SINUSOIDAL CODING OF SPEECH AT VERY LOW BIT RATES.

Thesis submitted in accordance with the requirements
of the University of Liverpool for the degree of
Doctor of Philosophy

by

Xiaoqin Sun

Department of Electrical Engineering and Electronics
University of Liverpool.

November 1996.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Barry M. G. Cheetham for his constant support, valuable advice, great encouragement and most importantly his patience and effort throughout these years.

My thanks are particularly expressed to my industrial supervisor, Dr. Kenneth W. T. Wong, who has given me expert advice, invaluable discussions, great encouragement and precious friendship.

I wish to thank Dr. Ken G. Evans for his constant help and expert consultations. I also thank Dr. Colin C. Goodyear for his valuable discussions and help.

I wish to record my thanks to Mr. Hung Bun Choi, Dr. Fabrice Plante, Mr. Matthew Pollard, Mr. Scott Watson, Mr. David Jones and other friends in the DSP group for their useful discussions and kind help. I acknowledge help also from previous members of the group including Dr. Jason Deveney, Dr. Michael Lo, Dr. Jonathan Tang, Dr. Dong-Bing Wei, Dr. Harry Fairhurst, Dr. Mark Davies, Dr. Laurence Nicolson, Dr. Gary Jamieson and Dr. Phil Crozier.

I acknowledge the help I have received from staff and technicians in the Department of Electrical Engineering and Electronics. In particular I wish to mention Mr. Song To, Mr. Fred Ormesher, Mr. Jim Room and Professor Jim Moruzzi, all of whom have given me much help and friendship. I also wish to mention Dr. John Earis who supported my research for Fazakerly Hospital.

I am especially grateful to all my friends in the Department and in Liverpool for the time and experience I have had and that I will remember forever.

I thank British Telecom Laboratories for providing facilities and financial support which have made this research possible. Thanks are expressed to Liverpool University for making facilities available and to the Overseas Research Studentship Committee for its financial assistance.

I am indebted to my beloved parents, my two sisters Xiao Yin and Xiao Ping, my husband Guang Xian and my son Liwei for their love, encouragement, support and sacrifice over the past years.

I dedicate this thesis to my beloved son Liwei Chen. His company and understanding has given me great encouragement, stimulation and motivation for my work.

Dedication

I dedicate this thesis to Liwei.

Abstract

There is currently much interest in the development of speech coders for representing speech at very low bit-rates, i.e. at bit-rates below 4 kb/s. Applications of such coders are to be found in digital mobile radio, internet communications and secure telephony in civil and military system. Among the techniques being actively researched for such applications are sinusoidal modelling which uses interpolation to regenerate segments of speech between representative sections. The purpose of this thesis is to study low bit-rate sinusoidal coding techniques, evaluate the performance of some coders based on these techniques and to develop new methods of improving their effectiveness.

Sinusoidal coding techniques are based on a fundamental sinusoidal model (FS model) which represents speech as the sum of sinusoids whose amplitudes, frequencies and phases are specified at regular update-points and are interpolated between update-points. This model, based on a seminal paper by McAulay & Quatieri, has been implemented, evaluated and modified. It has been confirmed that the FS model is capable of converting the original telephone bandwidth speech to a set of parameters which represent it without sacrificing speech quality. The synthetic speech quality is almost indistinguishable than that of the original speech, when the update points are at 10 ms intervals, and is only slightly degraded for 20 ms intervals. A new algorithm has been applied to improve the frequency-matching procedure required by the FS model.

A low bit-rate sinusoidal transform coder (STC) was developed from the FS model. To achieve the required low bit-rate representation of the FS model, a pitch frequency, spectral envelope and voicing probability are encoded, rather than the frequencies, amplitudes and phases of the sinusoids directly. Phases are derived at the decoder from the spectral envelope under the assumption that the vocal tract transfer function is minimum phase. The low bit-rate STC model was implemented and a quasi-pitch synchronous sinusoidal (QPSS) model was also developed as an alternative version of STC. Instead of transmitting the pitch-frequency, QPSS encodes the location of the first peak in each frame in an attempt to achieve a degree of synchronous between the original and synthesised speech.

It is found that distortion in the STC coder is introduced by the minimum phase assumption and other simplified representations. Three novel modification and optimisation techniques are proposed to reduce this distortion. Experiments have shown that the derived phases will be closer to the true phases by combining a minimum phase model with the phase spectrum of a Rosenberg pulse model or a second order all-pass filter. Optimising the shape of the spectral envelope as derived by smoothing the spectral peaks can also improve the accuracy of the phase derivation. Finally, an envelope correction method, which uses a compensation filter to reduce the frequency spreading effect due to pitch variation can further improve phase derivation.

Four fully quantised low bit-rate speech coders based on the modified STC model have been produced and evaluated. Scalar and vector quantisation techniques are used for coding the parameters which include 30 discrete cosine transform (DCT) coefficients for representing a spectral envelope at 20 ms update-points. A differential vector quantisation technique is proposed for the DCT coefficients. The four coders are designed for 4.4, 4, 3.4 and 2.4 kb/s operation. Encouraging results were obtained at the first three of these four bit-rates. For 2.4 kb/s operation it is concluded that an alternative speech envelope representation, based on all-pole LP analysis, would be more appropriate and suitable approaches are discussed.

Contents

Chapter 1
Introduction

1.1 Introduction to speech coding

Speech coding is the conversion of speech to digital form for transmission, storage and/or signal processing. Decoding is the process of converting from this digital form back to an analogue electrical signal corresponding to an acceptable version of the original speech. The purpose of much research in speech coding is to find more efficient and/or effective ways of digitising arbitrary segments of speech. Efficient generally means that the least possible memory is required for storing a given segment, or the minimum transmission channel capacity is required for continuous communications. Effective means that the highest possible quality is achieved at a given bit-rate [FSACJT, 79]. Although other applications are emerging, the most common application of low bit-rate speech coding is telephony where the speech is traditionally bandlimited between 300 Hz and 3400 Hz. It is conventional to sample such bandlimited speech at the rate of 8 kHz.

Speech coding techniques can be classified into three categories which are waveform coding, vocoding and hybrid coding.

Waveform coders, as the name implies, attempt to produce decoded speech, at the receiver, whose waveform shape is as close as possible to that of the original speech. This type of coder produces "telephone" or "toll" quality at intermediate or high bit-rates, usually more than 16 kb/s. Telephone or toll quality is defined as a level of sound quality roughly equivalent to what one would hear on a good domestic telephone line.

Waveform coders can be implemented using time-domain or frequency-domain processing techniques. The most common time-domain waveform coding technique is pulse code modulation (PCM) [JN, 84]. PCM is very widely used at 64 kb/s and is the basis of the first standard coding technique to be defined for line telephone networks. The 64 kb/s standard form of PCM uses A-law scalar quantisation with 8 kHz sampling rate and 8 bits/sample. This technique does not exploit many of the known characteristics of speech waveforms or perception and therefore it needs a high bit-rate to meet the requirements for high quality speech. In order to reduce the bit-rate without serious loss of quality, improved quantisation methods have been devised, such as differential pulse code modulation (DPCM) and adaptive differential pulse code modulation (ADPCM) [BBD, 86]. DPCM encodes the differences between successive samples rather than the samples themselves. ADPCM adapts the quantiser step-size of a DPCM coder according to the short-term speech power. Vector quantisation (VQ) techniques, which quantise groups of samples rather than single samples, have been applied to waveform coders, e.g. code-excited linear prediction (CELP), which achieves further bit-rate compression of speech and other signals [Ger, 94].

In the frequency-domain, speech may be divided into a number of frequency components or bands which may be more efficiently encoded separately [Xyd, 89]. Subband coders (SBC) [CWF, 76] [Kin, 87] and adaptive transform coders [ZN, 77] are based on this frequency-domain approach.

The term “vocoder" stands for “voice coder” or sometimes “source model coder”. In the design of vocoders, speech is considered to be the output of an excitation signal (source signal) passing through a time-varying filter which models a human's vocal tract [Atal, 82]. A set of parameters representing the characteristics of the speech are encoded by special techniques which try to minimise redundant information. A linear predictive coder (LPC) [RS, 78] [Mak, 75] [Sch, 85] is a typical example of a vocoder , and is illustrated in figure 1.1.

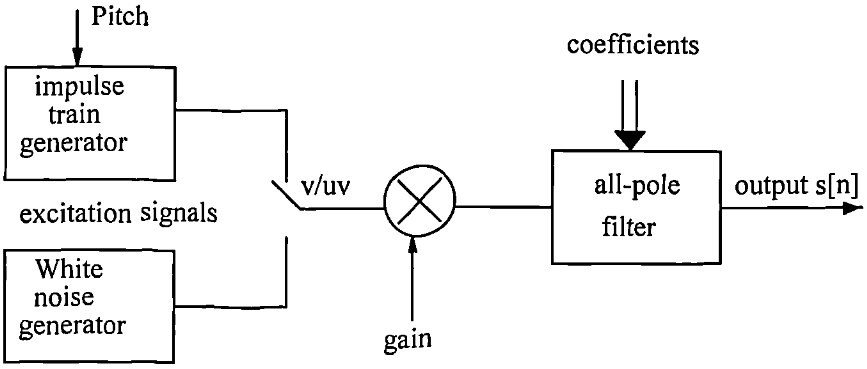


Fig. 1.1 A LPC synthesis process

The excitation signal in figure 1.1 is either a series of periodic pulses representing the periodicity of the vocal cords for voiced speech, or a noise-like random sequence for unvoiced speech. The excitation signal passes through an all-pole filter which models the spectral shaping effect of the vocal system comprising the glottis, vocal tract and lip radiation matching effect. The coefficients of the all-pole filter may be calculated for segments of speech by LPC analysis. Vocoders can encode speech at very low bit-rates, typically 2.4 kb/s, but usually with some distortion and loss of naturalness in the quality of the decoded speech.

Hybrid speech coding techniques are a combination of waveform coding techniques and vocoding techniques. Examples are multipulse LPC (MP-LPC) [AR, 82] [Boyd, 93] and code-excited linear prediction (CELP) [Ger, 94]. Instead of using a series of pulse or white noise as excitation signals, which an LPC vocoder uses, a MP-LPC coder produces a sparse sequence of amplitude (pulses) separated by zeros. The location and the amplitudes of the pulse are transmitted. A CELP coder stores residual waveforms in a pre-designed codebook to excite an all-pole LPC filter. CELP is a very important coding technique and has been adopted as the basis of several cellular radio standards. For example, the US department of defence (DOD) standard FS1016 4.8 kb/s for secure telephony and the ITU-G.728 16 kb/s standard are all based on CELP coders [WMCS, 96].

1.2 Low bit-rate speech coding and its applications

Over the last decade speech coding techniques, especially for low bit-rates, have been intensively studied and developed. The aim has been to provide toll quality at minimum bit rates for digital transmission or storage. The advantages to be gained are bandwidth efficiency, cost reduction, security, robustness to transmission errors and flexibility in the digital world.

Applications for low bit-rate speech coders are found in mobile telecommunications, secure commercial and military communications, satellite networks, voice-mail and multi-media services. [WMCS, 96]. Mobile communication equipment can be required to operate in very noisy environments. Its use places also quite severe limitations on processing complexity because of the need to preserve battery power. Speech coders designed for such applications must be robust to the addition to the speech of traffic noise and environmental noise (e.g. sounds from ventilation equipment and other machinery, background clatter, music, and speech from other talkers). The delay introduced by the algorithm must also be kept within some specified and usually very stringent limit. If a coder is to be used for military communications, it must be anticipated that some unusual types of background noise such as tank, helicopter and machine-gun sounds could be added to the speech.

Among the internationally recognised and standard coding techniques, the G.729 CELP coder [Sch, 95] achieves near toll quality at 8 kb/s. Its performance is considered comparable to the G.726 32 kbit/s ADPCM coder, which in many ways is better than 64 kb/s A law PCM. Other examples in the commercial field are the Inmarsat-B speech coding standard at 9.6 kb/s and 16 kb/s based on adaptive predictive coding (APC), the 9.6 kbit/s multipulse linear predictive coding (MPLPC) for Skyphone™ (also known as Inmarsat-aero) [WMCS, 96], and the 6.4 kbit/s improved multi-band excited (IMBE) coding standard for Inmarsat-M applications [MSDI, 91]. In the military field, important standards are the LPC-10e 2.4 kbit/s vocoder [Tre, 82] [FS, 84] based on linear predictive coding, and the FS1016 4.8

kbit/s CELP coder [WMCS, 96]. The LPC-10e vocoder provides intelligible but somewhat unnatural speech quality at a very low bit-rate.

Low bit-rate speech coding has a vital role in the future of telecommunications and data storage. There will be demands for ever lower bit-rate digital speech transmission and for greater quality and reliability over existing transmission systems. Some of these demands will arise from the need to make best use of the limited radio bandwidth available for a rapidly developing mobile telecommunications market. Experience with existing digital systems suggest that a more flexible trade-off between source and channel coding will be necessary to accommodate adverse operating conditions. The availability of lower bit-rate coding strategies will allow more bits per second to be used for channel coding where necessary, and it is likely that sophisticated variable bit-rate schemes will be used in the future to optimise the utilisation of transmission capacity under continually variable conditions.

Providing a good quality mobile telephony service at reasonable cost will lead to the number of domestic mobile customers continuing to increase dramatically. This will not be possible without increasingly efficient use of the radio transmission medium. The further development of speech coding techniques is a vital aspect of the work that is required to achieve this increase in efficiency.

The objectives for low-bit rate coding research are becoming more demanding and diverse as existing telecommunication networks grow and customer expectations increase. This is true in most areas of telecommunications, not only mobile communications.

Telephone voice services are already expanding well beyond the traditional person-to-person conversation call. Voice-mail, paging, call-minder and telephone banking are examples of such extended services. The range and availability of such services will greatly increase in the future and will become an increasing source of revenue, and perhaps more importantly a key ingredient in competitive and marketing strategies for service providers. Low bit-rate speech coding is already used in existing services and will continue to be important for the economic provision of these and future services.

Compression is increasingly being used for the storage and cataloguing of large quantities of conversational speech. Archiving speech is used for legal purposes in many professional activities. Business meetings, police interviews and financial, banking, betting and insurance transactions are examples of commonly discussed applications for speech compression. The advantage of digital rather than analogue storage for such applications is that the data is more secure and far more easily accessed.

Many speech-based systems, for example text-to-speech conversion systems [WMCS, 96] and speech recognition systems require large data-bases of speech to be made constantly available. The use of low bit-rate speech coding techniques to compress such data-bases can lead to more efficient systems in terms of storage capacity required and also in terms of the resources needed to access the data.

The use of low bit-rate speech coding combined with bit-rate compression applied to image coding, music and other signals will allow advanced multi-media services to be provided over public telephone networks. Some of these services will not require Integrated Services Digital Network (ISDN) access or a digital superhighway. The British Telecom "Videophone", now available to both business and domestic customers over ISDN, is an early example of this multi-media technology.

Low bit-rate speech coding has undoubtedly proved beneficial in many network applications. For example, it is now possible to hold a full duplex conversation over the internet. The speech transmission is via packets rather than a continuous bit stream and this aspect is likely to be increasingly important in many transmission media of the future which will be similarly based on packet switching technology. There is much interest in adapting current low bit-rate speech coding techniques to maximise their performance over the Internet medium. The adaptation to more general packet switched networks, for example ATM, will be an important area of future research in speech coding.

There are also requirements for 2.4 kbit/s speech coding schemes for a number of satellite applications [WMCS, 96] .

1.3 Thesis Organisation

This thesis introduces the principles of speech coding techniques and studies in detail the use of sinusoidal modelling techniques for coding band-limited speech at or below 4 kbit/s.

Sinusoidal modelling was introduced by McAulay and Quatieri in 1986 [McQ, 86A] and is currently receiving much world-wide research interest as a means of achieving better quality speech coders.

Chapter 1, i.e. this chapter, has given an introduction to speech coding and its applications.

Chapter 2 presents an overview of sinusoidal coding in general and surveys some of the most actively studied techniques. Comparing the common characteristics of these techniques and highlighting the important differences between them leads to an insight into some underlying principles.

Chapter 3 explains a fundamental sinusoidal (FS) model proposed by McAulay & Quatieri. It is the basis of sinusoidal coding, and is the core of the sinusoidal transform coding (STC), which is the main subject in this thesis.

Chapter 4 describes a low bit-rate sinusoidal transform coder (STC) which is based on the FS model in chapter 3. Some problems, e.g. inaccuracy of phase derivation, are raised and the need for improvements is pointed out.

Chapter 5 investigates the distortion in STC decoded speech that is due to the inaccuracy of the phase derivation. Three sources of error are idcntified and ways of reducing this error are presented. The reconstructed speech is thus improved without increasing the required bit-rate.

Chapter 6 outlines the principle of vector quantisation, and discusses techniques for code-book design, training and assessment. Details are given of vector quantisation processes for several low bit-rate coders based on the modified form of STC proposed in Chapter 5. Fully quantised coder simulations are described and evaluated.

Finally, chapter 7 summarises the whole thesis, gives a conclusion and proposes work to be done in the future.

Chapter 2
Sinusoidal coding techniques

2.1 Introduction

Sinusoidal coding techniques may be traced back to the "phase vocoder" [FG, 66] proposed by Flanagan and Golden in 1966. This was an early attempt to represent speech as a combination of narrow-band components; i.e. modulated sinusoids with a degree of randomised phase. In 1981, Hedelin [Hed, 81] proposed a pitch-independent sinusoidal model for coding a low frequency sub-band of telephone bandwidth speech. This sub-band, extending from 100 Hz to 800 Hz was represented by modulated sinusoids, the remaining spectrum being encoded by an LPC method. Hedelin's technique is pitch independent in the sense that the low frequency sub-band is represented by a set of sinusoids, typically 8, whose frequencies, amplitudes and phases are determined by an "extended Kalman" estimation process [Jaz, 70]. This is a very complicated process which, in practice, may be simplified by "peak picking" in the FFT domain, thus introducing a degree of pitch dependency. Later Almeida and Tribolet [AT, 82] developed a pitch-dependent "harmonic coding" algorithm for speech compression. With this technique, voiced speech is modelled as a sct of modulated sinusoids whose instantaneous frequencies are integer multiples of an assumed fundamental pitch-frequency. It is reported in a subsequent paper [MAT, 90] that the harmonic coding algorithm [AT, 82] can achieve very high quality digital speech transmission at 6 to 8 kb/s.

In recent years, sinusoidal coding has been the basis of many actively studied techniques for telephone band-width speech coding at very low bit-rates; i.e. at bit rates of 4.8 kb/s and below. Among these techniques are "sinusoidal transform coding" (STC) [McQ, 85] [McQ, 86A] [McQ, 92,], "multi-band excitation" coding (MBE) [GL, 88] [HL, 89] [MSDI, 91] [HL, 91], "prototype wavcform interpolation" (PWI) [Kle, 91A] [Kle, 91B] [KG, 91] [Kle, 93] [KH, 95] and "time frequency interpolation" (TFI) [Sho, 93A] [Sho, 93B].

The principle of all of these sinusoidal coding techniques is to characterise speech by spectrally analysing short representative segments extracted at suitable time intervals. The characteristics of each analysis segment are assumed to be "instantaneous" measurements which describe the speech waveform in the vicinity of a particular point in time referred to as an "update-point". These characteristics may be represented by instantaneous values of the amplitudes, frequencies and phases of a set of amplitude and frequency modulated sinusoids. The instantaneous values are quantised and efficiently encoded to constitute the low bit-rate speech representation. At the decoder, frames of speech are synthesised as sums of sinusoids with smoothly changing amplitudes, frequencies and phases obtained by interpolating between the encoded parameters. Synthesis frames begin and end at update-points and, in general, the lengths of analysis segments and synthesis frames will be different. The interpolation process is intended to approximate the changes that occur in the original speech, from one update-point to the next.

Fundamental differences between the sinusoidal coding techniques mentioned above lie in the way the spectral analysis is performed, the width of the analysis window, the use or non-use of an LPC synthesis filter, the treatment of voicing decisions and the interpolation techniques used at the decoder.

This chapter presents a brief introduction to sinusoidal coding techniques in general and surveys some of the details of the most actively studied versions i.e. STC, IMBE, PWI and TFI. An appreciation of the common characteristics of these techniques and important differences between them leads to an insight into some underlying principles and the possibility of combining the best features of different techniques into a new approach. The interpolation techniques used in these different versions have particularly interesting and important aspects which are described in some detail in this chapter. First the general features of each technique will be outlined.

2.2 Sinusoidal transform coding (STC)

Sinusoidal transform coding (STC) is based on a method [McQ, 86A] of representing speech as the sum of sinusoids whose amplitudes, frequencies and phases vary with time and are specified at regular update-points. The model was first applied to low bit-rate (8 kb/s referred to as "mid-rate") speech coding [McQ, 85] in 1985, and has subsequently been modified in various ways for use at lower bit-rates (4.8 kb/s and below) [McQ, 92].

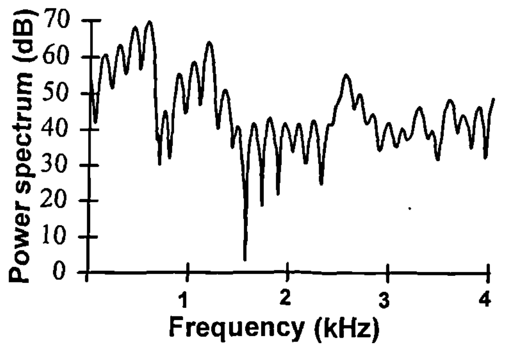


Fig. 2. 1 Voiced power spectrum

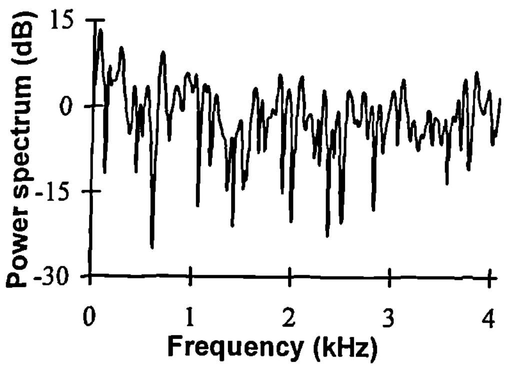


Fig. 2.2 Unvoiced power spectrum

At the analysis stage of STC, segments of the windowed speech, each of duration about two and a half pitch-periods and centred on an update-point, are zero padded and spectrally analysed via a fast Fourier transform (FFT). For voiced segments, the magnitude spectra will have peaks, in principle at harmonics of the fundamental frequency as illustrated in figure 2.1. Unvoiced segments, which are given a fixed duration, will have randomly distributed peaks, shown in figure 2.2. The frequencies of the peaks may be identified by applying a simple peak-picking algorithm and the corresponding magnitudes and phases are then obtained. These measurements become the parameters of the fundamental sinusoidal model. It has been found that speech synthesised from these parameters can be made essentially indistinguishable from the original speech when the parameters are unquantised.

For low bit-rate coding, it is not possible to encode, with sufficient accuracy, these parameters directly. An indirect method is adopted whereby an STC coder is represented by a small number of parameters. These parameters which are encoded at each update-point are:

- a spectral envelope represented by a set of cepstral coefficients
- an "onset time"
- an "ambiguity bit"
- a pitch-period
- a "voicing probability" frequency

The onset time is defined to be the location of the first vocal tract excitation point at or after the current update-point. The ambiguity-bit is said by McAulay & Quatieri [McQ, 92] to be needed as s(t) and -s(t) have the same spectral envelope and therefore -s(t) may occasionally be reconstituted instead of s(t) at the decoder. This single bit is therefore included in the encoded parameters to allow the discrepancy to be corrected when it occurs. The voicing probability frequency caters for partially and fully unvoiced speech by dividing the spectrum into two bands; a lower band, below the specified frequency, considered voiced and an upper band considered unvoiced. The onset time, the voicing probability frequency and the ambiguity-bit are determined by an "analysis-by-synthesis" procedure similar to that used by the MBE coder (see later) to determine its voicing decisions.

The STC decoder receives for each update-point a pitch-frequency measurement, a set of vector quantised parameters representing the spectral envelope, a "voicing probability" frequency, an onset time and an ambiguity-bit. Below the voicing probability frequency, the amplitudes required for a sinusoidal model are obtained at each update-point by sampling the decoded envelope at the pitch-frequency harmonics. The phase of each sinusoid is not available directly from the decoded parameters, but is deduced from the spectral envelope on the assumption that the envelope is the gain response of a minimum phase transfer function. The phase may be derived via Hilbert transform.

The process of generating frames of speech between each update-point requires natural and normally smooth evolution of the amplitudes and instantaneous phases of the modulated sinusoids constituting the sinusoidal model. This is an interpolation process, the details of which will be discussed later.

STC models speech above the voicing probability frequency by sinusoids closely spaced in frequency (say 100 Hz apart) synthesised with random phase. Again the amplitudes of these sinusoids are obtained by sampling the decoded envelope. The ambiguity bit applied to each sinusoid eliminates the possibility of random 180 degree phase reversals from one update-point to the next.

The STC coder produces one of the most promising low bit-rate representations of speech. Perhaps the most important contributions in the work of McAulay & Quatieri to the study of sinusoidal coding in general lie in the use of a pitch adaptive analysis window [McQ, 86A] and development of a frequency tracking algorithm [McQ, 86A]. As well as being effective for conventional telephone speech, STC has been reported [Spa, 94] to perform well with other types of signal such as multiple speaker and noise affected speech signals, music and biological sounds.

The study of STC is the main topic of this research thesis, and later chapters will present more detail about the issues introduced in this section. However, since STC has much in common with other sinusoidal coding techniques and since these other techniques are being proposed for the same or similar coding applications as STC, it is useful now to examine the main features of these other techniques.

2.3 Multi-band excitation (MBE)

The multi-band excitation coding technique was proposed by Griffin & Lim in 1988 [GL, 88] for 8 kb/s coding and was later improved and adapted to 4.8 kb/s by Hardwick & Lim in 1989 [HL, 89]. This improved version, referred to as "IMBE", is the basis of the Inmarsat-M voice codec for 6.4 kb/s speech transmission, with error protection, for a satellite mobile communication application [MSDI, 91].

The IMBE coder extracts parameters at update-points every 20 ms (160 samples with sampling rate 8 kHz). Each sample is represented by an integer in the range -32768 to 32767 (16-bit 2's complement) with the short term average power of the speech assumed to be scaled to a suitable level [MSDI, 91]. At the encoder, a two-stage pitch-period extraction procedure is applied. The first stage achieves an accuracy to within one half of a sampling interval, and the second stage refines the result of the first stage to an accuracy of one quarter of a sampling interval. Both stages are based on the same principle of "analysis-by-synthesis" whereby a set of pitch-period candidates, from 21 to 114 sampling intervals, are identified and an "error function" is calculated for each candidate. This error function measures the difference between the original speech segment and the same segment synthesised with the given pitch period. An important feature is that the synthesised segment has a spectrum optimised for the given pitch period; i.e. it is effectively recalculated for each candidate. When reformulated in the time-domain this procedure becomes rather like an autocorrelation measurement technique and is not much more complicated. It is made even more reliable and robust to noise affected speech by "pitch tracking" which aims to eliminate abrupt changes between successive frames. This smoothing procedure uses the pitch-period estimates for the previous two update-points and "look-ahead" estimates for the following two update-points. Looking ahead requires a buffer which contributes considerably to the overall delay of about 100 ms. An interesting feature of the pitch estimation procedure is that the initial pitch estimation algorithm uses a different analysis window from that used by the pitch refinement. The window used for initial pitch estimation is 281 samples, while the window used for pitch refinement is 221 samples. In contrast to the variable 2.5 pitch-period spectral analysis window used by STC, the spectral analysis window used by IMBE is fixed and in most cases will be considerably larger.

From the more accurate pitch-period estimate, non-overlapping frequency bands are defined, each of bandwidth equal to a specified number of pitch-frequency harmonics; typically three. The number of bands is therefore pitch-period dependent and may lie between 3 and 12. A separate voiced/unvoiced decision is now made for each of these bands in the original speech spectrum. This is done for each band by measuring the similarity between the original speech in that band and the closest approximation that can be synthesised using the (normally three) estimated pitch frequency harmonics that lie within the band. When the similarity is close, the band is declared voiced; otherwise it is declared unvoiced.

Once the voiced/unvoiced decision has been made for each band, three "spectral amplitudes" are determined for each band. For a voiced band, these spectral amplitudes are simply the amplitudes of the pitch frequency harmonics within the band. For an unvoiced band, the spectral amplitudes are determined by the root mean square of a "frequency bin" centred at the specified harmonic frequency and with bandwidth equal to the calculated fundamental frequency. Note that even unvoiced speech frames are assumed to have a fundamental frequency which will be determined by any small amounts of correlation within the unvoiced frame and, perhaps more importantly, correlation in the previous and next frames as examined by the pitch tracking procedure. The input speech is now assumed to be characterised at each update-point by a pitch-period estimate (from which the number of frequency bands may be determined) and, for each of these bands, a voiced/unvoiced decision and three spectral amplitudes.

At the decoder, the synthesised speech for each frame is a combination of voiced and unvoiced bands. Voiced speech is produced as the sum of sinusoids whose amplitudes, frequencies and phases vary across the frame. Unvoiced bands are generated by extracting appropriately band-limited portions from the DFT spectrum of a white pseudo-random sequence. Although earlier versions of MBE encoded phase information, the improved version (IMBE) does not encode phase information. For voiced bands, the phases at the update-points are chosen simply to maintain continuity with the waveform synthesised in the previous frame. Therefore there is no attempt to preserve the same phase relationship between the harmonics as existed in the original speech.

The MBE coder is based on the assumption that the short term spectra of speech, when decomposed into sub-bands, contain bands which are best considered voiced and others which are best considered unvoiced. There may be different reasons for this. For example, an analysis frame may truly contain a degree of mixed voicing, perhaps because it occurs at a transition between voiced and unvoiced utterances. Alternatively, the frame may be essentially voiced but with a rapidly changing pitch-frequency which produces apparent "spreading" [CSW, 95] of harmonics in certain frequency ranges. This spreading, caused by non-stationarity over the FFT time window, tends to flatten such regions of the spectrum and make it less like portions of a harmonic spectrum. This effect is more pronounced with IMBE than with other sinusoidal coders such as STC or PWI because the short term spectral analysis window length is generally much larger.

The IMBE approach has been very successful in representing perceptually good quality speech at bit-rates of 4.8 kb/s and lower. It is the only sinusoidal technique which has been used so far in a standard coder for commercial purpose [MSDI, 91].

2.4 Prototype Waveform Interpolation (PWI)

2.4.1 Prototype Waveform

Prototype waveform interpolation (PWI) was proposed [Kle, 91A] [KG, 91] in 1991 for speech coding at 3 to 4 kb/s. A prototype waveform is a segment extracted from a pseudo-periodic signal; i.e. a signal for which there is strong correlation, viewed over a suitably short time window, between itself and a truly periodic signal. At any point in time, the "instantaneous period" of the pseudo-periodic signal may be defined as the period of that truly periodic signal which produces the strongest correlation over a suitable time-window centred on the given point in time. For speech processing, a suitable time-window would contain at least two or three complete cycles of the periodic signal. The extracted segment must be centred on a given point in time and be of length equal to the instantaneous period at that point in time. This instantaneous period may be referred to as the instantaneous "pitch period" when the pseudo-periodic signal is voiced speech or an LPC residual derived from voiced speech. A prototype waveform may start at any point within a pitch cycle; i.e. not necessarily at a vocal tract excitation point.

The term "prototype waveform" may be applied to segments extracted directly from voiced speech waveforms or to segments extracted from residual signals obtained from an LPC analysis filter. The original idea of PWI [Kle, 91A] was to encode separately the lengths and shapes of prototype residual waveforms extracted, as illustrated in figure 2.3, at update-points which lie within voiced portions of the speech. It was proposed that a voiced/unvoiced decision should be made for each update-point, and that unvoiced segments should be encoded by switching to a form of CELP coder. The prototype residual waveform is analysed to produce a Fourier series:

$$e(t) = \sum_{\ell=1}^{P/2}\left[A_\ell \cos\left([2\pi / P]\ell t\right) + B_\ell \sin\left([2\pi / P]\ell t\right)\right] \qquad (2.1)$$

whose period P is equal to the estimated instantaneous pitch-period at the update-point. Over the analysis interval, i.e. an interval of length P centred on the update-point, e(t) is equal to the prototype waveform residual. "Time-alignment" is then applied to e(t) to make the Fourier series coefficients A_ℓ and B_ℓ as close as possible to the $A_{\ell-1}$ and $B_{\ell-1}$ coefficients of the previous update-point. The resulting A_ℓ, B_ℓ and LPC coefficients, the pitch period P, and a one-bit voicing decision are quantised and efficiently encoded for voiced speech.

Quantisation will distort not only the shape of each prototype waveform but also the way this shape changes from one update-point to the next; i.e. the waveform "dynamics". The result may be too much, too little and/or inappropriate periodicity in the decoded speech. An important innovation of PWI is the use of a "signal to change ratio"(SCR) in the quantisation procedure. This ratio measures the rate at which the periodicity is changing in the original speech, and attempts are made to quantise the parameters in such a way that this rate of change is preserved at the decoder, even when the decoded waveforms are considerably different from the original.

At the decoder, as Fourier series sine as well as cosine amplitudes are encoded, the correct phase relationships between pitch-frequency harmonics may be preserved. However, pitch synchronism with the original speech is not maintained because the prototype waveforms were time-aligned by the encoder for efficient quantisation. Further time alignment is needed at the decoder to maintain the continuity of the synthesised waveform. This is achieved by equating the phases at the update-point to the instantaneous phases attained at the end of the previous synthesis frame. A synthesised waveform similar to the original, but with an imperceptible time shift, which varies from frame to frame, is the characteristic of PWI encoded speech waveforms.

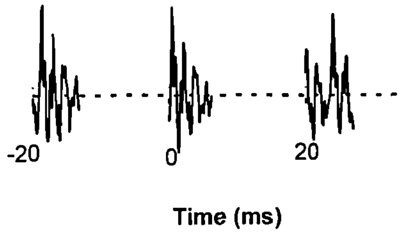


Fig 2.3 Prototype waveforms

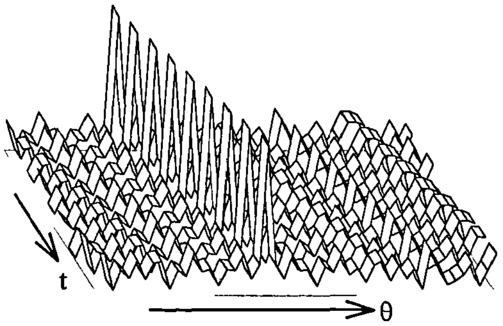


Fig. 2.4 Evolution of prototype waveform

2.4.2 Characteristic waveform (CW)

Recently [KH, 94A] [KH, 94B] [KH, 95], the concept of a prototype waveform has been generalised to include arbitrary length segments of unvoiced speech. The term "characteristic waveform" is now used. Instead of extracting a single characteristic waveform at each update-point, a sequence of about ten are extracted at regular intervals between consecutive update-points. When these waveforms are time-aligned as illustrated in figure 2.4, the changes that occur to their corresponding Fourier series coefficients are indicative of the nature of the speech. Rapid changes occur for unvoiced speech, slow changes for voiced. High-pass and low-pass digital filtering is applied to separate the effect of these changes and the resulting filtered Fourier series coefficients characterise a "slowly evolving waveform" (SEW) and a "rapidly evolving waveform" (REW) which sum to form the true characteristic waveform coefficients. The SEW is down-sampled to one waveform per update-point, rapid fluctuations having been eliminated by the low-pass filtering. The REW cannot be accurately represented at a low bit-rate. Fortunately it may be replaced at the decoder by a random waveform with similar spectral shape. The parameters for this generalisation of PWI, i.e. LPC coefficients, pitch-period and characteristics of the SEW and REW, may be encoded at 2.4 kb/s.

The quantisation of PWI parameters may be achieved with the use of a degree of differential coding combined with vector quantisation, using look-up tables, to characterise the slowly evolving component of the residual prototype waveform. The approach may be viewed as an evolution from CELP and inherits some of the advantages of both time-domain waveform coding and parameter coding techniques. The concept of characteristic waveforms with slowly and rapidly evolving components overcomes the disadvantage of having to switch between fundamentally different coding techniques for voiced and unvoiced speech.

2.5 Time-frequency interpolation (TFI)

The concept of time-frequency interpolation (TFI) [Sho, 93A] [Sho, 93B] can be viewed as a generalisation of a range of techniques which include PWI and STC. The term was used by Shoham [Sho, 93A] to describe an approach to speech coding which defines an "instantaneous" short term spectrum for each sample s[n] of a speech signal. These spectra may be assumed to evolve slowly from speech sample to speech sample. Samples of the evolving spectrum may be obtained by DFT analysis at suitable intervals. The aim is to efficiently encode these sampled spectra and to reconstruct speech by interpolating between them at the decoder. Particular forms of TFI are investigated by Sho ham [Sho, 93A] [Sho, 93B] which are similar to PWI but have distinguishable features.

The concept of TFI is to associate each speech or residual sample x[n] with a sequence of M[n] samples which are spectrally analysed to produce a short term DFT spectrum $\{X_n [k]\}$. If such spectra are computed at regular intervals of N samples, the "rate" of the TFI scheme is said to be 1/N spectra per sampling interval. The value of M[n] may be variable and is often, but not necessarily, related to the instantaneous pitch period.

Two forms of TFI are referred to by Sho ham: "low rate" and "high rate". Low rate TFI (LR-TFI) has N > M[n] which means that the sampled spectra are relatively far apart and the interpolation process must generate a time waveform between pairs of spectrum sampling points, whose instantaneous spectrum evolves in a suitable way. When M[n] is made equal to the instantaneous pitch-period for voiced speech, as in PWI, the interpolation is achieved by means of the inherent periodicity of the inverse DFT. Outside the DFT time window, the IDFT waveform repeats with period M[n] and will therefore tend to match the pseudo-period voiced waveform. When the M[n] is larger than the pitch-period, as with STC, a similar effect can be achieved by determining the pitch-frequency harmonics by peak picking, and interpolating between the parameters of these harmonics. Having M[n] smaller than the pitch-period raises considerable difficulties and is not considered

High rate TFI (HR-TFI) has N < M[n] which means that the spectra obtained at the spectral sampling points will be more similar to each other than with LR-TFI, though there will be more of them over a given time span. In most cases the spectral analysis windows will overlap. The periodicity of the inverse DFT plays a less critical part in the interpolation process, and in principle, a smoother and more accurate description of the signal can be obtained, though encoding the spectra at low bit-rates becomes a more challenging problem. The use of a differential vector quantisation scheme to encode the difference between successive spectra, as used with PWI for example, is still viable. The greater similarity between successive spectra with HR-TFI will enhance the effectiveness of the differential aspect of the scheme, thus compensating for the fact that there are more spectra to quantise in a given time span.

Whereas the original PWI concept could be described as a form of LR-TFI, the later work by Kleijn [KH, 95] on characteristic waveform interpolation conforms in some ways to the definition of HR-TFI. Spectra are obtained at frequent intervals and are quantised by decomposing their evolution into slowly and rapidly evolving components and applying different techniques to each.

As well as defining the general concept of TFI, Shoham [Sho, 93A] [Sho, 93B] also proposes a low bit-rate coder based on HR-TFI. This coder has two versions for 4.05 kb/s and 2.4 kb/s operation. The HR-TFI procedure is applied in the residual domain with N = 40 samples and DFT window length M[n] said to be approximately equal to [Sho, 93A] or equal to [Sho, 93B] the instantaneous pitch-period at time n. The LPC coefficients are updated less frequently than with PWI, for example, (every 60 ms for the 2.4 kb/s version) and are block-interpolated at intervals of N samples. A predictive vector quantisation process is used, the differences between successive spectra being quantised to a trained code book. Test results were reported [Sho, 93A] [Sho, 93B] to show that the 2.4 kb/s TFI coder performed very similarly to full rate (13 kb/s) GSM and 7.95 kb/s IS54 when coding IRS filtered telephone bandwidth speech.

2.6 Interpolation techniques

2.6.1 Introduction

Sinusoidal coding techniques, including those referred to in this chapter, resynthesise frames of speech at the decoding stage by interpolating spectral information from one update-point to the next as illustrated in the diagram below.

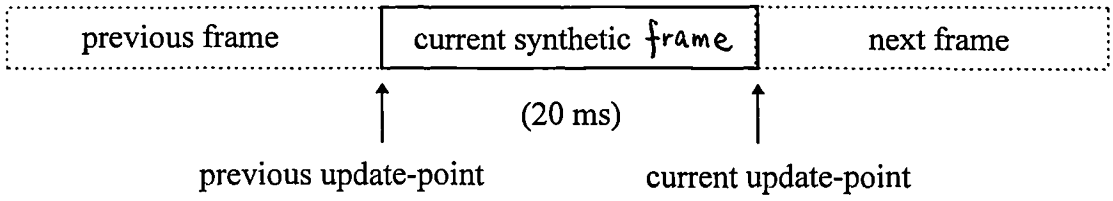


Fig. 2.5 Relationship of synthesis frames and update-points

The synthesis procedure may be described in terms of a generalisation of the concept of a Fourier series, where the amplitudes and frequencies of the harmonics may vary with time. These variable parameters have known instantaneous values at the update-points.

Such a generalised Fourier series is:

$$x(t) = \sum_{\ell=1}^{L}\left[a_\ell(t)\cos\left(\theta_\ell(t)\right) + b_\ell(t)\sin\left(\theta_\ell(t)\right)\right] \qquad (2.2)$$

where x(t) is a speech or LPC residual signal, $a_\ell(t)$ and $b_\ell(t)$ are the instantaneous amplitudes and $\theta_\ell(t)$ is the instantaneous phase of the ℓ^{th} sine and cosine term of the series. Since

$$a\cos(\theta) + b\sin(\theta) = A\cos(\theta + \phi) \qquad (2.3)$$

where $A = \sqrt{a^2 + b^2}$ and $\phi = \tan^{-1}$ (b/a), there would be no loss of generality if the generalised Fourier series were restricted to cosine terms only; i.e. if the speech or residual signal were expressed as:

$$x(t) = \sum_{\ell=1}^{L}\left[A_\ell(t)\cos\left(\sigma_\ell(t)\right)\right] \qquad (2.4)$$

Both forms of this generalised Fourier series are seen in sinusoidal coders; the first is used in the synthesis procedure proposed for PWI, and the second is used by STC, IMBE and TFI.

Let t = 0 and t = NT correspond to successive update-points. The objective is to synthesise a frame of speech or residual for values of t between 0 and NT. Consider the simpler all-cosine Fourier series first. Instantaneous amplitude measurements,

$A_\ell(0)$ and $A_\ell(NT)$, will be known at t = 0 and t = NT. To obtain intermediate values across the frame, a straightforward linear interpolation scheme may be used:

$$A_\ell(t) = A_\ell(0) + (A_\ell(NT) - A_\ell(0))\, t / NT \qquad (2.5)$$

Deriving formulae for the instantaneous phases $\sigma_\ell(t)$ is a little more complicated and there are important differences in the approaches adopted by different sinusoidal coding techniques. It must be arranged that the instantaneous phases of all sinusoids smoothly change across the synthesis frame and that discontinuities do not occur at synthesis frame boundaries. The derivative of the instantaneous phase $\sigma_\ell(t)$ of a sinusoid with respect to time is the instantaneous frequency denoted $\omega_\ell(t)$. For each sinusoid, $\omega_\ell(t)$ will be known at each update-point and must also be arranged to change smoothly across the synthesis frame. The instantaneous frequency of each sinusoid at an update-point will be determined by the encoded pitch-period. If the pitch-period is denoted by P(0) seconds at t = 0 and P(NT) seconds at t = NT, and the modulated sinusoids are assumed to remain harmonically related; i.e. $\omega_\ell(t) = \ell\ \omega_1(t)$ for all t, it follows that $\omega_\ell(0) = 2\pi\ell / P(0)$ and $\omega_\ell(NT) = 2\pi\ell / P(NT)$ for each value of ℓ. The continuity requirement means that the value of instantaneous phase $\sigma_\ell(0)$ for each sinusoid at the beginning of a synthesis frame must be equal to $\sigma_\ell^P(NT)$ where this denotes the value reached at t = NT by the formula for $\sigma_\ell(t)$ used in the previous synthesis frame. For each value of ℓ, the three conditions:

(i) $\omega_\ell(0) = 2\pi\ell / P(0) = d\sigma_\ell(t)/dt$ at $t = 0$

(ii) $\omega_\ell(NT) = 2\pi\ell / P(NT) = d\sigma_\ell(t)/dt$ at $t = NT$

(iii) $\sigma_\ell(0) = \sigma_\ell^P(NT)$

may be satisfied by a quadratic polynomial which may therefore be defined to be $\sigma_\ell(t)$ for t in the range 0 to NT. This is essentially the approach used by IMBE, as will be seen. It is referred to as "quadratic interpolation" of instantaneous phase.

When this technique is implemented, the instantaneous phases will likely be set to zero at the beginning of the first frame, and since the instantaneous frequencies $\omega_\ell(t)$ will always be exact harmonics of $\omega_1(t)$, the instantaneous phase $\sigma_\ell(t)$ for each ℓ will, in principle, remain equal to $\ell\,\sigma_1(t)$. However, rounding errors in fixed or floating point arithmetic will accumulate to unlock this relationship between instantaneous phases over a period of time. The phase relationships between sinusoids will therefore drift freely, albeit very slowly, over time resulting in gradual wave-shape changes unrelated to changes occurring in the speech or residual waveform. As the instantaneous phases were never correct in relation to the original speech signal anyway (being all set to zero initially) this does not detract from the technique in practice. However, it is useful to be aware that the drifting of phase relationships will take place. The IMBE coder, for example, uses this form of quadratic interpolation. It is possible to arrange that the drifting effect does not go beyond frame boundaries by replacing condition (iii) above by the following:

(iii) $\sigma_\ell(0) = \ell\ \sigma^P{}_1\,(NT)$

This modification has been used in some coding techniques, e.g. PWI [KG, 91], but has the disadvantage that exact continuity of higher frequency sinusoidal components is not guaranteed, and small discontinuities have indeed been observed in practice with this approach.

The quadratic interpolation technique as applied above to the cosine only Fourier series has no provision for making the phase relationships between the sinusoids correspond to the best possible model of the phase spectrum of the speech or residual waveform. It is well known that the resulting phase distortion introduces a degree of unnaturalness and loss of speech quality. Various coding techniques, including STC, attempt to make the instantaneous phases at the update-points correspond to the true phase spectrum. Considering again the three conditions on $\sigma_\ell(t)$ given above, this modifies the third condition to:

(iii) $\sigma_\ell(0) = \phi_{\ell 0}$

and introduces a fourth condition:

(iv) $\sigma_\ell(NT) = \phi_{\ell 1}$

where $\phi_{\ell 0}$ is the specified instantaneous phase of the ℓ^{th} sinusoid at the update-point corresponding to t = 0, and $\phi_{\ell 1}$ is the instantaneous phase of the ℓ^{th} sinusoid at the next update-point, corresponding to t = NT. If a formula for $\sigma_\ell(t)$ is to be found which is a polynomial in t and satisfies these four conditions, the polynomial must now be a cubic. A means of deriving the required cubic polynomial is discussed in the next section. The use of this polynomial is termed "cubic interpolation" of instantaneous phase. It is used by STC.

Instead of using the cosine only Fourier series, the original version of PWI [KG, 91] quantises, at each update-point, the a_ℓ and b_ℓ coefficients of the original form of the Fourier series. At the decoder, these coefficients are linearly interpolated individually to obtain:

$$a_\ell(t) = a_\ell(0) + (a_\ell(NT) - a_\ell(0))\,t/NT \qquad (2.6)$$

$$b_\ell(t) = b_\ell(0) + (b_\ell(NT) - b_\ell(0))\,t/NT \qquad (2.7)$$

for each value of ℓ, and the instantaneous phases $\theta_\ell(t)$ are interpolated by a technique which is similar to quadratic interpolation as described above. Now the generalised Fourier series:

$$x(t) = \sum_{\ell=1}^{L}\left[A_\ell(t)\cos(\theta_\ell(t)) + b_\ell(t)\sin(\theta_\ell(t))\right] \qquad (2.8)$$

may be expressed as

$$x(t) = \sum_{\ell=1}^{L} A_\ell(t)\cos(\theta_\ell(t) + \phi_\ell(t)) \qquad (2.9)$$

where $A_\ell(t) = \sqrt{a_\ell(t)^2 + b_\ell(t)^2}$

and $\phi_\ell(t) = \tan^{-1}(b_\ell(t)/a_\ell(t))$ for each ℓ.

This demonstrates that there may be a problem with the PWI instantaneous phase interpolation technique since the instantaneous frequency of the ℓ^{th} sinusoid is effectively the time derivative of $[\theta_\ell(t) + \phi_\ell(t)]$ rather than the time derivative of $\theta_\ell(t)$. This will be discussed later.

A reasonable approximation to the effect of interpolation, at much less computational expense, is sometimes obtained by a technique referred to as "overlap-and-add". This technique is used by IMBE for unvoiced to voiced transitions and elsewhere. For each update-point, a frame is synthesised using sinusoids and/or random signals with specified parameters which now remain constant instead of being modulated. This frame is then merged with a frame produced similarly for the previous update-point. The merging of the frames is achieved by multiplying the previous frame by a decaying window function, multiplying the current frame by a growing window function, and then adding the two frames together.

In keeping with the published literature [McQ, 85] [KG, 91], the generalised Fourier series has been so far defined as a function of the continuous time variable t. This allows the instantaneous frequency to be obtained by differentiating instantaneous phase with respect to t. In practice, the Fourier series will be a function of the discrete variable n. For the purposes of calculating instantaneous frequency, it is possible to differentiate a function of n, $\sigma[n]$ say, with respect to n as though n were also a continuous variable. If $\sigma[n]$ is an instantaneous phase, the resulting expression, $d\sigma[n]/dn$, becomes the corresponding instantaneous frequency in units of radians per sampling interval.

Now that the general idea of phase interpolation has been introduced, the following sections present more detail about the different techniques that are used in sinusoidal coders. We start with the phase interpolation technique used by STC.

2.6.2 Cubic interpolation of instantaneous phase

Cubic interpolation is used by STC for a Fourier series with cosine terms only. The instantaneous frequencies and instantaneous phases of the sinusoids are specified at the update-points. To consider just one of these sinusoids which are used to reconstruct synthetic speech, its formula is:

$$A_\ell[n]\cos(\sigma_\ell[n]) \tag{2.10}$$

where $A_\ell[n]$ is the time varying amplitude and $\sigma_\ell[n]$ is the instantaneous phase. For each synthesis frame we have an initial amplitude, instantaneous frequency and instantaneous phase, and a final amplitude, instantaneous frequency and instantaneous phase for each sinusoidal component. The instantaneous phase at the beginning of the frame may be referred to as the "phase offset". The initial set of parameters and the final set of parameters are those at the update-points at the beginning and the end of the synthesis frame. Interpolating the instantaneous phase for each sinusoid is to find a function $\sigma_\ell[n]$ which varies smoothly with n across the synthesis frame, and which has the appropriate instantaneous phase and instantaneous frequency at the beginning and also at the end of the synthesis frame. To obtain a smoothly changing formula for $\sigma_\ell[n]$which satisfies these four boundary conditions, a cubic polynomial is used by STC as follows:

$$\sigma_\ell[n] = \xi + \gamma\, n + \alpha\, n^2 + \beta\, n^3 \tag{2.11}$$

If the phase offset is $\sigma_{\ell 0}$, the initial instantaneous frequency (in radians/sample) is $\omega_{\ell 0}$, the final instantaneous phase is $\sigma_{\ell 1}$and the final instantaneous frequency is $\omega_{\ell 1}$, and the synthesis frame extends from n = 0 to n = N-1, then, at n = 0:

$$\sigma_\ell[0] = \xi = \sigma_{\ell 0} \qquad \therefore \quad \xi = \sigma_{\ell 0} \tag{2.12}$$

Since the derivative of the instantaneous phase with respect to n (assuming n to be a continuous variable) is the instantaneous frequency , then

$$\frac{d\sigma_\ell[n]}{dn} = \gamma + 2\alpha\, n + 3\beta\, n^2 \tag{2.13}$$

When n = 0,

$$\left.\frac{d\sigma_\ell[n]}{dn}\right|_{n=0} = \gamma = \omega_{\ell 0} \quad \therefore \quad \gamma = \omega_{\ell 0} \tag{2.14}$$

Therefore:

$$\sigma_\ell[n] = \sigma_{\ell 0} + \omega_{\ell 0} n + \alpha\ n^2 + \beta\ n^3 \tag{2.15}$$

where α and β depend on the final instantaneous phase $\sigma_{\ell 1}$ and the final instantaneous frequency $\omega_{\ell 1}$. We must choose α and β for each sinusoid to ensure that the value of the reconstructed speech, s[n], at the end of the current synthesis frame is as specified at the update-point and will therefore be equal to value of s[n] at the beginning of the next synthesis frame. If the frame-length is N samples, this means that

$$\cos\left(\sigma_\ell[\mathrm{N}]\right) = \cos\left(\sigma_{\ell 1}\right) \tag{2.16}$$

which is satisfied if

$$\sigma_\ell[\mathrm{N}] = \sigma_{\ell 1} + 2\pi\ M \tag{2.17}$$

where M is any integer. The 2πM term must be included because the interpolation formula (equation 2.15) for $\sigma_\ell[n]$ may pass through 2π several times as n goes from 0 to N. Therefore α and β must be chosen to ensure that

$$\sigma_{\ell 0} + \omega_{\ell 0}\mathrm{N} + \alpha\ \mathrm{N}^2 + \beta\ \mathrm{N}^3 = \sigma_{\ell 1} + 2\pi\ M \tag{2.18}$$

Further, the instantaneous frequency for each sinusoid at the end of the current synthesis frame must be as specified at the update-point and therefore equal to the instantaneous frequency at the beginning of the next frame:

i.e.
$$\left.\frac{d\sigma_\ell[n]}{dn}\right|_{n=N} = \omega_{\ell 1} \tag{2.19}$$

i.e.
$$\omega_{\ell 0} + 2\alpha\ N + 3\beta\ N^2 = \omega_{\ell 1} \tag{2.20}$$

Equations 2.18 and 2.20 may be solved to find α and β, each being dependent on the choice of the integer M. So the solutions are:

$$\alpha = \left(\frac{3}{N^2}\right)\left(\sigma_{\ell 1} - \sigma_{\ell 0} - \omega_{\ell 0}N + 2\pi\ M\right) - \frac{1}{N}\left(\omega_{\ell 1} - \omega_{\ell 0}\right) \tag{2.21}$$

$$\beta = \left(-\frac{2}{N^3}\right)\left(\sigma_{\ell 1} - \sigma_{\ell 0} - \omega_{\ell 0}N + 2\pi\ M\right) + \frac{1}{N^2}\left(\omega_{\ell 1} - \omega_{\ell 0}\right) \tag{2.22}$$

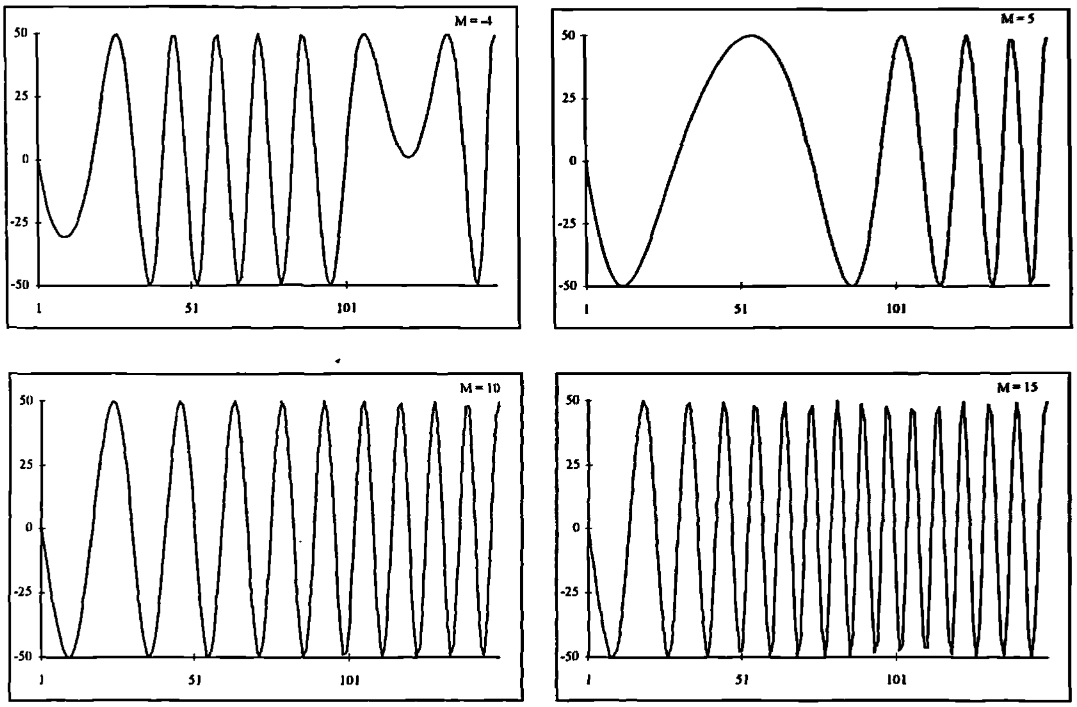


Fig. 2.6 Changes of a sinusoidal waveform for different M

The value of M chosen has a critical effect on the change in instantaneous frequency that occurs across the current synthesis frame. In all cases the amplitude and instantaneous frequency will be correct at the beginning and the end of the frame. However, for inappropriate choices of M, the change in frequency will be unnecessarily large across the frame. As an illustration, figure 2.6 shows how a particular sinusoidal component of a generalised Fourier series changes across the frame for different valuesof M. This waveform has $\sigma_{\ell 0} = \pi/2$, $\omega_{\ell 0} = \pi/20$, $\sigma_{\ell 1} = \pi/4$ and $\omega_{\ell 1} = \pi/5$. In this case, the best value of the M is 10 (A test program "PHASETEST.C" is listed in Appendix for investigating this phenomenon).

