Source Reliant Error Control
For Low Bit Rate
Speech Communications

by
Leh Kui ONG

Thesis submitted to the University of Surrey
for the degree of
Doctor of Philosophy

Centre for Satellite Engineering Research
University of Surrey
Guildford, Surrey
United Kingdom

August 1994

Copyright © 1994 L.K. Ong
In memory of

my late Mother...
Abstract

Contemporary and future speech telecommunication systems now utilise low bit rate (LBR) speech coding techniques in efforts to eliminate bandwidth expansion as a disadvantage of digital coding and transmission. These speech coders employ model-based approaches in compressing human speech into a number of parameters, using a well-known process known as linear predictive coding (LPC). However, a major side-effect observed in these coders is that errors in the model parameters have noticeable and undesirable consequences on the synthesised speech quality, and unless they are protected from such corruptions, the level of service quality will deteriorate rapidly. Traditionally, forward error correction (FEC) coding is used to remove these errors, but these require substantial redundancy. Therefore, a different perspective of the error control problems and solutions is necessary.

In this thesis, emphasis is constantly placed on exploiting the constraints and residual redundancies present in the model parameters. It is also shown that with such source criteria in the LBR speech coders, varying degrees of error protection from channel corruptions are feasible. From these observations, error control requirements and methodologies, using both block- and parameter-orientated aspects, are analysed, devised and implemented. It is evident, that under the unusual circumstances which LBR speech coders have to operate in, the importance and significance of source reliant error control will continue to attract research and commercial interests.

The work detailed in this thesis is focused on two LPC-based speech coders. One of the ideas developed for these two coders is an advanced zero redundancy scheme for the LPC parameters which is designed to operate at high channel error rates. Another concept proposed here is the use of source criteria to enhance the decoding capabilities of FEC codes to exceed that of maximum likelihood decoding performance. Lastly, for practical operation of LBR speech coders, lost frame recovery strategies are viewed to be an indispensable part
of error control. This topic is scrutinised in this thesis by investigating the behaviour of a specific speech coder under irrecoverable error conditions. In all of the ideas pursued above, the effectiveness of the algorithms formulated here are quantified using both objective and subjective tests. Consequently, the capabilities of the techniques devised in this thesis can be demonstrated, examples of which are: (1) higher speech quality produced under noisy channels, using an improved zero-redundancy algorithm for the LPC filter coefficients; (2) as much as 50% improvement in the residual BER and decoding failures of FEC schemes, through the utilisation of source criteria in LBR speech coders; and (3) acceptable speech quality produced under high frame loss rates (14%), after formulating effective strategies for recovery of speech coder parameters.

It is hoped that the material described here provide concepts which can help achieve the ideals of maximum efficiency and quality in LBR speech telecommunications.
Acknowledgements

First and foremost, to my parents, their moral support and love, I express my gratitude and indebtedness. It is through their visions and encouragements that I am able to overcome times of uncertainties and reach this stage of education without the comforts of home.

I am also grateful to Professor Evans and Dr. Kondoz for their guidance and the invaluable opportunities given to me, without which this thesis would not have eventually resulted. My acknowledgements to all my colleagues in the Speech Coding Research Group and the Centre for Satellite Engineering Research, especially Wei and Sam, whose friendship and advice has enriched my life during these years. Last, but not least, my appreciation to everyone at the University of Surrey, who have in one way or another, contributed to a most rewarding experience for me in the U.K.
Contents

1 Introduction .............................................. 1
   1.1 Motivation and Approaches .................................. 3
   1.2 Achievements and Contributions ................................ 4
   1.3 Thesis Outline ........................................... 5

2 Speech Coding For Telecommunications .................. 7
   2.1 Characteristics of Speech Coding Schemes .................... 8
      2.1.1 Transmission Rate versus Speech Quality .............. 8
      2.1.2 Robustness to Channel Impairments .................. 9
      2.1.3 Delay ............................................... 10
      2.1.4 Complexity ........................................ 10
   2.2 Linear Prediction of Speech .................................. 11
      2.2.1 Speech Production Model ............................... 12
      2.2.2 Linear Predictive Analysis .............................. 13
      2.2.3 Line Spectral Frequencies .............................. 16
   2.3 The CELP Coder ............................................ 19
      2.3.1 General Description ................................... 19
      2.3.2 Supporting Concepts .................................. 21
2.3.2.1 Analysis-by-Synthesis Optimisation ........................................ 21
2.3.2.2 Vector Quantisation ............................................................. 22
2.3.2.3 Perceptual Weighting ............................................................ 24
2.3.3 Short-Term Predictor ............................................................... 25
2.3.4 Long-Term Predictor ............................................................... 27
2.3.5 Codebook Excitation ............................................................... 30
2.4 The Multi-Band LPC Vocoder ......................................................... 31
2.4.1 General Description ............................................................... 32
2.4.2 Spectral Envelope Estimation .................................................. 35
2.4.3 Pitch Determination ............................................................... 36
2.4.3.1 Initial Pitch Estimation ....................................................... 37
2.4.3.2 Pitch Refinement ............................................................... 38
2.4.4 Voicing Determination ........................................................... 40
2.4.5 Excitation Gain ................................................................. 41
2.4.6 Speech Synthesis ................................................................. 41
2.5 Remarks ............................................................................... 42

