. It is seen that the increase in the spectral distortion in figures 3.18a, b and c for Burg's method are much less than that using the autocorrelation method.

In figure 3.19, the maximum spectral distortion (over all possible shifts in window positions) plotted against various pitch-periods from 20 to 150 samples (in steps of 10 samples) for Burg's method is fairly constant and it is not much different from what was obtained with the pitch-asynchronous analysis. As a result the computational cost of LP analysis using Burg's method can be reduced using the pitch-synchronous analysis instead of the pitch-asynchronous analysis, since a smaller speech segment is being analysis in using the pitch-synchronous approach.

It is concluded from sections 3.5.1 and 3.5.2 that pitch-synchronous autocorrelation method LP analysis is not recommended. Burg's pitch-synchronous analysis performs similarly to the asynchronous approach in case of large pitch-period speech. However when the pitch-period is small, pitch-synchronous analysis is not suitable since the possible maximum spectral distortion is always quite high.

3.5.3 Pitch-synchronous LP analysis using multiple pitch-cycles

Another way of implementing pitch-synchronous LP analysis is investigated in this section. In this method, two pitch-cycles are analysed instead of a single cycle. The synthetic speech used in the previous section with three different pitch-periods was analysed again. The spectral distortion measures obtained for the three pitch-periods are presented in figures 3.20a to c. The maximum and minimum spectral distortion plotted against various pitch-periods from 20 to 150 samples (in steps of 10 samples) is shown in figure 3.21.

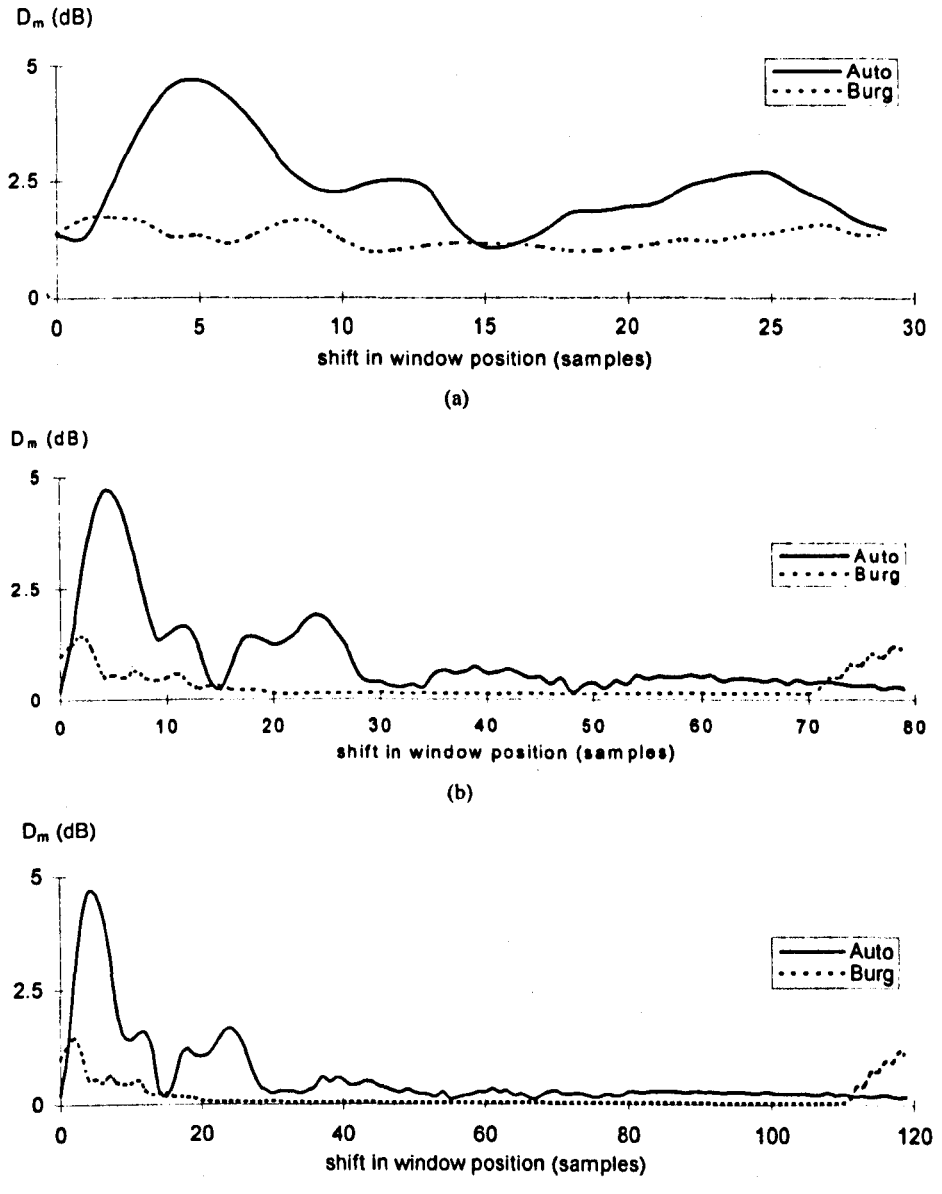


Figure 3.20 Spectral distortion measure D_m for the autocorrelation and Burg pitch-synchronous LP analysis methods using windows of length of two pitch-cycles. (a), (b) and (c) are the results for the vowel /a/ with pitch-periods 30, 80 and 120 samples.

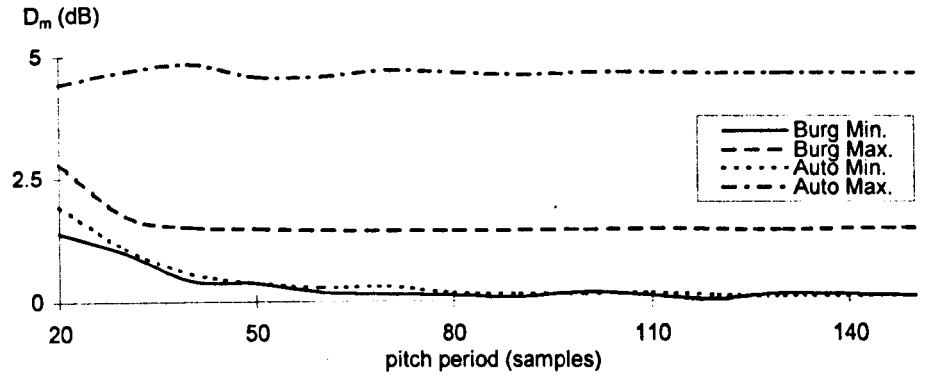


Figure 3.21 Maximum and minimum spectral distortion measurements using windows of length of double-pitch pitch-cycles for various pitch-periods.

The results suggest that the double-pitch approach has a much better performance than the single-pitch pitch-synchronous LP analysis. In the autocorrelation method, it is found that when the pitch-period of the input signal is small, the pitch-asynchronous LP analysis seemed to have a better performance than the double-pitch pitch-synchronous approach. However when the pitch-period is increased, the double-pitch approach worked as well as the pitch-asynchronous approach. It is seen in figure 3.21 that both the minimum and maximum spectral distortion are decreased in comparison to figure 3.19. A similar argument holds in Burg's method, in which the maximum spectral distortion (over all possible window shifts) is decreased when two pitch-cycles were used. Not much improvement is obtained in the minimum spectral distortion for Burg's method.

Therefore the performance of pitch-synchronous LP analyses can be enhanced by using two pitch-cycles instead of a single cycle. To further explore the problem, the number of pitch-cycles included in the pitch-synchronous analyses was increased to 3, 4 and then 5. The maximum and minimum spectral distortion measures (over all possible shifts in window positions) obtained for different number of pitch-cycles are tabulated in tables 3.3 to 3.5, for pitch-periods of 30, 80 and 120 samples.

Results from the autocorrelation method suggested that when the number of pitch-cycles used in the analysis is increased, the minimum spectral distortion (over all possible shifts) decreases. The number of pitch-cycles required to achieved a reasonably low minimum spectral distortion is about three. A point of saturation is

found when the number of pitch-cycles used is more than four, where the minimum spectral distortion started to rise. Furthermore by increasing the number of pitch-cycles used, the maximum spectral distortion (over all possible shifts) was reduced. It is seen in tables 3.2 to 3.5 that the difference between the minimum and maximum spectral distortion measures is always very large in all the three pitch-periods.

In case of Burg's method, there is not much effect on the minimum spectral distortion for large pitch-period speech. When the pitch-period is small, an improvement in the minimum spectral distortion is observed. Similarly to the autocorrelation method, the maximum spectral distortion reduced when the number of pitch-cycles included in the analysis was increased. It is also seen in tables 3.2 to 3.5 that the difference between the minimum and maximum measures obtained using Burg's method was smaller than that obtained using the autocorrelation method.

No. of pitch-cycles	Autocorrelation D_m (dB)		Burg D_m (dB)	
	min.	max.	min.	max.
2	1.065	4.682	0.968	1.703
3	1.042	4.505	0.981	1.397
4	1.033	4.135	0.959	1.304
5	1.035	3.849	0.939	1.252

Table 3.3 The maximum and minimum measure of spectral distortion measure using various number of pitch-cycles for pitch-period = 30 samples

No. of pitch-ycles	Autocorrelation D_m (dB)		Burg D_m (dB)	
	min.	max.	min.	max.
2	0.161	4.658	0.117	1.416
3	0.078	4.360	0.117	0.850
4	0.081	3.938	0.118	0.611
5	0.084	3.609	0.120	0.481

Table 3.4 The maximum and minimum measure of spectral distortion measure using various number of pitch-cycles for pitch-period = 80 samples

No. of pitch-cycles	Autocorrelation D_m (dB)		Burg D_m (dB)	
	min.	max.	min.	max.
2	0.116	4.650	0.042	1.442
3	0.081	4.376	0.042	0.865
4	0.045	3.928	0.043	0.618
5	0.058	3.600	0.038	0.482

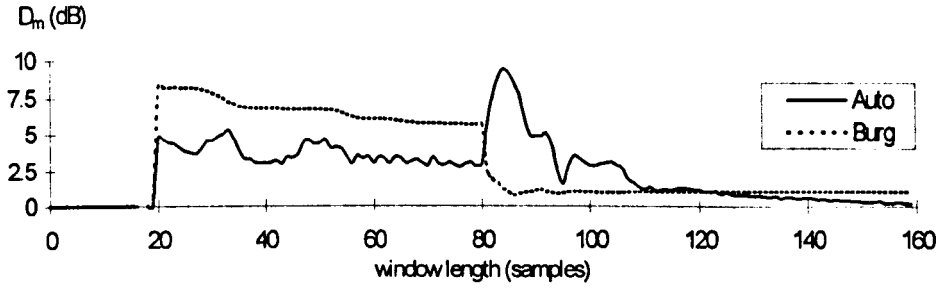
Table 3.5 The maximum and minimum measure of spectral distortion measure using various number of pitch-cycles for pitch-period = 120 samples

3.5.4 LP analysis with only a small number of speech samples available

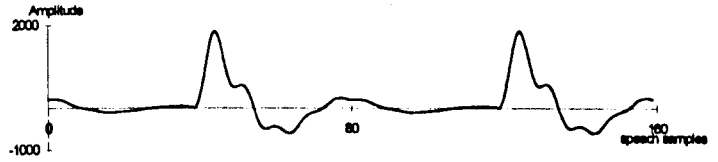
In this experiment, LP analysis was tested applied to segments of voiced speech including segments whose lengths were smaller than a complete pitch-cycle. To simplify the situation only the results obtained for a pitch-period equal to 80 samples are presented. The beginning of the analysis window was located at the beginning of a pitch-cycle for the synthetic speech. The length of the analysis window was initially set to 20 samples and this was increased sample by sample until it reached 160 samples. A window length of 160 samples is equivalent to pitch-asynchronous LP analysis with 2 pitch-cycles per analysis frame. The analysis was then repeated with the beginning of the analysis window shifted by 10 samples. This process was repeated until the total shift become equal to a complete pitch-cycle. Results obtained for three special cases are shown in figures 3.22a to f, together with the corresponding frame of 160 speech samples. The three frames of speech, in figures 3.22a, c and e, differ only in the starting position of the window within the pitch-cycle.

The results show that the autocorrelation method performs poorly when the number of speech samples available is less than a complete pitch-cycle. Optimal performance in the spectral distortion, referring to the results in table 3.4 which is 0.078dB, may be achieved when the number of pitch-cycles available is at least two pitch-cycles. Burg's method exhibits a very different property that when the position of the analysis window is carefully located, optimal performance in the spectral distortion, referring to the results in table 3.4 which is 0.117dB, may be obtained even if the number of available speech samples is less than a complete pitch-cycle. It is seen in figure 3.22f that the optimum performance was achieved when the window length was about 40 samples. This result was achieved when the beginning of the window was located in the pitch-cycle as shown in figure 3.22e. However, this is not the case when the beginning of the window is located on the pitch-cycle as shown in figure 3.22a. In this case, the optimum performance may only be achieved when the window length was about 80 samples, i.e. the complete pitch-cycle is accommodated in the analysis window. Experiments have also shown that this advantage of Burg's method will be gradually lost when the pitch-period of the voiced speech decreases.

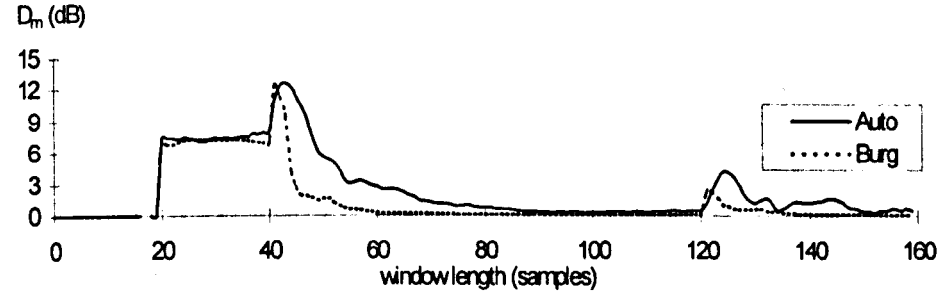
(a)



(b)

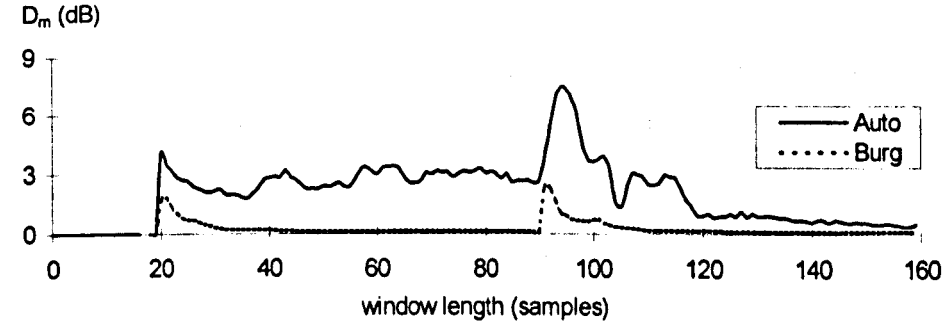


(c)



(d)

(e)



(f)

Figure 3.22 Spectral distortion measures D_m of the autocorrelation and Burg LP analysis method with variable analysis window length. The pitch-period of the voiced speech is 80 samples. (a), (c) and (e) are the speech signals under investigation (b), (d) and (f) are the corresponding D_m measures.

3.5.5 Conclusions of section 3.5

The results presented in section 3.5 suggest that the Burg LP analysis method always yields a more accurate estimation of the spectral envelope of synthetic voiced speech than the autocorrelation method. This is the case for both pitch-asynchronous and pitch-synchronous analysis. In the autocorrelation method, pitch-synchronous LP analysis is not capable of providing accurate spectral estimation unless more than one pitch-cycle is available in the analysis window. Although the performance could be improved by increasing the number of pitch-cycles available, this would not be appropriate in practice. The number of pitch-cycles required to achieve a reasonably low spectral distortion would be about three and the increase in computational cost required for voiced speech with large pitch-periods would be unacceptable. The accuracy of the autocorrelation method was found to be very sensitive to the location of the analysis window. Optimal performance may be achieved by ensuring that both ends of the analysis window coincide with small speech amplitudes.

Burg's method seemed to be better than the autocorrelation method in terms of spectral estimation accuracy and consistency. The accuracy of Burg's method was also found to be very sensitive to the location of the analysis window. Burg's pitch-synchronous LP analysis performed similarly to the pitch-asynchronous approach for large pitch-period voiced speech, by carefully adjusting the position of the analysis window. Burg's pitch-synchronous LP analysis method is recommended for voiced speech with large pitch-period since window positioning may be easier in this case and the computational cost can be reduced, compared to using the pitch-asynchronous analysis. When the pitch-period is small, window positioning is not so easy because the duration between successive pitch pulses is small. An increase in the number of pitch-cycles in the analysis window helps to reduce the possible maximum spectral distortion. Burg's method exhibited further merit in that accurate estimation is still possible when the number of speech samples is less than a complete pitch-cycle, through carefully locating the analysis window in the pitch-cycle.

3.6 Comparison of the autocorrelation and Burg's LP analysis method for clean natural speech

In this section, the performance of the autocorrelation and Burg LP analysis methods were compared for natural voiced speech using an average segmental prediction gain measure $\overline{G}_{sp}$ defined by equation 3.37. The higher the value of $\overline{G}_{sp}$, the more the vocal tract effect is removed from the speech signal and the more accurate is the spectral envelope estimation likely to be. The experiment was conducted on the speech file "OPERATOR.DAT" [21] (1800 frames). The experimental procedure is now described.

Prior to the LP analysis, the two-way pitch detector described in chapter 2 was used to estimate the nature of a 160 sample speech frame and to yield the pitch-period when the speech was classified as voiced. The analysis windows defined in equations 3.40 and 3.41 were used to extract the required speech samples for the autocorrelation method and Burg's method respectively. The LP analysis window was located in a way such that the centre of the analysis window coincided with an "update-point" between two consecutive 160 sample frames of the input speech as shown in figure 3.23.

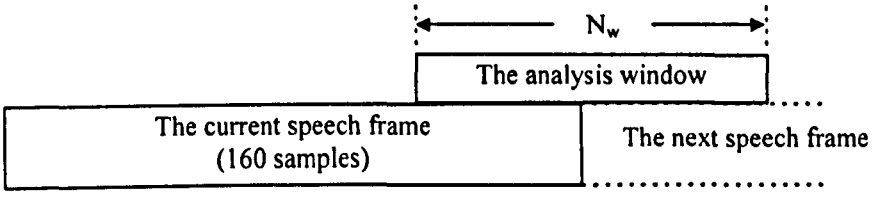


Figure 3.23 Positioning of the analysis window within a speech frame for LP analysis

Both pitch-asynchronous and pitch-synchronous LP analysis methods were tested. For pitch-asynchronous LP analysis, the size of the window was equal to the frame length, i.e. 160 samples. For Burg's pitch-synchronous LP analysis method, the window length was set equal to the instantaneous pitch-period at the update-point. Since it was seen that the autocorrelation method performs poorly for pitch-synchronous LP analysis, two pitch-cycles were used. This was done by repeating the pitch-cycle itself and thus the window length was twice the instantaneous pitch-period. The instantaneous pitch-period was computed by searching through pitch-

period candidates, in the range ±25% of the pitch-period estimated by the pitch detector described in chapter 2 applied to speech centered on the update-point. Suppose the instantaneous pitch-period is labelled as p. Then p is taken as the candidate which yields the maximum cross-correlation function among the group of pitch period candidates as,

$$p = \max_{m} \left\{ \frac{\sum_{n=N+\frac{p_e}{2}-1}^{N+\frac{p_e}{2}-m} s(n)\,s(n-m)}{\sqrt{\sum_{n=N+\frac{p_e}{2}-1}^{N+\frac{p_e}{2}-m} s^2(n) \sum_{n=N+\frac{p_e}{2}-1}^{N+\frac{p_e}{2}-m} s^2(n-m)}} \right\} \qquad (3.44)$$

$$0.75 * p_e \le m \le 1.25 * p_e \ (m \in \text{integer})$$

where

N is the frame length equals to 160 samples

p_e is the pitch-period estimated by the pitch detector.

For unvoiced speech frames the window length was always set to 160 samples, i.e. pitch-asynchronous was always used for unvoiced speech. A 10th order LP analysis was used. The LSF analysis filter structure shown in figure 3.8 was used to filter the speech signal and to obtain the residual signal for computing the segmental prediction gain G_{sp} for each speech frame. The average segmental predication $\overline{G}_{sp}$ measured over the 1800 speech frames of the speech file "OPERATOR.DAT" are listed in table 3.6,

	Pitch-asynchronous	Pitch-synchronous
Autocorrelation method	10.243 (dB)	9.815 (dB)
Burg's method	10.342 (dB)	10.244 (dB)

Table 3.6 The average segmental prediction gain obtained by autocorrelation and Burg LP analysis

The results suggested that Burg's method always yields a better performance than the autocorrelation method. For the pitch-asynchronous analysis, Burg's method is better than the autocorrelation method by about 0.1dB. For the pitch-synchronous analysis, Burg's method is better than the autocorrelation method by about 0.4dB

It is interesting to note that the prediction gain obtained from the pitch-synchronous method is always smaller than from the pitch-asynchronous method. This is because in the pitch-asynchronous analysis, the LP filter coefficients are derived to minimise the error signal across the entire speech frame. This is not the case in pitch-synchronous analysis, since the filter coefficients are calculated such that the error signal in the analysed pitch-cycle is minimised. Any rapid change in the speech signal which may occur within a speech frame would be unpredictable and the error signal could be large in those regions. An example of such a scenario is shown in figure 3.24. It is shown in figure 3.24a that there is a rapid shape change in the middle of the speech frame. Using Burg's pitch-asynchronous analysis, the error signal shown in figure 3.24b is minimised for the entire speech frame. However, the error signal is increased in the middle of the speech frame, shown in figure 3.24c, when Burg's pitch-synchronous analysis is implemented.

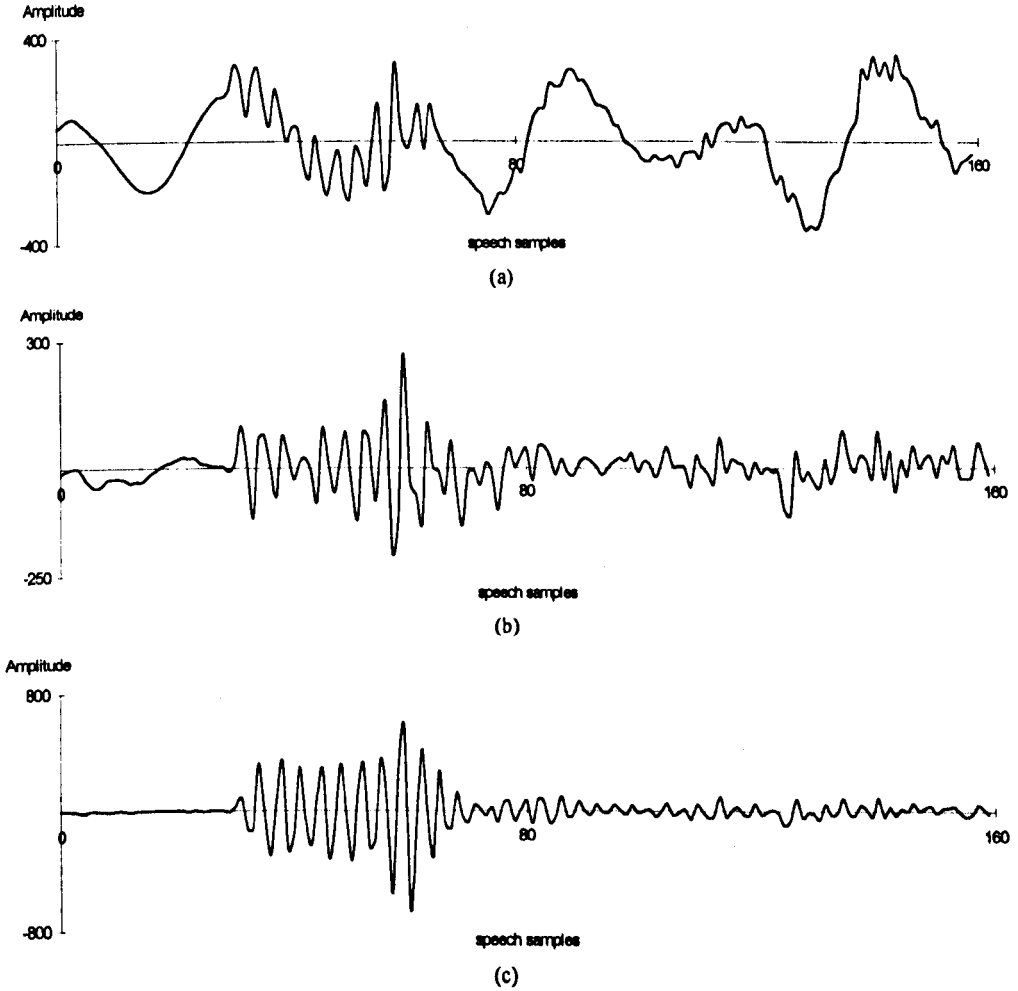


Figure 3.24 An example of the difference between pitch-asynchronous and synchronous LP analysis when there is a rapid shape change in a speech frame Note the different vertical scales). (a) the speech signal under analysed (b) the filtered residual obtained by Burg's pitch-asynchronous LP analysis (c) the filtered residual obtained by Burg's pitch-synchronous LP analysis

An example of spectral envelopes of a voiced segment resulting from the autocorrelation and Burg's pitch-asynchronous LP analyses is shown in figure 3.25. The original speech signal was analysed by a 512 point FFT with a Hamming window. It is shown that both the autocorrelation and Burg methods were able to estimate the location of the first two formants. The autocorrelation method seemed not to be able to track the speech formant when the frequency increased, where the third formant has been smoothed out. Using Burg's method, the third formant is well preserved. Note also that the formants estimated by Burg's method always have a narrower bandwidth than the autocorrelation method.

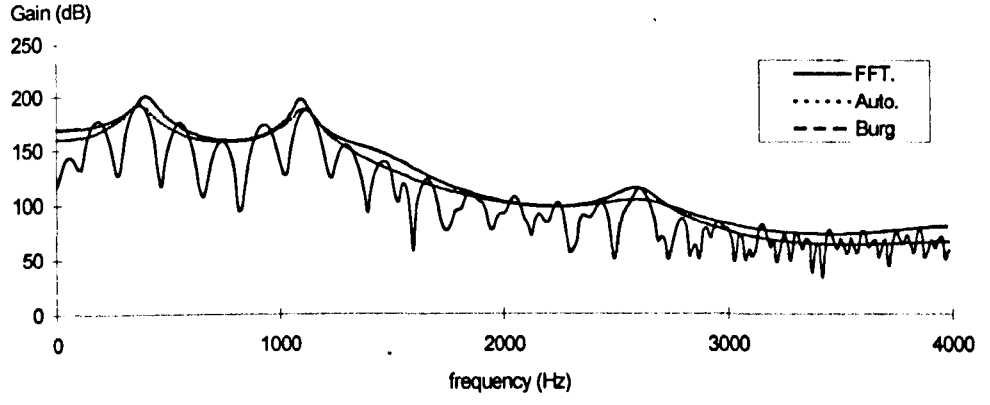


Figure 3.25 An example of the spectral envelopes estimated by the autocorrelation and Burg's methods.

3.6.1 An adaptive analysis window for Burg's pitch-synchronous LP analysis method

Pitch-synchronous LP analysis may be preferable in a speech coder which implements an interpolation technique. This is because the local properties of speech may be destroyed by pitch-asynchronous LP analysis. The scenario is important in voiced speech which has small pitch-periods since the speech production is a non-stationary process. Experimental results showed that Burg's method is the more suitable for pitch-synchronous LP analysis. An adaptive analysis window size is proposed in this project incorporating Burg's synchronous LP analysis method, in which,

$$N_w = \begin{cases} 3*p & p < 30 \\ 2*p & 30 \le p < 45 \\ p + P & otherwise \end{cases} \tag{3.45}$$

where

p is the pitch-period

P is the filter order.

The design was based on the experimental observation that when the pitch-period of the voiced speech is small, poor spectral estimation is obtained. Hence more pitch-cycles are used in order to reduce the maximum possible spectral distortion. Otherwise, a single pitch-cycle is used with an optimisation procedure. The optimisation procedure checks a number of speech samples around both ends of the analysis window. If a pitch pulse appears in either end of the analysis window, the analysis window is shifted forward to avoid the pitch pulse. In a 10th order LP analysis, the pitch pulse is not allowed to occur in the first and last 10 samples of the analysis window.

The adaptive window length scheme was tested using the same speech file [21] and the average segmental gain obtained was,

$$\overline{G}_{sp} = 10.304\ (dB)$$

Comparing the result with those in table 3.6, it is seen that the average segmental gain obtained here lies between those obtained for the pitch-asynchronous and the pitch-synchronous analyses.

3.7 Conclusions

A true inverse LSF analysis filter has been proposed for operating with the LSF synthesis filter. Experimental results suggested that the performance of the LSF analysis and synthesis filters is comparable with conventional ladder filter structures in both objective and subjective measurements. By using the LSF filters in a speech coder, the LSF's may be directly applied to the synthesis filter at the decoder. This eliminates the computational cost of converting the LSF's to a_i coefficients. It was found that the performance of the LSF analysis and synthesis filters is optimised by interpolating the LSF's, on a sample-by-sample basis between adjacent update intervals. As a result a smooth evolution of the vocal tract function is obtained. Filter stability is guaranteed throughout the interpolation process by interpolating LSF's. This would not be the case with a_i coefficients.

The autocorrelation and Burg LP analysis methods have been compared using both synthetic speech and clean natural speech. Results using the synthetic speech indicate that Burg's method has a better performance than the autocorrelation method, both when the analysis is pitch-asynchronous and when the analysis is pitch-synchronous. The spectral estimation accuracy of the autocorrelation method is very sensitive to the position of the analysis window. In case of pitch-synchronous analysis, the autocorrelation method is only recommended when the analysis window contains at least two pitch-cycles. The accuracy of Burg's method is not as sensitive to window positioning as the autocorrelation method. Experiments show that in Burg's method, an accurate spectral estimation is still possible even if the window length is smaller than a pitch-cycle, by carefully positioning the analysis window. Finally, an adaptive window length has been proposed for Burg's pitch-synchronous LP analysis when it is used with real speech.

Chapter 4

Quantisation of Line Spectral Frequencies

4.1 Introduction

Quantisation is a process which permits digital representation of a continuous signal. In the case of speech coding, this can be performed either directly on the analogue speech waveform or on parameters which characterise segments of the speech signal. There are two major categories of quantisation schemes, scalar quantisation (SQ) and vector quantisation (VQ). In scalar quantisation, code-words are assigned to individual signal parameters separately. The simplest form of scalar quantiser is known as a uniform quantiser. In a uniform quantiser, the probability density function (p.d.f.) of an input signal is assumed to be uniform and constant quantisation intervals are used across all the quantisation levels. The performance of a uniform quantiser deteriorates as the p.d.f. of the input signal moves away from a uniform distribution. To increase quantiser performance, more quantisation levels must be allocated to the range which has a high probability density and fewer levels to the statistically less probable values. A scalar quantiser based on this idea is known as a non-uniform quantiser and a typical example of this kind is Log-PCM.

It is known that the efficiency of quantisation schemes can be increased by quantising groups of consecutive samples together, i.e. vector quantisation [53], rather than quantising the samples individually. Vector quantisation (VQ) is a pattern matching technique which requires a code-book at the encoder and an identical copy of this code-book at the decoder [52]. The code-book is an array of vectors each with a unique index or address. The vectors, referred to as reference vectors, are the result of an exhaustive training procedure which analyses a large data-base of typical vectors and attempts to define a set of code-book vectors which best represents this data-base. It is believed [53] that the efficiency of a code-book is dependent on the suitability of the training vectors. The code-book vector which is best matched to the target vector is found and its index is encoded as the quantised

target vector. A simple application of vector quantisation to speech coding was realised by applying it directly to vectors of speech samples, i.e. VPCM. It is not surprising that this proved to be an inefficient form of vector quantisation since the dynamic range of the speech samples is so large that a huge code-book is necessary to cover this range. With the invention of linear prediction analysis, speech segments could be characterised by sets of source parameters [6] which describe the speech production model. Vector quantisation can be applied to these source parameters. Fewer bits are then needed to achieve a specified level of speech quality than if the parameters were scalar quantised [51]. Alternatively better speech quality can be achieved at a given bit-rate.

Vector quantisation is of considerable interest in both speech processing and image processing owing to its reliability in signal compression. Much research world-wide is going on to investigate various issues associated with VQ. This research includes work on quantiser structure, code-book training, code-book searching techniques and efficiency of training data. In the area of speech coding, associated problems include also the choice of the best set of source parameters. It has already been mentioned in chapter 3 that the speech production model can be decomposed into a vocal tract transfer function and an excitation signal. Many types of parameters which characterise the vocal tract transfer function have been suggested [55][56]. Each of these has a different sensitivity to quantisation noise which affects the quantised short-term spectral envelope. Among them, line spectral frequencies have been widely used. Line spectral frequencies are in some ways related to speech formants. By quantising LSF's, the spectral error caused by quantisation noise can be localised in frequency [56].

This chapter is structured as follows. A number of vocal tract parameters will be introduced in section 4.2. In section 4.3, problems associated with the design of vector quantisers will be discussed. These problems include quantiser complexity, code-book training, quantiser structure and performance assessment. In section 4.4, the design procedures for a 24-bit MS-LSF vector quantiser (MS - multi-stage split) will be presented. These include the effect of utilising a weighting factor during code-book training and searching. A re-ordering procedure which aims to preserve

the correct order of a set of LSF's after VQ will also be discussed. In section 4.5 the performance of the 24-bit MS-LSF vector quantiser is improved by introducing an interfame quantisation scheme to yield a 24-bit IMS-LSF vector quantiser (IMS - interframe multi-split).

4.2 Alternative representations of LP ladder filter coefficients

Conventional LP ladder filter coefficients, a_i, are not recommended for direct use in quantisation. This is because the frequency response of the LP synthesis filter is very sensitive to quantisation error in the LP ladder filter coefficients. Moreover, the stability of the synthesis filter cannot easily be guaranteed after quantisation. A number of alternative representations have been proposed in the literature [55]. Synthesis filter stability can be easily guaranteed when using LP poles or PARCOR coefficients to characterise the synthesis filter, even when these are quantised. LP poles are the roots of the denominator polynomial of the all-pole transfer function and are directly related to speech formants. Synthesis filter stability can be guaranteed by ensuring that the set of LP poles lie inside the unit circle. The drawback of using poles to represent the LP transfer function is that their determination is computationally expensive.

PARCOR coefficients may be obtained as the by-product of LP ladder filter coefficient computation. They are widely used in quantisation schemes because filter stability can be guaranteed by making sure that the quantised PARCOR coefficients lie between ±1. PARCOR coefficients can be used directly as the multiplier coefficients of a lattice filter. This eliminates the computational cost of converting between PARCOR coefficients and LP ladder filter coefficients. Despite the advantages of PARCOR coefficients, the spectral envelope becomes much more sensitive to quantisation error in PARCOR coefficients which are close to the boundaries ±1 than to the same error in PARCOR coefficients far from ±1. Therefore each PARCOR coefficient is transformed to another domain before quantisation. Among the transformations commonly used are [1],

a) Log-Area Ratios (LAR_i)

$$LAR_i = \log \frac{1 + k_i}{1 - k_i} \tag{4.1}$$

b) Inverse sine transform (IS_i)

$$IS_i = \arcsin(k_i) \tag{4.2}$$

In either transformation the scale of the PARCOR coefficient k_i is warped such that a uniform quantisation of the transformed parameter corresponds to a suitable non-uniform quantisation of a PARCOR.

LSF's are the most popular parameters used to quantise the short-term spectral information of segments of speech. They can be quantised in either scalar or vector form. By quantising LSF's, the spectral error caused by quantisation noise can be, to a degree, localised in the frequency-domain. Through the utilisation of LSF's, filter stability can be easily preserved by maintaining the interlacing property of the set of LSF coefficients. Finally, computational efficiency of a speech coder can be maintained by using the LSF's directly as the multiplier coefficients of the LSF analysis and synthesis filters discussed in chapter 3 section 3.4.

4.3 Vector quantisation

Vector quantisation is a pattern matching technique in which an input vector $\underline{x}$ is compared with each member of a set Γ of reference vectors stored in a code-book [52]. This is carried out by computing a distance (or distortion) measure between the input and the reference vector. Each reference vector in the code-book has an index (or address) referred to as a code-word. The code-word of the reference vector which produces the smallest distance or distortion is taken by the encoder to represent the input vector. At the decoder, the code-book index allows the required vector to be fetched from an identical copy of the code-book used by the encoder.

4.3.1 Complexity of a vector quantiser

The complexity of a vector quantiser depends on the computational load required for searching the code-book and the memory required to store the code-book at both the encoder and the decoder. These factors could be affected by: i) code-book size and dimension and ii) selection of distance or distortion measure [53].

i) Code-book size and dimension

The first issue to be decided in designing a vector quantiser is the size and dimension of the code-book. The size is the number of vectors in the code-book and the dimension is the length of each of the vectors. The larger the size of the code-book the more memory space will be required and the more computational effort will be needed for the code-book searching.

For a code-book of dimension N and size L which is a power of 2, the number of bits required to encode a vector is B where,

$$L = 2^B \text{ (levels)} \tag{4.3}$$

The number of bits per dimension (i.e. bits per single vector element) is,

$$R = \frac{B}{N} \text{ (bits/dimension)} \tag{4.4}$$

If the system requires F_c code-words per second, the transmission bit-rate T is,

$$T = B F_c \text{ (bits/sec)} \tag{4.5}$$

In a fully searched code-book, the number of distance or distortion computations required for a single input vector will be L. Assuming each distance or distortion computation requires a total number of N multiply-add operations (this is true for the mean-square error measure), the total computational cost for a single input vector is,

$$C = N L = N 2^{RN} \text{ (operations)} \tag{4.6}$$

Hence the computational cost for a quantiser using a full-search code-book, is linearly proportional to the size L of the code-book. Furthermore for a given number of bits per element R, the computational cost grows exponentially as the vector dimension increased.

If we assume a unit memory location is used to store a single code-word, the total memory locations required to store the entire code-book will be,

$$M = N\,L = N\,2^{RN}\ (\text{locations}) \tag{4.7}$$

Once again, the memory cost in the system grows exponentially with the dimension N of the code-word as well as the number of bits per dimension R.

ii) Selection of a distance or distortion measure

The definition of a distance or distortion measure $d(\underline{x}, \underline{y})$ for vectors $\underline{x}$ and $\underline{y}$ determines the performance of a quantiser. When $\underline{x}$ is a given input vector and $\underline{y}$ is a reference vector from the code-book set Γ, $d(\underline{x}, \underline{y})$ is said to be the distance or distortion from $\underline{x}$ to the reference vector $\underline{y}$. In the area of speech coding, the measure used must be subjectively meaningful. A low average distance or distortion for a portion of speech should be indicative of good subjective quality. There are two commonly used definitions of $d(\underline{x}, \underline{y})$. They are the mean-square-error distance measure and the Itakura-Saito distortion measure. Note that $d(\underline{x}, \underline{y})$ need not be symmetric, i.e. $d(\underline{x}, \underline{y})$ need not be equal to $d(\underline{y}, \underline{x})$. The term "distance" may be used when $d(\underline{x}, \underline{y}) = d(\underline{y}, \underline{x})$. Otherwise the term "distortion" is used and the second vector must always be $\underline{y}$ i.e. a member of the set Γ.

a. Mean-square-error measure (mse)

A general distance measure based on the L_Z norm is given as [53],

$$d_z(\underline{x}, \underline{y}) = \frac{1}{N} \sum_{i=1}^{N} \left| x_i - y_i \right|^z \tag{4.8}$$

where x_i and y_i are the element in the vectors $\underline{x}$ and $\underline{y}$.

In case of the mse measure, z is set to 2, i.e.,

$$d_2(\underline{x}, \underline{y}) = \frac{1}{N} \sum_{i=1}^{N} \left(x_i - y_i \right)^2 \tag{4.9}$$

The mse measure is widely used for waveform coding because of its simplicity in computation. However, the mse measure itself is not necessarily subjectively meaningful. To be useful in speech coding, weighting may be introduced to render

certain contributions to the distortion more important than others. Hence the mse measure can be redefined as,

$$d_W(\underline{x}, \underline{\gamma}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \gamma_i)^2 W_i \tag{4.10}$$

where $\underline{W} = [W_1, W_2, ..., W_N]^T$ is a weighting vector.

b. The Itakura-Saito distortion measure

The Itakura-Saito distortion measure is used in speech coding applications, where each code-book vector is a representation of a trained LP short-term spectral model. The measure is applied to vectors of LP ladder coefficients,

$$[a_1, a_2, \cdots, a_N]^T \tag{4.11}$$

where each a_i is the ith LP coefficient of a Pth order all-pole ladder filter

The Itakura-Saito measure of the distortion from a given vector $\underline{x}$ to a code-book vector $\underline{\gamma}$, i.e. a member of Γ, is defined as [51],

$$d_I(\underline{x}, \underline{\gamma}) = (\underline{x} - \underline{\gamma})^T \mathbf{R}_x (\underline{x} - \underline{\gamma}) \tag{4.12}$$

where $\mathbf{R}_x$ is a N×N normalised autocorrelation matrix corresponding to the code-book vector $\underline{\gamma}$. The matrix takes the form,

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(N-1) & r(N-1) \\ r(1) & r(0) & \cdots & r(N-2) & r(N-2) \\ r(2) & r(1) & \cdots & r(N-3) & r(N-3) \\ \vdots & \vdots & & \vdots & \vdots \\ r(N-1) & r(N-2) & \cdots & r(1) & r(0) \end{bmatrix}$$

Each r(i) is the normalised autocorrelation value,

$$r(i) = \frac{R(i)}{R(0)} \tag{4.13}$$

with R(i) is equal to the ith sample of the autocorrelation function corresponding to the vector $\underline{\gamma}$.

4.3.2 Code-book training using LBG algorithm

An important aspect in the effectiveness of a vector quantiser is the suitability of the code-book, i.e. whether there are sufficient code-book vectors and whether they are well placed in vector space. To achieve a satisfactory placement of code-book vectors a training procedure must be employed which requires a large set of typical vectors, referred to as training vectors, to be provided. The code-book is therefore trained by a procedure referred to as a clustering process applied to the set of training vectors. Generally, the robustness of a code-book will depend on the size of the training set and also on how representative of normal data the training vectors are. It was suggested [51] that a sufficiently long and representative training sequence should be used such that ideally the performance of the quantiser on new data produced by the same source should be roughly that achieved on the training data. Hence the more aspects of typical speech that the training vectors cover, for instance male voice, female voice, street environment, office environment and so on, the better will be the performance of the code-book. It was proposed [53] that in order to have a reliable quantiser, the ratio of the number of training vectors to the number of code-book vectors required should be at least ten and preferably much greater; up to 50 ideally.

The LBG clustering algorithm [54] is an iterative process for training vector quantiser code-books using some general distance or distortion measure. The algorithm arranges or "clusters" a given set of training vectors into a specified number of different groups referred to as cells. All the vectors in the same cell are considered to be close enough to a given "label" vector, such that they may be replaced by the label vector without causing excessive distortion. The label vector is the centroid of the cell defined as the center of gravity of the cell. The allocation of a training vector to an particular cell is carried out according to a distortion measure from the training vector to the label vector. The distortion measure may be a simple mse distance measure with perceptual weighting. The LBG algorithm is usually employed in conjunction with a centroid splitting (CS) procedure which allows the number of cells to be increased in stages to the required number. The resulting algorithm is referred to as LBG-CS. Suppose a L-size N-dimensional code-book Γ must be

generated using a set of N-dimensional training vector $\{\underline{t}_j\}_{1,\ T}$. The LBG-CS procedure for code-book training is as follows:

i) The entire training set is treated as a single cell and a label vector for the cell is computed. This label vector is required to be the centroid of the cell. The centroid is defined to be a vector such that the sum of the individual values of the distance or distortion measure from the centroid to each of the training vectors in the cell is minimum. In the case of the mse measure, the centroid will be the mean of all the training vectors in the cell, i.e. the centroid will be,

$$\underline{\gamma}_i = \frac{1}{T} \sum_{j=1}^{T} \underline{t}_j \tag{4.14}$$

where

$\underline{t}_j$ is the jth training vector

$\underline{\gamma}_i$ is the label vector in the code-book

T is total number of training vectors

ii) Split each label vector into two by multiplying it by a scalar close to unity, referred to as the splitting factor χ. The two vectors are the original label vector and its scaled version. Refer to these now as new label vectors.

iii) Define a cell for each new label vector by assigning each training vector to the new label vector for which the distortion from the new label vector to the training vector is minimum.

iv) Calculate the centroid for each of the new cells.

v) Optimise the arrangement of cells and their label vectors using an iterative process. This is carried out by redefining the new label vectors to be equal to the centroids just calculated and repeating steps (iii), (iv) and (v). The iteration continues until the relative change in total distortion at each iteration is reduced to an acceptable small level, i.e. until,

$$\left| \frac{\Xi - \Xi'}{\Xi} \right| \leq \varepsilon \tag{4.15}$$

with,

$$\Xi = \frac{1}{T}\sum_{j=1}^{T} \min_{i}\left[d\left(\underline{\gamma}_i, \underline{t}_j\right)\right] \tag{4.16a}$$

$$\Xi' = \frac{1}{T}\sum_{j=1}^{T} \min_{i}\left[d\left(\underline{\gamma}'_i, \underline{t}_j\right)\right] \tag{4.16b}$$

where

$\underline{t}_j$ is the jth training vector

$\underline{\gamma}_i$ and $\underline{\gamma}'_i$ are the new label vectors for the current and previous iterations

Ξ and Ξ' are the total distortion measures in the current and previous iterations.

ε is the predefined distortion threshold, and is typically set to 0.001

vi) Once the iteration process defined by steps (iii) to (iv) has been completed, a code-book has been designed with code-book vectors equal to the centroids of the set of cells. The set of label vectors are equal to these centroids. The code-book size starts at 2 and can now be doubled by going back to step (ii) which splits each label vector into two before the iterative process (iii) to (iv) is repeated for the increased number of cells. Therefore go back to step (ii) until the required code-book size is achieved.

4.3.3 Quantiser structures

Once the vector quantiser has been trained it can be used to quantise any given input vector of the correct dimension. To apply the vector quantiser, the input vector is replaced by the closest available code-book vector. The index of the chosen code-book vector is the coded quantised vector.

To find the closest available code-book vector, the input vector is compared with each individual reference vector in the code-book using a distance or distortion measure. This is referred to as a full-search VQ. A full-search VQ is computational costly when the size of the code-book is large, say a 12-bit codebook (4096 levels). To reduce the computational costs of the vector quantiser, alternative code-book arrangements must be used. Many code-book arrangements have been studied [51]-

[53], and it can be concluded that the choice of the code-book arrangements is a trade-off between quantiser performance and computational complexity. Three simple code-book arrangements will be introduced in the this section, they are known as split VQ, multi-stage VQ and interframe VQ.

Split VQ

In a split vector quantiser, each input vector is required to be partitioned into a number of sub-vectors. A separate code-book exists for each of these sub-vectors. These code-books are trained separately. When the number of split code-books is equal to dimension of the vector, i.e. when there is a code-book for each vector element, the quantiser becomes a scalar quantiser. The schematic diagram of a split vector quantiser using three code-books is shown in figure 4.1.

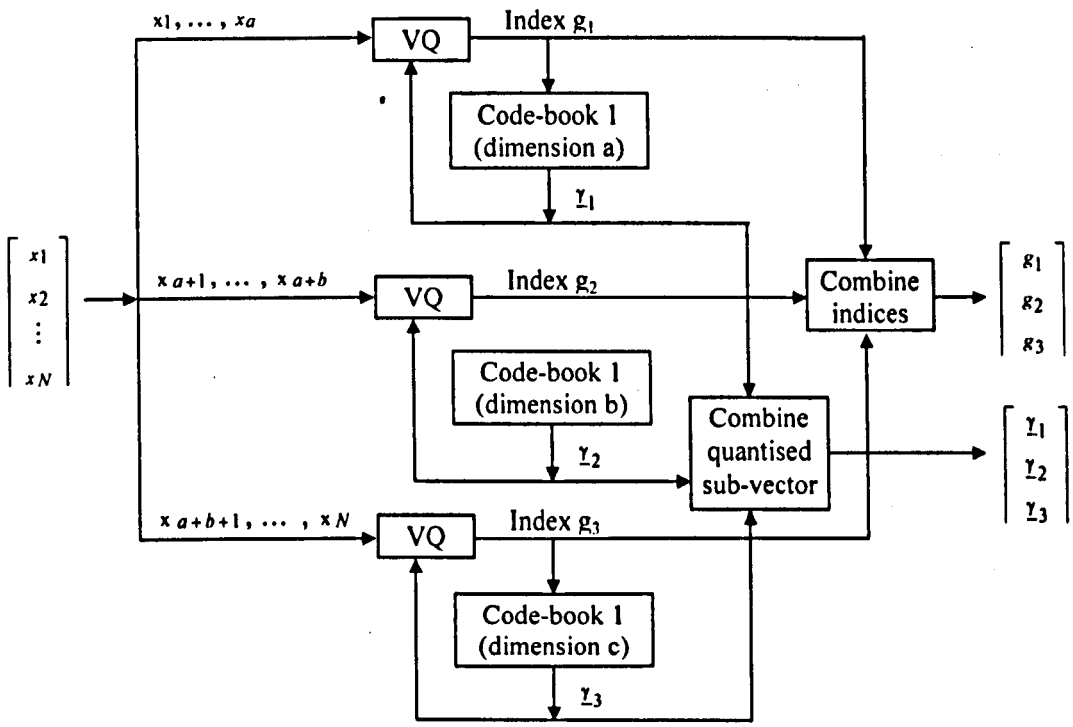


Figure 4.1 Schematic diagram of a split VQ using 3 code-books.

The N-dimensional input vector is partitioned into 3 sub-vectors with dimensions a, b and c where a+b+c=N. Each sub-vector is applied to a VQ algorithm with an appropriately trained code-book. Three code-book vectors $\underline{y}_1$, $\underline{y}_2$ and $\underline{y}_3$ with indices g_1, g_2, and g_3 respectively must be found for the sub-vectors. Once the optimal

vector in each code-book has been found, the three code-book vectors are recombined to form the quantised vector, i.e.

$$\hat{\underline{x}} = \begin{bmatrix} \underline{\gamma}_1 \\ \underline{\gamma}_2 \\ \underline{\gamma}_3 \end{bmatrix} \tag{4.17}$$

To code this quantised vector, a vector index $[g_1, g_2, g_3]^T$ is required. This enables the quantised sub-vectors γ_1, γ_2 and γ_3 to be read from identical copies of the three code-books at the decoder. For a given vector dimension and available bit-rate, the computational complexity of a split vector is approximately inversely proportion to the number of sub-vectors used. The quantiser performance deteriorates as the number of sub-vectors increased. A suitable compromise must be made between the complexity and the performance of the vector quantiser.

Multi-stage VQ

A multi-stage vector quantiser has the advantage of being able to reduce both the computational complexity and memory requirement of a vector quantiser with a relatively small decrease in performance [53]. In a multi-stage quantiser, an input vector of dimension N is quantised according to a main code-book of dimension N and a number of error code-books each of dimension N. The first error code-book is used to quantise the difference between the input vector and the chosen entry from the main code-book. The second error code-book is used to model the difference between the input to the first error stage and its quantised version, and so on for the third and any subsequent error code-books. In figure 4.2, the schematic diagram of a 3-stage vector quantiser is shown.

The input vector $\underline{x}$ is applied to the first-stage quantiser to obtain the optimal vector γ_1 with index g_1. The difference between $\underline{x}$ and γ_1 is then computed and applied to the second-stage quantiser to obtain the first error vector γ_2 and its index g_2. Now the difference between $\underline{x}$-γ_1 and γ_2 is computed and applied to the third stage quantiser to obtain the second error vector γ_3 and its index g_3. As a result, the quantised vector $\hat{\underline{x}}$ is obtained by adding the contributions from the three code-books, i.e.

$$\hat{\underline{x}} = \underline{\gamma}_1 + \underline{\gamma}_2 + \underline{\gamma}_3 \tag{4.18}$$

To encode $\hat{\underline{x}}$ a vector of the three indices g_1, g_2, and g_3 is required which allows γ_1, γ_2 and γ_3 to be read from identical copies of the 3 code-books at the decoder

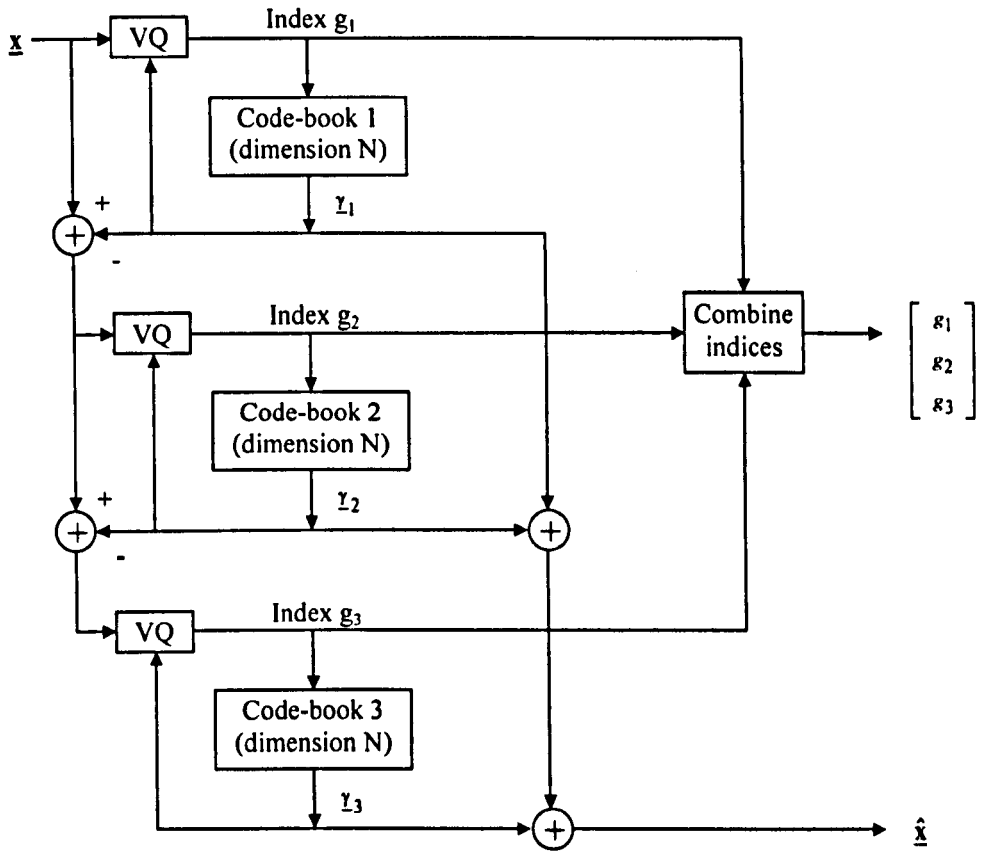


Figure 4.2 Schematic diagram of a multi-stage vector quantiser using 3 code-books.