To derive a formula for the best value of M, we must again assume n to be a continuous variable. The idea is to minimise the rate of change of the slope of ω[n] across the frame by minimising:

$$f[M] = \int_0^N \left[\frac{d^2 \sigma_\ell [\mathrm{n}, \mathrm{M}]}{dn^2} \right]^2 dn \qquad (2.23)$$

where $\sigma_\ell [\mathrm{n}, \mathrm{M}]$ is $\sigma_\ell[\mathrm{n}]$ for a particular value of M and

$$\frac{d^2\sigma_\ell[\mathrm{n,M}]}{dn^2} = 2\alpha + 6\beta\ n \qquad \text{for the given M} \qquad (2.24)$$

By straightforward integration,

$$f[M] = 4N\left(\alpha^2 + 3\alpha\ \beta\ N + 3\beta\ ^2 N^2\right) \qquad (2.25)$$

and to minimise this function, we differentiate f[M] with respect to M and set the resulting expression to zero; i.e. :

$$\frac{df[M]}{dM} = 4\mathrm{N}\left(2\alpha\ \frac{\mathrm{d}\alpha}{\mathrm{dM}} + 3N\alpha\frac{d\beta}{dM} + 3N\beta\frac{d\alpha}{dM} + 6\mathrm{N}^2\beta\frac{d\beta}{dM}\right) \qquad (2.26)$$

From equations 2.21 and 2.22,

$$\frac{d\alpha}{dM} = 2\pi\left(\frac{3}{N^2}\right) \quad \text{and} \quad \frac{d\beta}{dM} = 2\pi\left(-\frac{2}{N^3}\right) \qquad (2.27)$$

Substituting α, β, $\frac{d\alpha}{dM}$ and $\frac{d\beta}{dM}$ into equation 2.26 gives:

$$\frac{df(M)}{dM} = 4N\left[2\alpha 2\pi\frac{3}{\mathrm{N}^2} + 3\mathrm{N}\alpha 2\pi(-\frac{2}{\mathrm{N}^3}) + 3\mathrm{N}\beta 2\pi\frac{3}{\mathrm{N}^2} + 6\mathrm{N}^2\beta 2\pi(-\frac{2}{\mathrm{N}^3})\right] \qquad (2.28)$$

$$= -\ 24\pi\ \beta$$

It follows that ideally β must be zero and from equation 2.22, we obtain the corresponding optimum value of M; i.e.:

$$M = \frac{1}{2\pi}\left[\left(\sigma_{\ell 0} + \omega_{\ell 0}N - \sigma_{\ell 1}\right) + \left(\omega_{\ell 1} - \omega_{\ell 0}\right)\frac{N}{2}\right] \qquad (2.29)$$

This value will normally not be a whole number, and hence it must be rounded to the nearest integer. The coefficients α and β must now be recalculated from equations 2.21 and 2.22 for this integer value of M. Substituting these new values of α and β into equation 2.15, we obtain the required formula for $\sigma_\ell[\mathrm{n}]$ for the current frame. This procedure must be repeated for all values of ℓ for each frame.

This cubic instantaneous phase interpolation technique, as used by STC [McQ, 86A] [McQ, 92], requires phase information to be provided for each sinusoidal component at each update-point and produces synthetic speech whose phase spectrum will, in principle, remain locked to that of the original speech. Hence, in principle,

(a) The reconstructed speech will be synchronous with the original speech.

(b) The waveshape in a given synthesis frame will be similar to the corresponding frame in the original speech.

In practice, the phase information, as encoded at low bit-rates, may not always be entirely accurate. With STC, it is not directly encoded but is instead derived from the magnitude spectrum via a Hilbert transform as will be seen later. In addition to inaccuracies due to quantisation and deficiencies in the minimum phase assumption underlying the Hilbert transform, there will often be uncertainty about a linear phase component of the phases specified at update-points (the linear phase will be discussed in detail in chapter 4). Further interesting problems arise due to the fact that with the cubic interpolation techniques as described above, the instantaneous frequencies of the modulated sinusoids cannot be expected to remain harmonically related throughout a synthesis frame even when they are exactly harmonically related at the update-points. This is because the procedure for finding the optimal value of M, i.e. the required multiple of 2π in equation 2.29, does not attempt to preserve a harmonic relationship. The variation in instantaneous frequency across the frame can probably be made lower at higher frequencies than the variation that would be obtained by simply multiplying the formula for instantaneous phase obtained for the fundamental by the harmonic number ℓ.

Several possible variations of the cubic interpolation scheme used by STC are also worth exploring. Firstly, there may be advantages in setting β to be exactly zero when $\ell = 1$, thus truly minimising the variation of instantaneous frequency across the frame for the fundamental. The instantaneous phase of the fundamental at the end of the frame would be calculated from the formula for $\sigma_1[n]$ and the difference between this value and the instantaneous phase specified at the next update-point would correspond to a delay of the fundamental. This delay, equal to the phase difference divided by the specified instantaneous frequency of the fundamental, would then be implemented as a linear phase component in the cubic interpolation formulae for $\sigma_2[n]$, $\sigma_3[n]$, and so on. The phase relationships between the sinusoids would thus be preserved without rigidly specifying the phase of the fundamental

(ℓ = 1) at t = 0. Imperceptible time shifts would be introduced, as with PWI, but the additional freedom afforded may allow a more natural and gradual change in instantaneous frequency across a synthesis frame. We shall refer to this modified interpolation scheme as "non-aligned cubic interpolation".

Secondly, the rate of change of frequency can, with the cubic interpolation technique described above, change abruptly at update-points. There may be advantages in specifying that the second derivative of $\sigma_\ell(t)$ at t = 0 should be equal to the second derivative of $\sigma_\ell^p(t)$, i.e. the function used for the previous frame, at t = NT.

This section on cubic interpolation has raised many issues that will be taken into account throughout the development of this thesis. The next section discusses a simpler form of instantaneous phase interpolation which is in common use.

2.6.3 Quadratic interpolation of instantaneous phase

Quadratic interpolation may be used with a cosine - only generalised Fourier series where the phases of the sinusoids are not specified at the update-points. It is used by MBE and IMBE to synthesise voiced speech bands when they are relatively stationary. Since, in this case, there is no attempt to model the true phase relationships between sinusoids at the frame boundaries (update-points), the original waveshape is not necessarily preserved. It is known that disregarding phase relationships in this way entails some loss of naturalness.

The instantaneous frequencies required to resynthesise the speech will be known at the update-points and must be made to change smoothly and naturally throughout each synthesis frame. Let the instantaneous frequencies at time n for a given frame be $\omega_1[n]$, $\omega_2[n]$, $\omega_3[n]$, ..., $\omega_L[n]$. A formula for the instantaneous phase of the ℓ^{th} sinusoid may then be obtained by integrating the instantaneous frequency $\omega_\ell[n]$; i.e. :

$$\sigma_\ell[n] = \int_0^n \omega_\ell[\tau]\, d\tau + \sigma_{\ell 0} \qquad \text{for } 0 \le n < N \tag{2.30}$$

where $\sigma_{\ell 0}$ is an arbitrary constant. The value of this constant must be chosen to maintain continuity with the previous synthesis frame, i.e. $\sigma_{\ell 0} = \sigma_\ell^{\text{previous}}[N]$.

The instantaneous frequency of the ℓ^{th} sinusoid will be specified at the update-points occurring at n = 0 and n = N. Let the specified values be $\omega_{\ell 0}$ and $\omega_{\ell 1}$ respectively. A formula for $\omega_\ell[n]$ may be based on linear interpolation, i.e. :

$$\omega_\ell[n] = \omega_{\ell 0}\frac{N-n}{N} + \omega_{\ell 1}\frac{n}{N} \tag{2.31}$$

Substituting this expression into equation 2.30 and integrating gives the following quadratic formula for $\sigma_\ell[n]$:

$$\sigma_\ell[n] = \sigma_{\ell 0} + \omega_{\ell 0}n + \frac{\omega_{\ell 1} - \omega_{\ell 0}}{2N}n^2 \tag{2.32}$$

It may be verified that for each ℓ:

$$\left.\frac{d\sigma_\ell[n]}{dn}\right|_{n=0} = \omega_{\ell 0}, \qquad \left.\frac{d\sigma_\ell[n]}{dn}\right|_{n=N} = \omega_{\ell 1} \tag{2.33}$$

For low bit-rate sinusoidal coders, the instantaneous frequencies used to resynthesise the speech are normally harmonically related at the update-points, therefore $\omega_{\ell 0}$ and $\omega_{\ell 1}$ will be equal to $\ell\omega_{10}$ and $\ell\omega_{11}$ respectively for all ℓ.

It may be seen that with quadratic interpolation the sinusoids will remain exactly harmonically related throughout each synthesis frame. Therefore, for all values of n, $\omega_\ell[n] = \ell\omega_1[n]$ for all ℓ . As mentioned earlier, the instantaneous phases may drift over time with respect to each other due to the accumulation of rounding errors, though they can be locked together by defining $\sigma_\ell[n]$ to be equal to $\ell\sigma_1[n]$ for all ℓ and n, or equivalently by taking the phase offset $\sigma_{\ell 0}$ to be equal to $\ell\sigma_{10}$ for all ℓ. Equation 2.33 gives the correct instantaneous frequency at the beginning and the end of each frame for each sinusoid, and ensures continuity at the frame boundaries. It does not constrain the phase spectrum of the reconstructed speech in any other way. Because of the harmonic relationship, the formula may be re-written as:

$$\sigma_\ell[n] = \sigma_{\ell 0} + \ell\omega_{10}n + (\omega_{11} - \omega_{10})\frac{\ell n^2}{2N} \tag{2.34}$$

In IMBE, equation 2.34 is used for lower frequency sinusoids; i.e. over a frequency range which includes about a quarter of the total number of sinusoids. For the higher frequency sinusoids it is modified slightly to avoid the "buzziness" associated with too much phase coherence. The slight modification is achieved by replacing the interpolation formula 2.34 for instantaneous frequency by :

$$\omega_\ell[n] = \omega_{\ell 0}\frac{N-n}{N} + \omega_{\ell 1}\frac{n}{N} + \Delta\,\omega_\ell \tag{2.35}$$

where $\Delta\,\omega_\ell$ is a frequency offset which remains constant across the synthesis frame for each value of ℓ, but is different for each value of ℓ and varies from frame to frame. The frequency offset slightly perturbs the exact harmonic frequency relationship between the sinusoids and therefore, over time, unlocks the relationship between the instantaneous phases of the sinusoids. The instantaneous phases attained at the end of the frame will therefore be affected, and consequently the phase offset for the next frame must be adjusted to maintain continuity. The value of $\Delta\,\omega_\ell$ is chosen for each ℓ to be the smallest possible frequency deviation which makes the change in phase offset, in comparison to what it would have been with $\Delta\,\omega_\ell = 0$, a uniformly distributed random number, ρ_ℓ say, in the range $-k_\ell\pi$ to $k_\ell\pi$. The constant k_ℓ is made equal to the number of spectral amplitudes which are classed as unvoiced in the current frame, divided by the total number of sinusoids (Note that for fully voiced speech, $k_\ell = 0$). Integrating equation 2.35, we obtain:

$$\sigma_\ell[n] = \sigma_{\ell 0} + (\ell\omega_{10} + \Delta\omega_\ell)n + (\omega_{11} - \omega_{10})\frac{\ell n^2}{2N} \tag{2.36}$$

When n=N,

$$\sigma_\ell[N] = \sigma_{\ell 0} + (\ell\omega_{10} + \Delta\omega_\ell)N + (\omega_{11} - \omega_{10})\frac{\ell N}{2} \tag{2.37}$$

which for higher frequencies must be equal to $\psi_\ell[N] + \rho_\ell$ where $\psi_\ell[N]$ is the value of instantaneous phase that would have been obtained with $\Delta\,\omega_\ell = 0$, and ρ_ℓ

is the random number referred to earlier. $\psi_\ell[N]$ is obtained for each ℓ from a separate calculation which skips along from update-point to update-point, computing the phase offsets that the normal interpolation formulae would have produced; i.e. for each frame:

$$\psi_\ell[N] = \psi_\ell[0] + (\omega_{\ell 1} + \omega_{\ell 0})N/2 \qquad (2.38)$$

where $\psi_\ell[0]$ is the instantaneous phase obtained by the application of this formula to the previous frame; i.e. $\psi_\ell[0] = \psi_\ell^{\text{previous}}[N]$ + any integer multiple of 2π.

Therefore

$$\Delta\omega_\ell = (1/N)\,(\psi_\ell[N] + \rho_\ell - \sigma_{\ell 0} - (\omega_{10} + \omega_{11})\frac{\ell}{2} - 2\pi m) \qquad (2.39)$$

where m is any integer. Defining:

$$\Delta\phi_\ell = \psi_\ell[N] + \rho_\ell - \sigma_{\ell 0} - [\omega_{10} + \omega_{11}]\frac{\ell}{2} \qquad (2.40)$$

we can take m to be the integer part of:

$$\left[\frac{\Delta\phi_\ell + \pi}{2\pi}\right] \qquad (2.41)$$

to obtain a value of $\Delta\omega_\ell$ in the range -π/N to π/N.

With IMBE, the phase for a given harmonic is quadratically interpolated as described above only when the given harmonic and the corresponding harmonic at the previous update-point are both within bands declared voiced, and when the difference between the pitch frequencies is relatively small; i.e. when $|\omega_{11} - \omega_{10}| < 0.1\,\omega_{11}$. Otherwise an overlap-and-add technique is used. The following two effects will be observed with quadratic interpolation:

1. The phases of each sinusoid will drift freely in relation to the phases of the original speech.
2. The phase relationships between individual sinusoids will drift due to accumulated rounding error and also at times due to the injection of random phase.

These effects will cause the reconstructed speech waveshape not to resemble the original, though the phases changes are found to be, to a considerable extent, inaudible.

This section has concentrated on one form of quadratic interpolation which is perhaps the simplest. The technique used by PWI is in some ways an alternative form of quadratic interpolation, but because it is applied separately to the a_ℓ and b_ℓ coefficients of a generalised sine-cosine Fourier series, the effect is to preserve phase relationships as will be seen in the next section.

2.6.4 Phase interpolation used in PWI

Many variations of the quadratic interpolation technique mentioned above are possible. PWI employs linear interpolation of the pitch-period rather than the pitch-frequency, to determine a formula for the instantaneous pitch-frequency. Other instantaneous frequencies are assumed to be harmonics of the pitch-frequency and these are integrated in the normal way to obtain interpolation formulae for the instantaneous phases $\theta_\ell[n]$ for $\ell = 1$ to L. A generalised sine-cosine Fourier series of the form

$$x[n] = \sum_{\ell=1}^{L}\left[a_\ell[n]\cos(\theta_\ell[n]) + b_\ell[n]\sin(\theta_\ell[n])\right] \tag{2.42}$$

is used to synthesise an LPC residual x[n] for n = 0 to N-1, with $a_\ell[n]$ and $b_\ell[n]$ coefficients which are linearly interpolated between known values at $n = 0$ and $n = N$. It will be seen that this approach is broadly equivalent to quadratic interpolation, except that the use of sine as well as cosine terms in the Fourier series preserves the original phase relationships between harmonics. The phase offset $\theta_1[0]$ of the fundamental sine and cosine terms is calculated solely to maintain continuity and the phase offsets of the harmonics are locked to that of the fundamental. This means that although phase relationships between harmonics are preserved, imperceptible time shifts will occur from frame to frame because of the arbitrary phase offset of the fundamental.

Kl eijn [KG, 91] [Kle, 93] proposed two possible formulae for the instantaneous phase $\theta_\ell[n]$ of the ℓ^{th} sinusoid. The simpler of these is based on a linear interpolation of the instantaneous pitch period P[n] between its known values P_0 at the beginning of the synthesis frame and P_1 at the end. The interpolated value at time n within the frame is therefore:

$$P[n] = P_0 + (P_1 - P_0)\frac{n}{N} \tag{2.43}$$

Taking the ℓ^{th} sinusoid to be the ℓ^{th} harmonic of the pitch-frequency, its instantaneous phase must satisfy:

$$\frac{d\theta_\ell[n]}{dn} = \frac{2\pi\ell}{P[n]} \tag{2.44}$$

Integrating equation 2.43 for each sinusoid *l* we obtain:

$$\theta_\ell[n] = \begin{cases} \dfrac{2\pi\ell N}{P_1 - P_0}\ln\left[(P_1 - P_0)\dfrac{n}{N} + P_0\right] + K & : P_1 \neq P_0 \\[2ex] \dfrac{2\pi\ell n}{P_0} \;+\; K & : P_1 = P_0 \end{cases} \tag{2.45}$$

where K is an arbitrary constant of integration. Now $\theta_\ell[n]$ should equal to$\theta_{\ell 0}$ when n = 0, where $\theta_{\ell 0} = \theta_\ell^{\text{previous}}[N]$. Therefore

$$K = \theta_{\ell 0} - \frac{2\pi\ell N}{P_1 - P_0}\ln(P_0) \qquad :P_1 \neq P_0 \tag{2.46}$$

$$K = \theta_{\ell 0} \qquad :P_1 = P_0 \tag{2.47}$$

and it follows that:

$$\theta_\ell[n] = \begin{cases} \dfrac{2\pi\ell N}{P_1 - P_0}\ln\left[\dfrac{P_1 - P_0}{NP_0}n + 1\right] + \theta_{\ell 0} & : P_1 \neq P_0 \\[2ex] \dfrac{2\pi\ell n}{P_0} + \theta_{\ell 0} & : P_1 = P_0 \end{cases} \tag{2.48}$$

With PWI, the phase offsets are locked together by making $\theta_{\ell 0} = \ell\,\theta_{10}$ for all ℓ.

The generalised Fourier series coefficients $a_\ell[n]$ and $b_\ell[n]$ are linearly interpolated between their known values at the two update-points, i.e.:

$$a_\ell[n] = a_{\ell 0}\frac{N-n}{N} + a_{\ell 1}\frac{n}{N} \tag{2.49}$$

$$b_\ell[n] = b_{\ell 0}\frac{N-n}{N} + b_{\ell 1}\frac{n}{N} \tag{2.50}$$

which means that the formula for x[n] may be re-expressed as:

$$x[n] = \sum_{\ell=1}^{L} a_\ell[n]\cos(\theta_\ell[n] + \phi_\ell[n]) \tag{2.51}$$

where $$A_\ell[n] \;=\; \sqrt{(a_\ell[n])^2 \;+\; (b_\ell[n])^2} \tag{2.52}$$

and $$\phi_\ell[n] = \tan^{-1}\left(\frac{b_\ell[n]}{a_\ell[n]}\right) \quad \text{for each } \ell. \tag{2.53}$$

The PWI phase interpolation technique effectively separates the instantaneous phase of the ℓ^{th} sinusoid into two components $\theta_\ell[n]$ and $\phi_\ell[n]$. The first component, $\theta_\ell[n]$, forces the instantaneous frequency of the ℓ^{th} sine and the ℓ^{th} cosine term to be the ℓ^{th} harmonic of the pitch-frequency at each update-point, and to be smoothly interpolated between the update-points. This term also maintains the continuity of the sine and cosine terms at frame boundaries and locks the phases of the sinusoids together throughout the frame. Without the effect of the a_ℓ and b_ℓ coefficients, (i.e. assuming the a_ℓ and b_ℓ coefficients remain constant at unity and zero respectively) the waveform produced would be a series of impulses at intervals of the interpolated pitch period. The impulses will occur when the instantaneous phases of the harmonics are all integer multiples of 2π, and because of the phase locking, there will be no phase dispersal of these impulses; i.e. exact impulses will be produced. These impulses may be considered as an underlying excitation model as seen in most LPC based speech coders, though the impulses are not constrained to occur at sampling points; they may lie anywhere in between thus making the time-domain waveform a series of sinc-like pulses.

The second phase component, $\phi_\ell[n]$, along with the effect of $A_\ell[n]$ introduce spectral colouration which shapes the impulses to preserve the naturalness of the excitation signal being represented. The effect of $\phi_\ell[n]$ is to disperse the phases of the sinusoids.

However, this demonstrates that there may be a problem with the PWI instantaneous phase interpolation technique since the instantaneous frequency of the ℓ^{th} sinusoid is effectively the time derivative of $(\theta_\ell[n] + \phi_\ell[n])$ rather than the time derivative of $\theta_\ell[n]$. This is, in theory, satisfactory only if the time derivative of $\phi_\ell[n]$ is zero at frame boundaries. As $\phi_\ell[n]$ is derived from the $a_\ell[n]$ and $b_\ell[n]$ coefficients which are linearly interpolated there is no constraint placed on the derivative of $\phi_\ell[n]$ at frame boundaries. This effect must be expected to modify the instantaneous frequencies which will no longer be exactly as specified at frame boundaries. Further, it will introduce discontinuities of instantaneous frequency at frame boundaries. These frequency changes and discontinuities are hopefully not too serious as changes in the $a_\ell[n]$ and $b_\ell[n]$ coefficients should be relatively small from one frame to the next. They could be eliminated by cubically interpolating the $a_\ell[n]$ and $b_\ell[n]$ coefficients with the constraints imposed that their time derivatives at $n = 0$ and at $n = N$ are zero.

The PWI interpolation technique appears to have some interesting features that may be worth incorporating into other sinusoidal coders such as STC. This is quite possible even with cosine-only series. The fact that pitch-period rather than pitch-frequency is interpolated with PWI is, we believe, not significant in practice. This comment applies to the 'linear interpolation of pitch-period with respect to time' procedure outlined above and also to the more complicated 'linear interpolation of pitch-period with respect to pitch cycle phase' procedure proposed originally by Kleijn [Kle, 93]. Experiments have shown that modifying either of these procedures to 'linear interpolation of pitch-frequency with respect to time' has no discernible effect on speech quality. The latter is effectively quadratic interpolation applied to the sine-cosine series.

Advantages of the PWI approach may lie in the inherent impulse train excitation model introduced be the phase locking, the preservation of harmonic relationships throughout the synthesis frame and the flexibility gained by allowing (hopefully) imperceptible time shifts to occur without non-linear phase distortion. Disadvantages may arise from the linear interpolation of a_ℓ and b_ℓ coefficients causing frequency shifts and frequency discontinuities as described above, though this can be remedied.

2.6.5 Overlap-and-add

A reasonable approximation to the effects of interpolation, at much less computational expense, is obtained by an "overlap-and-add" technique which is used by IMBE and STC. For each update-point, a frame is synthesised using sinusoids and/or random signals with the specified parameters which now remain constant. This frame is then merged with a frame produced by the previous update-point by multiplying the latter be a gradually decaying window, the current frame by a gradually increasing window and adding them together. The details of the overlap-and-add technique will be discussed in the next chapter.

2.7 Comparisons of sinusoidal coding techniques

Now that four sinusoidal coding techniques have been introduced and their interpolation techniques at the receiver have been discussed in some detail, a more general comparison can be made between them.

Important differences lie in the lengths of the window used for short term spectral analysis at the encoder. For MBE, the relatively large analysis segment gives good spectral resolution and noise immunity though the effects of non-stationarities, due to pitch-frequency variation for example, will be apparent in the resulting spectra.

These effects are accommodated to a considerable extent by the multi-band approach which can declare a band unvoiced when frequency spreading causes it to appear not to be harmonically related to the estimated pitch frequency. A PWI analysis segment, being of length equal to a single pitch-period, may be expected to give a more accurate spectral representation of voiced speech at an update-point and be less affected by pitch-frequency variation and other non-stationarities. The variable length of the analysis segment adopted by STC is not a critical factor and is chosen to allow an accurate spectral envelope to be determined without serious distortion caused by non-stationarity.

Also of considerable importance is the fact that STC and IMBE directly encode the speech signal whereas PWI and TFI, as originally proposed, apply sinusoidal coding and interpolation to an LPC residual. This demarcation has been transgressed with STC-based coders designed to operate in the residual domain [YKE, 90] and there is probably no fundamental reason why PWI should not be applied directly to a speech waveform. However, the intricate details of each coder have been specifically adapted to the intended domain of operation, and would require careful modification if applied in a different domain. For example, the frequency discontinuity may be more significant if the PWI is applied directly to the speech domain.

IMBE, STC and TFI have the advantage of not including any phase information in the encoded output, whereas PWI attempts to encode the phase of the residual by sending Fourier series sine and cosine coefficients. IMBE and TFI disregard the phase spectrum entirely producing speech or residual waveforms that are not expected to resemble the original. The resynthesis procedure used by STC uses a minimum phase assumption to regenerate a phase spectrum from encoded magnitude-only information. This assumption will be questioned in this thesis. The addition of judicial amounts of random noise into the phase spectra of resynthisised speech, usually in higher frequency bands, has been found to subjectively improve speech quality. Most sinusoidal coders make some use of this fact.

At the decoder, there are important differences in the way each of the techniques adapt the model to unvoiced frames. IMBE applies band-limited portions from the

DFT spectrum of a white pseudo-random sequence to unvoiced bands; PWI switches to CELP for unvoiced frame; STC generates a set of sinusoids whose frequencies is 100 Hz apart and phases are random between -π to π to the unvoiced frequency range.

Vector quantisation [MRG, 85], i.e. the simultaneous quantisation of groups of parameters by indexing a look-up table or codebook, is used liberally by all the techniques, e.g. for LPC coefficients, spectral amplitudes, SEW and REW.
Among these sinusoidal coding techniques IMBE is the only coder has been adapted to be the commercial use, which is 6.4 kb/s Inmarsat-M voice codec including error correction. PWI and STC are still in a stage of continual development and TFI has no further publications recently. Variations of these coding techniques have been proposed, such as a version of PWI with the interpolation applied in the time-domain [LL, 94] rather than frequency-domain.

New lower bit-rates are being achieved by these sinusoidal coders. Both IMBE and STC 2.4 kb/s coders were submitted as candidates for latest DOD standard. It was also reported [KH, 95] that a 2.4 kb/s CW coder, which is the later form of PWI, has equivalent performance to the 4.8 kb/s FS1016 standard.

2.8 Conclusion

The use of sinusoidal coding techniques has enabled great advances to be made in the field of low bit-rate coding for telephone bandwidth speech. The computational complexity of the techniques described here is quite high, but generally within the capacity of modern single chip DSP microprocessors. IMBE has been implemented on such processors, for speech coding at 6.4 kb/s (with error coding). Lower bit-rate versions based on IMBE have been proposed. The development of the best possible 2.4 kb/s speech coder remains a topic for active research, and new ideas are being reported all the time. At present it seems possible that there is much potential for refinement and cross-fertilisation of the ideas behind the different coding techniques reported here. This thesis is predominantly concerned with the STC approach and the next chapter describes the fundamental sinusoidal model on which it is based.

Chapter 3
Fundamental Sinusoidal Model

3.1 Introduction

Sinusoidal transform coding (STC) is based on a fundamental sinusoidal (FS) model proposed by McAulay & Quatieri in 1986 [McQ, 86A]. This model represents voiced and unvoiced speech as the sum of a variable number of frequency and amplitude modulated sinusoids. For a typical speech signal, it may be necessary to add 30 or 40 such sinusoids together to obtain synthetic speech that sounds like the original.

At the analysis stage, the amplitudes, frequencies and phases of the sinusoids need to be determined at suitable intervals of time. This can be done by means of a short-time Fourier transform (STFT) applied to a segment of speech centred on each update-point. For voiced speech the STFT produces spectra with concentrations of energy at or around harmonically related frequencies. These energy concentrations appear as local maxima in the magnitude spectrum and are referred to as "peaks". A simple peak-picking algorithm may be used to locate the peaks in the magnitude spectrum, and their amplitudes may then be found. The frequencies and amplitudes thus determined from the magnitude spectrum become the frequencies and amplitudes of sinusoids constituting the FS model at a given update-point. The phases of these sinusoids at the update-point are obtained by sampling the short-term phase spectrum at the frequencies of the peaks.

For unvoiced speech, the same FS model may be successfully used with no voiced/unvoiced decision needed. In this case, the location of spectral peaks and hence the frequencies of the sinusoids will no longer be harmonically related. There will, in general, be more peaks, and they will change rapidly from update-point to update-point.

To resynthesise speech from the FS model, the amplitudes, frequencies and phases of the sinusoids must be interpolated between update-points to achieve a smooth progression. This requires links to be established between the sinusoids at one update-point and slightly different sinusoids at the next. It is referred to as the "peak tracking" problem. New sinusoids will appear from time to time which cannot be linked to previous ones. These are referred to as "births". Also, sinusoids at the previous update-point which cannot be linked to sinusoids at the current update-point are referred to as "deaths".

McAulay & Quatieri [McQ, 86A] proposed a "nearest neighbours" method for peak tracking and identifying births and deaths. In this chapter, a novel approach [Che, 93] is also described which will be referred to as the "warped frequency matching" approach. This new technique involves the shifting up or down of the frequencies of the peaks for a given update-point by a warping factor before attempting to match them to the peaks of the previous update-point. Various possible warping factors are tried until a "cross-correlation index" is maximised. This index measures the similarity between the magnitude spectrum at the previous update-point and the frequency warped magnitude spectrum at the current update-point, when only peak frequency components are taken into account. The optimally warped peak-frequencies are used as a guide during the "peak-tracking" procedure.

The resulting modulated sinusoids with interpolated parameters are summed to produce the required synthetic speech. As an alternative to the interpolation techniques, an overlap-and-add method may be applied to generate speech with much less computational complexity.

This chapter will explain the analysis and synthesis procedures of the FS model and details of their implementation.

3.2 Short-term FFT and peak-picking

The analysis procedure of the FS model essentially involves the location of peaks in the short-term magnitude spectra of frames of speech. The same procedure may be applied to the LPC residual which simplifies the peak picking procedure at the expense of some additional computation. The speech or residual is analysed around update-points at regular intervals of typically 20 ms, i.e. at intervals of 160 samples with a sampling rate of 8 kHz. For each update-point, a segment of speech or residual is extracted around the update-point, and a short-term spectrum is obtained by a fast Fourier transformation (FFT). Zero-padding is applied in the time-domain to make the window a convenient length. By analysing the FFT spectrum it is possible to identify and quantify certain kinds of regular structure in the speech or residual waveform, e.g. pitch harmonic structure.

The FFT expresses a signal segment containing time-domain samples x[n] for n = 0 to $\hat{M}$-1 as the linear combination of harmonically related complex exponentials. The complex valued FFT spectrum of order $\hat{M}$ may be defined as:

$$X[k] = \sum_{n=0}^{\hat{M}-1} x[n]e^{-j2\pi nk/\hat{M}} \qquad \text{for } k = 0, 1, \cdots, \hat{M}-1 \qquad (3.1)$$

and it follows by the inverse FFT that

$$x[n] = \frac{1}{\hat{M}} \sum_{k=0}^{\hat{M}-1} X[k]e^{j2\pi nk/\hat{M}} \qquad \text{for } n = 0, 1, \cdots, \hat{M}-1 \qquad (3.2)$$

When the FFT is applied to speech or residual, the segment $\{x[n]\}_{0,\ \hat{M}-1}$ may consist of a smaller segment of speech or residual samples $\{x[n]\}_{0,\ M-1}$ with $\hat{M}$-M zero valued samples for "padding out" the smaller segment make it of length $\hat{M}$. Typically, $\hat{M}$ will be 512 and M, which depends on the pitch period, will lie between about 50 to 280.

Once the FFT spectrum has been produced, a peak-picking procedure is used to find the frequencies and magnitudes of the peaks. The effect of the zero-padding will be to increase the number of frequency-domain samples thus making the location of peaks easier. A typical short term spectrum for voiced speech is shown as figure 3.1.

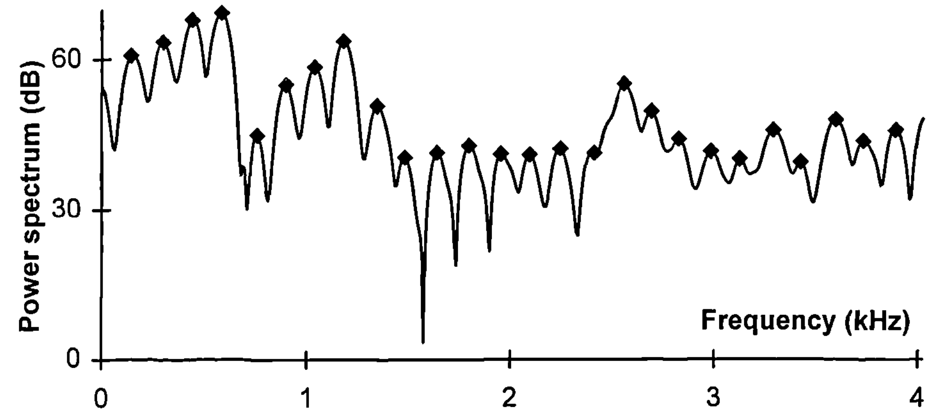


Fig. 3.1 The power spectrum of a typical voiced speech segment

A suitable peak picking procedure can be described as follows:-

- Calculate the FFT magnitudes and store them in an array md[0 : $\hat{M}$/2-1].
- Find the maximum value of magnitude.
- Set a magnitude threshold below which any peaks will be disregarded. Following McAulay and Quatieri's recommendation [McQ, 92] this threshold may be set at 80 dB below the maximum magnitude when this procedure is applied to speech. For a residual the threshold should be reduced depending on the order of the LPC analysis; values between 20 and 30 dBs were found to be appropriate.
- Find the peaks in the array md[0: $\hat{M}$/2-1] which are larger than the threshold or take the 80 largest peaks if the number of peaks exceeds 80.
- Store the frequencies of the peaks, i.e. the FFT frequency sample number for each peak, in an integer array.
- Store the corresponding magnitudes in a floating point array.
- Calculate the FFT phase spectrum and find the phase corresponding to each stored peak frequency. Store these phases, in the range of -π to π, in a floating point array.

3.3 Windowing and energy normalisation

3.3.1. Windowing

Since a segment of speech must be extracted for analysis by a process of windowing i.e. setting to zero all but the finite number of samples which form the segment, this will cause "leakage" in the frequency domain due to the effect of side-lobes of the chosen window. The frequency leakage will smear spectral peaks and affect the accuracy of the peak-picking.

The effect of the window can be illustrated by discussing the properties of two representative windows; i.e. a rectangular window and a Hamming window.

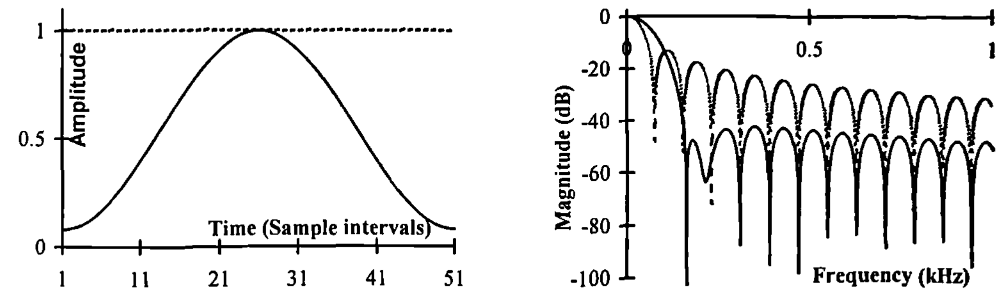
A rectangular window of length M is defined as:

$$w[n] = \begin{cases} 1 & : 0 \le n \le M-1 \\ 0 & :\text{otherwise} \end{cases} \tag{3.3}$$

A Hamming window of length M is defined as:

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(2\pi n/(M-1)\right) & : 0 \le n \le M-1 \\ 0 & :\text{otherwise} \end{cases} \tag{3.4}$$

The rectangular window applies equal weight to all samples in the chosen segment of speech or residual, and therefore truncates the signal abruptly at the edges of the window. The Hamming window applies different weights to different samples and thus introduces a gradual tapering of the amplitudes of samples towards the edges. Figure 3.2 (a) shows a 160 sample rectangular window and a 160 sample Hamming window in the time domain. Their DTFT power spectra in the range 0 Hz to 1 kHz are shown in figure 3.2 (b): dotted line for rectangular window, solid line for Hamming window. The sampling rate is 8 kHz.



(a) (b)

Fig. 3.2 (a) Waveforms of rectangular window and Hamming window

(b) Spectra of rectangular window and Hamming window

It can be seen that, in the frequency domain, the power spectrum of a rectangular window has a relatively narrow main-lobe while the width of the main-lobe of the same length Hamming window is about twice that of the rectangular window. However, the Hamming window gives much greater attenuation of the side-lobes than the comparable rectangular window. In general, the larger the window duration, the narrower the main-lobe width. However, for a given type of window, the attenuation of the worst side-lobe is to a large degree independent of window duration.

In principle, the speech or residual spectrum obtained with a rectangular window will have higher frequency resolution around pitch-harmonics due to the sharp main-lobe of the rectangular window. However, the spectrum will be seriously affected by side-lobes, which may sometimes reinforce each other and sometimes cancel each other out. This tends to produce rather random and confusing spectra particularly between pitch-frequency harmonics. The effect is called frequency leakage due to windowing. As a result, rectangular windows are not generally used in speech spectrum analysis, except when the analysis is pitch-synchronous.

Many types of non-rectangular windows exist [OS, 75] offering a range of different compromises between side-lobe reduction and loss of spectral resolution. A Hamming window has been adopted here for the FS model analysis. This gives about 20 dBs reduction of side-lobe amplitudes as compared with a rectangular window at the expense of approximately doubling the main lobe width. It may be remarked that an increase of main lobe width is not always a disadvantage of non-rectangular windows, especially when the DFT is used without zero-padding. Since zero-padding is used here it is true to say that there is loss of resolution.

Voiced speech is only quasi-periodic and pitch-frequency variations may also cause a form of frequency leakage or spreading. McAulay & Quatieri [McQ, 86A] recommended a window length which does not cause the effect to be too serious. The window length recommended by McAulay & Quatieri [McQ, 86A] for the FS model is 2.5 pitch-periods which means that the window length is pitch-period dependent even though the analysis is not pitch-synchronous.

Since each update-point is assumed to be located at the centre of its analysis segment, and the update-point should correspond to time n = 0 in the FFT window, the speech or residual segment lies effectively from -M/2 to M/2-1, where M is the window length (i.e. it should be equally distributed about the n = 0 point of the FFT). In practice, this means that the second M/2 samples of the segment become the first M/2 elements of the FFT time-domain sequence and the first M/2 samples become the last M/2 elements (i.e. elements $\hat{M}$-M/2 to $\hat{M}$-1). This is illustrated in figure 3.3 for a segment $\{s[n]\}_{0,\,M-1}$ of M speech samples.

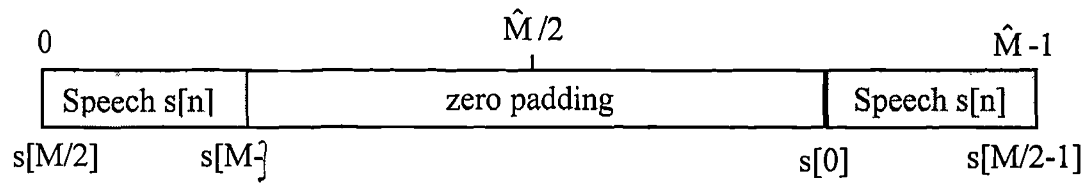


Fig. 3.3 FFT buffer in the time-domain

Therefore, the zero padded FFT samples x[n] must be defined as follows:-

$$x[n] = \begin{cases} s[M/2+n] & : 0 \le n \le M/2 \\ 0 & : M/2 < n < \hat{M}-M/2 \\ s[\hat{M}-M/2+n] & : \hat{M}-M/2 \le n < \hat{M}-1 \end{cases} \qquad (3.5)$$

This assumes M is even. If M is odd, the samples must be defined as follows:-

$$x[n]=\begin{cases} s[(M-1)/2+n] & :0 \le n \le ((M-1)-1)/2 \\ 0 & :(M-1)/2 < n < \hat{M}-(M-1)/2 \\ s[\hat{M}-(M-1)/2+n] & :\hat{M}-(M-1)/2 \le n < \hat{M}-1 \end{cases} \quad (3.6)$$

The arrangement described above will maintain the correct phase relationship between the original speech and speech synthesised by summing sinusoids devised from the calculated spectrum. If the speech segment was placed more conventionally at samples 0 to M-1 it would be centred on n = M/2, rather than n = 0 and this would introduce a phase error into each sinusoid, corresponding to a delay of M/2 samples. Since M is variable from frame to frame, the delay would vary, causing audible distortion.

3.3.2 Normalisation

Normalisation is necessary to avoid frame to frame fluctuations in the energy of reconstructed speech. Such energy fluctuations could arise from different window lengths being used as the pitch-period varies. They could also occur as the result of variations in the position of analysis windows relative to pitch cycles. For example, a window length of 2.5 pitch cycles could sometimes contain three vocal system excitation points and at other times may contain only two. There are therefore two different aspects of normalisation. One is referred to as "window normalisation" and the other is referred to as "windowed speech normalisation".

Window normalisation

Since the window length is pitch-dependent, the energy of the windowed segments of speech will vary with pitch-period. A normalisation process is therefore necessary to eliminate the effect of this variation on the estimates of the amplitudes of the sinusoidal model. A normalised Hamming window function $\{w[n]\}_{0,\,M-1}$ of length M may be defined as:

$$w[n] = \frac{0.54 \text{ - } 0.46\cos(2\pi n/(M-1))}{\sqrt{M}} \qquad \text{for } n = 0, 1, \ldots, M\text{-}1 \qquad (3.7)$$

If equation 3.7 is applied to extract M speech samples, the sum of the squared samples will be independent of M and indicative of the speech power in the vicinity of the extracted segment. This formula was used to produce the power graphs shown in figure 3.4(a) to be discussed later. If the segment is zero-padded and FFT transformed it follows by Parseval's theorem that the amplitudes of the spectral peaks will still be dependent on M. To eliminate this dependency, the input samples, windowed as above, are further divided by the square root of M prior to the FFT.

An alternative view of this problem is given in McAulay & Quatieri's paper [McQ, 92]. There it is recommended that the window function $\{w[n]\}_{0,M-1}$ be scaled for any given value of M to satisfy the condition:

$$\sum_{n=0}^{M-1} w[n] = 1 \qquad (3.8)$$

Since the sum of the un-normalised M th order Hamming window coefficients is:

$$\sum_{n=0}^{M-1}\left(0.54 \text{ - } 0.46\cos(2\pi n/(M-1))\right) \equiv 0.54M \qquad (3.9)$$

it follows that McAulay and Quatieri's normalisation of the Hamming window is equivalent to dividing each sample of the Hamming window function by 0.54M i.e. by a factor proportional to M, i.e. :

$$w[n] = \frac{0.54 \text{ - } 0.46\cos(2\pi n/(M-1))}{0.54M} \qquad \text{for } n = 0, 1, \ldots, M\text{-}1 \qquad (3.10)$$

This is equivalent to the approach described above apart from the constant scaling by 0.54 which will not affect the sinusoidal model.

Normalisation of the energy of windowed speech or residual segments

The other aspect of normalisation is that the energy of a windowed segment of voiced speech will vary with the position of the window relative to the position of pitch-cycles within the segment. This problem will occur even though the window function itself has been normalised as described above. It is a fundamental problem that is always seen with the asynchronous DFT analysis of pseudo-periodic signals, but it is potentially more serious here because the window length is not significantly greater than the fundamental frequency. Since the window length is chosen to be two and half pitch-periods for voiced speech, the window may include a segment speech with three excitation points, one close to the middle of the window with the another two close to the boundaries. At other times, it may only include two excitation points approximately symmetrically placed about the middle of the window. This effect will cause the energy of the windowed speech segment to vary from frame to frame even when the voiced speech itself is completely stationary.

The energy variation caused by window position is illustrated in figure 3.4(a) for a section of male voiced speech extracted from the data-file "Hello operator, operator, ..." [Gsp.pcm]. The section is shown in figure 3.4(b) and is about 400 samples in length with a pitch-period of about 55 samples. The dotted line in figure 3.4(a) plots the RMS value "RMS_S" of normalised Hamming windowed speech segments, each of length 2.5 pitch-periods, centred on the sampling instants specified by the horizontal axis. A value of RMS_S was calculated for each sampling instant from 100 to 300; i.e. at intervals of one sample.

The solid line in figure 3.4(a) plots the RMS value "RMS_f" of normalised Hamming windowed speech segments each with length exactly 3 pitch-periods and centred on the specified time-instant. Because it is calculated pitch-synchronously, RMS_f may be taken as a reliable indication of the short-term speech energy (square-rooted). Unnatural variations in the short-term energy from update-point to update-point will cause perceptible roughness in synthesised speech.

It can be seen that RMS_S varies by about 200 amplitude units due to the position of the window. Since the RMS value of the speech itself is around 5700 units, this is a variation of about 0.4 dB. This variation would be much larger (about 1.2 dB) with a rectangular window. The maximum RMS_S occurs when three excitation points are included in the window as shown in figure 3.5: the solid line is Hamming windowed speech and the dotted lines are the rectangularly windowed speech and the Hamming window. The minimum RMS_S occurs when only two excitation points are included in the window as shown in figure 3.6: again the solid line is the Hamming windowed speech and the dotted lines are the rectangularly windowed speech and the Hamming window.

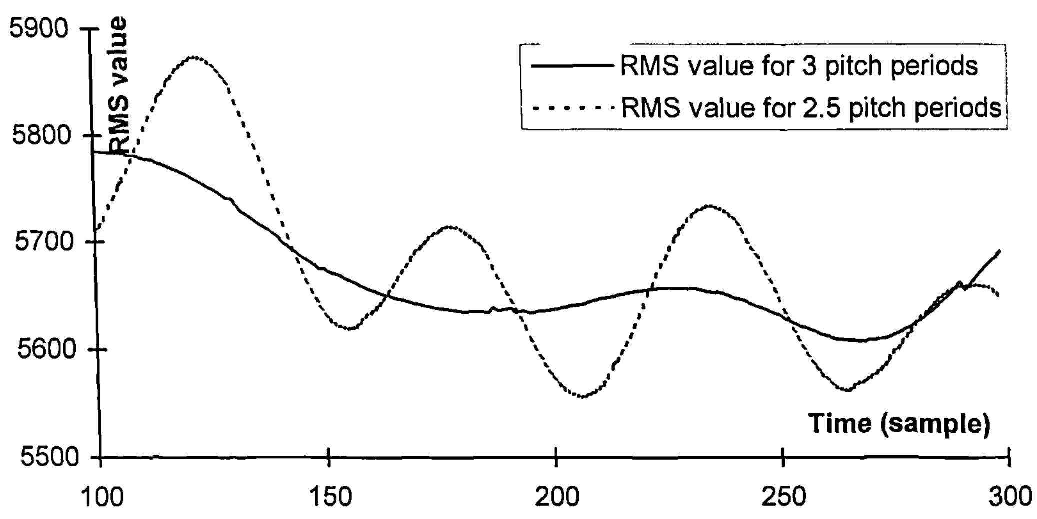


Fig. 3.4 (a) RMS values of speech with fixed and variable window lengths

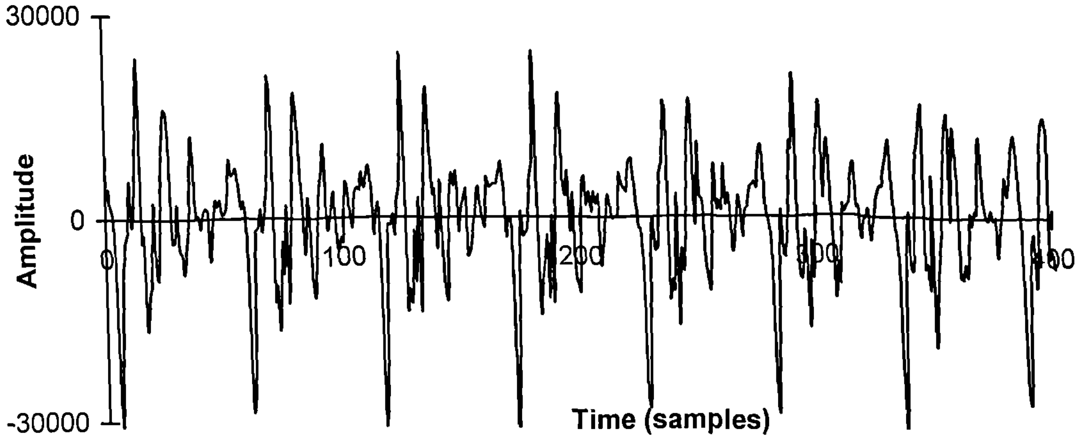


Fig. 3.4 (b) A section of male voiced speech (400 samples)

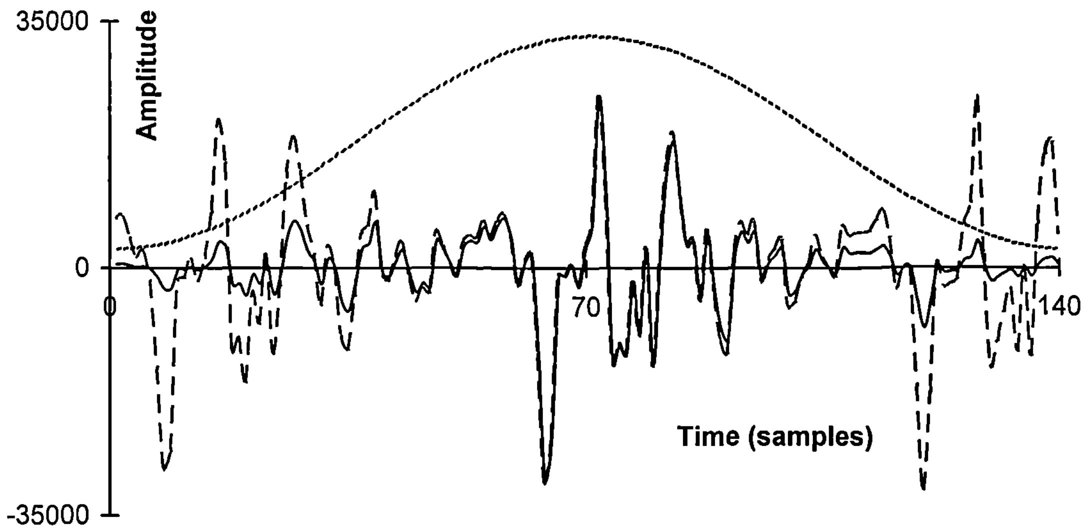


Fig. 3.5 Dotted lines: rectangular windowed speech and a Hamming window
Solid line: windowed speech.

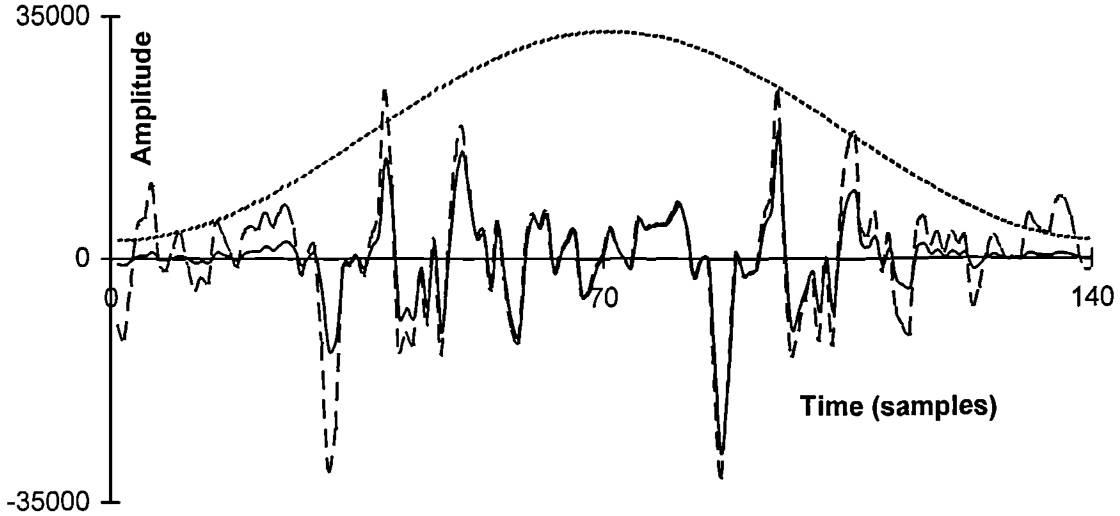


Fig. 3.6 Dotted lines: rectangular windowed speech and a Hamming window
Solid line: windowed speech.

Two ways of reducing the energy variation of windowed speech due to the positioning of the analysis window have been investigated. One way is to try, for each update-point, a range of possible analysis window positions whose central points lie, in each case, r samples to the left of the update-point with r an integer in the range -P/2 to P/2 where P is the estimated pitch period. The energy of the windowed speech is calculated for each window position and the value of r is found for which the energy is maximised. The window thus determined with maximum energy is taken as the analysis window for subsequent calculations. Experiments have shown that this maximum energy searching procedure tends to place the analysis window such that it includes three excitation points for voiced speech. Figure 3.4(a) illustrates this finding since it may be observed that the maxima in RMS_S correspond to segments containing three excitation points. The amplitudes of the peaks tend to remain an approximately constant amount above RMS_f i.e. a parallel shift, and therefore provide more reliable power estimates than would be the case if the RMSs curve were sampled arbitrarily. This procedure overcomes to a considerable extent the energy fluctuation from frame to frame. However the update-point will no longer always lie exactly in the centre of its window. The shift of the window centre with respect to the update-point does not seriously effect the measured time progression of the magnitudes and frequencies of the sinusoids and the improved accuracy in short-term magnitude spectra obtained more than make up for the slight time discrepancy introduced. The phase differences introduced are, of course, more serious and these must be corrected for by adding/subtracting a linear phase component 2πr/N from each estimated phase.

Another way of reducing the energy variation of windowed speech due to window positioning is to estimate the root mean-squared value, "RMS_f", of the speech over a window of length equal to an integer number of pitch-periods, e.g. three, or over a segment containing many pitch periods. The RMS value, "RMS_S" of each 2.5-pitch period analysis segment of speech, centred as normal on an update-point, is also calculated . The speech energy within the analysis window may then be normalised by multiplying each sample by the ratio RMS_f / RMS_S.

Comparing both methods, the second method is conceptually simpler and less computation is required. However the reliable estimation of RMS_f is not easy to achieve if it is not exactly calculated pitch-synchronously. The first method requires more computation and introduces a linear phase shift which must be corrected, but it is more reliable and practical.

3.4 Frequency matching

A set of parameters which are the amplitudes and frequencies of identified magnitude spectrum peaks and corresponding phases define the model at each update-point. A segment of speech will be synthesised between each pair of update-points by interpolating these parameters from one update-point to the next to obtain the time-varying amplitudes, frequencies and phases of the FS model. For each sinusoid, smooth and natural changes to these parameters are required across the synthesis frame and also at frame boundaries. The terms instantaneous amplitude, instantaneous frequency and instantaneous phase are used to describe the values of these variables at one particular point in time.

As mentioned before, the number of spectral peaks in the frequency range 0 Hz to half sampling frequency and the frequencies of these spectral peaks will in general not be the same from frame to frame. A matching algorithm must be applied to determine how peaks in the current frame are to be matched with peaks in the previous frame. It is also necessary to allow "births" and "deaths" to occur where peaks in the current or the previous frame cannot be matched.

Three possible effects will cause differences in the number of spectral peaks from frame to frame. Firstly, the number of peaks and the frequencies of these peaks will change as the pitch-period changes. Secondly, spurious peaks may occur from time to time due to the effect of window side-lobe interaction, and random components in the waveform. Thirdly, there may be rapid changes in the speech signal, e.g. at voicing transitions. The concept of "birth" and "death" was introduced [McQ, 86A] to account for these changes. If new peaks appear which cannot be linked to previous ones, they are referred to as "births". Peaks at the previous update-point which cannot be linked to appropriate peaks at the current update-point are referred to as "deaths".