3 Error Control In Speech Coders .................................................. 44
3.1 The Quality-Robustness Dilemma ............................................. 45
3.2 Forward Error Correction Coding ........................................... 47
3.2.1 BCH Codes ........................................................................ 49
3.2.1.1 Galois Field Basics ........................................................ 50
3.2.1.2 Description of BCH Codes .............................................. 51
3.2.1.3 Encoding ................................................................. 52
3.2.1.4 Decoding ................................................................. 53
4 Error Recovery Of Line Spectral Frequencies

4.1 Properties of LSFs
   4.1.1 Monotonicity Criterion
   4.1.2 Inter-frame Correlations
   4.1.3 Spectral Sensitivity
      4.1.3.1 Group Delay
      4.1.3.2 Power Spectrum

4.2 Interpolation Methods for Corrupted LSFs
   4.2.1 Vector Interpolation
   4.2.2 Pair-wise Interpolation
   4.2.3 Single Element Interpolation

4.3 Error Recovery Algorithm

4.4 Remarks

5 Source Aided Channel Coding

5.1 Limitations of Zero Redundancy Techniques

5.2 Conceptual View
   5.2.1 Error Control Configuration
   5.2.2 Improved Maximum Likelihood Decoding

5.3 SACC in the MB-LPC Coder
   5.3.1 Source Criteria in the MB-LPC Coder
   5.3.2 Block-based Decoding
   5.3.3 Trellis-based Decoding

5.4 SACC in the CELP Coder

5.5 Remarks
Chapter One

Introduction

Speech telecommunications have undergone dramatic transformations and advancements in scientific techniques and engineering technologies. Since the inception of the telephone in 1876 and the deployment of digital telephone transmission using Pulse Coded Modulation (PCM) in 1962, the world has and is still witnessing momentous events that will change the nature of communications technology and its impacts on the communications environment. From being a minority activity in the 1970s, to their implementations using analogue cellular systems in the 1980s and their ensured places in society through the definition of future digital systems, mobile radio communications has promised to be one of the biggest growth areas in electronics into the next century.

With first generation analogue systems, such as UK’s Total Access Communications (TACS) and USA’s Advanced Mobile Phone Service (AMPS), being unable to meet future capacity demands, concerted efforts are being made to address this challenge on a global scale. In Europe, work on radio transmissions using digital technology has resulted in the creation of the pan-European cellular network, named GSM (Global System for Mobile communications, formerly Groupe Speciale Mobile). As evident in the adoption of new standards in the USA and Japan, digital technology offers the immensely attractive capability of increasing the capacity of radio spectrum by carrying more voice links within the same bandwidth. Indeed, activities involving the use of improved digital speech coding techniques to reduce the voice transmission rate in GSM by half (H-GSM) are well under
In pursuing the future trends of these second-generation systems, transmissions groups such as the European Telecommunications Standards Institute (ETSI) and the International Radiocommunications Consultative Committee (CCIR), have begun research and standardisation activities into what is broadly known as third-generation personal communications networks (PCN). Called Universal Mobile Telecommunication Systems (UMTS) in ETSI terminology, these systems will signify an innovation no less than a complete revolution in telephony and data communications.

The UMTS represent the future of two powerful instruments of social change – the telephone and the computer – and will focus on meeting the communications requirements of people on the move. However, as experienced in digital cellular mobile networks, the environments in which UMTS operate are limited to an unusual degree by spectrum, interference and regulatory constraints. With voice communications being predicted as the largest proportion of the service offerings by UMTS, the ability to employ bit rate reduced representations of speech signals to circumvent spectral restrictions has spurred and/or renewed a large amount of interests in LBR speech coding research.