Interframe VQ

Interframe vector quantisation, which aims to exploit correlation between successive input vectors, can reduce the bit-rate requirement of a vector quantiser. In the interframe quantisation scheme shown in figure 4.3, the contribution of the previous quantised vectors is subtracted from the input vector and the difference vector is quantised. At the decoder, the quantised vector is reconstructed using equation 4.19.

$$\hat{\underline{x}}^{(l)} = \hat{\underline{x}}^{(l-1)} + \underline{\gamma} \tag{4.19}$$

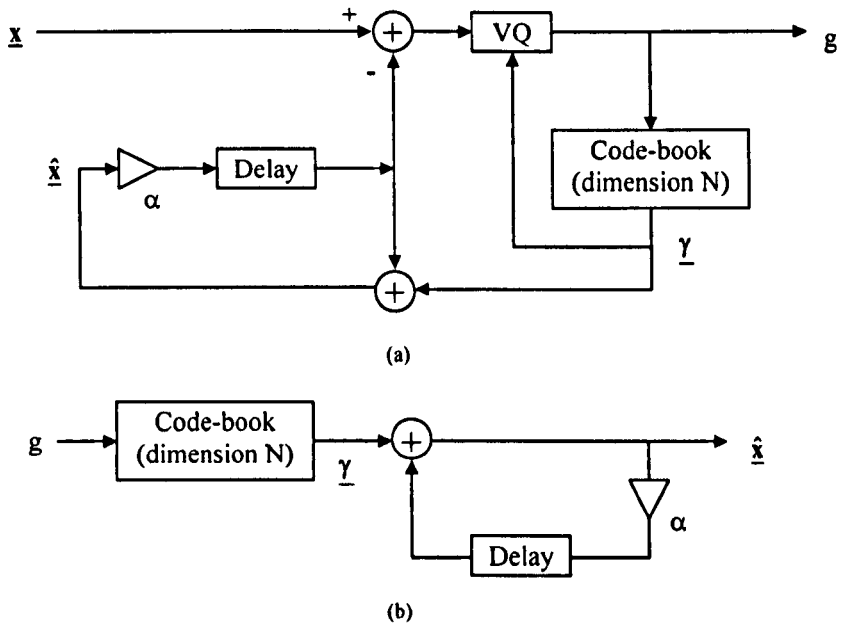


Figure 4.3 Schematic diagram of an interframe vector quantiser.
(a) encoder (b) decoder

A leakage factor α is included to scale the weighting of the previously quantised vector. The value of α is, generally, just less than one.

4.3.4 Objective assessment of a quantiser for short-term spectral coefficients

The performance of a vector quantiser for short-term spectral coefficients may be assessed by calculating the average spectral distortion measure $\overline{D}$ for a large set of typical and representative input vectors. The average spectral distortion measure $\overline{D}$ is defined as

$$\overline{D} = \frac{1}{M} \sum_{m=1}^{M} D_m \tag{4.20}$$

where the spectral distortion measure D_m is defined in equation 3.42 for a set of input vectors $\underline{x}_i$ for i=1, 2, ...N.

A short-term spectral coefficient quantiser is often considered to be spectrally transparent if it is able to achieve an average spectral distortion of less than about 1dB [57]. However, perceptible distortion may still occur in speech which has less than 1dB average spectral envelope distortion when there are occasional frames with high value of D_m [57]. This means that the statistical behaviour of a quantiser must

also be considered. It is suggested [57] that the performance of a quantiser can be evaluated using an average spectral distortion measure $\overline{D}$ and the percentage of two categories of statistical outliers. The two categories of statistical outliers are defined as,

i) the measurement of D_m with $2dB \leq D_m \leq 4dB$

ii) the measurement of D_m with $D_m > 4dB$

A quantiser is considered likely to achieve spectral transparent quality if the following three conditions are fulfilled [57]:

i) The average distortion $\overline{D}$ is about 1dB

ii) The percentage of testing vectors having $2dB \leq D_m \leq 4dB$ is less than 2%

iii) None of the testing vectors have $D_m > 4dB$

4.4 A 24-bit LSF quantiser using multi-stage split VQ

As applied to speech coding in general, VQ may be used for several different purposes. It is commonly used for coding the parameters which represent the short-term spectral envelope and also for coding segments containing time- or frequency-domain samples of speech or LP residual. CELP coders are strongly based on these ideas. We now consider the design of a scheme for the vector quantisation of LSF coefficients which will represent an LP spectral envelope at each update-point in a waveform interpolation coder.

A 24-bit ten-dimensional vector quantiser has been proposed for coding the 10 LSF coefficients of a 10th order LP synthesis filters. Such a quantiser has been designed by combining the two techniques of multi-stage and split VQ as described in the previous sections. The resulting LSF quantisation technique is referred to as MS-VQ. In this vector quantiser, the first stage is a full 10-dimensional vector quantiser which is populated by 1024 LSF vectors, requiring 10 bits. A10-dimensional quantised LSF vector is then obtained by finding the best matched vector to the input vector from the 10-dimensional codebook. Two 5-dimensional

vector quantisers are used at the second stage to quantise the difference between the best matched LSF vector from the first-stage codebook and the input vector, using 7-bit (size 128) code-books.

In this section, the procedures used to obtain the required LSF quantiser will be presented. The section starts by examining the effect of various weighting factors on the first stage of the quantiser. The performance of the quantiser is improved by introducing the second-stage code-books to encode the difference between the target vector and the optimum vector from the first code-book. A re-ordering process is employed in searching the second-stage code-books to preserve the interlacing property of the LSF's in the quantised vector. This is done to ensure the stability of the quantised all-pole filter.

An LSF training set containing 15000 ten-dimensional training vectors of LSF coefficients for voiced speech only was used in experiments performed to train the codebooks for a vector quantiser. The training vectors were generated by applying 10th order LP analysis to frames of voiced speech extracted from the speech file "GSP.DAT" [20]. The input speech signal was segmented into variable length frames centered on update point at intervals of 20ms. The two-way pitch detector described in chapter 2 was used to determine the nature of the input speech frame around each update-point. If voiced speech was indicated, Burg's pitch-synchronous LP analysis was performed on the variable length speech frame as described in chapter 3 to obtain a set of 10 LP ladder filter coefficients. A 10th order LP analysis was used. In case of unvoiced speech, the speech frame is discarded. As recommended in [57] a 10Hz bandwidth expansion was applied to each LP pole by modifying each LP ladder filter coefficient as follows,

$$a_i = a_i \gamma^i \qquad i = 1, 2, \ldots, 10 \tag{4.21}$$

where γ is set as 0.996.

Each LSF vector was computed from the set of bandwidth expanded LP ladder filter coefficients using the iterative process described in Appendix B. All the code-books were trained with the LBG-CS algorithm [60] using,

$$\text{splitting factor } \chi = 0.99$$
$$\text{distortion threshold } \varepsilon = 0.0001$$

To evaluate the performance an LSF vector quantiser, once it had been trained, a new set of LSF vectors was used. The testing vectors were extracted from the speech file "OPERATOR.DAT" [21]. Different from the training set, the 1800 testing vectors contained both voiced and unvoiced speech. The distortion measurements discussed in section 4.3.4 were used to assess the performance of a vector quantiser.

4.4.1 Utilisation of weighting factors during code-book searching

It was suggested [57] that the performance of an LSF quantiser may be enhanced by introducing appropriate weighting factors during code-book training and searching. The commonly used weighting factors are LSF distance [58][10], power weighting [57] and frequency weighting [57]. In this section, the means of applying weighting factors is considered and their effect on the performance of an LSF quantiser is evaluated. The evaluation was carried out using the single stage 10-bit LSF quantiser shown in figure 4.4 which is the first stage of the MS-VQ.

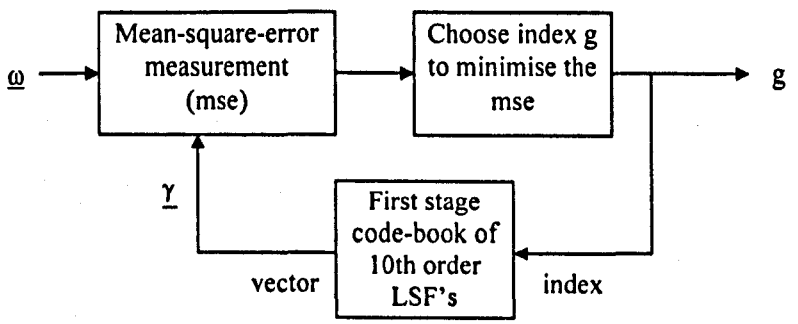


Figure 4.4 Schematic diagram for a single stage LSF vector quantiser.

The LSF code-book was first trained without weighting using the 15000 training vectors referred to earlier. It was then evaluated using 1800 test vectors, by code-book searching again without any weighting. For each test vector m in the range 0 to 1800, D_m was calculated, and the average $\overline{D}$ was calculated for the whole set of test vectors. The results of this evaluation are presented in table 4.1.

$\overline{D}$ (dB)	3.204
2-4 dB (%)	70.89
>4 dB (%)	21.00

Table 4.1 Distortion measure when only using the first stage of the 24bit MS-LSF quantiser without any weighting factor

To assess the effect of weighting the importance of some LSF coefficients more than others, a weighting vector $\underline{\mathbf{W}} = [W_1, W_2, ..., W_{10}]^T$ was introduced in the distortion measure as defined by equation 4.10 used for the code-book searching. The results obtained for different arrangement of weightings will be compared with those listed in table 4.1.

4.4.1.1 Weighting factors based on LSF distances

The first set of weighting factors under investigation here was suggested in the ITU recommendation for the 8kb/s CSA-CELP coder [10]. The weighing factors were derived from the distance of adjacent LSF's and are defined as,

$$W_1 = \begin{cases} 1.0 & \omega_2 - 0.04\pi - 1 > 0 \\ 10\left(\omega_2 - 0.04\pi - 1\right)^2 + 1 & otherwise \end{cases}$$

$$W_i = \begin{cases} 1.0 & \omega_{i+1} - \omega_{i-1} - 1 > 0 \\ 10\left(\omega_{i+1} - \omega_{i-1} - 1\right)^2 + 1 & otherwise \end{cases} \qquad i = 1, 2, \ldots, 9$$

$$W_{10} = \begin{cases} 1.0 & -\omega_9 + 0.92\pi - 1 > 0 \\ 10\left(-\omega_9 + 0.92\pi - 1\right)^2 + 1 & otherwise \end{cases} \tag{4.22}$$

The measurements of $\overline{D}$ and 2-4dB and > 4dB outliers obtained from a 10th order LSF VQ with these weighting factors are given in table 4.2,

$\overline{D}$ (dB)	3.319
2-4 dB (%)	68.39
>4 dB (%)	24.06

Table 4.2 Distortion measure when only using the first stage of the 24bit MS-LSF quantiser with the LSF distances as a weighting factor

Comparing these results with table 4.1, an increase of about 0.11dB in the average distortion has been obtained by introducing the weighting. However the total number of statistical outliers has been increased by more than 3%. The results suggest that LSF distance weighting introduced in this way does not necessarily improve the quantiser.

4.4.1.2 Weighting factors based on power weighting

A set of weighting factors W_i may be computed from the power (or energy) spectrum of the short-term spectral envelope of the speech segment under analysis. These "power" weighting factors are defined as [57],

$$W_i = \left[P(\omega_i) \right]^r \qquad i = 1, 2, \cdots, 10 \tag{4.23}$$

$P(\omega_i)$ is the power (or energy) at the frequency of the ith LSF coefficient ω_i of the short-term spectral envelope. The constant r is included to control the relative weighting given to the 10 LSF coefficients. Values of r in the range from 0.025 to 0.25 were tried and the results of these trials are listed in table 4.3

r	$\overline{D}$ (dB)	2-4 dB (%)	>4 dB (%)
0.025	3.168	72.17	19.61
0.050	3.154	72.83	18.83
0.075	3.147	73.28	18.44
0.100	3.153	73.39	18.39
0.125	3.173	72.83	19.17
0.150	3.204	71.39	20.78
0.175	3.248	70.17	22.28
0.200	3.291	68.50	24.06
0.225	3.353	66.83	25.89
0.250	3.419	64.72	28.11

Table 4.3 Distortion measure when only using the first stage of the 24bit MS-LSF quantiser with the range of values of r in the power weighting set from 0.025 to 0.25

The results in table 4.3 show that the power weighted vector quantiser performed best for values of r in the range 0.025 to 0.15. When r was larger than 0.15, the performance of the quantiser deteriorated rapidly. The optimum performance was obtained when $r = 0.075$.

4.4.1.3 Weighting factors based on power and frequency weighting

It was suggested [57] that the human perceptual system cannot resolve differences in high frequencies as accurately as at low frequencies. More weighting may be allocated to the lower LSF coefficients to increase their importance over the higher LSF coefficients. A weighting scheme which incorporates both power weighting and frequency weighting was proposed by Paliwal and Atal [57]. A value of $r = 0.15$ was suggested for the power weighting. Frequency weighting is allocated to the individual LSF coefficients as follows:

$$W_i = \begin{cases} 1.0 & \omega_1 \text{ to } \omega_8 \\ 0.8 & \omega_9 \\ 0.4 & \omega_{10} \end{cases} \qquad (4.24)$$

We tested this joint power and frequency weighting scheme by changing the value of the constant r in the power weighting. Values of r in the range 0.025 to 0.15 were tested. We chosen these values of r because they have given the best performance when the power weighting is used alone. The results of these trial are listed in table 4.4.

r	$\overline{D}$ (dB)	2-4 dB (%)	>4 dB (%)
0.025	3.273	70.56	22.28
0.050	3.272	70.28	22.39
0.075	3.271	70.22	22.33
0.100	3.298	69.44	23.22
0.125	3.323	68.50	24.43
0.150	3.357	67.72	25.22

Table 4.4 Distortion measure when only using the first stage of the 24bit MS-LSF quantiser with the joint power and frequency weighting for the range of values of r set from 0.025 to 0.15

Results in table 4.4 show that frequency weighting applied in this way did not improve the vector quantiser performance. The performance actually deteriorated when the frequency weighting was applied. The results indicate that the best performance of the first stage vector quantiser is likely to be obtained using a power weighting with $r = 0.075$.

4.4.2 Implementation of a re-ordering process during code-book searching

Using the power weighting with the value of r set to 0.075, a new code-book was trained for the first stage LSF quantiser. The optimum power factor r was used for the code-book training as well as the code-book evaluation. The measurements of $\overline{D}$ and the statistics of D_m are presented for the new quantiser in table 4.5. Table 4.5 also lists the measurements obtained without power weighting (as in table 4.1) for comparison.

	Code-book training methods	
	without the power weighting	with the power weighting
$\overline{D}$ (dB)	3.204	3.098
2-4 dB (%)	70.89	75.78
>4 dB (%)	21.00	16.50

Table 4.5 Distortion measure when only using the first stage of the 24bit MS-LSF quantiser with the code-book being trained and searched with and without the power weighting

Comparing the two set of results, about 0.1dB improvement in the average spectral distortion is obtained when the power weighting is introduced. Considering the statistics of the outliers, the percentage of the >4dB outliers is reduced by 4.5% using the power weighting. All of these >4dB outliers have been moved to the 2-4dB category.

After the weighting factors had been chosen and the first stage implemented, a 24-bit LSF vector quantiser was completed by introducing the second-stage split vector quantiser with code-books of LSF differences ΔLSF. Two 7-bit code-books were used, each of them populated by 5-dimensional vectors for ΔLSF's 1-5 and 6-10 respectively. To collect the training vectors for the split VQ code-books, each vector in the training set was quantised according to the first-stage code-book and subtracted from the optimal vector. The two code-books were trained without any weighting. The schematic diagram of the 24-bit MS-LSF quantiser is shown in figure 4.5. Note that the multi-stage VQ scheme discussed in section 4.3.3 has been modified when it is implemented in the 24-bit MS-LSF quantiser. This is because the interlacing property of the LSF's in a quantised LSF vector may not be preserved by simply quantising the differences, between the input LSF vector and the

optimum LSF vector from the first-stage code-book, at the second stage. The details of the 24-bit MS-LSF quantiser are described as follows.

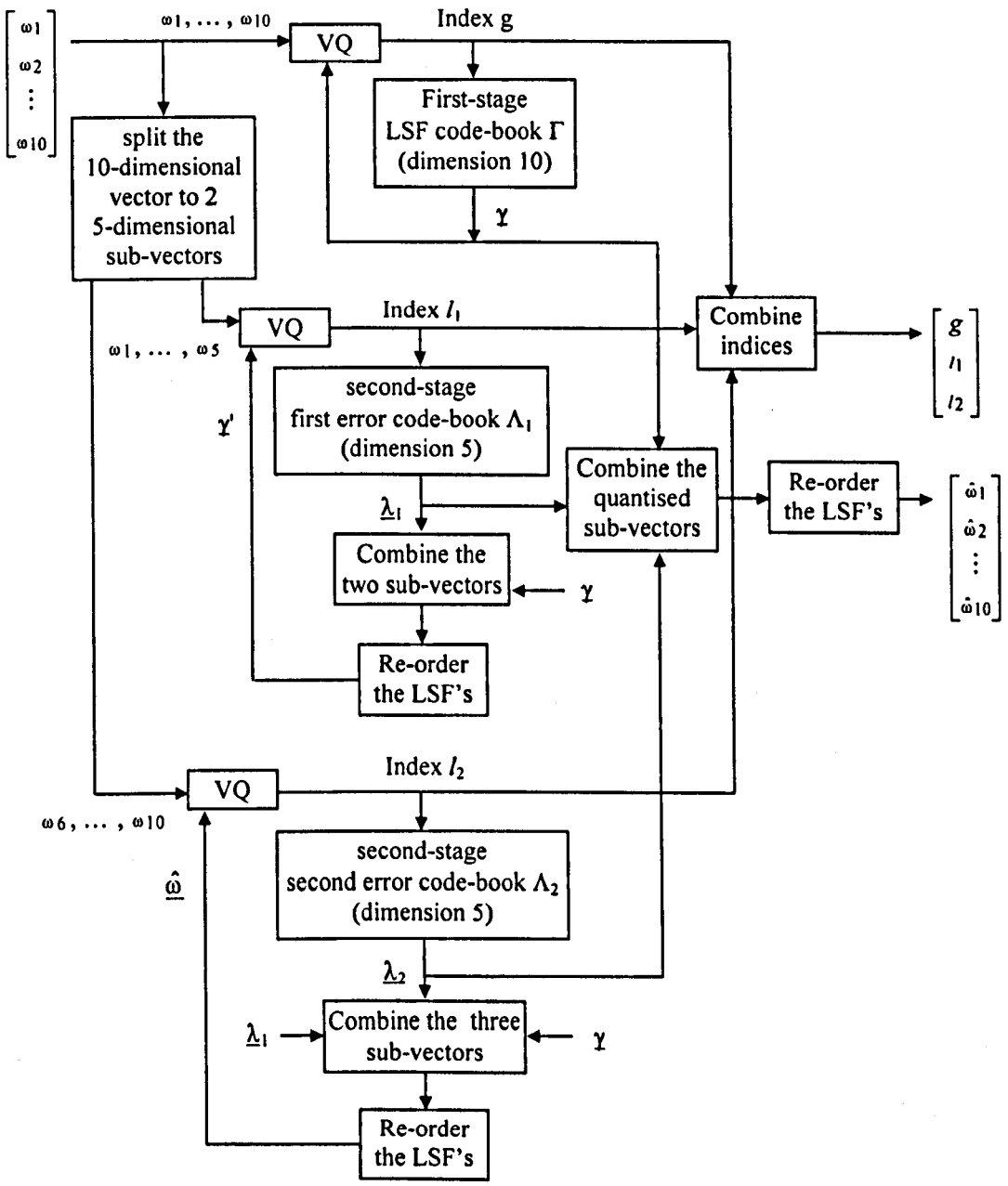


Figure 4.5 Schematic diagram for a 24-bit MS-LSF vector quantiser (MS - multi-stage split).

To implement the vector quantisation procedure, the first stage LSF code-book Γ is searched to find the best matched 10-dimensional vector $\underline{\gamma}$, using a power weighted mse measure. The first of the two second-stage code-books Λ_1 and Λ_2 is now searched to find the 5-dimensional vector $\underline{\lambda}_1$ of Λ_1 which, when combined with

the first five elements of $\underline{\gamma}$, produces a vector $\underline{\gamma}'$ which minimises the power weighted mse measure over all possible choices of $\underline{\lambda}_1$. By "combined with" we mean simply that the five elements of $\underline{\lambda}_1$ are added to the corresponding first five elements of $\underline{\gamma}$. A problem that must be addressed is that the resulting vector $\underline{\gamma}'$ will not necessarily preserve the interlacing property (see section 3.4.1) of LSF's which is necessary to make the all-pole LP system filter stable. This interlacing property that must be satisfied is that:

$$0 < \omega_1 < \omega_2 < \ldots < \omega_{10} < \pi$$

where LSF coefficients ω_1, ω_3, ω_5, ω_7, ω_9 are the frequencies of the zeros of P(z) and ω_2, ω_4, ω_6, ω_8, ω_{10} are the frequencies of the zeros of Q(z) with P(z) and Q(z) defined in equations 3.27a and b respectively.

A re-ordering process must therefore be used sometimes to re-arrange the LSF's in the vector $\underline{\gamma}'$, such that the LSF's are correctly interlaced. This re-ordering, which is described later, is applied at the encoder to produce the required test vector and in the same way at the decoder. After the optimal vector from the first of the second-stage codebooks has been found and therefore the best possible vector $\underline{\gamma}'$ is known, the second Λ_2 of the two second-stage code-books is searched to find the 5-dimensional vector $\underline{\lambda}_2$ of Λ_2 which, when combined with the second five elements of $\underline{\gamma}'$ produces a vector $\underline{\hat{\gamma}}$ which minimises the power weighted mse measure over all possible choices of $\underline{\lambda}_2$. Again a re-ordering process is sometimes necessary to preserve the required ordering property of LSF's. Thus a quantised LSF vector $\underline{\hat{\gamma}}$ is composed of contributions from the three code-books. Each element of $\underline{\hat{\gamma}}$ is equal to,

$$\hat{\gamma}_i = \begin{cases} \gamma_i + \lambda_{1i} & 1 \le i \le 5 \\ \gamma_i + \lambda_{2i-5} & 6 \le i \le 10 \end{cases} \tag{4.25}$$

with some re-ordering when necessary. The final quantised vector is,

$$\underline{\hat{\omega}} = re-order\left\{\underline{\hat{\gamma}}\right\} \tag{4.26}$$

4.4.2.1 A simple LSF re-ordering process

A possible re-ordering process is simply to re-arrange the elements of the quantised LSF vector such that they are in an ascending order. The results obtained using this method are:

$\overline{D}$ (dB)	1.587
2-4 dB (%)	18.50
>4 dB (%)	1.89

Table 4.6 Distortion measure for the 24-bit MS-LSF quantiser incorporating a simple LSF re-ordering process

The problem associated with this simple re-ordering process is that the distance between adjacent LSF's in a re-arranged LSF vector can be very small. This results in a sharp speech formant and hence the decoded speech may sound metallic [57].

4.4.2.2 An LSF re-ordering process designed from the statistics of the training vectors

The re-ordering process in section 4.4.2.1 may be improved by imposing restrictions on the minimum distances between adjacent LSF's in a re-arranged vector as has been done on the 8kb/s CSA-CELP [10]. To test this idea, the minimum LSF distances were determined from the statistics of the set of 15000 LSF training vectors. In table 4.7, the mean values $\overline{\omega}_i$ for each LSF element are presented, together with the minimum LSF ω_{min} and maximum LSF ω_{max} in set of 15000 LSF training vectors. In addition, the minimum distances between adjacent LSF's $\delta\omega_{i,j}$, with j=i+1 for i=1, 2, ..., 9, over the set of 15000 LSF training vectors are listed in table 4.8.

$$\omega_{min} = 0.059 \quad , \qquad \omega_{max} = 3.055$$

i	1	2	3	4	5
$\overline{\omega}_i$	0.175	0.284	0.543	0.840	1.206
i	6	7	8	9	10
$\overline{\omega}_i$	1.484	1.838	2.100	2.433	2.667

Table 4.7 The mean LSF of each vector element for the 15000 training vectors

i, j	1,2	2,3	3,4	4,5	5,6	6,7	7,8	8,9	9,10
$\delta\omega_{i,j}$	0.009	0.013	0.011	0.015	0.017	0.028	0.017	0.025	0.015

Table 4.8 Minimum LSF distance between adjacent LSF's for the 15000 training vectors

Based on the results in table 4.8, a modified re-ordering process may be proposed as follows,

$$\hat{\omega}_1 = \begin{cases} \hat{\gamma}_1 & \hat{\gamma}_1 \geq 0.059 \\ 0.059 & \text{otherwise} \end{cases}$$

$$\hat{\omega}_i = \begin{cases} \hat{\gamma}_i & \hat{\gamma}_i \geq \hat{\gamma}_{i-1} + \delta\omega_{i-1,i} \\ \hat{\gamma}_{i-1} + \delta\omega_{i-1,i} & \text{otherwise} \end{cases} \qquad (4.27)$$

$$i = 2, 3, \cdots, 10$$

$$\hat{\omega}_{10} = \begin{cases} \hat{\gamma}_{10} & \hat{\gamma}_{10} \leq 3.055 \\ 3.055 & \text{otherwise} \end{cases}$$

It is assumed that the vector $\underline{\hat{\gamma}}$ has first been pre-processed by the simple re-ordering process mentioned above, i.e. the LSF's in the vector $\underline{\hat{\gamma}}$ have been arranged in ascending order. The results obtained using this methods were,

$\overline{D}$ (dB)	1.587
2-4 dB (%)	18.44
>4 dB (%)	1.89

Table 4.9 Distortion measure for the 24-bit MS-LSF quantiser by restricting the minimum LSF distance in an LSF test vector according to the statistics of the 15000 training vectors

Comparing the results in table 4.9 with those in table 4.6 for simple re-ordering, not much improvement has been obtained by imposing the minimum LSF distances quoted in table 4.8.

4.4.2.3 An LSF re-ordering process using a fixed LSF distance

Instead of defining the minimum allowed distance between adjacent quantised LSF's according to the experimentally derived data in table 4.8, the use of a fixed minimum LSF difference $\delta\omega$ was considered. A number of possible values for this minimum difference $\delta\omega$ in the range 10Hz to 100Hz were tried and the results of this investigation are presented in table 4.10.

δω (in Hz)	$\overline{D}$ (dB)	2-4 dB (%)	>4 dB (%)
10	1.587	18.50	1.8 89
20	1.587	18.50	1.8 89
30	1.586	18.44	1.8 89
40	1.585	18.33	1.8 89
50	1.584	18.06	1.8 89
60	1.582	18.00	1.7 78
70	1.579	18.22	1.8 33
80	1.579	18.50	1.7 78
90	1.580	18.33	1.7 22
100	1.582	18.06	1.7 22

Table 4.10 Distortion measure for the 24-bit MS-LSF quantiser by restricting the minimum LSF distance in an LSF test vector to a fixed value

It is seen in table 4.10 that not much improvement over the results in table 4.9 was obtained when a small value of δω was used, e.g. 10Hz and 20Hz. The average spectral distortion and the percentage of statistical outliers decreases very slightly when δω is increased, from 20Hz to 60Hz. The average spectral distortion then increases again if δω is increased from 60Hz. It is interesting to observe that the performance of the quantiser appears to be similar to that obtained with simple re-ordering when δω=100Hz. However, the distances between adjacent LSF's are directly related to the nature of formant peaks [48]. The closer an adjacent pair of LSF's are around a particular formant frequency, the sharper will be the formant peak. This suggests that high values of δω may not be suitable since prominent formant peaks may not be well represented. Therefore a fixed minimum difference of 60 Hz (0.047rads) was chosen for the quantiser.

4.4.3 Conclusions of section 4.4

Three forms of weightings have been examined using the first stage of the 24-bit MS-LSF quantiser. The weighting factors were LSF distance weighting, power weighting and frequency weighting. Experimental results suggest that a quantiser performance may be enhanced using the power weighting defined in equation 4.23 with the values of r set in the range 0.025 to 0.15. Optimal performance was obtained when $r = 0.075$. LSF distance and frequency weighting

did not appear to offer the same advantages as power weighting. Therefore power weighting with $r = 0.075$ has been chosen for the LSF quantiser used in this thesis.

It must be pointed out that the combination of power and frequency weighting scheme has only been tested objectively. In order to have a clear picture about the perceptual effect of the frequency weighting scheme, subjective testing may need to be included.

In summary, a 24-bit LSF quantiser has been designed using the MS-VQ. The first stage of the quantiser employs a 10-bit 10-dimensional LSF code-book. Experiments showed that the quantiser performance is enhanced by using a power weighting in both code-book training and searching. Two 7-bit split code-books are used at the second-stage. These code-books contain 5-dimensional difference vectors and are trained without any weighting. To search for optimal vectors, a re-ordering process is used to maintain the ordering property of the LSF's in an LSF vector. In the re-ordering process, the value of the first LSF in a quantised LSF vector is restricted to be larger than 0.059rads and the value of the 10th LSF is restricted to be less than 3.055rads. In addition, a fixed minimum LSF distance of 60Hz (0.047rads) is imposed between the intermediate LSF's of the quantised LSF vector.

4.5 A 24-bit LSF quantiser using interframe multi-stage split VQ

It was demonstrated in section 4.4.2 that the performance of a quantiser could be enhanced by introducing second-stage code-books, which aim to make up the difference between the target vector and the optimal vector from the first-stage codebook. It is possible that performance of the LSF quantiser may be further improved by introducing third stage code-books. However, this would increase the system bit-rate. The function of the third stage code-books may be imitated by using interframe quantisation. In this section, the performance of the 24-bit MS-LSF quantiser discussed in section 4.4 is further improved by introducing an interframe quantisation scheme to yield a 24-bits IMS-LSF vector quantiser [61]. The 24-bit

IMS-LSF quantiser is able to achieve an almost spectral transparent quality. Finally, different code-book arrangements are examined for the 24-bit IMS-LSF quantiser.

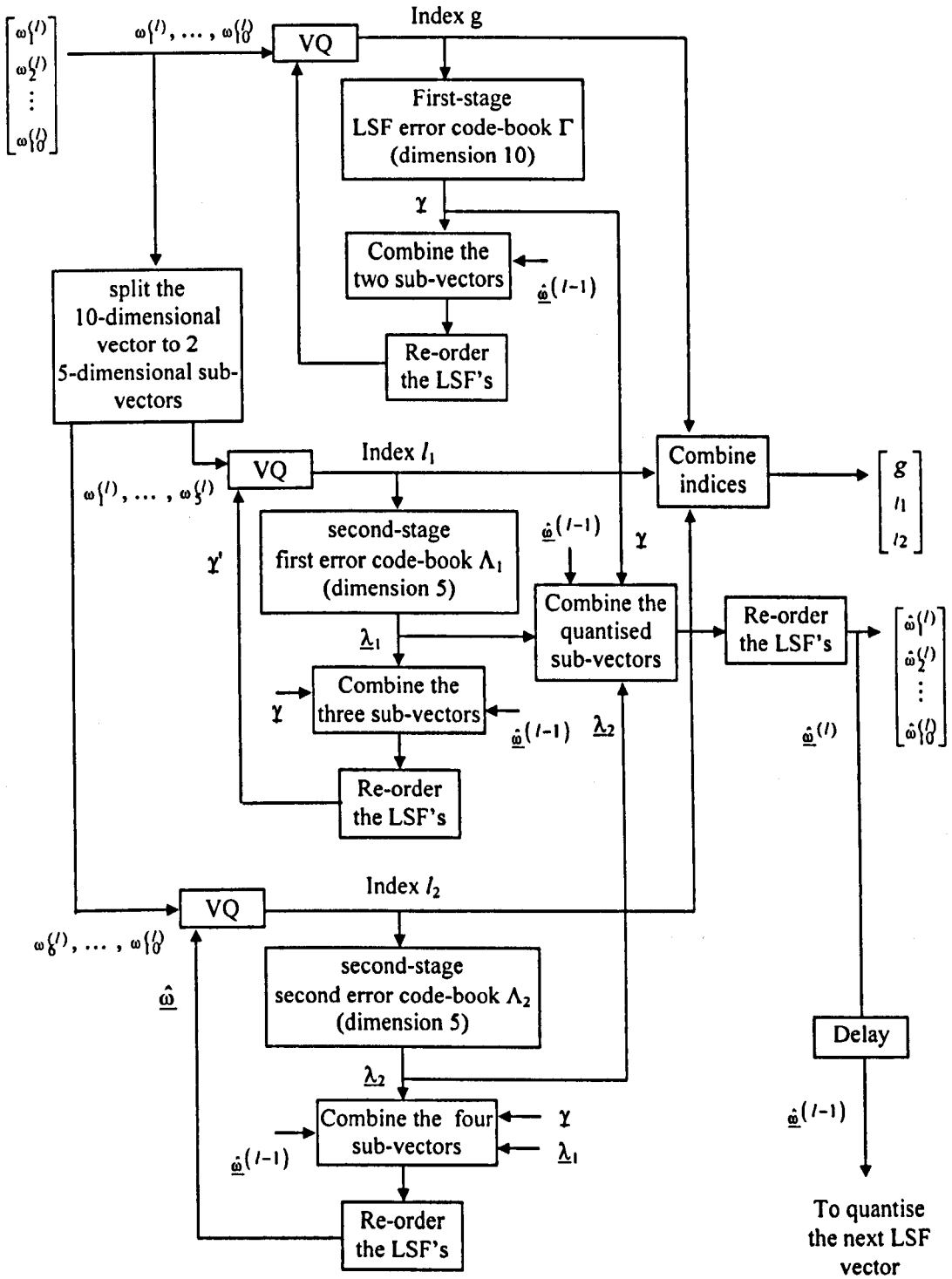


Figure 4.6 Schematic diagram for the 24-bit IMS-LSF vector quantiser (IMS - interframe multi-satge split).

The schematic diagram of the 24-bit IMS-LSF vector quantiser is shown in figure 4.6. The structure of the IMS-LSF quantiser is very similar to the MS-LSF quantiser. Instead of directly quantising the 10 LSF coefficients at the first stage, the difference between the previously quantised LSF vector and the input vector is quantised. To search the optimum vector $\underline{\gamma}$ from the first-stage codebook, the previously quantised LSF vector $\underline{\hat{\omega}}^{(l-1)}$ is combined with each vector from the first-stage code-book Γ to form a test vector $\underline{\gamma}'$. The vector $\underline{\gamma}'$ is compared with the input LSF vector using the power weighted mse measure. The vectors $\underline{\hat{\omega}}^{(l-1)}$ and $\underline{\gamma}$ are then evenly split into two sub-vectors and the same procedures as is done in the 24-bit MS-LSF quantiser is carried out to search the two second-stage code-books. Thus the quantised LSF vector $\underline{\hat{\omega}}^{(l)}$ is,

$$\underline{\hat{\omega}}^{(l)} = re-order\left\{\underline{\hat{\gamma}}^{(l)}\right\}$$

with,

$$\hat{\gamma}_i^{(l)} = \begin{cases} \hat{\omega}_i^{(l-1)} + \gamma_i + \lambda_{1i} & 1 \le i \le 5 \\ \hat{\omega}_i^{(l-1)} + \gamma_i + \lambda_{2i-5} & 6 \le i \le 10 \end{cases} \tag{4.28}$$

The training vectors for the first-stage code-book were obtained by subtracting the adjacent LSF vectors in the 15000 training vectors. The training data for the second-stage code-books were obtained by subtracting each of the 15000 training vectors from a quantised version of it, using the first-stage of the 24-bit IMS-LSF quantiser. The three code-books are trained without any weighting and the power weighting is only introduced during the code-book searching. Note that, the mean LSF's of the training vectors (listed in table 4.7) are used as the previous quantised vector when the quantiser is initialised.

4.5.1 Objective measurement of the 24-bit IMS-VQ

In searching the first-stage codebook when the two vectors $\underline{\hat{\omega}}^{(l-1)}$ and $\underline{\gamma}$ are combined, the result vector $\underline{\gamma}'$ may not be arranged in an ascending order. The LSF re-ordering process discussed in section 4.4.2 is assessed again for the IMS-LSF

quantiser and the results for different minimum LSF distance constraints δω are listed in table 4.11.

δω (in Hz)	$\overline{D}$ (dB)	2-4 dB (%)	>4 dB (%)
none	2.719	58.11	12.33
training statistical	2.722	57.78	12.72
10	2.756	58.89	13.28
20	2.727	56.44	13.28
30	2.715	56.06	12.89
40	2.694	58.28	12.00
50	2.679	55.50	12.67
60	2.698	58.17	11.56
70	2.683	58.39	11.56
80	2.661	56.28	12.28
90	2.648	58.39	10.89
100	2.645	56.94	10.78

Table 4.11 Distortion measures when using only the first-stage of the 24-bit IMS-LSF quantiser, for various minimum LSF distance constraints used in the re-ordering process.

Comparing to the results in table 4.5, an immediate improvement over the single-stage LSF quantiser is shown when the interframe scheme was introduced. A 0.4dB improvement in the average spectral distortion was obtained. The statistical outliers were reduced dramatically too. Considering the re-ordering process, results for various δω in table 4.11 are very much in harmony with those in tables 4.6, 4.9 and 4.10. In the IMS-LSF quantiser, the optimal performance was achieved when δω is 50 Hz (0.039rads).

After the optimal δω had been found, the two second-stage code-books were introduced to form the full 24-bit IMS-LSF quantiser. Finally, the tested results shown in table 4.12 suggest that the IMS-LSF quantiser is able to obtain almost transparent quality.

$\overline{D}$ (dB)	1.259
2-4 dB (%)	8.44
>4 dB (%)	0.22

Table 4.12 Distortion measures for the 24-bit IMS-LSF quantiser

4.5.2 Assessment of the LSF quantiser with various code-book arrangements

The 24-bit IMS-LSF quantiser was evaluated using 8_8^8 and 9_7^8 code-book arrangements. Here we use the notation A_{B2}^{B1} in which A is the number of bits in the first-stage code-book and B1, B2 are the numbers of bits in each of the second-stage code-books. Together with the original 10_7^7 formats, the three quantisers were also tested for a range of bit-rates from 18 bits to 24 bits. Note that the size of the first-stage code-book was unchanged during the experiment, i.e. 8-bit, 9-bit and 10-bit. The quantiser bit-rate was altered by adjusting the size of the second-stage code-books, in the way 10_4^4, 10_4^5, 10_5^5, 10_5^6 and so on. The experimental results are shown in figures 4.7a, b and c, corresponding to the average spectral distortion measures $\overline{D}$ and the percentage of the 2-4dB and >4dB statistical outliers respectively.

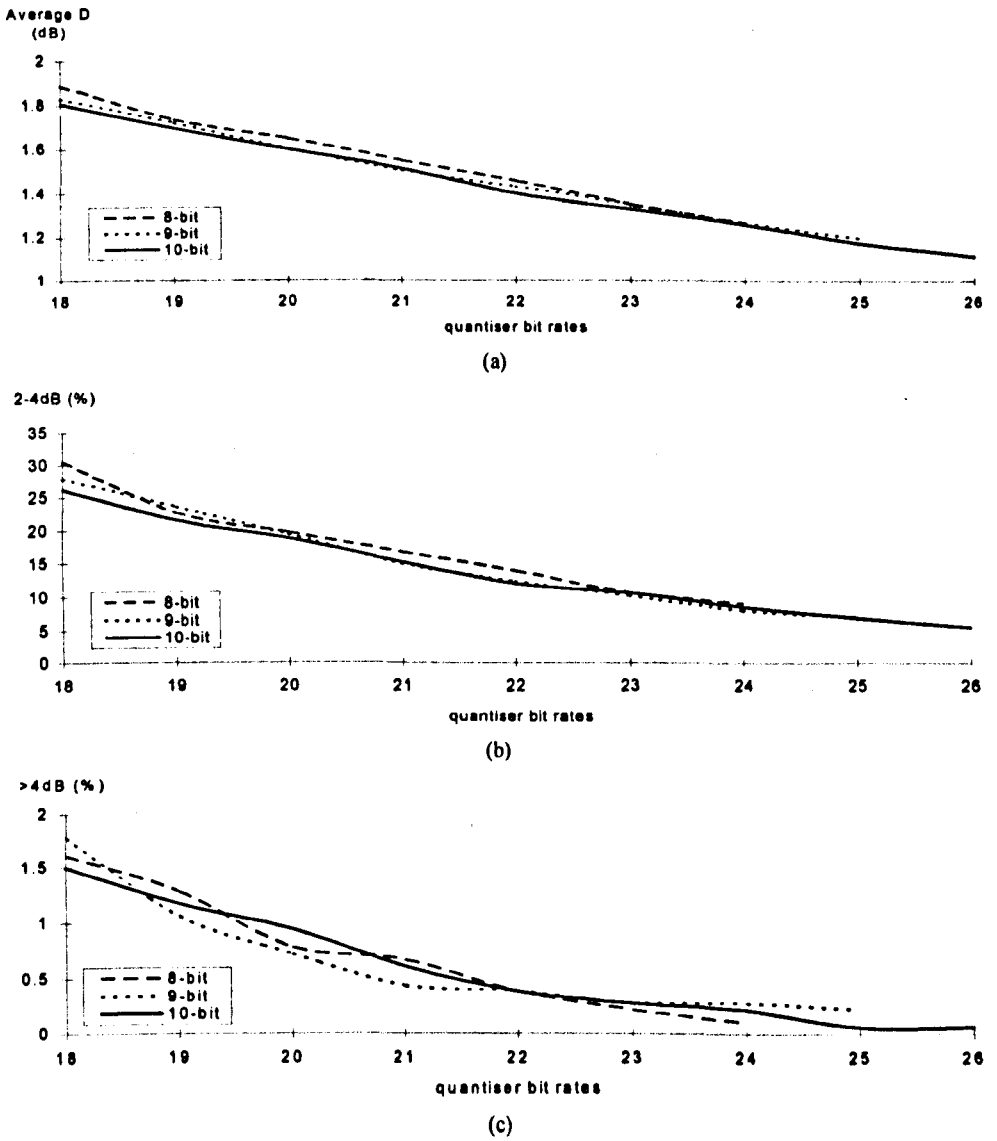


Figure 4.7 Performance of the IMS-LSF quantiser with different first-stage code-book sizes under different bit rates. (a) the average distortion measure (b) the percentage of 2-4dB outliers (c) the percentage of >4dB outliers

The results suggest that the quantiser may have a better performance by using a larger first-stage code-book. This was especially significant when the quantiser bit-rate was small. In figure 4.7a for an 18 bit quantiser, a 10_4^4 arrangement was about 0.08dB less in $\overline{D}$ than the 8_5^5 quantiser. In figures 4.7b and c, the statistical outliers obtained from the 10_4^4 quantiser is very much less than those obtained from the 8_5^5 arrangement. The performance of the three quantisers converged when the system bit rate was increased. The three quantisers performed comparably in a 24-bit arrangement. The 10_7^7 quantiser has the lowest average spectral distortion, whilst the 9_7^8 quantiser yielded the least statistical outliers. The difference in the $\overline{D}$ between the 10_7^7 and the 9_7^8 quantisers is about 0.01dB and the difference in the statistical outliers is about 0.5%. A 10_7^7 LSF quantiser was chosen as a candidate for the later speech coders. However computational savings might be obtained using the 9_7^8 or 8_8^8 arrangements.

4.5.3 Conclusions of section 4.5

An interframe quantisation scheme has been introduced into the 24-bit MS-LSF quantiser to form a 24-bit IMS-LSF quantiser. The IMS-LSF quantiser shows a distinct improvement over the MS-LSF quantiser. A fixed LSF distance of 50Hz is imposed between the intermediate LSF's of the quantised LSF vector. Various quantiser arrangements have been evaluated for the IMS-LSF quantiser for different quantiser bit-rates. Experimental results suggest that a larger first-stage code-book may be preferable when the quantiser bit rate is small. This restriction may be relaxed as the quantiser bit rate is increased. Finally a 24-bit LSF quantiser is designed using an interframe multi-stage split structure with a 10_7^7 arrangement.

4.6 Conclusions

Alternative representations of conventional LP ladder filter coefficients have been introduced. Line spectral frequencies are the most popular parameter for quantisation owing to their localised effect on the spectral envelope under quantisation noise. Various aspects in implementing a vector quantiser have been discussed. These include complexity consideration, code-book training, quantiser structure and performance assessment.

A 24-bits IMS-LSF quantiser which is able to achieve almost transparent quality has been proposed. The quantiser implements an interframe quantisation scheme incorporating a multi-stage split VQ, in which the difference between the current and the previous quantised LSF vector are quantised at each update-point. Experimental results suggested that the quantiser performance is improved by using a power weighting in the code-book searching. A re-ordering procedure is also designed for the quantiser. The objective of the re-ordering procedure is to preserve the interlacing property of the LSF's in an LSF vector and to ensure the stability of the quantised all-pole filter. In the re-ordering process, the first LSF is constrained to be larger than 0.059rads and the tenth LSF must be smaller than 3.055rads. In addition, a fixed minimum distance of 50Hz (0.039rads) is imposed on the intermediate LSF's.

Different code-book arrangements have been tested for the 24-bit IMS-LSF quantiser. The results suggest that a 9^8_7 or 8^8_8 arrangements are allowed in order to reduce the computational complexity of the quantiser.

Chapter 5

Two-mode Pitch-synchronous Waveform Interpolation (TPSWI) Model

5.1 Introduction

It has been reported [69] that the main source of degradation in a CELP coder during voiced speech comes from the insufficiently accurate reproduction of periodicity. Prototype Waveform Interpolation (PWI) coding was proposed in order to reinforce pitch-periodicity in the reconstructed voiced speech [68]. In an early form of PWI coder [70], the speech in the vicinity of each update-point is categorised as voiced or unvoiced. Voiced speech, being considered as a quasi-periodic signal which evolves slowly with time, is identified by the existence of strong correlation between samples in an analysis segment centred on the update-point and samples a certain fixed time later or earlier. For unvoiced speech there will be no such strong correlation. For voiced speech, a segment of length equal to a single pitch-period is extracted at each of the regular update-points. A description of each extracted segment is encoded. To reconstruct the voiced speech at the decoder, pitch-period length segments centred on adjacent update-points are interpolated to obtain a "synthesis segment" of speech between each pair of update-points. Unvoiced speech is generally encoded by switching to a simplified form of CELP coder [86]. This may result in the unvoiced speech being modelled by a pseudo-random sequence without periodicity, although provision for some periodicity may help with transitions and misclassifications.

PWI coding can either be applied directly to speech [17] or to an LP residual [70]. The LP residual for voiced speech is obtained by passing the input speech through an LP inverse filter, and may be expected to resemble a vocal tract excitation signal. A set of coefficients which characterises the LP inverse filter can be vector quantised to obtain an efficient low bit-rate representation. The use of line

spectral frequencies is recommended for this purpose. The LP inverse filter is so called because its transfer function is intended to be the inverse of an all-pole transfer function modelling the effect of the vocal tract. It is therefore intended to remove, as far as possible, the spectral envelope imposed on an assumed excitation signal by the vocal tract. At the decoder, once a voiced residual has been reconstructed by interpolating pitch-period length segments centred on adjacent update-points, the short-term spectral envelope is re-imposed on this signal to obtain the decoded voiced speech. It is claimed [70][71] that a PWI/CELP coder is able to produce good quality speech at around 4 kb/s.

A two-mode pitch-synchronous waveform interpolation (TPSWI) coder has been designed during the course of this project [83]. The TPSWI coder operates on the residual obtained using the LSF analysis filter discussed in chapter 3 as the LP inverse filter. The input speech is classified as voiced or unvoiced at each update-point by applying the two-way pitch detector described in chapter 2 to an analysis segment centred on the updated-point. A pitch-synchronous waveform interpolation (PSWI) modelling technique is used for voiced residual and when unvoiced speech is indicated only its power contour is encoded to allow a pseudo-random sequence to be used to replace it at the decoder.

The PSWI model is a PWI based technique for voiced speech in which pitch-period length segments are extracted from the residual at regular update-points. Each segment is represented using a gain-shape principle. The gain and shape factors may be encoded at different update rates. At the decoder, the wave-shapes at adjacent update-points, gain factors and the pitch-periods are interpolated separately. The interpolated wave-shapes are stretched to the correct period and scaled to the required gains to yield the reconstructed voiced speech residual. For unvoiced speech, the output of a pseudo-random noise generator producing a pseudo-Gaussian random sequence is scaled in amplitude to give it the power contour of the unvoiced residual. An overlap and add technique is employed to deal with voiced/unvoiced transitions in order to ensure a smooth signal evolution.

The next section of this chapter will be devoted to the concept of PWI coding. In section 5.3, the PSWI model used to synthesise voiced speech will be introduced. The technique used for unvoiced speech will be discussed in section 5.4. This is followed by a detailed description of the structure of the TPSWI coder. In section 5.6, a useful model for the phase spectra of voiced residual segments will be presented and evaluated. This model allows the phase spectra of voiced residual signal to be deduced at the decoder with little or no encoded information.

5.2 Fundamentals of prototype waveform interpolation (PWI) coding

As illustrated in figure 5.1, voiced speech is a quasi-periodic signal which means that at any point in time there is an exactly periodic signal which is strongly correlated with the voiced speech over a window centred on the given point in time. The degree of correlation depends on the window length and the degree of non-stationarity in the speech. The period of the exactly periodic waveform can be defined to be the instantaneous period of the speech. Both the instantaneous period and the shape of each cycle evolves over time. Clearly information is repeated from cycle to cycle of voiced speech. When update-points are close enough for the speech between them to be considered approximately stationary, a segment of pitch-period length centred on each update-point can be extracted from voiced speech, and an approximation to the speech can be recovered approximately by an interpolation process.

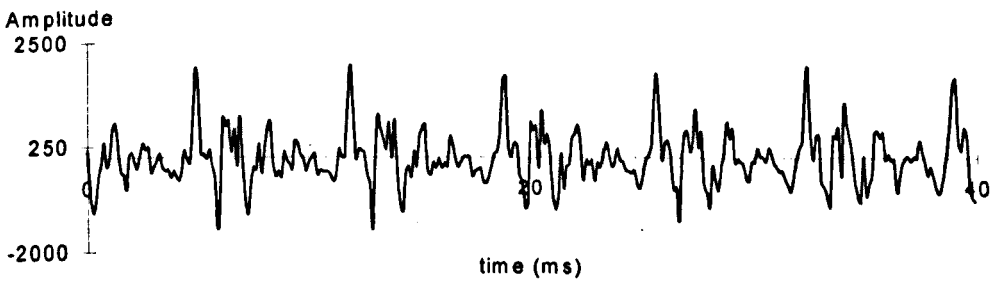


Figure 5.1 Example of a segment of voiced speech.

PWI coding exploits this property of voiced speech by extracting pitch-period length segments, namely prototype waveforms, at regular intervals of typically 20-30ms. A

description of each extracted prototype waveform is encoded along with the current pitch-period. At the decoder, the shapes and lengths of the prototype waveforms are interpolated from update-point to update-point. The interpolation process leads to a smooth evolution of the pitch-periods and the shapes of the prototype waveform.

5.2.1 Fourier series representation of quasi-periodic signals

A periodic signal can be represented as a Fourier series. The discrete time signal u(n) which is periodic with period p sampling intervals can be represented as:

$$u(n) = \sum_{k=0}^{K-1} C_k \cos\left(\frac{2\pi kn}{p}\right) + D_k \sin\left(\frac{2\pi kn}{p}\right) \tag{5.1}$$

where

C_k and D_k are the discrete Fourier series coefficients

K is the number of harmonics to be taken into account. Note that K must be less than p/2 to avoid aliasing.

The discrete Fourier series coefficients may be computed for a given periodic waveform using DFT analysis. The DFT analysis is performed over a single cycle, $\{u(n)\}_{0,p-1}$ say, which is defined from n=0 to p-1. In this case:

$$\begin{aligned} C_k &= \frac{1}{p}|U_k|\cos\left(\phi_{Uk}\right) \\ D_k &= -\frac{1}{p}|U_k|\sin\left(\phi_{Uk}\right) \\ & k = 0, 1, 2, \ldots, K-1 \end{aligned} \tag{5.2}$$

where $|U_k|$ and ϕ_{Uk} are the magnitude and phase respectively of the p-point DFT of $\{u(n)\}_{0,p-1}$.

Equation 5.1 may be adapted as an approximation to a quasi-periodic signal by making the set of Fourier series coefficients and the period change slowly over time. The quasi-periodic signal is then expressed as:

$$e(n) = \sum_{k=0}^{K(n)-1} C_k(n) \cos\left(\frac{2\pi kn}{p(n)}\right) + D_k(n) \sin\left(\frac{2\pi kn}{p(n)}\right) \tag{5.3}$$

5.2.2 Synthesis of voiced speech using PWI coding

In a PWI encoder, a description of a residual cycle is encoded at regular update-points. At the decoder, the Fourier series descriptions at adjacent update-points are interpolated to yield the recovered residual. The interpolation is performed on both the Fourier series coefficients and the pitch-period. Prototype waveforms will not necessarily be aligned in phase when they are extracted at the encoder. They must be phase aligned in order to maximise their similarity for efficient encoding. They must also be re-aligned at the decoder for interpolation. Phase alignment is carried out by comparing the cross-correlation functions between circularly shifted versions of the current prototype waveform and the previous prototype waveform. This can be performed in the Fourier series domain as [70], by calculating:

$$\xi' = \max_{\xi} \sum_{k=0}^{K'} \left\{ \left(\tilde{C}_k^{(l-1)} C_k^{(l)} + \tilde{D}_k^{(l-1)} D_k^{(l)} \right) \cos(k\xi) + \left(\tilde{D}_k^{(l-1)} C_k^{(l)} - \tilde{C}_k^{(l-1)} D_k^{(l)} \right) \sin(k\xi) \right\} \tag{5.4}$$

where $\tilde{C}_k^{(l-1)}$, $\tilde{D}_k^{(l-1)}$ are the Fourier series coefficients of the previous aligned prototype waveform and $C_k^{(l)}$, $D_k^{(l)}$ are the Fourier series coefficients of the current prototype waveform. Since the pitch-periods of the two cycles may not be the same, K' is the smaller number of harmonics of the two prototype waveforms. To calculate ξ', the summation is evaluated for a range of value of ξ between 0 and 2π.The Fourier series coefficients for the phase aligned prototype waveform are [70]:

$$\begin{aligned} \tilde{C}_k^{(l)} &= C_k^{(l)} \cos(k\xi') - D_k^{(l)} \sin(k\xi') \\ \tilde{D}_k^{(l)} &= C_k^{(l)} \sin(k\xi') + D_k^{(l)} \cos(k\xi') \\ & \qquad k = 0, 1, \ldots, K-1 \end{aligned} \tag{5.5}$$

At the decoder, the phase aligned prototype waveforms at the update-points are interpolated to produce an approximation to the signal between them. Interpolation can be applied to the Fourier series coefficients, each coefficient being modified on a time-sample by time-sample basis. Hence the synthetic residual signal over an update interval of N samples is obtained as [70],

$$\tilde{s}(n) = \sum_{k=0}^{K(n)-1} \left\{ \left((1-\psi(n)) \tilde{C}_k^{(l-1)} + \psi(n) C_k^{(l)} \right) \cos(k\omega(n)) + \left((1-\psi(n)) \tilde{D}_k^{(l-1)} + \psi(n) D_k^{(l)} \right) \sin(k\omega(n)) \right\} \tag{5.6}$$
$$n = 0, 1, \ldots, N-1$$

where $\sigma(n)$ is the fundamental instantaneous phase at time sample n and $\psi(n)$ is an interpolation function which can be linearly increased from 0 to 1 across the time interval n=0 to N-1 [70]. The interpolation function $\psi(n)$ may be a function of time sample n or phase sample $\sigma(n)$.