3.4.1 "Nearest neighbours" matching technique

McAulay & Quatieri [McQ, 86A] proposed a "nearest neighbours" matching technique for frequency matching or peak tracking between two update-points. A simplified process similar to that proposed was implemented for matching these two sets of peaks. It is described below:

Phase 1: Initialisation

1.1 Define arrays preamp[1:K], prefreq[1:K] and prephase[1:K] to be amplitudes, frequencies and phases of the peaks for the previous update-point, where K is the number of the peaks at the previous update-point.

1.2 Define arrays curamp[1:L], curfreq[1:L] and curphase[1:L] to be amplitudes, frequencies and phases of the peaks for the current update-point, where L is the number of the peaks at the current update-point. The frequencies stored in arrays prefreq[1:K] and curfreq[1:L] are represented by the FFT frequency sample indices in increasing order.

1.3 Define preampnew[1:K+L], prefreqnew[1:K+L] and prephasenew[1:K+L] for holding amplitudes, frequencies and phases for the previous update-point after frequency matching.

Define arrays curampnew[1:K+L], curfreqnew[1:K+L] and curphasenew[1:K+L] for holding amplitudes, frequencies and phases for the current update-point after frequency matching.

These "new" arrays are just for making the links and interpolation, and are calculated afresh for each new frame. Only the contents of arrays curfreq[1:L], curamp[1:L] and curphase[1:L] are preserved for the next frame.

1.4 Define a "matching threshold" δ to be a frequency difference (measured in FFT frequency intervals) that may be considered small. To simplify this explanation of the algorithm, it will be assumed that δ is constant, typically 5 samples. In practice δ can be made to vary with the frequency range; it is generally made smaller for lower frequencies and larger for high frequencies.

1.5 If $|$prefreq[k] - curfreq[*l*]$| \le \delta$, it is defined that prefreq[k] is "close" to curfreq[*l*]. If $|$prefreq[k] - curfreq[*l*]$| > \delta$, it is defined that prefreq[k] is "not close" to curfreq[*l*]. If $|$p - curfreq[*l*]$|$ < $|$q - curfreq[*l*]$|$ for frequency p and q then p is "closer" to curfreq[*l*] than q.

1.6 Initialise each of the indices I_p , I_c and I_m to one. These will be used for the "previous", "current" and "new" arrays respectively.

Phase 2: Matching procedure:-

2.1 If prefreq[I_p] is close to curfreq[I_c]:-

There are three possible cases:-

(a) If prefreq[I_p] is not closer to curfreq[I_c+1] than to curfreq[I_c], and curfreq[I_c] is not closer to prefreq[I_p+1] than to prefreq[I_p] then the frequency in prefreq[I_p] is matched to curfreq[I_c].

So prefreq[I_p] and curfreq[I_c] are copied into prefreqnew[I_m] and curfreqnew[I_m] respectively, and the corresponding amplitude and phase are copied from preamp[I_p], prephase[I_p], curamp[I_c] and curphase[I_c] into preampnew[I_m], prephasenew[I_m], curampnew[I_m] and curphasenew[I_m] respectively.
Then indices I_p, I_c and I_m are all increased by one and Phase 2 terminates.

(b) If prefreq[I_p] is closer to curfreq[I_c+1] than to curfreq[I_c]:-

if prefreq[I_p+1] is close to curfreq[I_c+1] then match prefreq[I_p] to curfreq[I_c] as described in (a) above; otherwise declare curfreq[I_c] as a birth and match prefreq[I_p] to curfreq[I_c+1]. Afterwards Phase 2 terminates.

To declare curfreq[I_c] as a birth, which means that the frequency curfreq[I_c] will be matched to the same frequency in the previous frame, the birth frequency is copied into curfreqnew[I_m] and the corresponding amplitude curamp[I_c] and phase curphase[I_c] are copied into curampnew[I_m] and curphasenew[I_m] respectively. A new element is created by storing the frequency curfreq[I_c] in prefreqnew[I_m], with an amplitude of zero and a phase of

$$\text{curphase}[I_c] - N \cdot (2\pi\ \text{curfreq}[I_c]/\hat{M})$$

into preampnew[I_m] and prephasenew[I_m] respectively. N is the interval between consecutive update-points and $\hat{M}$ is the FFT length.

Then indices I_c and I_m are incremented by one.

To match prefreq[I_p] to curfreq[I_c+1], the frequencies prefreq[I_p] and curfreq[I_c+1] are copied into prefreqnew[I_m] and curfreqnew[I_m] respectively. Their corresponding amplitudes and phases, which are preamp[I_p], curamp[I_c+1], prephase[I_p] and curphase[I_c+1], will be copied into preampnew[I_m], curampnew[I_m], prephasenew[I_m] and curphasenew[I_m] respectively.

Then indices I_p, I_c and I_m are each increased by one.

(c) If curfreq[I_c] is closer to prefreq[I_p+1] than to prefreq[I_p]:-

if curfreq[I_c+1] is close to prefreq[I_p+1] then match prefreq[I_p] to curfreq[I_c] as described in (a) above; otherwise declare prefreq[I_p] as a death and match prefreq[I_p+1] to curfreq[I_c]. Afterwards Phase 2 terminates.

To declare prefreq[I_p] as a death, which means that the frequency in prefreq[I_p] will be matched to the same frequency in the current frame, the death frequency is copied into prefreqnew[I_m] and the corresponding amplitude and phase are copied from preamp[I_p] and prephase[I_p] into preampnew[I_m] and prephasenew[I_m] respectively. A new element is created by storing the frequency prefreq[I_p] in curfreqnew[I_m], an amplitude of zero in curampnew[I_m] and a phase of

$$\text{prephase}[I_p] + N \cdot (2\pi \cdot \text{prefreq}[I_p]/\hat{M})$$

into curphasenew[I_m]. N is the interval between consecutive update-points and $\hat{M}$ is the FFT length.

Then the indices I_p and I_m are incremented by one.

To match prefreq[I_p+1] to curfreq[I_c], the frequencies prefreq[I_p+1] and curfreq[I_c] will be copied into prefreqnew[I_m] and curfreqnew[I_m] respectively. Their corresponding amplitudes and phases, which are preamp[I_p+1], curamp[I_c], prephase[I_p+1] and curphase[I_c], will be copied into preampnew[I_m], curampnew[I_m], prephasenew[I_m] and curphasenew[I_m] respectively.

Then indices I_p, I_c and I_m are each increased by one.

Now all possible conditions with prefreq[I_p] close to curfreq[I_c] have been dealt with.

2.2. If prefreq[I_p] is not close to curfreq[I_c]:-

There are two cases to consider:-

(a) if prefreq[I_p] > curfreq[I_c] then curfreq[I_c] is declared as a birth as described in step 2.1 (b).

(b) if prefreq[I_p] < curfreq[I_c] then prefreq[I_p] is declared as a death as described in step 2.1(c).

Then Phase 2 terminates.

Phase 3 Procedure control:-

Noting that the number of elements in arrays prefreq and curfreq are K and L respectively, there are three cases to consider:

(a) If $I_p > K$ then all elements in prefreq have been matched. If $I_c < L$ elements curfreq[I_c], curfreq[I_c+1], ..., curfreq[L] remain unmatched and are declared as births. The matching algorithm has now been completed.

(b) If $I_c > L$ then all elements in curfreq have been matched. If $I_p < K$ elements prefreq[I_p], prefreq[I_p+1], ..., prefreq[K] remain unmatched and are declared as deaths. The matching algorithm has now been completed.

(c) If $I_p \leq K$ and $I_c \leq L$ then repeat Phase 2.

At the end of this algorithm, the number of elements in each of "new" arrays, prefreqnew, etc., is I_m - 1.

3.4.2 Investigation of nearest neighbours matching technique

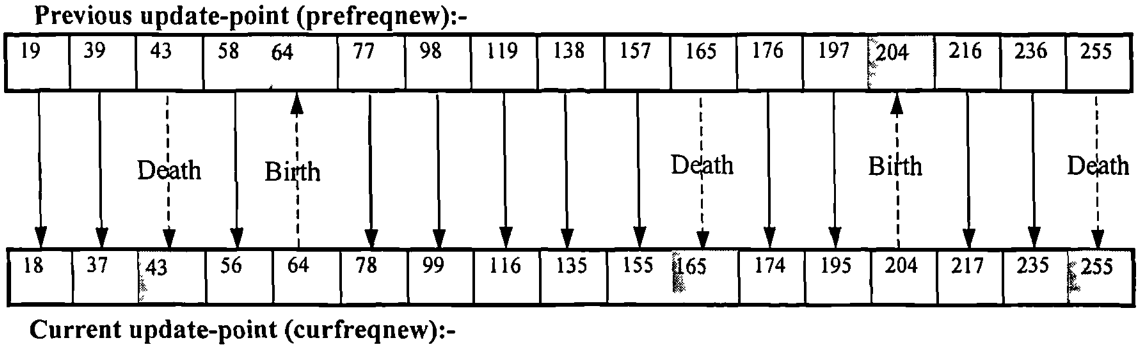
The concept of the nearest neighbours frequency matching is illustrated in figure 3.7. The numbers in the two rows labelled prefreq and curfreq are frequencies expressed as 512-point FFT sample indices. For each frequency array there are corresponding arrays of amplitudes and phases. The frequency range is 0 Hz to 4 kHz which corresponds to the sample indices from 0 to 255 corresponding to the frequency range 0 Hz to 4 kHz.

prefreq:-

19	39	43	58	77	98	119	138	157	165	176	197	216	236	255

curfreq:-

18	37	56	64	78	99	116	135	155	174	195	204	217	235



Previous update-point (prefreqnew):-

19	39	43	58 64	77	98	119	138	157	165	176	197	204	216	236	255

18	37	43	56	64	78	99	116	135	155 165	174	195	204	217	235	255

Current update-point (curfreqnew):-

Fig. 3.7 Illustration of nearest neighbours frequency matching

The matching method produces two new frequency arrays, one for the previous update-point and one for the current update-point. The new arrays are called prefreqnew and curfreqnew. The arrays have the same number of elements and elements with the same index are now matched; i.e. prefreqnew[1] is matched to curfreqnew[1], etc. For each "new" frequency array there are corresponding "new" arrays of amplitudes and phases. Note that births and deaths have been created and are shown as shaded elements. Interpolation may now be applied to the parameters of a sinusoid derived from each pair of elements.

The "nearest neighbours" method does not try to track the frequency changes of pitch-harmonics as the fundamental pitch-frequency changes. Changes in the peak-frequencies plotted against time for a 580 ms segment of mainly voiced speech are illustrated in figure 3.8. This graph was produced by spectrally analysing grossly overlapping frames each of length 150 samples. Frames overlap by 140 samples, i.e.

there are only 10 new samples each consecutive frame. A 1024-point FFT was used to perform the spectral analysis with zero-padding applied to each frame. Peak picking was then applied to each FFT spectrum and plotting the peaks for each frame produced the tracks shown in figure 3.8. The horizontal axis is frequency between 0 to 4 kHz and the vertical axis is time. It can be seen that for a given change in fundamental pitch-frequency, the changes in its harmonics become greater at higher frequencies, i.e. the high order harmonics change more than the low order harmonics.

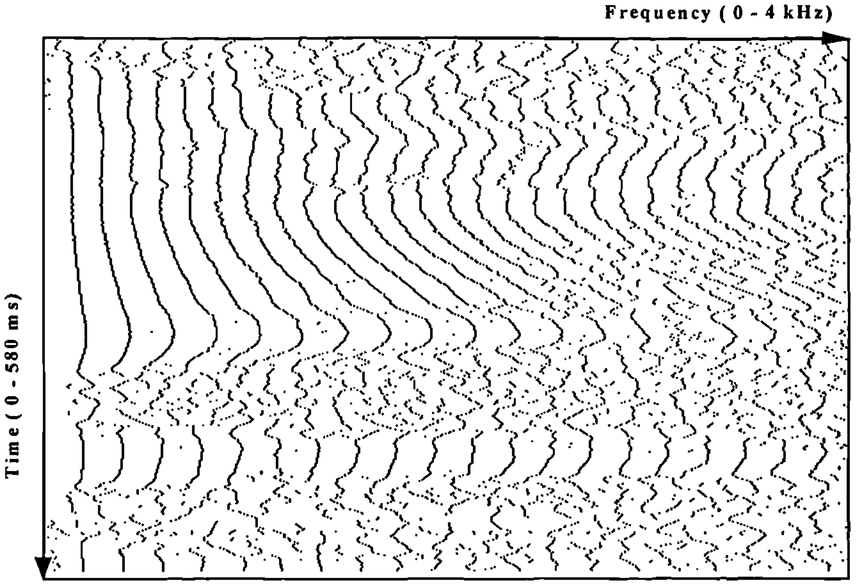


Fig. 3.8 Tracks of peak-frequencies of segments of speech

With the "nearest neighbours" method these high order harmonics are likely to be declared as "births" or "deaths" or inappropriately matched to each other, e.g. the 20th harmonic at the previous update-point may be matched to the 19th harmonic at the current update-point. This will produce an inappropriate and unnatural change of frequency from one update-point to the next. Figure 3.9 illustrates how peak frequencies may be matched between two 20 ms update-points for two voiced segments of speech using the "nearest neighbours" method. The algorithm used to produce this diagram was an exact implementation of McAulay and Quatieri's nearest neighbours matching technique by Pollard [Pol, 95]. The diagram indicates the amplitudes of the previous and current peaks by the lengths of vertical lines. The frequencies were found to be approximately harmonically related in each case with a fundamental frequency of about 220 Hz at the previous update-point increasing to about 230 Hz at the current update-point. Matching is indicated by joining the bases of the vertical lines, and births and deaths are indicated respectively by upward and downward arrow-heads. It may be seen that the matching is appropriate at low frequencies, but becomes less appropriate at higher frequencies. Unnecessary births and deaths occur when the change in frequency for a particular harmonic becomes higher than the threshold, and subsequently, inappropriate matching occurs i.e. the kth harmonic in the previous frame is matched to the (k+1)th harmonic in the current frame. A way of improving the matching in such situations is discussed in following section.

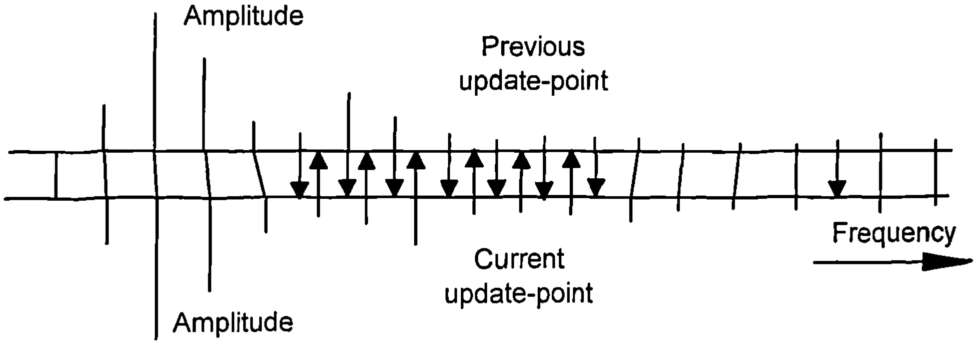


Fig. 3.9 Peak-tracking using "nearest neighbours" matching

3.4.3 "Warped frequency" matching technique

A novel approach [Che, 93] has been applied to the problem of peak tracking and of identifying births and deaths. It was devised to overcome the above-mentioned shortcoming of the "nearest neighbours" matching method and will be referred to as the "warped frequency" matching approach. The approach is to copy the frequencies in array curfreq[1:L] for the current update-point into array warpcurfreq[1:L] with each frequency multiplied by a constant "warp-factor". The choice of this constant, which lies between 0.9 to 1.1, will be discussed later. A frequency matching procedure similar to the "nearest neighbours" method described above is applied to arrays prefreq[1:K] and warpcurfreq[1:L] rather than prefreq[1:K] and curfreq[1:L].

The multiplication or frequency warping reduces or increases the peak frequencies measured for the current update-point by a small mount and is intended to cancel out the effect of any pitch frequency variation from the previous to the current update-points for the purpose of frequency matching. Multiplication by a constant ensures that harmonically related peaks remain harmonically related after frequency warping.

The cancellation of frequency variation is done only for the purpose of frequency matching to allow the progression of pitch-frequency harmonics, particularly the higher frequency ones, to be more easily identified. This approach is particularly effective when the pitch-frequency changes are relatively large. To find the required value of the warp-factor, various possible warping factors are tried, a form of "cross-correlation" index being calculated in each case. This index is the cross-correlation between line magnitude spectra obtained by (a) placing peaks with amplitudes as in preamp[1:K] at frequencies stored in prefreq[1:K] and (b) placing peaks with amplitudes stored as in curamp[1:L] at the warped frequencies stored in warpcurfreq[1:L]. The warping factor which maximises this "cross-correlation" index is taken as the final one. The optimally frequency warped peak-frequencies are then used as a guide during the "peak-tracking" procedure. Birth and death frequencies will be introduced by this procedure at appropriate points. Details of the "warped frequency" matching procedure are as follows:

Phase 1: Define arrays for previous, current and "new" amplitudes, frequencies and phases, as in Phase 1 of the "nearest neighbours" method. Define an additional array warpcurfreq[1:L].

Phase 2:

2.1 Set a warping factor equal to 0.9.

2.2 Calculate array warpcurfreq[1:L] by multiplying the frequencies in curfreq[1:L] by the warp-factor.

2.3 Calculate the "cross-correlation" index, i.e. the cross-correlation between a line magnitude spectrum with lines of amplitudes as in array preamp[1:K] at frequencies as in array prefreq[1:K] and a spectrum with amplitudes as in array curamp[1:L] located at the frequencies in warpcurfreq[1:L].

2.4 Increase the warping factor by 0.01 and go back to step 2.2, until the warping factor exceeds 1.1.

2.5 Take the value of the warping factor and the corresponding array warpcurfreq[1:L] to be the ones that maximise the cross-correlation index.

Phase 3: Apply the "nearest neighbours" matching procedure as described in Section 3.4.1 with curfreq[1:L] replaced by warpcurfreq[1:L] for the matching. When peaks have been matched, however, the array curfreqnew is loaded from the appropriate element of array curfreq rather than array warcurpfreq.

Figure 3.10 shows the same example given in figure 3.9, but using the warped frequency matching technique. As expected, it can be seen that the pitch-frequency and its harmonics are now correctly matched from one update-point to the next.

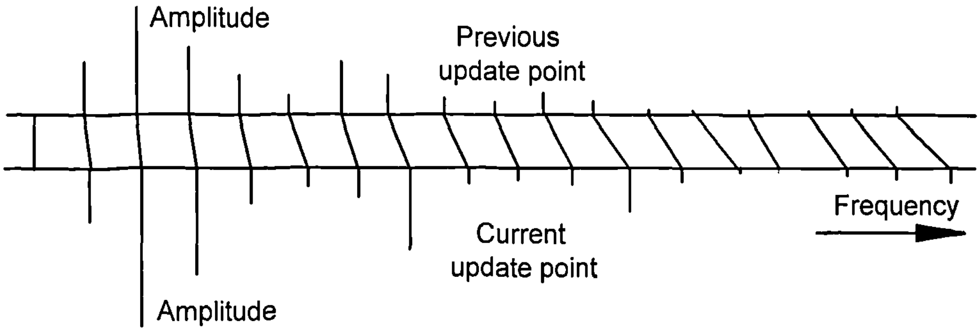
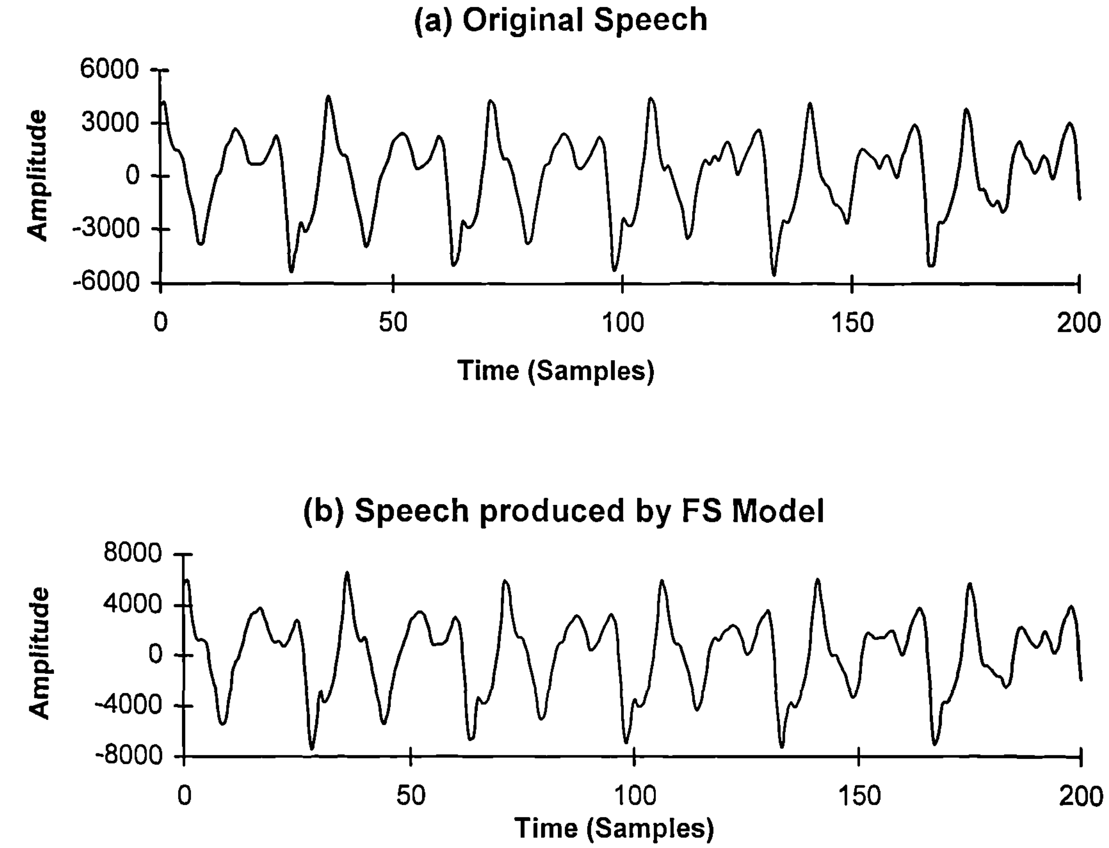


Fig. 3.10 Peak-tracking using "warped frequency" matching

3.5 Reconstruction

Once the "frequency matching" has been achieved, the cubic phase interpolation techniques published by McAulay & Quatieri [McQ, 86A] may be used to produce a smoothly changing instantaneous phase for each of the synthesising sinusoids between two update-points. Figure 3.11 shows 200 samples of voiced speech for both the original and synthetic speech. It is a male utterance of /tə/ in the word “Operator”. The synthetic speech is updated at intervals of 20 ms (160 samples). It can be seen the synthetic speech waveform is very similar to that of the original.


(a) Original Speech

(b) Speech produced by FS Model

Fig. 3.11 A segment of male speech waveform.

(a) original (b) synthetic with 20 ms update-points

3.6 Overlap-and-add method

3.6.1 Overlap-and-add method by summing sinusoids

As an alternative to the cubic interpolation technique for applications where this is computationally too expensive, e.g. when the speech coder is implemented on a DSP microprocessor, an "overlap-and-add" method can be applied to generate each frame of synthetic speech [McQ, 88] [McA, 89]. This method has been found to produce a similar speech quality to that obtained with cubic interpolation when the update interval is reasonably small (usually less than 10 ms).

The principle of the overlap-and-add method is to produce, for each update-point, a set of sinewaves each having constant amplitude, frequency and phase. These sinewaves are generated over a window of two synthesis frame-lengths centred on the current update-point, i. e. starting at the previous update-point and extending through the current update-point to the next. They are summed to produce a signal with constant amplitudes, frequencies and phases. A symmetrically tapering window such as a triangular window of twice the synthesis frame-length and centred on the current update-point, is applied to the samples thus produced. The similarly windowed samples obtained for the previous update-point will also be available at this stage. The later half of the previous '2-frame length window' will overlap in time with the earlier part of the current 2-frame window and corresponding samples are summed to obtain frames of synthesised speech. The later part of the double frame for the current update-point will be added to the earlier part for the next update-point when it is available.

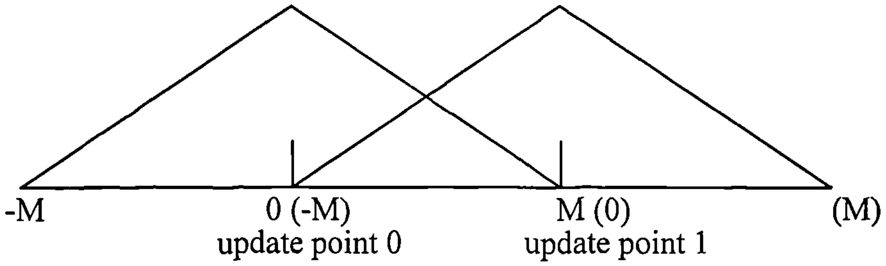


Fig. 3.12 Overlap M samples with triangular windows

An illustration of the arrangement of "overlap-and-add" windows is shown in figure 3.12. A frame of M samples of speech is to be generated between two update-points spaced M samples apart. For each update-point, a 2M samples frame of windowed signal is generated, starting M samples before the update-point and ending M samples after the update-point. For Update-point 0 as shown in figure 3.12, the signal generated is:

$$s_0[n] = \sum_{k=1}^{L_0} A_{0k} \cos(\omega_{0k} n + \phi_{0k}) \qquad \text{for } n = \text{-M to M-1}, \qquad (3.11)$$

where L_0 is the number of harmonics and A_{0k}, ω_{0k}, ϕ_{0k} are amplitudes, frequencies and phases at Update-point 0.

For Update-point 1, the signal generated is:

$$s_1[n] = \sum_{k=1}^{k=L_1} A_{1k} \cos(\omega_{1k} n + \phi_{1k}) \qquad \text{for } n = \text{-M to M-1}, \qquad (3.12)$$

where L_1 is the number of harmonics and A_{1k}, ω_{1k}, ϕ_{1k} are the amplitudes, frequencies and phases at Update-point 1.

So, the synthetic speech frame from Update-point 0 to Update-point 1 is reconstructed as follow:

$$s[n] = s_0[n]\, w[n] + s_1[n-M]\, w[n-M] \qquad \text{for } n = 0 \text{ to M-1}, \qquad (3.13)$$

The window sequence {w[n]} should satisfy the condition that w[n]+w[n-M] =1 for all n in the range 0 to M-1. This condition ensures that synthesised speech has been equally weighted for all samples through the synthesis frame. It is satisfied by a triangular window, a "trapezoidal" window as used by IMBE, a Hann window and other windows. The formula for a triangular window is:-

$$w[n] = 1 - \left|\frac{n}{M}\right| \qquad \text{for } -M \le n < M \qquad (3.14)$$

and since

$$w[n-M] = 1 - \left|\frac{n}{M} - 1\right| = \frac{n}{M} \qquad \text{for } -M \le n < M \qquad (3.15)$$

$$w[n-M] = \left|\frac{n}{M}\right| \qquad \text{for } 0 \le n < M \qquad (3.16)$$

it follows that the condition is satisfied for n in the range of 0 to M-1. The formula for a Hann window is:-

$$w[n] = 0.5 - 0.5\cos(2\pi(n+M)/2M) \qquad \text{for } -M \le n < M \qquad (3.17)$$

and

$$w[n-M] = 0.5 - 0.5\cos(2\pi n/2M) \qquad (3.18)$$

Therefore, for any value of n:

$$\begin{aligned} & w[n] + w[n-M] \\ & = 1 - 0.5\left[\cos(2\pi(n+M)/2M) + \cos(2\pi n/2M)\right] \\ & = 1 - 0.5\left[\cos(\pi n/M + \pi) + \cos(\pi n/M)\right] \\ & = 1 - 0.5\left[-\cos(\pi n/M) + \cos(\pi n/M)\right] \\ & = 1 \end{aligned} \qquad (3.19)$$

It follows that the condition is also satisfied by a Hann window.

Note that the sequences for Update-point 0 and 1 are respectively $\{s_0[n]\}_{-M,\,M-1}$ and $\{s_1[n]\}_{-M,\,M-1}$. The resulting synthetic speech sequence from update-points 0 to 1 is $\{s[n]\}_{0,\,M-1}$.

3.6.2 Overlap-and-add method using inverse FFT

As mentioned above, the double frames required by the overlap-and-add method may be synthesised by generating and summing individual sinusoids sample-by-sample. When the number of sinusoids are relatively large, an alternative and computationally more efficient approach may be used. This is to apply an inverse FFT to an array of frequency-domain samples constructed with non-zero values placed at the frequencies of the spectral peaks, all other frequency domain samples being set to zero. The amplitudes and phases of these non-zero values are made equal to the amplitudes and phases of the sinusoidal model. The length of the inverse FFT needs to be at least twice that of the synthetic speech frame. Then the first half of the windowed time sequence after inverse FFT at the current update-point overlaps with the second half of the windowed time sequence obtained for the previous update-point.

Experiments were carried out to evaluate the overlap-and-add method by synthesising speech from the sinusoidal model described in this chapter using the overlap-and-add method and comparing it with what was obtained by the cubic interpolation method. It was found that the speech quality was perceptually indistinguishable for both methods when update-points were placed at 10 ms intervals. However, the synthetic speech using the overlap-and-add method with 20 ms update-points sounded a little "rough". For 20 ms update-points, 40 ms synthetic speech has to be produced at each update-point which is too long a period for changes of the vocal tract and vocal cord activity to be considered negligible. The stationary assumption for the speech is no longer valid.

3.7 Implementation

Two versions of the FS model were implemented and tested. The first version was essentially as published by McAulay & Quatieri [McQ, 86A]. It represents an FS model by peak frequencies, amplitudes and phases derived from the original speech spectrum. The second version, which is based on a variation of McAulay & Quatieri's technique, termed SWELP (sine-wave excited linear prediction) by Yeldener, Kondoz & Evans [YKE, 90], produces an FS model of the LPC residual signal and passes it through an all-pole LPC synthesis filter with smoothly interpolated coefficients. Details of the implementation of the first version will be given in this section. Using this version of the FS model, some tests were carried out to investigate its performance. These tests include changing the update interval, randomising phases at high frequency and using different synthesis methods.

3.7.1 Software

A program, "FSMODEL.C" [SC, 93] was developed to evaluate the first version of the FS model. It is written in C++. Figure 3.14 shows the flow chart of this program.

The program reads non-overlapping input frames (160 samples per input frame) of 8 kHz sampled speech. At any given time, four complete input frames, i.e. 640 samples, are held available in an array. The beginning of the second frame is assumed to be the current update-point. Therefore the newest two input frames are "look-ahead" frames which are required for eliminating pitch-frequency extraction errors and for ensuring that a smooth and natural pitch frequency contour occurs. The arrangement of the four input frames is shown diagrammatically in figure 3.13.

A pitch detection algorithm based on the initial IMBE pitch detector [MSDI, 91] is applied to estimate the pitch-frequency in the vicinity of the current update-point by analysing three 281 sample frames, centred on the current update-point, 160 samples ahead of the current update-point, and 320 samples ahead. The pitch frequencies obtained for the two previous update-points are also taken into account when deciding, from the results of the analyses, what the pitch-frequency should be. Full details of the IMBE pitch-detector are given in the reference [MSDI, 91].

A short term speech spectrum is produced for the current update-point by applying a 512 point FFT to a 2.5 pitch-period Hamming windowed segment of speech which is centred on the update-point and zero-padded as described in Section 3.2. It is arranged that for unvoiced speech the window length is fixed at 20 ms. The energy density spectrum with a dB scale for energy is then calculated. A peak-picking algorithm is applied to find the frequencies of the peaks in the short-term energy density spectrum that are likely to correspond to pitch-frequency harmonics for voiced speech.

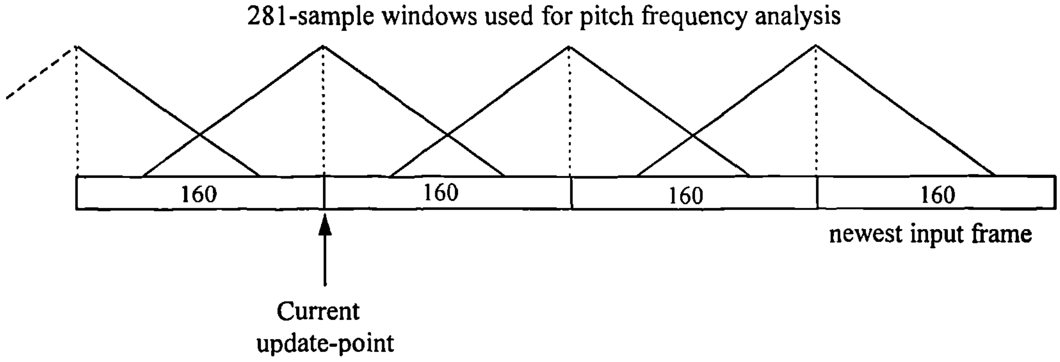


Fig. 3.13 Four input-frame buffers

For unvoiced speech, the peaks are likely to be randomly distributed. A threshold is set to be 80 dB below the maximum energy density over the 256 samples of the Nyquist frequency range. All peaks with energy density above the threshold are located. If the number of such peaks exceeds 80, as could happen for unvoiced speech, the smallest peaks are discarded until only 80 remain. Once the frequency of the peaks have been identified, their corresponding amplitudes and frequencies are obtained. Also, the phases at the frequencies of these peaks are derived from the FFT phase spectrum.

The following parameters represent the FS model at each update-point:-

- The number of peaks (L)
- Peak-frequencies: $\omega_1, \omega_2, ..., \omega_L$ radians/sample
- Peak-amplitudes: $A_1, A_2, ..., A_L$ volts
- Peak-phases: $\phi_1, \phi_2, ..., \phi_L$ radians

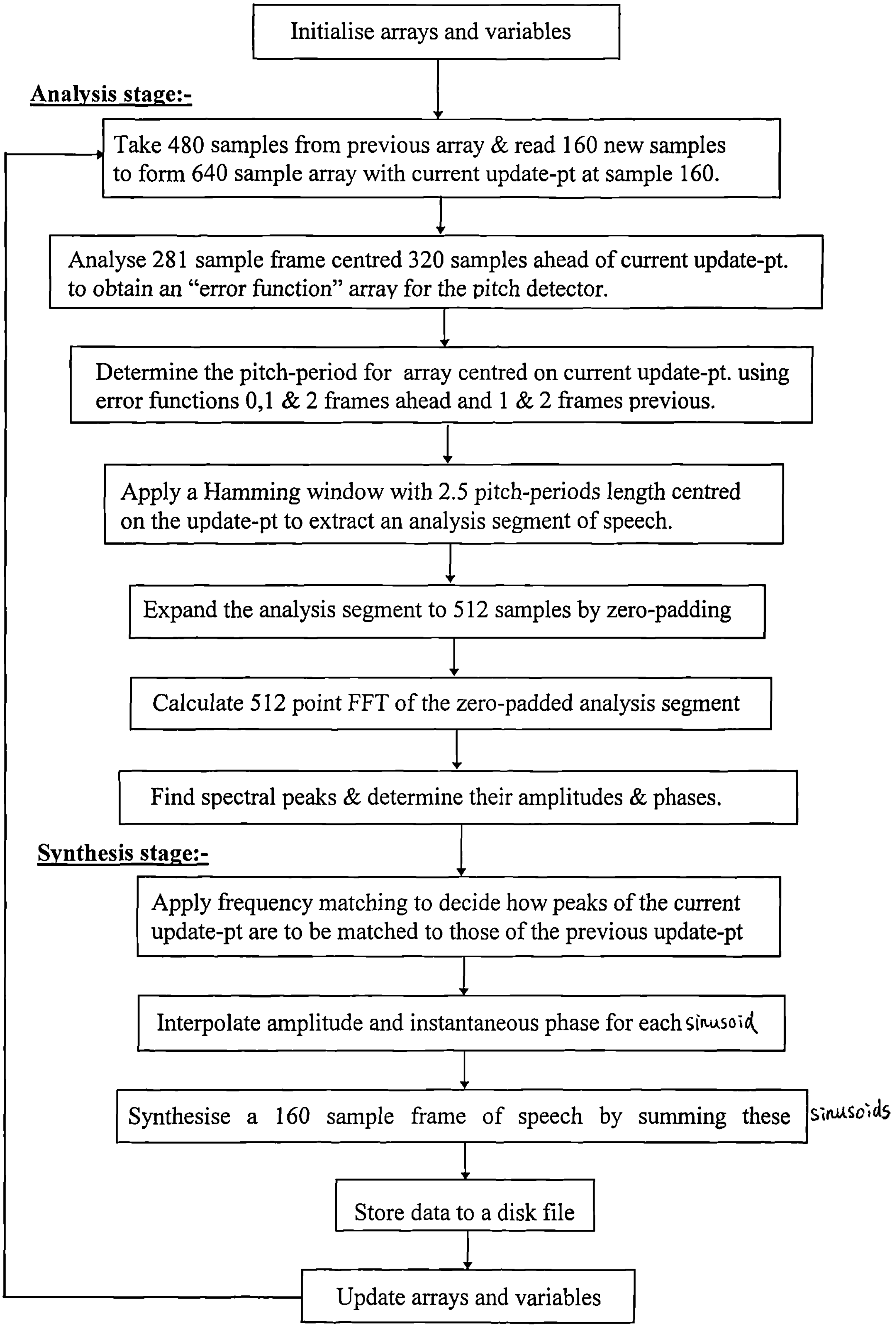

Initialise arrays and variables
Analysis stage:-
Take 480 samples from previous array & read 160 new samples to form 640 sample array with current update-pt at sample 160.
Analyse 281 sample frame centred 320 samples ahead of current update-pt. to obtain an "error function" array for the pitch detector.
Determine the pitch-period for array centred on current update-pt. using error functions 0,1 & 2 frames ahead and 1 & 2 frames previous.
Apply a Hamming window with 2.5 pitch-periods length centred on the update-pt to extract an analysis segment of speech.
Expand the analysis segment to 512 samples by zero-padding
Calculate 512 point FFT of the zero-padded analysis segment
Find spectral peaks & determine their amplitudes & phases.
Synthesis stage:-
Apply frequency matching to decide how peaks of the current update-pt are to be matched to those of the previous update-pt
Interpolate amplitude and instantaneous phase for each sinusoid
Synthesise a 160 sample frame of speech by summing these sinusoids
Store data to a disk file
Update arrays and variables

Fig. 3.14 Flow chart for the program "FSMODEL.C"

To synthesise a speech frame of length N between two update-points, the following generalised Fourier series formula is used:

$$s[n] = \sum_{\ell=1}^{L} A_\ell[n]\cos(\sigma_\ell[n]) \qquad \text{for } n = 0 \text{ to } N-1. \tag{3.20}$$

$A_\ell[n]$ and $\sigma_\ell[n]$ are interpolated as follows:

$$A_\ell[n] = A_{\ell 0} + (A_{\ell 1} - A_{\ell 0})\, n / N \tag{3.21}$$

$$\sigma_\ell[n] = \phi_{\ell 0} + \omega_{\ell 0}\, n + \alpha[\ell]\, n^2 + \beta[\ell]\, n^3 \tag{3.22}$$

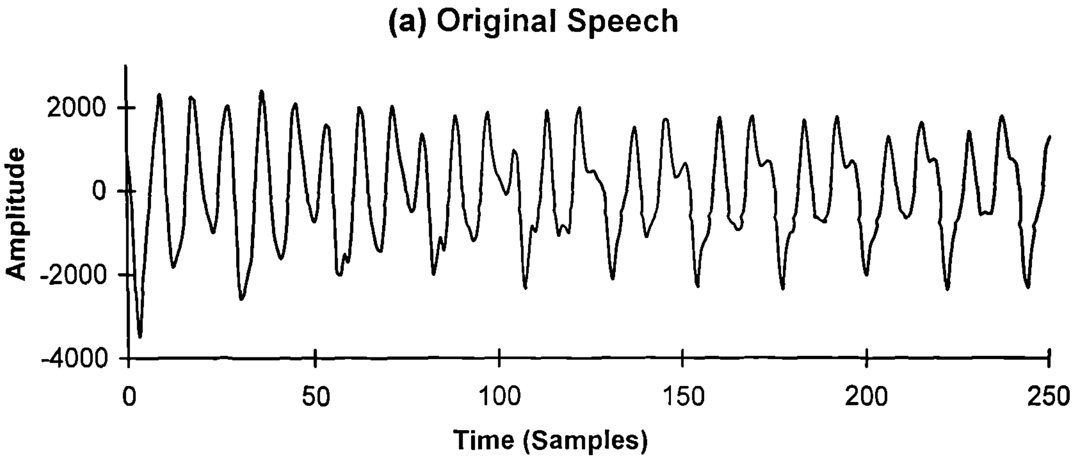
where $A_{\ell 0}$ and $A_{\ell 1}$ are the amplitude of ℓ^{th} sinusoid for the update-points at the beginning and the end of the synthesis frame respectively. $\phi_{\ell 0}$ and $\omega_{\ell 0}$ are the phase and the frequency respectively of the ℓ^{th} sinusoid at the previous update-point at the beginning of the synthesis frame. $\alpha[\ell]$ and $\beta[\ell]$ must be calculated from equations 2.21 and 2.22 using $\phi_{\ell 1}$ and $\omega_{\ell 1}$ which are the phase and frequency respectively of ℓ^{th} sinusoid at the current update-point at the end of the synthesis frame. By substituting equations 3.21 and 3.22 into equation 3.20, a frame of synthetic speech s[n] may be obtained.

3.7.2 Results

Experiments were carried out to evaluate the implementation of the sinusoidal model described above. These experiments anticipate the adaptation of the model to low bit-rate speech coding which is the subject of future chapters in this thesis. There is as yet no attempt to reduce the data rate. The aim at this stage is to verify that the model is a suitable basis for data compression and to establish which aspects of the model are likely to be critical or less critical to maintaining high speech quality. Speech was produced with different update intervals and variation in the synthesis methods used for both the magnitudes and phases of the component sinusoids. The experiments are summarised below:

- FS model with 10 ms update-points using cubic interpolation to interpolate instantaneous phase
- FS model with 10 ms update-points using quadratic interpolation to interpolate instantaneous phase
- FS model with 20 ms update-points using cubic interpolation to interpolate instantaneous phase
- FS model with 20 ms update-points using quadratic interpolation to interpolate instantaneous phase
- FS model with 10 ms update-points using overlap-and-add method to generate synthetic speech
- FS model with phases of the sinusoids replaced by uniformly distributed random values between - π to π at frequencies above 3 kHz.

Informal listening tests showed that the FS model with 10 ms update-points and cubic interpolation, produces very good synthetic speech which is essentially indistinguishable from the original. For 20 ms update-points, using cubic interpolation, the quality of the synthetic speech is very good for male speech, but some distortion for female speech is audible. The distortion for female speech may also be seen by comparing their waveforms. A segment of female speech, which is an utterance /je/ in the word "Yes", is shown in figure 3.15. The waveform produced by the 10 ms updated version is very close to that of the original whereas the waveform produced by the 20 ms updated version is clearly distorted.



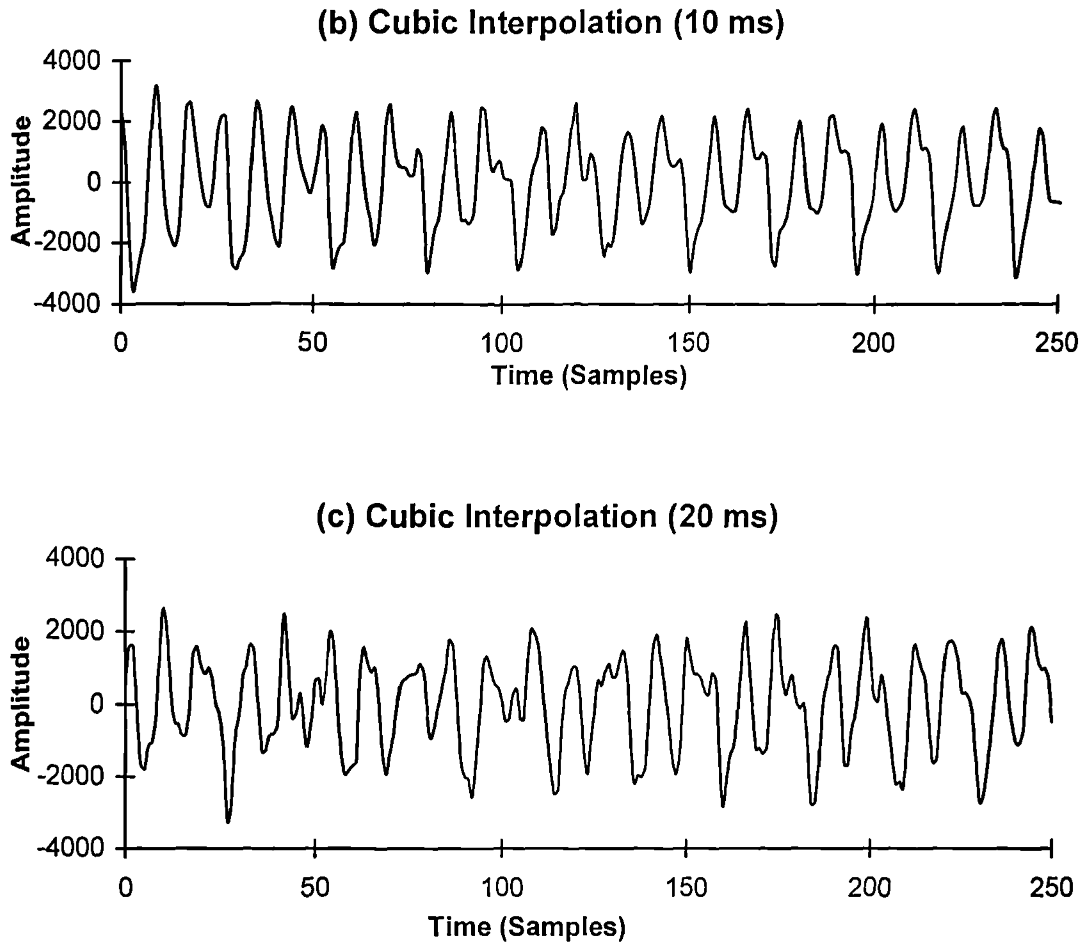


Fig. 3.15 Effect of 10 ms and 20 ms update-points for segment of female speech

The speech quality obtained using quadratic rather than cubic interpolation for both 10 ms and 20 ms update intervals was also quite good, but the degradation was clearly audible due to the loss of phase information.

As mentioned in Section 3.6, the speech quality obtained from the overlap-and-add method with 10 ms update intervals was very similar to that obtained with cubic interpolation with 10 ms update intervals. There was also no perceptual loss of synthetic speech quality when the speech was reconstructed using random phase rather than the true phase at frequencies above 3 kHz. This random phase experiment was suggested by Choi [Cho, 96] based on observations by other authors [e.g. McPQS, 91]. The result will be useful as it suggests a means of achieving some data reduction in the parameters of a low bit-rate sinusoidal speech coder.

3.8 Conclusion

The FS model of McAulay and Quatieri [McQ, 86A] has been implemented with some modifications. Evaluations have been carried out which confirm that it is capable of producing good quality speech which is essentially indistinguishable from the original for update-points 10 ms apart. With 20 ms update intervals, distortion was found in female speech. The FS model uses cubic interpolation, though when the update interval is 10 ms it had been found that the computationally simpler overlap-and-add method does not produce audible deterioration in quality. It has also been found that the phases at frequencies above 3 kHz can be set randomly without degrading the perceived speech quality. Experiments have indicated that the perceived speech quality is better using cubic interpolation than using quadratic interpolation, which confirms that phase information has a significant effect on the perceived quality of sinusoidally modelled speech. A “warped frequency” matching technique has been found to operate more effectively than “nearest neighbours” techniques.

Although it would not be possible to encode directly the amplitudes, frequencies and phases of all spectral peaks at the bit-rates required for low bit-rate speech coding, the FS model is a starting point and a useful basis for our understanding of the effect of low bit-rate coding techniques. By converting original speech to the parameters of a sinusoidal model its complexity may be reduced without greatly sacrificing quality. This conversion may be the first stage of a low bit-rate coding process. Further processing will then be required to reduce the bit-rate needed to represent the parameters of the model. This is the subject of the next chapter.

Chapter 4
Low bit-rate Sinusoidal Transform Coder

4.1 Introduction

Sinusoidal Transform Coding (STC) [McQ, 87] [McC, 90] [McQ, 92] is based on the idea that natural sounding telephone band-width speech can be produced by adding together sine waves whose amplitudes, frequencies and phases vary with the changing nature of the voice and the spoken utterance. As described in the last chapter, the FS model is able to produce telephone bandwidth speech that is barely distinguishable from the original speech. The model has many applications in its own right, e.g. time-scale and pitch-scale modification [QMc, 86] [QMc, 92]. For coding, it is useful as a starting point since it is much simpler than the original speech. However further techniques must be applied to this representation if it is to become the basis of a low or very low bit-rate coding scheme since there are too many parameters in the FS model to be encoded directly.

To achieve the low bit-rate representation of the FS model known as STC, some of the flexibility of the original model must be sacrificed. Firstly, it is assumed that the instantaneous frequencies of the sine-waves are harmonically related at the update-points. Secondly, explicit specification of the phases is considered not to be economic, and thirdly non-waveform based techniques are used to synthesise unvoiced speech and transition frames.

The first aspect means that instead of specifying a range of sinusoidal frequencies at each update-point, a pitch-period estimate is given. Pitch-period estimation is a crucial aspect of most low bit-rate coders, and the frequency domain approach [McQ, 90] used by STC, based on the FS model, has much to recommend it.

The second aspect, i.e. the treatment of the phases, constitutes one of the most interesting differences between STC and other sinusoidal coding techniques such as MBE and PWI. The STC approach is to regenerate the phases from magnitude only information on the basis of a minimum phase assumption. Phase information is not encoded and is derived at the decoder by adding together a linear phase component to achieve an appropriate delay and a minimum phase component derived from the spectral envelope, assuming a vocal system function which is minimum phase.

The third aspect leads to a voicing probability parameter to be introduced later, from which a "cut-off frequency" is derived. This variable cut-off frequency divides the frequency band of the speech into two sub-bands: a low frequency sub-band and a high frequency sub-band. The frequencies of the sinusoids used to synthesise the lower frequency sub-band are entirely harmonically related to the pitch-frequency. The frequencies of the sinusoids in the high frequency sub-band are fixed at 100 Hz apart, the instantaneous phase for each of these sinusoids at each update-point being taken as a uniformly distributed random number between $-\pi$ to π.

The amplitudes of higher frequency sub-band sinusoids are obtained from sampling the envelope at the frequencies of the sinusoids. Between the update-points, the amplitudes and instantaneous phases of the higher subband sinusoids are interpolated as in the lower sub-band. For fully voiced speech the cut-off frequency will be set to half the sampling frequency and the lower frequency sub-band will coincide with the entire speech band. For unvoiced speech or transitions the cut-off frequency will be reduced so that the randomised signal generated in the upper band will constitute an appropriately significant part of the synthetic speech.

Instead of encoding the amplitudes of the sinusoids directly, they are used to form a spectral envelope by a smoothing (or interpolation) technique such as cubic spline interpolation. The amplitudes of the sinusoidal model, at the decoder, are obtained by sampling the envelope at harmonics of the pitch-frequency. The spectral envelope may be represented by a small number of parameters such as cepstral coefficients or line spectral frequencies (LSFs) [McCQ, 93]. These parameters along with a pitch-period measurement and a "voicing probability" constant are efficiently encoded as a low bit-rate representation of the speech at each update-point.

4.2 Low bit-rate modelling

The principle of the low bit-rate STC coder is to take speech as a glottal excitation signal passing through a vocal tract filter which is a linear time invariant system in the short term. The excitation signal may be modelled as the sum of sinusoids whose amplitudes and instantaneous phases are interpolated between update-points.

It is thought that in the context of the sinusoidal model, a pitch pulse occurs when all of the sinewaves add coherently, i.e. when all their instantaneous phases are equal to zero or some integer multiple of 2π. Where all the sinusoids are harmonically related and fixed in frequency (as for a conventional Fourier series as would be obtained if the pitch-frequency remained constant) this can only occur if their instantaneous phases are proportional to frequency at the update-point. This means also that the instantaneous phases will remain proportional to frequency for each harmonic throughout the synthesis frame. In this case, assuming amplitudes of unity, the glottal excitation waveform can be modelled as follows:

$$e[n] = \sum_{\ell=1}^{L} \cos([n - n_0]\omega_\ell) \quad \text{for } n = 0, 1, ..., N\text{-}1 \tag{4.1}$$

or

$$e[n] = \sum_{\ell=1}^{L} \cos(n\omega_\ell + \phi_\ell) \tag{4.2}$$

where $\phi_\ell = -n_0\omega_\ell$ is a linear phase component of the instantaneous phase for each sinusoid, n_0 is called the onset time and ω_l is the frequency of each sinusoid. It is convenient to write:

$$e[n] = \mathrm{Re}\left(\sum_{\ell=1}^{L} \exp[jn\omega_\ell + j\phi_\ell]\right) \tag{4.3}$$

Let the transfer function of the vocal system be $H_s(\omega)$, and

$$H_s(\omega) = A_s(\omega)\exp[j\phi_s(\omega)] \tag{4.4}$$

where $A_s(\omega)$ is the gain and $\phi_s(\omega)$ is the phase response. Then for a given value of n_0, a frame of reconstructed speech $\hat{s}[n,n_0]$ can be written as follows:

$$\hat{s}[n,n_0] = \mathrm{Re}\left(\sum_{\ell=1}^{L} A_s(\omega_\ell)\exp[jn\omega_\ell + j\phi_\ell + j\phi_s(\omega_\ell)]\right) \tag{4.5}$$

i.e.

$$\hat{s}[n,n_0] = \sum_{\ell=1}^{L} A_s(\omega_\ell)\cos\left(n\omega_\ell + \hat{\theta}_\ell\right) \tag{4.6}$$

where

$$\hat{\theta}_\ell = \phi_s(\omega_\ell) + \phi_\ell = \phi_s(\omega_\ell) - n_0\omega_\ell \tag{4.7}$$

Assume now that the original speech s[n] can be sinusoidally modelled by the FS model:

$$s[n] = \mathrm{Re}\left(\sum_{\ell=1}^{L} A_\ell[n]\exp[j(n\omega_\ell + \theta_\ell[n])]\right) \tag{4.8}$$

where A_ℓ, ω_ℓ and θ_ℓ are the amplitude, frequency and phase offset respectively for each ℓ, as would be measured from the peak locations of the FFT spectrum of the speech. Since low bit-rate speech coding can not afford to encode the true measured phase offsets, θ_ℓ for ℓ = 1, 2, ..., L, we instead derive approximations to them based on equation 4.7. To enable this, an onset time n_0 is derived at the encoder, as will be described later. The system phase $\phi_s(\omega)$ is not encoded, the idea being to derive it from the magnitude spectrum at the decoder under a minimum phase assumption.

A set of cepstral coefficients obtained from the spectral envelope are encoded to represent the short term spectrum thus allowing the amplitude $A_s(\omega)$ and the phase $\phi_s(\omega)$ of the transfer function $H_s(\omega)$ to be determined at each ω_ℓ. Different ways of interpolating between peaks in the magnitude spectrum to determine a spectral envelope will be discussed in this chapter.

To achieve a low bit-rate representation, it is assumed that the frequencies of the peaks are harmonically related for voiced speech. For unvoiced and partially unvoiced speech, the concept of voicing probability as defined earlier is used.
Before discussing the details of this coder, the complex cepstrum and the minimum phase assumption must be introduced.

4.3 Complex cepstrum

This section introduces the complex cepstrum of a digital signal, and surveys some of its properties and applications [OS, 75] [SC, 94A]. The complex cepstrum is widely used in speech coding, synthesis and recognition, and has recently been applied to sinusoidal speech coding at very low bit-rates. It may be used for pitch-period detection and also as an alternative to LPC analysis for obtaining short-term spectral parameters. A problem that arises is that of deriving the phase spectrum from the magnitude spectral envelope of the vocal tract frequency response. This can be done via the complex cepstrum assuming the vocal system transfer function to be minimum phase. A system function in z-transform form is minimum phase if it has no poles or zeros outside or on the unit circle.

The complex cepstrum is defined as the sequence obtained by taking the inverse discrete time Fourier transform (inverse DTFT) of the natural logarithm of the frequency response of a system. The frequency response is in general complex. Its natural logarithm is also generally complex but the inverse DTFT of this function of ω, i.e. the complex cepstrum, is normally real when the impulse response is real as will be seen.