In all of these mobile voice communications systems, bit rates in the range of 2 to 8 kilobits per second (kb/s) are of high importance in the interests of the economy of frequency use and the significant overhead capacity needed for error protection. Thus as essential parts of future generation mobile communications systems, efficient speech and channel coding schemes will have to be optimised for robustness against channel impairments. While rapid advancements in speech coding technologies have enabled high speech quality at the desired bit rates, the progress in error control and/or channel coding innovations have been relatively gradual. This threatens the level of service quality expected, potentially debasing the role of such systems in telecommunications. The research material presented in this thesis is an effort to inject new ideas, concepts and techniques that will, hopefully, address the relative mismatch in advancements in speech and channel coding.
1.1 Motivation and Approaches

Advanced digital speech coding techniques responsible for the drastic reductions in transmission rates all utilise a common approach: parametric representations of speech signals. In public switched telephone networks (PSTN), sample by sample coding using PCM or adaptive differential PCM (ADPCM) means that errors incurred in the transmission medium have localised or limited effects. On the contrary, corrupted parameters in low bit rate (LBR) speech coders affect many, if not all, of the speech samples being coded. Thus, degradations in synthesised speech in LBR coders are often more noticeable, uncomfortable or even annoying. Equivalently, LBR speech coders are said to exhibit a higher degree of error sensitivity. To alleviate this, some form of error control must be introduced.

Traditionally, forward error correction (FEC) coding would have to be employed. This, however, has its costs: FEC introduces redundant bits in an information bit stream, and at a given gross bit rate, the speech coder would have to be operated at a lower rate than actually provided for. This is compounded by the fact that FEC schemes require substantial redundancy for reasonable performance. Clearly, this calls for other forms of effective error control or channel coding techniques.

In this thesis, related but distinctively different approaches have been undertaken. Foremost, human speech has intrinsic redundancies which can be exploited for error control. Although the aim in LBR speech coding algorithms is to extract these redundancies, the segmentation of speech signals into blocks of fixed lengths for processing means that there is some amount of residual redundancy left. Due to the frequent update rate of the speech coder parameters, these residual redundancies are effective indications of what the parameters would or are likely to be in the event of corruptions. Additionally, during quantisation and coding, speech coder parameters can be translated in ways such that errors can be detected in an intrinsic manner. In this thesis, these residual redundancies are collectively termed as source criteria. Source criteria can be utilised in zero redundancy schemes and for improved error correction performance of FEC coding schemes.
In mobile radio environments, the presence of a large number of errors (or error bursts) in the received bit stream at the decoder often go beyond the performance limits of the designed FEC coding schemes. To counteract such potentially disastrous situations for the LBR speech decoder, frame recovery or reconstruction techniques are also looked into so that graceful performance and/or shut-down of the speech decoder can be effected during irrecoverable error bursts. With the application of all or a combination of these techniques, it is hoped that contemporary and future voice communication systems will be able to provide high speech quality at very low transmission rates under various channel error conditions.

1.2 Achievements and Contributions

The original objective intended for this research was to formulate and implement advanced error control methods for a speech coder, namely, the Multi-Band LPC vocoder (voice coder). However, upon realisation of the limited nature and applicability of this approach, it was decided that a more generic, system-wise treatment of the problem in general be looked into. This has resulted in work of various emphasis, and is summarised as follows:

- A generalised approach to error control methodologies, considerations and techniques in low bit rate speech coders. This is presented using both block- and parameter-orientated approaches.

- The formulation and implementation of a new redundancy-less error-control technique for Line Spectral Frequencies by using intrinsic coding redundancy and the derived properties of vector quantisation.

- The identification of the error sensitivities of the MB-LPC parameters, and the apparent error control requirements needed.

- The concept of using source criteria in speech coders to extend the error correction capabilities of FEC codes to beyond that obtainable by soft decision decoding.
1.3 Thesis Outline

- A treatment of techniques for handling erased speech frames for the MB-LPC vocoder in irrecoverable error conditions.

Some of the above concepts and techniques have been published in conferences proceedings and journal papers, a list of which is given in Appendix A.

1.3 Thesis Outline

In Chapter 2, topics relating specifically to contemporary speech coding techniques are discussed. Two LPC-based low bit rate speech coding algorithms are used as example systems. These two speech coders, together with appropriate error control techniques, are used for analysing the effects of transmission corruptions on the speech synthesis process.

The general problem of error control in LBR speech coders is scrutinised in Chapter 3. Constraints affecting the channel performance, resulting in what is termed here as the quality-robustness dilemma in mobile communications, is described. A brief review of conventional FEC coding schemes is also given, following which considerations as to how effective error control can be implemented is detailed. Descriptions of system-level configurations and zero redundancy methods are also provided.

Chapter 4 is devoted to the description of an advanced technique developed for mitigating the subjective effects of errors in LSFs. Existing methods for stabilising LSFs are also discussed with and compared with the new algorithm. In Chapter 5, a new aspect in exploiting source information for channel coding, is introduced. The coding advantage of source aided channel coding, over that of conventional soft decision decoding, is presented in concept and then analysed through simulation results. Both block and convolutional codes are used to illustrate the benefits this technique can bring about for low bit rate speech communications.