An interpolation formula for the fundamental instantaneous phase $\sigma(n)$ at time n may be based on the relationship between the instantaneous pitch-period p(n) and the fundamental instantaneous phase $\sigma(n)$ at discrete time sample n:

$$\frac{2\pi}{p(n)} = \frac{d\,\sigma(n)}{d\,n} \tag{5.7}$$

Strictly speaking the integration must be with respect to the continuous time variable t, but the same answer is obtained by considering n to be continuous and integrating with respect to n. p(n) is obtained by linearly interpolating between the pitch-periods of the prototype waveforms at the previous and current update-point, i.e.:

$$p(n) = \big(1 - \psi(n)\big)\,p^{(l-1)} + \psi(n)\,p^{(l)} \tag{5.8}$$

Substituting equation 5.8 into equation 5.7, equation 5.7 becomes:

$$d\sigma(n) = \frac{2\pi\,dn}{\big(1 - \psi(n)\big)\,p^{(l-1)} + \psi(n)\,p^{(l)}} \tag{5.9}$$

In the PWI technique proposed by Kleijn [70], $\psi(n)$ is defined as the following function of $\sigma(n)$:

$$\psi(n) = \frac{\sigma(n) - \sigma^{(l-1)}(N)}{2\pi\,M} \qquad n = 0, 1, \ldots, N-1 \tag{5.10}$$

where M is the number of prototype waveforms within an update interval n=0 to N-1, and is evaluated using:

$$M = \frac{2N}{p^{(l)} + p^{(l-1)}} \tag{5.11}$$

The value of M is not necessarily an integer.

To compute the fundamental instantaneous phase σ(n) for each time sample n, the expression for ψ(n) given by equation 5.10 is substituted into equation 5.9 and thus:

$$\frac{d\sigma(n)}{dn} = \frac{2\pi}{p^{(l-1)} + \frac{\sigma^{(l)}(n) - \sigma^{(l-1)}(N)}{2\pi M}\left(p^{(l)} - p^{(l-1)}\right)} \tag{5.12}$$

i.e.

$$\left(\frac{p^{(l-1)}}{2\pi} - \frac{\sigma^{(l-1)}(N)\left(p^{(l)} - p^{(l-1)}\right)}{4\pi^2 M}\right)\frac{d\sigma(n)}{dn} + \frac{\left(p^{(l)} - p^{(l-1)}\right)}{4\pi^2 M}\sigma^{(l)}(n)\frac{d\sigma(n)}{dn} = 1 \tag{5.13}$$

By integrating equation 5.13, it may be shown that the fundamental instantaneous phase σ(n) for each time sample n, for n=0 to N-1, is given by:

$$\sigma(n) = \begin{cases} \sigma(0) + 2\pi \dfrac{-M p^{(l-1)} + \sqrt{M^2\left(p^{(l-1)}\right)^2 + 2nM\left(p^{(l)} - p^{(l-1)}\right)}}{p^{(l)} - p^{(l-1)}} & p^{(l)} \neq p^{(l-1)} \\ \sigma(0) + 2\pi \dfrac{n}{p^{(l-1)}} & p^{(l)} = p^{(l-1)} \end{cases} \tag{5.14}$$

In the following section, the basic TPSWI model will be presented. The TPSWI model uses a PSWI technique for voiced speech and a pseudo-random sequence generator for unvoiced speech. The PSWI technique, is a form of PWI which uses a simpler formula for the fundamental instantaneous phase σ(n) than equation 5.14. The pseudo-random sequence generator used for unvoiced speech will be introduced in section 5.4. In section 5.5, the two units are combined together to form the TPSWI coder.

5.3 Synthesis of voiced speech using a pitch-synchronous waveform interpolation (PSWI) model

The PSWI model developed in this project is a PWI based coder which operates in the residual-domain. The LSF analysis filter which is discussed in chapter 3 is employed at the encoder to provide an LP residual. A prototype waveform u(n) is extracted in the residual-domain centred on each update-point, using the known pitch-period p. The prototype waveform is decomposed into a gain factor and shape information. The gain factor λ is defined as the square root of the mean-square power of the prototype waveform, i.e. the rms value of the prototype waveform. The shape of the prototype waveform is characterised by the pitch-synchronous DFT magnitude and phase spectra of the power normalised prototype waveform. Each DFT spectral sample U_k may be expressed as:

$$U_k = |U_k| e^{j\phi_{Uk}} \qquad k = 0, 1, \ldots, p-1 \tag{5.15}$$

where $|U_k|$ and ϕ_{Uk} are the DFT magnitude and phase spectra of the prototype waveform of length p samples.

To reconstruct the residual signal, the normalised prototype waveform at each update-point must be phase aligned with the normalised prototype waveform at the previous update-point as used to synthesise the LP residual. The previous prototype waveform will therefore have been aligned with the one before that. The phase alignment is done using the DFT magnitude and phase spectra of the two normalised prototype waveforms using an adaptation of equation 5.4. Expressing equation 5.4 in the DFT frequency-domain (see Appendix C), it follows that to align it with the previous aligned prototype waveform the current prototype waveform must be delayed by ξ' sampling intervals where:

$$\xi' = \max_{\xi} \sum_{k=0}^{K'-1} \left|U_k^{(l-1)}\right| \left|U_k^{(l)}\right| \cos\left(\tilde{\phi}_{Uk}^{(l-1)} - \phi_{Uk}^{(l)} - k\xi \right) \tag{5.16}$$

where

$\left|U_k^{(l-1)}\right|$ and $\tilde{\phi}_{Uk}^{(l-1)}$ are the DFT magnitude and phase spectra of the previously aligned normalised prototype waveform.

$\left|U_k^{(l)}\right|$ and $\phi_{Uk}^{(l)}$ are the DFT magnitude and phase spectra of the current normalised prototype waveforms.

The phase aligned prototype waveform is obtained by modifying only the DFT phase spectrum of the current prototype waveform. Hence the aligned phase spectrum is:

$$\tilde{\phi}_{Uk}^{(l)} = \phi_{Uk}^{(l)} + k\xi' \qquad k = 0, 1, 2, \ldots, K-1 \tag{5.17}$$

To obtain the reconstructed voiced residual between adjacent update-points, the DFT real and imaginary coefficients of the phase aligned prototype waveform are computed and are interpolated as follows:

$$\tilde{s}(n) = \frac{1}{p(n)} \sum_{k=0}^{K(n)-1} \left\{ \left(\tilde{R}_k^{(l-1)} + \psi(n)\left(\tilde{R}_k^{(l)} - \tilde{R}_k^{(l-1)} \right) \right) \cos\left(k\alpha(n)\right) + \left(\tilde{I}_k^{(l-1)} + \psi(n)\left(\tilde{I}_k^{(l)} - \tilde{I}_k^{(l-1)} \right) \right) \sin\left(k\alpha(n)\right) \right\} \qquad n = 0, 1, \ldots, N-1 \tag{5.18}$$

where

$\psi(n)$ is an interpolation function.

$\tilde{R}_k$ and $\tilde{I}_k$ are the DFT real and imaginary coefficients of the aligned prototype waveforms.

In contrast to the PWI technique proposed by Kleijn [70], the interpolation function $\psi(n)$ used in the PSWI model is a linear function of time sample n. Hence $\psi(n)$ increases linearly from 0 to 1 as n increases from 0 to N-1 and is therefore:

$$\psi(n) = \frac{n}{N} \tag{5.19}$$

The fundamental instantaneous phase σ(n) is computed by a quadratic interpolation function. To derive the quadratic phase interpolation formula, the instantaneous fundamental pitch-frequencies at update-points *l-1* and *l*, i.e. $\varpi^{(l-1)}$ and $\varpi^{(l)}$, are linearly interpolated to obtain:

$$\varpi(n) = \varpi^{(l-1)} + \frac{n}{N}\left(\varpi^{(l)} - \varpi^{(l-1)}\right) \qquad n = 0, 1, \ldots, N-1 \tag{5.20}$$

Since σ(n) must be the integral of the fundamental instantaneous pitch-frequency at any point in time n, the fundamental instantaneous phase is then:

$$\sigma(n) = n\varpi^{(l-1)} + \frac{n^2}{2N}\left(\varpi^{(l)} - \varpi^{(l-1)}\right) + C \qquad n = 0, 1, \ldots, N-1 \tag{5.21}$$

Again the integration must be with respect to the continuous time variable t, but the same answer is obtained by considering n to be continuous and integrating with respect to n. The value of C is the initial condition of the integral and is set to the fundamental instantaneous phase value that would have been attained at the previous update-point, using the previous interpolation formula, i.e.:

$$\sigma(n) = n\varpi^{(l-1)} + \frac{n^2}{2N}\left(\varpi^{(l)} - \varpi^{(l-1)}\right) + \sigma^{(l-1)}(N) \qquad n = 0, 1, \ldots, N-1 \tag{5.22}$$

where $\sigma^{(l-1)}(n)$ is the formula for σ(n) obtained by interpolating between the previous update-point and the one before that.

This formula differs from equation 5.14 and is considerably simpler. Equation 5.14 was obtained [70] by linearly interpolating the fundamental pitch-period as a function of phase sample σ(n). In the new equation, the fundamental pitch-frequency is linearly interpolated as a function of time sample n. This simpler formula is essentially that used by IMBE coder [13] which does not attempt to preserve phase information. In the PSWI model, phase relationship between harmonics is preserved by attempting to encode the DFT real and imaginary coefficients: not just these

magnitudes. The interpolated instantaneous fundamental pitch-period p(n) may be determined at any time n as:

$$p(n) = \frac{2\pi}{\varpi(n)} \tag{5.23}$$

When the interpolated normalised residual has been obtained, it is scaled by a gain contour derived from the gain factors specified at the update-points to yield the reconstructed voiced residual. The gain factors are interpolated between update-points to obtain a smooth contour and hence a smooth evolution of the reconstructed voiced residual power. The interpolation of gain factors contour is done as follows:

$$\lambda(n) = \lambda^{(l-1)} + \frac{n}{N}\left(\lambda^{(l)} - \lambda^{(l-1)}\right) \tag{5.24}$$

where $\lambda^{(l-1)}$ and $\lambda^{(l)}$ are the gain factors at update-points *l-1* and *l*, i.e.

$$\hat{e}'(n) = \hat{e}(n)\left(\lambda^{(l-1)} + \frac{n}{N}\left(\lambda^{(l)} - \lambda^{(l-1)}\right)\right) \qquad n = 0, 1, \ldots, N-1 \tag{5.25}$$

An example of a 20ms segment of decoded voiced speech obtained using an unquantised implementation of the PSWI coder is shown in figure 5.2. The original 20ms segment is shown in figure 5.2a, the original LP residual is shown in figure 5.2b and the reconstructed residual is shown in figure5.2c. It may be seen that the reconstructed residual is similar in shape to the original, though there are easily identified differences. An important observation is that the reconstructed residual segment shown in figure 5.2c is slightly advanced in time compared with the original residual segment shown in figure 5.1b. This causes the reconstructed speech segment as shown in figure 5.2d to be similarly shifted in time. Such time shifts, which are sometimes advances and sometimes delays are characteristic of PWI techniques. The phenomenon is due to the effect of the quadratic phase interpolation and, in fact, is important for the effective operation of most forms of PWI. During the quadratic phase interpolation, only the fundamental pitch-frequency at the update-points and the frequency and the phase relationships between harmonics are preserved. The value of fundamental instantaneous phase at each update-point is left unconstrained by the encoded data and is calculated at the decoder, as described above with the

requirement of maintaining continuity of the fundamental frequency component at each update-point. This also to a large degree maintains the continuity of each harmonic since their instantaneous phases are $k\sigma(n)$ for k=2, 3, ... K-1. This means that the reconstructed speech should not be expected to be synchronous with the original speech. Experiments suggest that the perceptual quality of the resultant speech is not affected by this kind of "linear phase" drifting and that it is much more important to guarantee signal continuity at each update-point than to maintain synchronism with the original speech.

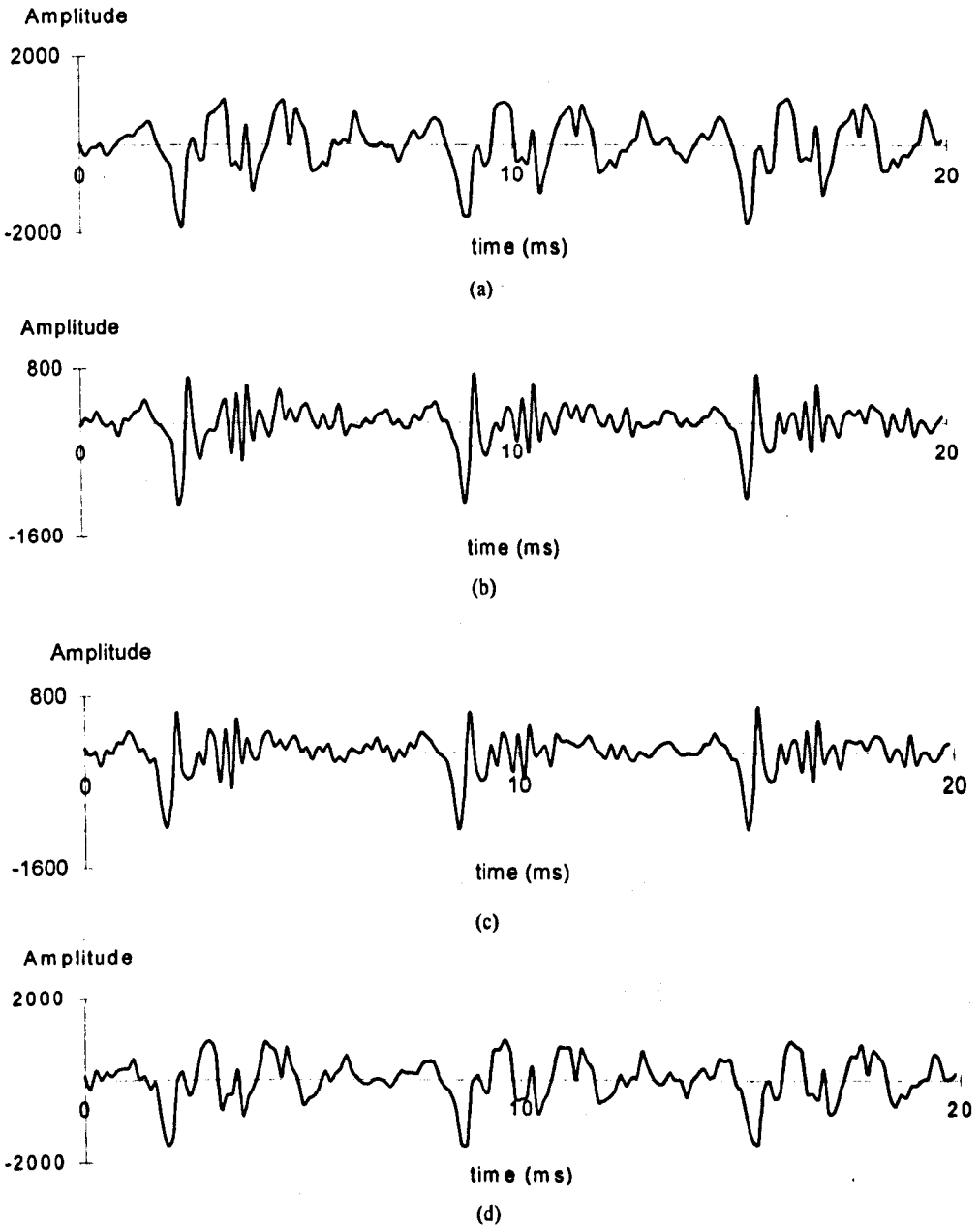


Figure 5.2 Example of a frame of voiced speech synthesised by the PSWI model.
(a) original speech (b) original LP residual (c) reconstructed residual (d) reconstructed speech

5.4 Synthesis of unvoiced speech using a pseudo-random Gaussian sequence generator

It is widely believed [72] that the LP residual for frames of unvoiced speech can be modelled simply by a pseudo-random Gaussian sequence. The power contour and frequency spectrum of the synthesised residual are important and experiments have shown [72] that unvoiced speech can be adequately modelled by exciting an LP synthesis filter by such a signal with the short-term average power updated every 5ms. It is known [75] that the perceptual quality of unvoiced speech is preserved by using a pseudo-random sequence with a roughly identical magnitude spectrum to the original LP residual and a similar power contour. Therefore unvoiced speech segments are modelled using an LSF synthesis filter excited by a power contoured pseudo-random Gaussian sequence which is, in principle, spectrally white. The LSF synthesis filter aims to model the spectral envelope of the unvoiced speech.

To encode an unvoiced speech segment, the unvoiced segment is first processed by an LSF analysis filter to provide a residual which has an approximately flat magnitude spectrum. The short-term power of the residual is then computed at 5ms intervals using:

$$G^2 = \frac{1}{N_s} \sum_{n=0}^{N_s-1} r^2(n) \tag{5.26}$$

where

G is the rms value

r(n) is the residual signal

N_s is a sub-window length equal to 40 samples for 8kHz sampling frequency

The power contour of the residual is encoded at 5ms intervals. In practice, an rms magnitude contour G rather than a power contour is used for more efficient computation. The LSF coefficients are updated, as usual, at a lower rate, normally 20ms, since only a roughly identical spectral envelope is required.

To reconstruct the unvoiced speech, a pseudo-random Gaussian sequence generator (PRGSG) produces random numbers with an approximately Gaussian distribution, a white frequency spectrum and constant average power normalised to unity. Each sample of the pseudo-random sequence is multiplied by a sample of the interpolated received power contour, i.e. multiplying a sample of the interpolated gain factor, to yield the required excitation as:

$$\hat{e}'(n) = \left(G^{(l-1)} + \frac{n}{N_s}\left(G^{(l)} - G^{(l-1)} \right) \right) \hat{e}(n) \qquad (5.27)$$

$$n = 0, 1, \ldots, N_s - 1$$

where

$\hat{e}'(n)$ is the scaled excitation signal

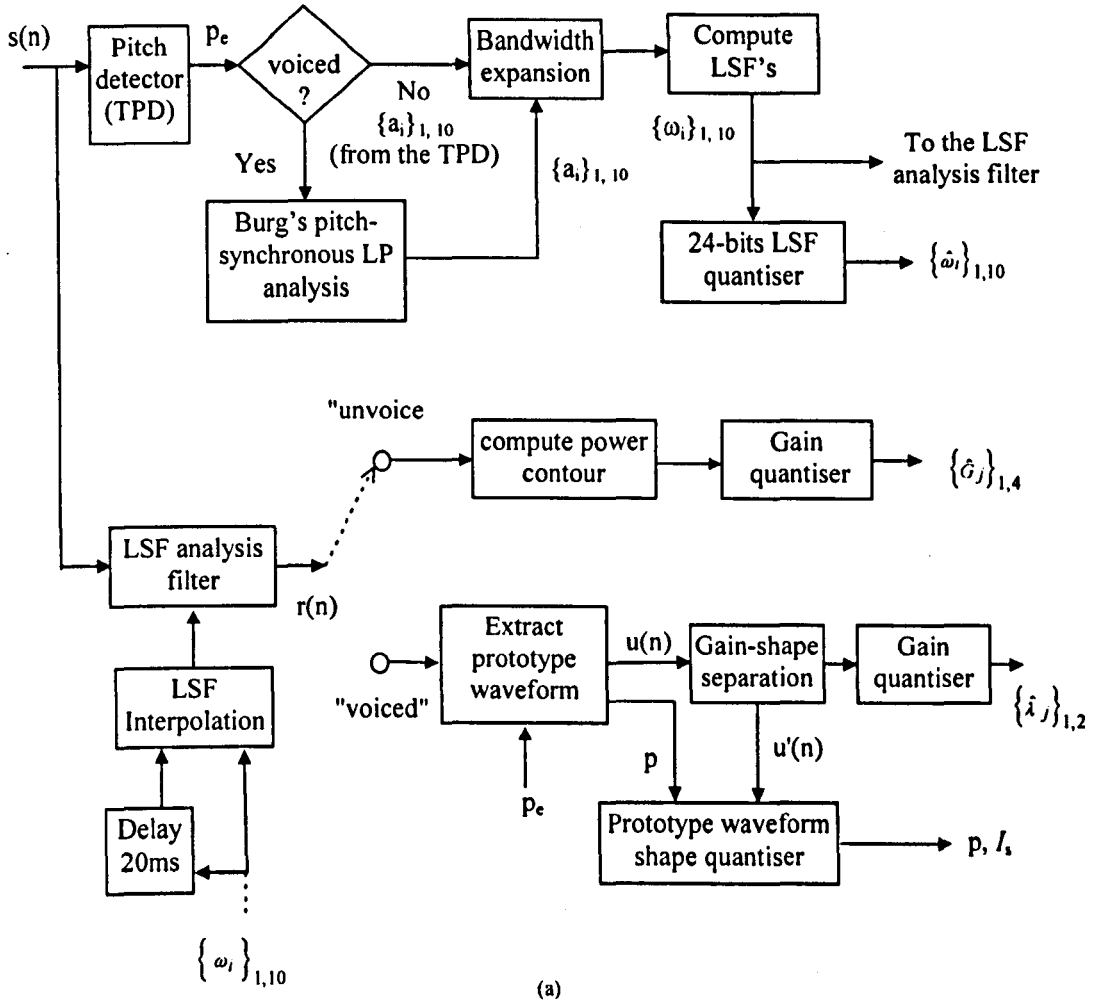
$\hat{e}(n)$ is the pseudo-random sequence produced by PRGSG

$\hat{G}^{(l-1)}$ and $\hat{G}^{(l)}$ are the encoded gain factors across a 5ms interval

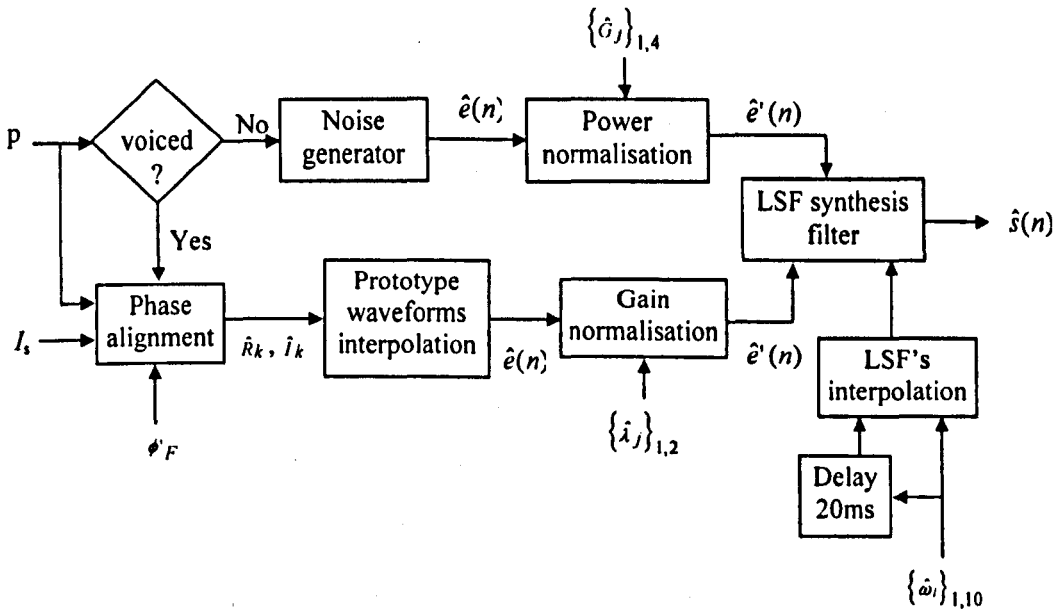
The excitation is then processed by the LSF synthesis filter which re-imposes the original spectral envelope of the unvoiced speech.

5.5 The two-mode pitch-synchronous waveform interpolation (TPSWI) coder

In figure 5.3 a schematic diagram of the TPSWI coder is shown. The quantisation of the TPSWI coder will be discussed in chapter 7. Unquantised parameters are used at this stage. The TPSWI coder is operated with update-points at regular intervals of 20ms which means that synthesis frames at the decoder are each of length 20ms. Analysis frames at the encoder are centred on each update-point. The nature of the input speech around each update-point is first determined by the two-way pitch detector discussed in chapter 2. When voiced speech is indicated a pitch-period is given and Burg's pitch-synchronous LP analysis method is performed to yield a set of 10 LP ladder filter coefficients. Otherwise, when unvoiced speech is indicated the 10 LP ladder filter coefficients produced by the pitch detector are directly used. A 10Hz bandwidth expansion is then applied to each LP pole as described in section 4.4 and the modified set of LP ladder coefficients is converted to a set of LSF coefficients for the current update-point. After the LSF coefficients are available, the frame of input speech is processed by the LSF analysis filter with the LSF coefficients interpolated between those obtained at update-points to yield the LP residual. The required parameters for the residual are extracted at the current update-point. These parameters are the gain and shape of a prototype waveform for voiced speech and four samples of the rms value contour for unvoiced speech. The decoder utilises these parameters to reconstruct a residual signal applied to excite an LSF synthesis filter with LSF coefficients interpolated as for the LSF analysis filter used at the encoder. The short-term spectral envelope is re-imposed on the reconstructed excitation signal by this LSF synthesis filter.



(a)



(b)

Figure 5.3 The schematic diagram of the TPSWI coder.
(a) the encoder (b) the decoder

5.5.1 The TPSWI encoder

For the TPSWI coder operating with 20ms update intervals, a 10ms look-ahead is required at the analysis stage. An asymmetric window used by the TPD is positioned such that the separating point between its two functions is located at the current update-point as illustrated in figure 5.4. The formulation of the asymmetric window was presented in chapter 3.

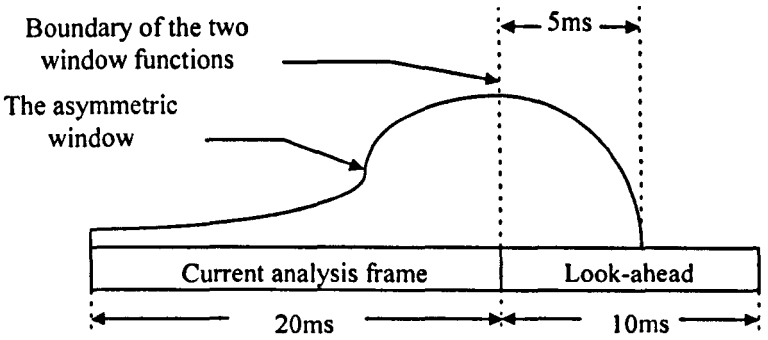


Figure 5.4 Positioning of the analysis window for the TPD on a speech frame.

If unvoiced speech is indicated, the pitch-period given by the TPD will be set to zero and the set of a_i coefficients produced by the TPD is used directly. In the case of voiced speech, an estimation of the true pitch-period is given by the TPD and Burg's pitch-synchronous LP analysis method is employed to re-estimate, this time more accurately, the short-term spectral envelope of the voiced speech. A rectangular window is used to extract the analysis speech samples for Burg's algorithm. The size of the window is adapted to the current estimated pitch value as described in chapter 3. The window is positioned such that it is centred on the current update-point as shown in figure 5.5.

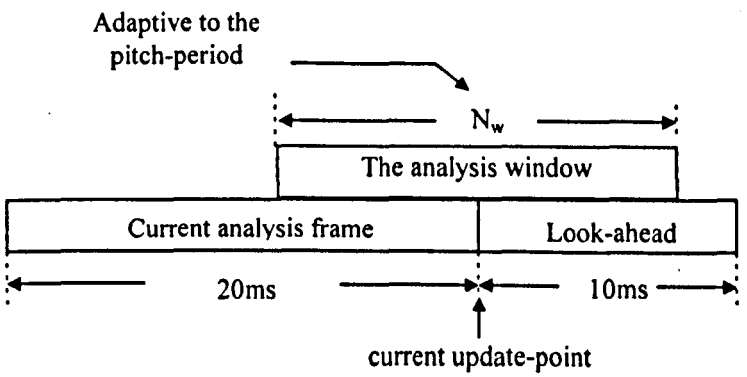


Figure 5.5 Positioning of the analysis window for Burg's pitch-synchronous LP analysis.

Once the set of bandwidth expanded LP ladder coefficients is available, the LSF coefficients are computed by an iterative process, the details of which are presented in Appendix B. To obtain the LP residual, the input speech signal is processed by an LSF analysis filter whose LSF coefficients are interpolated on a sample by sample basis between the new set of LSF coefficients and the set of LSF coefficients from 20ms earlier. Based on the nature of the input speech, different parameters are extracted from the speech residual, as explained in the following paragraphs.

5.5.1.1 Extraction of prototype waveforms from voiced speech

When voiced speech is detected, a segment of length one pitch-period is extracted from the residual. This is a prototype waveform. Since the pitch-period given by the TPD is an estimate calculated for a relatively wide analysis window, a more accurate and localised estimate of the instantaneous pitch-period at the current update-point must be found. Computation of this instantaneous pitch-period is carried out by examining the cross-correlation function between successive possible pitch-cycles for a range of potential pitch candidates. The range of potential candidates is set equal to ±5% of the estimated pitch-period, p_e say, provided by the TPD. The range of allowed instantaneous pitch-periods is restricted to between 15 and 150 samples. As a result, the more localised instantaneous pitch-period estimate is $p^{(l)}=m'$, with

$$m' = \max_m \left\{ \frac{\sum_{n=N-1}^{N-m} s(n)\, s(n-m)}{\sqrt{\sum_{n=N-1}^{N-m} s^2(n) \sum_{n=N-1}^{N-m} s^2(n-m)}} \right\} \qquad (5.28)$$

$$0.95 * p_e \le m \le 1.05 * p_e \quad (m \in \text{integer})$$

where N is the frame length and s(n) is the speech signal. The current update point is at n=N.

Once the accurate pitch-period estimate $p^{(l)}$ has been obtained for update-point l a prototype waveform with $p^{(l)}$ samples is extracted at the end of the residual frame in the way that is illustrated in figure 5.6.

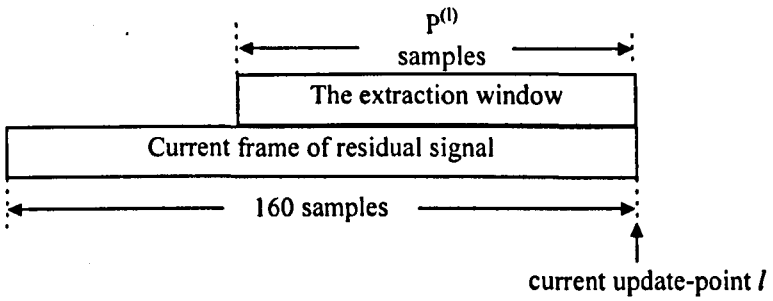


Figure 5.6 Extraction of prototype waveform from a frame of residual signal.

The prototype waveform should, in principle, be centred on each update-point. This would require look-ahead into a future residual frame thus increasing the total coding delay from 10ms to 30ms, as an extra 20ms speech segment is required to be analysed to provide the future frame of residual signal. To save the extra 20ms coding delay, extraction of the prototype waveforms are allowed to be slightly offset from the centre. Experimental results showed that the quality of the synthesised speech is not affected by this offset. The prototype waveforms are then decomposed into a gain factor and shape representation as described in section 5.3.

5.5.1.2 Extraction of gain factors from unvoiced speech

The unvoiced gain factors are updated in every 5ms, therefore four gain factors are required for each 20ms synthesis frame. The corresponding frame of LP residual at the encoder is divided into four sub-frames at the analysis stage and the required gain factors for each unvoiced sub-frame are computed using equation 5.26.

5.5.2 The TPSWI decoder

At the TPSWI decoder, an approximation to the LP residual which becomes the excitation signal to the LSF synthesis filter is constructed from the received parameters. The construction of the excitation signal for voiced and unvoiced speech has been discussed in sections 5.3 and 5.4.

A strategy for voicing transitions is one of the most important issues in a speech coder. Mishandling of voiced onset could lead to a loss of speech intelligibility. An overlap-add technique, which incorporates a triangular window,

was found to be suitable for dealing with voicing transitions. In the TPSWI coder, a transition frame is defined when a voiced frame is detected after an unvoiced frame (voiced onset), or vice versa (voiced offset). In both cases the speech excitation is reconstructed by summing together two windowed signals, $\hat{e}1_{w1}$ and $\hat{e}2_{w2}$. In the case of a voiced onset, $\hat{e}1$ is a pseudo-random sequence contoured using the gain factors of the previous unvoiced frame and $\hat{e}2$ is obtained by periodically repeating the current received prototype waveform from the current update-point backward to the previous one. For a voiced offset, $\hat{e}1$ is obtained by repeating the previous received prototype waveform to fill up the entire speech frame from the previous to the current update-point and $\hat{e}2$ is the sequence generated by the pseudo-random Gaussian sequence generator. The contour derived from the four gain factors is imposed on the normalised pseudo-random Gaussian sequence prior to the overlapping and adding. In the reconstructed voiced segments, i.e. $\hat{e}2$ for voiced onset and $\hat{e}1$ for voiced offset, the rms value of the prototype waveform is made equal to the received gain factor prior to the repeating process. When the two synthesis frame components have been generated, each is multiplied by the corresponding window function and then they are added together. Thus the "overlap-add" reconstructed excitation signal synthesis frame is,

$$\hat{e}(n) = \hat{e}1_{w1}(n) + \hat{e}2_{w2}(n) \qquad n = 0, 1, 2, \ldots, N-1 \tag{5.29}$$

where

$$\hat{e}1_{w1} = e1(n)\,w1(n) \text{ with } w1(n) = 1.0 - \frac{n}{N}$$

$$\hat{e}2_{w1} = e2(n)\,w2(n) \text{ with } w2(n) = \frac{n}{N}$$

In figure 5.7a, an example of a voiced onset frame is shown. To reconstruct the speech excitation, the unvoiced gain factors in the previous frame are recalled. These are scaled to a new pseudo-random Gaussian sequence to obtain $\hat{e}1$ shown in figure 5.7b. The current prototype waveform is reconstructed and scaled to the required rms value. The prototype waveform is repeated from the end of the frame towards the beginning to yield $\hat{e}2$, as shown in figure 5.7c. $\hat{e}1$ and $\hat{e}2$ are windowed and added together to form a speech excitation shown in figure 5.7d. A

triangular window is used in both cases and the windows are shown together with $\hat{e}1$ and $\hat{e}2$ in figures 5.7b and c.

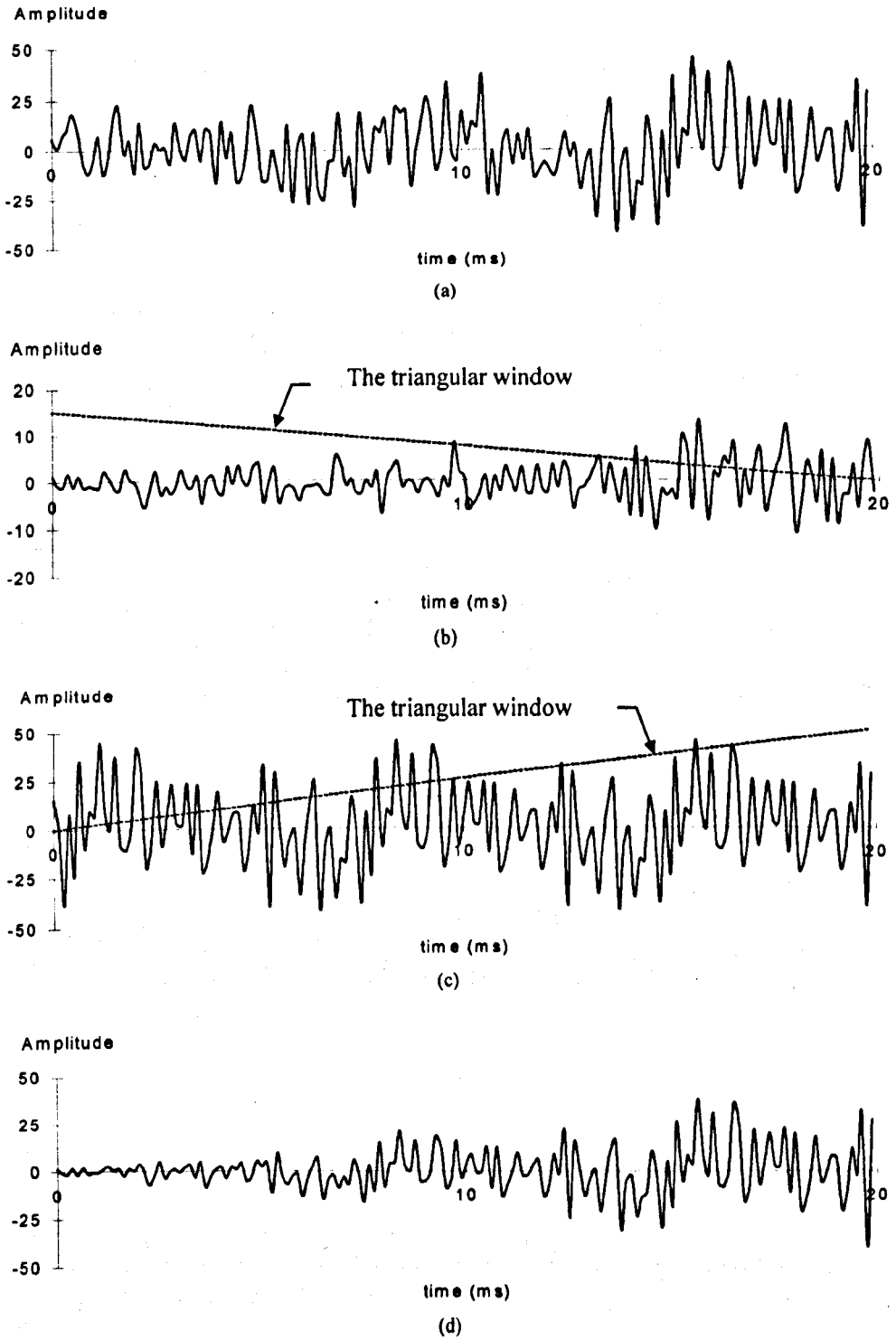


Figure 5.7 Overlap-add technique used for voiced onset.
(a) original speech segment (b) $\hat{e}1$ is a pseudo-random Gaussian sequence whose samples are scaled by the rms value contour specified in the previous frame (c) $\hat{e}2$ obtained by repeating the prototype waveform of the current speech frame (d) the reconstructed speech excitation

A voiced offset frame is shown in figure 5.8a. In this case, the pseudo-random Gaussian sequence generator creates a normalised sequence and its rms values is scaled to the required rms value contour to yield $\hat{e}2$, as shown in figure 5.8b. To obtain $\hat{e}1$, the scaled prototype waveform in the previous voiced frame is periodically repeated, as shown in figure 5.8b. Finally, $\hat{e}1$ and $\hat{e}2$ are windowed and added together to form a speech excitation synthesis frame of length N samples, shown in figure 5.8d.

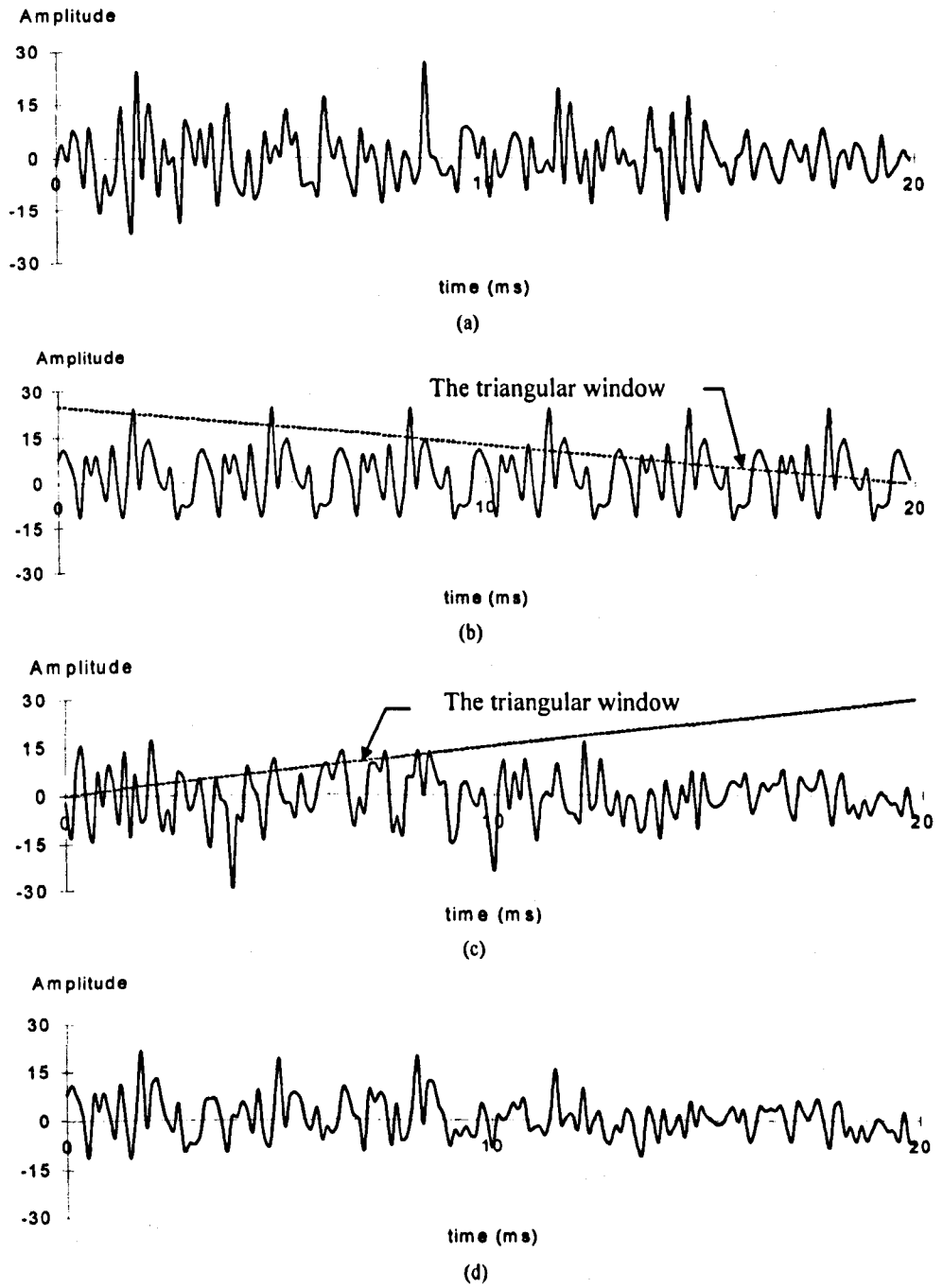


Figure 5.8 Overlap-add technique used for voiced offset.
(a) original speech segment (b) $\hat{e}1$ obtained by repeating the prototype waveform of the previous update-point (c) $\hat{e}2$ is the pseudo-random Gaussian sequence (d) the reconstructed speech excitation

5.5.3 Subjective evaluation of the TPSWI coder

The TPSWI model was implemented as a C program [83] simulating the operation of a real-time coder. At this stage the model parameters produced i.e. the LSF coefficients, the pitch-period and the residual parameters were not quantised. The effect of quantising these parameters to achieve the required low bit-rate will be considered in chapter 7. The C program simulation was assessed using the speech file "OPERATOR.DAT" [21] as input data. This file consists of about 90seconds of a conversation between a male and a female sampled at 8kHz with 16bits per sample. It was found that good speech quality can be achieved with update-points at 20ms intervals although a rather buzzy effect occurs from time-to-time which appears to be due to the enforcement of too much periodicity in the synthesised speech. This excess periodicity is known [70] to be the feature of PWI when information about only one prototype waveform is encoded every 20ms.

In addition, experiments have been performed on the TSPWI coder with the number of prototype waveforms per 20ms of voiced speech increased from one to two, four and eight. The main update-points at which the LSF coefficients and the pitch-period are specified remained always at 20ms intervals. The prototype waveforms extracted between the main update-points are referred to as "intermediate" prototype waveforms. The intermediate prototype waveforms were extracted from the LP residual using "intermediate" estimates of pitch-period. Each intermediate estimate of pitch-period was obtained by linearly interpolating the instantaneous pitch-frequencies at the 20ms update-points and converting the resulting frequency to a period. An immediate enhancement in the speech quality was obtained by encoding two, rather than one, prototype waveforms per 20ms of speech. This means that a prototype waveform is extracted at each update-point and mid-way between each pair of update-points. Further improvement is achieved by encoding four prototype waveforms per 20ms of speech. The synthesised speech obtained in this case was fairly natural. It was found that when eight prototype waveforms were extracted per 20ms of speech, except for a minor distortion in a few segments, the synthesised speech was almost indistinguishable from the original. In this case a prototype waveform is being extracted every 2.5ms, and this means that, especially for low pitched male speech there will be much overlap between prototype

waveforms. Although it may seem, at this stage, that there is a lot of data to be encoded with this high extraction rate, ways of reducing the data without sacrificing all the advantages gained will be discussed later. It must be pointed out that great care must be taken when the intermediate prototype waveforms are being extracted. A roughness in speech quality can result from error in any of the intermediate prototype waveforms. Details of how the intermediate prototype waveforms are extracted will be given in chapter 6.

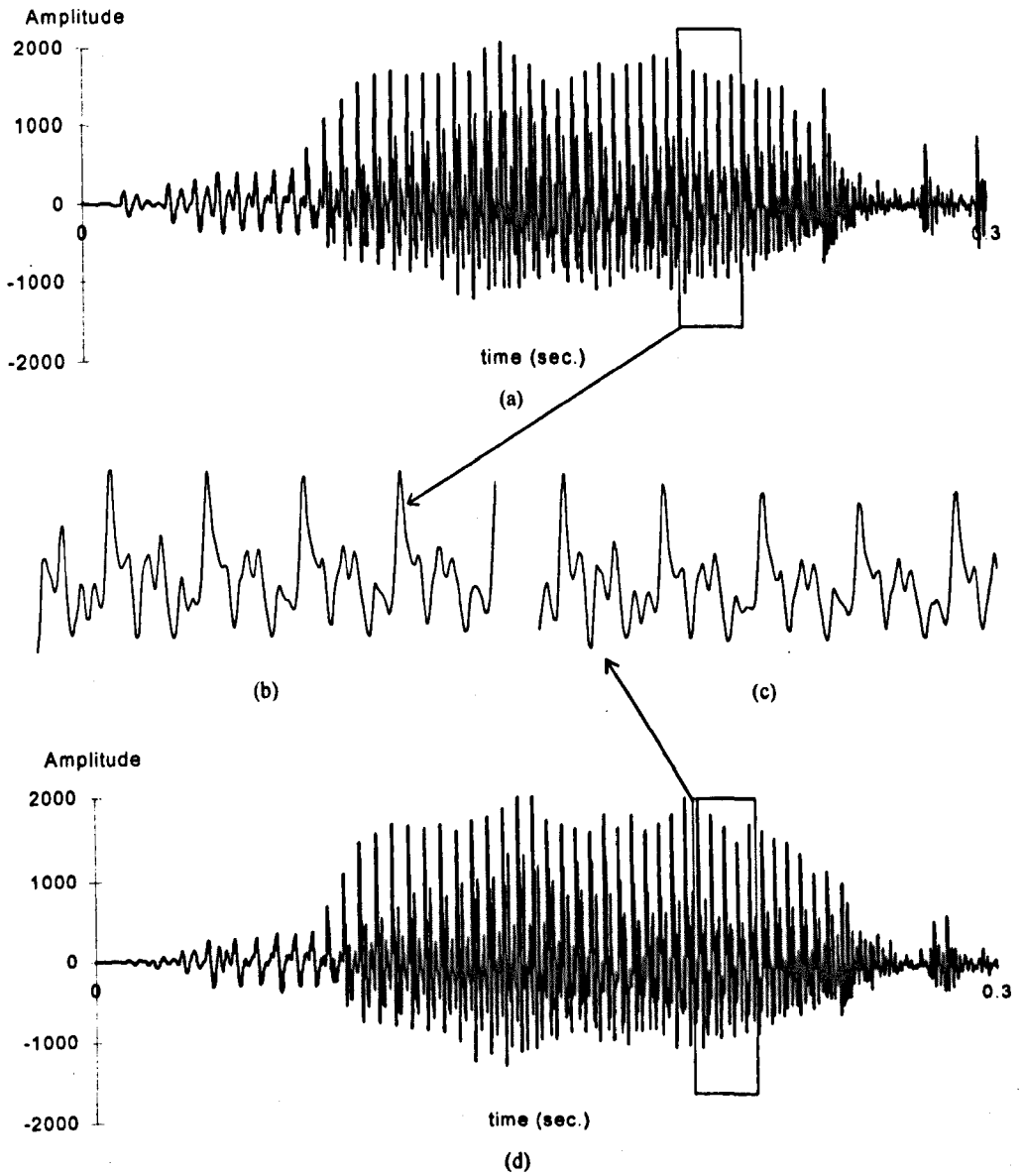


Figure 5.9 Example of voiced speech synthesised using the TPSWI coder with prototype waveforms extracted at 2.5ms intervals. (a) original speech segment (b) snap-shot of original voiced speech (c) snap-shot of synthesised speech (d) decoded speech segment

In figure 5.9a, an example of a 300ms speech segment is shown. The utterance is the word "NO" spoken by a male speaker. The segment is part of the test

file of speech applied to the TPSWI coder with eight prototype waveforms being encoded per 20ms. The synthesised speech obtained at the decoder is presented in figure 5.9d. It can be seen that the amplitude envelope of the synthesised signal follows very well the original shown in figure 5.9a. A smooth voiced onset and offset is preserved and although the unvoiced speech at the beginning and end of figure 5.9d looks in detail rather unlike the original, the distortion could not be perceived. In figure 5.9b and c, a segment of the original voiced speech and the corresponding segment of decoded speech are presented. It can be seen that the synthesised wave-shape is almost the same as the original one, and that there is, as expected, a time shift between them.

The perceptual quality of TPSWI encoded speech is considerably enhanced by having eight prototype waveforms per 20ms. Each prototype waveform is represented by a gain factor and the DFT magnitude and phase spectra of a power normalised version of it. It was also found that no noticeable loss of this enhancement occurred when only two gain factors per 20ms of voiced speech rather than eight were encoded. In this case the non-encoded gain factors are obtained by interpolation.

Using the speech file "OPERATOR.DAT" as test data, the TPSWI coder with 20ms prototype waveform extraction was compared with a PWI/CELP coder with the same prototype waveform extraction rate implemented by the author [82]. The main differences between these coders lie in the encoding technique used for unvoiced speech and the way that the TPSWI model handles the voiced/unvoiced transitions. Subjective tests suggested that the speech quality obtained from the two coders are comparable. However, the TPSWI coder had a smoother voiced/unvoiced transitions than the PWI/CELP coder. Many transient effects found in the PWI/CELP coder were eliminated. The TPSWI coder is considerably simpler in its approach to coding unvoiced speech since only the four short-term rms measurements of the unvoiced residual are required for 20ms rather than a CELP representation. This experiment demonstrated the importance of the rms value contour in the perception of unvoiced speech.

5.6 The TPSWI coder with phase derivation at the decoder

In the TPSWI coder, prototype waveforms are extracted at the encoder at a constant rate. The prototype waveforms can be represented by Fourier series cosine and sine coefficients [70], which are proportional to the real and imaginary parts of pitch-synchronous DFT spectra, or the magnitudes and phases of the DFT spectra [83]. Some efficiency can be gained by encoding only the magnitude spectrum of each prototype waveform on the assumption that the phase spectrum is likely to be less perceptually important and that an approximation to the true phase spectrum can be recovered at the decoder according to some assumption about the voiced speech production model.

In this thesis, a phase derivation scheme is proposed for estimating the phase spectrum of a prototype waveform. This is achieved by studying the conventional voiced speech production model and re-expressing it in terms of the time-reversed impulse response of the 2nd-order all-pole filter normally used to model the glottal excitation when cascaded with vocal tract and lip-radiation filters. Under the assumptions of this model of human speech production, the original speech, which is non-minimum phase, can be assumed to be equal to a minimum phase time-domain signal passed through a second-order all-pass filter. This means that, in theory, the phase spectrum of a prototype waveform can be modelled as the phase response of the 2nd-order all-pass filter. The basis for this theory is explained below. The phase derivation scheme allows quantisation of the PSWI model parameters to be concentrated on the magnitude spectra of prototype waveforms.

5.6.1 Deriving the phase spectrum of an LP residual signal using a voiced speech production model

A voiced speech production model is often expressed in terms of three cascaded filters as illustrated in figure 5.10. The filters are:-

i) a filter G(z), whose impulse response provides the glottal excitation pulse,

ii) an all-pole filter V(z), modelling the vocal tract resonances, and

iii) a filter L(z), modelling lip-radiation.

For voiced speech this set of filters is assumed to be driven by a signal e(n) made up of a pseudo-periodic series of discrete time impulses.

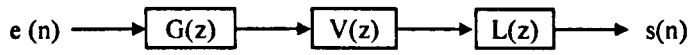


Figure 5.10 A simple voiced speech production model.

V(z) is assumed to be an all-pole model of the vocal tract which has the transfer function,

$$V(z) = \frac{1}{\prod_{i=1}^{P}\left(1 - \rho_i z^{-1}\right)} \tag{5.30}$$

where ρ_i are the poles of V(z) and P is total number of poles. The formant frequencies of voiced speech are dependent on the location of these poles within the unit circle.

L(z) is normally considered [6] to be a differentiator which has a single positive zero, α, on the real axis just inside the unit circle. L(z) is therefore defined as,

$$L(z) = 1 - \alpha z^{-1} \tag{5.31}$$

It is often assumed [2] that G(z) is a 2nd order all-pole filter with transfer function,

$$G(z) = \frac{1}{\left(1 - \beta_1 z^{-1}\right)\left(1 - \beta_2 z^{-1}\right)} \tag{5.32}$$

where β_1 and β_2 are the poles of G(z).

The first of these poles is assumed coincident with the lip-radiation zero ($\alpha=\beta_1$), and the effect of this pole-zero combination on the magnitude spectrum of speech is assumed to cancel [6]. As a result, the voice speech production model is often assumed to contain only poles within the unit circle as illustrated in figure 5.11a. Linear prediction aims to model the short-term spectral envelopes of voiced speech segments by estimating the positions of the assumed poles within the unit circle. The LP estimation technique can produce only the parameters of a minimum phase

model of speech production, i.e. an all-pole model where all the poles lie inside the unit circle. The spectral envelope is whitened by the LP inverse filter, which is assumed to place zeros on the locations of the poles as illustrated in figure 5.11b. The output of the inverse filter will have a flat magnitude spectrum if the all-pole model is valid and there are sufficient zeros to cover all the poles.