Given a discrete time linear time-invariant (LTI) system whose impulse response is {h[n]}, its frequency response is the DTFT of {h[n]}, i.e.

$$H\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} h[n]e^{-jn\omega} \qquad (4.9)$$

In polar form, this is expressed as

$$H\left(e^{j\omega}\right) = G(\omega)e^{j\,\phi(\omega)} \tag{4.10}$$

The complex cepstrum of the LTI system is a sequence $\left\{\hat{h}[n]\right\}$ which is the inverse DTFT of $\log_e\left(H\left(e^{j\omega}\right)\right)$, i.e.

$$\hat{h}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi}\log_e\left(H\left(e^{j\omega}\right)\right)e^{jn\omega}d\omega \qquad \text{for } -\infty < n < \infty \tag{4.11}$$

By equation 4.10,

$$\hat{h}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\log_e(G(\omega)) + j\phi\,(\omega)\right]e^{jn\omega}d\omega \tag{4.12}$$

As $\{\mathrm{h}[\mathrm{n}]\}$ is real, G(-ω) = G(ω) for all ω, and if $\phi\;(\omega) = 0$ when ω = 0, we can assume that $\phi\;(-\omega) = -\phi\;(\omega)$ for all ω. It follows that:

$$\hat{h}[n] = \frac{1}{2\pi}\int_0^{\pi}\left[\log_e(G(\omega)) + j\phi\,(\omega)\right]e^{jn\omega}d\omega \;+\frac{1}{2\pi}\int_{-\pi}^{0}\left[\log_e(G(\omega)) + j\phi\,(\omega)\right]e^{jn\omega}d\omega$$

$$= \frac{1}{2\pi}\int_0^{\pi}\left[\log_e(G(\omega)) + j\phi\,(\omega)\right]e^{jn\omega}d\omega \;+\frac{1}{2\pi}\int_0^{\pi}\left[\log_e(G(\omega)) - j\phi\,(\omega)\right]e^{-jn\omega}d\omega$$

$$= \frac{1}{2\pi}\int_0^{\pi}\log_e(G(\omega))\left[e^{jn\omega} + e^{-jn\omega}\right]d\omega \;+\frac{j}{2\pi}\int_0^{\pi}\phi\,(\omega)\left[e^{jn\omega} - e^{-jn\omega}\right]d\omega$$

$$= \frac{1}{\pi}\int_0^{\pi}\left[\log_e G(\omega)\cos(n\omega) - \phi\,(\omega)\sin(n\omega)\right]d\,\omega \tag{4.13}$$

We notice that $\hat{h}[n]$ is real for all n in the range -∞ to ∞, and

$$\hat{h}[-n] = \frac{1}{\pi}\int_0^{\pi}\left[\log_e G(\omega)\cos(n\omega) + \phi\;(\omega)\sin(n\omega)\right]d\omega \qquad \text{for all n} \tag{4.14}$$

Defining $$c[n] = \frac{\hat{h}[n] + \hat{h}[-n]}{2} \qquad \text{for } -\infty < \text{n} < \infty \tag{4.15}$$

from equations 4.13 and 4.14 we obtain:

$$c[n] = \frac{1}{\pi}\int_0^{\pi}\log_e G(\omega)\cos(n\omega)d\omega \qquad \text{for all n} \tag{4.16}$$

which can be rewritten as:

$$c[n] = \frac{1}{2\pi}\int_0^{\pi}\log_e G(\omega)\cos(n\omega)d\omega + \frac{1}{2\pi}\int_0^{\pi}\log_e G(\omega)\cos(n\omega)d\omega$$

$$= \frac{1}{2\pi}\int_0^{\pi}\log_e G(\omega)\cos(n\omega)d\omega + \frac{1}{2\pi}\int_{-\pi}^{0}\log_e G(\omega)\cos(n\omega)d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e G(\omega) e^{jn\omega} d\omega \qquad \text{for all n} \qquad (4.17)$$

Therefore, the sequence {c[n]} is the inverse DTFT of the logarithm of the gain response G(ω). It is a non-causal symmetric sequence, i.e. c[-n] = c[n] for all n, and is usually referred to as the "real cepstrum", or simply as the "cepstrum".

Similarly defining $$d[n] = \frac{\hat{h}[n] - \hat{h}[-n]}{2} \qquad \text{for all n}, \qquad (4.18)$$

then $$d[n] = -\frac{1}{\pi}\int_{0}^{\pi} \phi(\omega)\sin(n\omega) d\omega \qquad \text{for all n}, \qquad (4.19)$$

which can also be rewritten as:

$$\text{d[n]} = \frac{1}{2\pi}\int_{0}^{\pi} -\phi(\omega)\sin(n\omega) d\omega - \frac{1}{2\pi}\int_{-\pi}^{0} -\phi(\omega)\sin(n\omega) d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} j\phi(\omega) e^{jn\omega} d\omega \qquad \text{for all n} \qquad (4.20)$$

So the sequence {d[n]} is the inverse DTFT of $j\phi(\omega)$. It is a non-causal asymmetric sequence, i.e. d[-n] = -d[n] for all n. Clearly, the imaginary part of the DTFT of $\{d[n]\}$ is ϕ(ω).

4.4 Minimum phase assumption

It may be shown that a necessary and sufficient condition for the frequency response $H(e^{j\omega})$ of a causal and stable system to be minimum phase is that its complex cepstrum $\{\hat{h}[n]\}$ is causal [OS, 75]: i.e. $\hat{h}[n] = 0$ for n < 0. Only stable systems are considered in this analysis.

To verify this, let H(z) be the z-transform of {h[n]} and let $\hat{H}(z)$ be the z-transform of $\{\hat{h}[n]\}$ as defined by equation 4.13. H(z) can be expressed in terms of its poles and zeros as:

$$H(z) = \frac{K\prod_{k=1}^{M}\left(1 - z_k z^{-1}\right)\prod_{k=1}^{L}\left(1 - Z_k z^{-1}\right)}{\prod_{k=1}^{P}\left(1 - p_k z^{-1}\right)} \qquad (4.21)$$

where the zeros have been separated into two sets $\{z_k\}_{1,M}$ lying inside unit circle and $\{Z_k\}_{1,L}$ lying outside or on the unit circle. K is a real constant. Now:

$$\hat{H}(z) = \log_e\left(H(z)\right)$$

$$= \log_e K + \sum_{k=1}^{M}\log_e\left(1 - z_k z^{-1}\right) + \sum_{k=1}^{L}\log_e\left(1 - Z_k z^{-1}\right) - \sum_{k=1}^{P}\log_e\left(1 - p_k z^{-1}\right) \qquad (4.22)$$

Since $$\log_e(1 - x) = -\sum_{n=1}^{\infty}\frac{x^n}{n} \qquad \text{for } |x| < 1 \qquad (4.23)$$

when $|z| \geq 1$ we can express:

$$\sum_{k=1}^{M}\log_e\left(1 - z_k z^{-1}\right) = \sum_{k=1}^{M}\left(-\sum_{n=1}^{\infty}\frac{Z_k{}^n}{n}Z^{-n}\right) \qquad (4.24)$$

$$\sum_{k=1}^{P}\log_e\left(1 - p_k z^{-1}\right) = \sum_{k=1}^{P}\left(-\sum_{n=1}^{\infty}\frac{p_k^n}{n}Z^{-n}\right) \qquad (4.25)$$

Since $\left|z_k z^{-1}\right| < 1$ and $\left|p_k z^{-1}\right| < 1$ when $|z| \geq 1$ for poles and zeros inside the unit circle. Hence if there are no zeros outside or on the unit circle, i.e. if L= 0, both series in equation 4.22 converge for $|z| \geq 1$ and we can write:

$$\hat{H}(z) = \log_e K - \sum_{k=1}^{M}\left(\sum_{n=1}^{\infty}\frac{Z_k{}^n}{n}z^{-n}\right) + \sum_{k=1}^{P}\left(\sum_{n=1}^{\infty}\frac{p_k{}^n}{n}z^{-n}\right) \quad \text{for } |z| \geq 1 \qquad (4.26)$$

This is the z-transform of a convergent causal sequence.

$$\hat{h}[n] = \begin{cases} 0 & :n < 0 \\ \log_e K & :n = 0 \\ -\sum_{k=1}^{M}\frac{Z_k{}^n}{n} + \sum_{k=1}^{P}\frac{p_k{}^n}{n} & :n = 1,2,\ldots\infty \end{cases} \qquad (4.27)$$

Hence if all poles and zeros are inside the unit circle, $\left\{\hat{h}[n]\right\}$ is causal. i.e. $\hat{h}[n] = 0$, for n < 0. If H(z) is not minimum phase, then $L \neq 0$ in equation 4.22. Consider the term:

$$\sum_{k=1}^{L} \log_e\left(1 - Z_k z^{-1}\right) \tag{4.28}$$

Since $|Z_k| > 1$, it can not now be guaranteed that $|Z_k z^{-1}| < 1$ for $|z| \geq 1$. However, this expression may be written as:

$$\sum_{k=1}^{L} \left\{\log_e\left((1 - z / Z_k) / (-z / Z_k)\right)\right\}$$

$$= \sum_{k=1}^{L} \log_e(1 - z / Z_k) - \sum_{k=1}^{L} \log_e(-z / Z_k)$$

$$= \sum_{k=1}^{L} \left(-\sum_{n=1}^{\infty} \frac{z^n}{n Z_k^n}\right) - \sum_{k=1}^{L} \log_e(-z / Z_k) \quad \text{for } |z| \geq 1 \tag{4.29}$$

since $|z/Z_k| < 1$ for all $|z| \geq 1$. The first term is the z-transform of the non-causal convergent sequence:

$$h_1[n] = \begin{cases} 0 & : n \geq 0 \\ \displaystyle\sum_{k=1}^{L} \frac{1}{n Z_k^{-n}} & : n < 0 \end{cases} \tag{4.30}$$

Hence a convergent sequence $\left\{\hat{h}[n]\right\}$ will be non-zero for n < 0 if H(z) is not minimum phase; i.e. $\left\{\hat{h}[n]\right\}$ will be non-causal.

4.5 Hilbert transform between gain and phase responses

It may now be shown that the phase response of a minimum phase LTI system may be derived from the gain response and vice versa (apart from the gain at $\omega = 0$). When the complex cepstrum is causal, its coefficients $\hat{h}[n]$, which actually are real, can be deduced from {c[n]} or {d[n]}, apart from $\hat{h}[0]$, which cannot be deduced from d[0] and therefore from $\phi(\omega)$.

If the complex cepstrum is causal, i.e. if $\hat{h}[n] = 0$ for n < 0 then

$$\hat{h}[n] = \begin{cases} 2c[n] & :\mathrm{n} > 0 \\ c[n] & :\mathrm{n} = 0 \\ 0 & :n < 0 \end{cases} \tag{4.31}$$

Also $\hat{h}[n]$=2d[n] for n>0 (4.32)

Vice versa,

$$c[n] = \frac{\hat{h}[n] + \hat{h}[-n]}{2} = \begin{cases} \hat{h}[n]/2 & :n > 0 \\ \hat{h}[n] & :\mathrm{n} = 0 \\ \hat{h}[-n]/2 & :\mathrm{n} < 0 \end{cases} \tag{4.33}$$

$$d[n] = \frac{\hat{h}[n] - \hat{h}[-n]}{2} = \begin{cases} \hat{h}[n]/2 & :n > 0 \\ 0 & :n = 0 \\ -\hat{h}[-n]/2 & :n < 0 \end{cases} \tag{4.34}$$

It follows that:

$$d[n] = \begin{cases} c[n] & :\mathrm{n} > 0 \\ 0 & :\mathrm{n} = 0 \\ -c[n] & :\mathrm{n} < 0 \end{cases} \tag{4.35}$$

We know that the imaginary part of the DTFT of $\{d[n]\}$ is ϕ(ω). Therefore the steps to find ϕ(ω), when G(ω) is given, assuming the system is minimum phase, are as follows:

1. Find {c[n]} by taking inverse DTFT of $\log_e G(\omega)$.
2. Calculate $d[n] = \begin{cases} c[n] & :\mathrm{n} > 0 \\ 0 & :\mathrm{n} = 0 \\ -c[n] & :\mathrm{n} < 0 \end{cases}$
3. Calculate ϕ(ω) which is the imaginary part of the DTFT of {d[n]} for all values of ω.
4. Step 3 assumes that $\phi\ (0) = 0$, and $\phi\ (-\omega) = -\phi\ (\omega)$ for all ω. If the application requires that $\phi\ (0) = \pi$, simply add π to $\phi\ (\omega)$ for all ω ≥ 0 and subtract π from $\phi\ (\omega)$ for all ω < 0.

The minimum phase assumption is implicit in the use of LPC analysis to characterise a vocal system. The phase response of the vocal system can now be deduced from corresponding gain response. LPC analysis is not used to derive the spectral envelope therefore the vocal system is not restricted to be an all-pole model. Instead, the spectral envelope is deduced from the DFT spectrum by interpolating between the peaks. Once this magnitude function is obtained, the phase response may be deduced, under the minimum phase assumption, by methods described in this section.

The transformation from gain response to phase response is actually, a Hilbert transform. It may also be shown that for any a real causal sequence {x[n]}, not necessarily minimum phase, if its Fourier transform is

$$X\left(e^{j\omega}\right) = X_R\left(e^{j\omega}\right) + jX_I\left(e^{j\omega}\right) \qquad (4.36)$$

we can obtain the imaginary part from the real part or vice versa.

The cepstrum is used in the STC speech coder proposed by McAulay and Quatieri [McQ, 92], both as a means of efficiently quantising the spectral envelope and also for deriving the vocal system phase, as described in this section. Details will be given later in this chapter. Before this, we must consider some other aspects of STC. The following section examines the pitch estimation technique proposed for STC [McQ, 90].

4.6 Pitch estimation

The pitch estimation technique proposed [McQ, 90] for STC is a frequency-domain analysis-by-synthesis method. It compares two 0 to 1 kHz bandlimited time-domain signals $s_m[n]$ and $s_h[n, \omega_0]$ defined for n = -N/2, ..., 0, ..., N/2-1. The signal $s_m[n]$ is bandlimited "FS model speech" (FSMS) obtained by locating peaks in the original speech spectrum (not necessarily harmonically related) and resynthesising using the FS model. The FSMS is defined as the sum of sinusoids whose amplitudes and frequencies are assumed to be fixed and are made equal to the amplitudes and instantaneous frequencies specified at the current update-point. The phases of the FSMS components at n = 0 (in the centre of the framc) are made equal to the phases of the FS model at the current update-point. The FSMS will therefore be similar, but not identical to FS model speech; the difference being due to the fixing of the amplitudes and frequencies.

The signal $s_h[n, \omega_0]$ is bandlimited (0-1 kHz) "harmonically related sinusoidal model speech" (HSMS) obtained by summing a sinusoid at each harmonic of a proposed pitch-frequency (in the range 38 Hz to 400 Hz). The amplitude of each sinusoid is made equal to the maximum peak amplitude or the maximum value of the original speech magnitude spectrum in a “frequency slot” centred at the specified harmonic frequency and with bandwidth equal to the calculated fundamental frequency. The phase of each harmonic at n = 0 is taken to be the phase of the short-term spectrum of the windowed original speech at the frequency of this maximum amplitude within the frequency slot. Again the harmonic amplitudes and instantaneous frequencies do not change for n = -N/2, ..., 0, ..., N/2-1.

A mean squared error (MSE) between the segments of $s_m[n]$ and $s_h[n, \omega_0]$ is calculated as follows:

$$\varepsilon\left(\omega_0\right)=\frac{1}{N}\sum_{n=-N/2}^{N/2-1} | s_m[n]-s_h[n,\omega_0] |^2 \qquad (4.37)$$

for a range of pitch frequency candidates ω_0. When the MSE, $\varepsilon(\omega_0)$, is minimised, the corresponding pitch-frequency, ω_0, is chosen as the required estimate. Details of the computation required are now given.

The FSMS speech generated for a given frame by the FS model may be represented as:

$$s_m[n]=\mathrm{Re}\left(\sum_{\ell=1}^{L} A_\ell \exp\left[j\left(n\omega_\ell+\theta_\ell\right)\right]\right) \qquad \text{for } n = -N/2, ..., N/2-1 \qquad (4.38)$$

where L is the number of peaks within 0 to 1 kHz, and A_ℓ, ω_ℓ and θ_ℓ are the fixed amplitudes and frequencies and the phase offsets (instantaneous phases at n = 0) at the peaks. The HSMS speech segment derived as described above may be written as:

$$s_h[n,\omega_0]=\mathrm{Re}\left(\sum_{k=1}^{K}\tilde{A}_k\exp\left[j\left(nk\omega_0+\tilde{\theta}_k\right)\right]\right)\qquad \text{for } n=-N/2,\ldots,N/2-1 \qquad (4.39)$$

where ω_0 is the fundamental frequency, K is the number of harmonics within 0 to 1 kHz and $\tilde{A}_k$ and $\tilde{\theta}_k$ are the amplitude and phase offset (at the update-point) of the k^{th} sinusoid. The MSE, $\varepsilon(\omega_0)$, defined as the mean square difference between the segments of $s_m[n]$ and $s_h[n, \omega_0]$, within the frequency band 0 to 1 kHz, may be calculated for a possible pitch frequency candidate ω_0 as follows:

$$\begin{aligned}\varepsilon(\omega_0)&=\frac{1}{N}\sum_{n=-N/2}^{N/2-1}\left|s_m[n]-s_h[n,\omega_0]\right|^2\\&=\frac{1}{N}\sum_{n=-N/2}^{N/2-1}\left\{\left|s_m[n]\right|^2-2\,\mathrm{Re}\left[s_m[n]s_h^{*}[n,\omega_0]\right]+\left|s_h[n,\omega_0]\right|^2\right\}\end{aligned}\qquad (4.40)$$

where N is the length of a segment of speech.

According to Parseval's theorem and substituting from equation 4.39 for $s_h[n,\omega_0]$, we obtain:

$$\begin{aligned}N\cdot\varepsilon(\omega_0)&=\sum_{\ell=1}^{L}A_\ell^2-2\,\mathrm{Re}\left(\sum_{k=1}^{K}\tilde{A}_k e^{j\tilde{\theta}_k}\cdot\sum_{n=-N/2}^{N/2}s_m[n]e^{-jnk\omega_0}\right)+\sum_{k=1}^{K}\tilde{A}_k^2\\&=\sum_{\ell=1}^{L}A_\ell^2-2\,\mathrm{Re}\left(\sum_{k=1}^{K}\tilde{A}_k e^{j\tilde{\theta}_k}\cdot S_m[k]\right)+\sum_{k=1}^{K}\tilde{A}_k^2\end{aligned}\qquad (4.41)$$

where $S_m[k]$ is the N-point DFT of $\{s_m[n]\}_{-N/2,\,N/2-1}$. As mentioned before, the minimisation procedure chooses the phase of each harmonic to be the phase of the short-term spectrum of the windowed original speech at each peak frequency. So,

$$\begin{aligned}N\cdot\varepsilon(\omega_0)&=\sum_{\ell=1}^{L}A_\ell^2-2\sum_{k=1}^{K}\tilde{A}_k\,|\,S_m[k]\,|+\sum_{k=1}^{K}\tilde{A}_k^2\\&=P_s-2\rho(\omega_0)\end{aligned}\qquad (4.42)$$

where

$$P_s = \sum_{\ell=1}^{L} A_\ell^2$$

$$\rho(\omega_0) = \sum_{k=1}^{K} \tilde{A}_k \, |S_m[k]| - \frac{1}{2}\sum_{k=1}^{K} \tilde{A}_k^2 \qquad (4.43)$$

Since only $\rho(\omega_0)$ is dependent on ω_0, the required estimate of the pitch-frequency, i.e. the value of ω_0 which minimises $\varepsilon(\omega_0)$, is more easily found as the value of ω_0 for which $\rho(\omega_0)$ is maximised.

Once the optimising value of ω_0 has been determined, a signal-to-noise ratio (SNR) parameter may be calculated as follows:

$$\text{SNR} = \frac{\sum_{n=-N/2}^{N/2-1} |s_m[n]|^2}{\sum_{n=-N/2}^{N/2-1} |s_m[n] - s_h[n,\omega_0]|} = \frac{P_s}{P_s - 2\rho(\omega_0)} \qquad (4.44)$$

This SNR parameter may be used to calculate the voicing probability as will be described in section 4.8.

4.7 Derivation of onset time n_0

The onset time and its associated ambiguity bit are also derived by an analysis-by-synthesis method [McQ, 86B] [QMc, 87], but a slightly different one from that used for the pitch estimation. This method compares FS model speech with speech in which the phase is calculated from the cepstrum coefficients and the onset time n_0.

In STC the onset time n_0 is determined to make $\hat{s}[n,n_0]$ (equation 4.6) as close as possible to s[n] (equation 4.8), for n = -N/2, ..., N/2-1, according to a mean squared error (MSE) criterion. An "ambiguity bit" β also results from this calculation. The mean square error criterion is

$$\xi\,[n_0] = \frac{1}{N}\sum_{n=-N/2}^{N/2-1}\left| s[n] - \hat{s}[n,n_0] \right|^2$$

$$= \frac{1}{N}\sum_{n=-N/2}^{N/2-1}\left\{ \left|s[n]\right|^2 - 2\,\mathrm{Re}\left[s(n)\hat{s}^*(n,n_0)\right] + \left|\hat{s}[n,n_0]\right|^2 \right\} \qquad (4.45)$$

Substituting from equation 4.8 for s[n], and from equation 4.6 for $\hat{s}[n,n_0]$, we obtain:

$$\xi\,[n_0] = \sum_{\ell=1}^{L} A_\ell^2 - 2\sum_{\ell=1}^{L} A_\ell A_s(\omega_\ell)\cos\left[\theta_\ell + n_0\omega_\ell - \phi_s(\omega_\ell)\right] + \sum_{\ell=1}^{L} A_s^2(\omega_\ell) \qquad (4.46)$$

Since the magnitudes $A_s(\omega_l)$ of the speech waveform s[n] and -s[n] will be the same, an “ambiguity bit” β is introduced to cause π to be added to each of the derived phases when necessary to make them closer to the true phases. The value of β must be chosen to be either 0 or 1.

Then:

$$\xi\,[n_0] = \sum_{\ell=1}^{L} A_\ell^2 - 2\sum_{\ell=1}^{L} A_\ell A_s(\omega_\ell)\cos\left[\theta_\ell + n_0\omega_\ell - \beta\ \pi - \phi_s(\omega_\ell)\right] + \sum_{\ell=1}^{L} A_s^2(\omega_\ell)$$

$$= \sum_{\ell=1}^{L} A_\ell^2 - 2\eta(n_0,\ \beta) + \sum_{\ell=1}^{L} A_s^2(\omega_\ell) \qquad (4.47)$$

where $$\eta\left(n_{0,}\ \beta\right) = \sum_{\ell=1}^{L} A_\ell A_s(\omega_\ell)\cos\left[\theta_\ell + n_0\omega_\ell - \beta\ \pi - \phi_s(\omega_\ell)\right] \qquad (4.48)$$

Since only $\eta\left(n_{0,}\ \beta\right)$ is dependent on n_0 and β, we can minimise the MSE by maximising $\eta\left(n_{0,}\ \beta\right)$ which we call the "onset function". From equation 4.48

$$\eta\left(n_{0,}1\right) = -\eta\left(n_{0,}0\right) \qquad \text{for any value of } n_0$$

Therefore we only need to find the value of n_0 which maximises $\left|\eta\left(n_0,0\right)\right|$

where $$\eta\left(n_0,0\right) = \sum_{\ell=1}^{L} A_\ell A_s(\omega_\ell)\cos\left(\theta_\ell + n_0\omega_\ell - \phi_s(\omega_\ell)\right) \qquad (4.49)$$

and take β = 0, if $\eta\left(n_0,0\right)$ is positive at this maximum, or β =1, if $\eta\left(n_0,0\right)$ is negative at this maximum.

Because $\eta\,(\mathrm{n_0},0)$ is a highly non-linear function of n_0 it is not possible to get an analytical solution for the optimal value of n_0. The optimal value of n_0 can only be found by computing $\eta\,(\mathrm{n_0},0)$ over a range of onset times from the update-point minus half the largest expected pitch-period to the update-point plus half the largest expected pitch-period.

4.8 Voicing probability

The "voicing probability" parameter is a number in the range 0 to 1 which will be quantified to 3 to 4 bits in a fully quantised version of STC [McQ, 92]. The "voicing probability" was devised as a means of indicating the degree to which a frame of speech may be considered voiced. It is, in reality, a cut-off frequency relative to half the sampling frequency, fs/2 Hz. If $\mathrm{P_v}$ is the voicing probability, the spectrum below $\mathrm{P_v} \cdot \mathrm{fs}/2$ Hz (i.e. $\mathrm{P_v} \cdot \pi$ radians/sample) is considered to be voiced, and the spectrum above $\mathrm{P_v} \cdot \mathrm{fs}/2$ is considered to be unvoiced. There are two methods of deriving this parameter, and both are based on analysis-by-synthesis principles.

For the first method, the voicing probability $\mathrm{P_v}$ is derived from the SNR (equation 4.44) by the formula [McQ, 92]:

$$p_v = \begin{cases} 1 & \mathrm{SNR} > 10\ \mathrm{dB} \\ \frac{1}{6}(SNR-4) & 4\ \mathrm{dB} \le \mathrm{SNR} \le 10\ \mathrm{dB} \\ 0 & \mathrm{SNR} < 4\ \mathrm{dB} \end{cases} \qquad (4.50)$$

The second method is based on the result of the onset time derivation. To calculate the voicing probability factor P_v, we first define the "normalised onset function" $\tilde{\eta}$ as follows:

$$\tilde{\eta} = \frac{\left|\hat{\eta}\ (\mathrm{n_0},0)\right|}{\sum_{\ell=1}^{\mathrm{L}} \mathrm{A}_\ell^2} \qquad (4.51)$$

where $\left|\hat{\eta}(n_0,0)\right|$ is the maximum value of $\left|\eta(n_0,0)\right|$ as obtained earlier (equation 4.49). For voiced speech $\tilde{\eta}$ will be close to unity because, as can be seen in equation 4.49, the small differences between measured and derived phases will make the arguments of the cosine functions close to zero, i.e. the values of the cosine functions close to one. For unvoiced speech $\tilde{\eta}$ will be smaller as the values of the cosine functions in equation 4.49 will become smaller. The voicing probability factor P_v is then derived as follows:

$$P_v = \begin{cases} 1 & \tilde{\eta} > 0.75 \\ 4\left[\tilde{\eta} - 0.5\right] & 0.5 \le \tilde{\eta} \le 0.75 \\ 0 & \tilde{\eta} < 0.5 \end{cases} \qquad (4.52)$$

The relationship between P_v and $\tilde{\eta}$ is shown in figure 4.1.

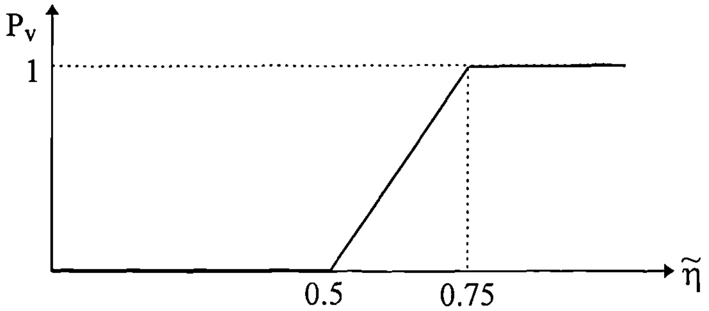


Fig. 4.1 Voice probability Pv against normalised onset function

At the decoder, P_v is used to derive a "voicing transition frequency" $\omega_c(P_v)$ which is actually a "cut-off" frequency. The cut-off frequency will be at least 1.5 kHz. For fully voiced speech the voicing-dependent cut-off frequency will be π and nothing is added to the derived phases. For unvoiced speech, the cut-off frequency will be close to 1.5 kHz and each sinusoidal component whose frequency is above this frequency will have a uniformly distributed random value added to its derived phase $\hat{\theta}_\ell$ at each sampling instant n. For transitions, the cut-off frequency will be somewhere between 3π/8 (1.5 kHz) and π (4 kHz) and the phases are assumed to be voiced (nothing added) below the cut-off frequency and to be unvoiced above it.

The voicing transition frequency i.e. the cut-off frequency is defined as follows:-

$$\omega_c\left(P_v\right) = \pi \ P_v \qquad (4.53)$$

4.9 Envelope derivation

To derive the spectral envelope at each update-point, a 512 point FFT is applied to a Hamming windowed segment of input speech, two and a half pitch periods in length, centred on the update-point and "zero padded" as described in section 3.2. The Hamming window and the resulting segment are energy normalised as described in section 3.3.

For voiced speech, the FFT derived spectrum will have a magnitude spectrum which, in principle, has peaks at pitch-frequency harmonics. To obtain the spectral envelope, a simple peak-picking method is applied to the linear magnitude spectrum to locate these peaks using the spectral envelope estimation vocoder (SEEVOC) technique [Pau, 81]. This technique picks the largest peak in each "frequency slot". If the fundamental frequency is ω_0, the kth "frequency slot" means the frequency range between $k\omega_0 - \omega_0/2$ and $k\omega_0 + \omega_0/2$ where k = 1, 2, ...,L and L is the number of harmonics. Smoothly interpolating these peaks produces a spectral envelope.

There are many interpolation methods which may be used to derive an envelope. One common method is linear interpolation and another is cubic spline interpolation. Linear interpolation method is simple and is applied to the magnitude spectrum (linear amplitude scale) as follows:

Taking each pair of adjacent peaks whose frequency locations are w_j and w_{j+1} with corresponding amplitudes a_j and a_{j+1} the amplitude 'a' at frequency 'w' between w_j and w_{j+1} can be obtained as:

$$a = a_j + (a_{j+1} - a_j)(w-w_j)/(w_{j+1}-w_j) \qquad (4.54)$$

The principle of cubic spline interpolation is to fit an envelope to the spectrum peaks which is smooth in its first derivative with respect to frequency and continuous in the second derivative, both within an interval between adjacent peaks and at its boundaries. The cubic spline interpolation method is commonly applied to the peaks of the log magnitude spectrum (natural logarithms) to form a smooth fitted envelope. Assume two adjacent peaks are located at w_j, w_{j+1} and their amplitudes are a_j, a_{j+1} respectively. Firstly, a set of tabulated values for the function's second derivatives, $\ddot{a}_j$ and $\ddot{a}_{j+1}$, are derived. The interpolation formula for the amplitude 'a' at frequency 'w' is a cubic polynomial:

$$a = Aa_j + B\,a_{j+1} + C\,\ddot{a}_j + D\,\ddot{a}_{j+1} \tag{4.55}$$

where

$$A \equiv \frac{w_{j+1} - w}{w_{j+1} - w_j}; \quad B \equiv \frac{w - w_j}{w_{j+1} - w_j}; \tag{4.56}$$

$$C \equiv \frac{1}{6}\left(A^3 - A\right)\left(w_{j+1} - w_j\right)^2; \quad D \equiv \frac{1}{6}\left(B^3 - B\right)\left(w_{j+1} - w_j\right)^2; \tag{4.57}$$

Equation 4.55 assumes that the second derivative varies linearly from its value $\ddot{a}_j$ at w_j to its value $\ddot{a}_{j+1}$ at w_{J+1}. This means that the second derivative has to be continuous across the boundary between the two intervals w_{j-1} to w_j and w_j to w_{j+1}. The second derivatives are derived from following equations [PTVF, 92]:

$$\frac{w_j - w_{j-1}}{6}\ddot{a}_{j-1} + \frac{w_{j+1} - w_{j-1}}{3}\ddot{a}_j + \frac{w_{j+1} - w_j}{6}\ddot{a}_{j+1} = \frac{a_{j+1} - a_j}{w_{j+1} - w_j} - \frac{a_j - a_{j-1}}{w_j - w_{j-1}}$$

$$\text{for } j=1, 2, \ldots, N \tag{4.58}$$

These are N-2 linear equations in the N unknown $\ddot{a}_j$, j=1, 2, ..., N. Another two additional conditions are obtained by setting $\ddot{a}_1$, and $\ddot{a}_N$ to be zero (this setting gives the so-called natural cubic spline).

A set of cepstral coefficients are then derived by performing a 512 point inverse FFT on the logarithm of the spectral envelope. The first 30 cepstral coefficients are intended to be encoded. To achieve greater economy in the required bit-rate, McAulay and Quatieri [McCQ, 93] apply a frequency warping procedure to the spectral envelope. This procedure compresses the spectral shape at higher frequencies so that it is less accurately represented than the shape of lower frequencies. A similar frequency warping procedure is investigated in Chapter 6 of this thesis.

A spectral envelope may also be derived simply by selecting only an appropriate number of cepstral coefficients from the cepstrum of the unmodified speech segment. The number would be pitch-period dependent and normally considerably less than 30. A non-rectangular window would also be necessary. The difference between this approach and the cubic spline method used in STC is that the envelope derived by cubic spline interpolation is guaranteed to have the correct amplitude at each harmonic frequency. The unmodified speech cepstrum windowing method produces an envelope which may not fit all harmonic peaks. Clearly truncating the cubic spline based cepstrum to 30 coefficients and the subsequent vector quantisation of these will distort the amplitudes to some extent. However the distortion is more predictable and is not pitch-frequency dependent. Finally, and perhaps most importantly, fixed length vector quantisation codebook is appropriate to this approach. The effect of truncation the cepstrum to 30 coefficients is investigated later in this chapter.

Yet another way of deriving a spectral envelope is to fit a high order all-pole model to the speech spectrum as proposed by McAulay & Quatieri [McCQ, 93]. This technique then allows the envelope to be represented as a set of LSF coefficients. Since many LSF-based vector quantisation algorithms have been well researched, such a algorithm may be employed to some advantage in STC. This approach will be discussed further in Chapter 6.

4.10 Speech synthesis

At the decoder, for each update-point, a set of frequencies for a sinusoidal model are determined from the fundamental frequency obtained from the encoded pitch-period and voicing probability. Below the "voicing transition frequency" $\omega_c(P_v)$ derived from the voicing probability P_V, the frequencies are made to be harmonically related to the fundamental frequency. Above $\omega_c(P_v)$, the frequencies are set to be 100 Hz apart. The spectral envelope and the system phase spectrum are derived from a set of encoded cepstral coefficients. These spectra are sampled at the frequency ω_ℓ of each sinusoid to obtain the required amplitudes A_ℓ and system phases $\phi_s(\omega_\ell)$. The instantaneous phase $\hat{\theta}_\ell$ can then be derived for each sinusoid ℓ as:

$$\hat{\theta}\ (\omega_\ell) = -n_0\omega_\ell + \phi_s(\omega_\ell) + \beta\ \pi \tag{4.59}$$

where $\phi_s(\omega)$ is the system phase derived from the envelope via a Hilbert transform and n_0 and β are the onset time and the ambiguity bit respectively.

For each sinusoid ℓ, $\hat{\theta}_\ell$ should be very close to the measured phase θ_ℓ for voiced speech. However we must expect there to be significant phase differences between θ_ℓ and $\hat{\theta}_\ell$ when the speech is unvoiced or a voiced/unvoiced transition. These phase differences probably cannot be encoded by a very low bit-rate coder, but assumptions can be made about their perceptual effects. These effects can be masked to some extent at the receiver by adding to the derived phase $\hat{\theta}_\ell$ for each sinusoid a "phase residual" error, $\mu\ (\omega_\ell)$, which is set to zero for voiced speech, but becomes a random variable of appropriate magnitude (frequency dependent) for unvoiced speech and transitions. The "phase residual" error $\mu\ (\omega_\ell)$ which is added to each $\hat{\theta}_\ell$ is therefore defined as follows:

$$\mu\ (\omega_\ell) = \begin{cases} 0 & \text{if } \omega_\ell \le \omega_c(P_v) \\ U[\pi,\ \pi] & \text{if } \omega_\ell > \omega_c(P_v) \end{cases} \tag{4.60}$$

where U[-π, π] is a uniformly distributed random value between -π to π. So

$$\hat{\theta}\ (\omega_\ell) = -n_0\omega_\ell + \phi_s(\omega_\ell) + \beta\ \pi + \mu(\omega_\ell) \tag{4.61}$$

The idea of this phase derivation is to replace measured phases θ_ℓ by derived phases $\hat{\theta}_\ell$. As we need to make the instantaneous phases coincide with the derived phases at the update-points, i.e. at the synthesis frame boundaries, the instantaneous phases have to be made equal to the derived phases.

The speech is then synthesised by linearly interpolating amplitudes and cubically interpolating instantaneous phases between two update-points as proposed for the FS model [McQ, 86A]. The frequency matching method also needs to be used before phase interpolation as for the FS model.

In principle, this low bit-rate approach may be expected to maintain synchronism with the original speech and preserve more characteristics of the original speech than techniques which do not attempt to maintain the correct phase relationships between pitch frequency harmonics.

The biggest difference between the low bit-rate STC phase derivation technique and the FS model phase *interpolation technique is that the former derives phase at the* decoder and the latter includes the phase information in its set of parameters.

4.11 Simulation of the STC model

A Turbo C++ implementation of the STC model has been developed [SC, 95]. It is basically as described by McAulay and Quatieri [McQ, 92] though some modifications have been introduced. It is as yet unquantised. The problem of quantising its parameters will be discussed in chapter 6. The IMBE pitch estimation technique [MSDI, 91] is applied to obtain an estimate of pitch-frequency and an FFT spectrum is then produced from a windowed segment of speech with two and half pitch-period length for voiced speech and fixed window length for unvoiced speech. A set of cepstral coefficients are derived from the spectral envelope of the speech, where the envelope is estimated using the SEEVOC technique [Pau, 81]. Using the analysis-by-synthesis methods described earlier to minimise the mean square error between the measured speech and the synthetic speech, the voice probability, onset time and ambiguity bit are derived.

4.11.1 Input data format and scaling

As specified for the IMBE pitch detector [MSDI, 91], the input speech data is assumed to be stored in 16-bit two's complement form in byte-reversed order (i.e. low-byte, high-byte order). The amplitude of the input speech is assumed to conform to the INMARSAT standard [MSDI, 91]; i.e. positive peaks of the voiced portions of the speech should ideally lie between 16,383 and 32,767, with negative peaks between -32,767 and -16,383. The dc level should be approximately zero. This means, essentially, that the speech samples should try to occupy all 16 bits without incurring overflow or clipping. The recommendation is also that the rms input speech level be made roughly 25 dB below the rms value of the largest possible sine-wave that can be accommodated without clipping. This sine-wave amplitude is 32,767 giving an rms value of $32{,}767/\sqrt{2}$ = 23,173 (approx). Therefore the recommended rms value of the input speech is 1303 (approx). In practice, a "front-end" AGC circuit will amplify the input speech with this ideal rms value as a guide. Another criterion, which should be roughly equivalent, is that ideally the average of the peak amplitudes should be about 6 dB below the clipping level. That means that if the maximum amplitudes are taken from a succession of pitch-cycles (one peak per pitch-cycle) the average of these amplitudes should ideally be 16,383. This program does not simulate the AGC and assumes the input speech file to be correctly scaled.

4.11.2 Parameters of the STC model

At the encoder, the following set of parameters are estimated at each update-point:

- Pitch-frequency
- A set of cepstral coefficients
- Voicing probability
- Onset time (n_0) and ambiguity bit

These parameters may be encoded at a total bit-rate around 3 kb/s. This allows about 8 bits per 20 ms frame for the pitch-frequency, 36 bits for the cepstral coefficients, 4 bits for the voicing probability, 9 bits for the onset time and one ambiguity bit. It was pointed out by McAulay and Quatieri [McQ, 92] that considerable accuracy (hence 9 bits) is required for the onset time as the phase of the reconstructed speech will be highly dependent on this parameter and errors in it will cause roughness. For a 2.4 kb/s version of this STC coder, it was recommended [McQ, 92], that only the pitch-frequency, cepstral coefficients and voicing probability be encoded. In this case, the onset time is not encoded, but is instead obtained by adding an appropriate number of pitch-periods (linearly interpolated), to the previous onset time as illustrated as figure 4.2. The appropriate number is determined as the minimum number of pitch-periods that must be added to go beyond the frame boundary.

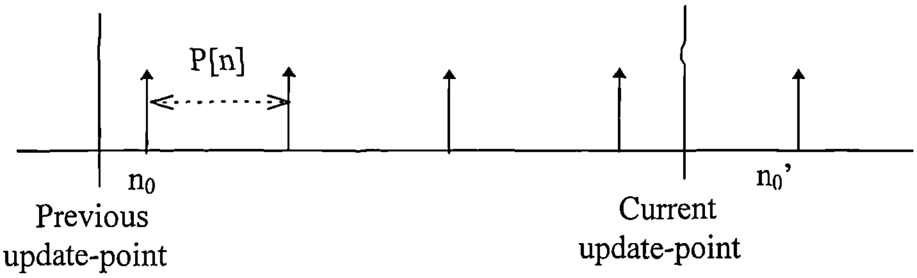


Fig. 4.2 Onset time derived by adding pitch periods to the previous one

where P[n] is the pitch-period linearly interpolated between two update-points. Since the ambiguity bit essentially caters for mistakes in the estimation of n_0 at the encoder, it is also not included in the 2.4 kb/s version. The omission of n_0 and the ambiguity bit from the encoded parameters means that the reconstructed speech will no longer be in synchronism with the original speech ; i.e. there is no attempt to preserve the absolute time locations of excitation points. Only their relative positions are preserved.

4.11.3 The effect of truncation of cepstral coefficients

Since only a sub-set of the cepstral coefficients are encoded, it is necessary to estimate the effect of truncating the complete set of cepstral coefficients. Figure 4.3 gives an indication of the mean magnitude spectral distortion that was caused by truncating the complete set of cepstral coefficients {c[0], c[1], ... c[255]} to some smaller subset {c[0], c[1], ... , c[K-1]} for a set of spectral envelopes typical of voiced speech. The measurements are with reference to magnitude spectra obtained using all 256 cepstral coefficients and are averaged over 1000 sets of cepstral coefficients obtained from segments of normal voiced speech. It may be seen in figure 4.3 that when the number, K, of cepstral coefficients was truncated to 30, the mean spectral distortion was about 1 dB. Over 1000 voiced frames, only 1.9% of outliers in the range 2-4 dB, and no outliers with spectral distortion greater than 4 dB were observed. On the basis of the objective criteria published by Paliwal [PA, 93] it may be anticipated that this degree of spectral envelope distortion should not be perceptible.

Figures 4.4, 4.5 and 4.6 illustrate three different spectral envelopes encountered in the experiment above, each obtained using 30 cepstral coefficients. In each case the spectral envelope obtained using all 256 cepstral coefficients are shown for comparison. The envelopes were selected with considerably different spectral distortion: i.e. 0.4 dB for the first, 1.12 dB for the second and 2.02 dB for the third. It may be observed that the 0.4 dB of spectral distortion is barely discernible, some flattening of peaks is seen at 1.12 dB, and more severe loss of detail is seen at 2.02 dB spectral distortion. The structure of the three envelopes is rather different: the first (producing the lowest distortion) is relatively simple and corresponds to a reasonably steady vowel. The second has rather more detail and rather more peaks, and the third is even more complicated and corresponds to a transition.

Figure 4.7 illustrates the effect on the spectral envelope used in figure 4.4 of truncating the number of cepstral coefficients from 256 to three different numbers: i.e. 30, 20 and 10 coefficients. The effect of truncating to 30 coefficients is, as observed previously, barely discernible. Twenty coefficients flatten some of the peaks noticeably, and ten coefficients fail to follow the detailed structure of the envelope and instead give only its general trend.

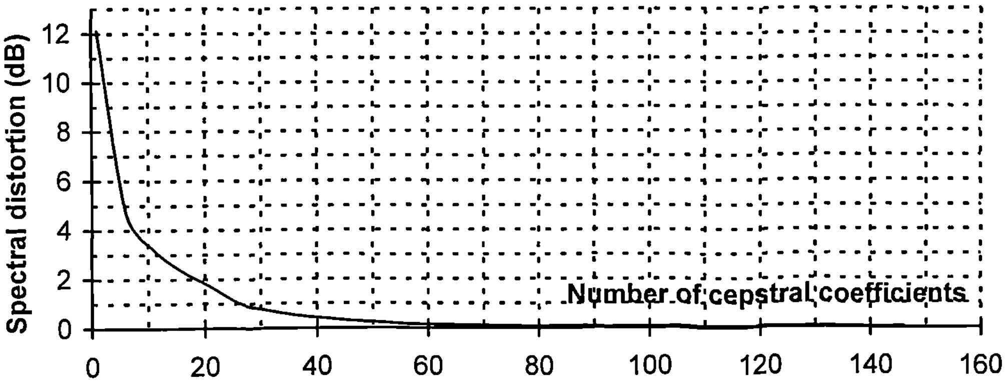


Fig. 4.3 The mean spectral distortion over 1000 voiced frames

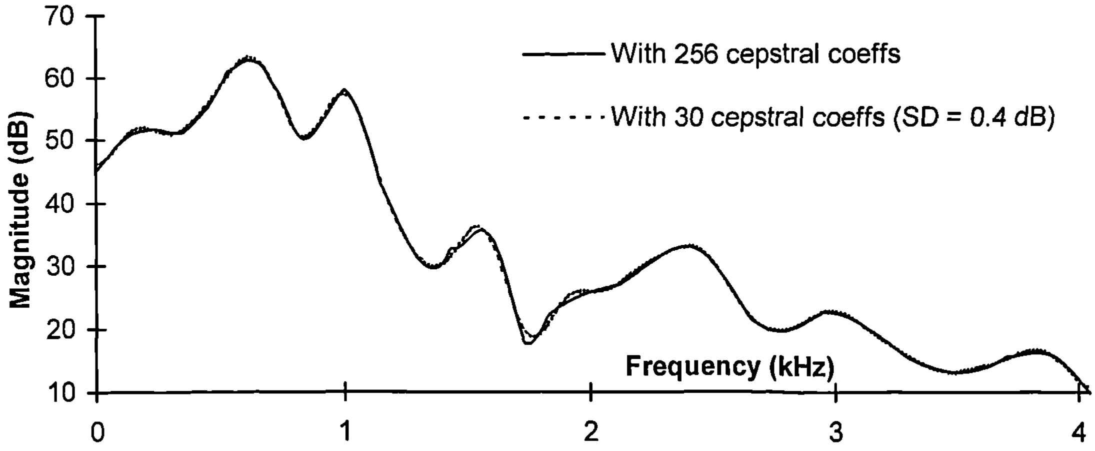


Fig. 4.4 A voiced spectral envelope with SD about 0.5 dB (30 cepstral coefficients)

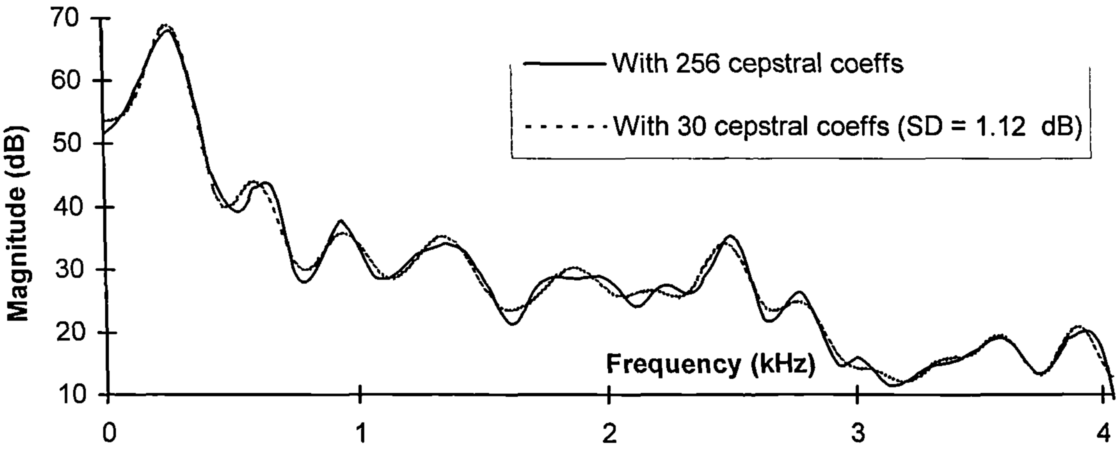


Fig. 4.5 A voiced spectral envelope with SD about 1 dB (30 cepstral coefficients)

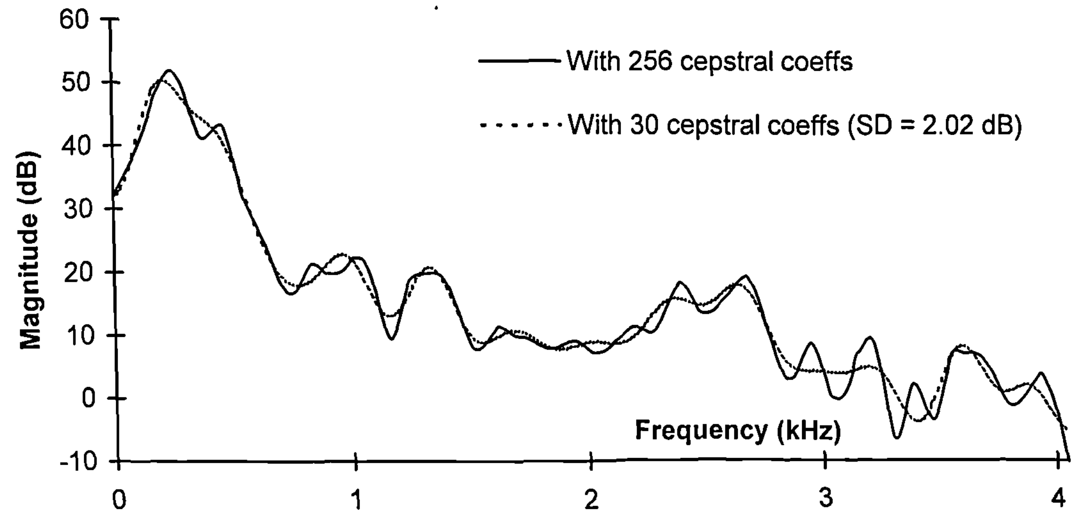


Fig. 4.6 A voiced spectral envelope with SD about 2 dB (30 cepstral coefficients)

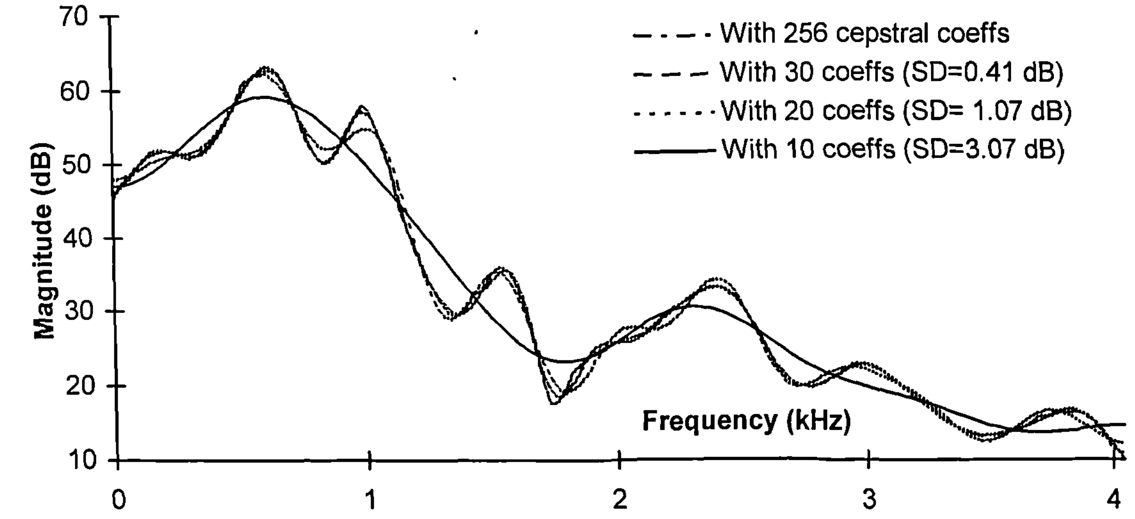


Fig. 4.7 Spectral envelope with 256, 30, 20 and 10 cepstral coefficients

Listening experiments were also carried out to further investigate the effect of truncating the number of cepstral coefficients. STC modelled speech was synthesised using different numbers of cepstral coefficients to represent the spectral envelope: i.e. 256, 150, 50, 30, 20 and 10 cepstral coefficients. Ten seconds of male and female speech [from Gsp.pcm] was used and the model was completely unquantised. It was found in informal listening tests that the synthetic speech obtained using 30 cepstral coefficients, or more, could not be distinguished from that obtained using 256 cepstral coefficients. With 20 coefficients, slightly distortion was heard, though the speech was still completely intelligible and had a degree of naturalness. With 10 coefficients, the quality was poor and there was some loss of intelligibility.

The graphs and results obtained in this section give us some confidence that taking 30 cepstral coefficients to represent speech spectral envelopes, as recommended by McAulay and Quatieri [McQ, 92], is not likely to be unreasonably inaccurate or extravagant. A final consideration is that, in a fully quantised STC coder as will be seen in chapter 6, the spectral envelope will be frequency warped to try to distribute quantisation noise on a perceptual basis. We therefore repeated some of the experiments above with frequency warping applied to the spectral envelope prior to it conversion to cepstral coefficients. No quantisation is yet applied, but it may be anticipated that the frequency warping will produce an envelope that is less smooth at higher frequencies and therefore more difficult to represent by a restricted number of cepstral coefficients.

A graph of the mean spectral distortion against the number of cepstral coefficients when these are derived from frequency-warped envelopes is shown in figure 4.8. This graph is very similar to that obtained from the unwarped envelope. For 20 and 10 coefficients it is lower by 0.13 and 0.05 dB respectively. For 30 coefficients it is higher by 0.08 dB. Therefore, with 30 cepstral coefficients, the mean spectral distortion is still about 1 dB. With 30 cepstral coefficients the percentage of outliers in the range 2-4 dB was found to have decreased to 1.8% but the percentage of outliers beyond 4 dB had increased to 0.1%. A comparison of spectral envelopes obtained using 10, 20, 30 and 256 frequency warped cepstral coefficients is shown in figure 4.9. It was concluded that the effect frequency warping on the accuracy of spectral envelopes represented by truncated cepstral coefficients is small when the number of coefficients is 30, and can be disregarded when designing a fully quantised STC coder.

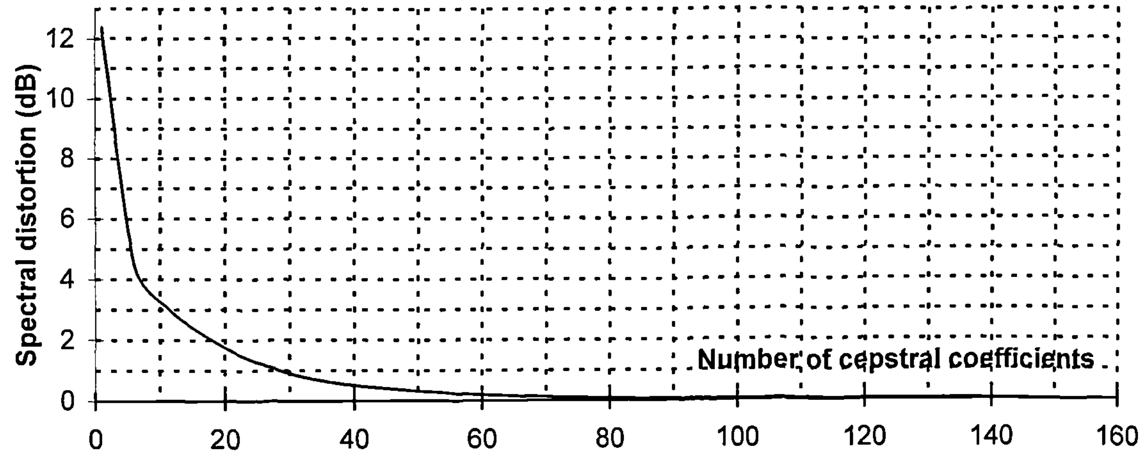


Fig. 4.8 The mean spectral distortion over 1000 voiced frames (frequency warped)

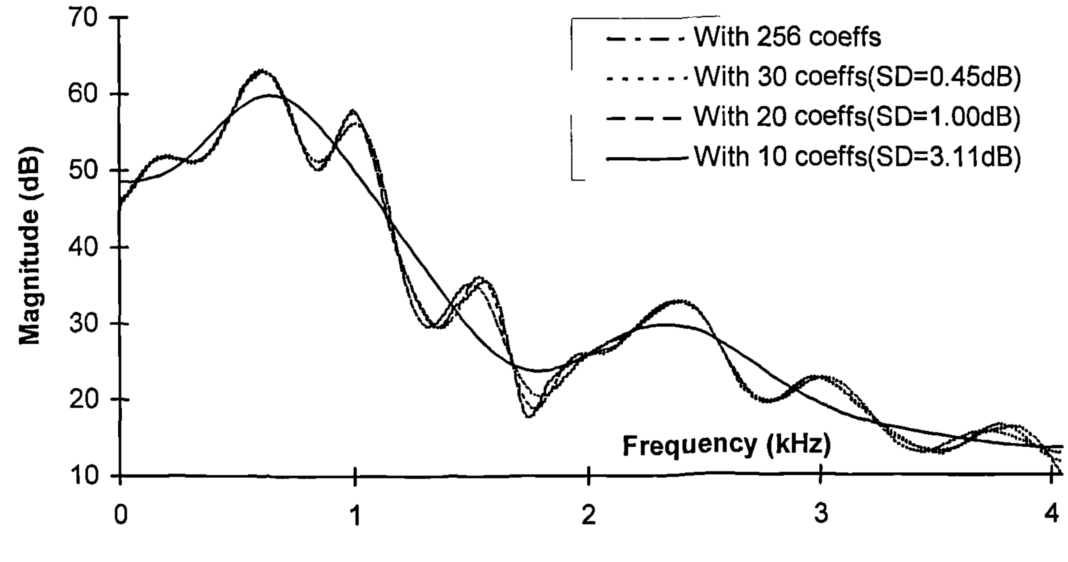


Fig. 4.9 Spectral envelopes with 256, 30, 20 and 10 cepstral coefficients (frequency warping was applied)

4.11.4 Evaluation of the STC model

Various examples of studio recorded (i.e. “clean”) speech were used to test the STC modelling technique. The synthetic speech produced was found to have good quality, preserved naturalness and speaker recognisability for both male and female voices. The quality was especially good for male speech, and for a short example (about 4 seconds) of male voiced only speech [Away.pcm] was found by many listeners to be barely distinguishable from the original. When tested with extended segments of voiced and unvoiced male and female speech [e.g. Gsp.pcm] the speech remained good, and appeared to be virtually toll quality when heard through hi-fi speakers in a fairly reverberant room. Only the effect of the occasional gross pitch estimation errors marred the perceived quality. However, when heard through good quality headphones *various other types of distortion was clearly heard* in the synthesised speech. The speech was easily distinguishable from the original under these conditions making it not quite toll quality. Male speech portions were generally better than female, though some reverberance was present at times. Female portions were considered slightly less natural than male. The 32 ms segment of male voiced speech shown in figure 4.11 may be compared with the corresponding STC modelled synthetic speech segment shown in figure 4.12. It may be seen that the two waveshapes are generally quite similar, though there is some loss of detail around the peaks of the synthetic waveform. It may also be observed that the synthetic speech is more “peaky” than the original in that the main peaks are more pronounced and the other peaks die away more rapidly within each pitch cycle. We believe this feature is important and it will be discussed in chapter 5.