Strategies for the recovery of speech from lost or erased frames is treated in Chapter 6. This aspect of error control is especially important in mobile radio applications where the
incidences of channel corruptions is high. The reconstruction of speech samples during such circumstances is demonstrated using a specific speech coder, namely, the MB-LPC vocoder. In the final chapter, a summary of the preceding chapters is given. In addition to the conclusions made, possible avenues for further and future research in this area are also proposed.
Chapter Two

Speech Coding For Telecommunications

Digital speech communications have, for thirty years, been dominated by the 64-kb/s PCM and the 32-kb/s ADPCM systems. These systems employ a broad class of digital coders known as waveform coders. In essence, these coders strive for facsimile reproduction of the signal waveform, irrespective of whether the signal is speech, music, tones or voiceband data. Waveform coders, thus, tend to be robust for a wide range of signal characteristics and input environments.

The drive for low bit rate speech coding has favoured a different class of coders known as source coders. These coders depend on a priori knowledge of how speech signals are generated. This implies that the signals to be coded are first matched with a speech-specific model, parameterised and then coded for transmission. Non-speech signals would have to be separately considered if source coders are to conform with the specifications in current PSTN and other systems based on waveform coders.

What, then, are the effects that source coding of speech signals have on voice communications, especially on a large scale? What is so specific about source coding such that it differs significantly from waveform coding? This chapter aims to provide some answers to these questions. In addition, two LBR speech coding algorithms, the CELP and the MB-
2.1 Characteristics of Speech Coding Schemes

LPC, are used to illustrate source coding and the ability to produce high quality speech at low bit rates.

2.1 Characteristics of Speech Coding Schemes

Speech coding is one of the most important sub-systems in modern digital speech communication systems and will play a key role in future mobile public networks. In the era of customer-driven quality standards and awareness of energy efficiency, the implications, requirements and constraints of a speech coding algorithm may mean either wide acceptance or immediate rejection. In addition, a well designed speech coding algorithm encompasses matching the system’s operational specifications and reconciling any conflicting algorithm-system interests. The following is a typical, but not exhaustive, list of such considerations.

2.1.1 Transmission Rate versus Speech Quality

It is generally accepted that for a given speech coding algorithm, a higher output speech quality can be obtained with a higher transmission or output bit rate. However, the converse may not be true as new and advanced speech coding techniques evolve. This is depicted in Figure 2.1 on page 9.

Regardless of the trends in speech coding science and technology, the speech quality requirements, laid down by regulatory bodies, such as CCITT (International Telegraph and Telephone Consultative Committee), must be met. For PSTN or systems which connect to the PSTN, a speech coder must produce speech of toll quality. Mobile radio telephony systems are less stringent as they require communications quality speech (although this may change in third-generation mobile systems). Military applications often view quality as of secondary importance pertinent to operations as when compared to bit rate reductions.
2.1 Characteristics of Speech Coding Schemes

2.1.2 Robustness to Channel Impairments

The robustness of a speech coding algorithm is the measure of the speech quality produced under different transmission channel conditions. Ideally, the input bit stream to the speech decoder should be free of errors. However, signal impairments in many speech communications applications often result in corrupted coder parameters, giving rise to degradations in the reproduced speech.

A good design of a speech coding algorithm should address the importance of error robustness in the developmental stages. For instance, the Multi-Band Excitation (MBE) vocoder and its variants (for example, the MB-LPC) can demonstrate graceful degradation as the channel worsens, whilst maintaining high intelligibility in the reproduced speech. On the other hand, the Code Excited Linear Predictive (CELP) coder and its variants do not exhibit similar tolerance to channel errors. Clearly, CELP-type coders impose a higher demand on the role of error control strategies and FEC coding techniques than MBE-type coders, the performance of which is seriously constrained by power and bandwidth.

Figure 2.1: Influence of coding rate on speech quality [13].
2.1 Characteristics of Speech Coding Schemes

2.1.3 Delay

The time displacement of a signal from its source (talker) to its sink (listener) is generally referred to as delay in speech communication systems. From a value of a few milliseconds to a few tenths of a second, delay is universally present in all communication systems. However, its effects are more pronounced in simultaneous duplex voice transmissions. Talker fatigue, echoes and other psycho-acoustic effects due to significant amounts of delay have been identified as subjectively annoying. It is, therefore, desirable to minimise delays in speech communication systems.

End-to-end delays in systems can be attributed to the speech coding algorithm and the length of the transmission path. Whilst the latter is a direct and unavoidable consequence of the distance between the users, LBR speech coders introduce an amount of delay which depends on the type of coding technique used. Delays in LBR speech coders can be further divided into that by algorithmic and processing components. Processing delays can be made insignificant with advances in technology, but algorithmic delays, due to buffering of speech samples, are the side effects of the block processing nature of speech coders by which low bit rates are achieved in the first place.

To ameliorate the effects of delay in speech communication systems, echo cancellers will have to be incorporated. It should be noted that the use of echo cancellers increases the overall system cost. Thus, the communications engineer selecting a LBR speech coder for a specified application will have to be aware of the total delay budget imposed if the use of echo cancellers is to be avoided.