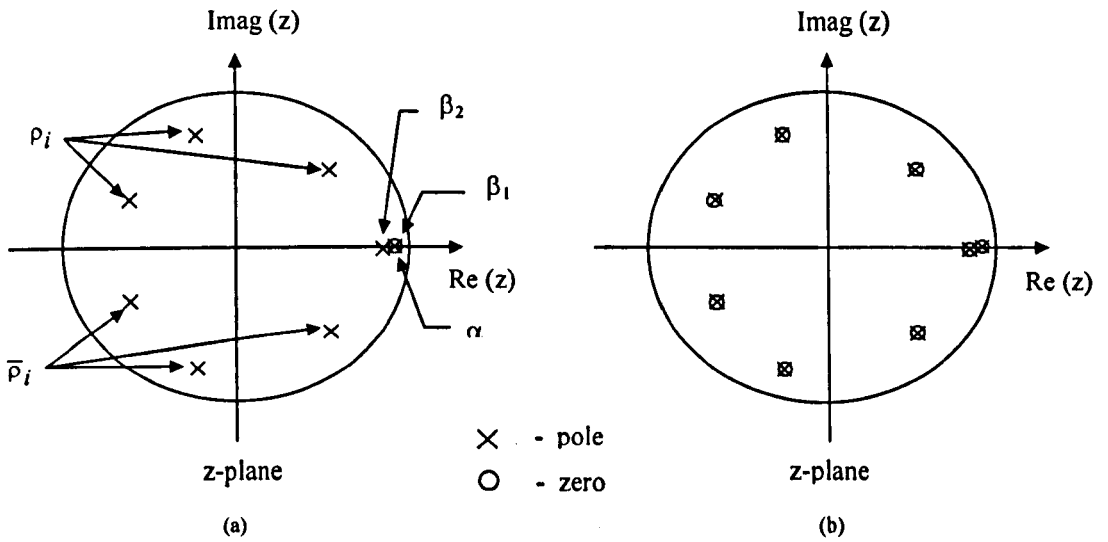


Figure 5.11 Poles and zero of a simple voiced speech production model. (a) voiced speech production model (b) poles and zeros of spectrally flattened residual as sometimes assumed

It has been suggested [77] that the naturalness of synthesised speech can be better preserved by using a transfer function G(z) which has an impulse response as shown in figure 5.12. This impulse response is consistent with theoretical models of the operation of the vocal cords and also to the waveforms obtained by direct measurement, for example using a laryngeograph. The slow rise of the response corresponds to the characteristically slow opening of the glottis and the sharp fall models its rapid closure. These characteristics are seen in the well known Rosenberg Pulse sometimes used to model glottal excitation [77], and formulated as:

$$g(t) = \begin{cases} A\left(3\left(\frac{t}{T_P}\right)^2 - 2\left(\frac{t}{T_P}\right)^3\right) & 0 \le t \le T_P \\ A\left(1 - \left(\frac{t - T_P}{T_N}\right)^2\right) & T_P < t \le T_P + T_N \\ 0 & T_P + T_N < t \le p \end{cases} \quad (5.33)$$

where T_P and T_N are the opening time and closing time and p is the pitch-period.

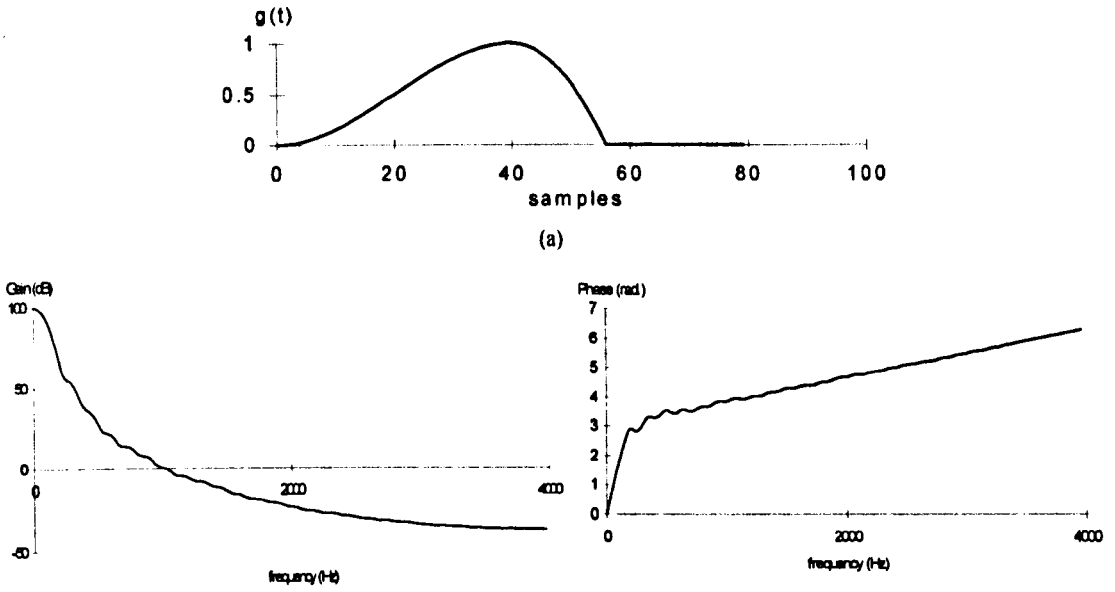


Figure 5.12 Example of Rosenberg pulse, with pitch-period = 100 samples.
(a) waveform shape (b) magnitude spectrum of the signal (c) phase spectrum of the signal

The Rosenberg pulse as illustrated in figure 5.12a is clearly not minimum phase. A minimum phase signal, in comparison to all other causal signals with the same magnitude spectrum will have its energy maximally concentrated at the beginning of the signal; i.e. for a given causal minimum phase signal {x(n)}, the sum $\sum_{n=0}^{L-1}\left(x(n)\right)^2$ for any value of L will be larger for {x(n)} than for any other signals with the same magnitude spectrum as {x(n)}. The Rosenberg pulse clearly cannot be the impulse response of the 2nd order minimum phase all-pole filter G(z) discussed earlier. However it does quite closely resemble the time-reversed impulse response of such an all-pole filter, G'(z) say. This is clearly the justification for the 2-pole glottal filter G(z) discussed earlier. An example of the time-reversed impulse response of a 2nd order all-pole filter, with poles at $\beta_1 = \beta_2 = 0.9$ is presented in figure 5.13a. The time reversal impulse response is delayed by 100 samples, after which samples for $n<0$ are assumed to be close enough to zero to be considered zero. The magnitude and phase spectra of the time reversed and delayed impulse response are shown in figures 5,13b and c respectively. It can be seen by comparing figures 5.12b and c and 5.13b and c that the two signals can have almost identical magnitude and phase spectra, if the two signals in figures 5.12a and 5.13a are properly phase aligned.

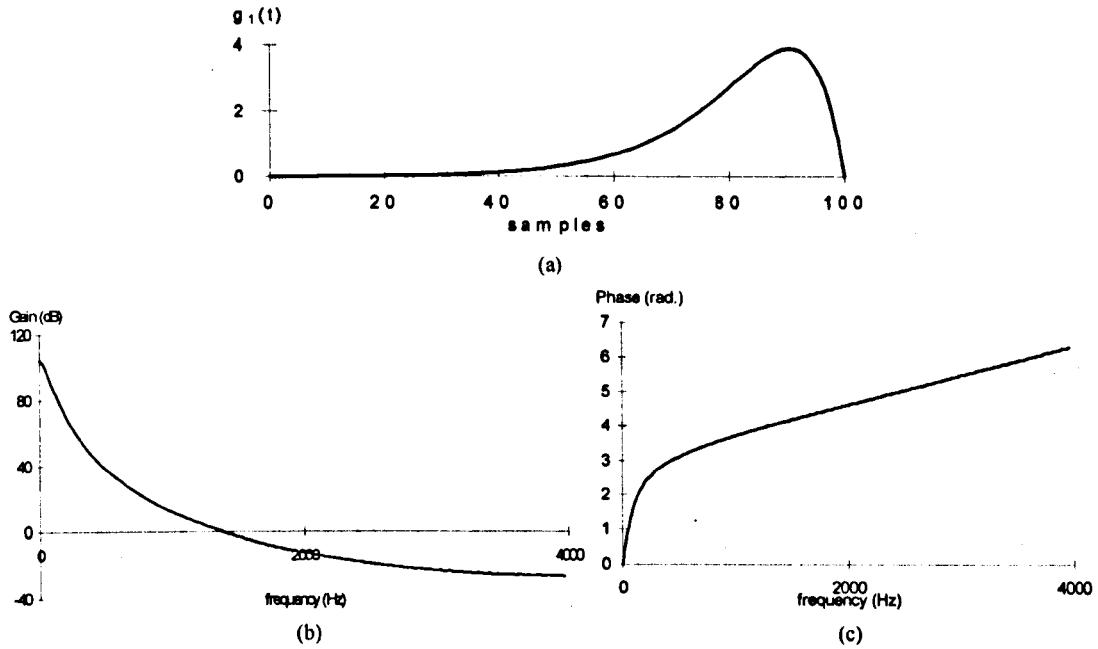


Figure 5.13 Time-reversed impulse response of a 2nd-order all-pole filter ($\beta_1 = \beta_2 = 0.9$, pitch-period=100samples). (a) wave shape (b) magnitude spectrum (c) phase spectrum

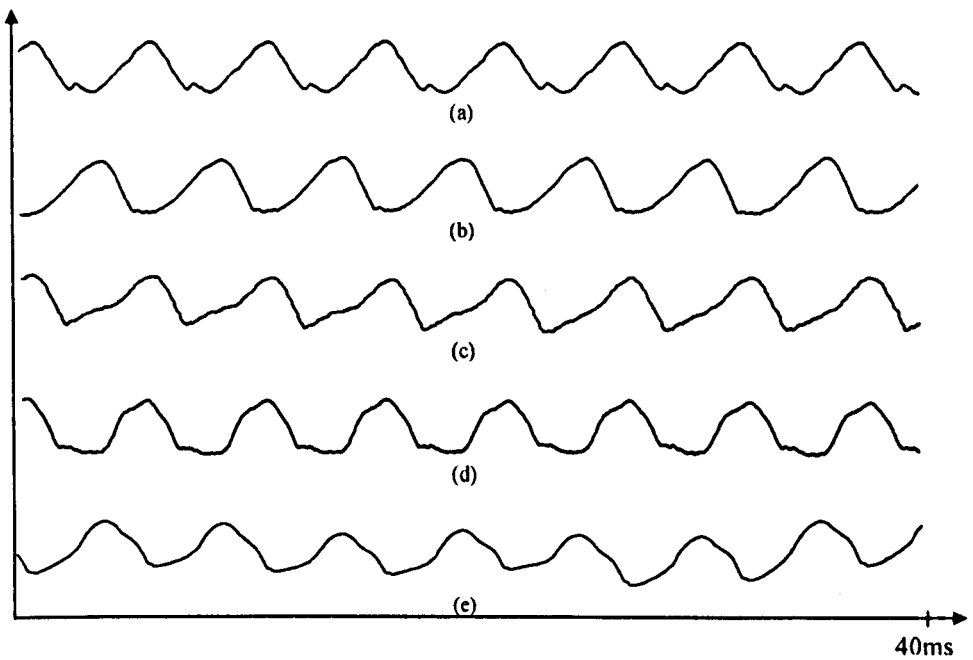


Figure 5.14 Glottal excitation signals extracted from 5 vowel sounds.
(a) /a/ (b) /e/ (c) /i/ (d) /o/ (e) /u/

In figure 5.14, the glottal excitations extracted from 5 vowels sounds, /a/, /e/, /i/, /o/ and /u/ spoken by the same speaker, are presented. The glottal excitations were extracted using the technique suggested in [78] and [79], using a 1st-order pre-emphasis filter which was a differentiator with a single zero at z=0.95. It may be seen that the glottal excitation for the vowels has a pulse-shape with some features similar

to those of a Rosenberg pulse or the time-reversed impulse response of a 2nd-order all-pole filter. Clearly there are striking differences also.

Under the assumption that the impulse response of a more realistic glottal filter G(z) resembles the time-reversed impulse response of a 2-pole transfer function G'(z), linear predictive analysis identifies the inverse of the minimum phase transfer function G'(z)V(z)L(z) rather than the inverse of the model shown in figure 5.10. This is because, denoting the magnitude and phase responses of G'(z) by $M(\omega)$ and $\phi(\omega)$ respectively, the time-reversed impulse response of G'(z) will have the same magnitude spectrum $M(\omega)$ and the phase spectrum $-\phi(\omega)$ disregarding a linear phase or delay component. Therefore LP analysis gives a good estimation of the magnitude spectrum of a glottal pulse but the LP phase spectrum will not be correct. It will have the phase contribution $\phi(\omega)$ rather than $-\phi(\omega)$. Therefore, in order to obtain an LP residual which is more likely to resemble the assumed impulse-like excitation to G(z), with its correct phase spectrum, we should augment the LP analysis filter by a 2nd-order all-pass section F(z) [80], as shown in figure 5.15a, where F(z) is defined as,

$$F(z) = \frac{\left(1 - \frac{1}{\beta_1} z^{-1}\right)\left(1 - \frac{1}{\beta_2} z^{-1}\right)}{\left(1 - \beta_1 z^{-1}\right)\left(1 - \beta_2 z^{-1}\right)} \tag{5.34}$$

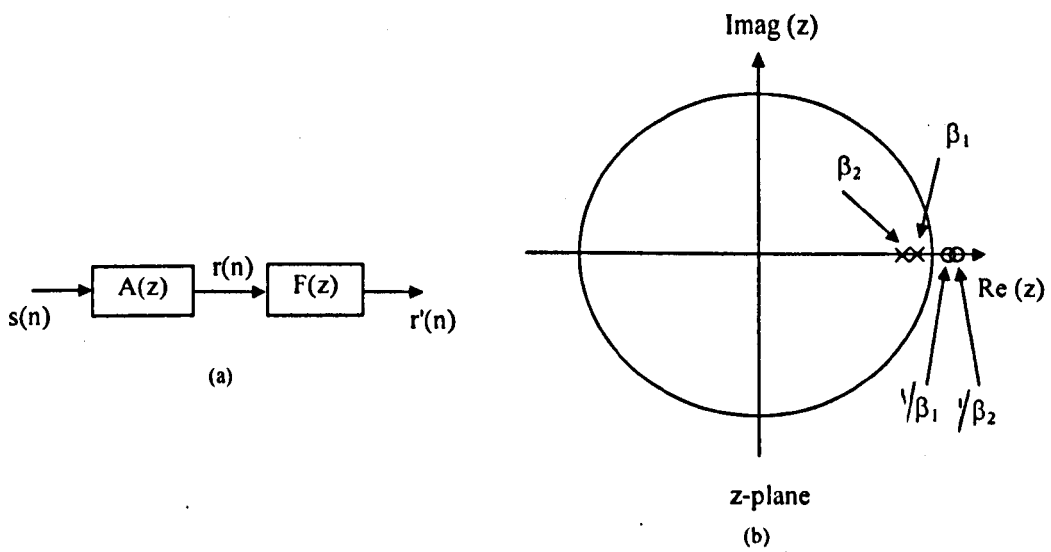


Figure 5.15 (a) An inverse filtering model for deriving the correct magnitude and phase spectrum of a speech residual. A(z) is found by LP analysis and F(z) is an all-pole filter as described in section 5.6.1 (b) pole-zero plot of the 2nd order all-pass filter F(z)

The poles and zeros of F(z) are as illustrated in figure 5.15b. The gain of F(z) is unity for all frequencies because the poles and zeros are in "mirror image" positions on the z-plane. The poles at β_1 and β_2 within the unit circle are intended to cancel out the incorrectly placed minimum phase analysis filter zeros obtained from the LP analysis. These minimum phase zeros are then effectively replaced by the non-minimum phase zero at $z=1/\beta_1$ and $z=1/\beta_2$ which more accurately model the glottal excitation. The phase spectrum of F(z) is computed as,

$$\begin{aligned}\phi_F(\omega) &= \tan^{-1}\left(\frac{\sin\omega}{\beta_1-\cos\omega}\right)+\tan^{-1}\left(\frac{\sin\omega}{\beta_2-\cos\omega}\right)-\tan^{-1}\left(\frac{\beta_1\sin\omega}{1-\beta_1\cos\omega}\right)-\tan^{-1}\left(\frac{\beta_2\sin\omega}{1-\beta_2\cos\omega}\right)\\ &=2\left(\tan^{-1}\left(\frac{\sin\omega}{\beta_1-\cos\omega}\right)+\tan^{-1}\left(\frac{\sin\omega}{\beta_2-\cos\omega}\right)\right)\\ &=-2\left(\tan^{-1}\left(\frac{\beta_1\sin\omega}{1-\beta_1\cos\omega}\right)+\tan^{-1}\left(\frac{\beta_2\sin\omega}{1-\beta_2\cos\omega}\right)\right)\end{aligned} \tag{5.35}$$

5.6.2 The effect of the voiced speech production model on an LP residual signal

The effect of the all-pass section F(z) on an LP residual is illustrated in figure 5.16. In the experiment, the parameter β_1 was set equal to the zero assumed for the lip-radiation filter α at 0.95. The value of β_2 was found by an optimisation procedure. A prototype waveform was extracted from the voiced residual, pitch-synchronous DFT analysis was applied to the prototype waveform to yield its phase spectrum $\phi_U(\omega)$ which was then matched as closely as possible by the negated phase spectrum $\phi_F(\omega)$ of F(z). Linear phase components in $\phi_U(\omega)$ must be accounted for, by phase aligning $-\phi_F(\omega)$ with $\phi_U(\omega)$ using the original magnitude spectrum. The closeness of $\phi_U(\omega)$ to $-\phi_F(\omega)$ was measured by a sum of squared phase differences. This was minimised by considering a range of values of β_2 and selecting the value which produced the lowest value.

After optimising β_2, the input speech shown in figure 5.16a was processed by a 10th order LP inverse filter and the all-pass filter. The resulting signal is shown in figure 5.16c. This graph illustrates that by passing the original LP residual signal

through F(z), the modified voiced residual can be made closer to an impulse train than the original LP residual shown in figure 5.16b.

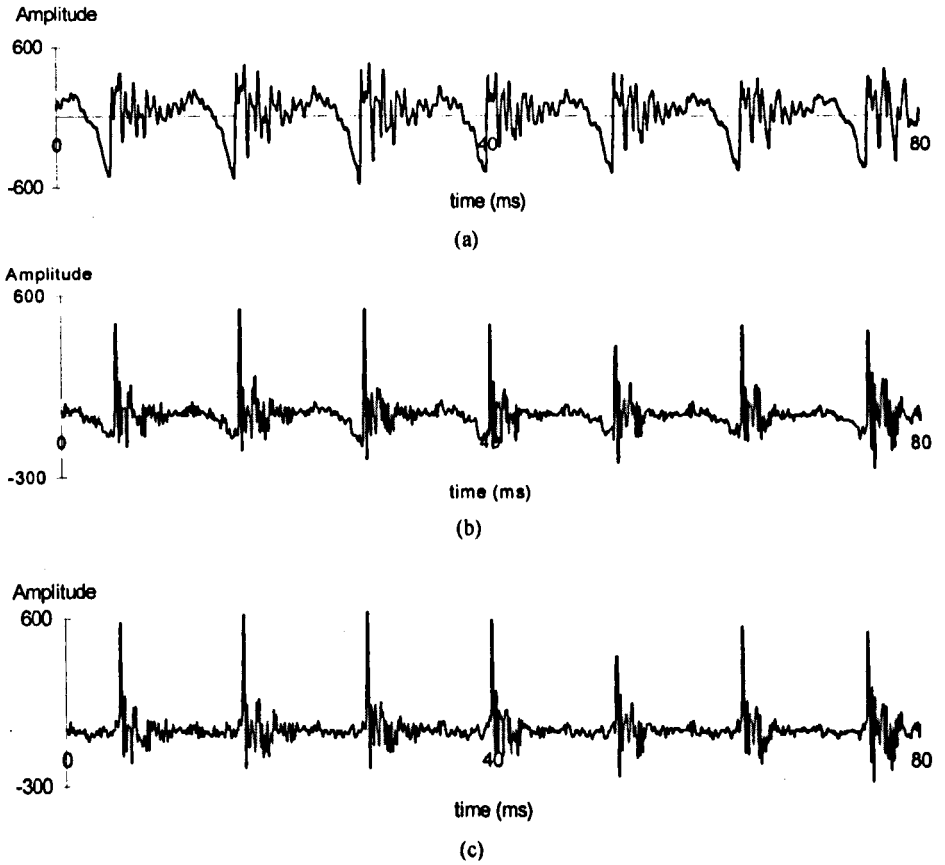


Figure 5.16 The effect of F(z) on an LP residual signal.
(a) original speech waveform (b) LP residual (c) residual signal after the 2nd order all-pass filter

5.6.3 Evaluation of the phase deviation scheme using synthetic speech

The phase model was also tested using synthetic voiced speech. The synthetic voiced speech segment shown in figure 5.17b was generated by the voiced speech production model shown in figure 5.10, with the period of the impulse-train excitation set to 150 samples. The glottal excitation was made to be a Rosenberg pulse train, as shown in figure 5.17a, by making G(z) an FIR filter whose impulse response was the required Rosenberg pulse. The constants T_P and T_N of the Rosenberg pulse are normally considered to be proportional to the period p and were in this case set at 0.4p and 0.16p, i.e. 60 samples and 20 samples respectively [77]. The vocal tract transfer function V(z) was defined to be 8th order all-pole model with 8 poles located within the unit circle. The pole positions were: $0.97e^{\pm j0.196}$,

$0.85e^{\pm j\ 1.178}$, $0.7e^{\pm j\ 1.767}$ and $0.5e^{\pm j\ 2.592}$. For the lip-radiation filter with transfer function $L(z)=1-\alpha z^{-1}$, α was set as 0.95.

The synthetic voiced speech was applied to a 10th order LP analyser to estimate its spectral envelope. The LP analyser implemented the autocorrelation LP analysis with a 240 samples Hamming window. The centre of the Hamming window is located at an update-point. The synthetic voiced speech was then passed through a 10th order LP analysis filter, to yield a residual signal which is shown in figure 5.17c.

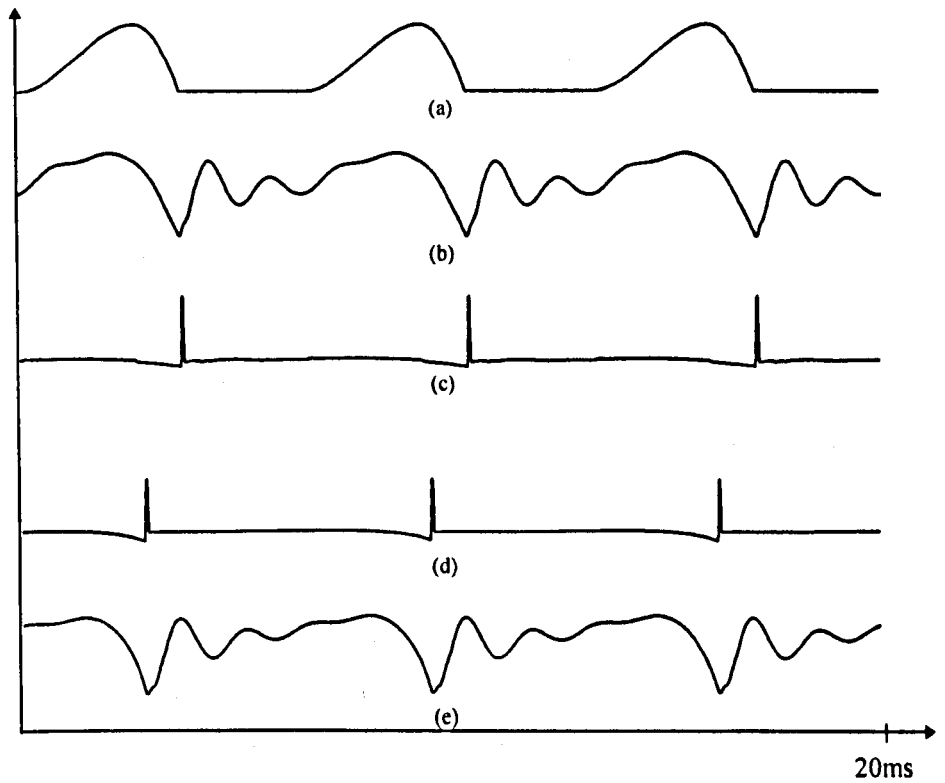


Figure 5.17 Example of synthesised voiced speech using the phase model
(a) glottal excitation (b) original voiced speech (c) residual signal (d) reconstructed speech excitation (e) reconstructed voiced speech

To recover the voiced speech, a period-length prototype waveform was extracted from the residual. DFT analysis of order p was performed on the prototype waveform to yield magnitude and phase spectra. The true DFT phase spectrum was discarded and the magnitude spectrum was used together with appropriate samples of $-\phi_F(\omega)$ to reconstruct the residual signal. The value of β_1 was taken to be 0.95 and a value of β_2 was found by solving for the zeros of the inverse filter transfer function

A(z), and choosing the real root which was closest to the unit circle for β_2. Using this approach, β_2 was found to be approximately 0.95 and the reconstructed residual was found to be as shown in figure 5.17d. The reconstructed residual is very close to the original shown in figure 5.17c. A time shift is present as expected due to the location of the prototype waveform extracted. When the reconstructed residual was passed through the LP synthesis filter, the synthesised speech shown in figure 5.17e was obtained. This is very close to the original voiced speech in figure 5.17b.

5.6.4 Incorporation of the phase derivation scheme into the TPSWI coder

An all-pass augmentation of the LP filter is not necessarily required at the analysis stage of a TPSWI coder, since the correct magnitude spectrum of the residual is all that is encoded. The all-pass phase spectrum need only be applied at the decoder to make the output speech correspond more closely to a naturally shaped glottal pulse excited speech production model.

In experiments with natural speech, a value of 0.95 was chosen for β_1 and β_2 was obtained in a more practical way than the method used in the earlier investigations. Suitable values for β_2 were obtained by matching a time-reversed Rosenberg pulse with the output of a 2nd-order all-pole filter, using a cross-correlation function. The Rosenberg pulse was adjusted according to the pitch-period p with parameters, T_P=0.4p and T_N=0.16p [77]. The values of β_2 found to be appropriate across the range of possible pitch-periods are listed in table 5.1.

pitch-period	β_2	pitch-period	β_2	pitch-period	β_2	pitch-period	β_2
16 - 52	0.64	65 - 68	0.76	82 - 84	0.84	103 - 107	0.90
53 - 54	0.65	69	0.78	85 - 87	0.85	108 - 114	0.91
54 - 56	0.66	70 - 72	0.79	88 - 89	0.86	115 - 124	0.92
57 - 59	0.70	73 - 74	0.80	90 - 93	0.87	125 - 132	0.93
60 - 62	0.71	75 - 79	0.82	94 - 99	0.88	133 -144	0.94
63 - 64	0.75	80 - 81	0.83	100 - 102	0.89	145 - 150	0.95

Table 5.1 Values of β_2 for different ranges of pitch-periods (in samples)

Although, in theory β_2 is embedded in the inverse LP filter parameters and so may be found by a method based on that used for artificial speech, experiments have been

shown a real root close to z=1 may not be always available. Although the approach may be modified to find appropriate values of β_2 on these occasions, the computational complexity involved in root solving and the additional complexity involved in processing these roots makes the approach very unattractive. Particular problems in finding the required root close to z=1 occur with telephone band speech which is high pass filtered below about 300Hz.

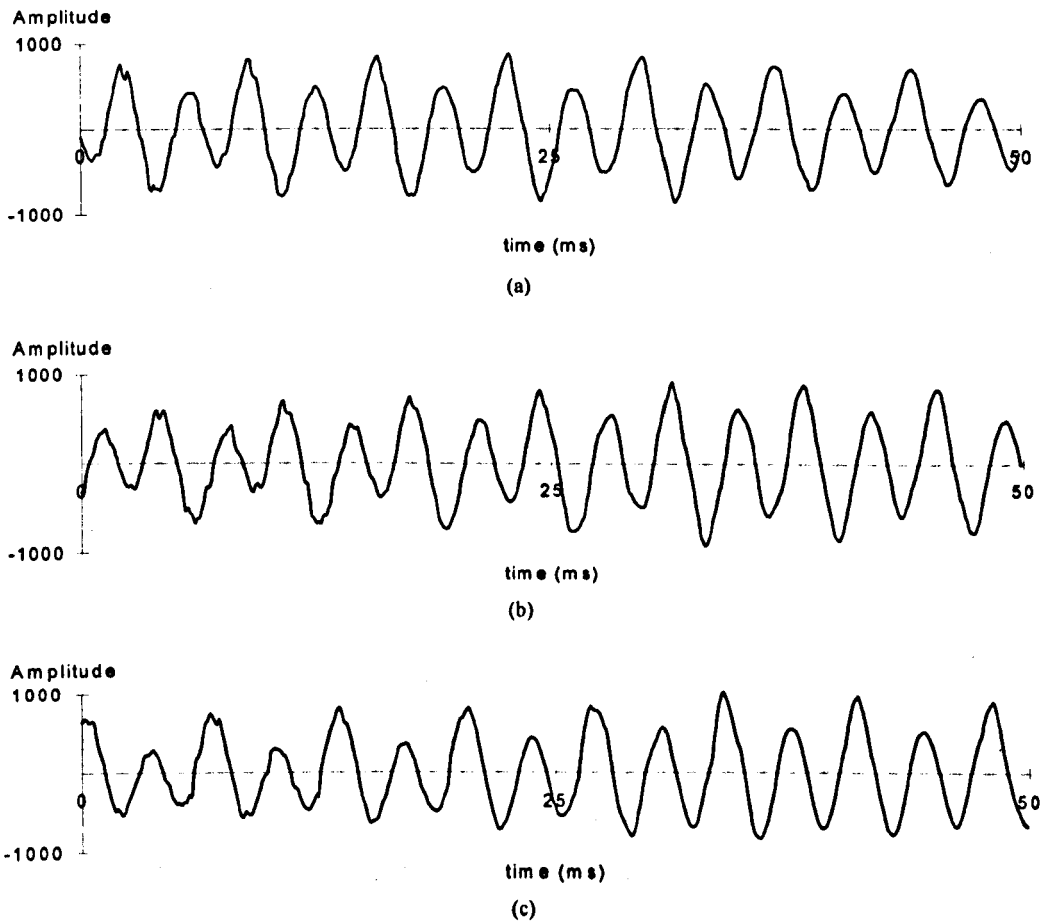


Figure 5.18 Synthesised voiced speech obtained by the two phase models.
(a) original speech signal (b) speech signal obtained using the original phase spectrum (c) speech signal obtained from the 2nd-order all-pass phase model

An example of a voiced speech segment synthesised using a TPSWI decoder is shown in figure 5.18. It can be seen that the synthesised speech obtained using the original magnitude and phase spectra, as shown in figure 5.18b, is very similar in shape to the original signal, as shown in figure 5.18a, though with the expected time shift. It may be seen in figure 5.18c that the speech waveform synthesised using the derived phase spectrum also closely resembles the original waveform.

5.6.5 Subjective evaluation of the new approach

When tested with female speech in the file "OPERATOR.DAT" [21] the new phase derivation scheme, incorporated into the TSPWI coder, was found to produce only minor degradation as compared with what was obtained using the true phase spectrum. However, a slightly unnatural speech quality was perceived when the pitch-period of the voiced speech was larger, as in the male speech. It was found that the quality of male speech was improved by randomising the phases at higher frequencies within the 0 to 4kHz bandwidth. The frequency range over which randomised phases rather than derived phases are used must be carefully chosen. A roughness in speech quality results if too many components with randomised phases are introduced. An adaptive frequency boundary was found to be effective and after much experimentation it was decided to make this equal to the frequency of the 9th LSF. The choice of the 9th LSF was made to arrange that the first three formants are in the derived phase region. As a result, the phase model is defined as,

$$\phi'_F(k\varpi) = \begin{cases} -\phi_F(k\varpi) & 0 \le k\varpi \le \omega_9 \\ \phi_r(k) & \omega_9 < k\varpi < \pi \end{cases} \qquad (5.36)$$

where

ω_9 is the 9th LSF (in radian per sample)

ϖ is the pitch-frequency (in radian per sample)

ϕ_F is the phase spectrum of the 2nd-order all-pass filter

$\phi_r(k)$ is a uniformly distributed pseudo-random phase between $-\pi$ to π, a different random phase being generated for each frequency bin k,

Using the new derived phase spectrum $\phi'_F(\omega)$, the TPSWI coder was tested again using the speech file "OPERATOR.DAT" [21]. Informal listening tests suggested that the perceptual quality of the synthesised male speech was improved by using the new phase spectrum $\phi'_F(\omega)$ instead of directly taking the phase spectrum $-\phi_F(\omega)$ derived from the 2nd order all-pass filter. No audible difference was found between synthesised female speech segments generated using, on the one hand, the phase spectra $\phi'_F(\omega)$ and on the other hand $-\phi_F(\omega)$.

5.7 Conclusions

A two-mode pitch-synchronous waveform interpolation (TPSWI) model designed during the course of this project has been described in the chapter. In this model, a pitch-synchronous PWI technique is used for voiced speech and unvoiced speech is modelled very simply by a power contoured pseudo-random sequence generator. Representing the contour accurately was found to be very important and much more important than trying to accurately model the residual waveform and any residual periodicity as CELP does. An overlap-add technique was found to be necessary to ensure smooth transitions between voiced and unvoiced speech. When the TPSWI coder was compared with a PWI/CELP coder developed by the author, smoother voicing transitions were found in the TPSWI coder and these greatly enhanced the perceived quality of the speech obtained. Furthermore, the computational complexity of the TPSWI coder for unvoiced speech is much less than that of the PWI/CELP coder.

The TPSWI coder has regular update-points at 20ms intervals. At each update-point a set of ten LSF coefficients and an estimate of the instantaneous pitch-period at the current update-point are encoded. Confirming the idea published by Kleijn [70], it was found that for voiced speech a very close approximation to the original speech was obtained when eight prototype waveforms were encoded per 20ms duration. Each prototype waveform is ideally characterised using a gain-shape approach where a gain factor and a normalised prototype waveform are separately encoded. Subjective tests suggested that no noticeable speech degradation occurs when only two gain factors are sent per 20ms, the missing gain factors being estimated by interpolation. Ways of effectively encoding the shape information will be discussed in a later chapter.

A theoretical model of human voiced speech production which, in principle, enables the phase spectra of prototype waveforms to be derived has been presented. This has implications towards the search for low bit-rate representation of prototype waveforms. It was concluded that the phase spectrum may be derived using a 2nd order all-pass filter thus potentially enabling the coding efficiency of the TPSWI coder to be increased by encoding only the magnitudes of the prototype waveforms.

Subjective tests suggested that even with the all-pass phase modelling a somewhat unnatural speech quality was produced by the TPSWI coder when the pitch-period of the voiced speech was large. It was found that in order to improve the speech quality, the phases of some of the higher frequency pitch harmonics may be randomised. Informal listening test suggested that the perceptual quality of the synthesised male speech was enhanced by this procedure and that no audible difference was occurred in the synthesised female speech. It was decided that frequency boundary above which harmonics have their phases randomised should be taken to be the 9th LSF as obtained for each update-point.

Chapter 6

Generalised Pitch-Synchronous Waveform Interpolation (GPSWI) model

6.1 Introduction

PWI coding was initially proposed [68] for voiced speech segments only. Unvoiced speech segments were originally intended to be coded by switching to a fundamentally different model such as a simplified form of CELP, or, even more simply, a suitably power contoured pseudo-random sequence. A voiced/unvoiced decision was thus required for choosing the appropriate model for a given segment of speech. Inevitably with such a coder speech quality degradation will occur from time to time due to speech classification errors. A very accurate voiced/unvoiced classifier is required to minimise the rate of occurrence of such errors. Deterioration of speech quality could be caused by mishandling of the switching between the two coding techniques. Also the 50Hz sampling of the waveform shape (by extracting prototype waveforms every 20ms) is used without regard to the nature of changes that may be occurring to the shape. More rapid elements of the changes will clearly cause a form of aliasing and further, the mis-representation of the more rapid changes in the synthesised speech can lead to excessive buzziness and unnaturalness. These difficulties provided the motivation for trying to achieve a generalised coding scheme which is capable of handling both voiced and unvoiced speech and which samples and represents the evolving speech or residual waveform in terms of different aspects of the waveforms which evolve at different rates and are perceived differently. Recently, such a generalised waveform interpolation (WI) coding algorithm was proposed by Kleijn *et al* [73]-[76] which is claimed to be able to achieve good speech quality at around 2.4kb/s.

The new coding algorithm generalises the concept of a prototype waveform to include fixed length segments of unvoiced speech or transitions. The generalised

prototype waveforms are called "characteristic waveforms". The new approach also aims to take account of the non-stationarity of speech and the way it is perceived by decomposing the changes that occur to characteristic waveforms into slowly evolving (SEW) and rapidly evolving (REW) waveform components [73]. The SEW components arise from the quasi-periodic content of the speech residual. The REW components arise from degrees of randomness embedded in all forms of speech but particularly in unvoiced speech [73]. For voiced speech segments, the SEW components will tend to be dominant whereas for unvoiced speech, the REW components will be more important. By modelling the SEW and REW components simultaneously and individually, the required generalisation of PWI techniques to random-like signals and an improved sampling and interpolation scheme for voiced speech is obtained.

It was suggested [74] that the human auditory system has a very different requirement for the perceptually indistinguishable reconstruction of the SEW (quasi-periodic) and REW (random-like) components. Coding efficiency can be gained by exploiting this characteristic of human perception. For SEW (quasi-periodic) components, the wave-shapes are likely to be important and generally need to be reproduced with some accuracy. For REW (random-like) components, the substitution of quite dissimilarly shaped pseudo-random waveforms is acceptable as long as the average spectrum and power contour are reasonable. Different update rates and quantisation schemes are used for the SEW (quasi-periodic) and REW (random-like) components.

A more recent type of waveform interpolation coder proposed by Kleijn *et al* [76] modifies the way in which characteristic waveforms are extracted from the residual and also encodes magnitude only information rather than the Fourier series cosine and sine coefficients. The missing phase information is regenerated artificially at the decoder. The quantisation schemes for the SEW and REW were refined to take into account their spectral characteristics and the way different spectral characteristics are perceived. The general spectral characteristics of a full-band REW are encoded at intervals of 2.5ms and the SEW in the range 800Hz to 4kHz is deduced by subtracting the REW in this band from a flat spectrum. Below 800Hz, the

SEW is encoded at intervals of 25ms by vector quantisation. Techniques for reducing the complexity of the coder were published [76] and these have led to real time implementations both on a DSP device and also on a general purpose workstation.

The main findings of Kleijn [74] are therefore that coding efficiency can be gained by separating the quasi-periodic and random-like components of the residual and that the quasi-periodic components and the random-like components may be regenerated at the decoder by interpolation and the substitution of pseudo-random sequences respectively. It was found that for voiced residuals the general shapes of the characteristic waveforms are slowly evolving over time and more detailed information embedded onto the general shapes is very different from one characteristic waveform to the next. For unvoiced residuals the general shape tends to disappear and the characteristic waveforms are very different from one to another. Decomposition of characteristic waveforms into quasi-periodic and random-like components can be achieved by exploiting the fact that the general shape of a characteristic waveform is characterised mainly by its lower frequency components and that its higher frequency components contribute mainly to its finer detail. A generalised pitch-synchronous waveform interpolation (GPSWI) model was devised [84] making use of this property of characteristic waveforms. It was developed from the TPSWI model described in chapter 5. The new approach now eliminates the switching to a pseudo-random sequence generator for unvoiced speech.

In the GPSWI model, eight characteristic waveforms are extracted from the 20ms residual segment between each pair of adjacent update-points. DFT analysis is performed on each characteristic waveform to yield the magnitude spectrum. Each DFT magnitude spectrum is then decomposed into a slowly evolving spectrum (SES) and rapidly evolving spectrum (RES). The SES is obtained by averaging eight successive magnitude spectra and is only defined from 0Hz to a separation frequency which is variable. The RES for each characteristic waveform is acquired by directly taking the higher frequency portion of the magnitude spectrum of the characteristic waveform at each sub-update point. The separation frequency is taken to be the 9th LSF for fully voiced speech and reduces towards 0Hz for unvoiced speech. The SES is used to characterise the general shape of the characteristic waveforms and is

encoded at a lower rate than the RES. The RES describes the finer detail of each of these characteristic waveforms and must be sampled at a higher rate to allow close tracking of the waveform dynamic. The phase spectra for the SES and RES are imposed artificially at the decoder. The phase spectra for the SES components are derived from the voiced speech production model discussed in chapter 5. For the RES components a pseudo-random phase spectrum is used. The spectral division of the characteristic waveforms in terms of the SES and RES leads to considerable economy in bit-rate and has led to good quality speech being obtained at 2.4kb/s. Full details of the GSPWI model are given in this chapter and its quantisation is discussed in chapter 7.

The sub-band approach to the decomposition of characteristic waveforms into slowly and rapidly evolving components is original, and allows good quality speech to be obtained from the GPSWI coder at 2.4kb/s. The main aim of this chapter is to present details of the design of the GPSWI coder. Firstly, the fundamental ideas of WI coding and its more recent developments will be briefly discussed in section 6.2. In section 6.3, a general description of the SES/RES decomposition scheme will be presented. The discussion in section 6.3 will begin with the spectral decomposition algorithm used for voiced speech and the algorithm will then be further developed to cover unvoiced speech. The structure and details of the GPSWI coder will then be presented in section 6.4.

6.2 Fundamentals of waveform interpolation (WI) coding

The generalised WI coding approach was developed from the concept of PWI. A single characteristic waveform is extracted from an LP residual at predefined update intervals, say every 2.5ms [75]. The terminology "prototype waveform" used in PWI coding is replaced by "characteristic waveform" in order to cover both voiced and unvoiced speech segments. The length of each characteristic waveform, even for unvoiced speech, is equal to the value of pitch-period given by a pitch track [74]. Following the explanation given in Kleijn [74], characteristic waveforms extracted at

2.5ms intervals may be normalised in time such that the length of each waveform always becomes 2π. The normalised time axis may then be referred to as the phase axis ϕ. The time-scale normalised characteristic waveforms are power normalised and then "phase aligned" by circularly shifting them in time (referred to as phase) to maximise the cross-correlation between them, thus making them as similar in shape as possible. Each of the phase-aligned time and power normalised waveforms is placed on a phase axis ϕ as shown in figure 6.1. A 2-D plane is then produced by interpolating the samples of the waveforms between the 2.5ms spaced sub-update points.

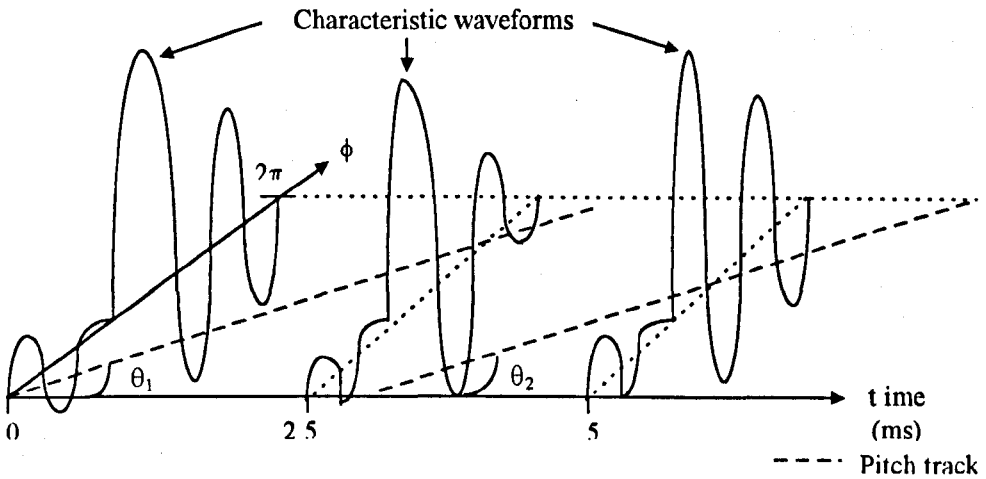


Figure 6.1 Periodic extension of a characteristic waveform in both the time and ϕ axes.

The time-normalised and power normalised characteristic waveform defined at a given point in time t, is therefore considered to be one period along the ϕ axis of a waveform which is periodic in ϕ (with period 2π) with Fourier series coefficients which depend on t as expressed by the following equation [74]:

$$u(t,\phi) = \sum_{k=0}^{p(t)-1} c_k(t)\, e^{jk\phi} \qquad (6.1)$$

where $c_k(t)$ are the complex Fourier series coefficients and p(t) is the pitch-period at time t.

The process of reconstructing the time-normalised residual by interpolation at the decoder can now be interpreted as re-sampling the 2-D surface along the diagonal lines shown dashed in figure 6.1. These dashed lines are referred to as pitch-tracks.

The angle of each diagonal line to the true time axis t, marked θ_1, θ_2, etc. is determined by the interpolated pitch-period. If the pitch-period is fixed, the lines will be straight, otherwise they will be slightly curved. The time taken for ϕ to increase from 0 to 2π along such a line is essentially equal to the pitch-period.

The accuracy of the reconstructed residual naturally depends on the update rate of the characteristic waveforms. It was reported [75] that the decoded speech obtained from a WI coder can be made approximately toll quality if the residual characteristic waveforms are encoded at sub-intervals of 2.5ms. To encode the characteristic waveforms efficiently, they are decomposed into slowly evolving (SEW) and rapidly evolving (REW) waveform components [73][74]. A different quantisation scheme is applied to each of the two component types based on known properties of human perception.

The human auditory system appears to have very different requirements for the perceptually acceptable reconstruction of the two types of waveform components. The SEW components represent quasi-periodic properties of the residual, the evolution of which is relatively slow. The SEW components can be sampled at a lower rate than the REW components, and, in principle, it appears possible to define the SEW components (i.e. to arrange the low-pass filtering) such that the original SEW information about the residual can be adequately recovered by interpolation from 20ms or 25ms spaced samples. An accurate description of the down-sampled SEW components which constitute a time-domain waveform referred to simply as the "SEW" is important. The SEW is therefore a slice of the smoothed 2-D plane taken parallel to the ϕ axis at intervals normally of 20ms.

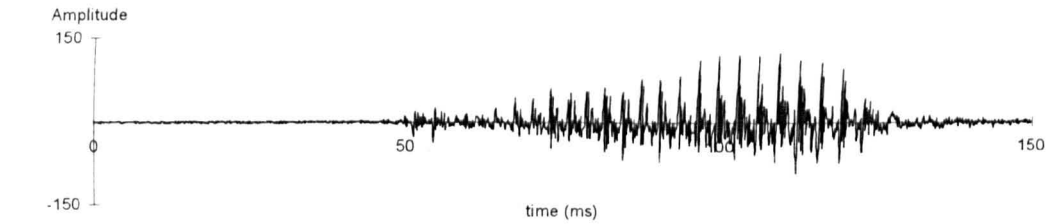
The REW components are much more random and rapidly changing than the corresponding SEW components and therefore can, in principle, carry much more information (the bandwidth of the evolution is much larger). Fortunately, human perception does not discern much information from such rapidly evolving signals apart from the fact that they are random, perhaps spectrally coloured and of varying amplitude. From the perception point of view, the perceptual quality of the REW

components can be preserved by substituting pseudo-random signals each with a combination of a roughly similar magnitude spectrum and a similar power contour to the original [75]. Because of the more rapid evolution, a higher update rate is required for the REW information than for the SEW (typically 2.5ms) though much less information needs to be provided at these updates. REW components, which constitute a time-domain waveform referred to simply as the "REW" may be regenerated at the decoder by inverse transforming the magnitude spectrum specified for the REW with a pseudo-random phase spectrum. A time-domain power contour is then imposed on the resulting signal.

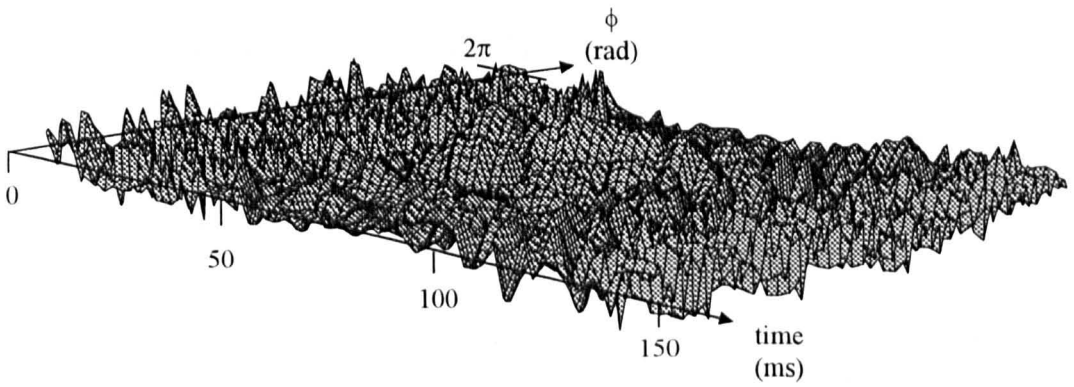
In a more recent type of WI coder proposed by Kleijn *et al* [76], characteristic waveforms are extracted from the LP residual only at the rate of about one per pitch-period. In principle, the extracted waveforms no longer overlap as they usually did with 2.5ms sub-update points. In practice, there may still be a very small amount of overlap due to changes in the pitch-period and slight inaccuracy in the interpolated pitch-period. The power and length of each characteristic waveform are, as before, normalised to unity and 2π respectively. The normalised characteristic waveforms are phase-aligned and plotted parallel to the ϕ axis on a 3-D graph of amplitude against phase and time. For each characteristic waveform a fixed value of time t is used corresponding to its update point. The update-point is the value of time t at the centre of the extracted waveform in the original residual signal. The update-points for the extracted waveforms are no longer regularly spaced in time, though this does not affect the main update-points which remain fixed, now at 25ms rather than 20ms. Samples of the normalised characteristic waveforms corresponding to the same phase ϕ may now be interpolated, as before, to obtain a 2-D surface as illustrated in figure 6.3b. The interpolation is a little more complicated now because of the unevenly spaced update-points. In practice, the 2-D surface is up-sampled in the time (t) domain but only to 1000Hz rather than 8kHz to save computation. The up-sampling is achieved by linearly interpolating between the phase-aligned characteristic waveforms at adjacent update-points. The up-sampled 2-D plane is now decomposed into slowly evolving and rapidly evolving planes by low-pass and high-pass filtering along the t axis for each value of ϕ. In fact the filtering is applied

to each of the Fourier series coefficients associated with each characteristic waveform rather than to the time-domain sample. If the filtering is applied to the time-evolution of the real and imaginary DFT coefficients (which are proportional to the Fourier series cosine and sine coefficients) rather than to the time-evolution of their magnitudes and phases, this gives the same result as applying the filtering to the time-evolution of each of the phase (ϕ) domain samples. The filtering is applied to the Fourier series coefficients simply for convenience since the coefficients must be derived anyway as part of the time normalisation process.

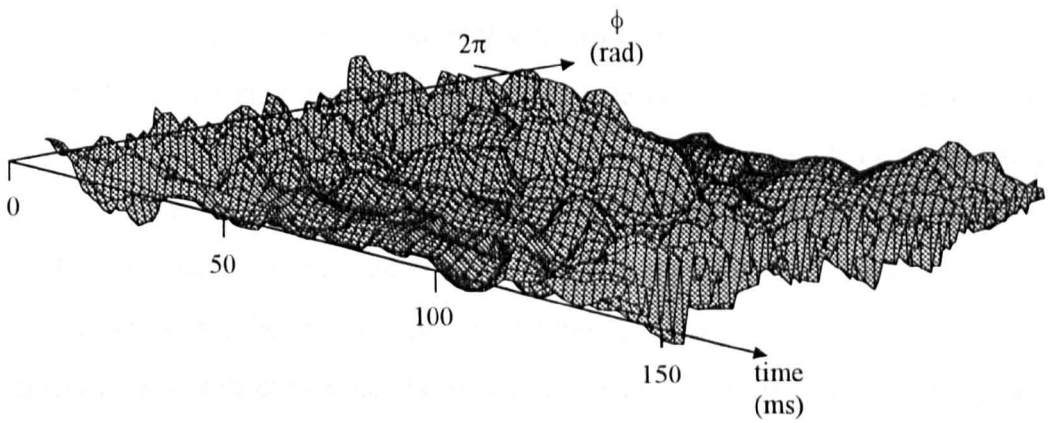
Figure 6.2 illustrates such a waveform decomposition scheme applied to the word "Yes" spoken by a female speaker. The LP residual signal is shown in figure 6.2a. Characteristic waveforms were extracted from the residual signal and were power normalised, time-scale normalised, phase aligned and interpolated in time t at 1kHz to form the 2-D time-evolution normalised characteristic waveforms plane shown in figure 6.2b. Applying one-dimensional low-pass and high-pass filtering to the time-evolution of each waveform sample (or Fourier series cosine and sine coefficient) the SEW and REW planes presented in figures 6.2c and d were generated. It is seen in figure 6.2c that the SEW components are changing quite gradually over time. The smooth characteristics of the SEW surface allow it to be sampled at a relatively low rate and to be reconstructed at the decoder using interpolation. The sampling is done at 25ms intervals along lines parallel to the ϕ axis. Figure 6.2d shows that the REW surface is random-like for the unvoiced speech portions and is very small for the voiced speech portions. The REW surface contains too much information to be economically sampled on a waveform basis but its general features, i.e. change in amplitude and short-term spectrum reflect the speech dynamic which are important to perception and naturalness. This information needs to be sampled at a relatively high rate. The REW components are regenerated at the decoder by injecting pseudo-random phase into the magnitude spectrum of a REW. The contributions from the REW and SEW are then added together. Finally, the short-term spectral envelope is re-imposed on the resulting signal to yield the decoded speech.



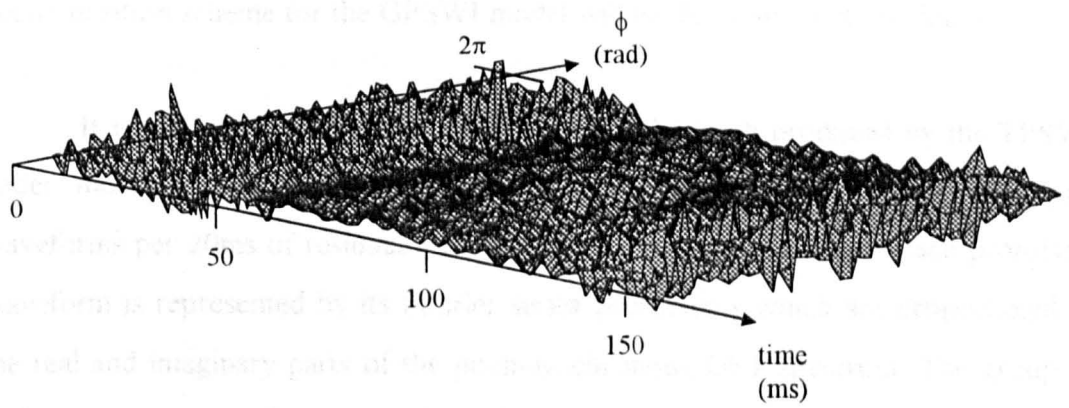
(a)



(b)



(c)



(d)

Figure 6.2 Example of waveform decomposition in a generalised waveform interpolation coder. (a) residual signal (b) time-evolution of normalised characteristic waveforms (c) slowly evolving waveforms (d) rapidly evolving waveforms

6.3 Spectral decomposition of characteristic waveforms

A new coding scheme, namely the generalised pitch-synchronous waveform interpolation (GPSWI) model, was devised during the course of this project. The GPSWI model was developed from the TPSWI model, two aspects of this development being the elimination of the switching to an alternative model for unvoiced speech and the introduction of time-evolution decomposition. The GPSWI model implements an original sub-band approach to the decomposition of the time-evolution of normalised characteristic waveforms into slowly and rapidly evolving components. The time-evolution is now viewed in terms of the spectra of the waveforms rather than the time-domain waveforms themselves.