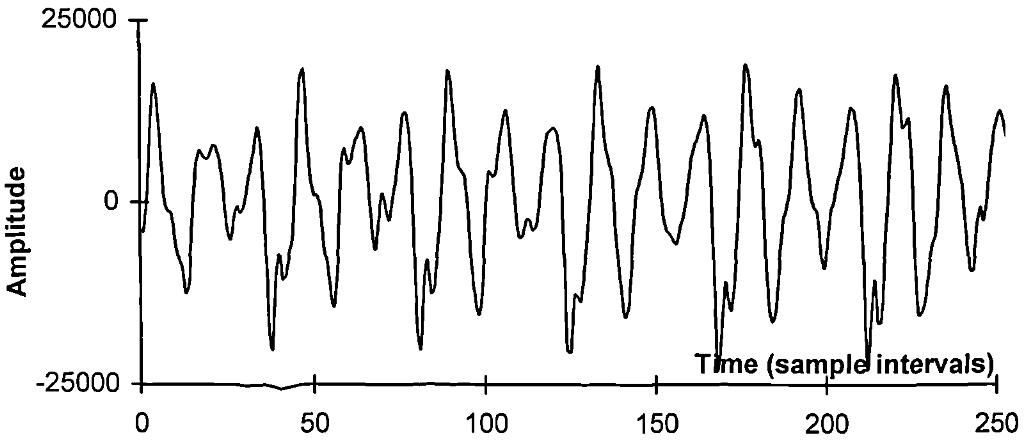


Fig. 4.10 A segment of original speech

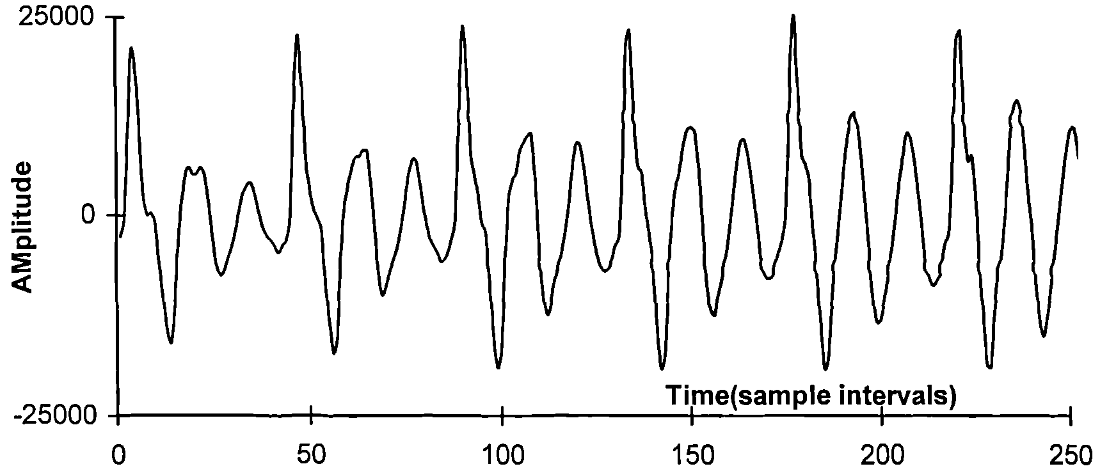


Fig. 4.11 A segment of synthetic male voiced speech obtain from STC

4.11.5 Performance of STC with noisy speech

Experiments were also carried out to investigate the performance of the STC model when representing speech produced in a noisy environment. Such experiments would be needed to determine whether a modelling technique could possibly be used as the basis of a speech coder for mobile telephony. The STC model was tested with noisy speech with a range of different signal-to-noise ratios (SNR). The noisy speech was produced by adding pseudo-random Gaussian noise to a recording of clean natural speech [Gsp.pcm]. The SNR was defined as the ratio of RMS value of the clean speech to the RMS value of the added noise over 10 seconds. In each case the RMS value was measured over the whole speech recording which includes voiced speech, unvoiced speech and silence. Files with SNR values of 30 dB, 20 dB, 10 dB and 0 dB were produced in this way. It should be pointed out that the male talker in the recording speaks rather louder than the female talker.

When the SNR was 30 dB, the STC synthesised speech itself did not suffer gross degradation in comparison to what was obtained from clean speech. A background noise signal was clearly audible, and its nature was changed considerably and was more tonal and variable, especially when heard through headphones. There appeared to be no additional pitch estimation errors, and the background noise did not seem to be more disturbing than in the unprocessed noisy speech file.

With 20 dB SNR, the male speech was still clear above the noise and the female voice though slightly less natural sounding was completely intelligible. With 10 dB SNR, male speech was slightly less natural but quite intelligible above the noise and the female speech had greater distortion and possibly some loss of intelligibility (the speech was not tested for intelligibility).

With 0 dB SNR, both male and female speech were perceivable above the noise and the male speech and even portions of the female speech could be considered intelligible. Examining the envelopes of the time-domain STC waveforms at 0 dB SNR, see figure 4.12 (a) and (b) reveals that the female speech is virtually lost in the noise as it is in the original noisy recording. However there still did not appear to be gross mistakes, for example in pitch determination. Even at -6 dB SNR, when almost none of the speech could be discerned in the envelope of the time-domain waveform, a semblance of possibly intelligible male and even female speech was heard and there was no audible evidence of gross pitch errors and other mistakes. The STC model therefore appears to degrade gracefully as the signal-to-noise ratio decreases. It should be noted that these results were obtained from the speech with white noise, different conclusions may be drawn when the input speech contains other type of noise.

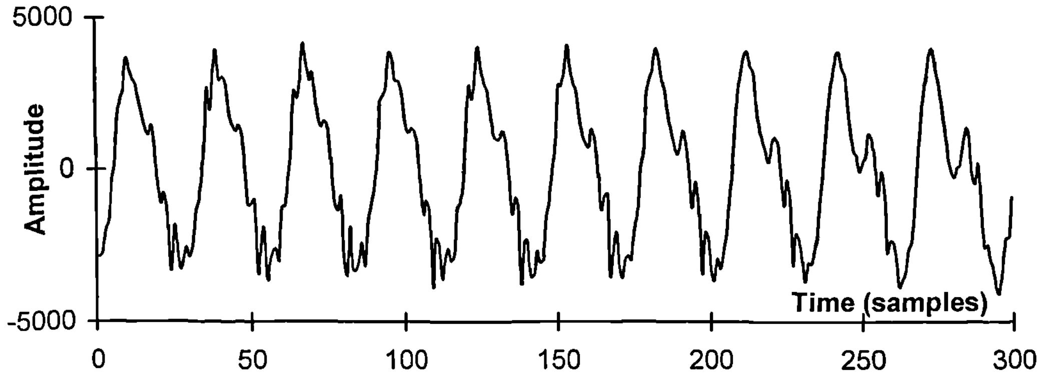


Fig. 4.12.(a) A segment of clean original female speech

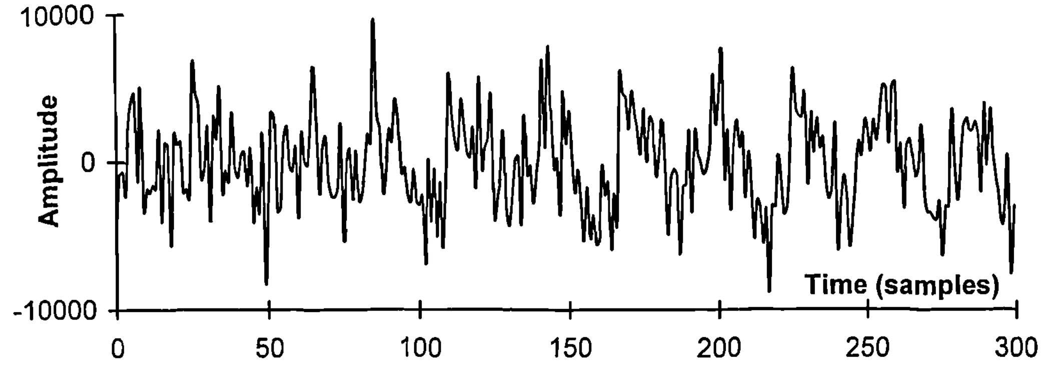


Fig. 4.12.(b) Same segment of speech with SNR 0 dB obtained from STC model

There was a suggestion that deriving the phase spectrum from the envelope would be inappropriate when the signal-to-noise ratio is very low and the envelope would be seriously distorted by the noise. In such cases, it may be better to dispense with the derived phase spectrum and instead use the non-phase quadratic interpolation technique used, for example, by IMBE. Experiments were tried to investigate this hypothesis.

Noisy speech was applied as before to the STC analyser, and synthetic speech was produced using first cubic interpolation and then quadratic interpolation. At an SNR of 30 dB, the cubically interpolated speech was still preferred as expected. Perhaps surprisingly the speech remained slightly better using cubic interpolation as the SNR was reduced to 20, 10 and 0 dB. The nature of the noise was very different, and appeared more tonal for quadratic interpolation than for cubic. It was concluded that there is likely to be no advantage in changing the interpolation scheme depending on the SNR.

4.12 Application of phase derivation technique to IMBE

To further investigate the phase regeneration technique proposed by McAulay & Quatieri, it was applied to the 4.1 kb/s standard Inmarsat IMBE decoder [MSDI, 91]. The synthetic speech produced by the traditional IMBE decoder uses quadratic interpolation which requires no phase information. It was noticed that the speech waveshape obtained from IMBE tends to be rather more peaky and sinc function like than that of the original speech. This is illustrated in figure 4.13(b).

The phase derivation used in STC was applied to the IMBE decoder, without requiring any extra information to be encoded. The normal IMBE received harmonic amplitudes, were used to produce a spectral envelope from which system phases were derived via a Hilbert transform. Cubic interpolation was then used to produce the instantaneous phase of each sinusoid replacing the quadratic interpolation used in the standard IMBE decoder. By introducing phase information in this way, the synthetic speech waveform shape, illustrated in figure 4.13(c), was made to look closer to the original speech waveshape in figure 4.13 (a) than that of the standard IMBE decoder. Listening tests indicated that under some listening conditions the synthetic speech quality was marginally improved by this approach. More investigation of this idea are needed.

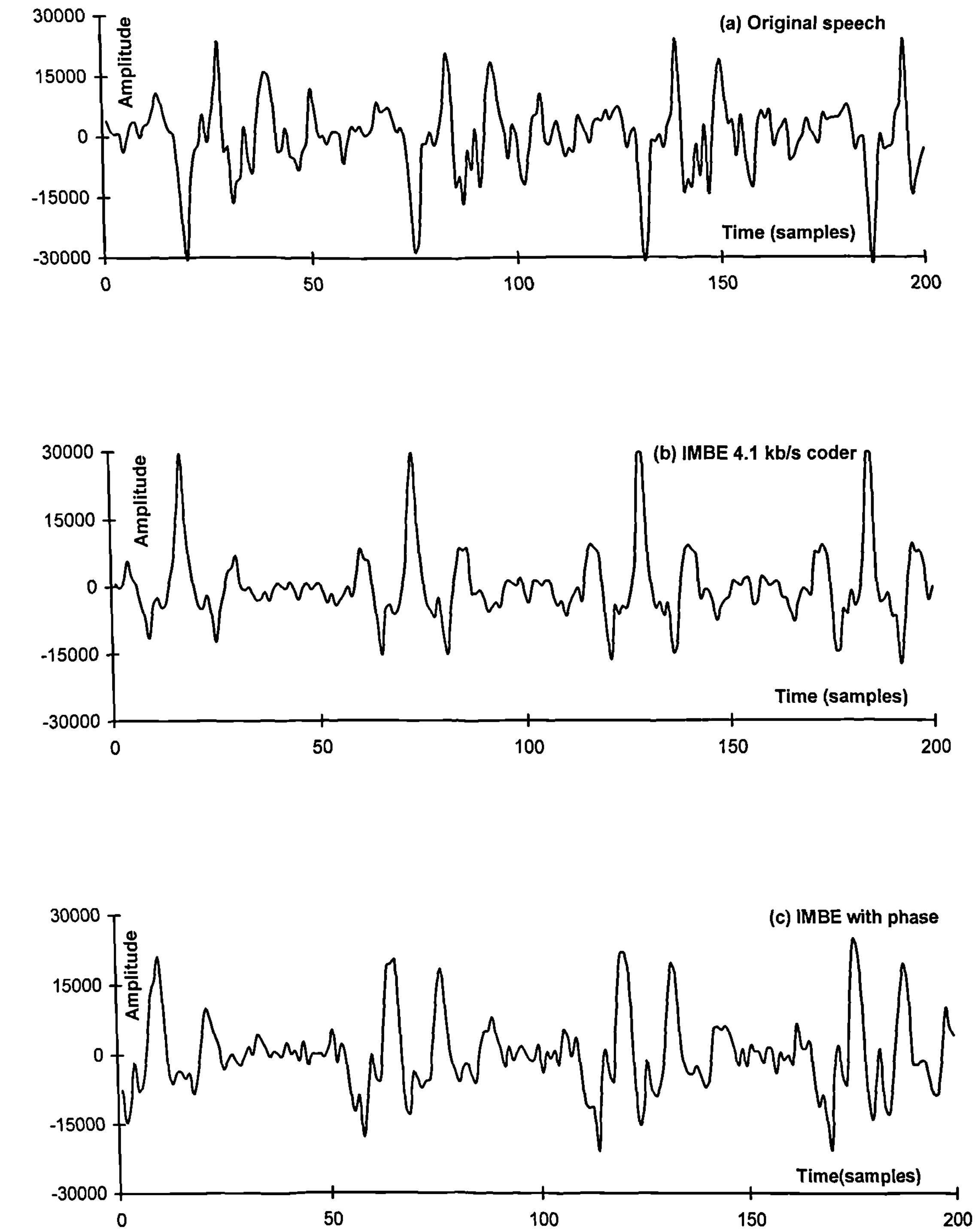


Fig. 4.13 (a) Original speech; (b) 4.1 kb/s IMBE coder;

(c) Phase information is added in 4.1 kb/s IMBE coder

4.13 Quasi-Pitch-Synchronous Sinusoidal (QPSS) model

4.13.1 The principle of the model

A new variation of STC, based on a modified interpolation formula has been investigated and implemented. The technique is referred to as quasi-pitch-synchronous sinusoidal (QPSS) coding. The principle of the technique is to separate the instantaneous phase of each sinusoidal component into two parts: (i) the vocal system phase and (ii) the excitation phase. These are interpolated separately. An advantage is that while interpolating the excitation signal phase, a degree of synchronism can be maintained between the synthesis excitation points (i.e. the points where all the phases coalesce) and the excitation points in the original speech. This is achievable without the need to encode both a pitch-frequency and an "on-set time" for each update-point.

Assume the instantaneous phase of the k^{th} sinusoid is $\sigma_k[n]+\alpha_k[n]$, where $\sigma_k[n]$ is the instantaneous phase of the k^{th} harmonic of an assumed excitation signal and $\alpha_k[n]$ is the instantaneous phase of the vocal system function which includes the effect of the glottal filter, the vocal tract and lip radiation. The speech over a synthesis frame for n = 0, 1, ..., N-1 can be represented as:

$$s[n] = \sum_{k=1}^{K} A_k[n]\cos\left(\sigma_k[n]+\alpha_k[n]\right) \tag{4.62}$$

where K is the number of sinusoidal components and $A_k[n]$ are the amplitudes of these components at time n for k = 1, 2, ..., K.

Formulae for $\sigma_k[n]$ and $\alpha_k[n]$ are derived separately. At each update-point, the vocal system phases $\alpha_k[n]$ are obtained as with STC from encoded cepstral coefficients via a Hilbert transform relationship assuming they represent the spectral envelope of a minimum phase vocal system. Between update-points, each $\alpha_k[n]$ can be interpolated between the system phases at the previous and the current update-point. Because the complex cepstrum is derived from the continuous complex logarithm [OS, 75], which means that both magnitude and phase spectra are

DFT samples of continuous functions of ω, there need be no provision for apparent discontinuities at multipl es of 2π. Cubic interpolation is used so that the constraints $\dot{\sigma}_k[0] = \dot{\sigma}_k^{prev}[N]$ can be satisfied thus discontinuities of instantaneous frequency are not created at synthesis frame boundaries.

For each k, a smoothly changing continuous formula for the excitation phase $\sigma_k[n]$, should be derived which satisfies the following conditions:

- For each sinusoid: $\sigma_k[n] = k\sigma[n]$ where $\sigma[n] = \sigma_1[n]$ is the instantaneous excitation phase for the fundamental frequency component.
- The value of $\sigma_k[n]$ at each frame boundary should be between -π and π.
- To maintain continuity at frame boundaries the value of $\sigma_k[n]$ at the beginning of a frame, i.e. when n = 0, should be equal to the value of $\sigma_k[n]$, as defined for the previous frame, at the end of the previous frame, i.e. when n = N.
- the derivative of $\sigma_k[n]$ with respect to time, i.e. the instantaneous frequency of the k^{th} sinusoid, should similarly maintain continuity at frame boundaries. (Note that unlike conventional STC and most other sinusoidal modelling techniques we do not directly encode the value of instantaneous frequency at frame boundaries.)
- Ideally $\sigma_k[n]$ should be equal to an integer multiple of 2π at each excitation point. (This means that all sinusoidal components are in phase at each excitation point. i.e. $\sigma_k[n]$ = 2πk at the first excitation point within the frame, 4πk at the second and so on.)

Following the practice of STC, we decided initially to investigate a version of QPSS which encodes the sample number of just the first excitation point within each frame. Additionally, the number of complete pitch-periods between this excitation point and the first excitation point in the previous frame must be encoded. These two items of data replace the pitch-period measurement encoded by the lowest bit-rate versions, i.e. versions which do not encode an onset time. The onset time is now obtained at little or possibly no increase in bit-rate.

Assume the phase interpolation formula for the fundamental sinusoid over a frame n = 0 to n = N-1 is:

$$\sigma[n] = A + Bn + Cn^2 \tag{4.63}$$

Therefore $$\sigma[n]\big|_{n=0} = A = \sigma^{prev}[N] \tag{4.64}$$

$$\dot{\sigma}[n]\big|_{n=0} = B = \dot{\sigma}^{prev}[N] \tag{4.65}$$

where $\sigma^{prev}[N]$ and $\dot{\sigma}^{prev}[N]$ are the values of instantaneous phase and instantaneous fundamental frequency at the end of the previous synthesis frame. The value C may be determined by a knowledge of the location of one excitation point and the number of pitch-periods between the beginning of the frame and that excitation point.

The value of C will determine the point at which all phases become integer multiples of 2π. However when this point is very close to the beginning of the frame, the equation for C can become ill-conditioned (i.e. highly dependent on $\sigma_k[0]$ and $\dot{\sigma}_k[0]$). Therefore, it was decided to define n_{next}as the location of the first peak of the next synthesis frame and to choose C such that if the formula for $\sigma[n]$ were extended from the current frame into next frame, the phases of the sinusoids would coalesce (i.e. become correct integer multiples of 2π) at n = n_{next}. In fact, $\sigma[n]$ is never extended into the next frame and it is possible that the phases will not actually coalesce with the new formula adopted for $\sigma[n]$ in the next frame. However, the phases should still approximately coalesce in the next frame and, more importantly, the instantaneous phase and frequency at the beginning of the next frame should have appropriate values. Denoting by L the number of pitch cycles (encoded) between the first excitation point in the current frame and the first excitation point in the next frame, to achieve coalescence at n = n_{next};

$$\sigma[n_{next}] = 2\pi L = A + B n_{next} + C n_{next}^2 \tag{4.66}$$

Therefore $$C = \frac{L \cdot 2\pi - \sigma[0] - \omega[0] \cdot n_{next}}{n_{next}^2} \tag{4.67}$$

The synthetic speech, therefore, equals:

$$s[n] = \sum_{k=1}^{K} \cos(k \cdot \sigma[n] + \alpha_k[n] + \mu_k[n]) \qquad (4.68)$$

where K is the number of sinusoidal components and $\mu_k[n]$ is a voicing dependent sequence based on the voicing probability P_v. The value of $\mu_k[n]$ will be zero for frequencies below πP_V and will be a uniformly distributed random value between $-\pi$ to π, on a sample-by-sample basis, for frequencies above πP_V.

QPSS also uses cepstral coefficients to represent the envelope and derives the system phase from the received envelope at the decoder assuming the vocal system to be minimum phase. The voicing probability is also calculated in the same way as in STC. So for QPSS, the following information needs to be encoded for each 20 ms (160 samples) frame:

- The number of pitch-cycles in the frame. (3 bits)
- The location of the first excitation signal peak in the frame (actually the first speech signal peak is used). (8 bits)
- 30 cepstral coefficients. (34 bits)
- Voicing probability. (3 bits)

A 2.4 kbits/s low bit-rate speech coder can thus be achieved.

4.13.2 Number of pitch cycles

To locate the first excitation point at or following each update-point and derive the number of pitch cycles between a pair of such excitation points, a measurement of the average pitch-period is required. The QPSS model uses the "IMBE" initial pitch detector to determine a pitch estimate using a 281 sample "pitch-period analysis window" centred on each update-point. The estimate thus obtained is also, in general, accurate enough for determining the spectral analysis window length (ideally 2.5 pitch periods).

Once a pitch-period estimate has been obtained for the speech around an update-point, it is used to help determine the number of excitation points. This is done by dividing the distance between the first excitation point in the current frame and the first excitation point in the next frame by the average pitch-period.

4.13.3 Location of first excitation point at/after current update-point

The location of the first excitation point at or after the current update-point is found in the current version of QPSS by simply searching for the largest positive peak (assuming appropriate polarity) among the first pitch-period of samples following the current update-point. This is not the true vocal tract excitation point and a small delay is thus introduced between the original and the synthetic speech. More accurate excitation point location techniques are available {Lo, 93}.

4.13.4 An implementation of the QPSS model

An unquantised version of the QPSS model has been implemented [SC, 94B]. This model synthesises speech as the sum of sinusoids whose parameters are specified at update-points separated by 20 ms intervals, i.e. 160 samples with the sampling rate 8 kHz. The input speech is read from a binary file with two bytes representing each sample as specified in section 4.11.1. The QPSS model requires a set of parameters at each update-point. These parameters are derived from segments of the input speech, with two and half pitch period duration, centred on each update-point.

At the decoder, a quadratic phase interpolation formula is defined to derive the instantaneous phase "$\sigma[n]$" of the excitation signal for the fundamental sinusoidal component. This formula uses the information of the number of excitation points and the location of the first excitation point at or after the current update-point. The instantaneous phase and instantaneous frequency for the fundamental sinusoidal component at the end of each frame are calculated to maintain waveform and derivative continuity. The formula is defined to allow all sinusoids to be in phase at the first excitation point at or after the current update-point without incurring rapid variations in instantaneous frequency.

The system phase $\alpha_k[n]$ is derived at each update-point from the encoded short-term spectral coefficients assuming the vocal system is minimum phase. For voiced portions, the synthetic speech is produced from the following formula:-

$$s[n] = \sum_{k=1}^{\text{Lstar}} A_k[n]\cos\left(k \cdot \sigma[n] + \alpha_k[n]\right) \cdot \text{filt}(k\omega[n]) \quad n = 0, \ldots N\text{-}1 \qquad (4.69)$$

where "Lstar" is the number of harmonics. $A_k[n]$ are samples of instantaneous spectral envelope at instantaneous frequency $\omega[n]$ and its harmonics, obtained by differentiating the interpolation formulae for $\sigma[n]$ at the synthesis frame boundaries. Each instantaneous spectral envelope is linearly interpolated between spectral envelopes at the update-points. The function $\alpha_k[n]$ is a cubically interpolated system phase spectrum sampled at each instantaneous fundamental frequency and its harmonics. The function "filt(ω)" is a low-pass filter which has following formula:-

$$\text{filt}(\omega) = \begin{cases} 1 & \omega < 0.95\pi \\ (\pi - \omega)/0.05\pi & \omega \geq 0.95\pi \end{cases} \qquad (4.70)$$

It is used to attenuate the effect of harmonics suddenly appearing or disappearing at frequencies close to π, when the pitch-frequency decreases or increases.
For unvoiced portions of speech or transitions, the overlap-and-add method is used to generate synthetic speech, as described in section 3.6.1.

The QPSS model is at present a voiced speech model only. During periods of silence or unvoiced speech, the estimated pitch period and the interpolated phase are more or less meaningless. Therefore the instantaneous frequency and phase offset at the beginning of the voiced frame immediately following a period of silence or unvoiced speech will be incorrect. The effect of this error will cause misalignment of the pitch cycles in this voiced frame and also, to a progressively less extent, in subsequent voiced frames. A minor but possibly useful modification to QPSS as defined here would be to encode the final excitation point rather than the first in each synthesis frame.

Results of the initial experiments carried out to evaluate QPSS indicate that the QPSS model can produce good quality voiced speech. However, in its current state of development, it does not yet produce speech whose quality is as good as that obtained from STC. We believe it has the potential for doing so with the investment of further development. There are interesting features of QPSS, i.e. the interpolation technique and the pitch-frequency modelling, which make it a worthy candidate for further research.

4.12 Discussion and Conclusions

This chapter surveys the basic ideas of sinusoidal transform coding (STC) which offers a low bit-rate implementation of the FS model discussed in Chapter 3. A pitch frequency, spectral envelope and voicing probability are encoded at each update-point rather than amplitudes, frequencies and phases directly. As originally proposed by McAulay and Quatieri [McQ, 92], cepstral coefficients are used to represent the spectral envelope at each update-point. Phases are derived from this envelope at the decoder via a Hilbert transform under a minimum phase assumption. A computer program implementing the unquantised STC model has been developed and tested in various ways. An alternative idea, called the QPSS, was studied and implemented in preliminary form. This idea was considered to be worthy of further investigation. The effect of applying the STC minimum phase derivation technique to a different sinusoidal coding technique, i.e. IMBE, was also investigated.

The fact that the synthetic speech waveshape is improved by introducing a minimum phase derivation technique into IMBE demonstrates the importance of phase information in the speech synthesised for low bit-rate sinusoidal coders. The synthetic speech waveform and short-term spectrum produced by the STC and QPSS models have, however, revealed some interesting features. One of the most striking features is that the reconstructed magnitude spectra may appear very close to those of the original speech, whilst the sinusoidally synthesised waveshape could be quite different from the speech waveform. The synthesised time-domain waveform tends to have pitch-cycles that decay in amplitude more rapidly than those of the original. The amplitudes of the peaks are also higher in the synthetic speech than in the original. The over-all energy tends to be approximately the same for both waveforms when the speech is voiced. The reason for this is the minimum phase assumption for the vocal system model which tends to concentrate the energy of its impulse response towards the beginning of each pitch cycle.

In practice, the resulting speech quality obtained from STC was found to be quite good but some synthetic effects still exist which may, to some extent, be due to the inaccuracy of the derived system phase. The phase regeneration in STC, and that of QPSS rely on the assumption that voiced speech is produced by applying a suitable excitation signal to a minimum phase vocal system transfer function. Much of LPC theory is based on this assumption also. The fact is that many different excitation signals and transfer functions can combine to give the same signal.

The results obtained from the STC model and QPSS *indicate that it may not be* adequate to derive a minimum phase transfer function with a series of impulses. This may be due to the true shape of glottal pulses, i.e. the sound pressure waveforms produced by the human vocal cords which excite the human vocal tract. Further, in voiced speech, especially female where the pitch period is short, there may be significant overlap in the vocal system response from one pitch cycle to the next. This overlap is known to affect the accuracy of LPC analysis and is also manifest in the frequency domain sampling of the vocal system frequency response at pitch frequency harmonics which are too widely spaced for an accurate spectral envelope to be determined. These factors are considered in the following chapter which aims to improve the accuracy of the phase derivation used by STC, QPSS and other models and also the spectral envelope derivation.

Chapter 5

Spectral envelope and phase optimisation

5.1 Introduction

Questions were raised in the previous chapter concerning the accuracy of the phase derivation technique used in low bit-rate STC. Inaccuracy in the derived phase spectrum as compared with the phase spectrum of the original speech will cause distortion to the reconstructed speech waveform even if the decoded envelope is absolutely correct at the pitch-frequency harmonics. Such distortion is illustrated by the waveform segments in figure 5.1 (solid line for original and dotted line for synthetic speech). Three sources of error which could contribute to this distortion have been identified. Firstly, since the phase spectrum is derived from the spectral envelope, phase error can arise from the inaccuracy of the minimum phase assumption. The transfer function of the vocal system, assumed to model the human vocal apparatus when excited by a spectrally flat signal whose magnitude spectrum corresponds to the spectral envelope of the speech, may not be entirely minimum phase. Secondly, the derived phase spectrum may be adversely affected by the shape of the smoothed spectral envelope between the pitch-frequency harmonics; this is a fundamental problem that will be discussed later. Finally, estimates of the envelope may be affected by pitch-period variations during voiced speech which will therefore also affect the phase spectra derived from the spectral envelopes.

This chapter aims to investigate these three sources of error and to discuss ways of modifying and optimising the envelope in order to reduce the errors and hence to improve the reconstructed speech without increasing the bit-rate.

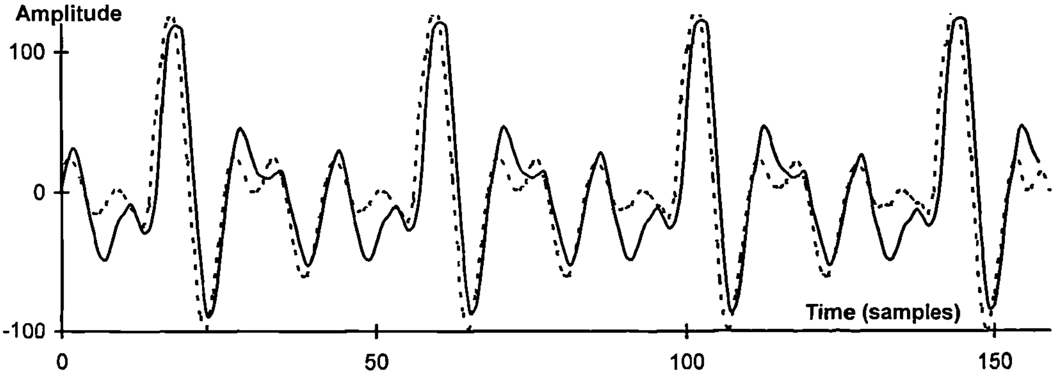


Fig. 5.1 Original speech (Solid line), Synthetic speech (dotted line)

5.2 Modification to the minimum phase assumption

5.2.1 The glottal excitation

The glottal excitation for voiced speech results from the periodic opening and closing of the vocal cords which modulates the air-flow from the lungs. The resulting changes in air volume velocity at the glottis produce sound which becomes the vocal tract excitation. Plotted as a graph against time, the glottal volume velocity (measured in cubic centimetres per second) produces a waveform consisting of a regular succession of pulses as illustrated in figure 5.2 (a). This waveform is pseudo-periodic in the short term, which means that, observed over a short period of time (say 20 to 30 ms), approximately the same pulse shape is repeated at approximately regular intervals of time referred to as the pitch-period. In figure 5.2 (a), the pitch period is 60 samples, i.e. 7.5 ms. Observed over longer periods (more than 50 ms), both the pulse shape and the pitch-period will be seen to be gradually changing. In the time-domain, each glottal pulse will tend to have a characteristic shape with a rather slowly rising leading edge caused by the vocal cords opening relatively slowly. The duration will typically be about half a pitch-period and the pulse will be terminated by a much more sudden trailing edge caused by a sudden reduction in volume velocity as the vocal cords snap sharply together. In the frequency-domain, a Hamming windowed 30 ms section of the glottal excitation waveform will exhibit a harmonic structure as illustrated in figure 5.2 (b) with energy concentrated at the pitch-frequency and its harmonics. The spectral envelope is determined by the spectral shape of an individual glottal pulse which is also illustrated in figure 5.2 (b) (solid line). This tends to have a spectral tilt of approximately -12 dB/octave.

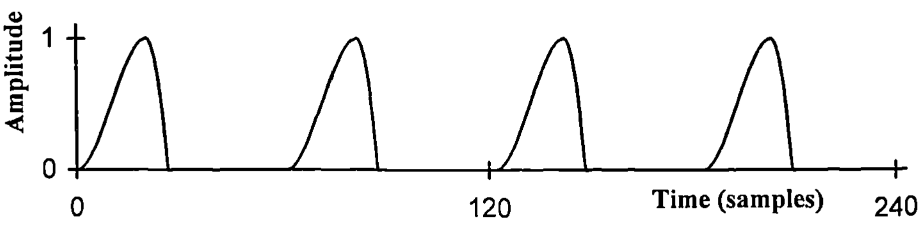


Fig. 5.2 (a) Glottal waveform

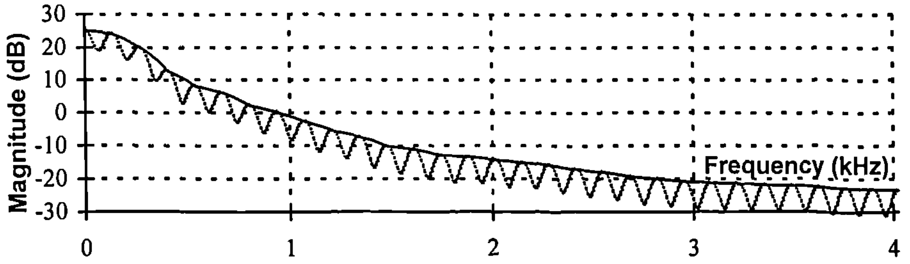


Fig. 5.2 (b) The magnitude spectrum and the envelope

Several different approaches, including Holmes [Hol, 62], Rothenberg [Rot, 73], Miller & Mathews [MM, 63] and Alku & Laine [AL, 89], have attempted to obtain such pulses by inverse filtering of the speech waveform. Also, Rosenberg [Ros, 71] performed perceptual tests to determine a suitable shape for the glottal pulse. The publications show that these methods can give good estimates of the glottal volume flow. One of the most common approximations of these glottal models is the perceptually based Rosenberg pulse shown in figure 5.3. This approximation is governed by three parameters which are the pitch-period, the "opening time" and the "closing time". It is defined by the following formula:

$$U_G(t) = \begin{cases} \alpha\left[3\left(\frac{t}{T_p}\right)^2 - 2\left(\frac{t}{T_p}\right)^3\right] & 0 \le t \le T_p \\ \alpha\left[1 - \left(\frac{t - T_p}{T_N}\right)^2\right] & T_p < t \le T_p + T_N \\ 0 & T_P + T_N < t < P \end{cases} \tag{5.1}$$

where α is the amplitude of the pulse, T_p is the opening time, T_N is the closing time, and P is the pitch-period. Typical values quoted [Rah, 91] for T_p and T_N are 33% and 10% of the pitch period respectively. Figure 5.3 shows a typical Rosenberg glottal pulse, assuming a pitch-period of 60 sampling intervals.

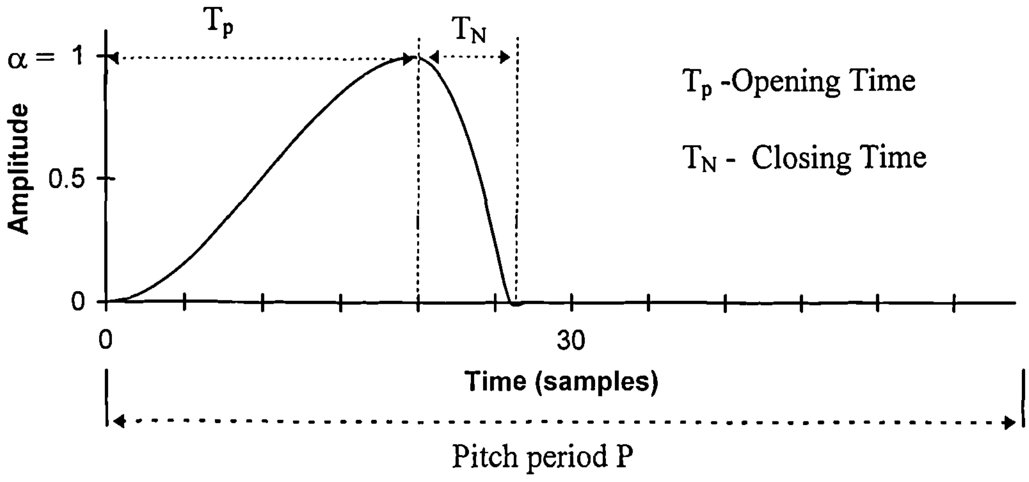


Fig. 5.3 A Rosenberg Pulse

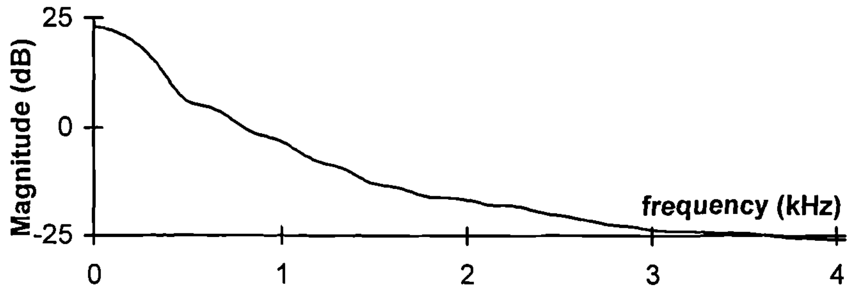


Fig. 5.4 (a) Magnitude spectrum of a Rosenberg pulse

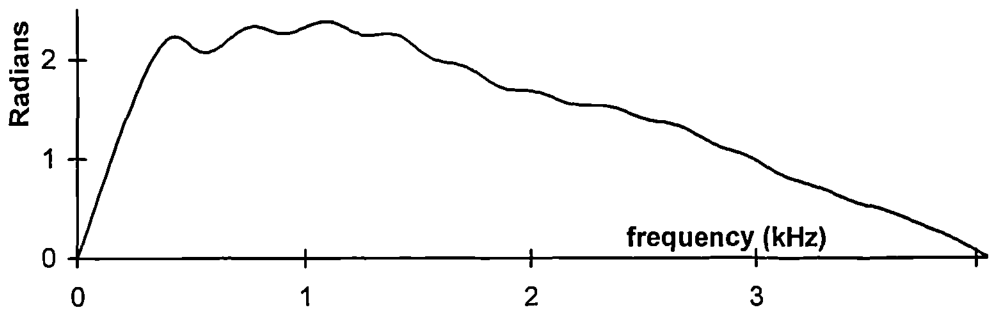


Fig. 5.4 (b) Phase spectrum of a Rosenberg pulse

The magnitude and phase spectra of the Rosenberg pulse may be calculated by applying a Fourier transform to $U_G(t)$ as given by equation 5.1. This was done for the pulse illustrated in figure 5.3 (with $P = 60$, $T_P = 0.33 \cdot P$ and $T_N = 0.1 \cdot P$) by zero-padding to 512 samples and applying a 512 point FFT. The magnitude and phase spectra thus obtained for the pulse are shown in figure 5.4

It is apparent from figure 5.3 that the shape of the Rosenberg pulse is not minimum phase [Good, 95]. A minimum phase signal, in comparison to all possible causal signals with exactly the same magnitude spectrum, will have maximum energy concentrated at the beginning of the waveform [OS, 75]. This means that for all real signals $\{x[n]\}$ with the same magnitude spectrum and for which $x[n] = 0$ for $n < 0$, given any time sampling point, m say, the expression

$$E_m = \sum_{n=0}^{m} x^2[n] \qquad (5.2)$$

will be the largest when the signal is minimum phase [OS, 75, Chapter 7]. The fact that the Rosenberg pulse is not minimum phase may be further confirmed by calculating the complex cepstrum of a single glottal pulse and observing its non-causality. The cepstrum obtained from the pulse shown in figure 5.3 is shown as figure 5.5 where 256 complex cepstral coefficients (from -128 to 127) are plotted.

This graph was obtained by zero-padding a 60 sample Rosenberg pulse to produce a 256 point sequence $\{x[n]\}$, applying a 256 point FFT to produce $\{X[k]\}$ and computing the complex cepstrum from this frequency-domain sequence. The direct method would be to calculate the sampled log-magnitude and phase spectra of $\{X[k]\}$ and then to take the 256 point inverse-FFT of a spectrum whose real part is this log-magnitude spectrum and whose imaginary part is this phase spectrum. The direct method can be programmed successfully [Mat, 95] , but it suffers from the complication that the phase spectrum of $\{X[k]\}$ must be "unwrapped" before it may be used as the imaginary part of a spectrum. Unwrapping means that apparent phase discontinuities arising from the calculation of arguments in the range $-\pi$ to π must be eliminated by the addition of a suitably chosen integer multiple of 2π to each phase.

To avoid the complications of phase unwrapping, the complex cepstral coefficients were in practice calculated using the "logarithmic derivative" method [OS, 75]. This method applies a 256 point FFT also to thc imaginary sequence $\{-j \cdot nx[n]\}$ to obtain $\{X'[k]\}$, i.e. frequency-domain samples of $X'(e^{j\omega})$ which is the derivative of $X(e^{j\omega})$ with respect to ω. Noting that the derivative of $\log_e(X(e^{j\omega}))$ with respect to ω is $j \cdot e^{j\omega} \cdot X'(e^{j\omega}) / X(e^{j\omega})$, a 256 point inverse-FFT is applied to the frequency-domain sequence $\{X[k]/X'[k]\}$ to obtain:

$$v[n] = \frac{1}{256} \sum_{k=0}^{255} \frac{X'[k]}{X[k]} e^{j(2\pi/256)nk} \qquad n = 0, 1, \ldots, 255 \qquad (5.3)$$

and the complex cepstrum $\{\hat{x}[n]\}$ is then calculated from the following formula:

$$\hat{x}[n] = \begin{cases} -\frac{1}{jn} v[n] & : n \neq 0 \\ \frac{1}{256} \sum_{k=0}^{255} \log_e |X[k]| & : n = 0 \end{cases} \qquad (5.4)$$

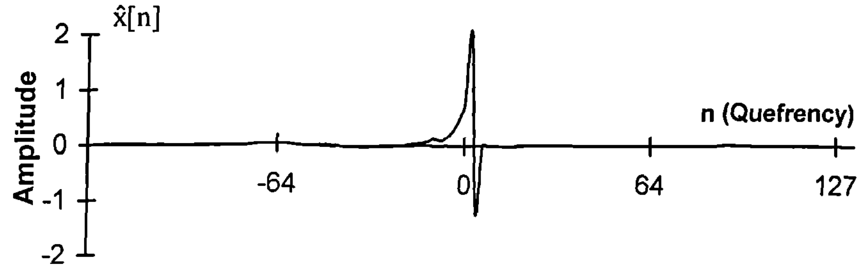


Fig. 5.5 Complex cepstrum of a Rosenberg pulse

The complex cepstrum shown in figure 5.5 clearly has energy at negative values of quefrency, in fact more than at positive values. Therefore it is non-causal which means that the Rosenberg pulse is non-minimum phase. Applying this cepstral analysis to segments of natural speech and LPC residual segments also confirms that they are not completely minimum phase.

5.2.2. The voiced speech production model

Voiced speech may be modelled by a sequence of impulses driving a glottal filter, an all-pole vocal tract model and a lip-radiation filter as illustrated below:

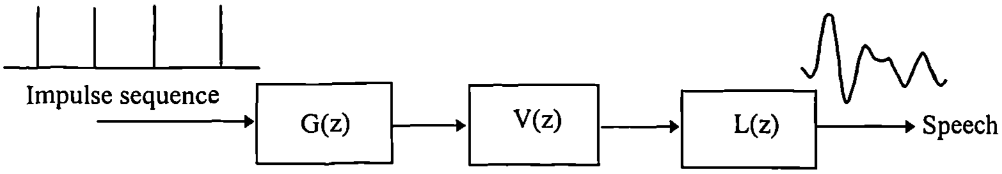


Fig. 5.6 A vocal system model

V(z) is normally assumed to be an all-pole filter, and L(z) may be assumed to be a differentiator with transfer function:

$$H(z) = 1 - \alpha z^{-1} \tag{5.5}$$

where α is a number close to 1, typically 0.95 [DPH, 93] [MG, 76]. V(z) and L(z) are both minimum phase, though L(z) is clearly not all-pole.

The magnitude response of G(z), i.e. $|G(e^{j\omega})|$, is often assumed [RS, 78] to be close to the magnitude response of a second order all-pole integrator with transfer function:

$$I(z) = \frac{1}{(1-\beta_1 z^{-1})(1-\beta_2 z^{-1})} \tag{5.6}$$

The constants β_1 and β_2 are again close to 1, and one of them, β_1 say, is often assumed [DPH, 93] [MG, 76] to be approximately equal to α so that the low-pass effect of one of the poles of I(z) on the magnitude spectrum of the speech is to some degree cancelled out by the high-pass filtering of the lip radiation model.

However, the shape of the glottal excitation pulses as proposed by [Ros,71] [Rot, 73] suggest that G(z) is not minimum phase. Their shapes may be better modelled by any of the excitation models mentioned in last section, such as Rosenberg pulses. To obtain an even simpler approximation, their shapes may be modelled as impulse responses of I(z) time-reversed and appropriately delayed. This time-reversal does not effect the magnitude spectrum of the signal, but its phase spectrum becomes the phase response of I(1/z) , disregarding the fact that I(1/z) is not a stable transfer function. This simply means that if $\phi(\omega)$ is the phase response of I(z), the phase spectrum of the time-reversed impulse response will be $\phi(-\omega)$ with an added linear phase component.

To confirm this, let {i[n]} be the real impulse response of I(z) as illustrated in figure 5.7 (a). Then $I(e^{j\omega})$ must be the DTFT of {i[n]}, i.e.

$$I(e^{j\omega}) = \sum_{n=-\infty}^{\infty} i[n]e^{-j\omega n} \tag{5.7}$$

The DTFT of the time-reversal impulse response {i[-n]}, as illustrated in figure 5.7 (b) is:

$$\sum_{n=-\infty}^{\infty} i[-n]e^{-j\omega n} = \sum_{n=-\infty}^{\infty} i[n]e^{j\omega n} = I(e^{-j\omega}) = \overline{I(e^{j\omega})} \tag{5.8}$$

where the bar denotes complex conjugate. Therefore the DTFT of the non-causal sequence {i[-n]} has magnitude $|I(e^{j\omega})|$ and phase $\phi(-\omega) = -\phi(\omega)$. *To relate these* spectra to a glottal pulse, i[-n] must be delayed as illustrated in figure 5.7(c), the samples occurring for n<0 being then considered negligibly small. The delay adds a linear phase component, $D\omega$ for some constant D, to the phase spectrum of i[-n]. This linear phase component can normally be disregarded since it corresponds only to a delay which , in STC, is dealt with by the onset time calculation.

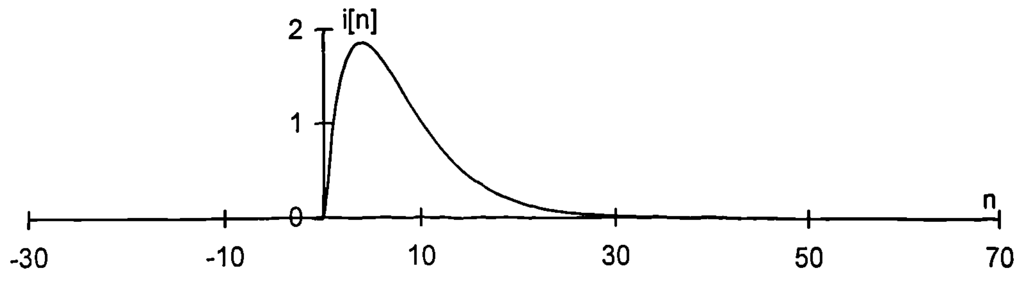


Fig. 5.7 (a) Impulse response of I(z)

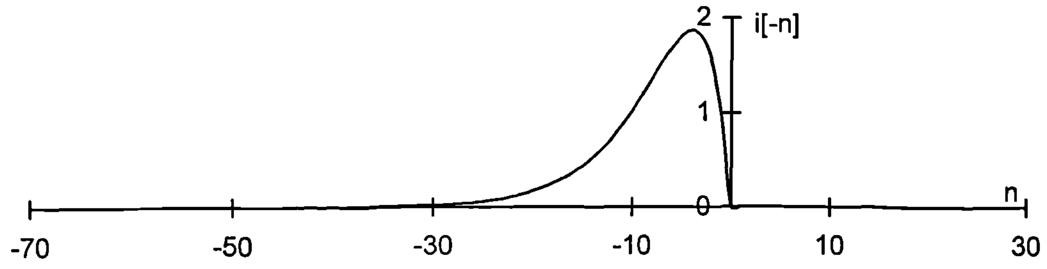


Fig. 5.7 (b) Time-reversed impulse response of I(z)

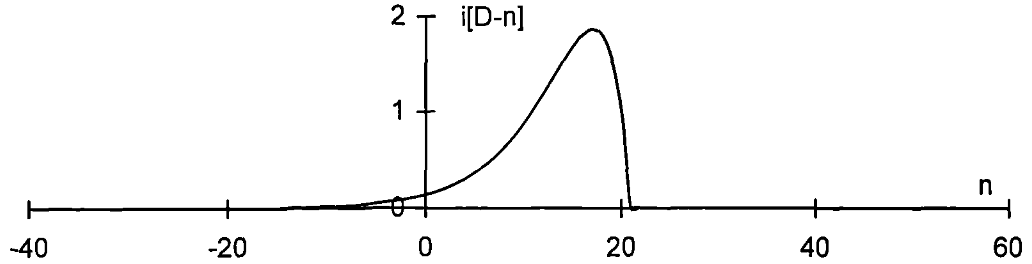


Fig. 5.7 (c) Delayed version of i[-n]

The magnitude and phase spectra of a voiced speech pitch-cycle can therefore be modelled as the magnitude and phase responses, respectively, of the following transfer function:

$$G(z)V(z)L(z) \tag{5.9}$$

where $|G(e^{j\omega})| \approx |I(e^{j\omega})|$ and $\mathrm{Arg}(G(e^{j\omega})) \approx D\omega - \mathrm{Arg}(I(e^{j\omega}))$ for some linear phase component $D\omega$. Strictly, G(z) cannot be realised by an all-pole filter, though it is often realised by an all-zero FIR filter. It is useful to visualise this speech model in terms of the non-LTI system diagram in figure 5.8.

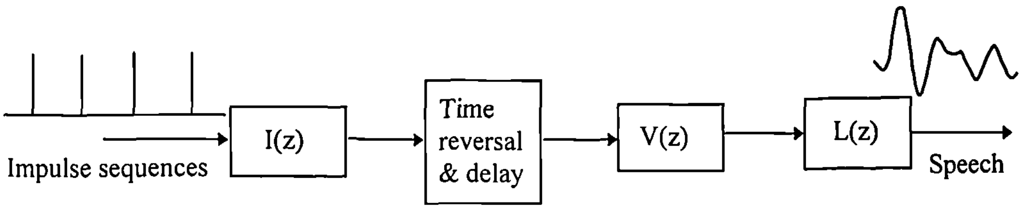


Fig. 5.8 A vocal system model

5.2.3 All-pass LPC inverse filter

If LPC analysis is applied to voiced speech assumed to conform to the model in figure 5.8, an all-zero "minimum-phase" LPC inverse filter will be obtained whose zeros will, in principle, cancel out the poles of the vocal tract model V(z). The LPC inverse filter will also remove the effect of G(z) on the speech magnitude spectrum by placing a zero at $z = \beta_2$. The effect on the speech magnitude spectrum of the pole of I(z) at $z = \beta_1$ is assumed to be approximately cancelled out by the lip-radiation zero of L(z) at $z = \alpha$. However, the effect on the speech phase spectrum of the poles of I(z) at $z = \beta_1$ and $z = \beta_2$ and the zero of L(z) at $z = \alpha$ will not be removed by the inverse filter.

The transfer function of the LPC inverse filter will be:

$$\frac{1}{I(z)V(z)L(z)} \qquad (5.10)$$

It follows that according to the model in figure 5.6, the LPC residual will be a pseudo-periodic series of impulses filtered by the following transfer function:

$$\frac{G(z)V(z)L(z)}{I(z)V(z)L(z)} = \frac{G(z)}{I(z)} \qquad (5.11)$$

Since $|G(e^{j\omega})| = |I(e^{j\omega})|$ this transfer function is all-pass, i.e. its gain is unity for all ω. Its phase response is:

$$\mathrm{Arg}(G(e^{j\omega})) - \mathrm{Arg}\,(I(e^{j\omega})) = -\phi(\omega) - \phi(\omega) - D\omega \quad \text{for some D} \qquad (5.12)$$

The effect of this transfer function on the LPC residual may be removed by augmenting the LPC inverse filter by an all-pass transfer function, $A_p(z)$ say, whose phase response is $2\phi(\omega)$, disregarding any linear phase component. Given that I(z) is defined by equation 5.6, such an all-pass transfer function is:

$$A_p(z) = \frac{(\beta_1 - z^{-1})(\beta_2 - z^{-1})}{(1-\beta_1 z^{-1})(1-\beta_2 z^{-1})} \qquad (5.13)$$

It may be verified that $|A_p(e^{j\omega})| = 1$ for all ω and that the phase response of $A_p(z)$ is:

$$\begin{aligned}\mathrm{Arg}(A_p(e^{j\omega})) &= 2\arctan\left(\frac{\sin\omega}{\beta_1-\cos\omega}\right)+2\arctan\left(\frac{\sin\omega}{\beta_2-\cos\omega}\right)+2\omega \\ &= 2\phi(\omega)+2\omega\end{aligned} \qquad (5.14)$$

where $\phi(\omega)$ is the phase response of I(z). A pole-zero plot for $A_p(z)$ is shown in figure 5.9. To avoid confusion, $A_p(z)$ will be referred to as an "all-pass inverse" filter which conforms to the terminology: "LPC inverse" filter.

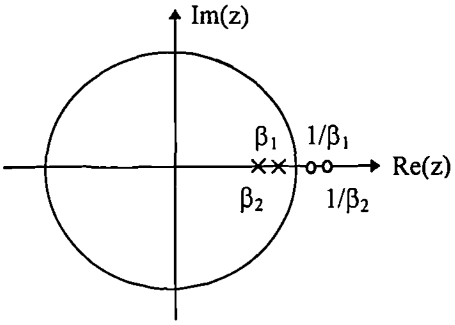


Fig 5.9 Pole-zero plot for the "all-pass inverse" filter $A_p(z)$

The phase response of $A_p(z)$ is shown in figure 5.10 for different values of β (β = 0.8, 0.9, 0.95), assuming $\beta_1 = \beta_2 = \beta$. The group-delay responses are shown in figure 5.11 for the same values of β. These graphs indicate that the phase and group-delay spectra of the residual will be most strongly affected at frequencies below about 1 radians/sample, i.e. below about 1.3 kHz.