2.1.4 Complexity

The complexity of a given LBR speech coding algorithm is a function of the number of computations required for the encoding and decoding processes. This is usually specified in million instructions per second (MIPS) or million floating point operations per second (MFLOPS). For implementation in real-time, the complexity requirements usually have to
match or not exceed the capabilities of the given platform, for example, a digital signal processor (DSP). Otherwise, the coding algorithm is said to be, in very general terms, too complex.

With newer and faster DSPs becoming available through advances in technology, it appears that, apart from inefficiently coded algorithms, most speech coding techniques are real-time implementable. This renders the above definition of complexity prone to misinterpretation. A more realistic alternative would be to include the power consumption of the target hardware as well. This is in line with the desire to keep mobile telephones and portables, which will all run on batteries, in operation for as long as possible between servicing. This will encourage speech coding researchers to concentrate not only on fewer-instruction algorithms, but also on simpler structures of DSPs (for example, fixed point versus floating point processors). The former aspect will enable applications to use processors with slower clock frequencies, whilst the latter will correspond to fewer number of transistors needed. Both will ultimately result in reduced power consumption. It will be undesirable if the projected UMTS were to be massively successful but uses technology that consumes a huge amount of power!

2.2 Linear Prediction of Speech

Source coding signifies the desire to encode not what, but *how* speech waveforms are being produced at a given state and time. As such, source coding requires considerably more knowledge about speech signals and their production mechanisms than in the case for waveform coding. In this section, an understanding of what source coding is, its advantages and drawbacks, and the analysis and encoding procedures that entail, is described. The discussions will concentrate on only the technique of linear prediction. The work, on which the theme of this thesis is based, involves coders which employ linear prediction.
2.2.1 Speech Production Model

In many sophisticated speech coding techniques, low coding rates have been achieved through the use of a well established principle of how speech signals are produced. Known as the speech production model, human speech results when the output of a suitable signal source (generally comprising of periodic and random noise signals) is used to excite a time-varying linear filter which has resonant properties similar to those of the human vocal tract. Equivalently, the frequency spectrum of the speech signal can be obtained by multiplying the source spectrum by the frequency characteristics of the vocal tract filter. This process is depicted in Figure 2.2.

![Source-Filter Model Diagram](Image)

Figure 2.2: The source-filter model of speech production.

The speech production model, also commonly called the source-filter model, is an oversimplification of the actual speech production process. In particular, the model assumes that the source is linearly separable from the vocal tract filter, and there is no interaction between the two. Although this is not strictly true, the successes of many speech coding techniques, based on this model, have proved that it is sufficiently accurate for voice coding.
2.2 Linear Prediction of Speech

The postulation of the source-filter theory of the production of speech leads to the concept of short-time analysis techniques fundamental in many LBR speech coding schemes. Speech is, over a long interval of time, essentially non-stationary. However, over a sufficiently short period (10 to 30 milliseconds), it can be considered as stationary. In effect, this enables the relatively slowly time-varying vocal tract filter to be parametrically represented. Amongst many other methods of representing this filter, the very important and powerful technique of linear prediction has emerged as the most extensively used technique in many speech coders.

The source-filter model gives a good description of the generation process of speech waveforms. Linear prediction only provides an approach in parameterising a part of this model, whilst making no attempt at describing how the excitation source may be generated. As speech coding evolves, numerous techniques have been suggested as to how the excitation source may be represented, leading to a large class of linear predictive coders. Two such coders, the CELP and the MB-LPC, will the subject of discussions in later sections. At the current stage, it is interesting to note that the excitation source is modelled in the time-domain in the CELP coder, and in the frequency domain in the MB-LPC coder.

2.2.2 Linear Predictive Analysis

The technique of linear prediction was known for a long time, but it was only in the early 1970s that it was shown, by Atal and Hanauer [2], to be applicable to speech. In essence, the idea behind the method is that samples of speech, $s(n)$, can be approximated as a linear combination of the past $p$ speech samples (see Figure 2.3).

$$s(n) = Gu(n) + a_1s(n - 1) + a_2s(n - 2) + \ldots + a_ps(n - p)$$

$$= Gu(n) + \sum_{k=1}^{p} a_k s(n - k)$$

(2.1)
2.2 Linear Prediction of Speech

To see how Equation 2.1 is arrived at, consider a sequence of speech samples with its (modelled) short-time vocal tract filter, \( H(z) \), as:

\[
\frac{S(z)}{U(z)} = H(z) = G \frac{1 + \sum_{i=1}^{q} b_i z^{-i}}{1 - \sum_{k=1}^{p} a_k z^{-k}}
\]

(2.2)

where \( G \) is an energy scaling factor for the excitation source, \( u(n) \).