In the new approach, the time-evolution of the magnitude spectra of the power-normalised characteristic waveforms are decomposed into slowly evolving and rapidly evolving magnitude spectra. The sub-band approach means that the magnitude spectra of the power-normalised characteristic waveforms are divided into two bands: a lower frequency band which describes the general shape of the characteristic waveforms and an upper frequency band which characterises its finer detail. The frequency boundary between the two spectral bands is made dependent on the LP filter characteristics and also on the voicing confidence level of the pitch-detector which has been discussed in chapter 2. The time-evolution of the lower frequency band will constitute the slowly evolving spectrum and that of the higher frequency band will form the rapidly evolving spectrum. In this section, the spectral decomposition scheme for the GPSWI model will be discussed in more detail.

It was reported in chapter 5 that the voiced speech produced by the TPSWI coder model is perceptually very close to the original when eight prototype waveforms per 20ms of residual are available at the synthesis stage. Each prototype waveform is represented by its Fourier series coefficients which are proportional to the real and imaginary parts of the pitch-synchronous DFT spectrum. The group of eight prototype waveforms per 20ms of residual must be decomposed into a slowly evolving spectrum (SES) and a rapidly evolving spectrum (RES) to allow efficient coding of the successive DFT spectra.

The SES is obtained by averaging the magnitude spectra of eight successive prototype waveforms and is defined from 0Hz to the frequency boundary referred to above. The SES is sampled once every 20ms. The RES is defined for the rest of the frequency band and is sampled at each sub-update point, i.e. at intervals of 2.5ms. An adaptive frequency boundary (f_{SR}) between the two bands is used to facilitate the coding of both voiced and unvoiced speech. The SES is used to represent the quasi-periodic components of the speech residual which are perceived in waveform terms and change relatively slowly and therefore may be encoded at a relatively low rate, in this case once every 20ms. The RES represents the rapidly changing information of the speech residual which is perceived in terms of more general characteristics which evolve rapidly and therefore must be encoded at a higher rate.

At the decoder, successive SES magnitude spectra are interpolated to compute an SES magnitude spectrum at each of the 2.5ms spaced sub-update points between the main update-points. At each sub-update point, the interpolated SES is combined with the corresponding RES to yield the magnitude spectrum of a modified power-normalised characteristic waveform. The phase spectrum of this waveform is taken as minus the phase spectrum of the 2nd order all-pass filter described in chapter 5. To reconstruct a speech residual, the phase spectra of the power-normalised characteristic waveforms are phase aligned and the real and imaginary DFT coefficients of the waveforms are computed. The real and imaginary DFT coefficients of adjacent characteristic waveforms are interpolated in the same way as for the TPSWI model. Quadratic instantaneous phase interpolation is applied to the frequency components belonging to the lower frequency band. For the higher frequency band, a randomised instantaneous phase is assigned to each DFT component at each sub-update point, before the quadratic instantaneous phase interpolation is applied in the normal way. Finally, the resulting signal is scaled to the required power and the short-term speech envelope is re-imposed to reconstruct the modelled speech.

6.3.1 Spectral decomposition of characteristic waveforms for voiced speech

In the development of the technique described above, voiced speech is regarded as a quasi-periodic signal, in which only the lower frequencies of characteristic waveforms extracted from the voiced residual signal will have a low evolution bandwidth, i.e. only the change in the general shape between adjacent characteristic waveforms is assumed to be small. Experiments have been carried out to justify this approach. A voiced residual, in theory a quasi-periodic impulse train, will generally in practice be a spiky signal as illustrated in figure 6.3 for which the detailed wave-shape between successive spikes may be substantially different from one period to the next.

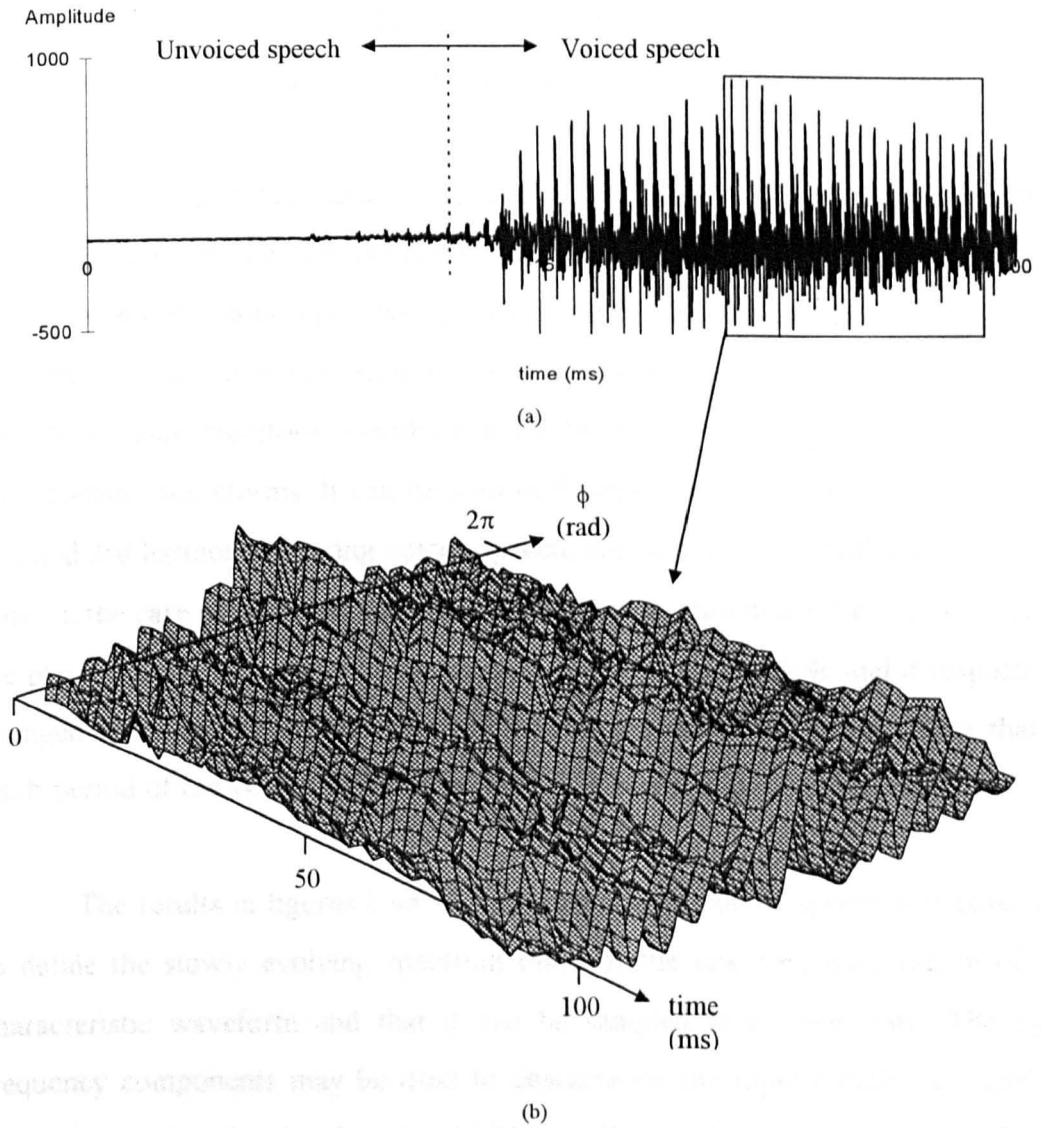


Figure 6.3 Example of a 2-D plane constructed by sampling a residual signal at a 2.5ms update rate. (a) residual obtained from part of the word "No" spoken by a male speaker (b) section of the 2-D plane during voiced speech

Figure 6.3a shows part of the residual signal for a word "No" spoken by a male speaker. The first half of the residual signal in figure 6.3a is unvoiced speech and the remaining part is voiced speech. Characteristic waveforms were extracted from the residual signal every 2.5ms and the power and length of each of these were normalised to unity and 2π respectively. A sampled 2-D plane was formed by phase aligning each normalised characteristic waveform to the previous one and plotting the phase aligned characteristic waveform against ϕ at each sub-update point on the time axis. In figure 6.3b, the 2-D plane corresponding to a segment of the voiced residual in figure 6.3a is presented. It can be seen that the detailed wave-shape is slightly different from sub-update point to sub-update point even though the general shape evolves only slowly with time. These rapid change in the fine structure in the characteristic waveforms would cause difficulties with the down-sampling of the characteristic waveform plane at 20ms and its re-generation by interpolation.

To investigate this rapid change in the fine structure, the time-evolution of the phases of particular pitch-frequency harmonics are plotted along the time axis. The phase evolution plots are investigated because the magnitude spectra of the characteristic waveforms are expected to be relatively flat [73]-[76]. The evolution of the phases may therefore contribute more to the evolution of the shapes of the characteristic waveforms. It can be seen in figures 6.4a and b that the phases of the 2nd and 3rd harmonics, during voiced speech, are slowly and smoothly evolving over time. In the case of unvoiced speech, a rapid phase evolution is observed. In contrast, the phases of the 16th and 17th harmonics, shown in figures 6.4c and d respectively, changed quite rapidly in both cases of voiced and unvoiced speech. Note that the pitch-period of the voiced speech was around 40 samples or 0.05 seconds.

The results in figures 6.4a to d suggest that in voiced speech it is reasonable to define the slowly evolving spectrum only for the low frequency region of each characteristic waveform and that it can be sampled at a lower rate. The higher frequency components may be used to characterise the rapidly evolving signal and they may need to be sampled at a higher rate in order to closely track the speech dynamic.

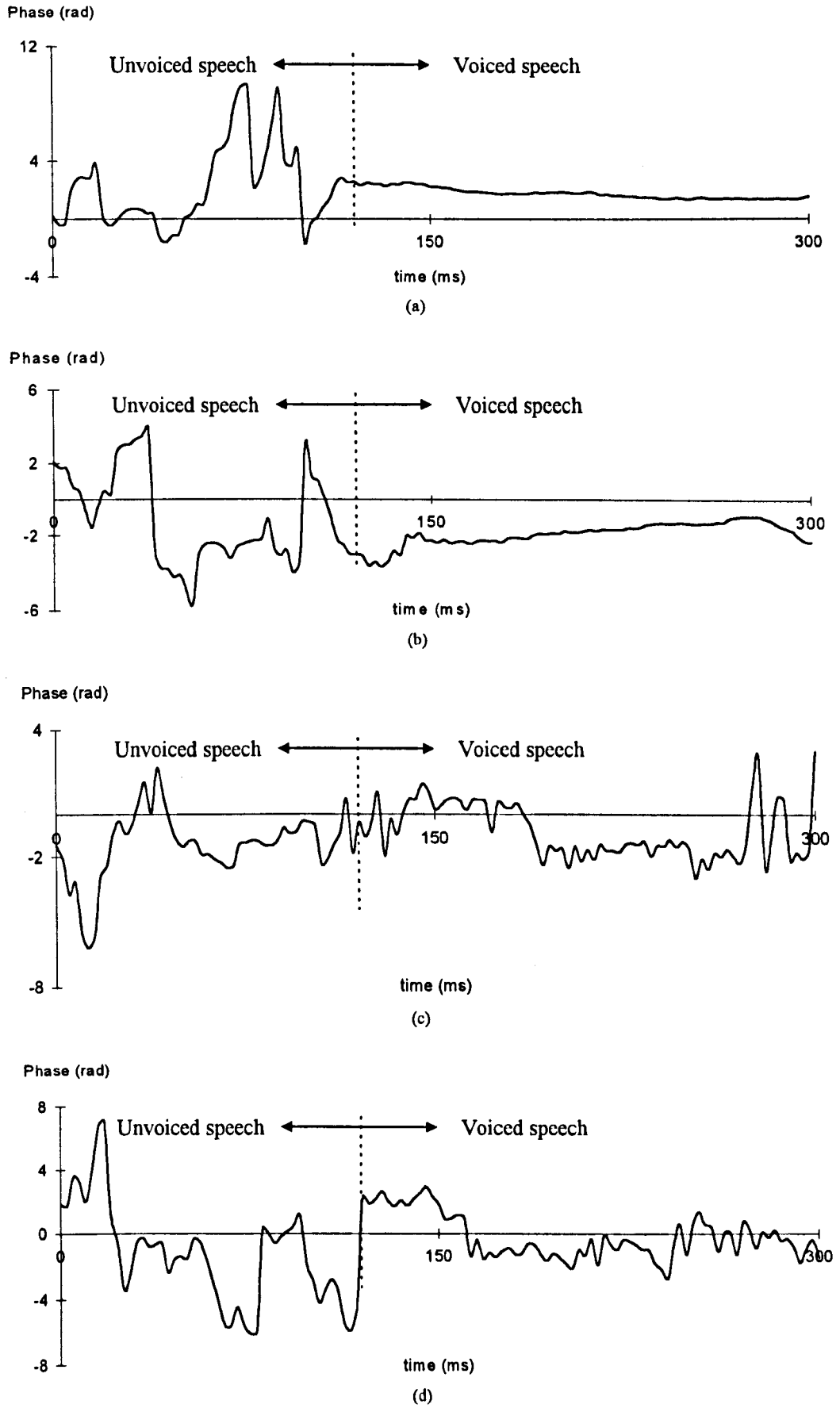


Figure 6.4 Example of how phases of some harmonics evolve with time.
(a) phase of 2nd-harmonic (b) phase of third harmonic (c) phase of 16th harmonic (d) phase of 17th harmonic (Note that the pitch-period of the voiced speech is about 40 samples)

6.3.1.1 Acquisition of a mean characteristic waveform

Suppose eight characteristic waveforms $\{u_j(n)\}_{n=0,\ p_j-1}$ for j=1, 2, ..., 8 are extracted from a residual signal at 2.5ms spaced sub-update points. DFT analysis is performed on each characteristic waveform to yield spectra $\{U_{jk}\}$. The power of each characteristic waveform is normalised to unity by normalising each magnitude spectrum according to equation 6.2:

$$|U'_{jk}| = \frac{|U_{jk}|}{\sqrt{\frac{1}{p_j^2}\sum_{k=0}^{p_j-1}|U_{jk}|^2}} \qquad (6.2)$$

$$k = 0, 1, \ldots, p_j - 1$$

where

p_j is the instantaneous pitch-period at the jth sub-update point

$|U_{jk}|$ is the kth sample of the DFT spectrum of the jth characteristic waveform, i.e. the characteristic waveform at the jth sub-update point where j=1 to 8.

$|U'_{jk}|$ is the kth sample of the power-normalised spectrum of the jth characteristic waveform.

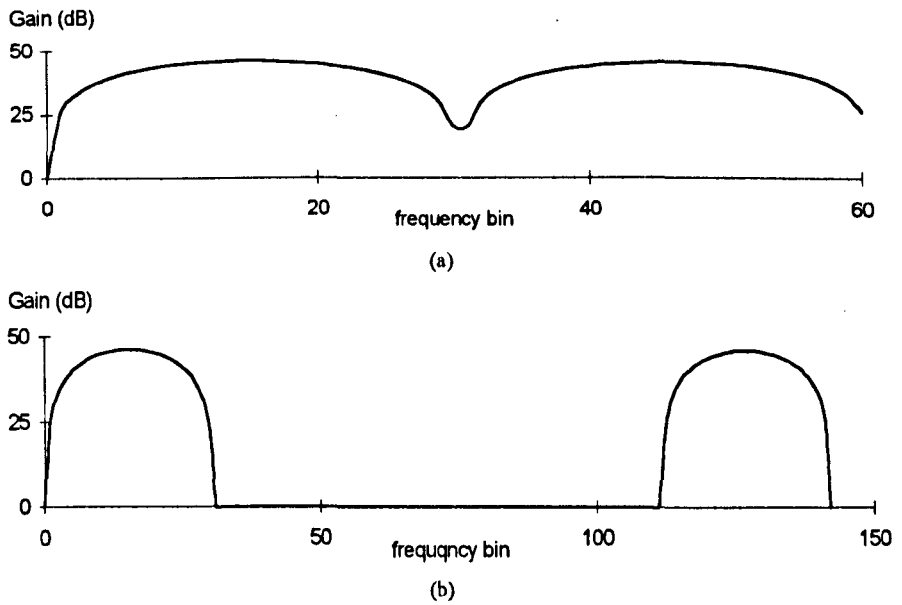


Figure 6.5 Pitch normalisation by zero padding the magnitude spectrum of a characteristic waveform. (a) original magnitude spectrum (60 DFT magnitudes) (b) zero padded magnitude spectrum (143 DFT magnitudes)

The length of each characteristic waveform is time-normalised to the maximum pitch-period p_{max}, which is equal to 143 samples for the GPSWI model.

This is done by zero-padding the original DFT spectrum to the length of the maximum pitch-period. The zero padding scheme that must be used is illustrated in figure 6.5. An original magnitude spectrum is shown in figure 6.5a, for which the characteristic waveform length was 60 samples. The zero padded spectrum for a version of the characteristic waveform time-normalised to a length of 143 samples is shown in figure 6.5b. Note that the mirror image property of the magnitude spectrum about the centre frequency sample $p_{max}/2+1$, which is equal to 72 samples, in this case must be maintained. The phase spectrum for the time-normalised characteristic waveform may be obtained by the same zero-padding scheme, except that it is anti-symmetric rather than symmetric about the centre, $p_{max}/2+1$.

To compensate for the effect of time-normalisation on the power of each magnitude spectrum, each time-normalised magnitude spectrum is modified using equation 6.3:

$$\left|U''_{jk}\right| = \frac{p_{\max}}{p_j}\left|U'_{jk}\right| \qquad (6.3)$$

$$k = 0, 1, 2, \ldots, p_{\max} - 1$$

$$j = 1, 2, \ldots, 8$$

where

$|U''_{jk}|$ is the spectrum with both time and power normalisation

p_{max} is the maximum pitch-period used in the speech coder (p_{max} is 143 samples in the GPSWI model)

The two power-normalisation processes can be combined to a single operation using equation 6.4:

$$\left|U''_{jk}\right| = \frac{\left|U_{jk}\right|}{\sqrt{\frac{1}{p_{\max}^2}\sum_{k=0}^{p_j-1}\left|U_{jk}\right|^2}} \qquad (6.4)$$

$$k = 0, 1, \ldots, p_{\max} - 1$$

When time- and power-normalised magnitude spectra have been obtained at a succession of sub-update points, a mean magnitude spectrum $|\overline{U}_k|$ is computed by averaging groups of eight normalised magnitude spectra as:

$$\left|\overline{U}_k\right| = \frac{1}{8}\sum_{j=1}^{8}\left|U''_{jk}\right| \qquad (6.5)$$
$$k = 1, 2, \ldots, \frac{p_{max}}{2}$$

where $\left|\overline{U}_k\right|$ is the kth magnitude component of the mean magnitude spectrum.

Thus a time- and power-normalised characteristic waveform which has the general shape of a group of eight consecutive time- and power-normalised characteristic waveforms has the magnitude spectrum $\left|\overline{U}_k\right|$. To produce an average phase spectrum which allows a mean characteristic waveform to be obtained via an inverse DFT, each time- and power-normalised waveform must be phase aligned. To do this, the phases for each sample on the ϕ axis must be unwrapped along the time axis t to avoid phase jumps of 2π due to phase wrapping. In the GPSWI model, the computation of such a mean time-domain characteristic waveform is not required. This saves a lot of computation. In other forms of WI coding [73]-[76] phase alignment of the characteristic waveforms must be performed prior to the waveform decomposition, at some computational cost.

Note that only half of the magnitude spectrum, $p_{max}/2+1=72$ samples, need be computed in equation 6.5 since the upper half frequency band is a mirror image of the lower half band. For the rest of the thesis, a magnitude spectrum is assumed to be defined only for the frequency range 0Hz to 4kHz (i.e. 0 to π radians).

6.3.1.2 Decomposition of characteristic waveforms to SES and RES

To compute the SES at each sub-update point, the mean magnitude spectra at adjacent update-points are linearly interpolated. For fully voiced speech, the SES at each sub-update point is defined, from 0Hz to the highest DFT frequency which is smaller than or equal to the 9th LSF, as:

$$\left|U_{Sjk}^{(l)}\right| = \left|\overline{U}_{jk}^{(l-1)}\right| + \frac{j}{8}\left(\left|\overline{U}_{jk}^{(l)}\right| - \left|\overline{U}_{jk}^{(l-1)}\right|\right) \qquad (6.6)$$
$$k \le K_j^{(l)}$$
$$j = 1, 2, \ldots, 8$$

where

l is the number of the current update-point

$\left|U_{S_{jk}}^{(l)}\right|$ is the kth SES magnitude component of the jth characteristic waveform

$K_j^{(l)}$ is the number of DFT bin which separates a DFT magnitude spectrum into SES and RES band and is computed from the 9th LSF ω_9 of the current analysed speech segment as:

$$K_j^{(l)} = \text{Integer}\left(\frac{\omega_9 p_j^{(l)}}{2\pi}\right) \qquad j = 1, 2, \ldots, 8 \tag{6.7}$$

where $p_j^{(l)}$ is the instantaneous pitch-period of the jth characteristic waveform

The RES at each update-point is defined only in the higher frequency region, i.e. for the DFT bins between $K_j^{(l)}$ and $\frac{p_j^{(l)}}{2}+1$ as:

$$\left|U_{R_{jk}}^{(l)}\right| = \left|U''^{(l)}_{jk}\right| \qquad k > K_j^{(l)} \qquad j = 1, 2, \ldots, 8 \tag{6.8}$$

where

$\left|U''^{(l)}_{jk}\right|$ is the jth time- and power-normalised characteristic waveform

$\left|U_{R_{jk}}^{(l)}\right|$ is the kth RES magnitude component of the jth characteristic waveform

A magnitude spectrum, which characterises a modified characteristic waveform $v_j(n)$ at each sub-update point is thus obtained as follow:

$$\left|V_{jk}^{(l)}\right| = \begin{cases} \left|U_{S_{jk}}^{(l)}\right| & k \le K_j^{(l)} \\ \left|U_{R_{jk}}^{(l)}\right| & k > K_j^{(l)} \end{cases} \qquad j = 1, 2, \ldots, 8 \tag{6.9}$$

where $\left|V_{jk}^{(l)}\right|$ is the kth magnitude component of the jth modified characteristic waveform.

An example of this spectral decomposition scheme applied to the group of characteristic waveforms shown in figure 6.3b is presented in figure 6.6. Using the spectral decomposition scheme, the slowly and rapidly evolving surfaces obtained from the SES and RES are illustrated in figures 6.6a and b. Note that, the original phase spectra, ϕ_{Uj} for j=1 to 8, of the characteristic waveforms are used. The handling of phase information will be discussed in next section.

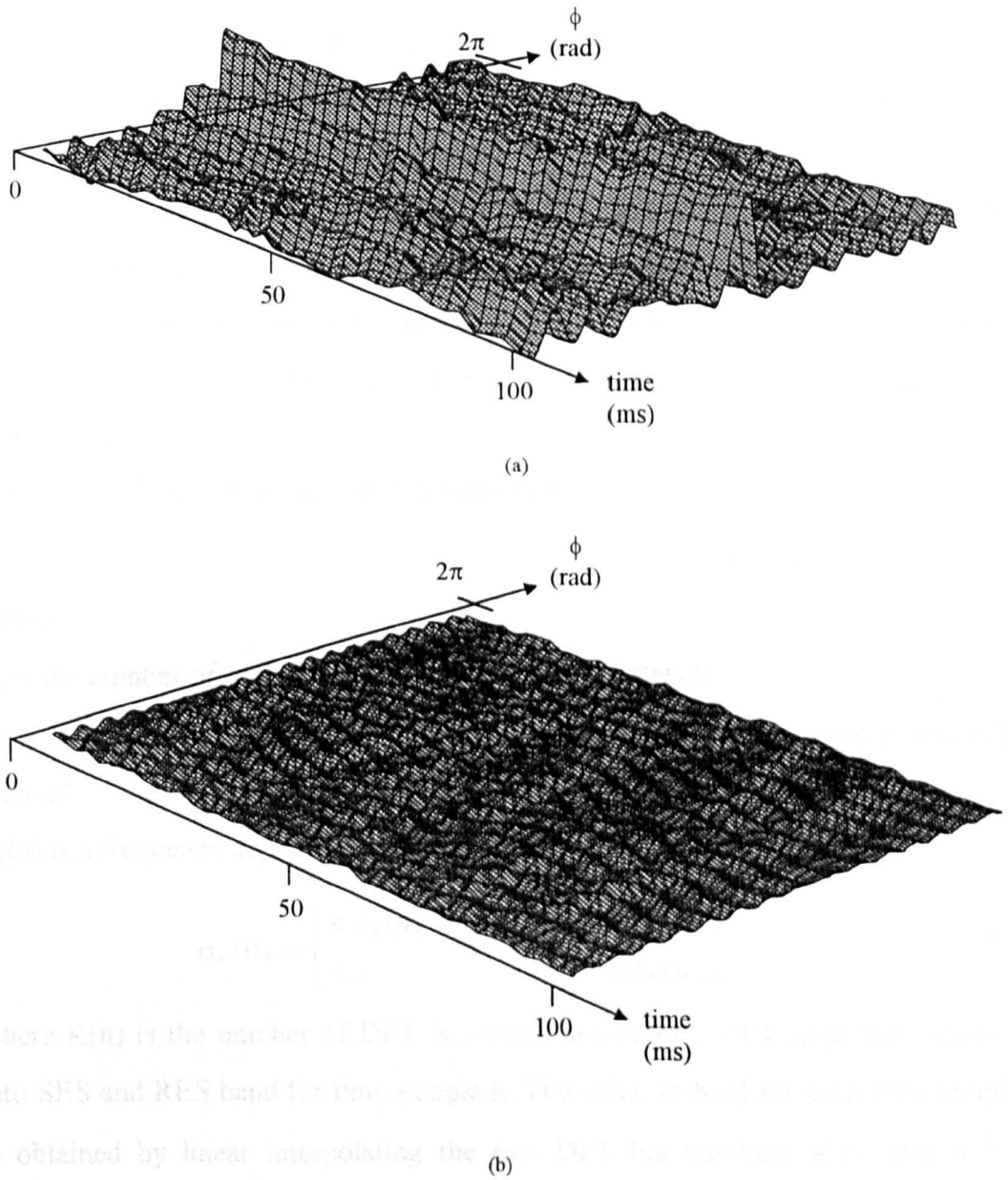


Figure 6.6 Example of spectral decomposition on the voiced portion of the word "No" spoken by a male speaker. (a) slowly evolving components of characteristic waveforms (b) rapidly evolving components of characteristic waveforms

It can be seen that by separating the characteristic waveforms in this way, a surface of more smoothly evolving characteristic waveforms can be obtained, as shown in figure 6.6a. This slow evolution property allows representative waveforms to be sent at a relatively low rate, allowing the slowly evolving signal to be reconstructed by interpolation. In figure 6.6b, the surface of the rapidly evolving characteristic waveforms exhibits a much faster rate of change and a higher update rate is required. The energy of the RES in this case is very small compared to that of the SES because the speech is fully voiced.

6.3.1.3 Reconstruction of the voiced speech

To recover the speech residual, each modified characteristic waveform may be phase aligned with the previous aligned one using the original phase spectrum ϕ_{Uj} at the sub-update point. Note that the original phase spectrum will be replaced by a derived phase spectrum when the GSPWI coder model is quantised. The real and imaginary coefficients of each aligned cycle are calculated and they are linearly interpolated with the previous ones to yield an approximation to the residual.

$$\tilde{s}(n) = \frac{2}{p(n)} \sum_{k=0}^{\frac{p(n)}{2}} \left\{ \left(R_k^{(j-1)} + \psi(n)\left(R_k^{(j)} - R_k^{(j-1)} \right) \right) \cos(\sigma_k(n)) + \left(I_k^{(j-1)} + \psi(n)\left(I_k^{(j)} - I_k^{(j-1)} \right) \right) \sin(\sigma_k(n)) \right\} \qquad n = 0, 1, \ldots, N_s - 1 \tag{6.10}$$

where

N_s is the number of speech samples in a sub-update interval

$\psi(n)$ is a linear interpolation function which rises from 0 to 1 across a sub-update interval

$\sigma_k(n)$ is a frequency dependent phase term which is defined as:

$$\sigma_k(n) = \begin{cases} k\,\sigma_q(n) & k \le K(n) \\ \sigma_{rk} & otherwise \end{cases} \tag{6.11}$$

where K(n) is the number of DFT bin which separates a DFT magnitude spectrum into SES and RES band for time sample n. The value of K(n) for each time sample n is obtained by linear interpolating the two DFT bin numbers $K_{j-1}^{(I)}$ and $K_j^{(I)}$ at adjacent sub-update points.

$\sigma_q(n)$ is the instantaneous phase obtained from the quadratic phase interpolation (i.e. interpolating the phase as for the TPSWI model). $\sigma_r(k)$ is a pseudo-random phase sample uniformly distributed between $-\pi$ and π. A pseudo-random phase spectrum is assigned for each sub-update interval, i.e. every 2.5ms. Using this phase arrangement, a quadratic interpolated phase is used for the SES frequency components and a pseudo-random phase is assigned to the RES frequency components.

Once $\hat{e}(n)$ has been calculated, it is scaled to the required power to yield the reconstructed residual. The resulting signal is then processed by the LSF synthesis filter to obtain the decoded speech.

6.3.1.4 Performance evaluation of the spectral decomposition scheme for voiced speech

In figure 6.7a, the 1-D speech waveform corresponding to the group of characteristic waveforms in figure 6.3b is presented. The reconstructed speech waveform obtained using the spectral decomposition technique is presented in figure 6.7b. It can be seen that the reconstructed waveform is able to track the original one quite closely.

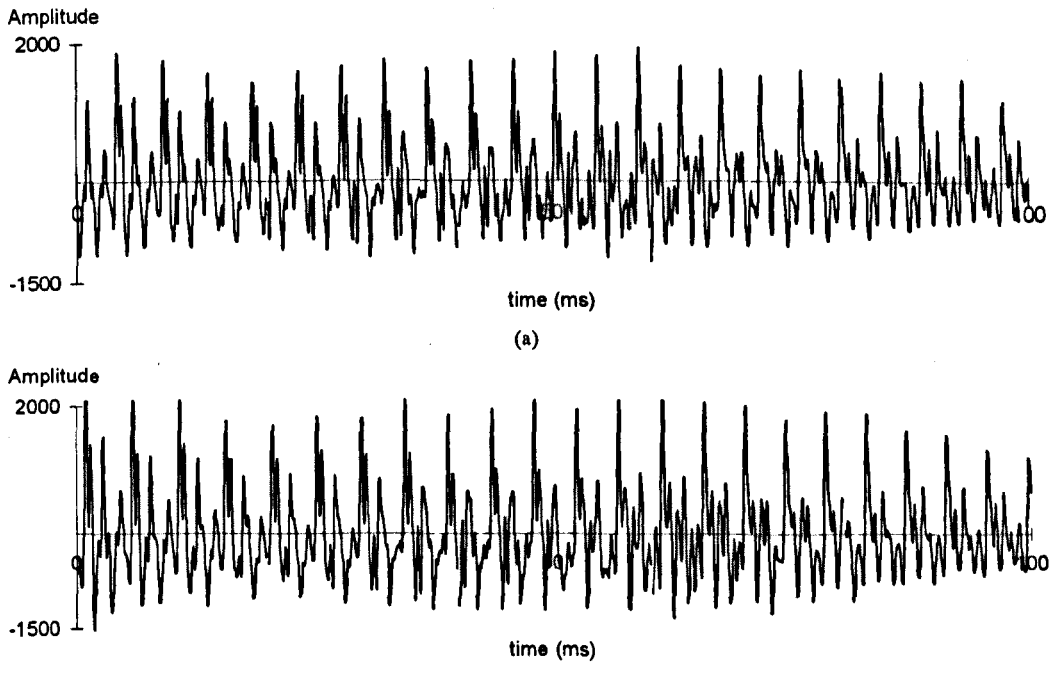


Figure 6.7 Example of voiced speech synthesised by the spectral decomposition scheme.
(a) original speech (b) synthesised speech

To evaluate the performance of the spectral decomposition scheme for voiced speech, the TPSWI model described in chapter 5 was recalled. The PSWI technique used for voiced speech in this model was replaced by the spectral decomposition scheme described in this chapter. The performance of the resulting model was compared with that of the original TPSWI model using the speech file "OPERATOR.DAT" [21] as input. The phase spectra of the characteristic waveforms in this experiment were taken as the true phase spectra as measured from the input speech. It was found that better speech quality was obtained from the new arrangement for voiced speech than was obtained from the original TPSWI model. The voiced speech synthesised by the spectral decomposition scheme was generally smoother. The speech quality of the resulting model was very close to the original speech when eight RES spectra were used per 20ms of speech at the synthesis stage. Furthermore, good speech quality was still obtained using only one RES per 20ms update-point.

6.3.2 Spectral decomposition of characteristic waveforms for unvoiced speech

An unvoiced residual is a random-like signal which possesses little or no periodicity. Reconstruction of unvoiced speech in the same way as voiced speech may result in a severe deterioration in the overall speech quality because of unwanted periodicity introduced into the reconstructed unvoiced residual, by the quadratic phase interpolation. In order to accomplish the coding of unvoiced speech, the frequency boundary between the bands which define the SES and RES is adapted. The frequency boundary f_{SR} will have a low value (for example f_{SR}=0Hz) for unvoiced speech, where random components would be dominant in the reconstructed residual. The way that this frequency boundary is chosen will be presented in the next section.

6.4 The generalised pitch-synchronous waveform interpolation (GPSWI) coder

The GPSWI coder was developed from the TPSWI coder, removing the need for a voiced/unvoiced switch and yielding a uniform coding algorithm for all speech types. In figure 6.8, a schematic diagram of the GPSWI coder is presented.

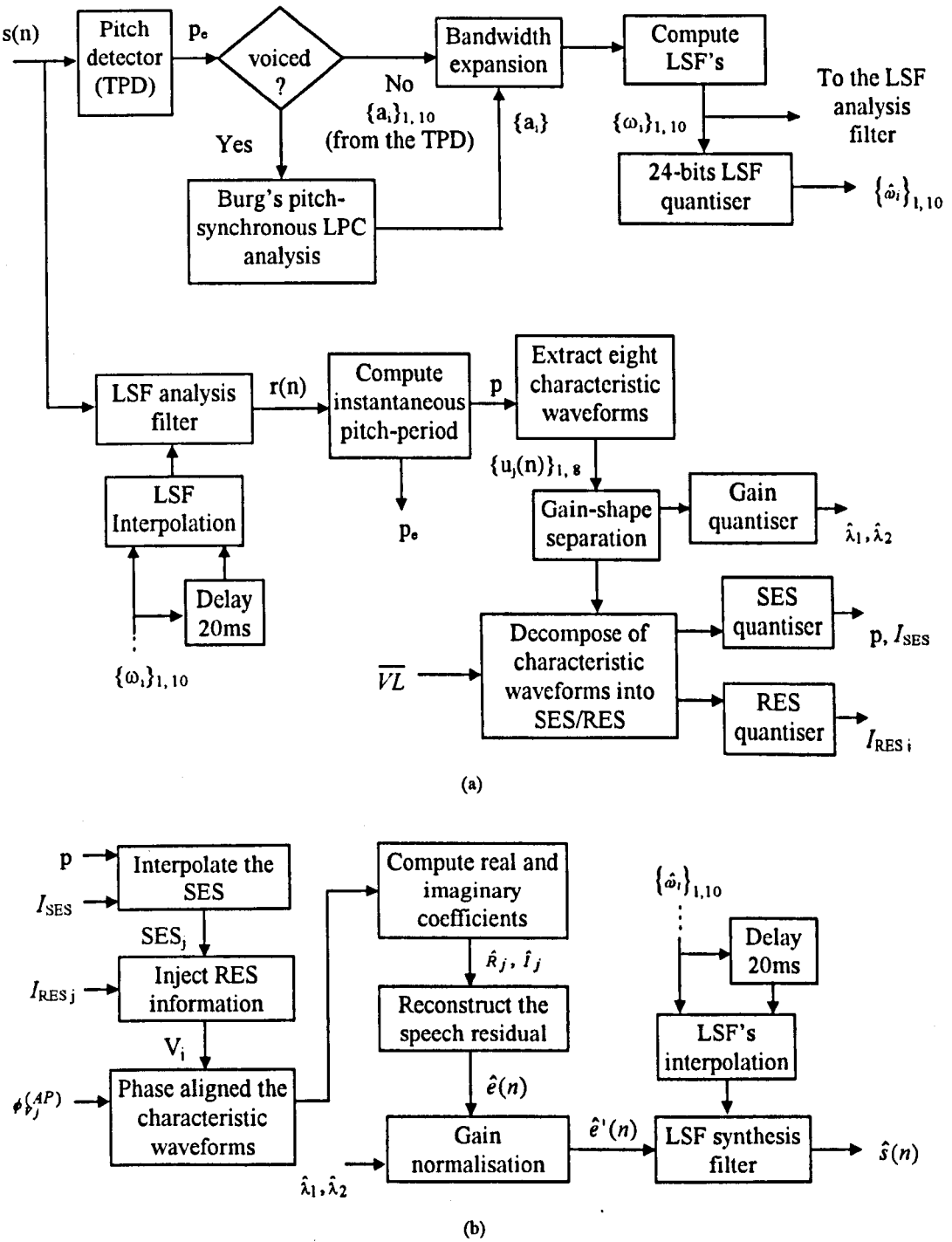


Figure 6.8 Schematic diagram of the GPSWI coder.
(a) encoder (b) decoder

6.4.1 Pitch estimation and spectral estimation

Like the TPSWI coder, the GPSWI coder operates with 20ms update intervals and 10ms look-ahead. For each update-point, a suitable length segment of speech is first assessed by the two-way pitch detector (TPD). The TPD evaluates the nature of the speech signal as well as giving a pitch-period estimate for both voiced and unvoiced speech. When unvoiced speech is indicated, the "pitch-period" is obtained by simply locating the delay corresponding to the global maximum in the cross-correlation function between two similar length segments extracted from the speech. The pitch post-processing unit in the TPD is disabled for unvoiced speech. In case of voiced speech, the pitch post-processing in the TPD is applied as usual. The range of possible pitch-periods assumed by the GPSWI model is set to be from 16 to 143 samples. Only 7 bits are needed to encode the pitch period at each update-point.

Once the nature of the input speech has been classified as voiced or unvoiced and an estimated pitch-period has been obtained, Burg's pitch-synchronous LP analysis is used to re-estimate, now more accurately, the short-term spectral envelope for voiced speech. Otherwise, the set of LP filter coefficients produced by the TPD is used directly. A 10 Hz bandwidth expansion is applied to the 10th order LP ladder filter coefficients and they are then converted to LSF's.

6.4.2 Extraction of characteristic waveforms

The GPSWI coder is operated in the residual-domain in which a segment of residual signal is obtained by passing the speech through an LSF analysis filter, with the unquantised LSF's as computed at the update-points being interpolated on a sample-by-sample basis between update-points. Eight characteristic waveforms are extracted from each segment of 20ms residual signal between consecutive update-points (i.e. at a sub-update interval of 2.5ms). The intermediate characteristic waveforms are extracted around each sub-update point using an intermediate pitch-period. The intermediate pitch-period $p_j^{(I)}$, for j=1 to 8, is computed by linearly interpolating between the instantaneous pitch-frequencies at adjacent update-points

and converting the resulting frequency back to a pitch-period. This is carried out using equation 6.12:

$$p_j^{(l)} = \left(\frac{1}{p^{(l-1)}} + \frac{j}{8}\left(\frac{1}{p^{(l)}} - \frac{1}{p^{(l-1)}} \right) \right)^{-1} \tag{6.12}$$

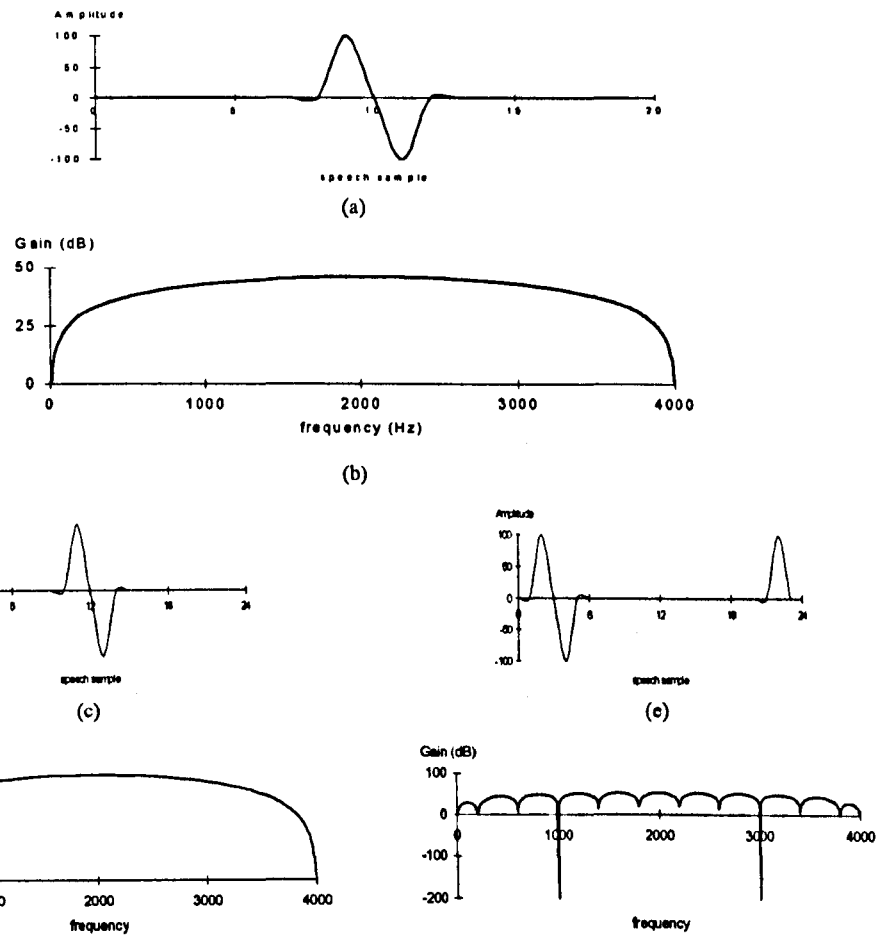


Figure 6.9 Example of characteristic waveform extraction error.
(a) original characteristic waveform (20 samples) (b) original magnitude spectrum (c) characteristic waveform extracted using a wrong pitch of 24 samples (d) magnitude spectrum of [c] (e) characteristic waveform extracted using the wrong pitch at a different location around the sub-update point (f) magnitude spectrum of [e]

Since the instantaneous pitch-period at each sub-update point is determined by interpolation, it may not be the most accurate value obtainable at that point. Direct extraction of a characteristic waveform at the sub-update point may then lead to error in the DFT magnitude spectrum. In figure 6.9, the effect of characteristic waveform extraction error is demonstrated. Suppose the true pitch-period at a sub-update point

is 20 samples and that the pitch-cycle and its magnitude spectrum are as shown in figures 6.9a and b respectively. To demonstrate the effect of a pitch error which may occur due to interpolation, an incorrect pitch-period estimate of 24 samples was assumed and therefore a rectangular window of length 24 samples was used to extract the characteristic waveform. Two different positions of the rectangular window relative to the pitch-cycle were used to extract characteristic waveforms around the sub-update point. The two extracted characteristic waveforms are shown in figures 6.9c and e. DFT analysis was performed on the characteristic waveforms to yield their magnitude spectra, which are presented in figures 6.9d and f respectively. The spectrum in figure 6.9d is the same as the original one even though the pitch-period is wrong. On the other hand, an incorrect spectrum was obtained from the characteristic waveform in figure 6.9f.

The conclusion is drawn that a second peak must be avoided from inclusion in the intermediate characteristic waveforms. A second peak can be effectively eliminated by avoiding large amplitudes at either ends of an intermediate characteristic waveform. This is done by comparing the rms power of a range of speech samples at both ends of an extraction window (10% of the pitch-period) to the rms power of the entire characteristic waveform. If any of the two rms powers is larger than the rms power of the characteristic waveform, the extraction window is slightly shifted around the current update-point until the condition is satisfied.

A characteristic waveform should be extracted by locating the centre of each extraction window around the sub-update point as shown in figure 6.10a. However, it may not be possible to fulfil this condition for sub-update points near the end of the input signal buffer since no future residual information may be available if overall delay is to be kept to a minimum. This will occur with $u_8(n)$, and may occur in $u_7(n)$, $u_6(n)$ and even $u_5(n)$ depending on how large instantaneous pitch-periods are. The extraction procedure is therefore modified such that for $u_8(n)$, the characteristic waveform is extracted at the end of the input buffer as shown in figure 6.10b. In the case of $u_7(n)$, the centre of the extraction window is allowed to be a little before the corresponding sub-update point and similarly for $u_6(n)$ and $u_5(n)$ if necessary. The time shift of the centre of the window from the true sub-update point is allowed

because of the periodicity of the voiced residual waveform. A suitable arrangement for the window is illustrated in figure 6.10b. Note that, precautions must be taken to avoid the same characteristic waveform being extracted for consecutive sub-update points.

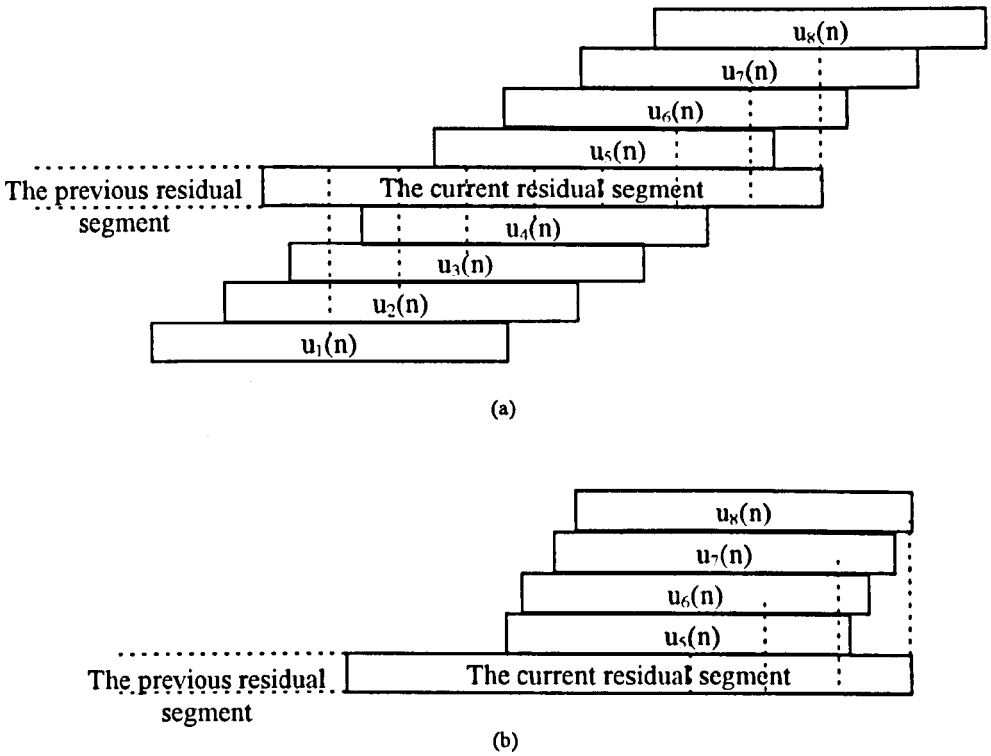


Figure 6.10 Extraction of eight characteristic waveforms in a residual segment.
(a) original approach (b) modification of the extraction procedure in order to avoid the need for future residual information

6.4.3 Decomposition of characteristic waveforms into gains, SES and RES

DFT analysis is performed on each characteristic waveform to yield its magnitude spectrum which is then zero-padded to the length of the maximum pitch-period samples, 143 samples, as described earlier. Power and time normalisation are then applied using equation 6.4. In the GPSWI coder, two gain factors $\lambda_1^{(l)}$ and $\lambda_2^{(l)}$ are encoded at each update-point for the 20ms synthesis frame preceding it. The values of $\lambda_1^{(l)}$ and $\lambda_2^{(l)}$ are taken as the rms values of characteristic waveforms $u_4(n)$ and $u_8(n)$ respectively. It was shown in section 3.6 that a sudden boost in the LP residual may occur in the middle of two update-points, when the speech waveform in the middle of the two update-points changes rapidly and pitch-synchronous LP analysis is being used. In this case, $\lambda_1^{(l)}$ may be substantially higher than $\lambda_2^{(l-1)}$ and

$\lambda_2^{(l)}$. A form of "bubbling" noise perceived in the decoded speech may result from direct interpolation between $\lambda_2^{(l-1)}$ and $\lambda_1^{(l)}$ then between $\lambda_1^{(l)}$ and $\lambda_2^{(l)}$ to form an rms contour under such a condition. The value of $\lambda_1^{(l)}$ must be restricted to avoid a sudden boost in the intermediate gain. This is achieved by defining:

$$\lambda_1^{(l)} = \begin{cases} 0.5*\left(\lambda_2^{(l-1)} + \lambda_2^{(l)}\right) & \lambda_1^{(l)} > 2\lambda \\ \lambda_1^{(l)} & otherwise \end{cases} \tag{6.13}$$

where λ is the larger value of $\lambda_2^{(l-1)}$ and $\lambda_2^{(l)}$.

After the power and time normalisation, the characteristic waveforms are decomposed into SES and RES spectrographs and sampled in the way described in section 6.3.2. In order to accomplish spectral decomposition for voiced and unvoiced speech, the boundary frequency (f_{SR}) is made adaptive to a mean voiced confidence level, $\overline{VL}$, which is defined as the average value of the voiced confidence levels at the current and the previous update-points:

$$\overline{VL}^{(l)} = \frac{VL^{(l-1)} + VL^{(l)}}{2} \tag{6.14}$$

The voiced confidence level, which is defined as the scaled sum of the voicing probabilities of a number of features measured from a segment of speech, is provided at each update-point by the TPD as described in chapter 2. The selection of f_{SR} is according to the list in table 6.1. The boundary frequency at each sub-update point is computed by linearly interpolating between the values of $\overline{VL}$ obtained at adjacent update-points.

$\overline{VL}$	frequency boundary for SES/RES
< 0.3	0
0.3 - 0.4	ω_3
0.4 - 0.5	ω_6
> 0.5	ω_9

ω_i - the ith LSF

Table 6.1 Adaptive frequency boundary scheme for SES/RES in the GPSWI coder

6.4.4 Speech reconstruction at the decoder

The reconstruction of the decoded speech has been discussed in section 6.3.1.4. In experiments carried out so far, the true phase spectrum of each characteristic waveform ϕ_{Uj} has been assigned to the corresponding modified characteristic waveform $v_j(n)$ at a sub-update point, i.e. ϕ_{Vj} has been taken to be equal to ϕ_{Uj} for j=1 to 8. In practice, it is intended that only the SES and RES spectral shapes will be encoded and that the phase spectrum for the modified characteristic waveforms, ϕ_{Vj} for j=1 to 8 will be artificially derived at the decoder. The phase spectrum for each characteristic waveform will be taken as the inverse of the phase spectrum of a 2nd-order all-pass filter, i.e.:

$$\phi_{V\,jk} = -\phi_F\left(k\,\varpi_j\right) \qquad 0 \le k\,\varpi_j < \pi \tag{6.15}$$

where ϖ_j is the pitch-frequency at sub-update point and $\phi_F(\omega)$ is as defined in equation 5.35. The value of α for the all-pass filter is set to 0.95 and the value of β is chosen from table 5.1 using the instantaneous pitch-period p_j at the sub-update point.

6.4.5 Performance evaluation of the GPSWI coder

6.4.5.1 The GPSWI coder with original phase spectrum

In the following experiments, the original phase spectrum of the characteristic waveforms ϕ_{Uj} are used:-

In figure 6.11, the speech waveform for the word, "No", synthesised by the GPSWI coder is illustrated. The original speech is presented in figure 6.11a. The decoded speech signal is presented in figure 6.11d. It can be seen in figure 6.11d that the reconstructed speech signal tracked the voiced onset and offset of the original speech very well. The amplitude envelopes of the two signals are almost the same. In figure 6.11b, a segment of original voiced speech is presented. Comparing this with the reconstructed voiced speech shown in figure 6.11c, the two voiced segments looked very alike. Informal listening tests suggested that the perceived speech quality of the GPSWI decoded speech is almost indistinguishable from the original when eight RES per 20ms of speech are used at the synthesis stage.

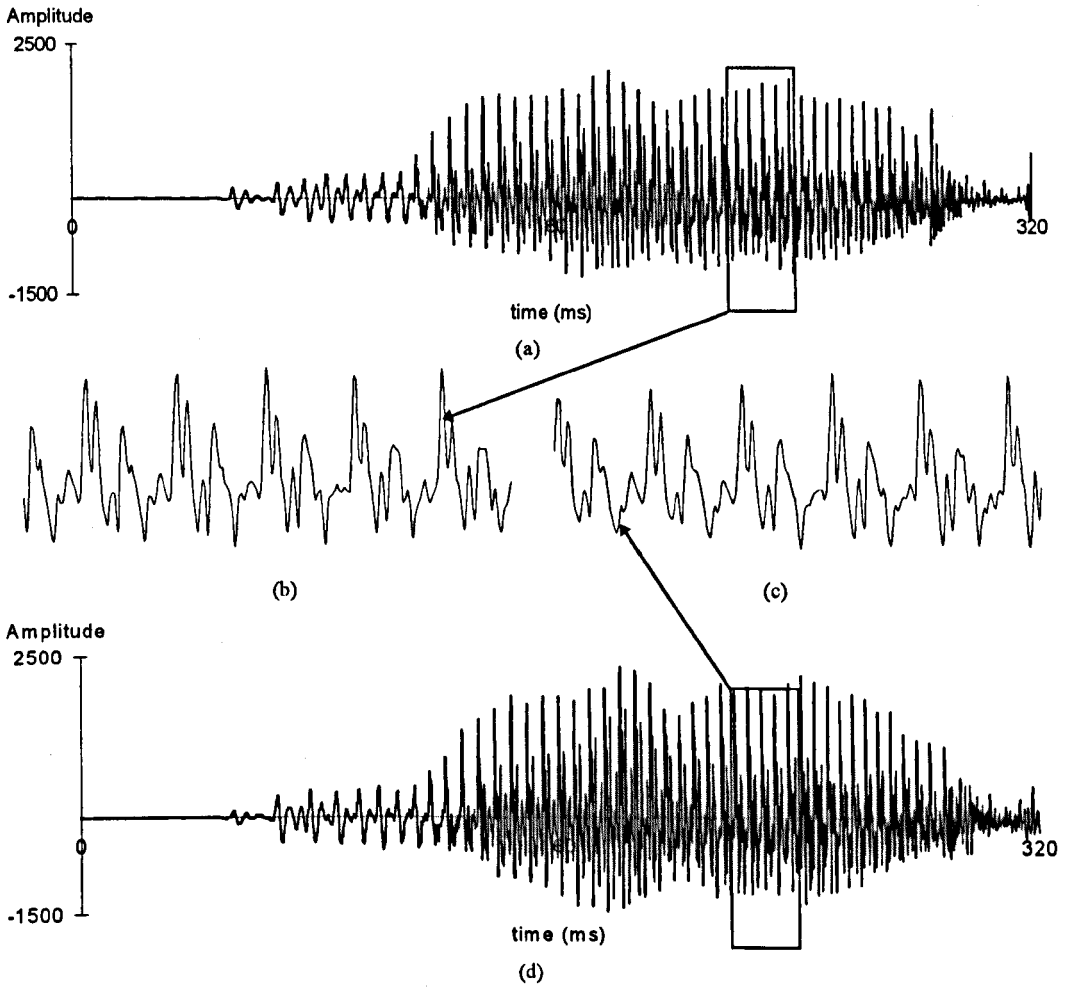


Figure 6.11 Example of decoded speech synthesised by the GPSWI coder.
(a) original speech (b) section of the original voiced speech (c) the corresponding section of the reconstructed voiced speech (d) the decoded speech

The GPSWI coder was compared with TPSWI coder using the speech file "OPERATOR.DAT" [21] as input. Informal listening tests suggested that the perceptual quality of the GPSWI coder is always better than the TPSWI coder, when the number of prototype waveforms used in the TPSWI coder is the same as the number of RES used in the GPSWI coder.