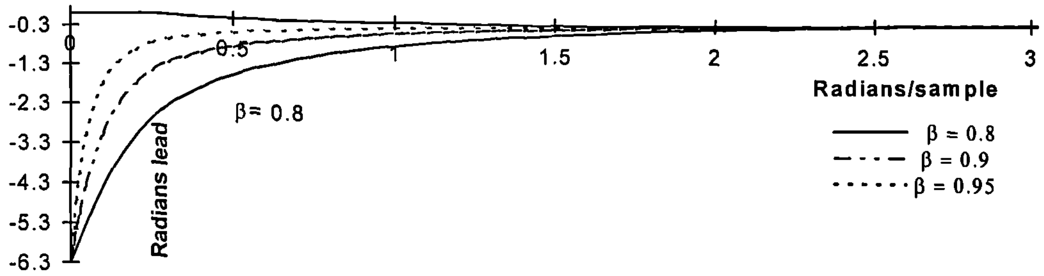


Fig. 5.10 Phase response of all-pass inverse filter $A_p(z)$

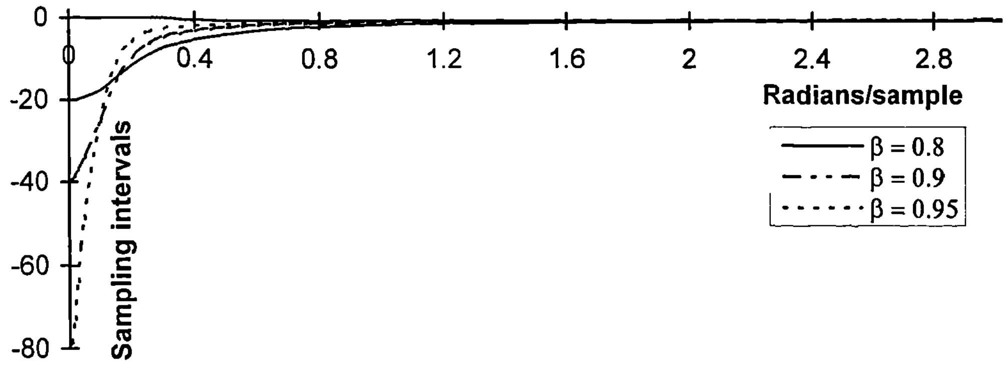


Fig. 5.11 Group delay response of all-pass inverse filter

According, to the ideas presented above, the magnitude spectrum of the LPC residual should be spectrally flat but its phase spectrum should not be expected to be purely linear phase. Therefore the residual will not be impulse-like. In theory, the phase spectrum may be linearised by passing the residual through a second order all-pass transfer function. The poles of this all-pass transfer function should be dependent on the effect of the glottal transfer function and the lip-radiation. In practice, suitable approximations to these poles may be made by an optimising search or curve-fitting procedure which aims to match either the phase response, or the group-delay to corresponding measurements obtained from the residual spectrum.

5.2.4. Modification of the minimum phase assumption for STC

The ideas outlined in the previous section may be applied to improve the accuracy of the phase derivation for STC. Two possible ways of doing this have been investigated. The first way is to remove the effect of a Rosenberg pulse from each spectral envelope before deriving a phase spectrum from it. The second way is to correct the phase spectrum derived from an unmodified envelope by including the effect of an all-pass inverse filter.

The approach in both cases is based on a separation of the voiced speech production model into a spectrally flat excitation signal and a vocal system model. If the speech production process is observed in the frequency-domain, this is equivalent to expressing the speech spectrum as the spectrum of a pseudo-periodic series of pulses multiplied by the frequency response of the vocal system transfer function which comprises G(z), V(z) and L(z).

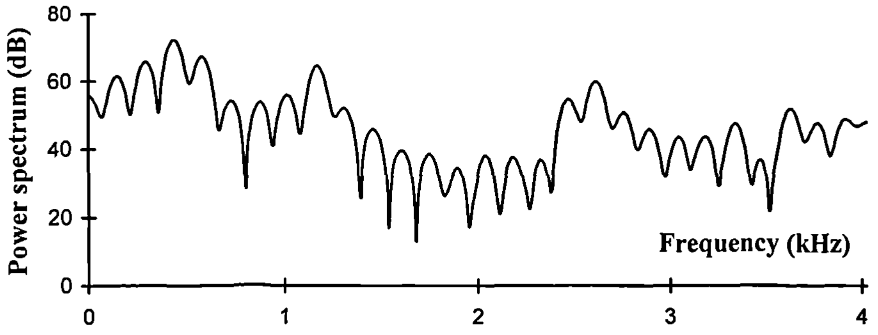


Fig 5.12 (a) Power spectrum of a segment voiced speech

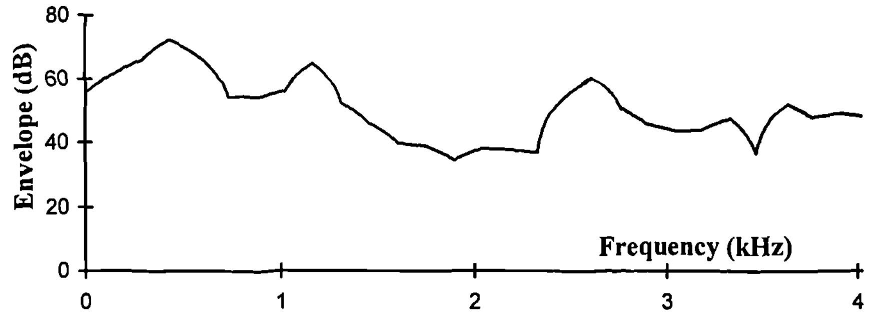


Fig 5.12(b) The spectral envelope of the speech segment

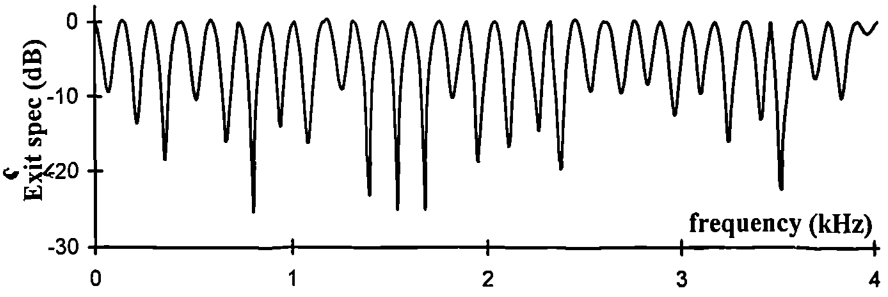


Fig 5.12(c) The spectrum of the series of impulses

Figure 5.12 (a) shows the magnitude spectrum of a segment of voiced speech. This spectrum is considered to be the multiplication of two spectra shown as figures 5.12 (b) and (c). Figure 5.12 (b) is the magnitude spectrum of the vocal system transfer function which should be identical to the envelope of the speech magnitude spectrum. Figure 5.12 (c) is the spectrum of a periodic series of discrete time unit impulses.

5.2.5 Rosenberg pulse modification method

If the short-term spectral envelope of the input speech is divided by the magnitude spectrum of a suitably shaped Rosenberg pulse, and if the phases of the input speech at harmonic frequencies have subtracted from them corresponding phases of the same Rosenberg pulse, the remaining magnitude and phase spectra are made more likely to correspond to a minimum phase transfer function. The effect of the non-minimum phase glottal excitation has thus been eliminated. The phase spectrum derived from the modified spectral envelope by the assumed Hilbert transform relationship is therefore likely to be closer to the modified true phase spectrum than was the case without the modifications. The STC decoder can therefore be modified to derive, at each update-point, the required phase spectrum for its sinusoidal model by adding together:

(i) a minimum phase component derived from the modified spectral envelope,
(ii) the phase spectrum of a Rosenberg pulse,
(iii) a linear phase component as usual.

The best performance is likely to be achieved if the Rosenberg pulse parameters are optimised to minimise a "phase error" defined as the squared Euclidean distance between original and derived phases at the pitch-frequency harmonics:

$$\varepsilon = \frac{1}{L}\sum_{k=1}^{L}[\phi_{o}[k] - \phi_{d}[k]]^{2} \tag{5.15}$$

where $\phi_o[k]$ and $\phi_d[k]$ are the original phase and the derived phase at the k^{th} harmonic and the L is the number of harmonics.

However, it has been found that considerable improvement in the accuracy of the derived phases as compared with standard STC may be obtained with a simplified approximation parameterised only by the pitch-period. The effectiveness of the phase correction is related to the "ambiguity bit" used in [McQ, 92] which is no longer required.

5.2.6 All-pass filtering method

An alternative way of compensating for inadequacies in the minimum phase assumption and thus improving the STC phase regeneration procedure is to derive the minimum phase response from the unmodified spectral envelope spectrum as usual at the STC decoder and then to subtract from it the phase response of an all-pass inverse filter whose transfer function is:

$$A_p(z) = \frac{(\beta_1 - z^{-1})(\beta_2 - z^{-1})}{(1-\beta_1 z^{-1})(1-\beta_2 z^{-1})} \tag{5.16}$$

The phase response of $A_p(z)$, $\varphi(\omega)$ say, given by equation 5.14. An all-pass filtering operation is therefore achieved in the frequency domain thus changing the magnitude and minimum phase spectra of I(z)V(z)L(z) to the magnitude and non-minimum phase spectra of G(z)V(z)L(z).

The required values of β_1 and β_2 will be strongly dependent on the pitch-period. We could get an idea of how to choose β_1 and β_2 by analysing Rosenberg pulses, and this would make the "all-pass" STC technique very similar to the technique outlined in the previous section. However, there may be advantages in trying to find the best values of β_1 and β_2 by an analysis-by-synthesis/optimisation technique carried out at the encoder.

To investigate the use of the all-pass technique with fixed parameters, the following initial experiment was carried out:

(i) The minimum phase spectrum was derived from the unmodified envelope as received at the decoder.

(ii) A value of say β=0.8 was specified, assuming $\beta_1 = \beta_2 = \beta$.

(iii) The phase spectrum $\varphi(\omega)$ was derived for the "all-pass inverse" filter.

(iv) $\varphi(\omega)$ was subtracted from the minimum phase spectrum

The results of this experiment are summarised in figures 5.13 and 5.14 and compared with results obtained from similar experiments using the Rosenberg pulse method. Figure 5.13 shows the average "mean phase error" over 240 frames of voiced speech. The "mean phase error" is defined as the squared Euclidean distance between original and derived phases at the pitch-frequency harmonics, divided by number of harmonics. The five bars represent the mean phase error obtained using different methods listed as follows:

(i) Minimum phase method

(ii) Rosenberg pulse approximation

(a) $T_p = 33\%$ and $T_N = 10\%$

(b) Optimised T_p and T_N by minimising the phase error

(iii) All-pass filter method

(a) $\beta = 0.8$

(b) Optimised β by minimising the phase error

Figure 5.14 shows the results of an informal listening test. A 10 seconds speech segment was analysed and synthesised by STC using four different methods. For each method the same spectral envelope representation was used with a different phase model. The speech segment included male and female voices. The phase models are listed below:

(i) True phase transmitted

(ii) Minimum phase method

(iii) Rosenberg pulse approximation with fixed $T_p = 33\%$ and $T_N = 10\%$ of the pitch period.

(iv) All-pass filter compensation with $\beta_1 = \beta_2 = 0.8$

Eleven subjects were used. Each subject had to give an order of preference for the perceived quality of the speech. A score was given to each method according to the order of preference. The scores were as follows: first = 4; second = 3; third = 2 and the fourth = 1. Figure 5.14 shows the total score obtained for each of the four phase regeneration methods.

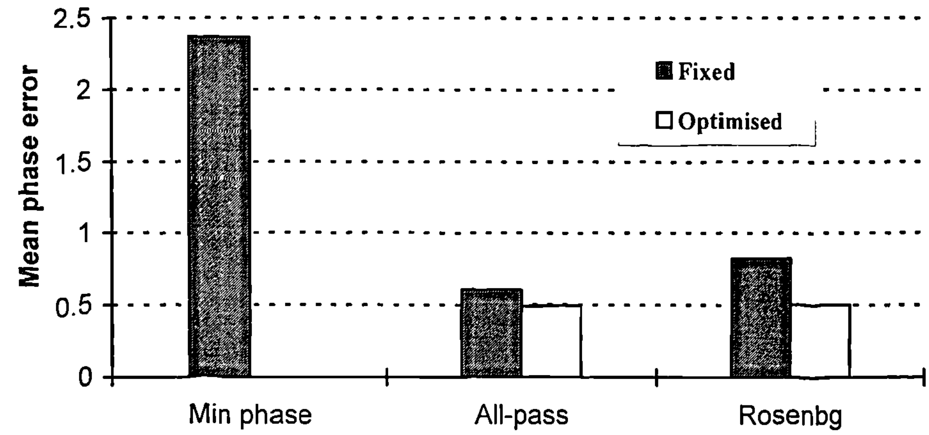


Fig 5.13 Comparison of mean phase error for three methods

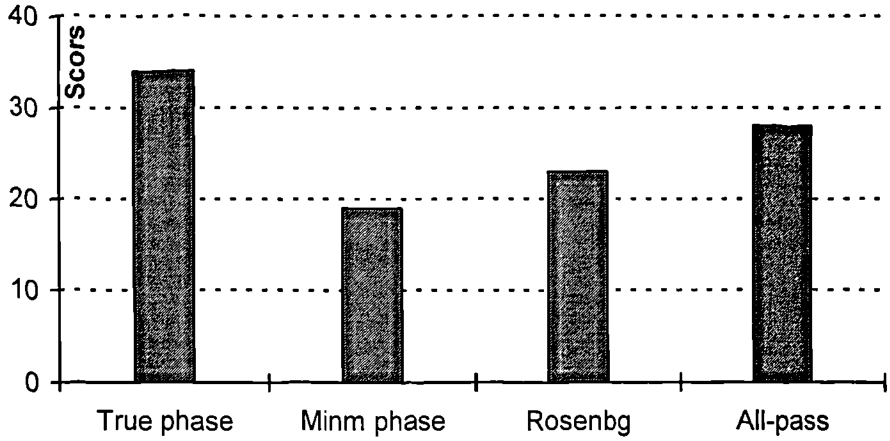


Fig. 5.14 informal listening for different phase modelling

Results indicate that both the Rosenberg pulse method and the all-pass filter method appear to be capable of giving better synthetic speech quality without increasing the required bit-rate. If the bit-rate can be increased slightly, optimised values of β_1 and β_2 can be encoded to give even better performance and smaller phase error than is obtained with fixed values of these parameters.

The modifications discussed above aim to compensate for the non-minimum phase characteristics of the glottal excitation. Distortion in the decoded speech also arises from other effects, including inaccuracies in the shape of the vocal system spectral envelope derived at the decoder. By improving the shape of the spectral envelope both the magnitude and phase of the decoded speech may be further improved. Ways of doing this will be discussed in the next section.

5.3 Optimisation of the spectral envelope

5.3.1. The effectiveness of the envelope derivation

The vocal tract is a resonant cavity situated between the glottis and the lips, which may be excited by sound generated at the vocal cords to produce voiced speech. The way it modifies the excitation sound to produce the speech sound is determined by its transfer function V(z) and a "lip-radiation" transfer function L(z). The frequency response, $V(e^{j\omega})$ of the vocal tract has a number of resonant peaks or "formants", whose centre frequencies and bandwidths are determined by the physical shape of the cavity. This shape is modified during speech by the control of the lips, tongue and jaw. An all-pole transfer function has been found to successfully model the properties of the vocal tract, which are very close to those of a lossless acoustic tube [Hol, 88]. The gain response of a typical vocal tract cavity, as modelled by a 12th order all-pole transfer function, is shown in figure 5.15 (a).

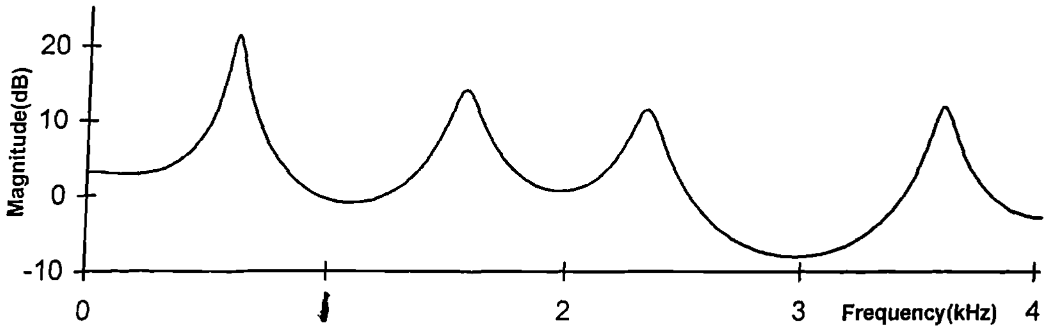


Fig. 5.15 (a) Gain response of 12th order all-pole vocal tract model

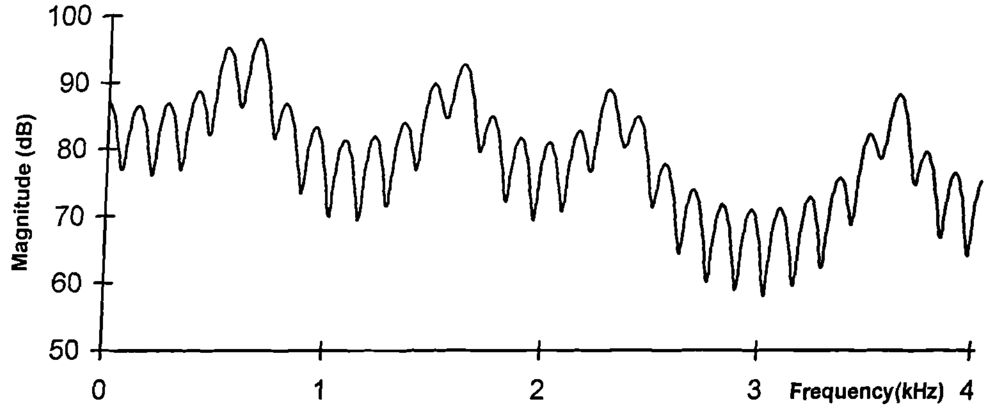


Fig. 5.15 (b) Spectrum of voiced speech

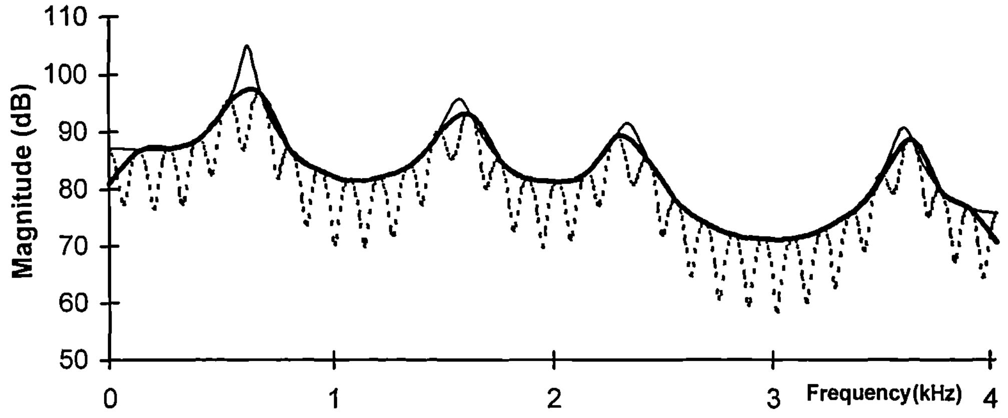


Fig. 5.15 (c) Inaccuracies due to cubic spline interpolation

This gain response $|V(e^{j\omega})|$ should , in principle, be similar in shape to the envelope of the short term spectrum of the voiced speech being produced (see figure 5.15 (b)). This spectrum was obtained for a 160 point Hamming windowed segment of speech zero-padded to 1024 points, when the pitch period remained fixed. In comparison to the LPC spectrum, it should be expected to be increased in energy by the excitation signal, and is also spectrally tilted by the pole $z = \beta_2$ of I(z) as discussed in Section 5.2.2.

In STC, the spectral envelope is derived by smoothly interpolating the harmonic peaks across frequency band. The results of two interpolation techniques are shown in figure 5.15 (c): the heavy solid line is cubically interpolated and the light solid line was obtained by fitting an all-pole envelope to the magnitude samples at pitch frequency harmonics. Cubic spline interpolation is reliable for low pitched male speech where the number of pitch frequency harmonics, i.e. samples of the envelope, is reasonably large. However it is noticeable that for higher pitched voiced speech, e.g. female speech, the frequency-domain interpolation or smoothing tends to produce rather flattened formant peaks, especially when the peaks lie between pitch-frequency harmonics. This is illustrated in figure 5.15.(c). Although it may be argued that the spectral envelope need be known accurately at the decoder only at the pitch-frequency harmonics, there are two possible objections to this argument. The first objection is that distorting the shape of the spectral envelope between the harmonics will distort the magnitude-phase relationship at the harmonics.

Experiments show that the estimated phase spectrum is particularly sensitive to the shape of the derived envelope in the vicinity of formant peaks, and this distortion will be more severe if the pitch frequency is high, such as in female voiced speech. The second objection is that when voiced speech is synthesised at the decoder, the pitch frequency may be changed from one update-point to the next. Therefore the envelope should be sampled at the evolving harmonic frequencies to obtain the true amplitude for each component of the sinusoidal model. These evolving frequencies will lie between the pitch-frequency harmonics as specified at the update-points. We have therefore investigated the possibility of deriving a more accurate spectral envelope from measurements at pitch-frequency harmonics, than is obtained by cubic spline interpolation. The *criterion of accuracy is that the phases at the pitch-frequency harmonics, as derived from the envelope, via a Hilbert transform, should* be as close as possible to the true phases of the speech, disregarding linear phase components.

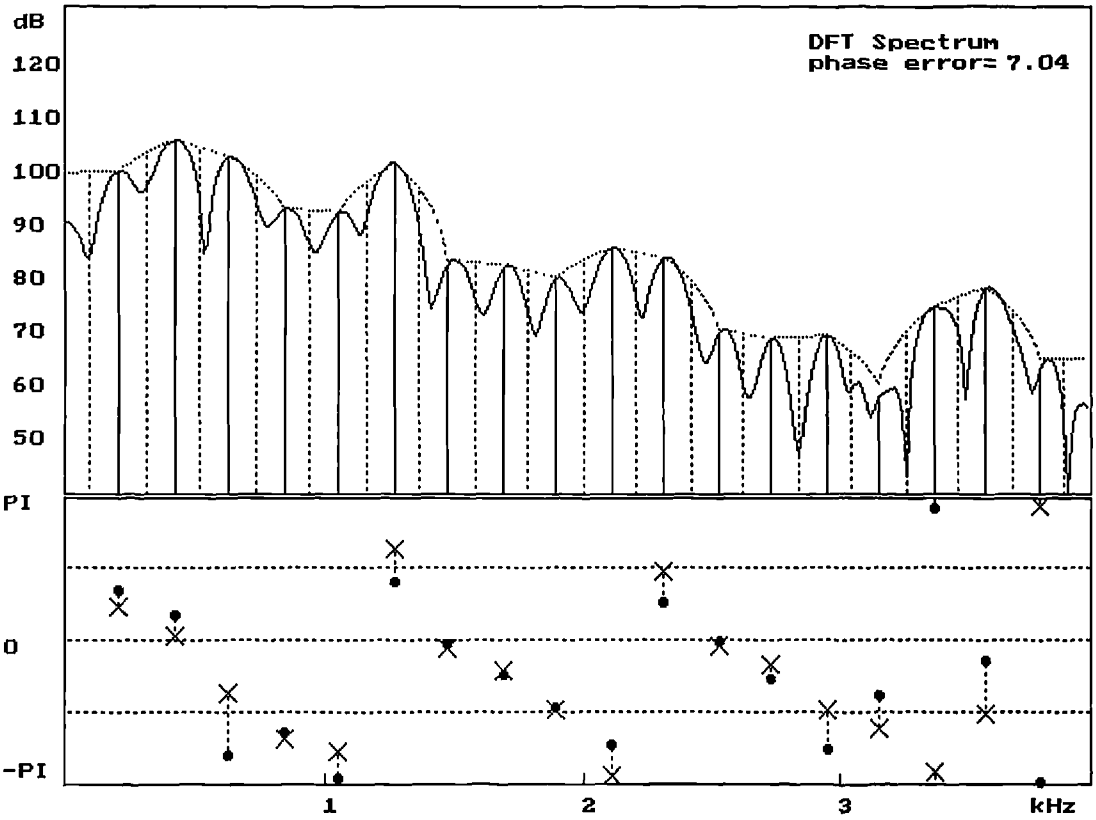


Fig. 5.16 (a) Power spectrum, the envelope and phase differences

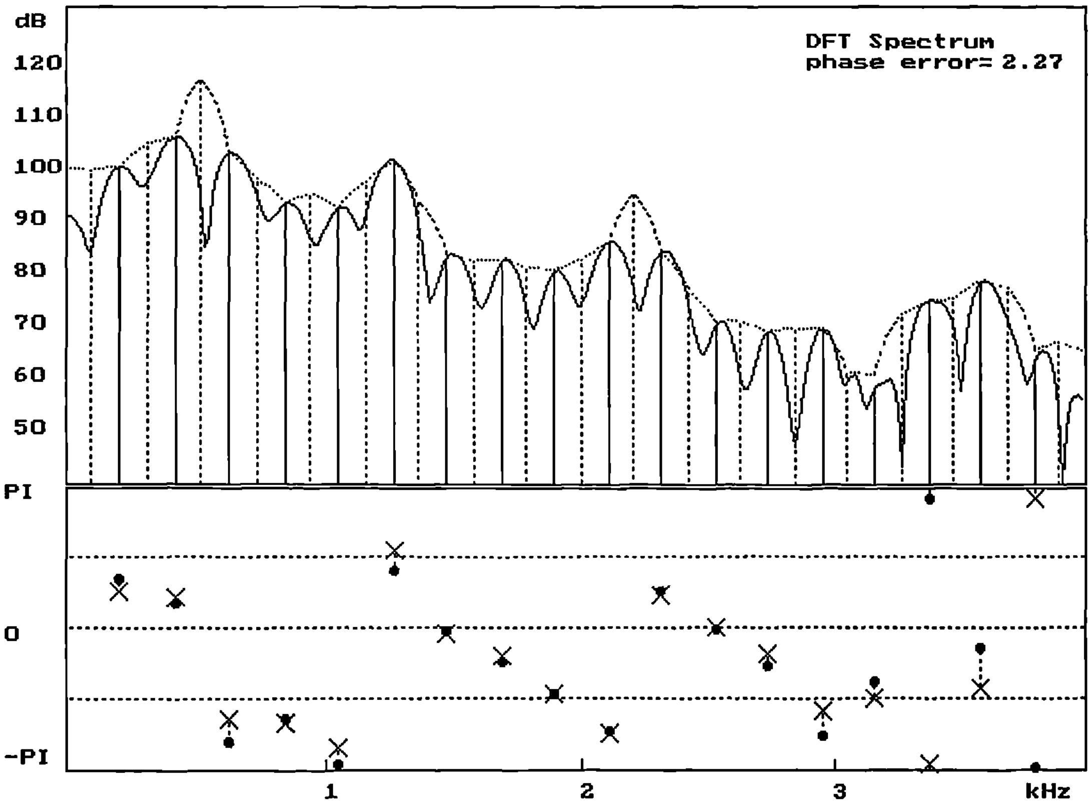


Fig. 5.16 (b) Power spectrum, modified envelope and improved phase differences

5.3.2. Optimisation of the envelope shape

To investigate errors in the Hilbert transform derived phase spectrum that are due to the non-ideal envelope shape between pitch-frequency harmonics, we introduced, mid-way between each pitch-frequency harmonic, an additional frequency at which the envelope is to be specified. An optimisation procedure was then applied to calculate the envelope magnitudes at these new frequencies. The optimisation procedure iteratively changes the envelope at these mid-way frequencies to reduce as far as possible the phase error at the pitch-harmonic frequencies. The envelope values at the pitch-harmonic frequencies do not change during the iterative procedure. The phase error is defined, as equation 5.15, as the squared Euclidean distance between the original and derived phases at the pitch harmonics.

To compute the phase error, a linear phase component must be subtracted from each derived phase, the corresponding delay being included as an optimisation parameter. This procedure was found to be successful in further reducing the derived phase error when combined with the non-minimum phase correction procedure described earlier. Figure 5.16 (a) shows the magnitude spectrum of a segment of voiced speech for which the pitch-frequency is about 200 Hz. Solid vertical lines indicate the pitch harmonics, and a spectral envelope derived by cubic spline interpolation is shown fitted to these harmonics. A dotted vertical line is shown mid-way between each pair of solid lines. It is at these mid-way frequencies that the spectral envelope is to be optimised, but in figure 5.16 (a), no optimisation has been applied and the amplitudes are simply samples of the normal cubic spline envelope. To derive the phase spectrum the magnitude spectrum was divided by that of a Rosenberg pulse, as described in Section 5.2.5, a discrete Hilbert transform was applied and the phase spectrum of the Rosenberg pulse was then added to the remain phase spectrum. The true phases at the pitch harmonics are shown as dots in the phase graph below the DFT magnitude spectrum. These are compared with the derived which are shown as crosses. It may be seen that the derived phases are generally quite close to the true phases. The phase error in this case is 7.04.

Figure 5.16 (b) is the same speech magnitude spectrum but with a spectral envelope whose amplitudes have been modified at the mid-way frequencies. It may be seen that spectral envelope peaks have been created by the optimisation procedure between certain pitch harmonics, and that deriving samples of the phase spectrum (crosses) from this modified envelope, again with Rosenberg pulse correction produces values which are considerably closer to the true phases than was the case in figure 5.16 (a). The phase error has been reduced by the optimisation procedure to 2.27.

The use of this optimisation procedure has allowed us to confirm that accurate phase derivation is possible with suitable interpolation for producing the spectral envelope. The cubic spline interpolation technique used by conventional STC to derive the envelope has clear deficiencies in that it tends to flatten formant peaks which lie between pitch harmonics. The capacity to raise these peaks appears beneficial. The optimisation technique used in these investigations was conceptually very simple, and computationally very time-consuming. To take advantage of the conclusions drawn, an improved and computationally less intensive algorithm is required.

One possibility is to employ the discrete all-pole (DAP) modelling proposed by Makhoul [EIM, 91]. There is much evidence that an all-pole transfer function can accurately model the effect of the vocal tract, though conventional LPC analysis techniques (e.g. Durbin's Algorithm) become less accurate for high pitch frequencies or more resonant formants. The principle of the DAP technique is to determine, by optimisation, the coefficients of an all-pole transfer function whose gain response will be as close as possible to the spectral envelope of the speech at the pitch harmonics. *It uses a modified form of the Itacura-Saito error measure [IS, 70]* [MG, 76], which is

$$E_{Is} = \frac{1}{L}\sum_{m=1}^{L}\frac{P(\omega_m)}{\hat{P}(\omega_m)} - \ln\frac{P(\omega_m)}{\hat{P}(\omega_m)} - 1 \qquad (5.17)$$

where $P(\omega_m)$ for m=1, 2, ..., L, is the given discrete spectrum i.e. the spectral envelope of the speech sampled at the pitch harmonics ω_m. $\hat{P}(\omega_m)$ is the all-pole model gain response at frequencies ω_m and L is the number of harmonics. The all-pole models required to minimise the Itacura-Saito error measure E_{Is} is calculated by an iterative procedure which will be discussed in Chapter 6.

The DAP model saves computational complexity in comparison to the optimisation method mentioned above. It also has the advantage that the envelope can be represented by LSF parameters which are readily vector quantised. The disadvantage of representing the DFT envelope by the DAP model is that the envelope is restricted to being that of an all-pole system rather than a general minimum phase system (with zeros as well as poles). The DAP envelope will not necessarily have precisely the right amplitude at each pitch harmonic , and it is optimised according to criteria based only on the magnitude spectrum i.e. not on phase. An advantage of representing an envelope by LSP coefficients is that the phase spectrum is derived from an all-zero inverse filter thus eliminating the need to compute a Hilbert transform.

It has been seen that the way the magnitude spectrum of speech is interpolated to derive the envelope will affect the accuracy of phase derivation. However the shape of the envelope will also be affected by other effects including frequency spreading due to pitch variation. The effect of pitch-frequency variation on the spectral envelope and the derived phase spectrum will be discussed in the next section.

5.4. Frequency spreading due to pitch variation

It is known that significant variations in pitch-frequency can occur even in an analysis frame-length as short as two and half pitch-periods. The effect is to produce frequency spreading in regions of the short-term DFT spectrum which smears out the spectral harmonics and results in local reductions of up to 3 dB in the spectral envelope. In general, the spreading will be apparent in frequency bands the centre frequencies and bandwidths of which depend on the degree of pitch-frequency variation. The frequency spreading would become even more serious if the FFT analysis window were widened, for example to make a sinusoidal coder more robust to background (source) noise.

A low bit-rate sinusoidal coder will encode this distorted envelope and at the decoder will derive from it the amplitudes of the sinusoidal components used to synthesise speech. The amplitudes will therefore be affected by the distortion and also the phases since they are derived from the envelope.

At the decoder, the sinusoidal frequencies are interpolated and therefore will change from one update-point to the next, thus causing the same type of apparent spreading as occurs with the original speech. If the synthesised speech were spectrally analysed there would appear to be a doubling of the energy reduction due to spreading once at the analysis stage (when finding the spectral envelope) and then again at the synthesiser. If several sinusoidal coders were connected in tandem (an important consideration in telephony) the energy reductions would accumulate in the same frequency bands, producing greater and greater distortion of the spectral envelope and the phases derived from it.

By simply boosting the reduced spectral envelope in certain frequency ranges, amplitude and phase error may be reduced. A means of predicting and compensating for the frequency spreading from pitch-period estimates at the update-points is now proposed.

5.4.1. Effect of frequency spreading on an artificial excitation signal

Experiments with artificial speech can be performed to demonstrate how frequency changes in the pseudo-periodic series of impulses assumed to excite the vocal tract can affect the speech magnitude spectrum as measured by a DFT. These experiments show how different degrees of pitch variation cause frequency spreading in different frequency ranges. Consider an excitation signal segment of length L which is approximately equal to 2.5 pitch-periods. Assume that this segment includes three vocal tract excitation points the second of which coincides approximately with the centre of the window, as illustrated in figure 5.17. In this illustration, the pitch period is approximately 60 sampling intervals and the window length L is 150 sampling intervals. Let the time interval between the first and second impulses be P-m samples and let the time interval between the second and third impulses be P + m. When m is non-zero, the effect of a changing pitch-period may be observed. It may be argued that placing an impulse close to the centre of the 2.5 pitch-period analysis frame models the worst case that will occur in practice since the effect of pitch frequency variation will be most noticeable.

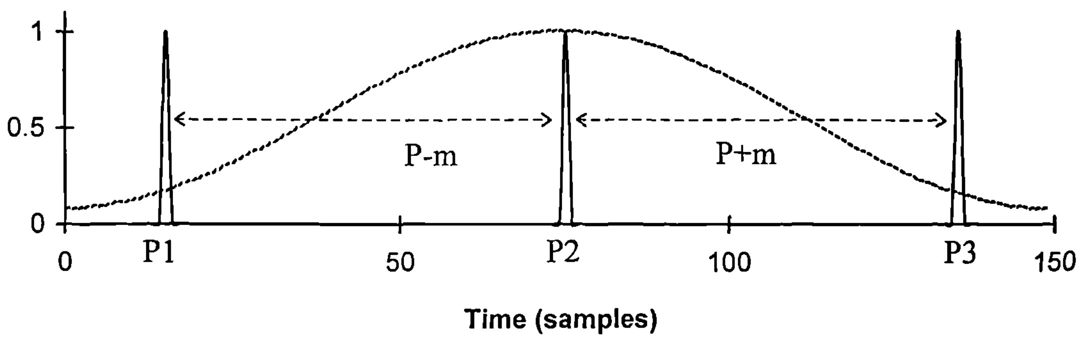


Fig. 5.17 Hamming windowed impulses

Segments as illustrated in figure 5.17 were synthesised for various values of m, P and L. Each was multiplied by an L-sample Hamming window, zero-padded to 1024 samples and FFT analysed. The magnitude spectra obtained for various values of m when P and L were fixed at 60 and 150 samples respectively are shown in figure 5.18 (a) to (f). The values of m used were as follows:

(a) m = 0; (b) m = 0.5; (c) m = 1; (d) m = 1.5; (e) m = 2; (f) m = 2.5;

Figure 5.18 (a) is a harmonic spectrum of fundamental frequency 133 Hz with the expected flat spectral envelope for a fixed pitch-frequency. When multiplied by the vocal system gain response, the resulting speech spectrum will have an envelope whose shape is close to that gain response.

When m = 0.5, the pitch-period increases by one sampling interval over the analysis frame. In this case, as may be seen in figure 5.18 (b), a strongly harmonic structure is clear at frequencies up to about 3 kHz, but a spreading of the harmonics occurs above this frequency resulting in an attenuation of the envelope by up to about 3 dB. The attenuation will be propagated to the speech spectrum causing the estimate of the vocal system gain response to be similarly attenuated.

When m = 1, the pitch-period changes by two samples over the analysis frame. Figure 5.18 (c) shows that the spreading now occurs in the middle of the frequency range, i.e. around 2 kHz, with clear harmonic structure occurring at lower and higher frequencies. A loss of up to 3 dB in the centre of the envelope will again be propagated to the speech spectrum thus distorting its spectral envelope around

2 kHz. In figures 5.18 (d), (e) and (f) the spreading and envelope distortion are seen in different frequency ranges depending on the value of m. In each of these figures there is more than one frequency range with reduced energy.

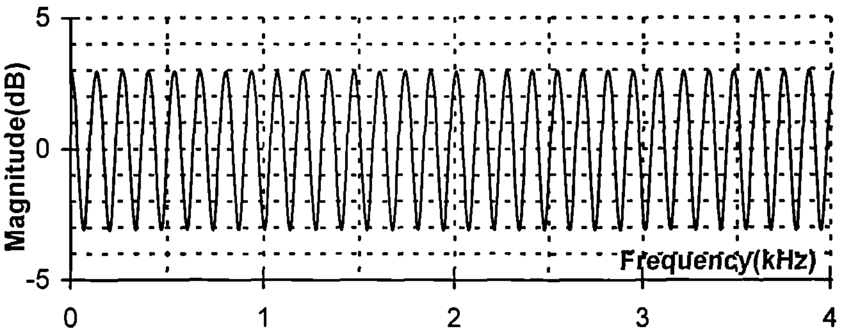


Fig. 5.18 (a) Magnitude spectrum of excitation when m=0

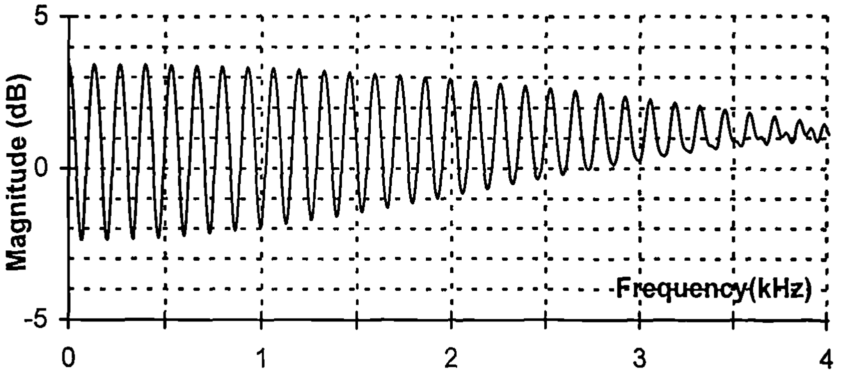


Fig. 5.18 (b) Magnitude spectrum of excitation when m=0.5

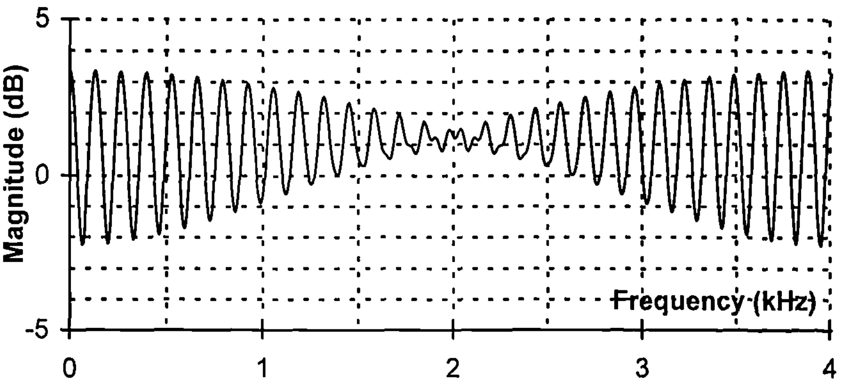


Fig. 5.18 (c) Magnitude spectrum of excitation when m=1

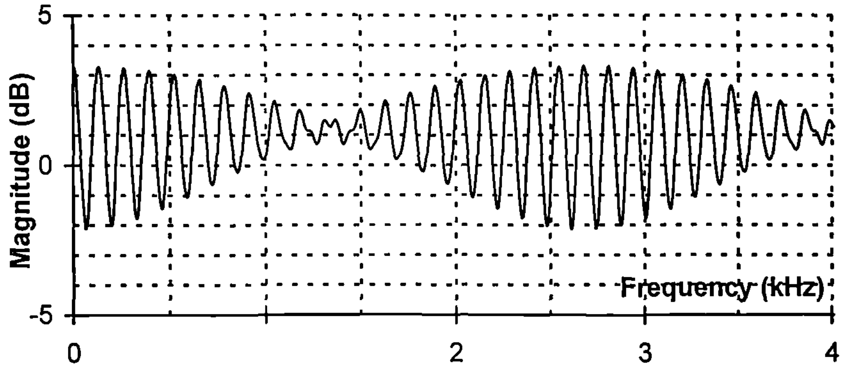


Fig. 5.18 (d) Magnitude spectrum of excitation when m=1.5

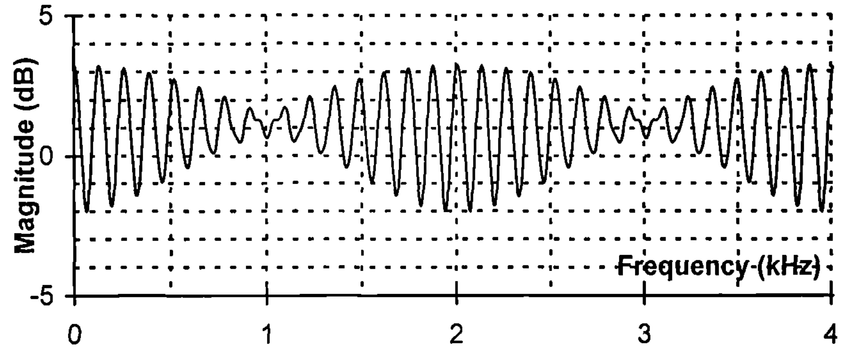


Fig. 5.18 (e) Magnitude spectrum of excitation when m=2

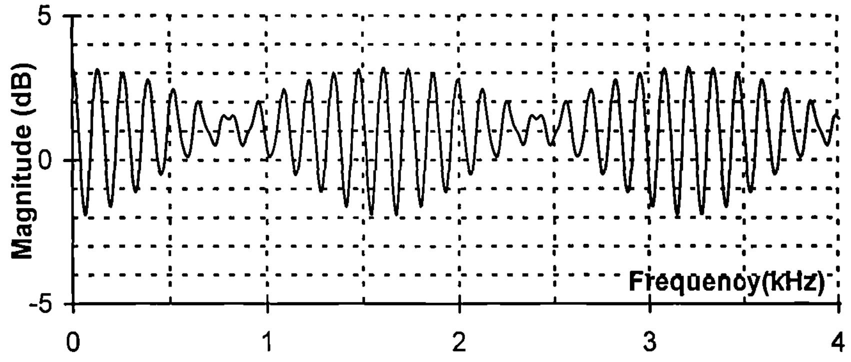


Fig. 5.18 (f) Magnitude spectrum of excitation when m=2.5

5.4.2. Prediction of the spreading

The L point DFT of the excitation signal in figure 5.17, multiplied by a Hamming window w[n], may be written as:

$$X[k] = w[P_1]e^{-j2\pi P_1 k/L} + w[P_2]e^{-j2\pi P_2 k/L} + w[P_3]e^{-j2\pi P_3 k/L} \qquad (5.18)$$

Assuming $w[P_2]$ to be approximately one, and $w[P_1]$ to be approximately equal to $w[P_3]$ = w say,

$$X[k] \approx e^{-j2\pi\cdot P_2\cdot k/L}\left(1 + w\,[e^{j2\pi\cdot(P-m)\cdot k/L} + e^{-j2\pi\cdot(P+m)\cdot k/L}]\right) \qquad (5.19)$$

where P is the average pitch-period and the pitch-period is assumed to change by 2m sampling intervals over the analysis window. It follows that:

$$|\,X[k]\,| \approx \left|\,1 + 2w\,e^{-j2\pi mk/L}\cos(2\pi\cdot P\cdot k\,/\,L)\right| \qquad (5.20)$$

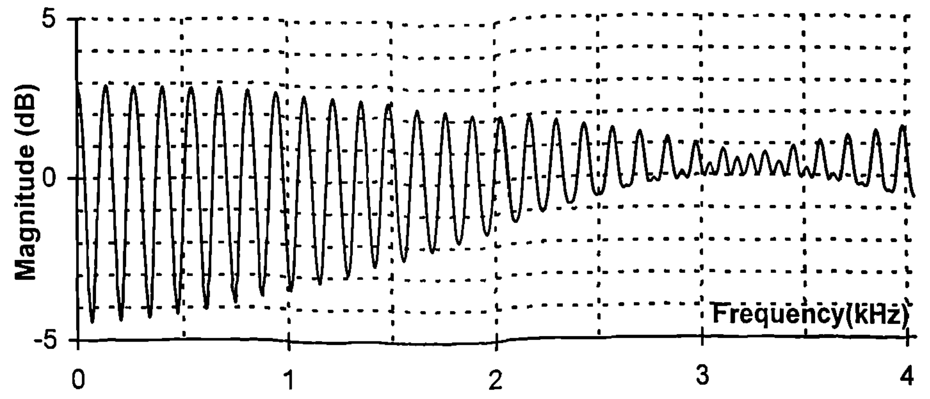


Fig. 5.19 Estimated magnitude spectrum when P=60, m=0.625 & w=.2

A graph of this approximation is shown in figure 5.19 for the case where P = 60, m = 0.625 and w = 0.2. It may be seen that the worst frequency spreading is around the frequency 3.2 kHz and if the envelope were to be deduced from the spectrum, a loss of up to about 2 dB around this frequency would be incurred. Equation 5.20 correctly predicts that, in general, there may be more than one frequency spreading band and that the centre of these bands will occur where 2πmk/L = π/2, 3π/2, 5π/2, etc., i.e. where k = L/(4m), 3L/(4m), etc. This is because the minimum value of the envelope will occur when 1 and $2w\,e^{-j2\pi mk/L}$ are in quadrature in equation 5.20, i.e. when $e^{-j2\pi mk/L}$ is purely imaginary. The corresponding frequencies in Hertz are

$f_s/(4m)$, $3f_s/(4m)$, etc. where f_s is the sampling frequency. From equation 5.20, it may also be deduced that the spectral envelope that would be obtained by interpolation of the harmonic peaks is:

$$\left|\Xi[k]\right| = \left|1+2we^{-j2\pi mk/L}\right|$$

$$= (1 + 4w^2 + 4w\cos(2\pi mk/L))^{0.5} \qquad (5.21)$$

Now $|\ \Xi\ [k]\ |$ may be obtained by sampling the gain response of a transfer function $\Xi\ (z) = 1 + 2\ w\ z^{-m}$ which is minimum phase when $w < 0.5$ and $m > 0$, (even when m is not an integer). Therefore the phase spectrum that would be obtained by applying a Hilbert transform to $|\ \Xi\ [k]\ |$ is equal to the phase response of Ξ (z), i.e.

$$\text{Arg}\ \Xi\ (e^{j\omega}) = \arctan(-2w\sin(2\pi mk/L)/(1+ 2\ w\cos(2\pi mk/L)) \qquad (5.22)$$

This phase spectrum is shown in figure 5.20 for the case where m = 0.625, w = 0.2 and L=150. The phase is reasonably linear up to about 3 kHz but then becomes non-linear because of the effect of spreading on the envelope. For an ideal excitation with constant pitch-frequency, the phase spectrum would be zero at all frequencies. The non-ideal phase spectrum will be added to the vocal system phase spectrum when speech is produced.

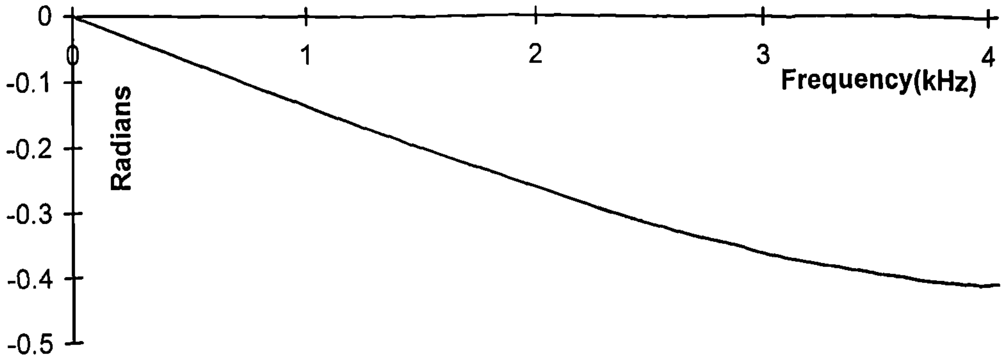


Fig. 5.20 Excitation phase spectrum derived from envelope approximation (m=0.625, w=0.2, L=150)

5.4.3. Compensation of the effect of frequency spreading

Frequency spreading has been seen to affect the accuracy of the DFT spectral envelope of speech and also the phases derived from the envelope. To compensate for this effect, a compensation filter may be applied to the DFT spectrum before the spectral peak interpolation procedure. By equation 5.21, the compensation filter should have the gain response:

$$G[k] = \frac{1}{\left|1 + 2we^{-2\pi jmk/L}\right|} \tag{5.23}$$

The parameters required for this filter are m and w. A suitable value of m may be obtained by estimating the rate of change of the pitch-period from the values calculated at the update-points. If the pitch-periods at the previous, current and next update-points, spaced N samples apart, are M_{-1}, M_0 and M_1 respectively, this estimated value of m could be:

$$m = \left(\frac{M_1 - M_{-1}}{2N}\right) M_0 \tag{5.24}$$

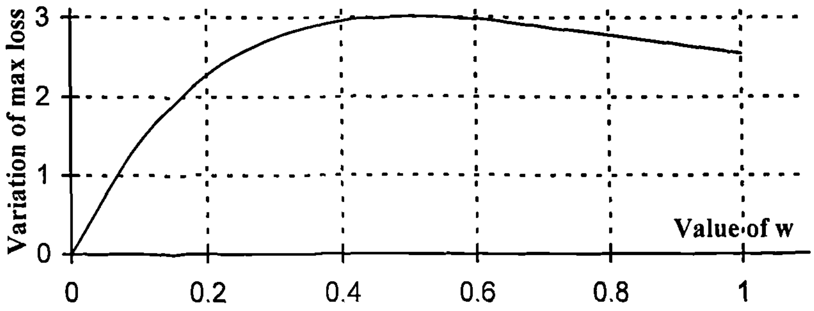


Fig. 5.21 The variation of maximum loss

To devise a suitable value of w, the variation of the maximum loss (V_m) can be measured since the value of the |Env[k] | changes from 1+2w to $(1+4w^2)^{0.5}$ within 0 to π radians. It is defined as:

$$V_m = 10 \log_{10}(1 + (4\,w)/(1+4w^2)) \tag{5.25}$$

It may be observed that the variation of the maximum loss is fairly gradual over the range w = 0.2 to w = 1, shown as figure 5.21. Noting that values of w less than 0.08 do not occur with a Hamming window, and in practice values of w at impulse position are most likely to lie between 0.2 to 0.8, it is reasonable to assume a representative value of 0.5 for w.

The filtering effect may be achieved by multiplying the DFT magnitude spectrum by G[k] in the DFT frequency domain. The effect is illustrated in figure 5.22, where (c) is the spectrum of the artificial speech generated by passing regular pulses through an all-pole model, (b) is the spreading affected spectrum obtained with m = 0.625 and (a) is the spectrum which has been compensated. It may be seen that the compensation filter enhances the spectral envelope around 3.2 kHz. The resulting envelope (a) fits the peaks of the regular pulse excited spectrum (c) much better than that obtained previously (b).

It can be seen that the distortion of the DFT spectral envelope which is caused by pitch-frequency variation can be reduced by this compensation technique.

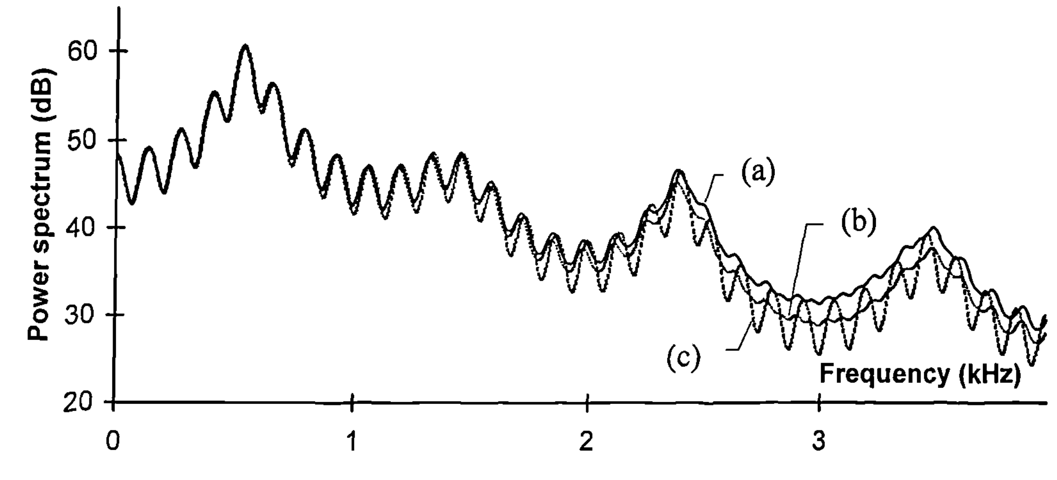


Fig. 5.22 (a) Compensated envelope

(b) Power spectrum due to the frequency spreading

(c) Power spectrum

5.5 Conclusions

This chapter has analysed three sources of distortion which affect the STC magnitude and phase spectra and proposes ways of reducing this distortion. Applying these modifications to STC synthesised speech produces waveforms closer in shape to the original and perceptually better sounding. The modification techniques are applicable to sinusoidal coders in general, and similar improvements have been observed with others, such as the IMBE coder.

The techniques described in this chapter have been built into a modified version of the original STC model which is to be the basis of fully quantised STC coders. The quantisation process and the implementation of these fully quantised speech coders in simulation is the subject of the next chapter.

Chapter 6

Quantisation of the STC coder

6.1 Introduction

Vector-quantisation [Pal, 90] [MRG, 85], also known as codebook quantisation, is a powerful data compression technique. It can usually encode more information at a given bit-rate than scalar-quantisation. In vector-quantisation, identical copies of one or more codebooks, each containing a set of vectors, are stored at both the encoder and the decoder. Each codebook is a table or array of vectors. If the table has V entries, each entry being a K-dimensional vector, it is a K-dimensional, V-level codebook. When K is equal to one, the vector-quantiser becomes a quantiser for individual samples, i.e. a scalar-quantiser. To quantise a given vector, it is necessary to compare it with each of the codebook vectors to find the closest one according to some criterion. The index number (or address) of the table entry is encoded to identify the chosen codebook vector. At the decoder, the codebook vector with the encoded index is read from an identical copy of the codebook.

The implementation of vector-quantisation procedures has three aspects:

(i) The design of the vector-quantiser (i.e. the codebook training).

(ii) The vector-quantisation process itself (i.e. codebook search).

(iii) Assessment of the performance of the vector-quantiser in operation.

This chapter surveys the principles of vector quantisation as required for the design of an efficient low bit-rate STC speech coder. The parameters to be encoded for a STC model are the pitch-frequency, the voicing probability and the spectral envelope. Both scalar quantisation and vector quantisation techniques are used to quantise these parameters. Procedures for quantising the parameters will be presented.