To minimise analysis complexity, linear predictive coding (LPC) assumes that \( H(z) \) is sufficiently approximated by an all-pole model:

\[
H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}}
\]

(2.3)

Equation 2.1 is simply the difference equation of Equation 2.3. Since this is an approximation, this leads to a prediction error, \( e(n) \):

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

(2.4)

By minimising the mean squared error between the actual speech samples and the linearly predicted ones, the LPC coefficients \( (a_1 \ldots a_p) \) can be determined by solving a set of linear
2.2 Linear Prediction of Speech

equations.

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

(2.5)

Setting the partial derivatives of \( E \) with respect to each coefficient, \( a_j \), to zero, gives:

\[
\sum_{k=1}^p a_k \sum_n s(n - j)s(n - k) = \sum_n s(n)s(n - j)
\]  

(2.6)

for \( j = 1, 2, \ldots, p \). Equation 2.6 represents a set of \( p \) linear equations for the \( p \) unknowns, \( a_k \), which can be solved by matrix inversion, not a trivial task if \( p \) is large. Two different efficient methods exist for finding the solution of this system of equations, namely the autocorrelation method and the covariance method. Only the autocorrelation method (AM) will be discussed here as it is more widely used.

The limits of the summation in the expressions

\[
\sum_n s(n - j)s(n - k), \quad \sum_n s(n)s(n - j)
\]

have, so far, been left unspecified. Since speech is assumed to be stationary over a given interval of time, the range of summation is then from 0 to \( N - 1 \), where \( N \) is the frame length of the specified time period in samples. With \( s(n) \) defined as zero for \( n < 0 \), then

\[
\sum_{n=0}^{N-1} s(n - j)s(n - k) = \sum_{n=0}^{N-1} s(n)s(n + j - k)
\]  

(2.7)

Therefore, Equation 2.6 becomes:

\[
\sum_{k=1}^p a_k \sum_{n=0}^{N-1} s(n)s(n + j - k) = \sum_{n=0}^{N-1} s(n)s(n - j)
\]  

(2.8)

for \( j = 1, 2, \ldots, p \). The right hand side of Equation 2.8 is in the form of autocorrela-
2.2 Linear Prediction of Speech

tion values of the speech signal for specific time shifts. Thus if $R(k)$ is defined as the autocorrelation:

$$R(k) = \sum_{n=0}^{N-1} s(n)s(n+k)$$

then Equation 2.8 results in:

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}$$

(2.10)

This form of a matrix is known as a Toeplitz matrix and a very efficient, recursive method due to Durbin and Levinson [33] for solving this special system of equations exists.

The LPC coefficients have to be updated for every speech frame or segment in order to reflect the quasi-stationary characteristics of the speech signal. Often, the samples are windowed using a soft window function (for example, a Hamming window) of duration $N$ samples in order to reduce the prediction errors at the start and end of the segment.

2.2.3 Line Spectral Frequencies

A closer examination of the LPC prediction filter (see Figure 2.3, page 14) will show that it corresponds closely to a class of digital filters termed Infinite Impulse Response (IIR) filters. An IIR filter has the notoriety of easily becoming unstable if its coefficients are not properly chosen. With the coefficients of the filter being updated for every speech frame, care has to be taken to ensure that $H(z)$ remains stable. This is guaranteed if the poles of the LPC filter:

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

(2.11)
all remain within the unit circle in the z-domain. This is known as the minimum phase property of the LPC filter. Fortunately, the autocorrelation method of calculating the LPC coefficients always guarantees this property.

For the construction of the LPC filter at the receiving end, the LPC coefficients have to be quantised, coded and transmitted. If they (that is, \( \{a_k ; k = 1 \ldots p \} \)) are quantised directly, the minimum phase property is not easily preserved. It is thus desirable to find an indirect but equivalent form of representing the LPC coefficients which averts the problem. Line spectrum pairs (LSPs), or line spectral frequencies (LSFs, as preferred in this thesis), were introduced by Itakura [21] in 1975 as a solution to this problem. The LSF representation of the LPC coefficients exhibits a very important property: evident filter stability through the natural ordering of the LSF parameters.

The transformation of the LPC coefficients to LSPs is as follows: given a stable LPC filter of an even order, the minimum phase LPC polynomial is defined as:

\[
A(z) = \frac{1}{H(z)} = 1 + \sum_{k=1}^{p} a_k z^{-k}
\]  

(2.12)

(The minus sign in front of the summation is changed to a plus purely for convenience.)