The GPSWI coder was also compared with the TPSWI coder using speech with additive noise samples including car noise [90], babble noise [91], multi-speaker noise [92] and white noise [93] at different SNR levels: 0dB, 10dB, 20dB and 30dB (refer back to chapter 2). Informal listening tests suggested that the GPSWI decoded speech generally preserved a smoother speech quality than that produced by the TPSWI coder and is always better. The difference in the decoded speech was

significant when the signal-to-noise ratio of the noisy speech was less than 20dB. Since an alternative model is used in the TPSWI coder for unvoiced speech, significant deterioration in the perceived speech quality results when voiced speech is mis-classified as unvoiced.

It was found that for the TPSWI coder in all noisy speech samples, the overall speech quality deteriorated substantially as the SNR level dropped, even though the intelligibility of the speech was still preserved. This is due to the way that the TPSWI coder handles unvoiced speech. Since unvoiced speech is modelled by pseudo-random sequence with a similar power contour, the noise samples under test may not be adequately modelled by the pseudo-random sequence, such as car noise and multi-speaker noise. Experimental results suggested that TPSWI worked better for white noise and babble noise than for car noise and multi-speaker noise. In testing the car and multi-speaker noise corrupted speech file, the noise content, i.e. the car and speech from the background speakers respectively, was replaced by a form of noise which was perceptually quite irritating.

In contrast to the TPSWI coder, the GPSWI coder was able to model both the speech content as well as the noise type. It seemed that the overall speech quality could be substantially increased by a better modelling of unvoiced speech when the SNR level dropped. Informal listening tests suggested that the GPSWI coder is able to give good speech quality even when the SNR level of the input noisy speech file is at 0dB for the four noise types.

The trade-off in the GPSWI coder is an increase in the computational complexity due to the requirement of extracting and analysing eight characteristic waveforms per 20ms speech. Furthermore the spectral decomposition scheme is applied to unvoiced speech as well as to voiced speech. This greatly increases the computational complexity for unvoiced speech as compared to the TPSWI coder.

6.4.5.2 The GPSWI coder with derived phase spectrum

In the following experiments, the phase spectrum of the modified characteristic waveforms is derived from the 2nd order all-pass filter:-

In testing the GPSWI coder with the speech file "OPERATOR.DAT" [21], minor degradation in the speech quality was revealed when comparing the decoded speech obtained with the synthetic phase with the decoded speech obtained using original phase spectrum. For male speech, the decoded speech with synthesis phase exhibited a slightly synthetic quality at some portions of the speech file, such as the end of the words "England" and "Bailham".

Surprisingly, the degradation in the speech quality seemed to be less noticeable when the SNR level of the speech file was reduced. For instance, the quality of the GPSWI decoded speech when using the original phase spectrum and the derived phase spectrum under 0dB car noise condition were virtually indistinguishable. It seemed that the synthetic quality was masked out by the background noise. This suggested that a reasonable modelling of unvoiced speech may be as important as modelling the voiced speech for a noisy environment.

6.5 Conclusions

The fundamental ideas behind recent approaches to WI coding have been introduced in this chapter. These ideas with various innovations have been applied to the design of a speech coder intended for 2.4kb/s operation when fully quantised. The model was developed from the TPSWI model described in chapter 5, and is referred to as the "GPSWI" model.

The TPSWI and GPSWI coders will encode magnitude only information. When the original phase spectrum is used to reconstruct the speech, the speech obtained from the GPSWI coder is always better than that obtained from the TPSWI coder, when the number of RES spectra in the GPSWI coder is equal to the number of prototype waveforms in the TPSWI coder used. The decoded speech from the GPSWI magnitude only model with the original phase spectrum is very close to the original when eight RES spectra are used per 20ms at the synthesis stage. In addition, the GPSWI coder performed substantially better than the TPSWI coder for noisy

speech. Unlike the TPSWI coder, the GPSWI coder is able to model, to some extent, the noise content as well as the speech content, and thus a smoother decoded speech can be obtained.

To finalise the unquantised GPSWI coder, the original (true) phase spectrum was replaced with a derived "all-pass" phase spectrum as will be used in the fully quantised 2.4kb/s version. Only minor degradation in the decoded speech resulted, some portions of male speech being perceived as being slightly less natural. It was concluded that the all-pass phase model was likely to be acceptable in the final coder. In the case of speech corrupted by noise, the degradation in speech quality became less noticeable as the SNR was reduced, in comparison with the perceived degradation with the true phase applied.

Chapter 7

Quantisation of the TPSWI and GPSWI coders

7.1 Introduction

In the original form of PWI coding [70], a pitch-period length residual segment, i.e. a prototype waveform, is quantised at each update-point using differential quantisation. This means that the prototype waveform is expressed as the sum of a proportion of the previous quantised prototype waveform and scaled versions of one or more waveforms read from vector quantisation code-books. The simplest form of such a quantiser employs only a white-Gaussian code-book. The problem associated with such a quantiser is that it is not ideal for quantising prototype waveforms containing predominant and perceptually very important pitch-pulses. Such pulses depend on the accumulated contributions of previous quantised prototype waveforms and therefore the pulses tend to build up and decay rather too slowly. This leads to a reverberant quality in the decoded speech. To enhance the quantiser performance, a single-pulse code-book may be included to model the peak structure in a prototype waveform.

PWI coding was originally proposed for voiced speech portions only, a switch to an alternative model such as CELP being advocated for unvoiced speech. With more recent waveform interpolation (WI) coding techniques [73]-[76], a unified coding algorithm is able to handle both voiced and unvoiced speech and the evolution of waveforms or their spectra is decomposed into slowly evolving and rapidly evolving components. Different quantisation schemes are used for the two components [75][76].

Quantisation of the TPSWI model is divided into two parts: the short-term spectral envelope and the residual information. The pitch-period and the speech

classification at each update-point are jointly quantised using 8 bits. The short-term spectral envelope is represented by ten LSF coefficients which are vector-quantised using the 24-bit IMS-LSF quantiser discussed in chapter 4. In case of unvoiced speech, the logarithms of four residual gain factors for the 20ms synthesis frame preceding each update-point are vector quantised at the update-point. For voiced speech, two gain factors and the magnitude spectrum of a prototype waveform are quantised. The gain factors are encoded as logarithms and are differentially quantised using scalar quantisers. The two gain factors are for the first and second 10ms sub-frame of the synthesis frame preceding the current update-point. For the purposes of vector quantisation the magnitude spectrum of each prototype waveform is evenly split into two frequency bands. Different code-book sizes are assigned to the DFT magnitude coefficients for the lower and upper frequency bands. A 2.3kb/s version of the TPSWI coder was obtained by encoding a single prototype waveform at every 20ms update. It was found that the performance of the TPSWI coder may be enhanced by increasing the update rate of prototype waveforms and hence a 2.4kb/s version was also produced by doubling the update rate of prototype waveforms. To achieve this increase in update-rate, only the lower frequency band of each prototype waveform is quantised, the upper band of the magnitude spectrum being assumed to be flat.

Quantisation of the short-term spectral envelope and the gain factors of the characteristic waveforms in the GPSWI coder is the same as for the TPSWI coder in voiced mode. In the GPSWI coder, there is no model switching according to speech classification. The pitch-period at each update-point is quantised to 7 bits. The SES and RES are vector-quantised using full-band code-books, more weight being assigned to the lower frequency components by the distance measure used for the SES code-book. A 2.4kb/s GPSWI coder was achieved by sending one SES and two RES spectra per 20ms speech segment.

This chapter will begin with a brief introduction to the problems of quantising PWI and more recent WI coders. In sections 7.4 and 7.5, quantisation schemes for the 2.4kb/s TPSWI and GPSWI coders developed in this project will be presented. For the 2.4kb/s TPSWI coder, quantisation of the gain factors for unvoiced speech

must be considered as well as quantisation of the gain factors and magnitude spectra of the prototype waveforms. For the GPSWI coder, ways of quantising the SES and RES must be devised.

7.2 Quantisation of PWI coder

In the PWI coding method employed by Kleijn [70], residual prototype waveforms are quantised using differential quantisation from one update-point to the next. Time-domain or Fourier series cosine and sine representations of the prototype waveforms may be used and each prototype waveform is represented by contributions from the previously quantised prototype waveform and one or more code-book candidates. The quantisation scheme is characterised by equation 7.1.

$$\underline{\hat{u}}^{(l)} = \hat{g}_0 \underline{\hat{u}}^{(l-1)} + \sum_{z=1}^{Z} \hat{g}_z \underline{\gamma}_z[I_z] \tag{7.1}$$

where

$\underline{\hat{u}}^{(l)}$ is the quantised residual prototype waveform at update-point *l*

$\underline{\hat{u}}^{(l-1)}$ is the quantised residual prototype waveform at the previous update-point *l-1*

$\hat{g}_0$ is a quantised gain factor determining the contribution from $\underline{\hat{u}}^{(l-1)}$

z is the number of code-books used in the differential quantiser

$\underline{\gamma}_z[I_z]$ is a vector with index I_z selected from zth code-book Γ_z

$\hat{g}_z$ is the quantised gain factor associated with the zth code-book entry

The simplest form of equation 7.1 is:

$$\underline{\hat{u}}^{(l)} = \hat{g}_0 \underline{\hat{u}}^{(l-1)} + \hat{g}_1 \underline{\gamma}_1[I_1] \tag{7.2}$$

where $\underline{\gamma}_1[I_1]$ is the I_1th candidate from a white-Gaussian code-book Γ_1.

Three parameters are required in quantising the prototype waveform: g_0, g_1 and I_1 (the code-book index). In figure 7.1 the schematic diagram of a time-domain prototype waveform quantiser is shown [18].

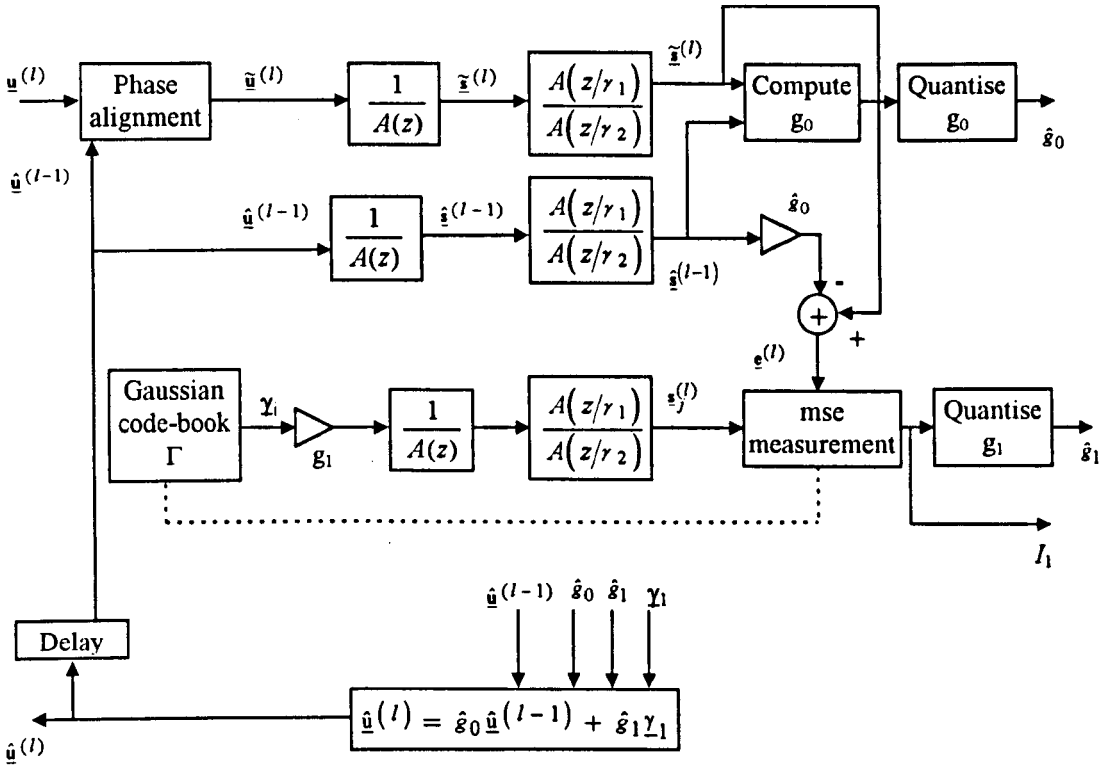


Figure 7.1 Schematic diagram of a prototype waveform quantiser

The prototype waveform quantiser works similarly to a CELP quantiser (Appendix A) in which the three parameters are evaluated using analysis-by-synthesis (a-b-s) with perceptual weighting. Prior to its quantisation, the previously quantised prototype waveform $\hat{u}^{(l-1)}(n)$ must be zero-padded or truncated to the same length as the current quantised prototype waveforms $u^{(l)}(n)$. The current prototype waveform $u^{(l)}(n)$ is then phase aligned with the previously quantised prototype waveform $\hat{u}^{(l-1)}(n)$ by circular shifting $u^{(l)}(n)$ according to the procedure described in Appendix C. The contribution from the previously quantised prototype waveform to the quantised version of $u^{(l)}(n)$ is now taken as $g_0 \hat{u}^{(l-1)}(n)$ where g_0 may be computed in the speech-domain using equation 7.3.

$$g_0 = \frac{\sum_{n=0}^{p-1} \tilde{s}^{(l)}(n)\, \hat{s}^{(l-1)}(n)}{\sum_{n=0}^{p-1} \hat{s}^{(l-1)^2}(n)} \tag{7.3}$$

In equation 7.3 $\tilde{s}^{(l)}(n)$ is the speech segment produced by passing $\tilde{u}^{(l)}(n)$ through an LP synthesis filter and a perceptual weighting filter (refer to the CELP section in Appendix A), using the original LP ladder filter coefficients. The speech segment $\hat{s}^{(l-1)}(n)$ is produced by passing $\hat{u}^{(l-1)}(n)$ through an LP synthesis filter and a perceptual weighting filter, using the quantised LP ladder filter coefficients.

Equation 7.3 is derived by minimising the mean-square-error distance measure between the two equal length segments: $\left\{\hat{s}^{(l-1)}(n)\right\}_{0,p-1}$ with each element scaled by g_0 and the sequence $\left\{\tilde{s}^{(l)}(n)\right\}_{0,p-1}$. The mse distance measure between the two segments is defined as:

$$E = \frac{1}{p}\sum_{n=0}^{p-1}\left(\tilde{s}^{(l)}(n) - g_0\hat{s}^{(l-1)}(n)\right)^2 \tag{7.4}$$

Differentiating E with respect to g_0 and setting the resulting expression to zero gives equation 7.3.

The gain factor g_0 is quantised using a scalar quantiser. The contribution of the previously quantised speech segment $\hat{s}^{(l-1)}(n)$ with the quantised $\hat{g}_0$ is subtracted from the current speech segment $\tilde{s}^{(l)}(n)$ to obtain the difference signal $e^{(l)}(n)$:

$$e^{(l)}(n) = \tilde{s}^{(l)}(n) - \hat{g}_0\hat{s}^{(l-1)}(n) \tag{7.5}$$

This difference signal is compared, in turn, with each of the speech segments $\left\{s_j^{(l)}(n)\right\}_{0,p-1}$ for j=1, 2, ..., L, where L is the number of code-book entries. These speech segments are generated by passing each of the innovation sequences $\underline{\gamma}_j$ stored in the Gaussian code-book through the LP synthesis filter and perceptual weighting filter both with the quantised LP ladder filter coefficients. A mean-square-error (mse) distance measure is used to quantify the comparison. The minimum mse distance measure between $\left\{e^{(l)}(n)\right\}_{0,p-1}$ and one of the speech segments $\left\{s_j^{(l)}(n)\right\}_{0,p-1}$ for j=1, 2, ..., L can be found, as discussed in section 2.2, as by choosing the index j

to the code-book entry which maximises the normalised cross-correlation coefficient between the two segments. This function is defined as:

$$C(j) = \frac{\sum_{n=0}^{p-1} e^{(l)}(n)\, s_j^{(l)}(n)}{\sqrt{\sum_{n=0}^{p-1} e^{(l)^2}(n) \sum_{n=0}^{p-1} s_j^{(l)^2}(n)}} \tag{7.6}$$

The gain associated with the Gaussian innovation sequence g_1 can be found by differentiating the distance measure $E = \frac{1}{p}\sum_{n=0}^{p-1}\left\{e^{(l)}(n) - g_1 s_j^{(l)}(n)\right\}^2$ with respect to g_1 and setting the differentiation to zero. The resulting expression for the gain factor g_1 is then:

$$g_1 = \frac{\sum_{n=0}^{p-1} e^{(l)}(n)\, s_j^{(l)}(n)}{\sum_{n=0}^{p-1} s_j^{(l)^2}(n)} \tag{7.7}$$

The logarithm of the gain factor g_1 is scalar-quantised. The index I_1 of the best matched code-book sequence is encoded.

One of the disadvantages of this quantisation scheme is the mechanism by which spiky residual signals, with clear pitch excitation pulses must be represented. This may result in a reverberant synthesised speech quality as mentioned in the introduction. Figure 7.2 shows an example of a speech segment obtained from the differential quantisation scheme mentioned above. The quantised prototype waveform shown in figure 7.2c was obtained by summing an approximate contribution from the previous quantised prototype waveform shown in figure 7.2b and a suitably weighted vector selected from the Gaussian code-book. Compared with the unquantised prototype waveform shown in figure 7.2a, the pitch-pulse in the quantised prototype waveform is not as strong as it should be and the energy of the random-like components of the quantised prototype waveform is higher than it should be as compared with the original prototype waveform. The reconstructed residual signal shown in figure 7.2d, which was obtained by linearly interpolating between the two quantised prototype waveforms is rather too random. As a result, the pitch-pulses in the synthesised speech waveform shown in figure 7.2e are not as prominent as they

should be relative to the overall synthesised speech waveform. This may be seen by comparing the decoded speech, as shown in figure 7.2e, to the original shown in figure 7.2f.

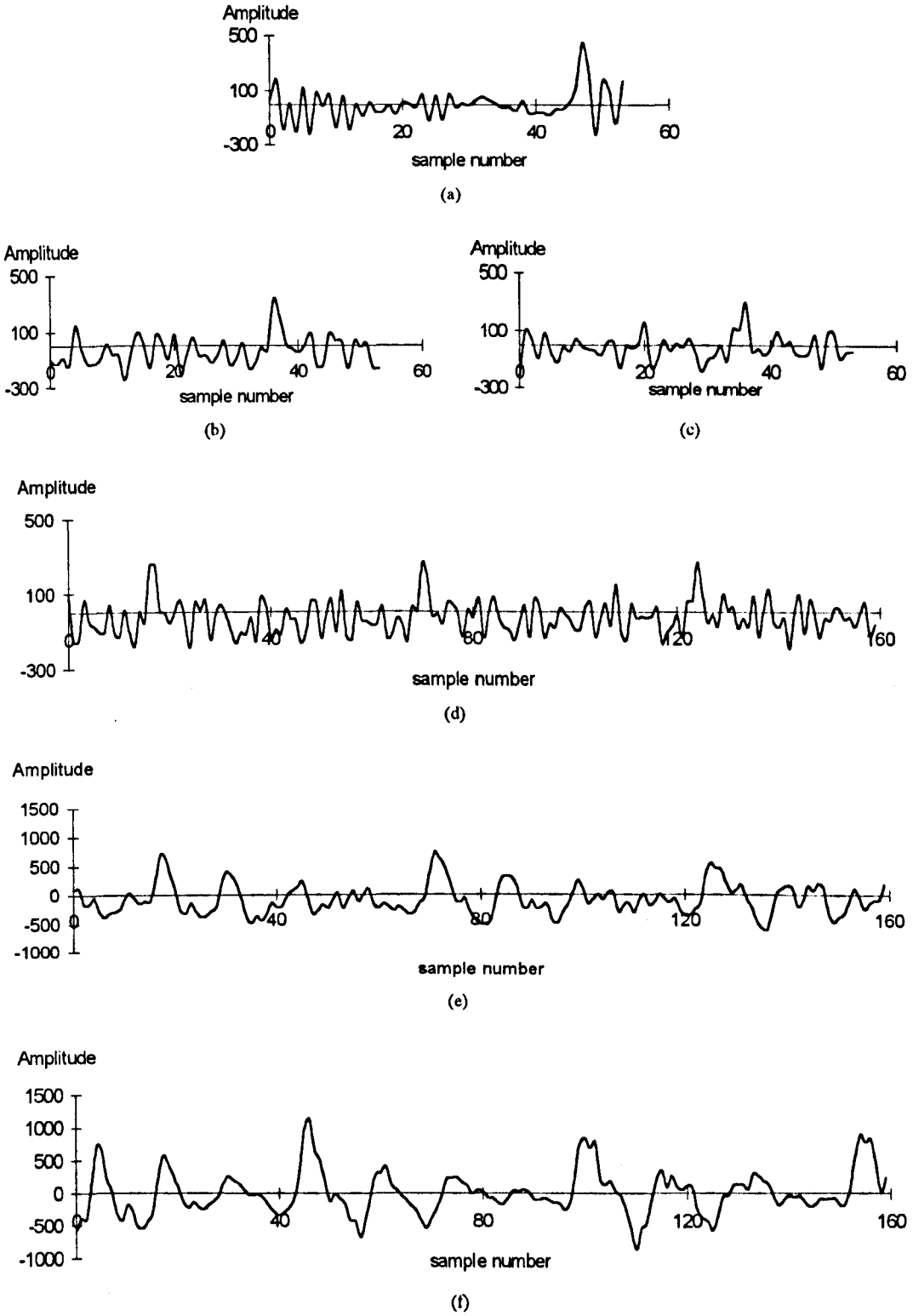


Figure 7.2 Example of a speech segment as represented by a differential quantiser.
(a) original prototype waveform (b) quantised previous prototype waveform (c) current quantised prototype waveform (d) reconstructed residual (e) synthetic speech (f) original speech

It was reported by Kleijn [70] that the performance of a prototype waveform quantiser can be enhanced by including a single-pulse code-book at the first-stage. The single-pulse code-book is populated by band-limited pulses centred on different locations. Hence, equation 7.1 becomes:

$$\underline{\hat{u}}^{(l)} = g_0 \underline{\hat{u}}^{(l-1)} + g_1 \underline{\gamma}_1[I_1] + g_2 \underline{\gamma}_2[I_2] \tag{7.8}$$

where $\gamma_1[I_1]$ is the optimum candidate from the single-pulse code-book and $\gamma_2[I_2]$ is the optimum candidate from the Gaussian code-book.

It was reported that a system with a single-pulse code-book and a sparse-pulse code-book, each of 7 bits, can provide excellent quality at excitation bit-rates of about 2kbits/s. This would necessitate an overall bit-rate of more than about 3.6kb/s which is too high for our application. Very little further detail about the single-pulse code-book is given in the literature [70]. Some work on the single-pulse code-book for the prototype waveform quantiser can also be found in Tang [17].

A fully quantised C-language implementation of the PWI coder described above has been developed [81] according to the block diagram shown in figure 7.1. The implementation has the optimised contribution from the previous quantised prototype waveform and the 8-bit Gaussian code-book, but does not have a single-pulse code-book. The bit-allocation table for this 2.4kb/s PWI coder is given in table 7.1. The constant g_0 is scalar-quantised to 4 bits. The logarithm of the constant g_1 is scalar-quantised to 4 bits. The speech classification and pitch-period is jointly quantised to 8 bits corresponding to values from 0 to 255. If the input speech is unvoiced the value zero (0000 0000) will be encoded. Otherwise values from 15 to 150, corresponding to the pitch-periods in samples, will be encoded.

Parameters	number of bits	update interval (ms)	bits/second
LSF's	24	20	1200
speech class & pitch-period	8	20	400
g_0	4	20	200
g_1	4	20	200
Gaussian code-book	8	20	400
Total			2400

Table 7.1 Bit allocation table for the 2.4kb/s PWI coder

Heard over good quality head-phones the speech was clearly intelligible and maintained speaker recognisibility. However, the reverberence referred to above proved to be a major problem and the speech was far from toll quality. The female speech was less reverberant than the male. Heard over loud-speakers, the reverberence was less noticeable and the speech quality appeared closer to the original.

This section has described a first attempt at quantising the original PWI/CELP coder as described by Kleijn [70]. A later section (section 7.4) will describe a more sophisticated quantisation scheme for the modified version of PWI, referred to as the TPSWI coder, devised during the course of this project and described in section 5.5 of this thesis. Before this, the next section briefly shows how quantisation techniques have evolved from that described above for PWI/CELP coder to the more advanced schemes now being applied to recent types of WI coder.

7.3 Quantisation of more recent WI coding techniques

As described in chapter 6, current WI techniques decompose speech into regular updates of a set of LP filter coefficients, the pitch-period, samples of the speech power contour and the spectral shapes of the REW and SEW [75][76]. These parameters are assumed to be independent of each other and different update rates may be used for different parameters. In one of the implementations [75][76], tenth order LP analysis is performed every 25ms. The ten LP filter coefficients are converted to LSF coefficients which are vector-quantised using 30-bit split VQ. The pitch-periods, assumed to be in the range from 20 to 147 samples, are quantised to 7 bits, again at intervals of 25ms. The power of each characteristic waveform extracted from the residual is computed and transformed to a measurement of speech power by an LP synthesis filter. The speech-domain power measurements are then converted to a logarithmic scale and are differentially quantised to 4 bits. An update rate of 80Hz is assigned to the power factors, i.e. two power factors are encoded for each 25ms of speech.

The REW at suitable update-points, referred to as sub-update-points, is characterised only by the general shape of its magnitude spectrum. A 4th order polynomial is used to characterise the shape of each REW magnitude spectrum. Therefore each shape-vector is 5-dimensional with each element a coefficient of the 4th order polynomial. The shape-vectors are quantised at a rate of 240Hz [75], i.e. at intervals of about 34 samples, using a 3-bit code-book. The REW code-book therefore contains 5-dimensional vectors, each vector element containing the coefficients of a 4th order polynomial. To retrieve the REW magnitude spectrum at the decoder at each sub-update-point, the set of polynomial coefficients are retrieved from the REW code-book and the polynomial is evaluated. The time-domain REW may then be synthesised by interpolating between Fourier series coefficients obtained at the sub-update points using quadratic instantaneous phase interpolation in the usual way. Each Fourier series coefficient is obtained by combining the DFT magnitude spectrum given by a 4th order polynomial with a pseudo-random phase spectrum [76]. In one publication [75], the Fourier series coefficients are produced twice per sub-update point with an interpolated or simply repeat ... magnitude spectrum and a further random phase spectrum placed mid-way between adjacent sub-update points. A REW generated in this way would simply be added to the corresponding SEW. Alternatively, the REW Fourier series coefficients may be combined with interpolated SEW parameters at each sub-update point so that the time-domain waveform may be generated by a single interpolation process. Computationally less intensive methods for performing the interpolation process have been recently proposed by Kleijn [76].

It was found [76] that the perceptual quality of the SEW can be reasonably well preserved at very low bit-rates by directly encoding its spectral magnitudes only up to 800Hz. The magnitudes above 800Hz are calculated by subtracting the magnitude spectrum of the REW from a flat spectrum, i.e. assuming that the magnitude spectrum of each characteristic waveform is flat above 800Hz. The 0Hz to 800Hz bandwidth SEW is quantised using an 8-bit code-book at intervals of 25ms. The SEW code-book contains 8-dimensional vectors, each vector element representing the magnitude spectral density in a frequency bin at a multiple of 100Hz. The SEW is regenerated in the time-domain using one of two phase spectra

depending on the proportion of the SEW in a characteristic waveform [74]. These phase spectra are reported [76] to represent on the one hand a pulse or on the other a "spread-out waveform". No further detail about the phase spectra is given in the literature [76]. However more detail about the quantisation of current WI coders can be found [75] and [76].

The ideas outlined in this survey have influenced the quantisation techniques applied to the two main coders developed in this thesis. The next section describes the quantisation of the "two-mode pitch synchronous waveform interpolation" (TPSWI) coder which was described in section 5.5. Section 7.5 deals with the quantisation of the "generalised pitch synchronous waveform interpolation" (GPSWI) coder which was described in section 6.4.

7.4 Quantisation of the TPSWI coder

Quantisation of the TPSWI coder developed in this project requires, at regular update-points, the quantisation of the short-term spectral envelope, the speech classification, the pitch-period and either a prototype waveform for voiced speech or a set of four gain factors for unvoiced speech. The short-term spectral envelope is represented by ten LSF coefficients which are vector-quantised at intervals of 20ms using the 24-bit IMS-LSF quantiser as presented in chapter 4. The pitch-period and the speech classification (voiced or unvoiced) are jointly quantised to 8 bits, at intervals of 20ms, as will be explained. The four gain factors for the frame of unvoiced speech preceding each 20ms update-point are expressed as logarithms and vector-quantised using an 8-bit code-book, with a mean-square-error (mse) distance measure.

For voiced speech, each prototype waveform is quantised using a gain-shape principle which means that the shape of a power normalised version of the prototype waveform and a gain factor are quantised separately. The logarithm of the gain factor is differentially quantised at intervals of 10ms using a 3-bit scalar quantiser. Each

prototype waveform is normalised to have an rms value of unity and the DFT magnitude spectrum of the resulting normalised waveform is transformed to a logarithmic magnitude scale to yield a shape-vector. For the purposes of vector quantisation, the shape-vector is evenly partitioned into two sub-vectors. The lower frequency sub-vector is quantised using a 6-bit code-book and the upper frequency sub-vector is quantised to 2 bits. The two shape code-books are searched using an analysis-by-synthesis principle the optimal vectors being chosen according to a perceptually weighted mean-square-error (mse) distance measure applied to the speech segments that would be produced by each candidate. A 2.3kb/s fully quantised TPSWI coder was obtained by encoding prototype waveforms at intervals of 20ms.

A 2.4kb/s version was also produced with the perceptual speech quality enhanced by doubling the update rate of prototype waveforms to 100Hz (10ms intervals). In the 2.4kb/s TPSWI coder, a 5-bit code-book was assigned to quantise the lower frequency sub-vector and the upper frequency sub-vector was assumed to be flat and required no bits. In the following sections, the details of the quantisation scheme for the 2.3kb/s coder will be presented. This quantisation scheme will then be further developed to achieve the 2.4kb/s TPSWI coder.

7.4.1 Quantisation of unvoiced gain factors

In the 2.3kb/s TPSWI coder, each 20ms frame of unvoiced speech preceding an update-point is divided into four sub-frames in order to determine a power contour. The rms values, P_1, P_2, P_3 and P_4 say, are computed for the sub-frames, transformed to logarithms and vector quantised using an 8-bit code-book. The vector $\underline{G}$ of unvoiced "log-gain" factors is defined as:

$$\underline{G} = \left[\log_{10} P_1, \log_{10} P_2, \log_{10} P_3, \log_{10} P_4 \right]^T$$

The log-gain factor code-book for unvoiced speech was trained with 15000 training vectors extracted from unvoiced portions of the speech file "GSP.DAT" [20]. A mean-square-error (mse) distance measure was used as defined below:

$$d(\underline{G}, \underline{\gamma}_j) = \frac{1}{4}\sum_{k=1}^{4}\left(G_k - \gamma_{jk}\right)^2 \tag{7.9}$$

where $\underline{\gamma}_j$ is the jth vector in the log-gain code-book whose elements are γ_{jk} for k=1, 2, 3 and 4. Some examples of the 256 different contours in the code-book are given in figure 7.3.

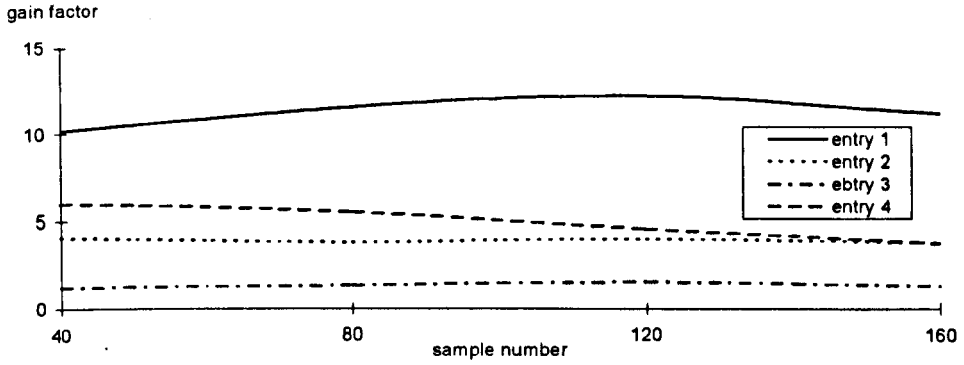


Figure 7.3 First four gain contours in the unvoiced log-gain code-book.

At the decoder, the code-book entry with the given index is fetched for the unvoiced log-gain vector at each update-point. The four rms values are recovered by:

$$\hat{\underline{p}} = \left[10^{\gamma_{j1}}, 10^{\gamma_{j2}}, 10^{\gamma_{j3}}, 10^{\gamma_{j4}}\right]^T$$

where $\underline{\gamma}_j$ is the optimal code-book vector

A pseudo-random sequence generator produces a 20ms normalised sequence which is multiplied sample-by-sample by a gain contour obtained by interpolating between the four quantised gain factors. This produces the reconstructed unvoiced residual.

7.4.2 Quantisation of prototype waveform gain factors

In the 2.3kb/s TPSWI coder, two gain factors are encoded per 20ms segment of voiced speech. One of these refers to the speech at the current update-point and the other is for an intermediate point mid-way between the current and the previous update-points. The intermediate gain factor is computed from an intermediate prototype waveform extracted using an interpolated pitch-period which is the harmonic mean of the pitch-periods at the current and the previous update-points. As discussed in section 6.4.3, a sudden boost in the intermediate gain factors may

occasionally arise with pitch-synchronous LP analysis, and therefore the restriction proposed in equation 6.13 must be applied.

At the encoder after the intermediate gain has been adjusted using equation 6.13, the two prototype waveform gain factors, $\lambda_1^{(l)}$ and $\lambda_2^{(l)}$, are transformed to a logarithmic scale to obtain $g_1^{(l)} = \log_{10}\lambda_1^{(l)}$ and $g_2^{(l)} = \log_{10}\lambda_2^{(l)}$. These "log-gain" factors are differentially scalar-quantised using a 3-bit code-book as will be now explained.

The difference between $g_1^{(l)}$ and $\hat{g}_2^{(l-1)}$ is compared with each candidate γ_j in the voiced log-gain difference code-book to calculate a distance measure defined as:

$$d_j = \left(g_1^{(l)} - a\,\hat{g}_2^{(l-1)} - \gamma_j\right)^2 \qquad (710a)$$

$$j = 0, 1, \ldots, L-1$$

where L is the number of code-book candidates, and a is a constant slightly less than one (0.95). The value of j which minimises d_j is taken as the index to the required quantised difference γ_j. $g_2^{(l)}$ is now differentially quantised by computing:

$$d_k = \left(g_2^{(l)} - a\,\hat{g}_1^{(l)} - \gamma_k\right)^2 \qquad (7.10b)$$

for each k and finding the value that minimises d_k.

At the decoder, the gain factors $\lambda_1^{(l)}$ and $\lambda_2^{(l)}$ are recovered from the previously quantised log-gain factor $\hat{g}_2^{(l-1)}$ and the optimal code-book entries γ_j and γ_k as:

$$\lambda_1^{(l)} = 10^{\left(a\hat{g}_2^{(l-1)} + \gamma_j\right)} \qquad (7.11a)$$

$$\lambda_2^{(l)} = 10^{\left(a\hat{g}_1^{(l)} + \gamma_k\right)} \qquad (7.11b)$$

The voiced log-gain factor code-book was trained using 5000 voiced frames extracted from the speech file "GSP.DAT" [20]. Two prototype waveforms were extracted from each 20ms voiced frame and their rms values were computed and adjusted using equation 6.13. The 10000 rms values were converted to logarithms

and thus became log-gain factors. Each logarithmic gain factor was then subtracted from the previous one to obtain a set of testing scalars. The scalar codebook for the log-gain factor were trained using the LBG-CS algorithm discussed in section 4.3.2, with the vector dimension set to one. The 3-bit code-book has 8 entries which were found to be as follows:

-0.580, 0.829, -0.147, 0.139, -0.297, 0.360, -0.053 and 0.027

The code-book training took about half an hour on a P-90 personal computer. The splitting factor and the distortion threshold in the LBG-CS algorithm were set to 0.99 and 0.0001 respectively.

7.4.3 Quantisation of prototype waveform shapes

In figure 7.4 a schematic diagram of the spectral shape quantiser used in the 2.3kb/s TPSWI coder is shown. In this coder, the shape of a prototype waveform is characterised by its power-normalised logarithmic DFT magnitude spectrum. The power normalisation means that prior to quantisation, the prototype waveform is scaled in amplitude such that its power becomes unity. The DFT magnitude spectrum of the unity power prototype waveform is quantised according to two shape code-books, each populated with 38-dimensional vectors. The first shape code-book contains shape-vectors for characterising the lower frequency components and the second shape code-book contains shape-vectors for the upper frequency components. The shape code-books are searched using an analysis-by-synthesis approach (a-b-s), in which each code-book vector is used to construct a prototype waveform transformed to the speech-domain by the LP synthesis filter. The speech segment is perceptually weighted and compared with a perceptually weighted and phase standardised version of the corresponding segment of original speech. The phase standardisation is carried out to place the vocal tract excitation point in the centre of the segment. Unquantised LP ladder filter coefficients are used to synthesise the original prototype waveform and quantised LP ladder filter coefficients are used to transform the shape code-books vectors. A mse distance measure is used to search for the optimal code-book candidate. The mse distance measure can be minimised by maximising the normalised cross-correlation coefficient between the two speech

segments as will be seen. The lower half-band code-book is first searched to find the optimal vector. The optimal vector selected from the first code-book is combined with each candidate in the second code-book to find the best overall vector.

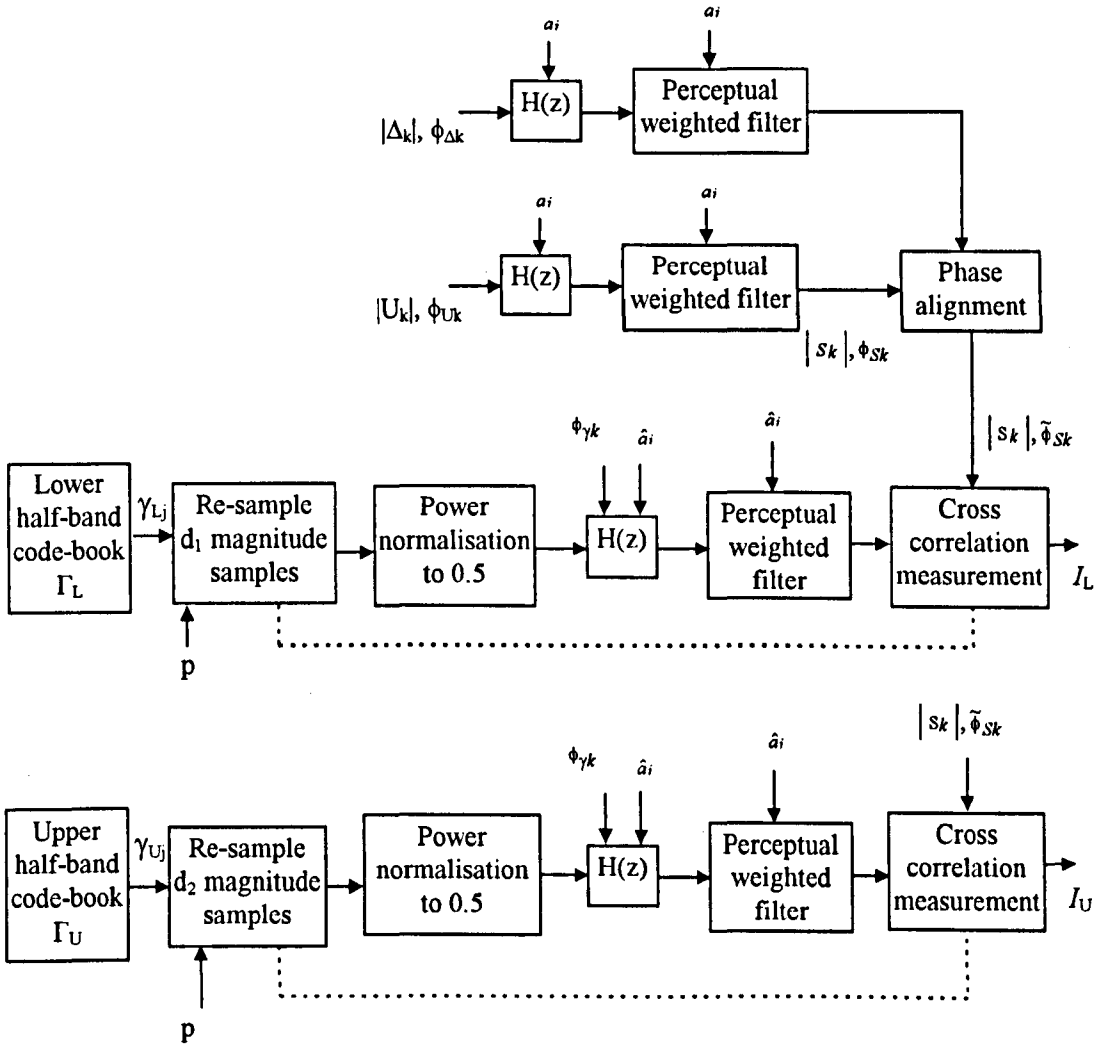


Figure 7.4 A spectral shape quantiser for the TPSWI coder.
(Details of the quantiser is discussed in section 7.4.3.1 to 6)

In sections 7.4.3.1 to 6, the design of the shape quantiser will be presented. A justification will be given for splitting each magnitude shape-vector into two sub-bands. The fact that each magnitude shape-vector may be of different length, determined by the pitch-period, means that a shape-vector in the codebook must be re-sampled to the required length, i.e. half of the current pitch-period. The way this is done will be described in section 7.4.3.4. Also the derivation of a frequency-domain method of applying the DFT magnitude and phase coefficients will be given.

7.4.3.1 Quantisation of shape-vectors using split VQ

Vector quantisation of prototype waveform shapes as represented by DFT magnitudes requires a high computational cost. The maximum length of a shape-vector is 76 samples since the maximum allowable pitch-period is made to be 150 samples and the DFT magnitude spectra are mirrored. Experiments were carried out to evaluate the relative importance of different frequency components in the DFT magnitude spectrum of a prototype waveform. For each prototype waveform a modified magnitude spectrum was defined as follows:

$$|U'_k| = \begin{cases} |U_k| & \frac{k\varpi f_s}{2\pi} < f_T \\ \sqrt{p} & otherwise \end{cases} \qquad (7.12)$$

where k = 0, 1, ... , p-1

p is the pitch-period in samples

$|U_k|$ is the original magnitude at DFT bin k

ϖ is the pitch-frequency normalised in radians per sample interval

f_T is a boundary frequency (Hz) below which the DFT magnitudes are unmodified and above which they are replace by a constant value $\sqrt{p}$

The modified magnitude spectrum therefore had the original magnitudes for frequencies smaller than the threshold f_T and a flat magnitude spectrum $\sqrt{p}$ otherwise. The value $\sqrt{p}$ was used because a completely flat pth order DFT magnitude spectrum with unity-power would have each magnitude sample equal to $\sqrt{p}$. The power of the modified magnitude spectrum was normalised to unity. A speech segment was then synthesised using the modified magnitude spectrum with the original phase spectrum to produce a prototype waveform which was then passed through the LP synthesis filter. The speech segment thus obtained was compared with the corresponding segment of original speech using a normalised cross-correlation coefficient to measure the similarity of the two speech segments. The normalised cross-correlation coefficient may be defined using frequency-domain representation:

$$C = \frac{\sum_{k=0}^{p} |S_k||\hat{S}_k| \cos\left(\phi_{Sk} - \hat{\phi}_{Sk}\right)}{\sqrt{\sum_{k=0}^{p} |S_k|^2 \sum_{k=0}^{p} |\hat{S}_k|^2}} \tag{7.13}$$

where

p is the pitch-period

$|S_k|$ and ϕ_{Sk} are magnitude and phase spectra of the original speech segment

$|\hat{S}_k|$ and $\hat{\phi}_{Sk}$ are magnitude and phase spectra of the speech segment synthesised by the modified prototype waveform

About 1000 prototype waveforms were considered and the average cross-correlation coefficient obtained for different values of the threshold f_T are presented in figure 7.5.

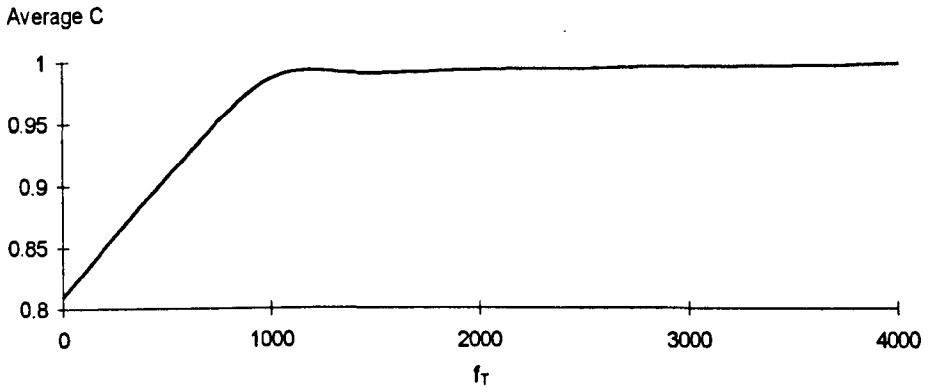


Figure 7.5 Average cross-correlation measure of a modified speech segment compared to the original using different f_T

Figure 7.5 shows that the average cross-correlation coefficient was always close to 1 for values of f_T above 1500Hz. In fact the average cross-correlation coefficient was always greater than 0.99 for f_T>1500Hz. The average cross-correlation coefficient dropped rapidly when f_T was moved towards the low frequency end. This suggests that the low frequency components in the DFT magnitude spectra of a residual prototype waveform are more important than the high frequency ones in order to regenerate a speech segment which looks similar to the original speech segment.

Using the above argument, the magnitude spectrum of a prototype waveform is evenly split into two equal bands. The first sub-band, which characterises the

magnitudes at the lower half frequency band, is quantised using a larger code-book. The upper half frequency band is quantised using a smaller code-book.

7.4.3.2 LP filtering in frequency-domain

Suppose a prototype waveform has magnitude and phase spectra $|U_k|$ and ϕ_{Uk} for k=0, 1, ..., p-1. The magnitude and phase spectra, $|S_k|$ and ϕ_{Sk} for k=0, 1, ..., p-1, of the corresponding speech segment can be calculated as follows:

$$|S_k| = \frac{|U_k|}{\left\{\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)\right)^2 + \left(\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)\right)^2\right\}^{1/2}} \tag{7.14a}$$

$$\phi_{Sk} = \phi_{Uk} + \tan^{-1}\left\{\frac{\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)}{\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)}\right\} \tag{7.14b}$$

where P is the order of the LP predictor and the ladder filter coefficients are 1, a_1, a_2, ..., a_p.

Conversely, $|U_k|$ and ϕ_{Uk} may be expressed in terms of $|S_k|$ and ϕ_{Sk} as follows:

$$|U_k| = |S_k|\left\{\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)\right)^2 + \left(\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)\right)^2\right\}^{1/2} \tag{7.15a}$$

$$\phi_{Uk} = \phi_{Sk} - \tan^{-1}\left\{\frac{\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)}{\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)}\right\} \tag{7.15b}$$

The derivations of equations 7.14 and 7.15 are given in appendix C.

7.4.3.3 Perceptual weighting in frequency-domain

Perceptual weighting may be included in the distance measure between two speech segments by passing each of them through a perceptual weighting filter prior to the distance measurement. The perceptual weighting filter aims to exploit the characteristics of the human ear that it is more sensitive to the noise around the spectral valleys than in the formant regions [95]. This is due to the noise masking

effect of the high spectral energy at a formant. The perceptual weighting filter is made adaptive according to the short-term spectral envelope of the input speech as determined by the LP analysis. If the prediction is 10th order and the LP ladder coefficients are 1, a_1, a_2, ..., a_{10}, then a suitable perceptual weighting filter has transfer function:

$$P_w(z) = \frac{1 - \sum_{i=1}^{10} \alpha_1^i a_i z^{-1}}{1 - \sum_{i=1}^{10} \alpha_2^i a_i z^{-1}} \tag{7.16}$$

Such perceptual weighting is commonly used in CELP [86] the principle being to place a pole and a zero close to each pole, $re^{j\theta}$ say, identified by LP analysis. The zero is at $\alpha_1 re^{j\theta}$ and the pole is at $\alpha_2 re^{j\theta}$. The effect of the zeros are to flatten the spectral valleys and to remove spectral tilt. The poles re-introduce formants which are bandwidth expanded, i.e. they have lower Q-factors. Suitable values for α_1 and α_2 have been found to be 0.9 and 0.6 [87] respectively. The perceptual weighting filter can be implemented in the time-domain by cascading an LP analysis filter $A(z/\alpha_1)$ with an LP synthesis filter $1/A(z/\alpha_2)$, or using frequency-domain computations. An example of the power spectral envelope of a speech segment processed by the perceptual weighting filter with transfer function $P_w(z)$ as defined above is shown in figure 7.6.

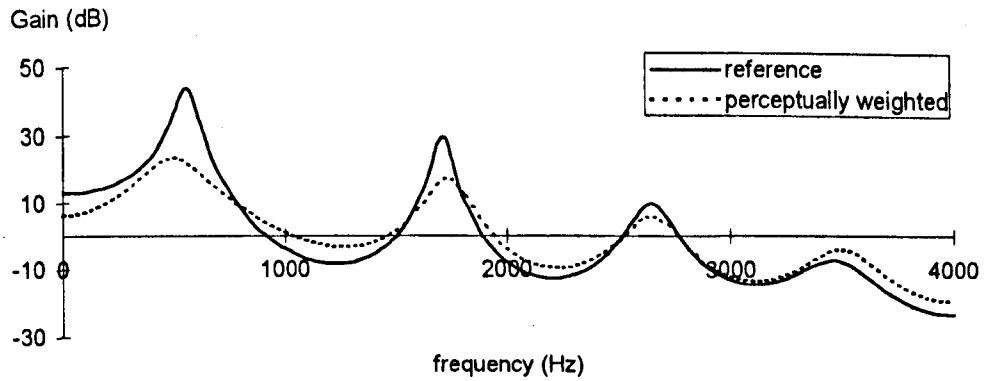


Figure 7.6 Spectral envelope of a speech segment modified by the perceptual weight filter with (α_1 = 0.9 and α_2=0.6)

It can be seen in figure 7.6 that the energy at speech formants in the perceptually weighted spectral envelope is attenuated while there is an increase in energy in the spectral valleys. This effectively de-emphasises the differences in the formant regions and assigns relatively more weight to the differences in the spectral valleys.

The use of perceptually weighted time-domain differences is fundamental to the principle of analysis-by-synthesis and is commonly used in speech coding especially CELP and PWI. Experiments were carried out to confirm the advantages of the perceptual weighting and a clear improvement in perceptual speech quality was found to be obtained through its use. Further, although the use of an error measure in the residual-domain simplifies the code-book search it was found not to be as effective as the speech-domain approach.

7.4.3.4 Searching of shape code-books

Prior to searching the shape code-books, the residual prototype waveform to be quantised is processed by the LP synthesis filter to yield a "reference" speech segment. This segment is perceptually weighted and phase-aligned with a synthetic speech segment, of the same length p, produced by passing a synthesised pulse-like prototype waveform through the same LP synthesis filter and perceptual weighting filter. This is done to align the reference speech segment such that it corresponds as closely as possible to a waveform that would be produced when the vocal tract excitation point occurs exactly in the centre of the waveform. Minimising a perceptually weighted speech domain error rather than a residual-domain error was found most effective in searching for the best match. The pulse-like prototype waveform is defined in the frequency-domain in such a way that its main peak in the time-domain will occur exactly in the middle of the p-length segment. Its magnitude spectrum is defined to be flat and its phase spectrum is made to conform to the assumed phase model developed in section 5.6.1. Its pth order DFT magnitude and phase spectra are therefore:

$$|\Delta_k| = 1 \tag{7.17a}$$

$$\phi_{\Delta k} = k\pi - \phi_F(k\varpi) \qquad k = 0, 1, \ldots, p-1 \tag{7.17b}$$

where ϖ is the pitch-frequency. $\phi_F(k\varpi)$ is the phase spectrum of a 2nd-order all-pass filter and is computed from equation 5.35 with coefficient α set to 0.95 and β is chosen according to table 5.1 using the current pitch-period. The linear term $k\pi$ places the main peak of the pulse exactly in the middle of the segment,

The magnitude and phase spectra, $|S_k|$ and $\tilde{\phi}_{Sk}$, of the perceptually weighted and phase standardised reference speech segment are used as a means of searching the two shape code-books to quantise the residual-domain prototype waveform.

The lower frequency-band shape code-book, which contains 38-dimensional frequency-domain vectors, is first searched. The construction and training of the shape code-books will be discussed in the next section. Each 38-dimensional shape-vector in the code-book is re-sampled using linear interpolation to provide a shape-vector of the required dimension, which is:

$$d_1 = Integer\left(\frac{p}{4} + 0.5\right) \tag{7.18}$$

where p is the current pitch-period in samples, and the integer function rounds down.

The re-sampled shape-vector is converted from its logarithmic scale back to a linear magnitude spectrum and is then normalised to have a power of 0.5. A phase spectrum $\phi_{\gamma k}$ is now defined as in equation 7.19.

$$\phi_{\gamma k} = \phi_{\Delta k} = k\pi - \phi_F(k\varpi) \tag{7.19}$$

where $\phi_{\Delta k}$ is as defined in equation 7.17b, for the pulse-like waveform that was used to phase-standardise the reference waveform.

The power-normalised lower band magnitude spectrum of each code-book entry is multiplied by the corresponding lower band gain response of the LP synthesis filter followed by that of the perceptual weighting filter to produce the lower band magnitude spectrum $|S_{Lk}|$, for k=0, 1, ... ,d_1-1, of a perceptually weighted speech segment. The phase response in equation 7.19 is added to that of the LP synthesis filter as in equation 7.14b and also to that of the perceptual weighting filter to produce a phase spectrum $\phi_{\Gamma k}$. A full band phase spectrum is then defined as:

$$\hat{\phi}_{Sk} = \begin{cases} \phi_{\Gamma k} & 0 \le k\varpi < \omega_9 \\ \tilde{\phi}_{Sk} & otherwise \end{cases} \tag{7.20}$$

This phase spectrum is required for computing the speech-domain cross-correlation coefficient which measures the similarity between the code-book entry and the prototype waveform which is to be quantised. Actually, the phase spectrum $\hat{\phi}_{Sk}$ is the same for all code-book entries and therefore needs to be calculated only once. Above the 9th LSF, $\hat{\phi}_{Sk}$ is made the same as in the reference waveform so that only magnitude differences become significant. This is done because the phase will be randomised at the decoder at frequencies above the 9th LSF, as explained in section 5.6.5. Equation 7.20 is valid across the frequency range 0 to π and will be used when searching the upper band codebook as well as the lower band code-book.