6.2 The design of a vector-quantiser

6.2.1 Basic principles

The design of a vector-quantiser is concerned with selecting an appropriate set of vectors for the codebook entries. To do this, a large set of training vectors must be provided which are assumed to be representative of the range of vectors that are likely to be encountered when the vector-quantiser is used in practice. For a γ-bit vector quantiser, the aim is to find a set 2^{γ} of codebook vectors such that the sum of the distances between each training vector and its closest codebook vector is as small as possible.

Usually, the required set of codebook vectors are generated using a “clustering” algorithm applied to a large set of training vectors obtained from extended segments of natural speech. The clustering algorithm must divide the training vectors among a number of different “cells”, a cell being a set of training vectors which may be considered close to one-other according to some measure of “distance” or “distortion”. Each cell will have a unique “centroid”. This is defined as the vector with the property that the sum of the distances or distortions from all training vectors of the cell to this particular vector is a minimum over all possible vectors.

The allocation of a particular training vector to a cell is decided according to its “distance” from a single “reference vector” for the cell, or according to a measure of the distortion that would be caused to some quantity (e.g. the short term speech spectral envelope) by replacing the training vector by the reference vector. The reference vectors should ideally be the centroids of the cells. However, these centroids cannot be known in advance since they depend on the cell allocations. Therefore an iterative process is required which starts with reference vectors believed to be not too dissimilar from the centroids of the cells they will create. The iterative process divides the training vectors into cells according to the current reference vectors (i.e. "clusters" the training vectors around the current set of reference vectors), calculates the centroids of the cells thus created, redefines the reference vectors according to these centroids and continues in this fashion until the reference vectors and the centroids become sufficiently close to each other.

The "distance" measure adopted for vector-quantisation could simply be the Euclidean distance between two vectors, i.e. if $\underline{v} = \{v_1, v_2, \ldots, v_k\}$ is a training vector and $\underline{u} = \{u_1, u_2, \ldots u_k\}$ is a reference vector, the Euclidean distance between $\underline{v}$ and $\underline{u}$ is :

$$d(\underline{u},\underline{v}) = \sqrt{(v_1 - u_1)^2 + (v_2 - u_2)^2 + \cdots + (v_k - u_k)^2} \qquad (6.1)$$

Alternative distance and distortion measures are often used, such as the squared Euclidean distance measure:

$$d(\underline{u},\underline{v}) = (v_1 - u_1)^2 + (v_2 - u_2)^2 + \cdots + (v_k - u_k)^2 \qquad (6.2)$$

A third measure often used for vector quantisation of LPC ladder filter coefficients is a form of the Itakura-Saito distortion measure [MRG, 85] [ElM, 91] applied to LPC coefficients. It is referred to as the IS distortion measure. The IS measure of distortion that occurs when a given P × 1 vector $\underline{u}$ of LPC ladder coefficients is changed to the P × 1 reference vector $\underline{v}$ of LPC ladder coefficients is defined as :

$$\mathrm{IS}(\underline{u},\underline{v}) = (\underline{u} - \underline{v})^{\mathrm{T}} \mathrm{R}(\underline{v})(\underline{u} - \underline{v}) \qquad (6.3)$$

where $\mathrm{R}(\underline{v})$ is the $\mathrm{P} \times \mathrm{P}$ Toeplitz normalised auto-correlation matrix [Lo, 93] corresponding to the set of LPC filter coefficients $\underline{v}$. T denotes "matrix transpose". This is said to be "the IS distortion from $\underline{u}$ to $\underline{v}$". The matrix R($\underline{v}$) is calculated with unity diagonal elements from the LPC coefficient vector $\underline{v}$. IS($\underline{u}$, $\underline{v}$) is referred to as a "distortion" rather than a "distance" measure since it is not necessarily symmetric i.e. IS($\underline{v}$, $\underline{u}$) is not necessarily equal to IS($\underline{v}$, $\underline{u}$).

The position of the centroid for a given set of training vectors forming a cell depends on the distance or distortion measure used. Different centroids would be obtained for the Euclidean distance measure than would be obtained for the squared Euclidean distance measure. For the squared Euclidean distance measure, the centroid of a cell is very easily calculated. It is the mean vector of the cell, i.e. the vector whose elements are the mean values of the corresponding elements of all the training vectors in the cell. For the IS distortion measure, the centroid of a cell can be easily calculated by Durbin's algorithm [Lo, 93].

Better results are often obtained when distance or distortion measures are based on more direct measurements of the consequences of using particular vectors. In speech, perceptually based distance measures are often used. The IS distortion measure is better than the Euclidean distance measure for training vector quantisers for LPC ladder coefficients as it is to some extent perceptually based.

6.2.2 The LBG algorithm

Several algorithms are available for designing vector quantisers given a large set of training vectors [GWZ, 92]. The most popular is the LBG algorithm with centroid splitting (LBG-CS) which is based on a paper published in 1980 by Linde, Buzo and Gray [LBG, 80]. For the LBG-CS algorithm, a set of training vectors are initially defined as a single cell whose centroid is calculated such that the sum of the distances or distortions from all the training vectors to the centroid is minimised. This centroid is then split into two "split-centroid" vectors by creating a new vector equal to the original centroid plus a small vector (or the original centroid multiplied by a scalar constant close to unity). These split-centroid vectors are used as the initial reference vectors for the training of a 2-level vector-quantiser.

The training vectors are re-clustered into two cells around the reference vectors. This is done by calculating the distance or distortion from each training vector to each of the two reference vectors and assigning the training vector to a cell associated with one or other of these reference vectors depending on which distance or distortion measurement is smaller. At this stage, the clustering into two cells is not optimal and an iterative procedure is required to optimise this two-cell clustering.

To begin this iterative procedure, the centroid of each of the two cells is calculated. The distance or distortion from each training vector (regardless of which of the two cells it is originally in) to each centroid is calculated. The centroids are then used as reference vectors and each training vcctor is re-assigned to one or other of the reference vectors according to which has the lowest distance or distortion. A new arrangement of the training vectors into two cells is thus obtained. A new centroid is then calculated for each cell and the iteration process continues.

At each iteration, the sum of the distance or distortion measurements from each of the training vectors to the centroid of its assigned cell is calculated. The iteration will continue until this summed distortion is reduced to an acceptably low level or changes very little from one iteration to the next. Assuming that the summed distortion of the current iteration is D and the summed distortion of the previous iteration is D', a suitable condition for termination is:

$$\frac{D'}{D} - 1 \leq \varepsilon \qquad (6.4)$$

where ε is a suitably small constant. Typically, ε =0.001.

At this stage, two "2-level" centroid vectors will have been found which are suitable as the codebook vectors for a "2-level" vector quantiser. The next stage of the LBG-CS algorithm is to "split" these two centroids into four split-centroids and to use these to initiate the training of a “four-level” quantiser. The training procedure is similar to that used for the 2-level quantiser; i.e. each training vector is associated with its nearest split-centroid thus defining four cells of training vectors, the centroid of each of these four cells is computed , the training vectors are re-clustered around these centroids used as reference vectors to obtain four re-defined cells, the centroids of these redefined cells are computed and the process continues iteratively until the total distortion ceases to be significantly reduced at each step. The four centroids obtained at the end of this iterative process are suitable as the codebook vectors for a 4-level (2-bits) vector quantiser.

The four centroids thus obtained may be split again to initiate the training of an "eight-level" quantiser by a similar iteratively re-clustering procedure. The centroid splitting procedure will continue until the number of the centroids becomes equal to the required number of quantisation levels. The final centroids become the required codebook vectors.

The LBG-CS algorithm for training a 2^{Y} level codebook is summarised as follows:

1) Set a distortion threshold ε.
2) Calculate the single centroid of the whole set of training vectors.
3) Split each centroid into two split-centroids. Set these as "reference vectors".
4) Calculate the distance or distortion from each training vector to each of the reference vectors and thus re-cluster the training vectors into cells; one cell for each reference vector.
5) Calculate the summed distance or distortion for the current iteration:

$$D = \sum_{j=1}^{T} \min_{i}\left(d\left(\underline{x}_j, \underline{y}_i\right)\right) \qquad (6.5)$$

where the training vectors are defined by $\underline{x}_j$ for j=1, 2, ..., T and the reference vector used for cell i is defined by $\underline{y}_i$.

6) Calculate the centroid of each cell.
7) If $\frac{D'}{D} - 1 \le \varepsilon$, where D' is the previous value of D, then the iteration is finished and a vector quantiser has been produced with the centroids as the required codebook entries. Otherwise replace each reference vector by the corresponding centroid and go back to step (4).
8) The LBG-CS algorithm finishes when sufficient levels have been obtained, otherwise go back to step (3) to double the number of cells and initiate a new LBG iteration.

6.3 The vector quantisation process

Once a codebook has been designed, any given unquantised input vector of the correct order may be quantised by comparing it with each of the codebook vectors to find the closest match, i.e. to find the codebook vector, which is closest according to the specified distance or distortion measure. This measure should be the same as was used in the codebook training procedure. Then, in place of the vector itself, the index of the selected codebook vector is encoded. If the vector quantiser has V levels, the index can be encoded using $\gamma = \log_2 V$ bits. This index is used at the decoder to summon the appropriate vector from an identical codebook. This vector will then be the quantised version of the original input vector.

One problem is the computational complexity of the process required to quantise each input vector. The required index is the result of searching through the complete codebook, calculating for each entry the distance or distortion from the input vector to a codebook vector to find the smallest value. For a V-level quantiser, to quantise a K-dimension input vector, V distortion calculations are needed each requiring a total of K operations involving a certain number of multiplications and additions [MRG, 85]. Therefore, the computational cost of quantising each input vector is proportional to $K \cdot V$, i.e. $K \cdot 2^{\gamma}$. Thus the computational cost is linearly dependent on the vector dimension and exponentially grows with the number of bits available to represent the index. Since the number of vector elements for a V-level and K-dimensional quantiser is $K \cdot V$, the storage cost is also linearly dependent on the vector dimension and exponentially dependent on the number of bits.

Considerable computational and storage costs are also associated with the design of the codebook. Assuming that a K-dimensional, V-level quantiser is trained with T training vectors, and I LBG iterations per level doubling stage, the computational cost for training a V-level codebook is proportional to $K \cdot V \cdot T \cdot I$, i.e. $K \cdot 2^{\gamma} \cdot T \cdot I$ [MRG, 85]. The storage required is proportional to $K \cdot V + K \cdot T$. For reliable design of the codebook, the number of training vectors should be at least 10 times and ideally 50 times the number of codebook vectors, i.e. T > 10V. So, it is often necessary to find ways of reducing the computational complexity and storage cost.

There are many ways of doing this. One simple way is to partition each of the training vectors into two or more smaller dimensional sub-vectors. Instead of training a single large dimensional codebook, several smaller dimensional codebooks, each for a particular sub-vector range of the training vectors, are trained separately. The number of bits available are divided among the smaller codebooks thus making the number of levels available to each of them considerably less than the number of levels that would be possible with a single codebook.

6.4 Assessment of the performance of the vector quantiser

The performance of a vector-quantiser may be objectively assessed and quantified by calculating the accumulated distortion caused by replacing each input vector taken from a typical set of input vectors by its vector-quantised version. The distortion measure could be the same as that used to train the vector quantiser, but not necessarily so. For speech, when the objective is to represent the short-term spectral envelope by a vector of parameters (e.g. LSF coefficients) a commonly used distortion measure [PA, 93] is the spectral distortion (SD) measure defined as follows:-

$$SD_i^2 = \frac{1}{K}\sum_{k=0}^{K-1} \left|20\log_{10}(\underline{u}_{ik}) - 20\log_{10}(\underline{q}_{ik})\right|^2 \tag{6.6}$$

where $\underline{u}_{ik}$ and $\underline{q}_{ik}$ for k = 0, 1, ..., K-1 are the samples of the DFT magnitude spectra obtained from the i^{th} unquantised vector and the i^{th} quantised vector respectively obtained from a set of typical input vectors. K is the order of the DFT. Let I be the number of typical input vectors. The units of SD_i^2 are squared dBs and the value obtained from equation 6.6 is square-rooted to obtain SD_i in dB. It is often useful to calculate a mean spectral distortion:

$$SD = \frac{1}{I}\sum_{i=1}^{I} SD_i \tag{6.7}$$

a minimum and maximum spectral distortion:

$$SD_{min} = \underset{i=1}{\overset{I}{Min}}(SD_i) \quad \text{and} \quad SD_{max} = \underset{i=1}{\overset{I}{Max}}(SD_i) \tag{6.8}$$

and the percentage, SD_{out}, of “outliers” i.e. vectors for which a certain spectral distortion limit is exceeded. In general, the lower these spectral distortion values, the better the quality of the quantised speech that should be obtainable. When re-synthesised speech obtained for unquantised vectors of short-term spectral parameters and for corresponding quantised vectors are not distinguishable in subjective tests, it is said that "transparent quantisation" of the vectors has been achieved. Objective criteria, based on the spectral distortion measures defined above, have been for deciding whether a vector quantisation scheme for short-term spectral parameters is likely to be transparent or not [PA, 93]. These objective criteria for transparent quantisation are:-

- The mean spectral distortion, SD, should be about 1 dB.
- The percentage of outliers between 2 to 4 dB should be less than 2 %.
- The percentage of outliers larger than 4 dB should be zero.

These three criteria were proposed for the vector quantisation of 10 LSP coefficients [PA, 93] with 24 bits/frame. They are often used more generally and will be used as a guideline in assessing the quantisation schemes discussed in this chapter.

There are many other distortion measures suitable for quantifying the performance of vector-quantisers for short term spectral parameters. Among these are long term averaged squared Euclidean distance measure, the mean-square error (MSE) measure, a weighted mean-squared error measure, a linear prediction distortion measure and some perceptually motivated *distortion measures [Pal, 90] [MRG, 85].* In this chapter the spectral distortion based criteria will be adopted to objectively assess the quantised parameters of the spectral envelope

6.5 Quantisation of parameters for STC

This section is concerned with the quantisation of the parameters for the STC coder discussed in Chapter 4 and modified in Chapter 5. The parameters to be quantised are the pitch-frequency, the voicing probability and the spectral envelope. First, details of scalar quantisers designed for the pitch-frequency and the voicing probability will be given. Then the procedures for vector-quantising the spectral envelope are presented with methods for evaluating their effectiveness. Before presenting details of the quantisation procedures, the concepts of frequency warping and post-filtering need to be discussed since they both are used to improve the effectiveness of the quantisers for STC.

6.5.1 Frequency warping

Frequency warping is a technique proposed for STC [McQ, 92] to improve quantisation efficiency taking advantage of human perception. Human hearing is less sensitive to spectral detail in sound at higher frequencies than at lower frequencies. The DFT samples a spectrum on a linear frequency scale and gives equal importance to all frequency ranges. This even distribution of samples is required to be changed to a non-linear distribution with closer frequency spacing at low frequencies than at high frequencies. A convenient way of achieving this effect is to apply a frequency warping transformation to the spectrum, keeping a linear spacing for the samples of the transformed spectrum.

Given a spectrum $S(e^{j\omega})$, the transformation defines a warped spectrum $S'(e^{j\omega})$ which, for any ω in the range $-\pi$ to π, is equal to $S(e^{j\omega'})$ where ω' is said to be the "warped frequency". The relationship between ω and ω' proposed by McAulay and Quatieri [McQ, 92] is:

$$\omega' = \begin{cases} \omega & 0 \le \omega \le \omega_L \\ \omega_L(1+\alpha)^{\omega-\omega_L} & \omega_L \le \omega \end{cases} \qquad (6.9)$$

where ω_L and α are constants.

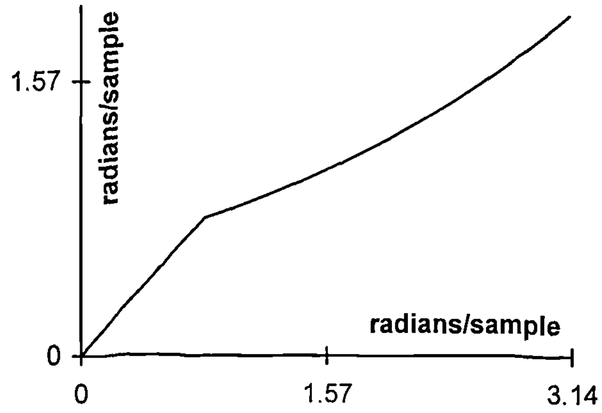


Fig. 6.1 Frequency warping function for STC

This relationship is shown in figure 6.1 (with $\alpha = 0.475$, $\omega_L = \pi/4$), and appears to be discontinuous in slope. We have preferred to use an alternative relationship based on the "Mu-law" companding formula. This formula aims to be approximately linear for low values of ω and becomes more compressed as ω increases. The relationship between ω and ω' is therefore as follows:

$$\omega' = \pi \frac{\log_e(1+\alpha\omega)}{\log_e(1+\alpha\pi)} \tag{6.10}$$

where α is a constant which determines the degree of warping. The larger α, the more severe the warping. Figures 6.2 illustrates the relationship between ω' and ω when $\alpha = 0.9$. In this case, frequencies in the range 0 to $\pi/8$ (i.e. 0 Hz to 500 Hz) map approximately linearly to warped frequencies in the range 0 to 0.71 radians/sample (i.e. 0 Hz to about 904 Hz). This approximately doubles the effective frequency resolution when the warped frequency scale has the same number of samples as the original. The warping compresses the $7\pi/8$ to π range (i.e. 3.5 to 4 kHz) to the warped frequency range 2.9 radians/sample to π (i.e. 3.7 to 4 kHz), thus approximately halving the frequency resolution. This value of α was adopted in our experiments.

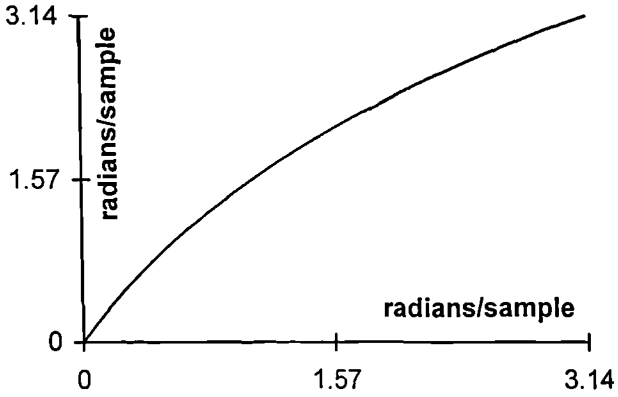


Fig. 6.2 Warping function (warped frequency against frequency)

The inverse warping formula corresponding to equation 6.10 is given in equation 6.11, and illustrated graphically in figure 6.3 for the case where α = 0.9.

$$\omega = \frac{1}{\alpha}\left((1+\alpha\pi)^{\omega'/\pi} - 1\right) \tag{6.11}$$

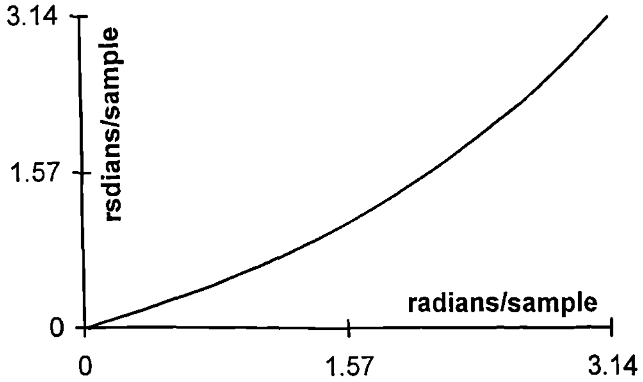


Fig. 6.3 Inverse warping function (frequency against warped frequency)

In practice with STC, frequency warping is applied to the 512 point FFT spectrum by converting each index to a frequency and applying equation 6.10 to calculate a warped frequency. Since the warped frequency will not correspond exactly to an FFT sampling point, linear interpolation is applied between the values of original spectral envelope placed at the warped frequencies to determine values of a new spectral envelope at the exact FFT frequencies. The effect of frequency warping on a typical voiced spectral envelope is shown in figure 6.4. The spreading out at lower frequencies and compression at higher frequencies may be clearly seen.

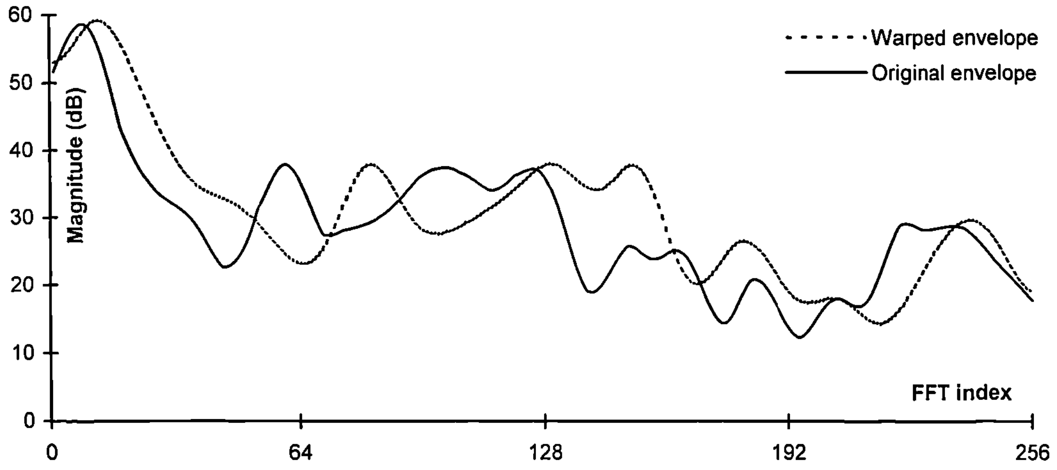


Fig. 6.4 Original and warped spectral envelopes

At the encoder, each FFT spectral envelope is warped in this way before being converted to a cepstrum and vector-quantised. At the decoder, the inverse warping procedure is applied to the received spectral envelope before the sinusoidal model parameters are derived from it.

6.5.2 Post filtering

When a spectral envelope is quantised at the encoder the distortion that occurs will affect the speech quality at the decoder in various ways. For example, distortion of the spectral shape around troughs between formants will cause muffling in the synthetic speech [McQ, 92].

Post-filtering is a technique commonly used in low bit-rate speech coders to improve the perceived quality of synthesised voiced speech by attenuating the spectral envelope troughs. According to studies of human hearing [SAH, 79], a particular sound will be less audible when a louder sound at similar frequencies is heard simultaneously. This is called the "frequency masking effect" [WMCS, 96]. The level of the particular sound which is just masked by the louder sound is called "masking threshold". Ideally, if the quantisation noise can be reduced below the masking threshold, the noise will not be heard in the synthetic speech. For low bit-rate speech coders, there may be two stages to the process of reducing the perceived noise level. The first stage is noise spectral shaping applied at the encoder. This aims to reduce the quantisation noise spectral level at the formants at the expense of raising the quantisation noise level in the spectral envelope troughs. At the decoder, a digital filter, referred to as a "post-filter" may be applied to modify the quantised spectral envelope to reduce both speech and noise levels in spectral envelope troughs between the formants without affecting the formants. It has been reported [GG, 86] that the spectral envelope of voiced speech can often be significantly attenuated between formants without introducing perceptible distortion. The "just noticeable difference" (JND), i.e. the attenuation that may be applied in a spectral trough before the effect is noticed can be as large as 10 dB. Therefore the effect of the post-filter on the quantisation noise will be beneficial and its effect on the speech can hopefully be made not very noticeable

Post-filtering has been traditionally applied to LPC based speech coders, especially CELP. The principle has also been applied to sinusoidal based coders, e.g. STC [McQ, 92] , and it is feasible to implement a post-filter in the frequency domain to reshape a short term spectral envelope as represented by quantised coefficients. Assume that S(ω) is a given voiced spectral envelope represented by a subset of cepstral coefficients distorted by the effects of vector quantisation. Let T(ω) be a measure of the "spectral tilt" defined such that:

$$\log_e T(\omega) = c_0 + 2c_1\cos(\omega) \qquad (6.12)$$

where c_0 and c_1 are first two cepstral coefficients. Also let R(ω) be a "flattened residual envelope" [McQ, 92] defined such that:

$$\log_e R(\omega) = \log_e S(\omega) - \log_e T(\omega) \qquad (6.13)$$

The frequency response of a post-filter, P(ω), defined by McAulay & Quatieri [McQ, 92] is:

$$P(\omega) = \left[\frac{R(\omega)}{R_{max}}\right]^{0.2} \qquad (6.14)$$

where R_{max} is the maximum value of R(ω). The post-filtered envelope can be expressed as following equation:

$$\hat{S}(\omega)=P(\omega)S(\omega) \tag{6.15}$$

From equations 6.14 and 6.15, we can see that at formant peaks, R(ω) may be expected to be close to R_{max}, thus making P(ω) = 1. In principle, the spectral envelope should not be greatly affected at the formants. At spectral envelope troughs, P(ω) should be small, therefore the levels of the post-filtered spectral envelope at such regions should be decreased thus also attenuating the effect of quantisation noise.

Figure 6.5 illustrates the effect of post-filtering on a received voiced spectral envelope. The received envelope is shown before and after post-filtering. The post-filter spectrum defined in equation 6.14 is also shown. It may be seen that the post-filtered envelope around the troughs is attenuated as expected. Unfortunately, the peaks, apart from the first, are also affected due to a general spectral tilt in the post-filter spectrum. The reason for this is that R(ω) is not sufficiently flat and better ways of flattening the envelope of S(ω) can probably be found. Nevertheless the attenuation within the troughs is generally greater than at the peaks. It has been reported [CG, 95] that many post-filters do not exactly maintain the amplitudes of the formant peaks.

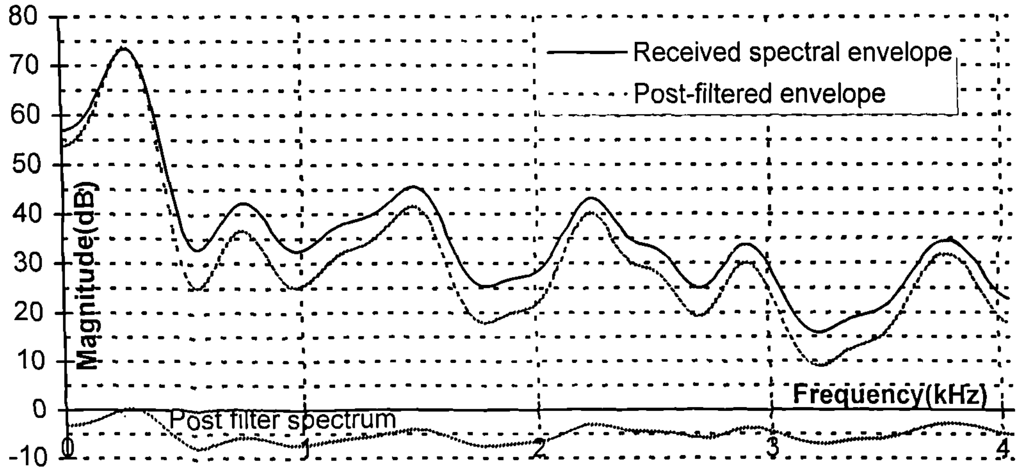


Fig. 6.5 Effect of post filter on the spectral envelope

For STC, since the phase spectrum is derived from the spectral envelope, the post-filtering technique should not be applied to the received envelope until the phase spectrum has been obtained. Experiments with the quantised STC model indicated that a marginal improvement in perceived quality is obtained through the use of a post-filter.

6.5.3 Quantisation of pitch-frequency and voicing probability

Scalar-quantisation is used for the pitch-frequency $\hat{\omega}_0$ (using 8 bits) and the voicing probability V_p (using 4 bits).

The pitch-frequency $\hat{\omega}_0$ is integerised by the following equation which is used in the In marsat-M IMBE standard [MSDI, 91]:-

$$\hat{b}_0 = \left\lfloor \frac{4\pi}{\hat{\omega}_0} - 39 \right\rfloor \qquad (6.16)$$

where $\lfloor\ \rfloor$ denotes "integer part of". Assuming that $\hat{\omega}_0$ is a frequency between $\frac{4\pi}{295}$ and $\frac{\pi}{10}$ radians/sample (i.e. between about 54 Hz and 400 Hz), it follows that $\hat{b}_0$ is a integer between 0 and 255, which can be represented by 8 bits in binary form. At the decoder, the received pitch-frequency $\tilde{\omega}_0$, i.e. the decoded version of $\hat{\omega}_0$, can be deduced from $\tilde{b}_0$, which is the decoded version of $\hat{b}_0$, using the following equation:

$$\tilde{\omega}_0 = \frac{4\pi}{\tilde{b}_0 + 39.5} \qquad (6.17)$$

This scheme quantises the pitch-period in the range 20 to 147 to an accuracy of ±0.25 samples which means that the pitch-frequency is quantised to an accuracy ranging from about ± 0.00007 radians/sample (±0.1 Hz) around 54 Hz to ±0.004 radians/sample (± 5 Hz) around 400 Hz. The error in the highest frequency harmonics within the band 0 to 4 kHz ranges from about ±7 Hz for the 70th harmonic when the pitch-frequency is 54 Hz, to about ±50 Hz for the 10th harmonic when the pitch-frequency is 400 Hz .

Voicing prob (Vp)	Index
0.425	0 (0000)
0.463	1 (0001)
0.501	2 (0010)
0.54	3 (0011)
0.578	4 (0100)
0.616	5 (0101)
0.655	6 (0110)
0.693	7 (0111)
0.731	8 (1000)
0.77	9 (1001)
0.808	10 (1010)
0.846	11 (1011)
0.885	12 (1100)
0.923	13 (1101)
0.962	14 (1110)
1.0	15 (1111)

Table 6.1 Codebook for voice probability

The voicing probability V_p is converted to a 4-bit index by a simple uniform scalar quantiser. The voiced/unvoiced cut-off frequency ω_c as defined in Chapter 4 is equal to $V_p\pi$ radians/sample and the V_p range is from 0.425 to 1 since the ω_c is constrained between $3\pi/8$ and π. Therefore, the voicing probability can be uniformly encoded according to Table 6.1. The voiced/unvoiced cut-off frequency ω_c is quantised to an accuracy of $\pm$ 0.038 π radians/sample i.e. 152 Hz.

Results have shown that when speech is synthesised from the quantised pitch-frequency and voicing probability, with unquantised spectral envelope coefficients, it is indistinguishable from the speech synthesised from totally unquantised parameters. Therefore the scalar quantisers appear to be effective.

6.5.4 Vector-quantisation of the spectral envelope

In published versions of STC [McQ, 92] [McPQS, 91] scalar quantisation is proposed for differentially quantising the short-term spectral envelope coefficients. In the scheme being proposed in this thesis, coefficients representing the spectral envelope at each update-point are required to be encoded by vector-quantisation. To accommodate the relatively large number of coefficients, i.e. 30, a number of different code-books must be used, each trained by the LBG algorithm with centroid splitting (LBG-CS) [LBG, 80].

The short-term spectral envelope obtained at each update-point is represented by a vector of 30 cepstral coefficients. It was claimed [McQ, 92] that cepstral coefficients have a large dynamic range and an uncorrelated nature. Therefore, it was proposed [McQ, 92] *that instead of quantising the cepstral coefficients directly*, they should first be converted to discrete cosine transform (DCT) coefficients and then quantised. The DCT coefficients are very closely related to the spectral envelope and in fact are close to what would be obtained simply by down-sampling the envelope to obtain 30 uniformly spaced out samples in the frequency range 0 to π radians/sample. In a recent paper [SMcDQ, 96], a set of DCT coefficients is actually derived directly by down-sampling the spectral envelope. However, the DCT coefficients derived from the truncated cepstrum are effectively samples of a smoothed natural log spectrum which makes the down-sampling process more reliable and predictable (without aliasing effects). Also [McQ, 93], accurate interpolated values of the spectral envelope at non-integer FFT frequency sampling points are obtained without difficulty. The transformation between cepstral and DCT coefficients is as follows:-

$$d_k = c_0 + 2\sum_{m=1}^{K-1} c_m \cos\left(2\pi\frac{mk}{2K-1}\right) \qquad (6.18)$$

$$c_m = \frac{1}{2K-1}\left[d_0 + 2\sum_{k=1}^{K-1} d_k \cos\left(2\pi\frac{km}{2K-1}\right)\right] \qquad (6.19)$$

where d_k is k^{th} DCT coefficient , c_m is m^{th} cepstral coefficient and K is the number of coefficients.

Experiments were carried out to find an efficient way of vector-quantising 30 DCT coefficients. To achieve reasonable computational complexity in the training and quantisation procedures, the 30 DCT coefficients must be partitioned into a suitable number of sub-vectors using multi-codebooks. In general, the use of multi-codebooks will cause distortion due to the boundary conditions between codebooks. This means that the optimal sub-vector chosen individually from each codebook may not actually be the best vector to use when all sub-vectors are combined together. However, since the DCT coefficients are directly related to the spectral envelope, each sub-vector represents a particular frequency range and sub-vector search is therefore localised in the frequency-domain. In this case, it is likely that the optimal sub-vectors will produce *a satisfactory over-all vector.*

6.5.5 Vector quantisation for DCT coefficients

To train the vector-quantisers for the spectral envelope coefficients, about 18,400 training vectors each containing 30 DCT coefficients were produced by analysing 368 seconds of natural speech. The speech was taken from a specially selected file of natural speech spoken by a diversity of speakers [Gsp.pc]. Vectors from silent speech frames were eliminated from the training set. The 30 DCT coefficients per frame were produced by the normal STC process; i.e. FFT analysing 2.5 pitch-periods with zero-padding, peak-picking, cubic spline smoothing, cepstral analysis, truncation and DCT transforming.

The LBG-CS training procedure was applied individually to sub-vectors of the vector of 30 DCT coefficients, with the squared Euclidean distance measure (equation 6.2) used for each sub-vector. This distance measure corresponds closely to the spectral distortion measure defined in Section 6.4, the minor difference being the reduction of the number of DFT frequency samples to 30. The training procedure was implemented as described above with two minor modifications [BT, 94].

Firstly, rather than always splitting each centroid into two as described in Section 6.2.2, the splitting was made dependent on the contents of the cells. Only cells for which the number of vectors was larger than the average number had their centroids split into two reference vectors. The centroids of other cells were not split. When the number of reference vectors became equal to the required number of code-book entries the splitting procedure was terminated. For an "N level" quantiser, if there are M training vectors, the average number per cell is M/N.

Secondly, during the LBG training procedure for any of the intermediate stages, if a cell was found to be empty after the re-clustering step (step 4 in Section 6.2.2), i.e. if no vectors were in the cell, the program eliminated this cell and deleted its reference vector as a potential code-book entry. Hence the number of reference vectors could reduce during the LBG training. Any reductions were eventually made up at a centroid splitting stage.

At bit-rates of between 2.4 and about 4 kb/s and with 20 ms update-points, between 36 and about 68 bits are available per frame for quantising the 30 DCT coefficients, allowing about 12 bits per frame for the pitch-frequency and voicing probability. Preliminary experiments applying vector-quantisation directly to the 30 zero-mean DCT coefficients, with the mean value of the 30 DCT coefficients scalar quantised and the 30 coefficients divided into 3 sub-vectors proved unsuccessful at 2.4 kb/s. Various vector sub-divisions and allocations of the 36 available bits were tried, but in all cases the spectral distortion measurements and percentage of outliers was very large. According to the objective criteria discussed in Section 6.3, and using a natural speech file with about 1000 frames, typically average spectral distortion measures of greater than 3.5 dB were obtained with around 30% of 4 dB outliers.

Rather than attempting to quantise DCT coefficients directly, McAulay and Quatieri [McQ, 92] proposed a scheme for quantising the differences between adjacent samples of the DCT magnitude spectrum in increasing order of frequency. Their scheme is based on the idea of DPCM but is applied in the frequency-domain. This is advantageous because the correlation between adjacent DCT coefficients is high for voiced speech. As with DPCM the difference encoded is the difference between the actual coefficient and the adjacent quantised coefficient rather than the straightforward difference between two unquantised coefficients. We refer to the former as a "backward" difference and the latter as a "forward" difference. This is an "intra-frame" differential quantisation scheme rather than an inter-frame scheme. An inter-frame scheme, i.e. a scheme involving coefficients from the previous frame, would clearly have possible benefits also.

The differential quantisation scheme proposed by McAulay and Quatieri uses scalar quantisation for the differences. In this thesis, we investigated the idea of using vector-quantisation for sub-vectors of differences. Since we anticipate a smaller dynamic range of vector norms than was obtained with direct DCT quantisation, greater vector-quantisation efficiency [GWZ, 92] was hoped for.

Despite the fact that backward differences were to be used for the vector-quantisation process, forward differences had to be used to derive the training vectors from the speech data-base. This was necessary to simplify the training procedure allowing a standard LBG-CS algorithm to be employed. Training vectors were defined to contain, for each frame, the first DCT coefficient, the difference between the first DCT coefficient and the second, the difference between the second and the third, and so on.

Figure 6.6 shows the mean square value minus the squared mean (referred to as the variance) of each vector element over 18400 frames for both vectors of zero-mean DCT coefficients and vectors of differences of DCT coefficients. The variance is defined as:

$$\mathrm{Var}_k = \frac{N\sum_{i=1}^{N} D_{ik}^2 - \left(\sum_{i=1}^{N} D_{ik}\right)^2}{N^2} \qquad \text{for k=0, 2, ..., K-1} \qquad (6.20)$$

where D_{ik} is the kth coefficient at the ith frame, K is the number of coefficients and N is the number of frames. It may be seen that the variances for the differences are consistently lower than for the corresponding DCT coefficients.

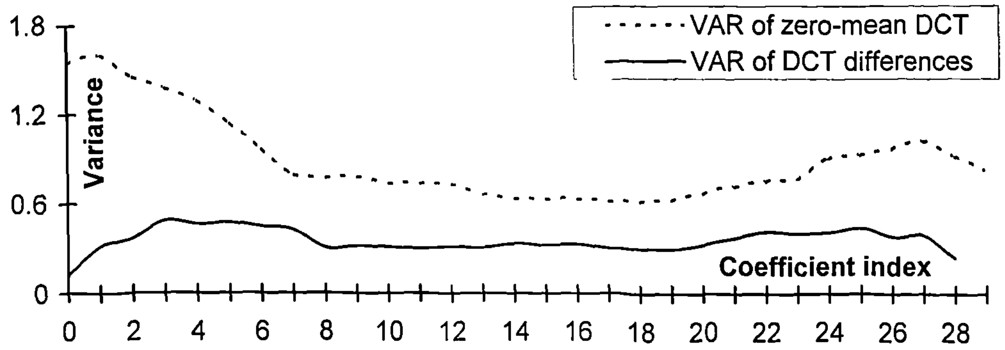


Fig. 6.6 Variance of coefficients for zero-mean DCT and difference of DCT

6.5.6 Training of vector quantisers

Firstly, the design of a 30 coefficient differential vector quantiser for a 2.4 kb/s STC coder was considered. Thirty-six bits per frame are available for the spectral envelope. Several sub-vector partitions and bit-allocations were tried. The best result achieved was using five code-books with the following vector partition and bit allocation:-

Vector element	Bit allocation
• 0	5 bits
• 1-6	8 bits
• 7-12	8 bits
• 13-20	7 bits
• 21-29	8 bits

These code-books were trained on a UNIX main-frame. All five took a total of about 20 hours of training time. The first code-book required 9 centroid splitting stages with a total of 320 LBG iterations; The second code-book required 15 centroid splitting stages and 333 LBG iterations. The third, fourth and fifth code-books required 15, 13 and 18 centroid splitting stages and totals of 225, 165 and 207 LBG iterations respectively.

Multiple vector-codebooks were also trained for three higher bit-rate versions of STC. The bit-rates were 3.4, 4 and 4.4 kb/s, allowing 56, 68 and 76 bits respectively for coding the DCT coefficient differences at each 20 ms update-point and a provision of 600 b/s for the pitch-frequency and the voicing probability. The vector partitions and bit-allocations for each sub-vector are given for the three coders in Table 6.2.

	56-bit quantiser		68-bit quantiser		76-bit quantiser	
Sub-vector	partition	bit alloc.	partition	bit alloc.	partition	bit alloc.
1	0	6 bits	0	6 bits	0	6 bits
2	1-6	10 bits	1-4	9 bits	1-4	10 bits
3	7-12	10 bits	5-8	9 bits	5-8	10 bits
4	13-18	10 bits	9-12	9 bits	9-12	10 bits
5	19-24	10 bits	13-16	8 bits	13-16	10 bits
6	25-29	10 bits	17-21	9 bits	17-21	10 bits
7	--	--	22-25	9 bits	22-25	10 bits
8	--	--	26-29	9 bits	26-29	10 bits

Table 6.2 Vector partitions and bit-allocations

After the code-books had been trained, one thousand test vectors were generated for evaluation purposes from a different file containing 20 seconds of voiced natural speech [Mvoice.pcm]. Each test vector contained 30 unquantised DCT coefficients. A corresponding set of differentially vector-quantised sub-vectors was then created by finding the optimal code-book indices, and determining the corresponding code-book sub-vectors. Backward differences were used during the search procedure, as intended in the final coder. Usually, the distortion measures used for code-book training and ultimate use should be identical The use of "backward" differential quantisation introduces some differences.

6.5.7 Backward differential quantisation

To explain why backward differential quantisation is necessary, let "ud " be a vector containing 30 unquantised DCT differences and let "cd" be a 30 element vector composed of the closest appropriate code-book sub-vectors. If the code-book search used for each sub-vector of ud is programmed to minimise the Euclidean distance between the sub-vector and the chosen code-book vector, it will introduce levels of quantisation error which become increasingly large for the higher frequency sub-vectors due to the accumulation (or integration) of quantisation error. This is because the process of converting back from DCT differences to DCT coefficients is an accumulation or integration process as illustrated by the following equations. Let q_i for i = 0, 1, 2, ..., 29 be the elements of the vector q of received DCT coefficients reconstituted from the quantised differences. Also let cd_i be the elements of cd. Then,

$$q_0 = cd_0$$

$$q_1 = q_0 + cd_1 = cd_0 + cd_1$$

$$q_2 = q_1 + cd_2 = cd_0 + cd_1 + cd_2 \qquad \text{etc.}$$

Summing the elements of cd as illustrated will clearly increase the effect of quantisation error.

The vector-quantisation procedure is therefore modified so that when searching the code-book for an optimal sub-vector of cd, the distance between the quantised sub-vector of DCT coefficients it would produce and the original sub-vector of DCT coefficients is minimised. Note that the distance measure now being used is different from that used in the code-book training procedure. To illustrate the use of backward differences consider the following example:-

Assume that instead of 30 DCT coefficients, there were only eight, partitioned into 3 sub-vectors with 1, 3 and 4 elements. Let the unquantised DCT coefficients be u_i for $i = 0, 1, 2, .., 7$. The first, u_0, is scalar-quantised to obtain cd_0 from the first code-book. Let $q_0 = cd_0$. Then to find the optimal vector $[cd_1, cd_2, cd_3]$ the second code-book is searched to find the sub-vector which minimises the distance between $[u_1, u_2, u_3]$ and a vector $[q_1, q_2, q_3]$ where:

$$q_1 = q_0 + cd_1$$

$$q_2 = q_1 + cd_2 \quad (= q_0 + cd_1 + cd_2)$$

$$q_3 = q_2 + cd_3 \quad (= q_0 + cd_1 + cd_2 + cd_3)$$

An element of recursion has now been introduced into the search procedure, but this does not really make it any more complicated. The recursion involves calculation of the received DCT sub-vector $[q_1, q_2, q_3\ ]$ also at the encoder and is in some ways an analysis by synthesis procedure. Note that $[q_1, q_2, q_3\ ]$ is dependent on q_0 and the chosen sub-vector $[cd_1, cd_2, cd_3]$ will attempt to compensate for quantisation error in q_0. To find the optimal vector $[cd_4, cd_5, cd_6, cd_7]$, the third code-book is searched to find the entry which minimises the distance between $[u_4, u_5, u_6, u_7]$ and a vector $[q_4, q_5, q_6, q_7]$ where:

$$q_4 = q_3 + cd_4 \quad \text{etc.}$$

The chosen entry will be dependent on q_3 as obtained from the previous sub-codebook and will attempt to compensate for quantisation error in q_3.

Better results are obtained by minimising backward rather than forward differences as illustrated in figure 6.7 (a), (b) and (c). This figure shows a comparison between unquantised and quantised DCT coefficients produced by the forward difference and the backward difference methods. The DCT coefficients are for a single analysis frame of voiced speech. It may be seen that the quantised DCT coefficient curve in figure 6.7(a) drifts away from the unquantised one as the frequency increases while the quantised DCT coefficient curves in figure 6.7(b) and (c), are much closer to the same unquantised DCT coefficient curve. The 36-bit quantiser discussed earlier was used to produce figure 6.7(a) and (b). The 76-bit quantiser was used to produce figure 6.7(c).

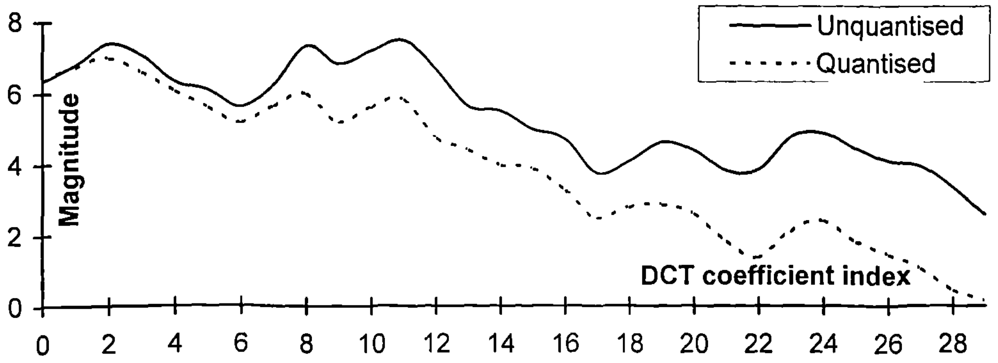


Fig. 6.7 (a) Received DCT coefficients from quantised forward differences

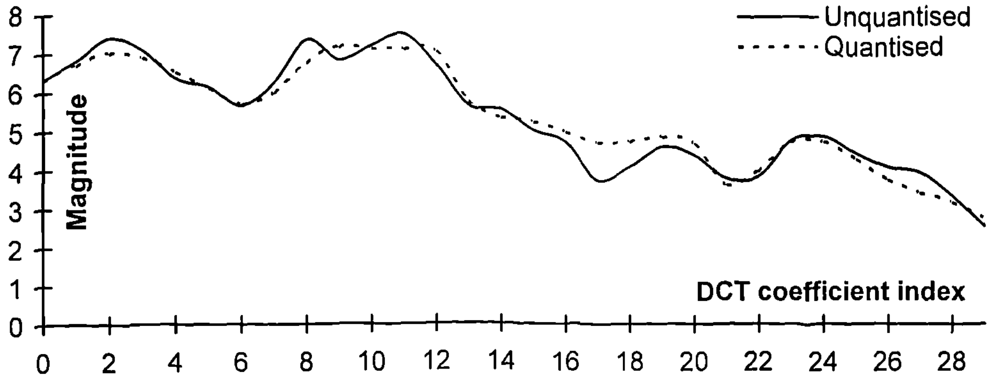


Fig. 6.7 (b) Received DCT coefficients
from quantised backward differences for 36 bits

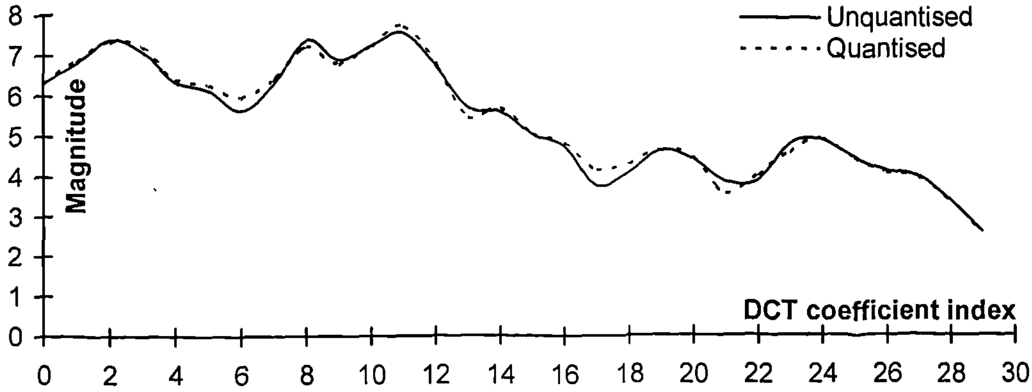


Fig. 6.7 (c) Received DCT coefficients
from quantised backward differences for 76 bits

6.5.8 Evaluation of the vector quantisers

The performance of the multiple code-book quantisers were evaluated by calculating the spectral distortion produced by quantised DCT vectors $\underline{q}_i$ with reference to the corresponding unquantised DCT vectors $\underline{u}_i$ for the 1000 frame test file [Mvoice.pcm] referred to above. The spectral distortion SD_i between $\underline{u}_i$ and $\underline{q}_i$ for each frame i was calculated, as in Section 6.4, as the square root of:

$$SD_i^2 = \frac{1}{K}\sum_{k=0}^{K-1} \left|u_{ki} - q_{ki}\right|^2 \tag{6.21}$$

where u_{ki} and q_{ki} are the k^{th} coefficients of the unquantised and the quantised DCT vectors respectively for the i^{th} frame. K is the number of DCT coefficients, i.e. 30. The mean spectral distortion over non-silent frames selected from the file is, as in Section 6.4:

$$SD = \frac{1}{I'}\sum_{i=1}^{I'} SD_i \tag{6.22}$$

where I' is the number of frames considered not to be silent, these being omitted from consideration.

The results obtained from this objective evaluation of the 36-bit quantiser were as follows:-

Average spectral distortion:	3.0 dB
2-4 dB outliers:	82.8%
>4 dB outliers:	9.6%
maximum distortion:	8.9 dB
minimum distortion:	1.2 dB

It may be seen that the mean spectral distortion and the percentage of outliers are still much worse than the guidelines [PA, 93] mentioned in Section 6.3. When speech was synthesised from the data file containing quantised DCT coefficients (with unquantised pitch-frequency and voicing probability) and compared with speech produced by unquantised parameters it was found that additional perceptible distortion, such as reverberance, had been introduced by the vector quantisation.

The higher bit-rate quantisers mentioned in Section 6.5.6 were evaluated to attempt to obtain better quality, with the hope of approaching transparency. STC coders with bit-rates 3.4 kb/s, 4 kb/s and 4.4 kb/s were thus developed with the vector-quantisation schemes given in table 6.2. The results obtained from the objective evaluation of the higher bit-rate quantisation schemes were as follows:-

	56-bit quantiser	68-bit quantiser	76-bit quantiser
average SD (dB)	1.83	1.55	1.34
2-4 dB outliers (%)	30.9	14.3	7.1
>4 dB outliers (%)	0.2	0.0	0.0
max distortion(dB)	4.69	3.67	3.37
min distortion (dB)	0.59	0.61	0.45

Table 6.3 Spectral distortion and outliers

As expected, the mean spectral distortion and the percentage of outliers significantly improved as the bit-allocation was increased. The synthetic speech quality for the 56-bit quantiser (3.4 kb/s) was more natural and less distorted than for the 36-bit quantiser, but over headphones, differences between the unquantised and quantised synthetic speech could still be heard. The synthetic speech for the 68-bit quantiser was even better, but, as the objective measurements predicted, the quantisation did not appear to be quite transparent over good quality headphones.

As recorded in table 6.3, the 76-bit quantiser had a mean spectral distortion close to 1 dB, with no 4-dB outliers and 7% of 2-4 dB outliers. These measurements still do not quite satisfy the objective criteria for transparency mentioned in Section 6.3 (apart from the "4-dB outliers" criterion which is satisfied). However, they are quite close, and informal listening tests revealed that the synthetic speech derived from the

76-bit quantiser was virtually indistinguishable from the speech obtained from the unquantised spectral envelope, even over good quality headphones.

It may be remarked that the objective criteria used here as a guideline were proposed by Paliwal [PA, 93] for the vector quantisation of LPC parameters. The 30 DCT coefficients allow considerably more flexibility in the attainable spectral envelope shapes and it is possible that Paliwal's criteria are slightly pessimistic as predictions of perceived speech quality.

The objective measurements and subjective evaluations discussed above indicate that vector-quantisation of backward differences of DCT coefficients can probably be made effective at bit-rates of 4 kb/s and above. The objective measurements are close to what is required, and there are many possibilities for further improving the quantisation schemes; for example by including a degree of inter-frame as well as intra-frame prediction [EC, 94]. Also, using a larger training set may produce better results, since the training set used in this work was only about 18 times larger than the code-book size. Ideally, the training set size should be 10 to 50 times larger than code-book size.

Objective measurements obtained at lower bit-rates, e.g. at 2.4 kb/s, are less promising in this respect. The spectral envelope quantisation is so far from being transparent that it cannot be predicted that transparency is achievable even with sophisticated enhancements. This finding suggests that at 2.4 kb/s, all-pole modelling techniques may be the only option at present despite the advantages of cepstral-based envelope modelling. Several ways of modelling spectral envelopes by all-pole modelling techniques are discussed in next section.

6.5.9 All-pole modelling for deriving STC spectral envelope

In 1993 [McCQ, 93] and again in 1994 [McQC, 94], McAulay, Champion and Quatieri suggested the use of high order LPC coefficients for representing spectral envelopes in very low bit-rate STC coders. This suggestion was made despite the fact that the spectral envelopes are spectrally warped and therefore no longer conform, in theory, to an all-pole model. Preliminary results [McQC, 94] suggested that scalar-quantisation of 22 LSF (line spectral frequency) coefficients representing LPC spectra approximating the required spectral envelopes produced improved quality over the cepstral based approach in 4.8 kb/s and 8 kb/s STC coders. At 2.4 kb/s an improvement in quality was also reported. [McQC, 94] using 14 LSF coefficients to approximate the frequency warped spectral envelopes.

There are several ways of deriving a high order LP analysis filter whose all-pole frequency response approximates a given spectral envelope. Applying auto-regressive analysis of high order (i.e. higher than about 12) directly to time-domain speech will generally not be successful since the order of the analysis filter will often be close to the smallest pitch-period (e.g. 20), and a degree of long term as well as short term prediction will be attempted. However, converting the required spectral envelope to a power spectrum (simply by squaring the linear scaled frequency domain samples), and then performing an inverse DFT produces an auto-correlation sequence whose samples can be supplied to a "Levinson-Durbin" LP analysis algorithm which can then compute the LP coefficients in the normal way. This is a straightforward application of the "Weiner-Khintchine" theorem which states that the Fourier transform of the auto-correlation function is the power spectrum.

The We.ner-Khintchine calculation referred to above is easily carried out, but it has some problems in practice. Firstly, if frequency-domain smoothing, for example by cubic spline interpolation, is applied to derive the spectral envelope from which the auto-correlation samples are derived, this will not correctly reveal any formant peaks that lie between harmonics. Therefore the spectral envelopes will not necessarily resemble those of all-pole spectra, even if the effect of frequency-warping is disregarded in this respect. If frequency-domain smoothing is not applied, the higher order auto-correlation coefficients used for the LPC analysis will start to be affected by the long term as well as the short term periodicity of higher pitched and especially female voiced speech. The frequency-warping will exaggerate this effect.

Secondly, although the LP analysis will produce all-pole spectra approximating the required smoothed envelopes reasonably well, more weight is given to the accuracy of the approximations around formants rather than in troughs between formants. This is due to the least squares error measure used in the Levinson-Durbin algorithm which, expressed in terms of speech samples $\{s[n]\}_{0,N-1}$, is:

$$E_{lp} = \sum_{n=0}^{N-1} (s[n] \otimes a[n])^2 \tag{6.23}$$

where $\{a[n]\}$ is the impulse response of the linear prediction filter and " $\otimes$ " denotes convolution. The DTFT of $s[n]\otimes a[n]$ is $S(e^{j\omega})A(e^{j\omega})$ with power spectrum $P(\omega)/\hat{P}(\omega)$ where $P(\omega)$ and $\hat{P}(\omega)$ are the power spectra of s[n] and the all-pole LP envelope respectively at frequency ω. It follows from Parseval's theorem applied in the DFT domain that:

$$E_{lp} = \frac{1}{N} \sum_{k=0}^{N-1} \frac{P(\omega_k)}{\hat{P}(\omega_k)} \tag{6.24}$$

where the DFT frequencies are ω_k. Since $\hat{P}(\omega)$ is the power spectrum of an all-pole impulse response it is constrained [MG, 76] to satisfy:

$$\sum_{k=0}^{N-1} 10\log_{10} \hat{P}(\omega_k)\,\mathrm{dB} = 0 \tag{6.25}$$

which means that values of $\hat{P}(\omega)$ above unity must exist with values less than unity such that the overall average of $10 \log_{10}(\hat{P}(\omega))$ on a dB scale becomes zero.