Two artificial \((p + 1)\)th degree polynomials are set up:

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

\[
= z^{-(p+1)} \prod_{i=0}^{p+1} (\alpha_i + z)
\]  

(2.13)

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

\[
= z^{-(p+1)} \prod_{i=0}^{p+1} (\beta_i + z)
\]  

(2.14)

where \(\alpha_i, \beta_i\) are the roots of the two polynomials. It can be observed that:

\[
A(z) = \frac{P(z) + Q(z)}{2}
\]  

(2.15)

The two polynomials have been shown to possess the following properties [44]:
2.2 Linear Prediction of Speech

1. all roots of $P(z)$ and $Q(z)$ are on the unit circle,

2. the roots of $P(z)$ and $Q(z)$ are interlaced.

The second property given above is a necessary and sufficient condition for the stability of the LPC synthesis filter.

Since the roots of $P(z)$ and $Q(z)$ are of the form $e^{j\omega}$, and both polynomials have real coefficients, the roots must then occur in complex-conjugate pairs. Thus, only values of $\omega_i$ between 0 and $\pi$ need to be considered. With $-1$ and 1 being one of the roots of $P(z)$ and $Q(z)$ respectively, both polynomials each now have $p/2$ distinct $\omega_i$'s. Denoting the $p/2$ normalised frequencies of $P(z)$ by $\{\omega_i ; i = 0, 2, 4, \ldots \}$, and that of $Q(z)$ by $\{\omega_i ; i = 1, 3, 5, \ldots \}$, then the second property becomes:

$$0 < \omega_0 < \omega_1 < \ldots < \omega_{p-2} < \omega_{p-1} < \pi$$  \hspace{1cm} (2.16)

This is known as the *monotonicity criterion* of the LSPs in this thesis. The LSFs are the frequency equivalents of the LSPs:

$$\text{LSF}(i) = \frac{\omega_i}{2\pi T} = \frac{\omega_i F_s}{2\pi} \text{ (Hz)}, \hspace{0.5cm} 0 \leq i \leq p - 1$$  \hspace{1cm} (2.17)

where $F_s$ is the sampling frequency.

The representation of the LPC filter coefficients using LSFs has many distinct advantages in addition to the ones described earlier. In particular, LSFs are related to the formants (that is, peaks of the LPC spectral envelope), explicatively depicting the LPC spectral shape. Also, efficient quantisation schemes have been developed for LSFs, using vector quantisation [40], and effective error control techniques [35] found for robustness against channel errors.
2.3 The CELP Coder

In modelling the excitation source of the speech production process, various methods have been suggested and successfully implemented. Good examples of such schemes include the multi-pulse excited LPC (MPELPC) and the residual excited linear predictive (RELP) coders. These LPC-based speech coders offer a good quality of the synthesised speech at rates of 8 kb/s or more. However, at lower rates, these speech coders become inadequate in achieving the required quality. The Code-Excited Linear Predictive (CELP) coder, first introduced by Atal [1] [42], represented a major breakthrough in the coding efficiency bottleneck. The realistic implementation of the CELP algorithm, with its high computational requirements (50 to 100 MIPS) [22], was only made possible with advances in technology. Today, many speech coders operating at rates between 4 and 8 kb/s are either CELP or CELP-based.

The successes of the CELP coder and its copious variants would not have been possible without some key concepts, such as the analysis-by-synthesis (AbS) optimisation procedure. Upon briefly reviewing the operation of the CELP coder, the key concepts exploited by CELP will be described. Some detailed, but not extensive, formulations in the CELP coder will also be discussed, together with some remarks on the error sensitivities of its parameters.

2.3.1 General Description

The block diagram of a “standard” CELP coder is as shown in Figure 2.4.

The CELP algorithm operates as follows:

1. Input speech, \( s(n) \), is first partitioned into analysis frames of around 20 to 30 milliseconds, using a soft window such as a Hamming window. Speech is then analysed and coded on a frame by frame basis.

2. Linear prediction analysis is performed on the current frame to give a set of LPC
coefficients. These are used to set up a short term predictor (STP) which models the spectral envelope of the current speech samples.

3. A zero-valued excitation is then used to "clock" out the memory (also known as initial conditions) of the STP, the result of which is subtracted from the speech input to produce the reference signal. The STP is now memoryless for subsequent analysis.

4. Pitch or long term prediction (LTP) is then performed on sub-multiples of the refer-
ence, for example, 5 to 10 milliseconds. The LTP analysis produces a value for the delay (D) and associated scaling factor(s) \( \beta_i \). The LTP injects voice periodicity into the synthesised speech.

5. Determination of the secondary excitation, also known as codebook sequence, is next performed, also on sub-multiples of the frame size. The excitation signal, \( u(n) \), is selected from a codebook, usually containing random Gaussian sequences. As there are many candidate entries in the codebook, the sequence is selected through enumerating all possibilities in the codebook by generating the synthetic speech through the LTP and STP filters and obtaining that which minimises a pre-defined objective error. An associated gain factor is then computed.