Therefore a perceptually weighted lower band frequency-domain representation of a speech segment for each code-book entry is compared with the lower band of the reference vector using a mse distance measure. The mse distance measure is minimised by maximising the normalised cross-correlation coefficient between the two segments. The higher the normalised cross-correlation coefficient is, the more similar to the reference will be the speech segment produced by the code-book candidate. The normalised cross-correlation coefficient between the two speech segments is calculated in the frequency-domain as follows:

$$C(j) = \frac{\sum_{k=0}^{d_1} |S_k||S_{Ljk}| \cos\left(\tilde{\phi}_{Sk} - \hat{\phi}_{Sk}\right)}{\sqrt{\sum_{k=0}^{d_1} |S_k|^2 \sum_{k=0}^{d_1} |S_{Ljk}|^2}} \tag{7.21}$$

The lower band code-book candidate which yields the maximum cross-correlation coefficient is chosen as the quantised version of the lower band components of the reference vector. Its index, I_L say, is recorded.

The upper frequency-band code-book is searched in the same way as the lower one. Each 38-dimensional code-book vector is re-sampled to the required dimension:

$$d_2 = Integer\left(\frac{p}{2}\right) + 1 - d_1 \tag{7.22}$$

where p is the pitch-period.

The vector is then converted from a logarithmic scale to linear. For each upper frequency range code-book vector, the power of the corresponding upper half magnitude spectrum is normalised to 0.5. The upper band magnitude spectrum is processed by the LP synthesis and perceptual weighting filter to obtain $|S_{Uk}|$, for $k=d_1, d_1+1, \ldots, d_1+d_2-1$. By maximising the cross-correlation coefficient:

$$C(j) = \frac{\sum_{k=d_1}^{d_1+d_2-1} |S_k||S_{U_jk}| \cos\left(\tilde{\phi}_{Sk} - \hat{\phi}_{Sk}\right)}{\sqrt{\sum_{k=d_1}^{d_1+d_2-1} |S_k|^2 \sum_{k=d_1}^{d_1+d_2-1} |S_{U_jk}|^2}} \tag{7.23}$$

the optimum candidate, with index I_U say, from the upper frequency-band code-book is found. The indices of the two optimum code-book entries are encoded to represent the shape of the prototype waveform.

Refer to figure 7.4 for a summary of the TPSWI shape quantisation process.

7.4.3.5 Training of the shape code-books

The shape code-books were trained using 10000 training vectors extracted from voiced speech portions of a clean speech file "GSP.DAT" [20]. The training vectors were assembled by extracting prototype waveforms from the LP residual. Each prototype waveform was normalised to have unity power, and pitch-period length DFT analysis was performed to yield its magnitude spectrum. The logarithmic magnitude spectrum of each prototype waveform was up-sampled to a fixed length using linear interpolation. The fixed length is $p_{max}/2+1$ where p_{max} is the maximum anticipated pitch-period, which is 150 samples for the TPSWI coder. Therefore the fixed length is 76 samples. After time-scaling to the fixed length, the logarithmic magnitude spectrum is normalised so that the maximum value is zero. The resulting magnitude spectrum is then evenly split into 2 sub-bands each represented by a 38-dimensional shape-vector. The shape-vectors thus obtained were stored as training vectors.

Using the 10000 lower frequency and 10000 upper frequency training vectors, the LBG-CS algorithm discussed in section 4.3.2 was used to train a lower frequency and an upper frequency shape code-book. In each case, a mse distance

measure was used and the splitting factor and the distortion threshold set to 0.99 and 0.0001 respectively.

7.4.3.6 Reconstruction of the excitation signal

To reconstruct each prototype waveform, the decoder uses the received code-book indices to fetch the required shape sub-vectors. The two shape sub-vectors are re-sampled to the required lengths d_1 and d_2 determined from the received pitch-period p and transformed back to a linear scale to yield the quantised half-band magnitude spectra. The two half-band magnitude spectra are combined to form the quantised full-band magnitude spectrum $|\hat{U}_k|$ for k=0, 1, ... , p/2. The power of the full-band magnitude spectrum $|\hat{U}_k|$ is normalised to unity. The synthetic phase spectrum $\hat{\phi}_{Uk}$ is defined as follows:

$$\hat{\phi}_{Uk} = \begin{cases} \phi_{\gamma k} & 0 \le k\varpi < \hat{\omega}_9 \\ \phi_r(k) & \hat{\omega}_9 \le k\varpi < \pi \end{cases} \qquad (7.24)$$

where $\phi_{\gamma k}$ is as given by equation 7.19, $\phi_r(k)$ is a uniformly distributed pseudo-random phase between $-\pi$ to π, a different random phase being generated for each frequency bin k, and $\hat{\omega}_9$ is the 9th quantised LSF at the current update-point. This phase spectrum will randomise the phases above $\hat{\omega}_9$ as described in section 5.6.5. The magnitude and phase spectra $|\hat{U}_k|$ and $\hat{\phi}_{Uk}$ are now taken as the DFT frequency-domain description of the current quantised prototype waveform.

The current quantised prototype waveform is phase-aligned, as described in section 5.3, with the previous aligned prototype waveform. The real and imaginary DFT coefficients of the current aligned prototype waveform are computed and these are interpolated with the previous DFT coefficients and combined with the interpolated pitch-period to yield an interpolated signal. The interpolated signal is scaled to the power contour specified by the quantised gain factors to yield the reconstructed residual signal. The gain factors are linearly interpolated on a sample-by-sample basis.

7.4.4 Bit allocation of the 2.3kb/s TPSWI coder

A 2.3kb/s TPSWI coder was constructed using the quantised scheme discussed in the above. The bit allocation scheme of the coder is listed in table 7.2. The speech classification and pitch-period are jointly quantised at each update-point using 8 bits. A value of zero (0000 0000) means that the current synthetic speech frame is unvoiced. Values 15 to 150 indicate that the speech is voiced and represent the pitch-period in sampling intervals. This leaves some values undefined and available for error protection.

Parameters	number of bits	update interval (ms)	bits/second
LSF's	24	20	1200
speech class & pitch-period	8	20	400
power contour	3	10	300
lower half-band code-book	6	20	300
upper half-band code-book	2	20	100
Total			2300

Table 7.2 Bit allocation table for the 2.3kb/s TPSWI coder

Comparing the decoded speech from the fully quantised TPSWI coder with the unquantised version of it, using the speech file "OPERATOR.DAT" [21], the decoded male speech from the quantised coder was found to be a little more synthetic and transient distortion occurred from time to time. The coder tended to preserve the naturalness and general quality of female speech rather better than for male speech. Only minor degradation was found in the decoded female speech obtained from the quantised coder in comparison to the same speech obtained from the model.

7.4.5 Bit allocation of the 2.4kb/s TPSWI coder

It was suggested in chapter 5 that the perceptual quality of speech from the TPSWI coder can be enhanced by increasing the update rate of prototype waveforms. A 2.4kb/s TPSWI coder was constructed in which a prototype waveform is encoded at intervals of 10ms rather than 20ms. The main update-points at which the LSF's and the pitch-period are encoded remain at intervals of 20ms. The pitch-period for the intermediate prototype waveform between the main update-points is taken to be the harmonic mean of the two instantaneous pitch-periods at the update-points. The

intermediate prototype waveform is extracted mid-way between the two main update-points as discussed in section 6.4.2. The inclusion of a second peak must be avoided in the intermediate prototype waveform by locating the peak in the centre of the segment as described in section 6.4.2. To achieve the required low bit-rate, the upper frequency-band shape code-book is replaced by a flat spectrum which has a magnitude of $\sqrt{p}$, where p is the current pitch-period. In searching the lower half-band code-book, each code-book candidate is processed to yield a half-band magnitude spectrum. The power of the magnitude spectrum is normalised to 0.5. The power normalised spectrum is combined with the upper half-band spectrum (i.e. the flat spectrum), to form a full-band magnitude spectrum and the processes outlined above then proceeds as for the 2.3kb/s coder. The bit allocation table of the 2.4kb/s TPSWI coder [83] is presented in table 7.3.

Parameters	number of bits	update interval (ms)	bits/second
LSF's	24	20	1200
speech class & pitch-period	8	20	400
power contour	3	10	300
lower half-band code-book	5	10	500
upper half-band code-book	0	10	0
Total			2400

Table 7.3 Bit allocation table for the 2.4kb/s TPSWI coder

Comparing the fully quantised 2.3kb/s and 2.4kb/s TPSWI coders using the speech file "OPERATOR.DAT" [21], the decoded male and female speech from the latter coder seemed to contain less transient distortion than the 2.3kb/s coder. However, the quality of the decoded male speech remained a little synthetic, and as before, the female speech was generally more natural.

C-language programs implementing, fully quantised, the 2.3kb/s and 2.4kb/s versions of the TPSWI coder are presented in two internal reports [96] and [83]. These programs are simulations in the sense that although they are intended as prototypes for real-time DSP implementations, the problems of real-time implementation have not been addressed. The programs take as input a computer file of 8kHz sampled 4kHz bandwidth speech with 16 bits per sample and produce a

corresponding output file. Problems of channel errors have also not yet been considered.

The C-programs are not optimised in any way. Clearly the TPSWI technique as quantised in this section is rather complex and would require much simplification if it were to become the basis of a real-time coder. Many simplifications are possible following suggestions made by Kleijn [76]. As mentioned in the introduction to this chapter fundamental difficulties with the TPSWI approach lie in the switched mode (voiced/unvoiced) operation and in the sampling of the prototype waveform evolution. The "generalised pitch-synchronous waveform interpolation" (GPSWI) coder described in chapter 6 addresses these problems and the quantisation of this coder will be considered in the next section.

7.5 Quantisation of the GPSWI coder

Quantisation of the GPSWI coder requires the quantisation of the short-term spectral envelope, the pitch-period, the characteristic waveform gain factors, the SES, the RES and the "separating" frequency, f_{SR}, which divides the 0 to 4kHz (0 to π radians/sample) frequency band into the SES and RES sub-bands. The separating frequency may be specified as being equal to either 0 radians/sample, the 3rd, 6th or 9th LSF. The short-term spectral envelope is quantised at 20ms intervals using the 24-bit IMS-LSF quantiser discussed in chapter 4. Two characteristic waveform gain factors are quantised and encoded for each 20ms synthesis frame in the same way as for the TPSWI coder in voiced mode. The pitch-period is quantised using 7 bits at each 20ms update-point, the range of possible pitch-periods being set from 16 to 143 samples. The SES is defined from zero to the separating frequency which may be as high as the 9th LSF. The SES code-book is composed of full-band (0 to 4kHz) shape-vectors stored as 72 log-magnitude spectral samples per vector. An mse distance measure is used to compare perceptually-weighted versions of the speech segments generated by the original SES magnitude and by each of the SES code-book candidates. Once again the mse distance measure is minimised by maximising

the normalised cross-correlation coefficient between the two speech segments as defined in equation 7.13. Different weightings are assigned to different frequency bands of the SES when the normalised cross-correlation coefficient is computed. The RES is defined from the separating frequency to 4kHz. The separating frequency can be as low as 0Hz, therefore the RES code-book contains full-band (0 to 4kHz) shape-vectors. A fully quantised 2.4kb/s version of the GPSWI coder has been designed and tested. This version encodes one SES and two RES per 20ms of speech. In the following sections, the quantisation of the SES and RES will be discussed.

7.5.1 Assignment of weighting factors in searching the SES code-book

To evaluate the importance of different frequency components in a SES, a similar experiment to that described in section 7.4.3.1 was performed. A frequency threshold f_T was defined and random errors were injected into the magnitudes of the SES at frequencies from f_T to 4kHz. The error corrupted SES was defined as:

$$\hat{U}_{sk} = \begin{cases} U_{sk} & \dfrac{k\varpi f_s}{2\pi} < f_T \\ \left(1+\hat{U}_{err}(k)\right) U_{sk} & otherwise \end{cases} \qquad (7.25)$$

where U_{Sk} is the magnitude of the original SES for k=0, 1, ..., p-1

$\hat{U}_{err}(k)$ was generated as a uniformly distributed pseudo-random number between 0 and some constant U_{err} which lies between 0 and 1. A different random number was generated for each value of k. U_{err} is expressed as a percentage of unity. The noise corrupted SES was combined with the original phase spectrum to synthesise a characteristic waveform which was then converted to the speech-domain. The speech segment was compared with the one produced by the original SES, the similarity between the two segments being quantified by a cross-correlation coefficient as defined in equation 7.13. Different values of U_{err} and f_T were tested. The average cross-correlation coefficient measures over 1000 SES magnitude spectra typical of natural speech are presented in figure 7.7.

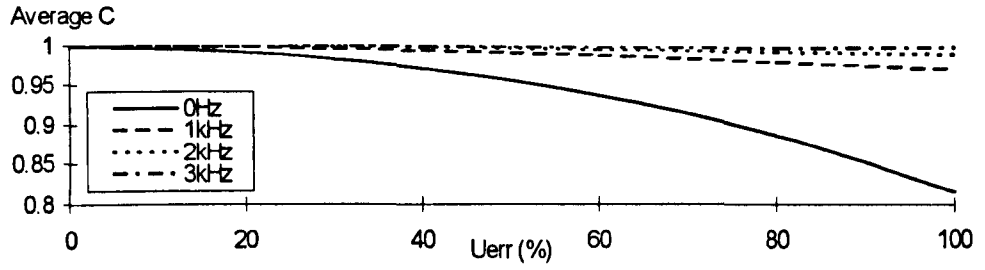


Figure 7.7 Average cross-correlation coefficient from modified SES compared to original for different values of f_T under various magnitude error criteria

Results in figure 7.7 show that the average cross-correlation coefficient remained close to 1 for all values of U_{err} when f_T was above 2kHz. When f_T was set to 1kHz, the average cross-correlation coefficient measure started to drop as the percentage error U_{err} increased. The average cross-correlation coefficient dropped rapidly as the percentage error increased when f_T was defined as 0Hz. This suggests that magnitudes around 0Hz to 1kHz are the most important. Magnitudes between 1kHz and 2kHz are less important than those below this range. Finally, magnitudes above 2kHz are the least important and large magnitude error is allowed in this range.

A second experiment was conducted, in which case the magnitude errors were introduced only in a fixed frequency band. Four frequency bands were tested, they were 0-1kHz, 1-2kHz, 2-3kHz and 3-4kHz. Different values of U_{err} were tested and the average cross-correlation coefficient (equation 7.13) measured are presented in figure 7.8.

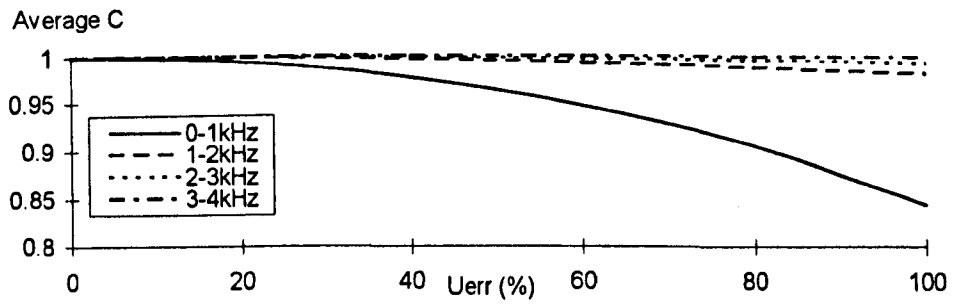


Figure 7.8 Average cross-correlation coefficient between original SES and an SES obtained by injecting magnitude errors into different frequency bands.

The results in figure 7.8 re-confirm that larger magnitude error is allowed at the high frequency bands of a SES than in the low frequency band. In case of the 2-3kHz and 3-4kHz bands, the reduction in the average cross-correlation coefficient is minute

even when U_{err} was 100%. A reduction in the average cross-correlation coefficient was observed in the 1-2kHz band, when U_{err} was increased. The average cross-correlation coefficient dropped substantially when U_{err} was introduced into the 0-1kHz band. According to the results of the two experiments, different weights may be assigned to different frequency bands in a SES during code-book training as well as code-book searching in order to increase the efficiency of the SES quantiser.

7.5.2 The SES/RES quantiser

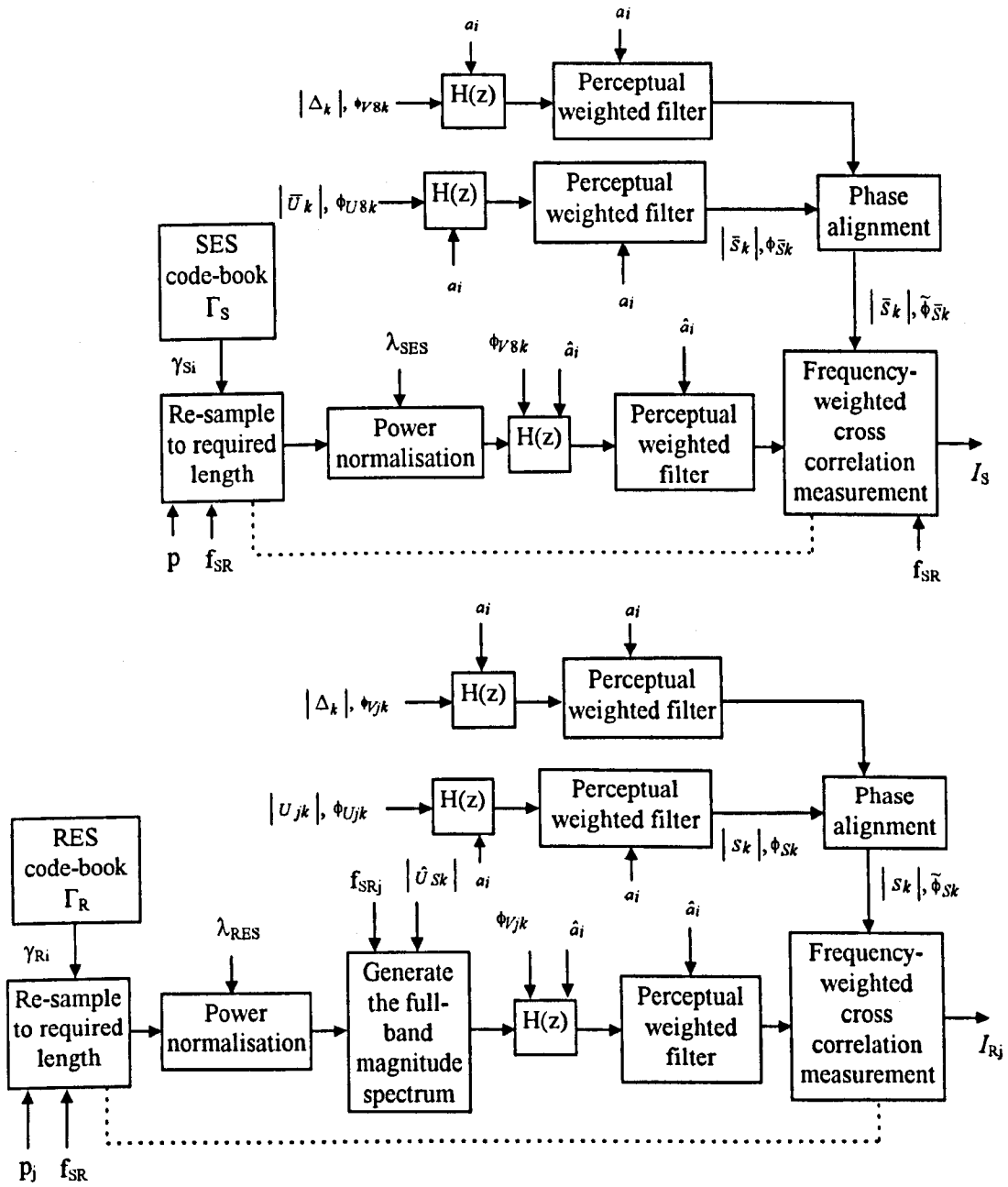


Figure 7.9 Spectral shape quantiser for the GPSWI coder

In figure 7.9, the schematic diagram of a SES/RES quantiser for the 2.4 kb/s GPSWI coder [84] is shown. The value of the separation frequency, f_{SR} in radians per sample, which is equal to 0, the 3rd, 6th or 9th LSF, is determined from the average voicing confidence level, $\overline{VL}$, as described in section 6.4.3. The current value of $\overline{VL}$ is obtained by averaging the previous and the current voicing confidence levels as produced by the two-way pitch detector (TPD) described in chapter 5.

The SES and RES quantisers operate similarly to the shape quantiser in the TPSWI coder. The full-band SES code-book is first searched. Each SES code-book candidate is re-sampled to obtain p/2+1 samples and these are transformed to a linear magnitude scale. The magnitude samples beyond the current value of f_{SR} are set to zero. The power of the SES candidate is scaled to λ_{SES}, where

$$\lambda_{SES} = \frac{f_{SR}}{\pi} \tag{7.26}$$

The value of λ_{SES} is proportional to f_{SR} with a constant of proportionality such that it would become unity if f_{SR} were equal to π, i.e. half the sampling frequency.

The synthetic phase spectrum:

$$\phi_{V_{jk}} = k\pi - \phi_F\left(k\varpi_j\right) \qquad 0 \le k\varpi_j \le \pi \tag{7.27}$$

is assigned to each of the power-normalised SES candidates where $\phi_F\left(k\,\varpi_j\right)$ is as defined in equation 5.35. The resulting magnitude and phase spectra are processed by the LP synthesis filter and the perceptual weighting filter to yield the magnitude and phase spectra of a perceptually weighted speech segment.

The speech segment thus obtained is compared with the reference segment by computing a normalised cross-correlation coefficient defined only from zero to f_{SR}. Weighting factors are used to increase the weighting of the low frequency components. The resulting weighted cross-correlation coefficient between the perceptually weighted speech segment obtained from SES code-book candidate i and the reference segment is:

$$C(i) = \frac{\sum_{k=0}^{K} W_k \left| S_k \right| \left| S_{Si,k} \right| \cos\left(\tilde{\phi}_{Sk} - \phi_{\gamma k} \right)}{\sqrt{\sum_{k=0}^{K} \left| S_k \right|^2 \sum_{k=0}^{K} \left| S_{Si,k} \right|^2}} \tag{7.28}$$

where the magnitude and phase spectra $\left| S_{Si,k} \right|$ and $\phi_{\gamma k}$ are as obtained from SES code-book candidate i. The reference magnitude spectrum $\left| S_k \right|$ at the given update-point is the mean magnitude spectrum $\left| \overline{U}_k \right|$ transformed to the perceptually weighted speech-domain. $\left| \overline{U}_k \right|$ is the average of the magnitude spectra of eight characteristic waveforms centred at approximately 2.5ms intervals within a 20ms segment preceding the update-point. The reference phase spectrum, $\tilde{\phi}_{Sk}$, is the phase spectrum, ϕ_{U8k}, of the single characteristic waveform closest to the update-point also transformed to the perceptually weighted speech-domain. (Phase averaging was tried, deriving $\tilde{\phi}_{Sk}$ from the average of 8 consecutive phase spectra rather than the single one used here, but this was unsuccessful). The DFT bin number K corresponds to the separating frequency f_{SR} at the current update-point and is calculated as follows:

$$K = Integer\left(\frac{f_{SR}\, p}{2\pi} \right) \tag{7.29}$$

The weighting factors are set as:

$$W_k = \begin{cases} 1.0 & 0 \le k\varpi < \pi/4 \\ 0.5 & \pi/4 \le k\varpi < \pi/2 \\ 0.25 & \pi/2 \le k\varpi < \pi \end{cases} \tag{7.30}$$

The quantised SES at each update point is interpolated at the encoder with the quantised SES at the previous update-point to yield an SES magnitude at each of 8 sub-update points. This is done for the purpose of searching the RES code-book at each sub-update point as will now be described.

To search the RES code-book, the SES at each sub-update point j is combined with each of the candidates in the RES code-book to generate a trial full-band magnitude spectrum at the sub-update point. Each RES code-book candidate is re-sampled from its standard length of 72 samples to a length equal to $p_j/2+1$ where p_j is the instantaneous pitch-period for the current sub-update point obtained by interpolation. The re-sampled RES shape-vector is transformed to a linear magnitude scale and the magnitudes at frequencies below f_{SR} are set to zero. The power of the modified RES candidate is scaled to λ_{RES}, where $\lambda_{RES} = 1.0-\lambda_{SES}$. The resulting RES is then combined with the interpolated SES to form the magnitude spectrum of a trial characteristic waveform at the sub-update point. The magnitude spectrum is multiplied by the gain response of the LP synthesis filter and perceptual weighting filter to generate the magnitude response $|S_{R_i k}|$ of a speech segment. This magnitude spectrum is compared with the perceptually weighted magnitude spectrum of the reference segment, $|S_k|$ say, at each sub-update point using the cross-correlation coefficient defined as:

$$C(i) = \frac{\sum_{k=0}^{\frac{p_j}{2}} |S_k||S_{R_i k}|}{\sqrt{\sum_{k=0}^{\frac{p_j}{2}} |S_k|^2 \sum_{k=0}^{\frac{p_j}{2}} |S_{R_i k}|^2}} \tag{7.31}$$

The index of the RES code-book candidate which results in the highest cross-correlation coefficient, i.e. produces a speech segment whose magnitude spectrum looks maximally like the original in a mean square error sense, is encoded.

The procedure described above can be simplified as a result of the RES and SES being defined for different frequency bands, i.e. 0 to f_{SR} and f_{SR} to π. The simplification is achieved by comparing only the RES part of the spectrum. Note that smoothing of the reference RES is implied by this process since the code-book entries are themselves smoothed and allow only the general shape of the RES to be represented in a minimum mean-square-error sense.

7.5.3 Bit allocation of the 2.4kb/s GPSWI coder

The 2.4kb/s GPSWI coder [84] encodes a SES and two RES spectra at intervals of 20ms. The bit allocation table for the coder is given in table 7.4.

Parameters	number of bits	update interval (ms)	bits/second
LSF's	24	20	1200
pitch-period	7	20	350
power contour	3	10	300
SES code-book & f_{SR}	7	20	350
RES code-book	2	10	200
Total			2400

Table 7.4 Bit allocation table for the 2.4kb/s GPSWI coder

In contrast to the TPSWI coder, there is no direct speech classification and model switching for the GPSWI coder. Although the separating frequency f_{SR} is dependent on a voicing confidence level, the coder performance will be much less critically dependent on this than it would be on a model switching voiced/unvoiced decision. The pitch-period is encoded in 7 bits representing a range from 16 to 143 samples. Two characteristic waveform gain factors are encoded for each 20ms synthesis frame. As for the TPSWI coder in voiced mode, the first gain factor refers to the speech for an intermediate point mid-way between the current and the previous update-points. The second gain factor refers to the speech at the update-point. The separating frequency between the SES and RES is embedded in the 7-bit SES code-book index as will now be described.

A SES code-book index of zero means that no SES is required, i.e. the separating frequency is 0Hz. For indices 1 to 15 the SES is non-zero only from 0Hz to the 3rd LSF. For indices 16 to 63, the SES is non-zero up to the 6th LSF and the remaining indices, i.e. from 64 to 127, are devoted to the SES which is defined from 0Hz to the 9th LSF. When the SES code-book is searched, only the candidates corresponding to the particular selection of separating frequency are tested, i.e. entries 1 to 15 are tested when f_{SR} is equal to the 3rd LSF, entries 16 to 63 are tested when f_{SR} is equal to the 6th LSF and entries 64 to 127 are tested when f_{SR} equals the 9th LSF. When f_{SR} is equal to zero, searching SES code-book is unnecessary. The coder directly searches the 2-bit RES code-book containing four RES shape-vectors

as described in section 7.5.2. The training procedures adopted for the SES and RES code-books are presented in the next section.

7.5.4 Training SES and RES code-books

The SES has a variable bandwidth extending from 0 radians/sample to the separating frequency f_{SR}. The magnitude spectrum of the SES from f_{SR} to π radians/sample is taken to be zero. A set of SES training vectors may be generated from a reference file of natural speech by computing the mean magnitude spectra obtained from eight consecutive characteristic waveforms centred on update-points at 2.5ms intervals. Each mean magnitude spectrum must be transformed to logarithmic form and up-sampled to the standard length of 72 samples using linear interpolation. The magnitudes at frequencies higher than f_{SR} must be set to zero (effectively) by setting the log-magnitudes exactly equal to -B where B is a suitably large integer (300). The spectrum thus obtained should be normalised such that the maximum value is zero. The value of f_{SR} is recorded with each training vector.

The required SES code-book comprises three sub-codebooks: one for f_{SR} = LSF3, one for f_{SR} = LSF6 and one for f_{SR} = LSF9. The training vectors must therefore be divided into three sets corresponding to the three possibilities for f_{SR}. Training the SES code-book for f_{SR} = LSF9, say, by directly applying the LBG-CS algorithm to training vectors obtained as described above may not be appropriate since the effect of the zero magnitudes (log-magnitudes equal to -B) above f_{SR}, which varies as LSF9 varies, would be included in the distance measure where they would have no effect on the speech in practice. The LBG-CS algorithm must therefore be modified such that the zero magnitudes are excluded from the computation of distance.

Suppose we have a set of N-dimensional training vectors for the "f_{SR} =LSF9" SES codebook. Each training vector may contain "zero-elements" beyond the value of LSF9 applicable to the speech segment that produced the training vector. These elements, being artificially inserted as exactly -B in the log-magnitude domain, are

assumed to be distinguishable from any naturally occurring elements which are extremely unlikely to be exactly -B. With a conventional mse distance measure, the centroid of a cell of training vectors is simply obtained by computing a vector whose elements are the mean values of the corresponding vector elements across all the vectors in the cell. This calculation of the centroid must now be modified slightly. To obtain the kth element, γ_k, for the centroid $\underline{\gamma}$ of a cell of training vectors $\{\underline{t}_1, \underline{t}_2, \ldots, \underline{t}_M\}$, the mean of only the non-zero kth elements is calculated as follows:

$$\gamma_k = \frac{1}{M'_k} \sum_{j=1}^{M} t_{jk} \tag{7.32}$$

where M'_k is the total number of non-zero kth elements for the training vectors in the cell. For assigning a training vector $\underline{t}$ to a cell with representative vector $\underline{\gamma}$, the mse distance measure is now modified from equation 4.9 to become:

$$d(\underline{t}, \underline{\gamma}) = \frac{1}{N'} \sum_{k=1}^{N} (\gamma_k - t_k)^2 \tag{7.33}$$

where N' is the number of non-zero elements in the training vector $\underline{t}$ which is of dimension N. The number of non-zero elements is easily ascertained by counting the number of elements with log-magnitude exactly equal to -B.

The LBG-SC procedure with mse distance measure modified as described above was used to train the three SES code-books using the three sets of training vectors extracted from the speech file "DSP.DAT" [20]; i.e. a set for f_{SR} = LSF3 with 15 entries, one for f_{SR} = LSF6 with 48 entries and one for f_{SR} = LSF9 with 64 entries. A further modification to the training procedure was necessary to obtain the first two of these code-books since they contain numbers of entries which are not powers of two.

The first training set contained 1000 shape-vectors each with non-zero elements only from 0Hz to the 3rd LSF. A 16-level sub-codebook was trained using the first training set, the centroid whose cell contained the lowest number of training vectors was removed and LBG procedure was repeated to obtain the required sub-codebook with 15 levels. The second training set contained 5000 training vectors each with non-zero elements only up to the 6th LSF. These vectors were used to

train a sub-codebook with 64 levels. The 16 centroids whose cells contained the lowest number of training vectors were taken away one by one, the LBG algorithm being repeatedly applied until a 48-level sub-codebook was obtained. The third training set contained 10000 shape-vectors with non-zero elements from 0Hz to the 9th LSF. These vectors were used to train a 6-bit, 64 levels, sub-codebook for the SES corresponding to f_{SR} = LSF9.

The RES also has a variable bandwidth extending from f_{SR} radians/sample to π radians/sample. The magnitude spectrum of the RES from zero to f_{SR} will be taken to be zero when searching the RES codebook. A set of RES training vectors may be generated from the reference file of natural speech by extracting characteristic waveforms centred on update-points at 2.5ms intervals, computing the DFT magnitude spectra and transforming these to logarithmic form. Cepstral smoothing is then applied by taking the inverse DFT to obtain the real cepstrum, windowing to zero all cepstral samples beyond the fourth, zero-padding to 144 samples (maintaining mirroring as described in section 6.3.1) and transforming back to the frequency-domain via a 144-point DFT. This produces a smoothed log-magnitude vector of the required standard length of 72 frequency domain samples, with its mirror image. The log-magnitudes at frequencies lower than f_{SR} are now set to -B and the spectrum thus obtained is normalised such that the maximum value is zero. The smoothing approach used here is an alternative to the fitting of a polynomial as in Kleijn[76]. The value of f_{SR} is recorded with each training vector.

The required RES code-book comprises four sub-codebooks: one for f_{SR} = zero, one for f_{SR} = LSF3, one for f_{SR} = LSF6 and one for f_{SR} = LSF9. The training vectors must therefore be divided into four sets corresponding to the four possibilities for f_{SR}. Training the RES sub-codebook for f_{SR} = zero proceeds using the standard LBG-SC algorithm. For the other sub-codebooks, the LBG-CS algorithm must be modified such that the zero magnitudes below f_{SR} are excluded from the computation of distance, in a similar manner as for the training of the SES sub-codebooks.

The RES code-book is therefore populated with 72-dimensional smoothed spectral envelope shape-vectors, for the convenience of combining them with the SES vectors which are also 72-dimensional. About 5000 training vectors were used to train the 2-bit RES sub-codebook for the case where $f_{SR} = 0$. The training vectors were extracted from the unvoiced speech portions of the speech file "GSP.DAT" [20]. The training vectors for the other three 2-bit RES sub-codebooks were extracted as for the corresponding SES sub-codebooks. The SES and RES codebooks were trained with the splitting factor and the distortion threshold set to 0.99 and 0.0001 respectively.

A C-program implementing the fully quantised 2.4kb/s version of GPSWI is presented in the internal report [84]. The 2.4kb/s GPSWI coder requires about 20% more processing time than the 2.4kb/s TPSWI coder. The GPSWI coding algorithm and the C-program would therefore need to be considerably simplified, for example using methods recently proposed by Kleijn [76], to achieve real-time application. In the following section, the speech quality obtained from the 2.4kb/s TPSWI and GPSWI coders will be evaluated and compared with that obtained from five international standardised coders.

7.6 Performance evaluation of the 2.4kb/s TPSWI coder and GPSWI coder

The 2.4kb/s GPSWI coder [84] was compared with the 2.4kb/s TPSWI coder [83] using the speech file "OPERATOR.DAT" [21] and other reference speech files. These files were different from those used to train the quantisers. As previously mentioned, the TPSWI coder produced decoded speech which was generally of good quality though there was clearly some loss of naturalness and transient distortion occurred from time to time. The speech was of "communications" rather than "toll" quality these terms being defined in Boyd [97]. The speech synthesised by the GPSWI coder was more natural and less distorted as compared to that from the TPSWI coder, although a degree of unnaturalness could still be found in the male

voice when the pitch-period was particularly large. Overall, the speech quality from the GPSWI coder was clearly better than that from the TPSWI coder. The disadvantage of the GPSWI coder is its high computational complexity as compared to the TPSWI coder. This is due to the application of spectral decomposition to both voiced and unvoiced speech, the fact that eight DFT's are required per 20ms of speech and the need to search two full-band shape code-books to quantise the characteristic waveforms at each update-point.

The 2.4kb/s GPSWI coder was compared with the 32kb/s ADPCM (G726) coder using a file of 32kb/s ADPCM decoded speech provided by BT laboratory. Informal listening tests showed that the decoded speech from the GPSWI coder is not as good as that produced by the ADPCM coder, there being some loss of naturalness and also occasional transient distortion.

The 2.4kb/s GPSWI coder was also compared with 4 coders:-

a) 2.4kb/s LPC-10e coder[14],
b) 4.1kb/s IMBE coder [13],
c) 2.4kb/s ME-LPC coder [67],
d) 2.4kb/s AT&T WI coder [76].

The LPC-10e coder is currently, and has been for many years the American DoD standard 2.4k/s speech coder. The IMBE coder has been adopted by INMARSAT for the "Sky-phone" system. The ME-LPC coder was the winner of the competition based on the specification for the new American DoD 2.4kb/s speech coder. The AT&T WI coder was one of the candidates for the 1996 DoD standardisation competition.

Informal listening tests showed that the speech quality obtained from the 2.4kb/s GPSWI coder was substantially better than that produced by LPC-10e. Furthermore it was also better than that obtained from the 4.1kb/s IMBE coder. The decoded speech obtained from the 2.4kb/s GPSWI coder was found to be very close to that of the ME-LPC and WI coders, though the speech quality obtained was perhaps not quite as natural.

7.7 Conclusions

Quantisation schemes proposed in the literature for PWI and WI coders have been considered in this chapter and quantisation strategies for 2.4kb/s versions of the TPSWI and GPSWI coders have been devised.

In the TPSWI coder, the pitch-period and speech classification are jointly quantised using 8 bits. The 2.4 kb/s TPSWI coder encodes a prototype waveform every 10ms for voiced speech. A prototype waveform is quantised using a gain-shape approach in which the gain factors are quantised in 3 bits using differential VQ. Experiments showed that the magnitudes of the lower frequency components of a prototype waveform are likely to be more important than those in the higher frequency region. The magnitudes in the lower half of the DFT frequency spectrum are therefore quantised in 5 bits and the magnitudes in the upper half DFT frequency spectrum are assumed to be flat. In the case of unvoiced speech, four gain factors are vector-quantised using an 8-bit code-book.

It was concluded that, fully quantised, the 2.4kb/s TPSWI coder is capable of producing communication quality speech and that the use of 10 ms rather than 20 ms update intervals for the prototype waveforms was beneficial even though the upper 2kHz frequency band was not encoded but was instead replaced by a flat spectrum. The speech contained a degree of buzziness and transient distortion which, it was concluded, may be reduced by the use of a generalised WI model and better modelling of the evolution of spectral features.

In the GPSWI coder, the model-switching according to a voiced/unvoiced decision is eliminated and the evolution of spectral features is decomposed into slowly and rapidly evolving components which are quantised separately and in different ways. The pitch-period is encoded in 7 bits. Two gain factors are encoded for a 20ms synthesis frame, these being quantised as for the TPSWI coder. A slowly evolving magnitude spectrum (SES) is encoded at intervals of 20ms using 7 bits, and a rapidly evolving magnitude spectrum (RES) is quantised using 2 bits at 10ms

update intervals. The SES and RES code-books are populated with full-band shape-vectors each standardised to a length of 72 samples.

Informal listening tests suggested that the decoded speech obtained from the fully quantised 2.4kb/s GPSWI coder is better than that from the fully quantised 2.4kb/s TPSWI coder. However the computational complexity of the GPSWI coder is about 20% higher. The decoded speech produced by the 2.4 kb/s GPSWI coder was found to be not as natural as that from 32 kb/s ADPCM. It was substantially better than that from the LPC-10e coder and better than that from the 4.1kb/s IMBE coder. The decoded speech produced by the 2.4kb/s GPSWI coder was very close to that from the 2.4kb/s ME-LPC and AT&T WI coders, though the latter two had a more natural speech quality.

Chapter 8

Conclusions, achievements and future work

8.1 Conclusions

This thesis is concerned with the use of waveform interpolation techniques for speech coding at very low bit-rates i.e. 2.4kb/s. The original idea of prototype waveform interpolation (PWI) has been explored and its evolution towards current generalised waveform interpolation (WI) techniques has been investigated and followed up. The major innovations in current WI techniques lie in the generalisation of the voiced model to unvoiced speech and the decomposition of the evolution of spectra and periodicity into slowly and rapidly evolving components. In principle, and in practice with many implementations of WI, the decomposition is achieved by one-dimensional low-pass and high-pass filtering to achieve full 0 to 4kHz band slowly and rapidly evolving waveforms or spectra.

In this thesis, to achieve the required 2.4kb/s bit-rate, a sub-band approach is proposed for the decomposition where a slowly evolving spectrum is obtained only for a lower frequency band and a rapidly evolving spectrum is obtained for a higher frequency band. A magnitude spectrum only is modelled, the phase being regenerated according to an "all-pass" phase model developed at Liverpool University. Justification for the sub-band approach is given in the thesis. The sub-band approach is similar in some ways to the most recent approach proposed by Kleijn [75][76] though it was discovered by the author before Kleijn's paper [75] appeared.

The operation of WI coders depends critically on accurate pitch detection and linear prediction (LP) analysis. Work on this thesis began with the development of a reliable pitch determination algorithm and a pitch synchronous LP analysis technique

for voiced speech, based on Burg's algorithm, to refine the analysis obtained from the more conventional autocorrelation method.

A two-way pitch detector (TPD) has been designed in which segments of speech are classified as either voiced or unvoiced and an estimate of the true pitch-period is given for voiced speech. The nature of the speech is classified on the basis of a voicing confidence level and the power ratio between the speech and a band-pass filtered version of it. The voicing confidence level is computed from the voicing probabilities of four features determined from the speech. The pitch-period for the voiced speech is determined using a backward-mode cross-correlation function which is applied on a band-pass filtered speech residual. Experimental results showed that the cross-correlation function computed from the band-pass filtered residual signal had more predominant peaks around the location of the true pitch-period as well as integer multiples of it. Furthermore, the amplitudes of the sidelobes in the cross-correlation function were effectively attenuated. A pitch post-processing unit has been included in the TPD, the function of which is to eliminate possible multiple pitch-period errors and to provide a smooth pitch-contour.

The TPD was tested for clean nature speech and speech corrupted by different SNR levels of white, car, babble and multi-speaker noise. The results suggested that a 97.6% speech classification accuracy and a 90.5% pitch estimation accuracy were obtained for clean natural speech. The performance of the TPD was fairly consistent for noisy speech with SNR levels higher than 20dB. The performance of the TPD deteriorated seriously when the SNR dropped below 20dB.

Two line spectral frequency (LSF) analysis filter structures have been proposed and incorporated into an LSF synthesis filter. It was discovered that if the LSF analysis filter was not a "true inverse" of the LSF synthesis filter, transient effects occurred at the output of the synthesis filter when the LSF coefficients were made adaptive. The term "true inverse" was applied to arrangements which eliminate these transient effects. A true inverse LSF analysis and synthesis filter pair has been obtained. Objective and subjective tests have been conducted to investigate the performance of the LSF filter pair. The results indicated that the LSF analysis and

synthesis filters worked as well as conventional ladder filter structures. By using LSF filters in a speech coder, the computational complexity required in some coders (e.g. CSA-CELP [10]) of converting LSF's to LP ladder filter coefficients is avoided. Furthermore, maximum smoothness of spectral envelope evolution is achieved by linearly interpolating the LSF coefficients across adjacent update-points on a sample-by-sample basis.

LP analysis using the autocorrelation method and Burg's method have been investigated and compared. Experimental results showed that the two methods were sensitive to the position of the analysis window location within the speech. In all the experiments performed, Burg's LP analysis method always yielded a more accurate spectral estimation than the autocorrelation method. Burg's method has further merits that an even more accurate spectral estimation is possible for pitch-synchronous LP analysis and that the accuracy is maintained when the length of the analysis window becomes smaller than a complete pitch-cycle. A variable length rectangular window is proposed for use with Burg's pitch-synchronous LP analysis method. The size of the analysis window is decided according to the pitch-period of the voiced speech at the update-point.

A 24-bit IMS-LSF quantiser, which is able to achieve an almost spectral transparent quality, has been designed. The quantiser implements an interframe quantisation scheme in which the difference between the current and previous quantised LSF vectors is quantised using a multistage-split VQ. A 10-bit code-book is used to search the full dimension difference vector at the first stage. At the second stage, two 5-dimension 7-bit code-books are used. Power weighting is included in the code-book searching to enhance the quantiser performance. A re-ordering scheme is proposed to maintain the ordering property of LSF's in an LSF vector and hence to ensure the stability of the all-pole filter. Different code-book arrangements have also been examined for the 24-bit quantiser and the results suggested that comparable performance can also be achieved by arranging the code-books in either 9_7^8 and 8_8^8 formats. The advantages of the latter arrangements is a reduction in the computation complexity.

A two-mode pitch-synchronous waveform interpolation (TPSWI) coder has been developed. The TPSWI coder is composed of two modules: a PSWI model for voiced speech and a pseudo-random sequence generator for unvoiced speech. An overlap-add technique is employed to preserve a smooth switching between the two models. Informal listening tests suggested that the overall perceptual speech quality of the TPSWI decoded speech was better than that obtained from the PWI/CELP coder. The perceived quality of the TPSWI modelled speech is very close to the original when updating a prototype waveform every 2.5ms, i.e. 8 prototype waveforms within each interval of 20ms.

A prototype waveform can be characterised by a gain-shape approach. The gain factor is the rms value of the prototype waveform. The shape is characterised by its DFT magnitude and phase spectra. Subjective tests suggested that no noticeable degradation in speech quality occurs when only 2 gain factors per 20ms of speech are encoded instead of all eight, when eight prototype waveforms are extracted. This is beneficial in keeping the bit-rate to a minimum.

Coding efficiency can be increased by sending only the magnitude spectrum of the prototype waveforms and deriving the phase spectrum at the decoder. To devise a way of artificially generating a phase spectrum at the decoder the commonly used fundamental voiced speech production model has been studied. It has been shown to be more reasonable to take the glottal excitation to be the time-reversed impulse response of a 2nd-order all-pole filter rather than the true impulse response of such a filter. Under this assumption, the phase spectrum of a prototype waveform can be assumed to resemble the negated phase spectrum of the 2nd-order all-pass filter. One of the two coefficients for the 2nd-order all-pass filter is fixed as 0.95. The other coefficient is adjusted according to the pitch-period of the prototype waveform. Informal listening tests suggested that only minor degradation in the female speech was caused by using the all-pass derived phases instead of the original. However a degree of unnaturalness was introduced in male speech with particularly low pitch frequency. It was found that the perceptual speech quality of the decoded male speech can be enhanced by randomising the phases in the higher frequency region.

This high frequency region was defined to be from the 9th LSF at the current update-point to 4kHz.

A generalised pitch-synchronous waveform interpolation (GPSWI) devised in this project utilises a sub-band approach to decompose characteristic waveforms to slowly and rapidly evolving components. The sub-band approach means that the magnitude spectrum of each characteristic waveform is separated into two frequency bands. The lower frequency band is used to characteristic the general shape of the characteristic waveform which is considered to be slowly evolving and may be sampled at a relatively low rate. The general shape of the characteristic waveform is recovered by interpolation at the decoder. The higher frequency band is used to characterise the more random-like structure of a characteristic waveform which is considered to be changing more rapidly. Only the general features of the higher frequency band are encoded, the random-like signal being re-generated at the decoder by injecting random phases into the decoded general magnitude spectral shape.

Informal listening tests suggested that the speech obtained from the GPSWI model was better than that obtained from the TPSWI model. The GPSWI modelled speech was virtually indistinguishable from the original when eight rapidly evolving (RES) spectra were encoded every 20ms. Good speech quality was still maintained when only one RES was encoded every 20ms. The GPSWI model worked well for noisy speech even when the SNR was as low as 0dB.

A fully quantised 2.4kb/s TPSWI coder was obtained by updating a prototype waveform every 10ms. In this coder the gain factor for each prototype waveform is differentially quantised to 3 bits. The magnitudes in the lower DFT magnitude spectrum is quantised to 5 bits. The magnitudes at the upper DFT magnitude spectrum is assumed to be flat. A 2.4kb/s GPSWI coder is obtained by transmitting a SES, 2 RES and 2 gain factors every 20ms. Quantisation of the gain factors is the same as for the TPSWI coder. The SES and RES are quantised to 7 and 2 bits respectively.

Informal listening tests suggested that the perceptual quality of the 2.4kb/s GPSWI decoded speech is better than that obtained using the 2.4kb/s TPSWI coder, 2.4kb/s LPC-10e coder and 4.1kb/s IMBE coder. It is close to the 2.4kb/s ME-LPC coder and 2.4kb/s AT&T WI coder, though the two latter coders have a slightly more natural speech quality. Comparing the 2.4kb/s GPSWI coder with the 32kb/s ADPCM coder, the decoded speech from the former was noticeably more synthetic.

8.2 Summary of achievements

1. A two-way pitch detector with the following features:-
   a) A voicing confidence level is given as a form of speech classification.
   b) An exponential probability density function is used to compute the probability of voicing.
   c) The pitch detection is done on the basis of a cross-correlation function defined for an LP residual obtained from a speech signal bandlimited from 100 Hz to 1kHz.
   d) The use of a conditional based 3-point median smoother applied to the current estimate of the pitch-period and the pitch-periods obtained from the two previous frames.

2. Implementation of a "true inverse" LSF analysis and synthesis filter pair in a speech coder.

3. LP analysis using Burg's pitch-synchronous method with an adaptive analysis window. The window size is dependent on the current pitch-period within the speech frame.

4. A fully quantised 2.4kb/s two-mode pitch-synchronous waveform interpolation (TPSWI) coder:-
   a) Implementation of the PSWI model for voiced speech.
   b) Implementation of a pseudo-random sequence generator for unvoiced speech.

c) Implementation of an overlap-add technique to ensure a smooth switching between the two coding models.

5. A means of deriving the phase spectrum of a prototype waveform based on the phase response of a 2nd-order all-pass filter. The phase derivation scheme was developed from a study of the voiced speech production model, in which the glottal excitation signal is assumed to be the time reversed impulse response of a 2nd-order all-pole filter.

6. A fully quantised 2.4kb/s generalised pitch-synchronous waveform interpolation (GPSWI) coder with the following features:-
   a) A sub-band approach to the definition of slowly and rapidly evolving components of characteristic waveforms.
   b) Adaptation of the separating frequency of the slowly and rapidly evolving spectrum according to the voicing confidence level given by the pitch detector.

8.3 Future work

The TPSWI coder works promisingly well for clean natural speech. However, the perceptual quality of TPSWI decoded speech deteriorates seriously when the speech is corrupted by noise. One of the sources of the degradation that occurs with noisy speech is the model used for unvoiced speech which tends to be used also for background noise which is very dissimilar to speech. Difficulties arise since the background noise may not necessarily be random-like as, for example, with car noise and multi-speaker noise. To improve the robustness of the TPSWI coder, the pitch detector in the TPWSI coder may be modified such that it classifies the input speech as quasi-periodic or random-like rather than as voiced or unvoiced speech. The advantage of this is that background noise with strongly periodic components may be better modelled as periodic signals rather than random signals.

Although the perceptual quality of the GPSWI coder is better than the TPSWI coder, it is computationally more complex. Some effort must now be devoted to reducing the complexity of the GPSWI coder. Obvious improvements that may be made are the standardisation of shape-vectors to a length which is a power of two (rather than 72) thus allowing the FFT to be used for interpolation, smoothing and some other operations. There is much scope for improving the efficiency of the analysis-by-synthesis comparisons for the code-book searches. Also the need for sample-by sample interpolation of LSF coefficients must be questioned, it being likely that some form of block-wise interpolation will prove satisfactory. More efficient phase alignment procedures are possible than are currently used in the GPSWI coder, and many other procedures could be streamlined. Once these obvious improvements have been made, probably at little or no cost to speech quality, there are approximations that may be made as in Kleijn [76] to achieve further savings.

Experimental results have suggested that when the pitch-period of the speech is especially large, rather synthetic speech quality is obtained when the "two-pole all-pass" derived phase spectrum is used instead of the original phase spectrum. Further research must be carried out to improve the phase derivation scheme. This research may find better ways of locating the pole/zero positions, and of deciding how and when to randomise phase at higher frequencies.

More effort is needed to improve the shape quantisers in the TPSWI and GPSWI coders. The efficiency of the existing vector quantisers is questionable since the dimension of the shape-vector is large, i.e. 38-dimensions for the TPSWI coder and 72 for the GPSWI coder. One possible solution is to transform the shape-vectors into other domains and to perform quantisation on the parameters in the new domain. For example the shape-vectors may be parameterised using polynomial curve fitting and quantising polynomial coefficients. Alternatively, windowed cepstral coefficients as produced during the RES spectral smoothing and interpolation process (section 7.5.4) may be used to characterise the shape-vectors.

This thesis has been concerned with the design of very low bit-rate (2.4kb/s) speech coders. Although the ultimate goal of research in this area is to achieve toll

quality at 2.4kb/s, it is still a very difficult goal to achieve. The types of coder being proposed to approach the goal, including the ones in this thesis, are extremely complex and are based on concepts that are in general not yet fully understood. For example, the models of perception on which the REW and SEW separation are based need further study. Therefore the work in this thesis is just a further stage in a long term research effort which is currently the subject of world-wide research interest. Further research will undoubtedly lead to the ultimate goal.

There is commercial interest in speech coders operating at 2.4kb/s and below which do not provide toll quality. Such applications may be found within the internet system for example and in some military communication systems. There is also much interest in speech coders which are specifically designed for operation at variable frame rates over various types of packet switched networks. Waveform interpolation techniques are clearly applicable to such commercial applications and the ideas presented in this thesis hopefully contribute to the knowledge that will be successfully be exploited in this way. Further research will find ways of adapting WI coding techniques of realisable complexity to specific possibly less demanding applications.

There are also important applications for the use of waveform interpolation coders at bit-rates higher than 2.4kb/s. One such application is half rate GSM requiring a speech coder at around 4.8kb/s, allowing for bit-error protection. The requirement here is definitely toll quality and there would be a considerable challenge in deciding how best to adapt the coders presented in this thesis to the increased bit-rate capacity.

Waveform interpolation appears to be the most promising approach to the future development of low bit-rate speech coding. It is still a new technique and requires much further development and study at a fundamental level. Applications for low bit-rate coding at even increasing bit-rates will continue to exist for the foreseeable future.

References

[1] P. E. Papamichalis, "Practical Approaches to Speech Coding", Prentice-Hall, 1987.

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

[3] S. Furui and M. M. Sondhi, "Advances in speech signal processing", Marcel Dekker Inc., 1992.

[4] R.W. Schafer and J. D. Markel, "Speech Analysis", New York: IEEE Press 1979.

[5] M. R. Schroeder, "Pitch Determination of Speech Signal", New York Springer 1983.

[6] J. D. Markel and A. H. Gray, "Linear Prediction of Speech", New York, Springer-Verlag, 1976.

[7] CCITT Recommendation G712, "Transmission performance characteristics of Pulse code modulation", 1992.

[8] CCITT Recommendation G726, "32kb/s adaptive differential pulse code modulation (ADPCM)", 1990.

[9] CCITT Recommendation G728, "Coding of speech at 16kbit/s using low-delay code excited linear prediction", 1992.

[10] ITU Draft Recommendation G729, "Coding of speech at 8kbit/s using conjugate-structure algebraic-code-excited linear predictive (CS-ACELP) coding", 1995.

[11] ETSI/GSM Recommendation 06.10, "European digital cellular telecommunication system (Phase 1) : Full-rate speech transcoding", 1992.

[12] EIA Interim Standard IS-54, "Cellular system : Dual-mode subscriber equipment-network equipment compatibility specification", 1989.