From equation 6.24 it may be seen that across frequency range, differences between $P(\omega)$ and $\hat{P}(\omega)$ will be more highly weighted at higher amplitudes of $P(\omega)$ than at lower amplitudes. Perceptually, this may not be a bad thing, but the nature and accuracy of the spectral match will clearly be different from what is obtained using cepstral or DCT coefficients.

Despite these theoretical reservations, the approach has been tried in its basic form and encouraging results have been obtained. The graph below shows a 14th order all-pole spectrum which was derived as described above from the envelope of the voiced male speech spectrum shown. It can be seen that in this case the envelope peaks appear to be reasonably represented, though the deep troughs at around 2 kHz and 2.8 kHz are, as expected, badly modelled.

The speech quality obtained from the STC model, with 14th order unquantised all-pole LP coefficients representing the spectral envelope, sounded very similar to that obtained with the 30th order cepstral-based envelope. This may be due to the fact that perceptually the accuracy of the envelope is less significant in spectral troughs between formants than at the formants.

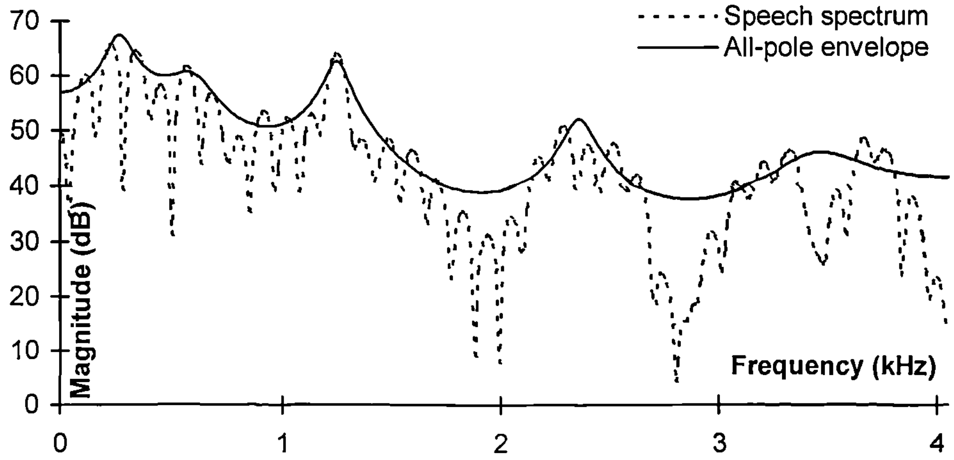


Fig. 6.8 A 14 th all-pole envelope fitting on the speech spectrum

Vector-quantisation of the LPC coefficients has not yet been carried out, though there is evidence that transparent vector-quantisation of 14 LSP coefficients is achievable within the 36 bits available with a 2.4 kb/s STC coder. Transparency has been achieved [PA, 93] for 10 LSP coefficicnts using 24 bits.

There are several alternative methods of deriving an all-pole model of a given spectral envelope sampled at discrete values of frequency. Two such methods are known as discrete all-pole modelling (DAP) [EIM, 91] and iterative all-pole modelling (IAP) [PWC, 95]. Both these methods are applicable in the situation when the spectral envelope is known only at pitch-harmonics, though interpolation may be necessary to derive extra points when the fundamental pitch-frequency is particularly high and a very high order model is required.

The DAP modelling technique uses a form of "all-pole" linear prediction (LP) analysis in the frequency-domain which is fundamentally different from conventional LP analysis as seen either in the time-domain or in the frequency-domain. It has the capability of more accurately representing an assumed all-pole spectral envelope than the conventional method, especially when the pitch-frequency is high and consequently few pitch-frequency harmonics are available to sample the envelope. It is a technique which assumes that a power spectral envelope P(ω) is known at a relatively small number of discrete frequencies $\omega = \Omega_1, \Omega_2, \dots, \Omega_L$ which would normally be the pitch-frequency harmonics. The object is, as usual, to derive an all-pole filter with transfer function 1/A(z), with $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}$ such that the power spectrum, $\hat{P}(\omega)$, of $G/A(e^{j\omega})$ is close to P(ω) according to some distance measure. The value of the "gain factor" G is to be determined also.

Conventional LP analysis attempts to match $\hat{P}(\omega)$ to P(ω) by minimising equation 6.24 which measures the distance between them over a complete set of DFT frequencies; this is like measuring the distance for a continuous range of ω from -π to π. It has been shown that the power spectrum $\hat{P}(\omega)$ thus obtained minimises the continuous Itakura-Saito measure:

$$E_{IS} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\frac{P(\omega)}{\hat{P}(\omega)} - \ln_e\left(\frac{P(\omega)}{\hat{P}(\omega)}\right) - 1\right]d\omega \qquad (6.26)$$

which when expressed in the DFT domain is:

$$E_{Is} = \frac{1}{N}\sum_{k=0}^{N-1}\left[\frac{P(\omega_k)}{\hat{P}(\omega_k)} - \ln_e\left(\frac{P(\omega_k)}{\hat{P}(\omega_k)}\right) - 1\right] \qquad (6.27)$$

The DAP method takes the samples of P(ω) at the discrete frequencies $\omega = \Omega_1, \Omega_{2,} \dots, \Omega_L$ and attempts to find G and coefficients of A(z) such that $\hat{P}(\omega)$ and P(ω) are close at these discrete frequencies. The measure of closeness used is a modified form of the Itakura-Saito measure defined as follows:

$$E_{Is} = \frac{1}{L}\sum_{m=1}^{L}\left[\frac{P(\Omega_m)}{\hat{P}(\Omega_m)} - \ln_e\left(\frac{P(\Omega_m)}{\hat{P}(\Omega_m)}\right) - 1\right] \tag{6.28}$$

In general minimising this modified Itakura-Saito distance measure will give a different result from conventional LP analysis, though the more discrete frequencies there are, the closer will be the two results.

From the discrete samples of P(ω) an auto-correlation function may be calculated by the following formula:

$$R[i] = \frac{1}{L}\sum_{m=1}^{L} P(\Omega_m)e^{j\Omega_m i} \tag{6.29}$$

This is, in fact, the auto-correlation function of a "line" power spectrum with the power between the discrete frequencies considered zero. The auto-correlation function R[i] will therefore have a shape which repeats after P points, and is symmetric about i = P/2 where P is the pitch-period.

Conventional LP analysis based on this auto-correlation function, truncated before it is repeated, will produce a reasonable all-pole envelope, but the envelope cannot be expected to be as close as possible to P(ω) at the discrete frequencies. This observation applies even if P(ω) corresponds exactly to an all-pole transfer function of the same order as the LPC analysis. Even then, $\hat{P}(\omega)$ will generally be different from P(ω) despite the fact that they are both all-pole and have the same order. Making $\hat{P}(\omega)$ equal to P(ω) for all ω in this case would clearly minimise the modified Itakura-Saito measure (6.28), but not the original one (6.27).

The reason for this discrepancy is that the auto-correlation function defined by equation 6.29 is distorted as a delay-domain representation of the required envelope P(ω). The distortion is caused by aliasing because P(ω) is essentially under-sampled in the frequency-domain. As a result, conventional LP analysis tends to produce envelopes with formant peaks shifted towards one or other of the discrete frequencies. This is a particular disadvantage for STC coding where LPC modelling is to be used as efficient way of representing the envelope at pitch-frequency harmonics.

By giving a closer match between P(ω) and $\hat{P}(\omega)$ at discrete frequencies, the DAP analysis method appears to have a considerable advantage for STC. Further, where the speech can be considered to truly conform to an all-pole model, the increased accuracy in envelope derivation may also lead to greater accuracy in the derived phase spectrum. Glottal excitation modelling as discussed in Chapter 5 can increase the conformity of the speech to an all-pole model and thus in principle make the DAP technique even more applicable.

DAP modelling requires an iterative technique for calculating G and the coefficients of A(z). It is convenient to re-express G/A(z) in the form 1/A(z) where the coefficients of A(z) are "de-normalised", i.e. $a_0 = G$ rather than one, and $a_1, a_2, ..., a_M$ are also scaled accordingly. Given an initial estimate, obtainable by conventional LP analysis, the frequency response of 1/A(z) is down-sampled at the discrete frequencies $\omega = \Omega_1, \Omega_{2,} ..., \Omega_L$. This is the same down-sampling process that is assumed to have happened to the true envelope P(ω). From the down-sampled frequency response, a form of impulse response $\hat{h}[n]$ is calculated by the inverse transform:

$$\hat{h}[n] = \frac{1}{L}\sum_{m=1}^{L} e^{-j\Omega_m n} / A(e^{j\Omega_m}) \qquad \text{for } n = 0, \pm 1, ..., \pm M \qquad (6.30)$$

This is a form of impulse-response which corresponds to the "line" spectrum obtained as a result of the down-sampling. As compared with the true impulse-response of 1/A(z) it will have suffered aliasing distortion in the time-domain. The effects of circular convolution also mean that it cannot be assumed that $\hat{h}[n] = 0$ for $n < 0$. The auto-correlation function R[i], as given by equation 6.29, is considered to be a similarly distorted version of the true auto-correlation function that could be obtained from P(ω) if P(ω) were known for all ω. The DAP iterative process observes that:

$$\sum_{k=0}^{M} a_k \hat{R}[i+k] = \hat{h}[i] \quad \text{for } i = 0, \pm 1, \ldots, \pm M \tag{6.31}$$

where $\hat{R}[i]$ is the auto-correlation function corresponding to $\hat{h}[i]$ and $A(z) = a_0 + a_1 z^{-1} + \ldots + a_M z^{-M}$. Also, by substituting $|1/A(e^{j\omega})|^2$ for $\hat{P}(\omega)$ in the modified Itakura-Saito formula and setting $\partial E_{IS}/\partial a_k = 0$ for k = 0, 1, 2, ..., M, it follows that to minimise this distance measure, we must have:

$$\sum_{k=0}^{M} a_k \left(R(i-k) - \hat{R}(i-k) \right) = 0 \quad \text{for } i = 0, \pm 1, \pm 2, \ldots, \pm M \tag{6.32}$$

These equations are re-expressed by substituting from equation 6.29 to obtain:

$$\sum_{k=0}^{M} a_k R(i+k) = \hat{h}[i] \quad \text{for } i = 0, \pm 1, \pm 2, \ldots, \pm M \tag{6.33}$$

An iterative solution to this set of simultaneous equations is now possible by solving for a second set of a_k coefficients assuming constant $\hat{h}[i]$, re-computing $\hat{h}[i]$ using these a_k coefficients, solving again for a third set of a_k coefficients, and so on until convergence is obtained. Since only M+1 simultaneous equations are required, the DAP reference solves equation (6.33) for i = 0, -1, ..., -M at each step. A rigorous justification for this iterative process and an investigation of its convergence is given in the reference [ElM. 91].

Therefore a solution to the DAP modelling process appears to be possible at the expense of considerably increased computational complexity as compared to conventional LP analysis. Although the technique is clearly worthy of further evaluation for STC, there are many uncertainties about it that must be explored. It is pointed out [ElM, 91] that DAP modelling will not necessarily always provide a better spectral envelope for real speech than other techniques. It is based on an all-pole assumption which will not be appropriate for all segments of natural speech. A pole-zero version of the technique has been proposed [ElM, 89], and this may have further benefits for STC.

The IAP technique [PWC, 95] has the same objective as the DAP modelling technique, but uses a different iterative process. Given the power spectrum of a segment of voiced speech, the process calculates auto-correlation coefficients via an inverse DFT applied to the whole power spectrum; i.e. not just to a number of frequency-domain samples. An initial all-pole envelope is then obtained by applying conventional LP analysis to a truncated set of these auto-correlation coefficients. As discussed above, this all-pole envelope will not necessarily be close to the original power spectral peaks at the pitch-frequency harmonics. Therefore the envelope obtained from the all-pole model is modified, at the harmonic locations only, by replacing the modelled power by the true (original) speech power. A new truncated set of auto-correlation coefficients is then calculated from this "spiky" all-pole power spectrum, a new all-pole envelope is derived by conventional LP analysis, and this iteration procedure is carried on until a convergence criterion is satisfied.

The three different all-pole modelling approaches discussed in this section produce similar results for lower pitch-frequency (male) speech. Significant differences occur for higher pitch-frequency (female) speech. The application of these and other all-pole and possibly pole-zero modelling techniques to very low bit-rate STC clearly has much potential and will be the subject of future research.

6.5.10 Simulation of the STC low bit-rate coders

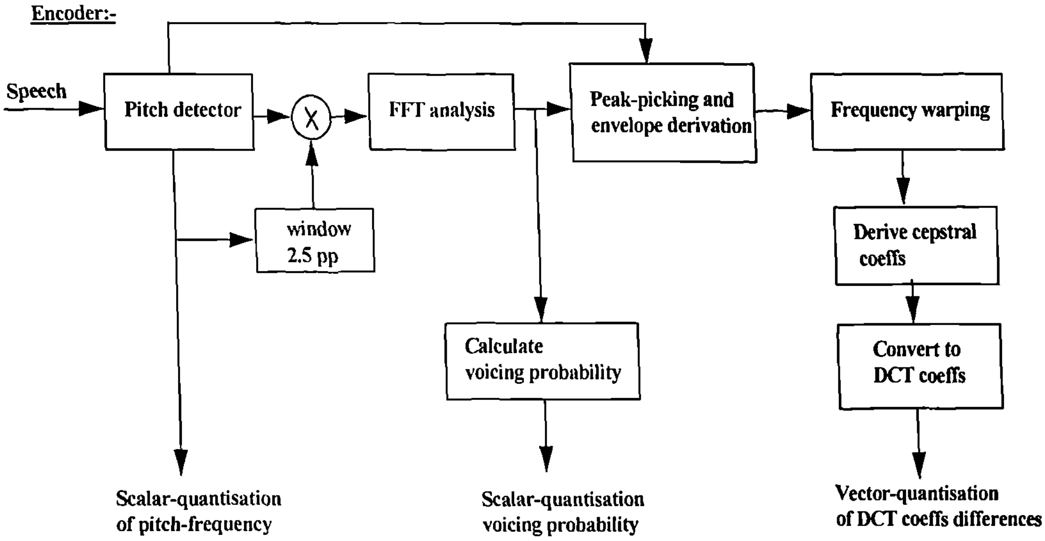
A "Turbo C++" program [SC, 96] has been developed as part of this project to simulate the operation of four fully quantised STC coders. The term "simulation" is applied since, although the software is intended as a prototype for a real time DSP implementation, the problems of real time implementation have not yet been properly addressed. The program takes as its input a computer file of speech samples and produces a corresponding output file. Problems of channel errors have also not yet been considered.

The coders are applied to 8 kHz sampled input speech which is analysed at update-points at 20 ms intervals. The four coders differ in that the parameters are encoded at bit-rates of 2.4 kb/s, 3.4 kb/s, 4 kb/s and 4.4 kb/s which are equivalent to 48, 68, 80 and 88 available bits respectively per update-point. Twelve of the available bits are used to encode pitch-frequency and voicing probability by scalar quantisation as described in Section 6.5.3. The spectral envelope coefficients are vector-quantised using the remaining available bits. The bit allocation schemes for these four STC coders are summarised in table 6.4.

Parameters	2.4 kb/s	3.4 kb/s	4 kb/s	4.4 kb/s
Pitch-freq.	8 bits	8 bits	8 bits	8 bits
Voicing prob.	4 bits	4 bits	4 bits	4 bits
DCT coeffs	36 bits	56 bits	68 bits	76 bits
Total	48 bits	68 bits	80 bits	88 bits

Table 6.4 Bit allocation for the four STC coders

A simplified flow diagram for the fully quantised coders is given in figure 6.8.


Encoder:-
Speech
Pitch detector
X
FFT analysis
Peak-picking and envelope derivation
Frequency warping
window 2.5 pp
Derive cepstral coeffs
Calculate voicing probability
Convert to DCT coeffs
Scalar-quantisation of pitch-frequency
Scalar-quantisation voicing probability
Vector-quantisation of DCT coeffs differences

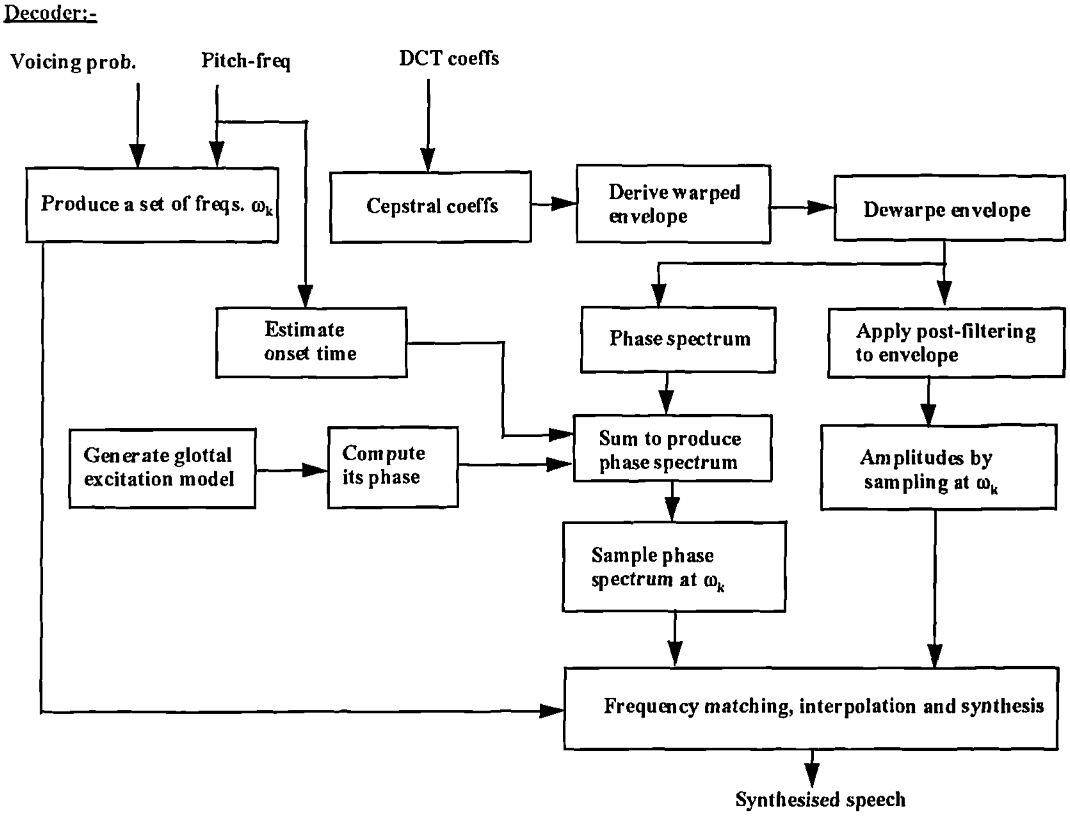

Decoder:-
Voicing prob.
Pitch-freq
DCT coeffs
Produce a set of freqs. ωk
Cepstral coeffs
Derive warped envelope
Dewarpe envelope
Estimate onset time
Phase spectrum
Apply post-filtering to envelope
Generate glottal excitation model
Compute its phase
Sum to produce phase spectrum
Amplitudes by sampling at ωk
Sample phase spectrum at ωk
Frequency matching, interpolation and synthesis
Synthesised speech

Fig. 6.8 Flow diagram for the STC coder implementations

The complexity of STC coders is relatively high, and the program discussed here was not optimised in any way for speed. However it is known [McQ, 92] that STC coders have been implemented on single DSP microprocessors and we anticipate that the STC coder versions developed in this thesis could be similarly implemented. At present, the program requires 0.27 seconds per 20 ms speech frame, running on a Pentium 90 personal computer and compiled non-optimally by the "Turbo C++" compiler. This is about 13 seconds of processing time per second of speech. Out of the 13 seconds, 2.5 seconds are required for the code-book search, 2.5 seconds for the pitch detector, 3.5 seconds for calculating the voicing probability and 1 second for the sinusoidal synthesis. These are the most time-consuming operations. An exercise carried out with an IMBE coder showed that an initial implementation of similar computational complexity can be speeded up five times by simplifying the C code; e.g. by using pointers rather than arrays and using look-up tables.

The simulation program was developed for evaluating the performance of the STC coders using files of natural speech as test data. These files contain speech which was different from any of the examples used to train the vector-quantisers. For example, a file of male mainly voiced speech [Away.pcm] of duration about 5 seconds was used to test the voicing aspects of the coders. As expected, the synthetic speech produced from the coders was most noticeably affected by the quantised short-term spectral envelope though any pitch estimation errors produced severe transitory distortion.

As reported in Section 6.5.8, comparing the fully quantised coders with the unquantised STC model, it was found that, for the 4.4 kb/s coder, the speech obtained was virtually indistinguishable, the 4 kb/s coder produced speech which was quite similar except for occasional bursts of coarseness and the 3.4 kb/s coder produced reasonably natural speech but with distortion clearly audible through headphones. For the 2.4 kb/s coder, despite the overall speech quality still being reasonably good, the existence of reverberance made it far from toll quality.

The simulation program was also modified to model the spectral envelope at each update-point by an all-pole gain response derived by LP analysis. The first of the three LP analysis techniques mentioned above was used. The program calculates a set of auto-correlation coefficients by applying a 512-point inverse FFT to the frequency-warped envelope derived by cubic spline interpolation as for the cepstral and DCT methods. The auto-correlation coefficients were then truncated to 14, and an LP filter was derived using the Levinson-Durbin algorithm. With unquantised LP coefficients, the speech quality was almost indistinguishable from the quality obtained using 30 unquantised cepstral coefficients. Since many techniques are available for efficiently vector-quantising LSP coefficients [PA, 93], and 30 bits are available for quantising the 14 coefficients in a 2.4 kb/s coder, this experiment indicated that the use of all-pole modelling is probably quite viable.

6.6 Discussion and conclusions

This chapter has addressed the problem of quantising the parameters of a low bit-rate STC speech coder and has discussed in particular the use of vector-quantisation with trained code-books. Various quantisation schemes have been investigated for coding the STC model introduced in Chapter 4 and modified in Chapter 5, at bit-rates 2.4 kb/s, 3.4 kb/s, 4 kb/s and 4.4 kb/s. A backward differential vector-quantisation technique was developed for the 30 DCT coefficients representing the spectral envelope at each update-point.

Results indicated that the DCT coefficient quantisation schemes implemented in this chapter were likely to be successful STC for coders at bit-rates of about 4 kb/s or above. However they were not promising for coders with bit-rates below about 3 kb/s. It was concluded that at such bit-rates, all-pole LP derived spectral representations may be more successful. Conventional LP methods produce reasonable results, but the use of more sophisticated all-pole modelling techniques such as DAP and IAP may be even better.

Chapter 7
Conclusions and future work

7.1 Conclusions

This thesis has surveyed the field of sinusoidal modelling techniques as are currently being applied to low bit-rate speech coding. Among these are STC, IMBE and PWI. All these techniques are active subjects of current research and commercial interest. Among the possible commercial applications are mobile radio communications since band-width will be limited and in ever increasing demand. The thesis has compared these sinusoidal modelling techniques and pointed out that there are interesting similarities and differences between them. There is much that can be learned from PWI and adapted to STC and vice-versa.

This thesis concentrates on STC and the various approaches adopted by McAulay and Quatieri to representing the STC model efficiently. STC is based on a fundamental sinusoidal model [McQ, 86A] which greatly simplifies the signal and, as we have confirmed, is capable of representing speech which is virtually indistinguishable from the original for update-points 10 ms apart. With 20 ms update-intervals, some distortion was found in female speech.

It has also been found that phase information at frequencies below 3 kHz has a significant effect on the perceived quality of sinusoidally modelled speech. Sinusoidally modelled speech reconstructed using phase information derived directly from the original speech sounds much more natural than the speech synthesised without this phase information. Replacing the original phase spectrum above 3 kHz by a random phase spectrum does not appear to affect this naturalness.

STC reduces the information contained in the fundamental sinusoidal model by various means, e.g. assuming the instantaneous frequencies of the sine-waves are harmonically related at the update-points for voiced speech, deriving phase spectra from magnitude only information at the decoder, and synthesising unvoiced and transition speech on a non-waveform basis.

As a preliminary stage to later work, an unquantised version of STC was developed in this thesis incorporating most of the features recommended by McAulay and Quatieri. The update-points are separated by 20 ms and the short-term spectral envelope is represented at each update-point by a truncated set of cepstral coefficients. Fundamental to the concept of STC is the regeneration of phase spectrum on the basis of a minimum phase assumption. Unvoiced speech is synthesised by summing 100 Hz apart sinusoids with random phases. Synthetic speech obtained from our version of STC was good, especially for male speech, apart from the effect of the occasional gross pitch estimation errors which marred the perceived quality.

The minimum phase assumption underlying STC phase regeneration neglects the effect of the glottal excitation and has been demonstrated in this thesis to be not totally accurate. To compensate for the inadequacy of the minimum phase assumption, two ways of including a glottal excitation model have been proposed. The first approach assumes that the glottal excitation signal is a Rosenberg pulse and attempts to remove its effect from the speech spectra to obtain truly minimum phase spectra. The second approach modifies the minimum phase spectra derived from the encoded envelopes by a second order all-pass filter to correct for the non-minimum phase glottal excitation. Methods of reducing other forms of distortion to the spectral envelope shape have also been investigated. Results have shown that the STC synthetic speech wave-forms are made closer to the original by these corrections . Perceptually better speech quality was also obtained.

The STC model has been fully quantised in various ways to achieve 2.4 kb/s, 3.4 kb/s, 4kb/s and 4.4 kb/s coders. This requires the parameters at each 20 ms update-point to be encoded using 48, 68, 80 and 88 bits respectively. The parameters are the pitch-frequency and voicing probability which are scalar quantised using 8 bits and 4 bits respectively and the spectral envelope DCT coefficients which are encoded by differential vector quantisation using the remaining bits. Frequency warping is applied to the spectral envelopes to try to improve the efficiency of the quantisation on a perceptual basis. An alternative to the use of DCT coefficients is to use high order LPC coefficients to represent the frequency warped spectral envelope.

Results confirmed that the speech quality obtained from the fully quantised coders was mainly dependent on the accuracy of the quantised spectral envelope. It was concluded that vector-quantisation of backward differences of DCT coefficients can probably be made effective at bit-rates of 4 kb/s and above. At lower bit-rates, e.g. at 2.4 kb/s, better speech quality can probably be obtained by an LPC modelling technique for the spectral envelopes. Preliminary experiments using basic high order LPC modelling technique were encouraging in this respect. Several more sophisticated, though quite computationally intensive LPC modelling techniques, e.g. discrete all-pole modelling [ElM, 91], are available for improved accuracy.

7.2 Future work

The application of low bit-rate speech coders to mobile telephony will continue to be an important research topic for many years to come. The performance of these coders with noise affected speech, the effect of channel errors, and the effect of tandeming, i.e. where speech is coded and decoded many times, should be considered during the design and development of low bit-rate coders. Low bit-rate speech coding techniques are also being applied to wide-band speech coding applications for example in conferencing where the speech band-width is increased from the normal approximately 3 kHz telephone bandwidth (300 Hz to 3.4 kHz) to a band-width of about 7 kHz (50Hz to 7.2 kHz).

Significant improvements have been made recently [KSSH, 96] in the representation of evolving speech spectra in terms of slowly and rapidly evolving waveforms as discussed in Chapter 2 in the context of PWI coding. PWI differs fundamentally from STC in that it is completely pitch synchronous with analysis window lengths of exactly one pitch-period rather than approximately 2.5 pitch-periods. We believe there is nothing to prevent the idea of REW and SEW being adapted to STC, and there are possible benefits which should be investigated.

The results in Chapter 3 revealed that the synthetic speech derived from the fundamental sinusoidal model analysed at 20 ms update-points (rather than 10 ms) has already some distortion, especially for female speech. This distortion will be inherited by the STC model which also uses 20 ms update-points. If an efficient way of encoding the STC model parameters at 10 ms update-points could be found this may be a good way of improving the speech quality. Since there will be only very few bits available in low bit-rate versions, we could not afford to encode all parameters at each 10 ms update-point. Some parameters, such as spectral envelope coefficients, may have to be encoded at 20 or even 30 ms intervals possibly with a very few bits at 10 ms intervals to give a little intermediate information. McAulay and Quatieri [McQ, 88] [McQ, 92] have investigated such a "frame-fill interpolation" technique which also allows the overlap-and-add technique to be used at the synthesis stage. This approach appears to be worthy of further investigation.

The idea of QPSS investigated briefly in Chapter 4 has possible advantages for the lowest bit-rate coders. It combines the coding of pitch-period and onset-time and thus tries to maintain a degree of pitch synchronism with the original speech. At present, it suffers degradation when speech changes from unvoiced to voiced as described in Chapter 4. One way of overcoming this problem is to transfer an estimated value of pitch-frequency for voice transitions and excitation point position for purely voiced frames of speech. A higher order phase interpolation formula may be needed to satisfy the boundary conditions for unvoiced to voice transition frames. Future work on QPSS may prove it to be a useful variation of STC.

In Chapter 5, the use of a Rosenberg pulse with fixed opening and closing times or a second order all-pass filter with fixed double poles has been shown to improve the phase derivation. Allowing the Rosenberg pulse or all-pass filter parameters to vary has been shown to produce modest further improvement with a simple optimisation scheme. A more sophisticated optimisation scheme may allow more significant improvements over the fixed parameter versions. The development and efficient implementation of such a scheme would be a worthwhile research topic. For the all-pass filter, the two poles would be simultaneously optimised to minimise the sum of squared phase differences. With an analytic formula for the phases, a gradient function may be available thus allowing a "conjugate gradients" optimisation technique to be employed.

It is feasible and likely to be beneficial to apply the improved phase model discussed in Chapter 5 to other sinusoidal coding techniques, such as PWI and IMBE. Preliminary results have been obtained for IMBE and further work is planned in this area.

For very low bit-rate versions of the STC coder, e.g. for a 2.4 kb/s STC coder, an all-pole LPC based modelling technique appears be an effective way of representing the spectral envelope. Although the simple technique tried in this thesis was reasonably effective, there are clearly advantages to be gained by investigating more sophisticated techniques such as the discrete all-pole model [ElM, 91]. The possibility of devising a new technique which jointly optimises the envelope approximation and the corresponding minimum phase spectrum may also be viable and this would be a valuable research goal. The use of the Itakura-Saito (I-S) distortion measure has well known advantages for the measuring the closeness of spectral envelope approximations. Perhaps a corresponding perceptually based phase distortion measure could be devised for the STC derived phase spectra.

The computational complexity of the STC modelling and encoding techniques investigated has not been made a major issue in this thesis. Indeed the software produced has been written for clarity rather than efficiency. If any of the techniques were to be adopted for a practical coder running in real time, for example on a floating point DSP microprocessor such as the TMS320C30 or the DSP32C, many modifications would have to be made. Nevertheless, we believe that the STC coders could be made within the capacity of a single DSP chip, and the optimisation of the code and some of the procedures to achieve this would be a useful part of any programme to continue this work on STC.

References

[AL, 89] Aiku, P. & Laine, U. K., "A New Glottal LPC Method for Voice Coding and Inverse Filtering", ISCAS'89, Proc. 1989, IEEE, pp. 1831 - 1834.

[AR, 82] Atal, B.S., & Remde, J.R., " A new model of LPC excitation for producing natural-sounding speech at low bit rates", IEEE Proc. ICASSP, 1982, pp. 614-617.

[AT, 82] Almeida, L. B. & Tribolet, J. M., "Harmonic Coding: A Low Bit-Rate, Good-Quality Speech Coding Technique", Proc. IEEE 1982, pp. 1664 - 1667.

[Atal, 82] Atal, B.S., " New directions in coding at low bit rates", Proc. IEEE, pp.1083-1986.

[Away.pcm] "We are away a year, ...", binary data file containing about 5 seconds of male speech sampled at 8 kHz with 16 bits/sample.

[BBD, 86] Benvenuto, N., Bertocci, G. & Daumer, W. R., " The 32 kbit/s ADPCM coding standard", AT&T Techn J., 65, No. 5, September, 1986, pp-12-21.

[BGGM, 80] Buzo, A., Gray, A.H., Gray, R.M. & Markel, J.D., " Speech Coding Based Upon Vector Quantisation", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP -28, No. 5, Oct. 1980, pp. 562 - 574.

[Boyd, 93] Boyd I, "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).

[BT, 94] British Telecom Laboratories, "Software implementation of vector quantiser", 1994.

[CG, 95] Chen, J.H. & Gersho, A., " Adaptive postfiltering for quality enhancement of coded speech", IEEE Tran. on speech and audio processing, vol. 3, No. 1, pp 59-71, January, 1995.

[Che, 93] Cheetham, B.M.G., "Speech analysis and resynthesis by a sinusoidal representation", Report for BT Laboratories, Liverpool University, August, 1993.

[Cho, 96] Choi, H.B., " The effect of phase randomisation at high frequency", Private communications, Liverpool University, 1996.

[CSW, 95] Cheetham, B. M. G., Sun X. Q. & Wong W. T. K., "Spectral envelope estimation for low bit-rate sinusoidal speech coders", Proc. Eurospeech'95, Madrid, Spain, pp. 693-696, September, 1995.

[CWF, 76] Crochiere, R.E., Webber, S.A. and Flanagan, J.L., "Digital coding of speech in sub-bands", Bell Syst Techn J, pp. 1069-1085, (October 1976).

[DPH, 93] Deller, J.R. Jr, Proakis, J.G. & Hansen, J.H.L., " Discrete-time processing of speech signals", Macmillan, Inc. 1993.

[EC, 93] Erzin, E. & Cetin, A.E., " Interframe differential vector coding of line spectrum frequencies", Proc. ICASSP'93, pp. II25-II28, Minneapolis, April, 1993.

[ElM, 91] El-Jaroudi, A. & Makhoul, J., " Discrete All-Pole Modelling", IEEE trans. on Signal Processing, Vol. 39, No. 2, 1991, pp. 411- 423.

[ElM, 89] El-Jaroudi, A. & Makhoul, J., " Discrete Pole-Zero Modelling and Applications", IEEE ICASSP, May, 1989, pp. 2162- 2165.

[FG, 66] Flanagan, J. L. & Golden, R. M., " Phase Vocoder", The Bell System Technical Journal, vol. 45, Nov 1966, pp. 1493 - 1509.

[FS, 84] Federal Standard: FED_STD-1015, "Telecommunications: Analog to digital conversion of voice by 2,400 bit/s linear predictive coding", Nov. 1984.

[FSACJT, 79] Flanagan, J., Schroeder, M.R., Atal, B., Crochiere, R.E., Jayant, N.S. & Tribolet, J.M., " Speech Coding", IEEE Trans. on communications, vol. com-27, No.4, April, 1979, pp. 710-737.

[Ger, 94] Gersho, A, "Advances in Speech and Audio Compression", Proc. IEEE, Vol. 82, 6, June, 1994, pp. 900 -918.

[GG, 86] Ghitza, O. & Gildstein, J. L., " Scalar LPC quantization based on formant JNDs", IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-34, Agu. 1986, pp. 697-708.

[GL, 88] Griffin, D.W. & Lim, J.S., "Multi-Band Excitation Vocoder", IEEE Trans. Accoust., Speech, Signal Process., vol. 36, No. 8, pp. 1223-1235, Aug. 1988.

[Good, 95] Goodyear, C.C., " Glottal excitation model", Private communications, Liverpool University, 1995.

[Gsp.pc] British Telecom supplied binary data file containing about 10 minutes of male and female speech sampled at 8 kHz with 16 bits/sample.

[Gsp.pcm] "Hello operator, operator, ..." British Telecom supplied binary data file containing about 100 seconds of male and female speech sampled at 8 kHz with 16 bits/sample.

[GWZ, 92] Gersho, A., Wang, S.H. & Zeger, K., "Vector quantization techniques in speech coding", in S. Furui & M. M. Sondhi (Eds): "Advances in Speech Signal Processing", Marcel Dekker Inc. 1992, (New York), pp. 49 - 83.

[Hed, 81] Hedelin, P., "A Tone-Oriented Voice-Excited Vocoder", Proc. IEEE, 1981, pp. 205 - 208.

[HL, 89] Hardwick, J,S. & Lim, J.S., "A 4800 bps improved multi-band excitation speech coder", Proc. IEEE Workshop on Speech Coding for Telecoms, Vancouver, Canada, 1989.

[HL, 91] Hardwick, J. S. & Lim, J. S., "The application of the IMBE speech coder to mobile communications" , Proc. ICASSP'91, pp. 249-252.

[Hol, 62] Holmes, J. N., "An investigation of the volume velocity waveform at the larynx during speech by means of an inverse filter", Proc. Speech communication seminar, vol. 1 (B4), Stockholm, Sweden, 1962.

[Hol, 88] Holmes, J.N., "Speech synthesis and recognition", Van Nostrand Reinhold (UK) Co. Ltd., 1988.

[IS, 70] Itakura, F. & Saito, S., "A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies", Electro. & Comm. in Japan, Vol. 53-A, No. 1, 1970, pp. 36 - 43.

[Jaz, 70] Jazwinsky, A. H., " Stochastic Processes and Filtering Theory", Academic Press, New York, 1970.

[JN, 84] Jayant, N. S. & Noll, P., "Digital coding of waveforms", prentice-hall Inc., 1984.

[KG, 91] Kleijn, W. B. & Granzow, W., "Methods for Waveform Interpolation in Speech Coding", Digital Signal Processing, Vol. 1 No. 4, Oct. 1991, pp. 215 - 230.

[KH, 94A] Kleijn, W. B. & Hssgen, J., "A general waveform-interpolation structure for speech coding", "Signal processing VII: Theories and applications", proc. of EUSIPCO-94, vol. III, September, 1994, Edinburgh, UK, pp. 1665-1668.

[KH, 94B] Kleijn, W. B. & Hssgen, J., "Transformation and Decomposition of the Speech Signal for Coding", IEEE Trans. Signal Processing Letters. Vol. 1, No. 9, Sept. 1994, pp. 136 - 138.

[KH, 95] Kleijn, W.B. & Haagen, J., "A speech coder based on decomposition of characteristic waveforms", Proc. ICASSP'95, pp.508-511.

[Kin, 87] Kingsbury, N.G., "Robust 8000 bit/s subband speech coder", IEE Proc. F, 134, pp. 352-366, July 1987.

[Kle, 91A] Kleijn, W. B., "Continuous Representations in Linear Predictive Coding", Proc. IEEE, 1991, pp. 201 - 204.

[Kle, 91B] Kleijn, W. B., "Analysis-by synthesis speech coding based on relaxed waveform-matching constraints", Ph.D. dissertation, Delft University of Technology Delft, The Netherlands, 1991, pp. 55 - 62.

[Kle, 93] Kleijn, W. B., "Encoding speech using prototype waveforms", IEEE Trans. SAP-1 no. 4, pp 386-399, 1993.

[KSSH, 96] Kleijn, W.B., Shoham, Y., Sen, D. & Hagen, R., " A low-complexity waveform interpolation coder", Proc. of ICASSP, 1996, Atlanta, US, pp. 212-215.

[LBG, 80] Linde, Y., Buzo, A. & Gray, R., " An Algorithm for Vector Quantizer Design", IEEE trans. on Communications, Vol. COM.- 28, No. 1, January 1980, pp 84 - 95.

[LL, 94] Li, H. & Lockhart, G.B., " Non-linear interpolation in prototype waveform interpolation (PWI) encoders", IEE Colloquium, 1994.

[Lo, 93] Lo, K.Y., "Pitch-synchronous Speech Coding at Very low Bit Rates", Ph.D thesis, Liverpool University, UK, 1993.

[Mak, 75] Makhoul, J., "Linear Prediction: A Tutorial Review", Proc. IEEE, Vol. 63, No. 4, April 1975, pp. 561 - 580.

[MAT, 90] Marques, J. S., Almeida, L. B. & Tribolet, J. M., "Harmonic Coding at 4.8 KB/S", Proc. IEEE, 1990, pp. 17 - 20.

[Mat, 95] "The student Edition of MATLAB", Ver. 4, User's Guide, Prentice Hall, 1995.

[McA, 89] McAulay, R. J., "Computationally efficient sine wave synthesis for acoustic waveform processing", Patent Co-operation treaty (PCT) WO 89/09985, 19 October, 1989.

[McC, 90] McAulay, R. J. & Champion, T., " Improved interoperable 2.4 kb/s LPC using sinusoidal transform coder techniques", Proc. IEEE, 1990, pp. 641-643.

[McCQ, 93] McAulay, R. J., Champion, T. & Quatieri, T. F., "Sinewave amplitude coding using line spectral frequencies", Proc. IEEE Workshop on Speech Coding for Telecommunications Speech Coding for the Network of the future, Canada, Oct. 1993, pp. 53 - 54.

[McPQS, 91] McAulay, R.J., Parks, T., Quatieri, T.F. & Sabin, M., " Sine-Wave Amplitude Coding at Low Data Rates", Extraction from the book "Advances in Speech Coding", Editors, B. S. Atal, V. Cuperman & A. Gersho, 1991.

[McQC, 94] McAulay, R.J., Quatieri, T.F. & Champion, T. G., " Sinewave amplitude coding using high-order all pole models", Proc. of EUSIPCO-94, Edinburgh, UK, September 1994, pp. 395-397.

[McQ, 85] McAulay, R. J. & Quatieri, T. F., "Mid-Rate Coding Based on a Sinusoidal Representation of Speech", Proc. IEEE, 1985, pp. 945 - 948.

[McQ, 86A] McAulay, R. J. & Quatieri, T. F., "Speech Analysis/Synthesis based on a Sinusoidal representation", IEEE Trans. vol. ASSP-34, No. 4, August, 1986. pp.744-754.

[McQ, 86B] McAulay, R.J. & Quatieri, T.F., " Phase Modelling and its Application to Sinusoidal Transform Coding", Proc. IEEE, ICASSP, Tokyo, Japan, April, 1986. pp. 1713-1715.

[McQ, 87] McAulay, R.J. & Quatieri, T.F., "Multirate sinusoidal Transform Coding at Rates from 2.4 kbps to 8 kbps", Proc. IEEE, 1987. pp. 1645-1648.

[McQ, 88] McAulay R.J. & Quatieri, T.F., " Computationally efficient Sine-Wave and its Application to Sinusoidal Transform Coding", IEEE 1988, pp. 370-373.

[McQ, 90] McAulay R.J. & Quatieri, T.F., " Pitch Estimation and Detection

Based on a Sinusoidal Speech Model", Proc. IEEE, ICASSP, Albuquerque, NM, April, 1990, pp. 249-252.

[McQ, 92] McAulay R. J. & Quatieri, T. F., "Low-Rate Speech Coding based on the Sinusoidal Model", in "Advance in Speech Signal Processing" Edited by Sadaoki, Furui, M. Mohan Sondh, Marcel Dekker, Inc. 1992, (New York), pp. 165 - 208.

[McQ, 93] McAulay R. J., Quatieri, T. F., "The application of subband coding to improve quality and robustness of the sinusoidal transform coder", IEEE proc. Vol. 2, 1993, pp. 439-442.

[MG, 76] Markel, J.D. & Gray, A.H., "Linear prediction of speech", Springer, Berlin, 1976.

[MM, 63] Miller, J. E. & Mathews, M. V., " Investigation of the glottal waveshape by automatic inverse filtering ", J. Acoust. Soc. Am. Suppl. vol. 35, pp. 1876(A), 1963.

[MRG, 85] Makhoul, J., Roucos, S. & Gish, H., "Vector Quantization in Speech Coding", Proc. IEEE, Vol. 73, No. 11, Nov. 1985, pp. 1551 - 1588.

[MSDI, 91] Maritime System Development Implementation: "Inmarsat M Voice Codec, version 3.0", 1991.

[Mvoice.pcm] British Telecom supplied binary data file containing about 20 seconds of male speech sampled at 8 kHz with 16 bits/sample.

[OS, 75] Oppenheim, A. V. & Schafer, R. W., "Digital Signal Processing", Prentice-Hall, Englewood Cliffs, N.J., 1975.

[PA, 93] Paliwal, K. K. & Atal, B.S., "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", IEEE Trans. on speech and audio processing, vol. 1, No. 1, January, 1993, pp. 3-14.

[Pal, 90] Paliwal, K. K., "Speech Processing Techniques", in "Advances in speech, hearing and language processing", edited by W.A.Ainsworth, Vol.1, 1990, JAI press Ltd., pp. 1-78.

[Pau, 81] Paul, D.B., " The Spectral Envelope Estimation Vocoder", IEEE Trans. vol. ASSP-29, No. 4, August, 1981. pp. 786-794.

[Pol, 95] Pollard, M.P., "Waveform interpolation methods for high quality speech synthesis", Internal report, Liverpool University, 1995.

[PTVF, 92] Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B. P.,

" Numerical Recipes in C ", Cambridge University Press, Second edition, 1992.

[PWC, 95] Parris, C.I., Wong, D. & Chambon, F., " A robust 2.4 kb/s LP-MBE with iterative LP modelling", Proc. Eurospeech'95, Madrid, Spain, pp. 677-680, September, 1995.

[QMc, 86] Quatieri, T. F. & McAulay, R.J., "Speech Transformations based on a Sinusoidal Representation", IEEE Trans, vol. ASSP-34, No. 6. December 1986. pp.1449-1464.

[QMc, 87] Quatieri, T.F. & McAulay, R. J., "Mixed-phase deconvolution of speech based on a sine-wave model", IEEE Proc., 1987. pp. 649-652.

[QMc, 92] Quatieri, T. F. & McAulay, R. J., "Shape Invariant Time-Scale and pitch Modification of Speech", IEEE Trans. on signal processing. vol. 40, No. 3, March, 1992. pp. 497-510.

[Rah, 91] Rahim, M., " Neural networks in articulatory speech synthesis", Ph.D. thesis, Liverpool University, UK, 1991.

[Ros, 71] Rosenberg, A. E., "Effect of Glottal Pulse Shape on the Quality of Natural Vowels", The Journal of the Acoustical Society of America, Vol. 49, No. 2, 1971, pp.583 - 590.

[Rot, 73] Rothenberg, M., " A new inverse- filtering technique for deriving the glottal air flow waveform during voicing", J. Acoust. Soc. Am. Vol. 53, No. 6, 1973, pp. 1632 - 1645.

[RS, 78] Robiner, L.R. & Schafer, R.W., "Digital processing of speech signals", Prentice-Hall, Inc., 1978.

[SAH, 79] Schroeder, M.R., Atal, B. S. & Hall, J. L., "Optimizing digital speech coders by exploiting masking properties of the human ear", J. Acoust. Soc. Amer., vol. 66, Dec. 1979, pp. 1647-1652.

[SC, 93] Sun, X. Q. & Cheetham, B.M.G., " Software of fundamental sinusoidal model ", Internal report , Liverpool University, 1993.

[SC, 94A] Sun, X. Q. & Cheetham, B.M.G., "Complex Cepstrum", Internal report to BT Laboratories, Liverpool University, 1994.

[SC, 94B] Sun, X. Q. & Cheetham, B.M.G., " Software of quasi-pitch-synchronous sinusoidal (QPSS)", Internal report , Liverpool University, 1994.

[SC, 95] Sun, X. Q. & Cheetham, B.M.G., " Software of STC model", Internal report , Liverpool University, 1995.

[SC, 96] Sun, X. Q. & Cheetham, B.M.G., " Software of 2.4 kb/s fully quantised STC coder", Internal report , Liverpool University, 1996.

[Sch, 85] Schroeder, M. R., "Linear Predictive Coding of Speech: Review and Current Directions", IEEE Communication Magazine, Vol. 23, No. 8, August, 1985, pp. 54 - 61.

[Sch, 95] Schroder, G., "The standardisation process for the ITU-T 8-kbit/s speech codec", Proc. IEEE Workshop on Speech Coding for Telecommunications, pp. 1-2, Annapolis, USA, September 1995.

[SMcDQ, 96] Singer, E., McAulay, R.J., Dunn, R.B. & Quaieri, T.F., " Low rate coding of the spectral envelope using channel gains", Proc. of ICASSP, 1996, Atlanta, US, vol. 2, pp. 769-772.

[Sho, 93A] Shoham, Y., "High-Quality Speech Coding at 2.4 to 4.0 KBPS based on Time-Frequency Interpolation", Proc. IEEE, ASSP, April, 1993, Vol. 2, pp. 167-170.

[Sho, 93B] Shoham, Y., "High-quality speech coding at 2.4 kbps based on time-frequency interpolation", in Proc. European Conf. on Speech Communication and Technology, Berlin, Germany, September 1993, vol. 2, pp. 741-744.

[Spa, 94] Spanias, A. S., "Speech Coding: A Tutorial Review", Proc. IEEE, Vol. 82, No. 10, Oct. 1994, pp. 1541 - 1582.

[Tre, 82] Tremain, T.E., "The Government Standard linear predictive coding algorithm: LPC-10", Speech Technology, vol. 1, No. 2, April 1982, pp.40-49.

[WMCS, 96] T. K. Wong, R. M .Mack, B. M. G.Cheetham & X. Q. Sun, "Low rate speech coding for telecommunications", BT Technology Journal, Vol. 14, No.1, January 1996, pp. 28-44.

[Xyd, 89] Xydeas, C., "Modern Developments in Speech Coding", IEE Colloquium on "Spectral Estimation Techniques for Speech Processing", Feb.1989, pp. 1 - 8.

[YKE, 90] Yeldener, S., Kondoz, A. M. & Evans, B. G., "Sine Wave Excited Linear Predictive Coding of Speech", International Conference on

Spoken Language Processing, Kobe, Japan, November, 1990, pp. 4.2.1-4.2.4.

[ZN, 77] Zelinski, R. and Noll, P., "Adaptive transform coding of speech signals", IEEE Trans Acoustics, Speech and Signal Processing, ASSP-25, No. 4, pp. 299-309, August 1977.

Appendix

PROGRAM ----PHASTEST.C

Test program for phase interpolation technique by McAulay et al. Generates a cos wave of starting frequency "w0" rad/sample and starting phase ph0 radians. At the end, the frequency is w1 & phase is "ph1". The aim is to generate a cosine wave whose starting frequency & phase is w0 & ph0 respect., whose frequency & phase at the end of the frame (LBLOCK=150 samples) is w1 & (ph1+2*PI*MM), and whose variation in instantaneous frequency across the frame is minimised. The program demonstrated the effect of choosing different values of MM, and then calculates the best value to produce the final display.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include <bios.h>
#include <ctype.h>

#define  PI         3.141592654
#define  LBLOCK     150

float output[LBLOCK], FLBLOCK;
float w0,w1,ph0, ph1;
FILE *prt;
```

```
int init_graphics(void)
{  int driver, mode;
   detectgraph(&driver,&mode); if(driver<0) return(-1);
   if((mode != EGAHI)  && (mode != VGAHI)) return(-1);
   if(registerbgidriver(EGAVGA_driver)<0)  return (-1);
   initgraph(&driver,&mode,""); if(graphresult()<0) return 0;
   return(1);
}
```

```
Comments: Draw time-varying cosine waveform
void Drawcos(void)
{ int n,ix;

  setviewport(0,0,602,400,1);
  clearviewport();
  setcolor(15);
  rectangle(0,0,450,200);

  for(n=0; n<LBLOCK; n++)
  { ix= 100-(int)(output[n]);
    if(n==0) moveto(0,ix); else lineto(3*n, ix);
  }
}
```

```
Comment: Calculate alfa & beta by equn 34 with M=MM
void Calculate( int MM)
{  int n;
   float temp1, temp2, fn, alfa, beta,theta;

   temp1 = ph1 - ph0 - w0*FLBLOCK + 2.0*PI*(float)MM;
   temp2=w1 - w0;
   alfa=(3.0*temp1/FLBLOCK - temp2)/FLBLOCK;
   beta= ( -2.0*temp1/FLBLOCK + temp2)/(FLBLOCK*FLBLOCK);
   fprintf(prt,"M=%d\n", MM);

  Comment: Calculate theta[n] by eqn 37 and hence output cos waveform:-
  for(n=0;n<LBLOCK; n++)
  { fn=(float)n;
    theta = ph0 + w0*fn + alfa*fn*fn + beta*fn*fn*fn;
    output[n]=50.0*cos(theta);
    fprintf(prt,"%f\n", output[n]);
  }
}
```

```
=====================================================================
void main(void)
{  int MM;
   float xstar;
   char outbuf[20];

   w0=PI/20;
   ph0=PI/2.0;
   w1=PI/5.0;
   ph1=PI/4.0;
   FLBLOCK=(float)LBLOCK;

   prt=fopen("c:\\xiaoqin\\sinusoid\\bigm.txt", "wt");
   if(prt==NULL) printf("file deos not open!");

   init_graphics( );

  Comment:  Try different values of MM :-
  printf("\n\n\n\n\n\n\n\n\n\n\n\n" );

  for(MM=-4; MM<21; MM++)
  { Calculate(MM);
    Drawcos();
    outtextxy(5,210,"MM=");
    sprintf(outbuf, "%4d", MM);
    outtextxy( 25, 210, outbuf);
    getch( );
  }

  Comment: Calculate xstar (eqn 36) and integerise to get correct value of MM:-
   xstar=(0.5/PI)*((ph0+w0*FLBLOCK-ph1)+((w1-w0)*FLBLOCK/2.0));
   MM=(int)(xstar+0.5);
   Calculate(MM);  Drawcos();

   printf(" MM= %d.  THE BEST ONE!\n",  MM);
   printf(" w0=%6.4f,  ph0=%6.4f,    w1=%6.4f, ph1=%6.4f\n", w0,ph0,w1,ph1);
   printf(" x* = %7.4f.  Therefore, the best value of MM is %d\n", xstar, MM);
   printf(" The variation in instantaneous frequency is minimised.\n");
   getch( );

   clearviewport( );   closegraph( );   fclose(prt);
}
=====================================================================
```

List of publications

- Cheetham B. M. G. & Sun X. Q., " Spccch proccssing Tcchniquc applied to the analysis of pulmonary sounds", invited paper presented at the 18th International Conference on Lung Sounds, Alberta, Canada, 1993.

- Spence D. P. S., Rees K., Sun X. Q., Cheetham B. M. G., Calverley P. M. A. & Earis J. E., "Upper airways modelling by LPC filtering in heavy snoring and obstructive sleep apnea (OSA)", Proc. 18th Int. Conf. on Lung Sounds, Alberta, Canada, 1993.

- Cheetham, B. M. G., Sun X. Q. & Earis J. E., "Real time analysis of lung sounds", Proc. First CORSA WP III Symposium on Signal Processing in Lung Sound Analysis, p2/1-p2/6, Helsinki, Finland, June, 1995.

- Cheetham, B. M. G., Sun X. Q. & Wong W. T. K.,"Spectral envelope estimation for low bit-rate sinusoidal speech coders", Proc. Eurospeech'95, Madrid, Spain, pp. 693-696, September, 1995.

- Sun X. Q., Cheetham B. M. G. & Wong W. T. K., "Spectral envelope and phase optimisation for sinusoidal speech coding", Proc. IEEE Workshop on Speech Coding for Telecommunications, pp. 75-76, Annapolis, USA, September 1995.

- Wong, W.T. K., Mack, R. M., Cheetham, B. M. G. & Sun, X. Q., "Low rate speech coding for telecommunications", BT Technology Journal, Vol. 14, No.1, January 1996, pp. 28-44.

- Sun, X. Q., Cheetham, B. M. G., Evans, K. G. & J.E.Earis, J. E. , "Estimation of analogue pre-filtering characteristics for CORSA", Proc. Second CORSA WP III Symposium on Signal Processing in Lung Sound Analysis, Helsinki, Finland, June, 1996.

- Plante, F., Kessler, H., Sun, X. Q., Cheetham, B. M. G. & Earis, J. E., "Inverse filtering applied to upper airway sounds", Proc. Second CORSA WP III Symposium on Signal Processing in Lung Sound Analysis, Helsinki, Finland, June, 1996.

- Sun, X.Q., Evans, K.G., Cheetham, B.M.G. & Earis, J.E., "Characterisation of pre-filter response for lung sounds measurements", Proc. 21st Int. Conf. on Lung Sounds, Chester, England, September, 1996.

- Plante, F., Kessler, H., Sun, X.Q., Cheetham, B.M.G. & Earis, J., "The analysis of upper airway sounds by inverse filtering", Proc. 21st Int. Conf. on Lung Sounds, Chester, England, September, 1996.

- Sun, X. Q., Plante, F., Cheetham, B. M. G. & W. T. K. Wong, " Phase modelling of speech excitation for low bit-rate sinusoidal transform coding", Proc. IEEE ICASSP'97, Munich, April, 1997. (Accepted for publication)

- Cheetham, B. M. G., Choi, H. B., Sun, X. Q., Goodyear, C. C., Plante, F. & Wong, W. T. K., "All-pass excitation phase modelling for low bit-rate speech coding", IEEE Symposium on Circuit and Systems, ISCAS'97, Hong Kong, June, 1997, (under review)