[13] Inmarsat Satellite Communications Services, "Inmarsat-M system definition, Issue 3.0 - module 1: system description", 1991.

[14] T. E. Tremain, "The government standard linear predictive coding algorithm : LPC-10", Speech technology, vol.1, pp. 40-49, April 1982.

[15] W. T. K Wong, R. M. Mack, B. M. G. Cheetham and X. Q. Sun, "Low rate speech coding for telecommunications", BT Technology Journal, vol. 14, pp. 28-44, Jan. 1996.

[16] M. K. Y. Lo, "Pitch-synchronous speech coding at very low bit rates", Ph.D. dissertation, Dept., of Electrical Engineering and Electronics, The University of Liverpool, 1993.

[17] J. K. W. Tang, "Variable frame length harmonic coding at very low bit rates", Ph.D. dissertation, Dept., of Electrical Engineering and Electronics, The University of Liverpool, 1995.

[18] H. B. Choi, "British Telecom full-year technical report 1994", Dec. 1994.

[19] H. B. Choi, "British Telecom full-year technical report 1995", Dec. 1995.

[20] B. T. Laboratory, "Speech file - GSP.DAT".

[21] B. T. Laboratory, "Speech file - OPERATOR.DAT".

[22] L. R. Rabiner and M. R. Sambur, "Application of an LPC distance measure to the voiced-unvoiced-silence detection problem", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-25, pp. 338-343, Aug. 1977.

[23] L. R. Rabiner, C. E. Schmidt and B. S. Atal, "Evaluation of a statistical approach to voiced-unvoiced-silence analysis for telephone-quality speech", The Bell System Technical Journal, pp. 455-482, March 1977.

[24] D. G. Childers, M. Hahn and J. N. Larar, "Silent and voiced/unvoiced/mixed excitation (four-way) classification of speech", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-37, pp. 1771-1774, Nov. 1989.

[25] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg and C. A. McGonegal, "A comparative performance study of several pitch detection algorithms", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-24, pp. 399-417, Oct. 1976.

[26] C. A. McGonegal, L. R. Rabiner and A. E. Rosenberg, "A semiautomatic pitch detector (SAPD)", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-23, pp. 570-574, Dec. 1975.

[27] M. M Sondhi, "New methods of pitch extraction", IEEE Trans. Audio Electroacoust., vol. AU-16, pp. 262-266, June 1968.

[28] L. R. Rabiner, M. R. Sambur and C. E. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-23, pp. 552-557, Dec. 1975.

[29] J. J. Dubnowski, R. W. Schafer and L. R. Rabiner, "Real-time digital hardware pitch detector", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-24, pp. 2-8, Feb. 1976.

[30] L. R. Rabiner, "On the use of autocorrelation analysis for pitch detection", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-25, pp. 24-33, Feb. 1977.

[31] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg and H. J. Manley, "Average Magnitude Difference Function Pitch Extractor", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-22, pp. 353-362, Oct. 1974.

[32] J. D. Markel, "The SIFT algorithm for fundamental frequency estimation", IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 367-377, Dec. 1972.

[33] A. M. Noll, "Cepstrum Pitch Determination", J. Acoust. Soc. Amer., vol. 41, pp. 293-309, Feb. 1967.

[34] D. J. Hermes, "Measurement of pitch by subharmonic summation", J. Acoust. Soc. Amer., vol. 83, pp. 257-264, Jan. 1988.

[35] P. Martin, "Comparison of pitch detection by cepstrum and spectral comb analysis", Proc. ICASSP 1982, pp.180-183.

[36] A. Gorin and R. J. Mammone, "Introduction to the special issue in neural networks for speech processing", IEEE Trans. Speech and Audio Processing, vol. 2, pp. 113-114, Jan. 1994.

[37] E. Barnard, R. A. Cole, M. P. Vea and F. A. Alleva, "Pitch detection with a neural-net classifier", IEEE Trans. Signal Processing, vol. 39, pp. 298-307, Feb. 1991.

[38] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of a two-way pitch detector", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1995.

[39] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Pitch-period reference file for the speech file - GSP.DAT", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1994.

[40] J. Makhoul, "Linear prediction : a tutorial review", IEEE Proc. Vol. 63, pp. 561-580, 1975.

[41] J. Makhoul, "Spectral linear prediction : properties and applications", IEEE Trans. Acoust, Speech and Signal Processing, Vol. ASSP-23, pp. 283-297, June 1975.

[42] L. R. Rabiner, B. S. Atal and M. R.Sambur, "LPC prediction error - analysis of its variation with the position of the analysis frame", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-25, pp. 434-442, Oct. 1977.

[43] K .K. Paliwal and P. V. S. Rao, "Windowing in linear prediction analysis of voiced speech", Instn. Electronics & Telecom. ENGRS. Vol. 27, pp. 165-171, 1981.

[44] T. P. Barnwell III, "Windowless techniques for LPC analysis", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-28, pp. 421-427, Aug. 1980.

[45] J. Makhoul, "Stable and efficient lattice methods for linear prediction", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-25, pp. 256-261, Oct. 1977.

[46] M. S. Song, R. M. Montalvo and A. M. Peterson, "IIR filter implementation in LSP structure", The 18th Asilomar conference on circuit. systems and computers (1985), pp. 428-433.

[47] D. D. Parikh, "Comparison of LPC vs. LSP synthesis", The 17th Asilomar conference on circuit. systems and computers (1984), pp. 216-221.

[48] B. M. G. Cheetham and P. M. Hughes, "Formant estimation from LSP coefficients", Proc. 5th Int. Conf. on Digital Signal Processing in Communications, Loughborough, UK, 1988, pp. 183-189.

[49] K. K. Paliwal, "Interpolation properties of linear prediction parametric representations", EUROSPEECH'95, pp. 1029-1032.

[50] H. B. Choi, W. T. K. Wong, B. M. G. Cheetham and C. C. Goodyear, "Interpolation of spectral information for low bit rate speech coding", EUROSPEECH'95, pp. 1033-1036.

[51] R. M. Gray, "Vector quantization", IEEE ASSP Magazine, pp. 4-29, April 1984.

[52] A. Gersho and V. Cuperman, "Vector quantization : A pattern-matching technique for speech coding", IEEE Communications Magazine, pp. 15-21, Dec. 1983.

[53] J. Makhoul, S. Roucos and H. Gish, "Vector quantization in speech coding", IEEE Proceedings, vol. 73, pp. 1551-1588, Nov. 1985.

[54] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design", IEEE Trans. Communications, Vol. COM-28, pp. 84-95, Jan. 1980.

[55] R. Viswanathan and J Makhoul, "Quantization properties of transmission parameters in linear predictive systems", IEEE Trans. Acoust, Speech and Signal Processing, Vol. ASSP-23, pp. 309-321, June 1975.

[56] F. K. Soong and B. H. Juang, "Optimal quantization of LSP parameters", IEEE Trans. Speech and Audio Processing, Vol. 1, pp. 15-24, Jan. 1993.

[57] K. K. Paliwal and B. S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame", IEEE Trans. Speech and Audio Processing, Vol. 1, pp. 3-13, Jan. 1993.

[58] R. P. Ramachandran, M. M. Sondhi, N. Seshadri and B. S. Atal, "A two codebook format for robust quantization of line spectral frequencies", IEEE Trans. Speech and Audio Processing, Vol. 3, pp. 157-168, May. 1995.

[59] F. K. Soong and B. H. Juang, "Line spectrum pair (LSP) and speech data compression", IEEE Proc. ICASSP 1984, pp. 1.10.1 - 1.10.4.

[60] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of the LBG clustering algorithm", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1995.

[61] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of an interframe multistage split vector quantiser", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1995.

[62] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-34, pp. 744-754, Aug. 1986.

[63] D. W. Griffin and J. S. Lim, "Multiband excitation vocoder", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP-36, pp.1223-1235, Aug. 1988.

[64] J. C. Hardwick and J. S. Lim, "A 4800bps improved multi-band excitation speech coder ", IEEE Proc. Workshop on Speech Coding for Telecommunications 1989, pp. 5-8.

[65] Y. Shoham, "Low-rate speech coding based on time-frequency interpolation", IEEE Proc. ICSLP 1992, pp. 20-40.

[66] Y. Shoham, "High-quality speech coding at 2.4 to 4.0 kbps based on time-frequency interpolation", IEEE Proc. Workshop on Speech Coding for Telecommunications 1993, pp.167-170.

[67] A. McCree, K. Truong, E. B. George, T. P. Barnwell and V. Viswanathan, "A 2.4kbit/s MELP coder candidate for the new U. S. Federal standard", IEEE Proc. ICASSP 1996, pp. 200-203.

[68] W. B. Kleijn and W. Gransow, "Methods for waveform interpolation in speech coding", Digital signal processing , vol. 1, pp. 215-230, 1991.

[69] W. B. Kleijn, "Continuous representations in linear predictive coding", IEEE Proc. ICASSP 1991, pp. 201-204.

[70] W. B. Kleijn, "Encoding speech using prototype waveforms", IEEE Trans. Speech and Audio Processing, Vol. 1, pp. 386-399, Oct. 1993.

[71] I. S. Burnett and R. J. Holbeche, "A mixed prototype waveform / CELP coder for sub 3kb/s", IEEE Proc. ICASSP 1993, pp. 175-178.

[72] G. Kubin, B. S. Atal and W. B. Kleijn, "Performance of noise excitation for unvoiced speech", IEEE Proc. Workshop on Speech Coding for Telecommunications 1993, pp. 35-36.

[73] W. B. Kleijn and Jesper Haagen, "Transformation and decomposition of the speech signal for coding", IEEE Signal Processing Letter, Vol. 1, pp. 136-138, Sept. 1994.

[74] W. B. Kleijn and Jesper Haagen, "A general waveform-interpolation structure for speech coding", EUSIPCO-1994, pp. 1665-1668.

[75] W. B. Kleijn and Jesper Haagen, "A speech coder based on decomposition of characteristic waveforms", IEEE Proc. ICASSP 1995, pp. 508-511.

[76] W. B. Kleijn, Y. Shoham, D. Sen & R. Hagen, "A low-complexity waveform interpolation coder", IEEE Proc. ICASSP 1996, pp. 212-215.

[77] A. E. Rosenberg, "Effect of glottal pulse shape on the quality of natural vowels", J. Acoust. Soc. Amer., vol. 49, pp. 583-590, April 1970.

[78] P. Alku, "Low bit rate speech coding with glottal linear prediction", Proc. ICASSP'90 pp. 2149 - 2152.

[79] P. Alku and U. K. Laine, "A new glottal LPC method for voice coding and inverse filtering ", Proc. ISCAS'89 pp. 1831 - 1834.

[80] X. Q. Sun and B. M. G. Cheetham, "Speech excitation modelling for low-bit rate speech coding", To be published.

[81] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of a 2.4kb/s PWI coder", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1995.

[82] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of a PWI/CELP coder", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1994.

[83] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of a 2.4kb/s TPSWI coder", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1996.

[84] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of a 2.4kb/s GPSWI coder", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1996.

[85] B. S. Atal, "Predictive coding of speech at low bit rates", IEEE Trans. Communications, Vol. COM-30, pp. 600-614, April 1982.

[86] M. R. Schroeder and B. S. Atal, "Coded-excited linear prediction (CELP) : high-quality speech at very low bit rates", IEEE Proc. ICASSP 1985, pp. 937-940.

[87] R. Salami, C. Laflamme, J. P. Adoul and D. Massaloux, "A toll quality 8kb/s speech codec for the personal communications system (PCS)", IEEE Trans. Vehicular Technology, Vol. 43, pp. 808-816, Aug. 1994.

[88] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of a CELP coder", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1994.

[89] P. Kroon and B. S. Atal, "Quantization procedures for the excitation in CELP coders", IEEE Proc. ICASSP 1987, pp.1649-1652.

[90] B. T. Laboratory, "Noise file - CAR.DAT".

[91] B. T. Laboratory, " Noise file - BABBLE.DAT".

[92] B. T. Laboratory, "Noise file - MULTI.DAT".

[93] "White Gaussian noise file - WHITE.DAT".

[94] X. Q. Sun, "Sinusoidal coding of speech at very low bit rates", Ph.D. dissertation, Dept., of Electrical Engineering and Electronics, The University of Liverpool, 1996.

[95] J. H. Chen and A. Gersho, "Adaptive postfiltering for quality enhancement of coded speech" IEEE Trans. Speech and Audio Processing, Vol. 3, pp. 59-71, Jan. 1995.

[96] H. B. Choi, B. M. G. Cheetham and C. C. Goodyear, "Software implementation of a 2.3kb/s TPSWI coder", Dept. of Electrical Engineering and Electronics, The University of Liverpool, 1996.

[97] I. Boyd, "Speech coding for telecommunications" in Westall F A and Ip S F A (Eds), "Digital signal processing in telecommunications", Chapman & Hall, pp. 300-325, 1993.

Appendix A Coded-Excited Linear Prediction (CELP) Coding

A.1 The CELP algorithm

Coded-Excited Linear Prediction (CELP) coding is a form of Adaptive Predictive Coding (APC) [85], where a reconstructed speech sample is generated by a scaled sum of past reconstructed speech samples and a quantised innovation sequence. In figure A.1 the system block diagram of a CELP coder is shown.

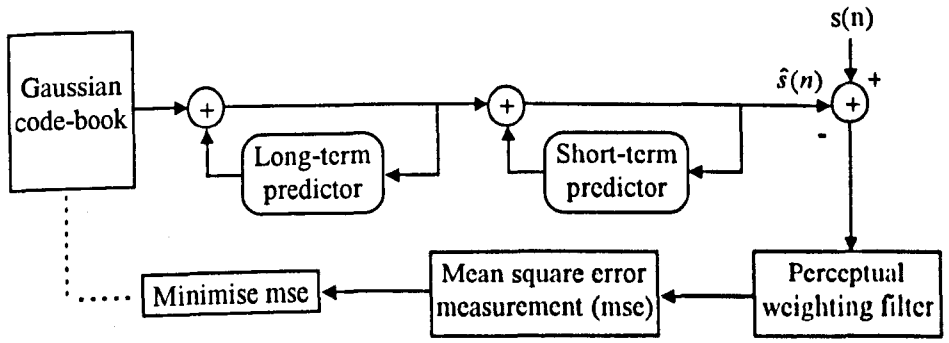


Figure A.1 The structure of a CELP coder

A number of innovation signals, which model speech excitation, are stored in a Gaussian code-book. During encoding, each innovation signal in the Gaussian code-book is processed by a long-term predictor (LTP) and a short term predictor (STP). The function of the long-term predictor is to introduce voiced periodicity into an innovation signal. At the short-term predictor, a short-term spectral envelope is imposed onto the innovation signal. The resulting signal from the short-term predictor would be an estimation of the original speech segment. This estimated signal is compared with the original speech segment to yield an error signal. The error signal is perceptually weighted and its mean square value is computed. The index of the innovation signal which yields a minimum error measure is transmitted. At the decoder, a copy of the identical Gaussian code-book, long-term predictor and short-term predictor is available. The decoder uses the code-book index to fetch the optimal innovation signal. The innovation signal is then processed by the two predictors to reconstruct the speech segment.

A.1.1 Long-term prediction

When a voiced speech is generated, the excitation signal is driven by the vocal cords. This results in a quasi-periodic signal in which the adjacent pitch-cycles are highly correlated. The function of the long-term predictor is to model this long-term correlation in an excitation signal. The long-term predictor has a transfer function as,

$$\frac{1}{P(z)} = \frac{1}{1 - \sum_{i=1}^{Q} b_i \, z^{-(M+i)}} \tag{A.1}$$

where

b_i is the predictor coefficient

Q is the number of taps used in the long-term predictor

M is the value of delays which is related to the pitch-period of voiced speech

Two configurations of LTP have been reported [89], single-tap LTP's and three-tap LTP's, depending on the number of delay taps Q used. Single-tap LTP's have been widely used in CELP coders owing to the computational simplicity and lower bit-rate requirement. A single-tap LTP is described by two parameters: predictor coefficient b and predictor delay M.

A.1.2 Short-term predictor

The short-term predictor aims to impose the short-term spectral envelope onto an excitation signal and thus the output from the short-term predictor is an estimation of the original speech segment. The short-term predictor is implemented by an all-pole vocal tract filter H(z) as,

$$H(z) = \frac{1}{1 - \sum_{i=1}^{P} a_i z^{-i}} \tag{A.2}$$

where a_i's are the LP ladder filter coefficients and P is the filter order.

A.1.3 Perceptual weighting filter

After an innovation signal is processed by the long-term and the short-term predictor. The resulting signal is compared with the original speech segment to yield an error signal. The error signal is processed through a perceptual weighting filter.

The purpose of the perceptual weighting filter is to attenuate the frequency components in the error signal which are perceptually less important and to amplify those frequency components which are perceptually more important. The transfer function of a perceptual weight filter is defined as,

$$w(z) = \frac{1 - \sum_{i=1}^{P} a_i \alpha_1^i z^{-i}}{1 - \sum_{i=1}^{P} a_i \alpha_2^i z^{-i}} \tag{A.3}$$

The parameters of the perceptual weighting filter α_1 and α_2 may be fixed to constant as α_1=0.9 and α_2=0.6 [87]. Alternatively, it may be adjusted according to the sampling frequency f_s as α_1=1.0 and $\alpha_2 = e^{-\frac{200\pi}{f_s}}$ [86].

A.2 A CELP coder implemented in the project

A.2.1 The CELP encoder

In figure A.2, the schematic diagram of a CELP encoder implemented in this project [88] is shown. The encoder consists of an adaptive code-book, a Gaussian code-book and a copy of the decoder (the local decoder). The adaptive code-book aims to model the long-term correlation of a speech excitation and is consistently updated by the reconstructed excitation signal from the local decoder. The Gaussian code-book is used to model the difference between the contribution from the optimum adaptive code-book candidate and the original speech segment. The Gaussian code-book is populated with white-Gaussian noise sequences. The two code-books are searched separately using an analysis-by-synthesis approach. A mean square error measure is used to compare the reconstructed speech segment from each code-book candidate with the original. The code-book candidate yields the minimum error measure is chosen and its optimal gain is computed. The indices of the two code-books and their associated gains are quantised. The details of the CELP encoder will be described in the follows.

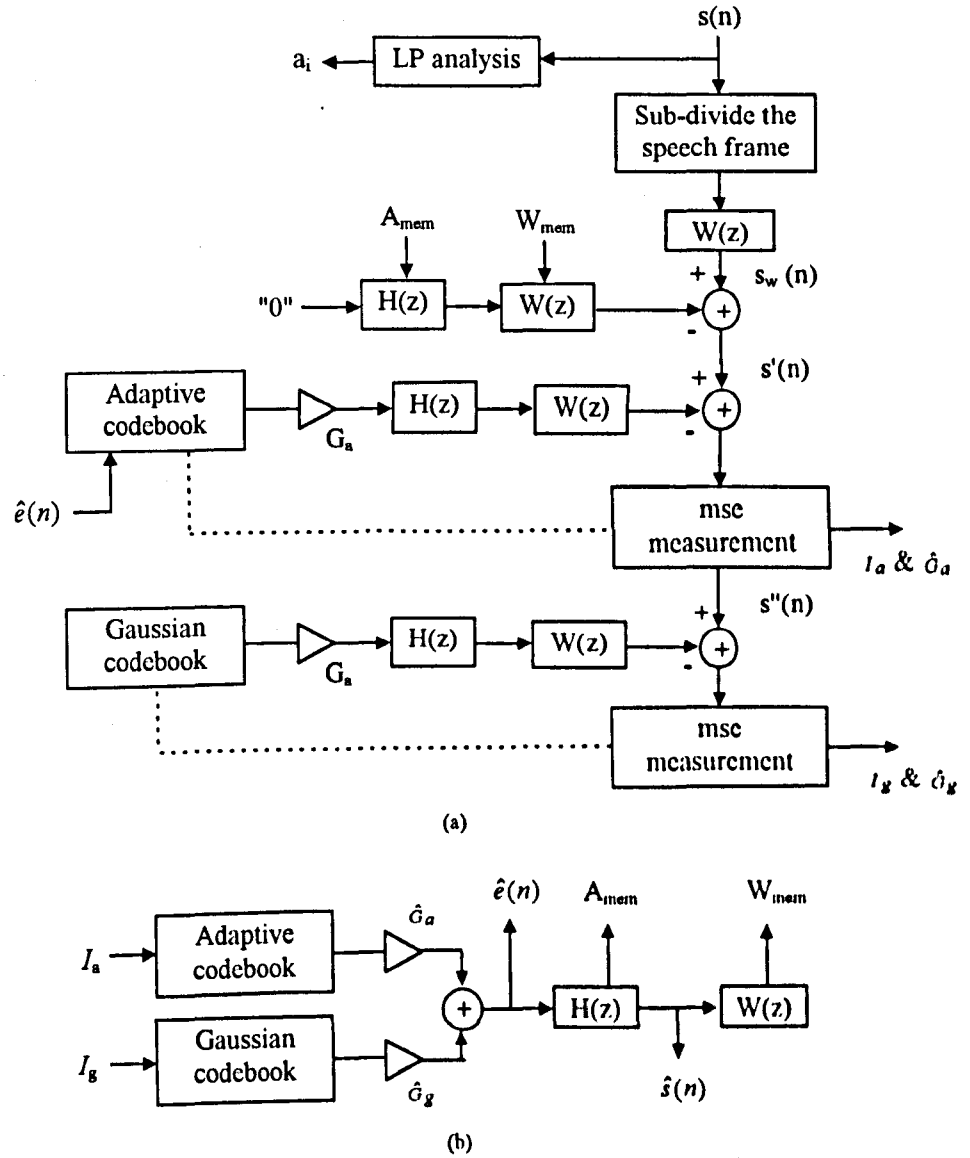


Figure A.2 The schematic diagram of a CELP encoder.
(a) the encoder (b) the local decoder

A.2.2 The zero impulse response filter

During encoding, the input speech is segmented into speech frames with 160 samples. LP analysis is applied to an input speech frame to yield a set of LP ladder filter coefficients. The frame of speech segment is then sub-divided into four speech sub-frames, each contains of 40 speech samples. To search the optimal candidates from the two code-books for a speech sub-frame, the speech sub-frame is first processed by a perceptual weighting filter with the filter parameters set as $\alpha_1 = 0.9$ and $\alpha_2 = 0.6$ [87]. The resulting signal $s_w(n)$ is used as the reference to search the two code-books.

In searching a code-book, each code-book candidate is called up and processed by an LP synthesis filter and a perceptual weighting filter. The memory content in the two filters must be reset to the memory content left over from the last speech sub-frame (the overhang memories A_{mem} and W_{mem}), for each code-book innovation signal. This increases the computational costs of the coder.

The effect of the overhang memory can be modelled by a zero impulse response (ZIR) filter. The ZIR filter consists an identical copy of the LP synthesis and perceptual weighting filter. Prior to the code-book searching for a new speech sub-frame, the memory contents in the ZIR filter are set to the overhang memories. The ZIR is then excited by a zero input. To eliminate the effect of the overhang memories, the output of the ZIR filter is subtracted from the perceptually weighted input speech sub-frames $s_w(n)$, to yield a modified speech sub-frame s'(n).

The overhang memories are provided by the local decoder. At the encoder after the required quantised parameters are available, the local decoder utilises the current quantised parameters to reconstruct a version of the quantised excitation signal. The reconstructed excitation signal is then passed through an identical copy of the LP synthesis filter and perceptual weighting filter. The memory contents remain in the two filters are the overhang memories required for the next speech sub-frame.

A.2.3 Adaptive code-book searching

In the CELP coder, the long-term predictor is implemented by an adaptive code-book. The adaptive code-book consists of a delayline which is constantly updated by the reconstructed excitation signal at the local decoder every 5ms (i.e. every 40 samples). The updating scheme is shown in figure A.3.

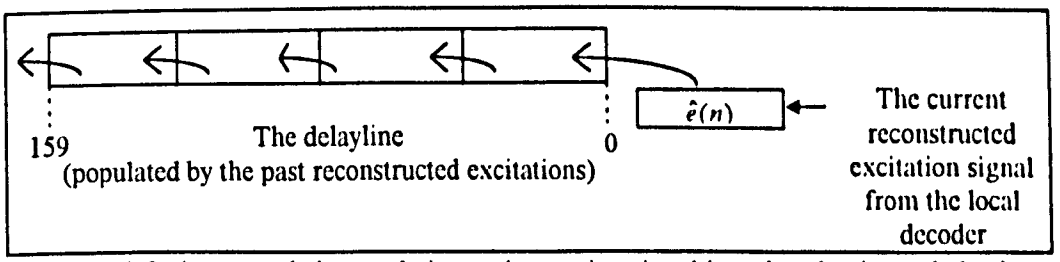


Figure A.3 An example in populating an innovation signal into the adaptive code-book

In searching the adaptive code-book, the value of the delay tap is used as an indicator to extract an innovation signal from the delayline. This is demonstrated in figure A.4. Suppose the current delay value is 60 samples, the adaptive code-book used this as the starting point and extracted 40 consecutive samples towards the beginning of the delayline.

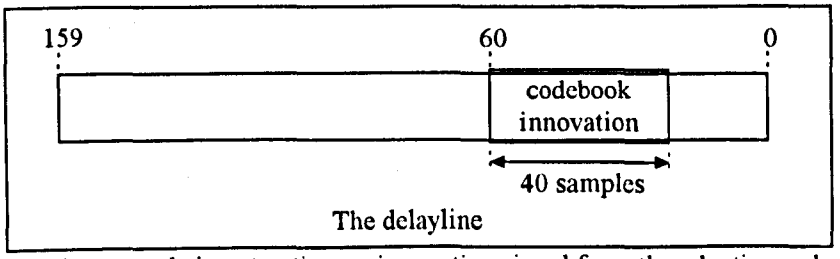


Figure A.4 An example in extracting an innovation signal from the adaptive code-book

A single tap LTP is used in the CELP coder and thus two predictor parameters, the delay value M and the predictor coefficient b , are required. The range of delay values M chosen depends on the pitch-period of voiced speech under consideration. A range M which takes values from 20 to 147 speech samples (8kHz sampling frequency) has been chosen.

To search the best matched code-book candidate, each innovation signal is processed by the LP synthesis filter and perceptual weighting filter. The resulting signal is compared with the modified speech sub-frame s'(n) using a mean square error measure which is defined as,

$$E = \frac{1}{40} \sum_{n=0}^{39} \left(s'(n) - G\hat{s}(n) \right)^2 \tag{A.4}$$

where

E is the total error

G is the gain factor

$\hat{s}(n)$ is the synthesised speech sub-frame

To compute the optimal gain, the mean square error E is differentiated with respect to the gain factor G as,

$$\frac{\partial E}{\partial G} = \frac{2}{40} \left(\sum_{n=0}^{39} G\hat{s}^2(n) - \sum_{n=0}^{39} \hat{s}(n)s'(n) \right) \tag{A.5}$$

By setting $\frac{\partial E}{\partial G} = 0$, the optimal gain factor is,

$$G = \frac{\sum_{n=0}^{39} \hat{s}(n)s'(n)}{\sum_{n=0}^{39} \hat{s}^2(n)} \tag{A.6}$$

By substituting equation A6 into A4, the mean square error is computed as,

$$E = \frac{1}{40}\left(1 - \frac{\left(\sum_{n=0}^{39} \hat{s}(n)s'(n)\right)^2}{\sum_{n=0}^{39}\hat{s}^2(n)\sum_{n=0}^{39}s'^2(n)}\right)\sum_{n=0}^{39}s'^2(n) \tag{A.7}$$

Equation A7 suggested that the mean square error is minimum when the normalised cross-correlation function between the two signals is maximum. The normalised cross-correlation function between the two signals s'(n) and $\hat{s}(n)$ is defined as,

$$C = \frac{\sum_{n=0}^{39} \hat{s}(n)s'(n)}{\left(\sum_{n=0}^{39}\hat{s}^2(n)\sum_{n=0}^{39}s'^2(n)\right)^{1/2}} \tag{A.8}$$

As a result, the optimum adaptive code-book candidate is searched by choosing the delay value, across the range of possible delay values, which yields the maximum cross-correlation values. The optimal gain is computed from the optimum adaptive code-book candidate using equation A6.

A problem arises when the delay value M is smaller than 39 samples, where the number of samples at the beginning of the delayline is no enough to fill up the entire innovation sequence, i.e. less than 40 samples. In this case the beginning of the delayline is extended by repeating the current updated excitation signal in a way shown in figure A5.

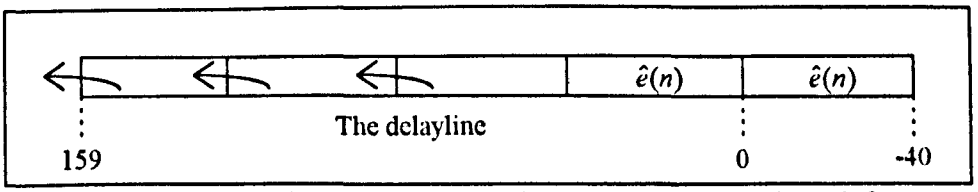


Figure A.5 The delayline is extended by repeating the updated excitation sub-frame

A.2.4 Gaussian code-book searching

After the optimum adaptive candidate has been found, the contribution from the adaptive code-book is subtracted from the modified speech sub-frame s'(n), to yield the reference signal s''(n) to search the Gaussian code-book. The Gaussian code-book is populated with white-Gaussian noise sequences, each has a fixed length of 40 samples. An 8-bits Gaussian code-book is used. The Gaussian code-book is searched in the same way as it is done for the adaptive code-book. The index of the optimum code-book candidate is sent to the decoder together with its optimal gain factor.

A.2.5 The CELP decoder

After code-book searching, the code-book indices I_a and I_g are sent to the decoder. The two code-book gains are vector quantised using an 8-bit code-book [18]. The LP ladder filter coefficients are converted to the LSF coefficients. The LSF coefficients are quantised using a 24-bit split vector quantiser [57].

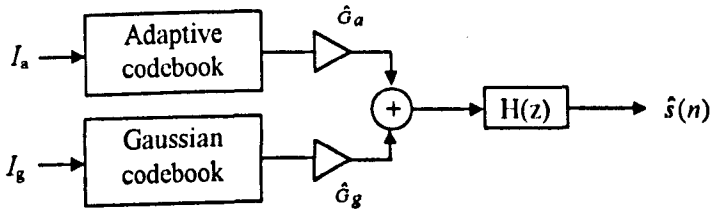


Figure A.6 A CELP decoder

The schematic diagram of the CELP decoder is shown in figure A6. The reconstructed excitation signal is obtained by summing the optimum innovation signals from the adaptive code-book and the Gaussian code-book, each scaled by the quantised gain $\hat{G}_a$ and $\hat{G}_g$ respectively. Finally the decoded speech is obtained by processing the reconstructed excitation signal through an LP synthesis filter.

Appendix B Conversion between LP ladder filter coefficients and LSF's

B.1 Line Spectral Frequencies

Suppose we have an all-zero filter polynomial A(z) which is obtained by applying linear prediction analysis to a segment of speech samples. The all-zero filter polynomial A(z) has a transfer function defined as,

$$A(z) = \sum_{i=0}^{P} a_i z^{-i}$$
$$= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_P z^{-P} \quad \text{(B.1)}$$

where P is the filter order and a_i is the ladder filter coefficients

The Pth order all-zero filter polynomial A(z) can be decomposed into two LSF polynomials P(z) and Q(z), using the relationship defined in equations B2a and b,

$$P(z) = A(z) + z^{-(P+1)} A\left(z^{-1}\right) \quad \text{(B.2a)}$$

$$Q(z) = A(z) - z^{-(P+1)} A\left(z^{-1}\right) \quad \text{(B.2b)}$$

P(z) is a (P+1)th order symmetric polynomial and Q(z) is a (P+1)th order anti-symmetric polynomial. The zeros of P(z) and Q(z) are the required LSF's and they have the properties that the all-pole filter polynomial H(z), which is equal to 1/A(z), is stable if and only if,

a) All the roots of P(z) and Q(z) lie on the unit circle,

b) The roots of P(z) and Q(z) are interlaced, i.e. if φ_i and θ_i are the roots for P(z) and Q(z) respectively for i = 1, 2, ..., P/2, then,

$$0 \le \varphi_1 < \theta_1 < \cdots < \varphi_{P/2} < \theta_{P/2} \le \pi$$

Furthermore, the original all-zero filter polynomial A(z) is recovered from the two LSF polynomials P(z) and Q(z) using equation B3,

$$A(z) = \frac{P(z) + Q(z)}{2} \quad \text{(B.3)}$$

B.2 From LP ladder filter coefficients to LSF's

To simplify the analysis, only even filter order is considered, i.e. P is always even. P(z) has a real root at $z^{-1} = -1$ and Q(z) has a real root at $z^{-1} = 1$. The rest of the roots for P(z) and Q(z) are in complex conjugate pairs.

To compute the roots for P(z), equation B.2a is expanded as,

$$\begin{aligned} P(z) &= 1 + (a_1 + a_P)z^{-1} + (a_2 + a_{P-1})z^{-2} + \cdots + (a_{P-1} + a_2)z^{-(P-1)} + (a_P + a_1)z^{-P} + z^{-(P+1)} \\ &= 1 + p_1 z^{-1} + p_2 z^{-2} + \cdots + p_{P-1} z^{-(P-1)} + p_P z^{-P} + z^{-(P+1)} \end{aligned} \quad \text{(B.4)}$$

where

p_i is the coefficient of the polynomial P(z)

$$p_1 = a_1 + a_P = p_P$$
$$p_2 = a_2 + a_{P-1} = p_{P-1}$$
$$\vdots$$
$$p_{P/2} = a_{P/2} + a_{P/2+1} = p_{P/2+1}$$

Since P(z) has a real root at $z^{-1} = -1$, equation B.4 becomes,

$$P(z) = (1 + z^{-1})\, P'(z) \quad \text{(B.5)}$$

with

$$P'(z) = 1 + p'_1 z^{-1} + p'_2 z^{-2} + \cdots + p'_{P-1} z^{-(P-1)} + p'_P z^{-P} \quad \text{(B.6)}$$

where

$$p'_1 = p_1 - 1 = p'_{P-1}$$
$$p'_2 = p_2 - p_1 + 1 = p_2 - p'_1 = p'_{P-2}$$
$$\vdots$$
$$p'_P = p_P - p'_{P-1} = 1$$

Equation B.6 can be simplified as,

$$\begin{aligned} P'(z) &= 1 + p'_1 z^{-1} + p'_2 z^{-2} + \cdots + p'_{P-1} z^{-(P-1)} + p'_P z^{-P} \\ &= 1 + p'_1 z^{-1} + p'_2 z^{-2} + \cdots + p'_2 z^{-(P-2)} + p'_1 z^{-(P-1)} + z^{-P} \\ &= z^{-P/2}\left(\left(z^{P/2} + z^{-P/2}\right) + p'_1\left(z^{(P/2-1)} + z^{-(P/2-1)}\right) + p'_2\left(z^{(P/2-2)} + z^{-(P/2-2)}\right) + \cdots + p'_{P/2}\right) \end{aligned} \quad \text{(B.7)}$$

By substituting $z = e^{j\vartheta}$ into equation B.7, we have,

$$P'(e^{j\vartheta}) = 2\,e^{-j\frac{P\vartheta}{2}}\left(\cos\left(\frac{P}{2}\right)\vartheta + p_1'\cos\left(\frac{P}{2}-1\right)\vartheta + p_2'\cos\left(\frac{P}{2}-2\right)\vartheta + \cdots + 0.5p_{\frac{P}{2}}'\right) \quad \text{(B.8)}$$

Hence the locations on the unit circle of the roots of P(z) is solved by setting the right hand side of equation B.8 to zero, i.e.

$$\cos\left(\frac{P}{2}\right)\vartheta + p_1'\cos\left(\frac{P}{2}-1\right)\vartheta + p_2'\cos\left(\frac{P}{2}-2\right)\vartheta + \cdots + 0.5p_{\frac{P}{2}}' = 0 \quad \text{(B.9)}$$

Similarly to solve the roots for Q(z), equation B.2b is expanded as,

$$\begin{aligned} Q(z) &= 1 + (a_1 - a_P)z^{-1} + (a_2 - a_{P-1})z^{-2} + \cdots + (a_{P-1} - a_2)z^{-(P-1)} + (a_P - a_1)z^{-P} - z^{-(P+1)} \\ &= 1 + q_1 z^{-1} + q_2 z^{-2} + \cdots + q_{P-1}z^{-(P-1)} + q_P z^{-P} - z^{-(P+1)} \end{aligned} \quad \text{(B.10)}$$

where

q_i is the coefficient of the polynomial Q(z)

$q_1 = -q_P$, $q_2 = -q_{P-1}$ and so on.

Q(z) has a real root at $z^{-1} = 1$, and equation B.10 becomes,

$$Q(z) = (1 - z^{-1})\,Q'(z) \quad \text{(B.11)}$$

with

$$Q'(z) = 1 + q_1'z^{-1} + q_2'z^{-2} + \cdots + q_{P-1}'z^{-(P-1)} + q_P'z^{-P} \quad \text{(B.12)}$$

where

$$\begin{aligned} q_1' &= q_1 + 1 = q_{P-1}' \\ q_2' &= q_2 + q_1 + 1 = q_2 + q_1' = q_{P-2}' \\ &\vdots \\ q_P' &= q_P + q_{P-1}' = 1 \end{aligned}$$

Equation B.12 can be simplified as,

$$\begin{aligned} Q(z) &= 1 + q_1'z^{-1} + q_2'z^{-2} + \cdots + q_2'z^{-(P-2)} + q_1'z^{-(P-1)} + z^{-P} \\ &= z^{-P/2}\left(\left(z^{P/2} + z^{-P/2}\right) + q_1'\left(z^{(P/2-1)} + z^{-(P/2-1)}\right) + q_2'\left(z^{(P/2-2)} + z^{-(P/2-2)}\right)\cdots + q_{P/2}'\right) \end{aligned} \quad \text{(B.13)}$$

By substituting $z = e^{j\vartheta}$ into equation B.13, we have,

$$Q\left(e^{j\vartheta}\right) = 2e^{-j\frac{P\vartheta}{2}}\left(\cos\left(\frac{P}{2}\right)\vartheta + q_1'\cos\left(\frac{P}{2}-1\right)\vartheta + q_2'\cos\left(\frac{P}{2}-2\right)\vartheta + \cdots + 0.5q_{P/2}'\right) \quad \text{(B.14)}$$

Hence the locations on the unit circle of the roots of Q(z) is solved by setting the right hand side of equation B.14 to zero. Thus,

$$\cos\left(\frac{P}{2}\right)\vartheta + q_1'\cos\left(\frac{P}{2}-1\right)\vartheta + q_2'\cos\left(\frac{P}{2}-2\right)\vartheta + \cdots + 0.5q_{P/2}' = 0 \quad \text{(B.15)}$$

The roots of equations B.9 and B.15 can be computed through an iterative method [59]. If a sign change is detected between f(ω) and f(ω+Δω), an odd number of root(s) will exist between ω and Δω. By using a sufficiently small grid (Δω), a single root can be gauranteed in the region and the root can be computed,

$$x = \omega + \frac{(\Delta\omega)f(\omega)}{f(\omega+\Delta\omega) - f(\omega)} \quad \text{(B.16)}$$

The disadvantage of using a smaller grid is the increase in computational complexity. Hence a trade-off must be made between the accuracy of LSF's and computational complexity.

B.3 From LSF's to LP ladder filter coefficients

To retrieve the a_i coefficients from the LSF's. The LSF polynomial P(z) is arranged in form of,

$$P(z) = \left(1 + z^{-1}\right)\prod_{i=1}^{P}\left(z^{-1} - e^{j\varphi_i}\right) \quad \text{(B.17)}$$

Since the roots of P(z) are in complex conjugate pairs, we arrange each conjugate pair together as,

$$\begin{aligned}\left(z^{-1} - e^{j\varphi_i}\right)\left(z^{-1} - e^{j\varphi_{P-i}}\right) &= \left(z^{-1} - e^{j\varphi_i}\right)\left(z^{-1} - e^{-j\varphi_i}\right) \\ &= 1 - 2z^{-1}\cos\varphi_i + z^{-2}\end{aligned} \quad \text{(B.18)}$$

By substituting equation B.18 into B.17, equation B.17 becomes,

$$P(z) = \left(1 + z^{-1}\right)\prod_{i=1}^{P/2}\left(1 - 2z^{-1}\cos\varphi_i + z^{-2}\right) \quad \text{(B.19)}$$

By expanding equation B.19, the coefficients p_i for the LSF polynomial P(z) are obtained.

Similarly the coefficients q_i for the LSF polynomial Q(z) is calculated by expanding equation B.20,

$$Q(z) = \left(1 - z^{-1}\right) \prod_{i=1}^{P} \left(z^{-1} - e^{j\theta_i}\right) \tag{B.20}$$

By grouping the complex conjugate pairs together, equation B.20 becomes,

$$Q(z) = \left(1 - z^{-1}\right) \prod_{i=1}^{P/2} \left(1 - 2z^{-1}\cos\theta_i + z^{-2}\right) \tag{B.21}$$

To compute the a_i coefficients, the coefficients of the LSF polynomials P(z) and Q(z) are summing together as,

$$a_i = \frac{1}{2}\left(p_i + q_i\right) \qquad i = 0, 1, 2, \ldots, P \tag{B.22}$$

Appendix C Manipulation of pitch-cycles in the frequency-domain

C.1 Spectral representations for a pitch-cycle

The discrete Fourier transform coefficients of a pitch-cycle u(n) are given by,

$$U_k = \sum_{n=0}^{p-1} u(n) e^{-\frac{j2\pi kn}{p}} \tag{C.1a}$$

with inverse transform,

$$u(n) = \frac{1}{p} \sum_{k=0}^{p-1} U_k e^{\frac{j2\pi kn}{p}} \tag{C.1b}$$

where p is the pitch-period.

If we separate the U_k into real and imaginary parts, thus,

$$U_k = R_k - jI_k \tag{C.2}$$

then equation C.1b becomes the expression in sine and cosine terms as follows,

$$u(n) = \frac{1}{p} \sum_{k=0}^{p-1} \left(R_k \cos\left(\frac{2\pi kn}{p}\right) + I_k \sin\left(\frac{2\pi kn}{p}\right)\right) \tag{C.3}$$

where

$$R_k = \sum_{n=0}^{p-1} u(n) \cos\left(\frac{2\pi kn}{p}\right) \tag{C.4a}$$

$$I_k = \sum_{n=0}^{p-1} u(n) \sin\left(\frac{2\pi kn}{p}\right) \tag{C.4b}$$

We may also write U_k in term of magnitude and phase, then,

$$U_k = |U_k| e^{j\phi_{U_k}} \tag{C.5}$$

C.2 Phase alignment of pitch-cycles in the frequency-domain

Phase alignment of pitch-cycles is carried out using a cross-correlation function. Suppose we have two pitch-cycles x(n) and y(n), each with the same pitch-period p, i.e. $\{x(n)\}_{n=0,\,p-1}$ and $\{y(n)\}_{n=0,\,p-1}$. In the time-domain, assume the two pitch-cycles are periodically extended for $-\infty<n<\infty$, the unnormalised cross-correlation function between the two pitch-cycles x(n) and y(n) for a range of sample shifts m, where m = 0 to $p-1$, is defined as,

$$C(m) = \sum_{n=0}^{p-1} x(n)y(n+m) \tag{C.6}$$

The number of sample shift m' which yields the maximum cross-correlation value is found. The phase aligned pitch-cycle, say $\tilde{y}(n)$, is obtained by circularly shifting y(n) by m' samples.

In the frequency-domain, equation C.6 is re-called. Using the IDFT expression in equation C.1b for the pitch-cycles x(n) and y(n+m), equation C.6 becomes,

$$C(m) = \sum_{n=0}^{p-1} \frac{1}{p}\sum_{l=0}^{p-1} U_{xl}e^{j\left(\frac{2\pi ln}{p}\right)} \frac{1}{p}\sum_{k=0}^{p-1} U_{yk}e^{j\left(\frac{2\pi k(n+m)}{p}\right)}$$

$$= \frac{1}{p^2}\sum_{n=0}^{p-1}\sum_{l=0}^{p-1}\sum_{k=0}^{p-1} U_{xl}U_{yk}e^{j\left(\frac{2\pi(k+l)n}{p}+\frac{2\pi km}{p}\right)} \tag{C.7}$$

Summing first over n, only terms for $k = -l$ remain. Hence equation C.7 becomes,

$$C(m) = \sum_{k=0}^{p-1} U_{xk}U_{y-k}\,e^{-j\frac{2\pi km}{p}} \tag{C.8}$$

Using the magnitude and phase expressions in equation C.5, equation C.8 becomes,

$$C(\xi) = \sum_{k=0}^{p-1} |U_{xk}||U_{yk}|e^{j(\phi_{xk}-\phi_{yk}-k\xi)} \tag{C.9}$$

where ξ is phase shift, with $\xi = \frac{2\pi m}{p}$.

Equation C.9 shows that phase alignment in the frequency-domain is realised by injecting a linear phase components kξ into the phase spectrum of pitch-cycle y(n). This corresponds to circularly shifting the pitch-cycle y(n) in the time-domain. The range of phase shift ξ is from 0 to 2π, in a finite number of steps. The greater the number of steps used, the better the two pitch-cycle may be aligned. However, this also increases the computational cost of the system.

Assume the amount of phase shift that achieving the maximum cross-correlation value is ξ', where,

$$\xi' = \max_{\xi} \sum_{k=0}^{p-1} |U_{xk}||U_{yk}| \cos\left(\phi_{xk} - \phi_{yk} - k\xi\right) \tag{C.10}$$

Note that: the imaginary part is an odd function and thus only the real part is valid.

To obtain the aligned pitch-cycle $\tilde{y}(n)$, only the phase spectrum of y(n) need to be modified and the new phase spectrum is defined as,

$$\tilde{\phi}_{yk} = \phi_{yk} + k\xi' \tag{C.11}$$

where $\tilde{\phi}_{yk}$ is the phase spectrum of the aligned pitch-cycle

We may choose to work with real and imaginary parts of the U_k. By substituting equation C.2 into C.8, the cross-correlation function is expressed as,

$$C(m) = \sum_{k=0}^{p-1} \left\{ \left(R_{xk} - jI_{xk}\right)\left(R_{yk} + jI_{yk}\right)\left(\cos\left(\frac{2\pi km}{p}\right) - j\sin\left(\frac{2\pi km}{p}\right)\right)\right\} \tag{C.12}$$

Thus the cross-correlation function for the phase shift ξ is defined as,

$$C(\xi) = \sum_{k=0}^{p-1} \left\{ \left(R_{xk}R_{yk} + I_{xk}I_{yk}\right)\cos k\xi + \left(R_{xk}I_{yk} - I_{xk}R_{yk}\right)\sin k\xi \right\} \tag{C.13}$$

$$0 \le \xi < 2\pi$$

Assume the phase shift which yields the maximum cross-correlation value is ξ', the real and imaginary parts of the phase aligned pitch-cycle become,

$$\tilde{R}_{yk} = R_{yk}\cos k\xi' + I_{yk}\sin k\xi' \tag{C.14a}$$

$$\tilde{I}_{yk} = I_{yk}\cos k\xi' - R_{yk}\sin k\xi' \tag{C.14b}$$

C.3 LP filtering of pitch-cycles in the frequency-domain

Suppose we have a Pth order all-zero filter A(z) which is obtained by applying LP analysis to a section of speech samples. The impulse response for the all-zero filter A(z) is defined from i = 0 to P as a_i. In the time-domain, the LP residual can be computed by a convolution process between the input speech signal and the filter impulse response a_i as,

$$r(n) = \sum_{i=0}^{P} a_i \, s(n-i) \tag{C.15}$$

Assume the input speech signal is a periodic signal which is stationary over time. Equation C.15 can be treated as a circular convolution operation between the filter impulse response and a speech-domain segment s(n), for n = 0 to p-1. Hence the residual-domain pitch-cycle u(n), for n = 0 to p-1, is defined,

$$u(n) = \sum_{i=0}^{P} a_i \, s(n-i) \qquad n = 0, 1, \ldots, p-1 \tag{C.16}$$

Using the IDFT expression in equation C.1b for s(n) and substituting this expression into equation C.16, the residual-domain pitch-cycle u(n) becomes,

$$\begin{aligned} u(n) &= \sum_{i=0}^{P} a_i \frac{1}{p} \sum_{k=0}^{p-1} S_k e^{j\left(\frac{2\pi k(n-i)}{p}\right)} \\ &= \frac{1}{p} \sum_{k=0}^{p-1} \sum_{i=0}^{P} a_i S_k e^{j\left(\frac{2\pi k(n-i)}{p}\right)} \\ &= \frac{1}{p} \sum_{k=0}^{p-1} \sum_{i=0}^{P} a_i \left|S_k\right| e^{j\left(\phi_{sk} - \frac{2\pi ki}{p}\right)} e^{j\left(\frac{2\pi kn}{p}\right)} \\ &= \frac{1}{p} \sum_{k=0}^{p-1} \left|U_k\right| e^{j\phi_{Uk}} e^{j\left(\frac{2\pi kn}{p}\right)} \\ &= \frac{1}{p} \sum_{k=0}^{p-1} U_k e^{j\left(\frac{2\pi kn}{p}\right)} \end{aligned} \tag{C.17}$$

where

$|U_k|$ and ϕ_{Uk} are the magnitude and phase of the pitch-cycle in residual-domain

$|S_k|$ and ϕ_{Sk} are the magnitude and phase of the pitch-cycle in speech-domain

From equation C.17, the magnitude and phase of a residual-domain pitch-cycle can be computed from the magnitude and phase of the corresponding pitch-cycle in the speech-domain using the relationship that,

$$\frac{1}{p}\sum_{k=0}^{p-1}\sum_{i=0}^{P} a_i \left|S_k\right| e^{j\left(\phi_{Sk}-\frac{2\pi ki}{p}\right)} e^{j\left(\frac{2\pi kn}{p}\right)} = \frac{1}{p}\sum_{k=0}^{p-1} \left|U_k\right| e^{j\phi_{Uk}} e^{j\left(\frac{2\pi kn}{p}\right)}$$

Hence we have,

$$\left|U_k\right| = \left|\sum_{i=0}^{P} a_i \left|S_k\right| e^{j\left(\phi_{Sk}-\frac{2\pi ki}{p}\right)}\right| \tag{C.18a}$$

$$\phi_{Uk} = Arg\left(\sum_{i=0}^{P} a_i \left|S_k\right| e^{j\left(\phi_{Sk}-\frac{2\pi ki}{p}\right)}\right) \tag{C.18b}$$

i.e.

$$\left|U_k\right| = \left|S_k\right| \left\{\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2\pi ki}{p}\right)\right)^2 + \left(\sum_{i=0}^{P} a_i \sin\left(\frac{2\pi ki}{p}\right)\right)^2\right\}^{1/2} \tag{C.19a}$$

$$\phi_{Uk} = \phi_{Sk} - \tan^{-1}\left\{\frac{\sum_{i=0}^{P} a_i \sin\left(\frac{2\pi ki}{p}\right)}{\sum_{i=0}^{P} a_i \cos\left(\frac{2\pi ki}{p}\right)}\right\} \tag{C.19b}$$

To recover the speech-domain pitch-cycle from the residual-domain pitch-cycle, equation C.17 is re-called. Using the relationship that,

$$\frac{1}{p}\sum_{k=0}^{p-1}\sum_{i=0}^{P} a_i S_k e^{j\left(\frac{2\pi k(n-i)}{p}\right)} = \frac{1}{p}\sum_{k=0}^{p-1} U_k e^{j\left(\frac{2\pi kn}{p}\right)}$$

we have,

$$U_k = \sum_{i=0}^{P} a_i S_k e^{-j\left(\frac{2ki\pi}{p}\right)} \tag{C.20}$$

then,

$$S_k = \frac{U_k}{\sum_{i=0}^{P} a_i e^{-j\left(\frac{2ki\pi}{p}\right)}} \tag{C.21}$$

From equation C.21, the magnitude of the speech-domain pitch-cycle is now,

$$|S_k| = \frac{|U_k|}{\left|\sum_{i=0}^{P} a_i e^{-j\left(\frac{2ki\pi}{p}\right)}\right|}$$

$$= \frac{|U_k|}{\left\{\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)\right)^2 + \left(\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)\right)^2\right\}^{1/2}} \qquad \text{(C.22a)}$$

Furthermore, the phase of the speech-domain pitch-cycle can be derived from equation C.21 ,

$$\phi_{Sk} = \frac{Arg(U_k)}{Arg\left(\sum_{i=0}^{P} a_i e^{-j\left(\frac{2ki\pi}{p}\right)}\right)}$$

$$= \phi_{Uk} - Arg\left(\sum_{i=0}^{P} a_i e^{-j\left(\frac{2ki\pi}{p}\right)}\right)$$

$$= \phi_{Uk} + \tan^{-1}\left\{\frac{\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)}{\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)}\right\} \qquad \text{(C.22b)}$$

In case of real and imaginary parts, equation C.16 is now become,

$$u(n) = \sum_{i=0}^{P} a_i \frac{1}{p} \sum_{k=0}^{p-1} \left(R_{Sk} - jI_{Sk}\right) e^{j\left(\frac{2\pi k(n-i)}{p}\right)}$$

$$= \frac{1}{p} \sum_{k=0}^{p-1} \sum_{i=0}^{P} a_i \left(R_{Sk} - jI_{Sk}\right) e^{-j\left(\frac{2\pi ki}{p}\right)} e^{j\left(\frac{2\pi kn}{p}\right)}$$

$$= \frac{1}{p} \sum_{k=0}^{p-1} \left(R_{Uk} - jI_{Uk}\right) e^{j\left(\frac{2\pi kn}{p}\right)} \qquad \text{(C.23)}$$

where

R_{Uk} and I_{Uk} are the real and imaginary parts of the residual-domain pitch-cycle

R_{Sk} and I_{Sk} are the real and imaginary parts of the speech-domain pitch-cycle

Hence the real and imaginary parts of a residual cycle can be computed from the real and imaginary parts of the corresponding speech-domain pitch-cycle as,

$$R_{Uk} = R_{Sk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right) - I_{Sk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right) \tag{C.24a}$$

$$I_{Uk} = R_{Sk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right) + I_{Sk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right) \tag{C.24b}$$

Conversely to compute the real and imaginary parts of a speech-domain pitch-cycle from the corresponding residual-domain pitch-cycle, we use the relationship $R_{Uk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right) + I_{Uk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)$ such that,

$$R_{Uk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right) + I_{Uk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right) = R_{Sk}\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)\right)^2 + R_{Sk}\left(\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)\right)^2 \tag{C.25}$$

Hence the real part of the speech-domain pitch-cycle R_{Sk} is,

$$R_{Sk} = \frac{R_{Uk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right) + I_{Uk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)}{\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)\right)^2 + \left(\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)\right)^2} \tag{C.26}$$

Similarly imaginary part of the speech-domain pitch-cycle I_{Sk} can be determined using the relationship $I_{Uk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right) - R_{Uk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)$,

$$I_{Uk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right) - R_{Uk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right) = I_{Sk}\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)\right)^2 + I_{Sk}\left(\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)\right)^2 \tag{C.27}$$

and the imaginary part I_{Sk} is,

$$I_{Sk} = -\frac{R_{Uk} \sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right) - I_{Uk} \sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)}{\left(\sum_{i=0}^{P} a_i \cos\left(\frac{2ki\pi}{p}\right)\right)^2 + \left(\sum_{i=0}^{P} a_i \sin\left(\frac{2ki\pi}{p}\right)\right)^2} \tag{C.28}$$