6. At the decoder in the receiving end, the initial conditions of the filters are restored. Synthetic speech is then generated by filtering the scaled optimal codebook sequence through the cascaded LTP and STP filter without using perceptual weighting (see Section 2.3.2.3).

2.3.2 Supporting Concepts

The ability of the CELP coder to produce high quality speech at rates lower than 8 kb/s is due to the utilisation of several concepts borrowed from other branches in engineering, including systems identification and rate distortion theory. Consequently, CELP is able to surpass precursors (such as RELP) at the expense of a dramatic increase in computation complexity.

2.3.2.1 Analysis-by-Synthesis Optimisation

The analysis-by-synthesis (AbS) optimisation procedure is a general technique used in many other areas of estimation and identification of signals and systems. Due to its sheer complexity, it was not used extensively in speech coding until the MPLPC coder was
2.3 The CELP Coder

described [3]. Since then, AbS is used to enhance the analysis and coding procedures in many low bit rate speech coders, particularly CELP.

![Block diagram of the analysis-by-synthesis optimisation loop](image)

Figure 2.5: Block diagram of the analysis-by-synthesis optimisation loop

The basic idea behind AbS is as follows: first, the reference signal to be parameterised is assumed to be observable in some form, for example, time or frequency domain. With a theoretical form of the signal production model, its parameters can then be varied and permuted, in a trial and error fashion, to generate different estimations of the reference signal. A cost function is then applied to the difference between the reference and the estimated signals. The set of the model parameters which minimises the cost function is then taken as that which best generates the reference signal. As can be realised at this stage, this "closed loop" approach is the main source of computation complexity in CELP. The outline of the CELP operation discussed in Section 2.3.1 is actually a sub-optimal parameter-by-parameter implementation of the AbS concept.

2.3.2.2 Vector Quantisation

One of the main reasons why LPC-based speech coders, such as RELP and MPLPC, begin to deteriorate in terms of synthesised speech quality at lower bit rates is that the scalar quantisation (SQ) of the parameters becomes coarse and therefore inaccurate. CELP
2.3 The CELP Coder

overcame this particular difficulty by quantising the secondary excitation on a vector basis. Based on Shannon’s rate distortion theory, vector quantisation (VQ) offers high capacity compression with a high degree of fidelity in reproduction. Many speech coding schemes now use VQ as a means of achieving even lower bit transmission rates.

VQ [14, 31] aims to quantise parameters as a vector instead of individual scalars. To see how VQ is able to provide high coding efficiency, assume that a vector, \( \mathbf{v} \), consists of \( N \) real-valued, continuous or discrete variables:

\[
\mathbf{v} = [x_1 \ x_2 \ \ldots \ x_N]
\]

A VQ procedure would then map \( \mathbf{v} \) into an entry, \( y_i \), in a reproduction set codebook, \( \mathbf{Y} \):

\[
\mathbf{Y} = [y_1 \ y_2 \ \ldots \ y_C]
\]

where \( C \) denotes the number of vectors in \( \mathbf{Y} \), and the components, \( \{y_{ij} \ ; i = 1 \ldots C, j = 1 \ldots N\} \), are real-valued continuous or discrete values. Figure 2.6 is an example of how a two-dimensional space can be partitioned for this purpose.

Thus to recall what the quantised version of \( \mathbf{v} \) is (that is \( y_i \)), one would only need the index to the codebook entry. The number of bits needed to code this index (commonly termed as the rate of the vector quantiser) is:

\[
R = \log_2 C \ \text{(bits per vector)}
\]

and

\[
r = \frac{R}{N} \ \text{(bits per sample)}
\]

Hence, if a sufficiently small codebook is used, with no significant distortion due to quantisation errors observed, then fractional bits per sample coding would have been achieved.
2.3 The CELP Coder

Figure 2.6: A two-dimensional visualisation of vector quantisation.

An understanding of the construction of the VQ codebook, \( Y \), and its required size, \( C \), requires considerable treatment of the subject, and is well provided in many texts on speech coding [41].

2.3.2.3 Perceptual Weighting

The CELP algorithm (Figure 2.4, page 20) in an AbS optimisation process derives its optimal codebook excitation through the minimisation of the error between the reference signal, \( s(n) \), and the synthesised signal, \( \hat{s}(n) \). A commonly used cost function would be the squared error:

\[
E = \sum_{n=0}^{N-1} e^2(n) = \sum_{n=0}^{N-1} [s(n) - \hat{s}(n)]^2
\]
This error cost function therefore emphasises on matching the reference signal in the time domain.

It has been known that an important aspect of the human perception of speech is the phenomenon of masking [39], in which the threshold of noise can be increased in the presence of high energy signals. Thus, the errors in the formant frequency ranges can be de-emphasised (that is, higher quantisation noise can be tolerated) since they are perceptually less important than the frequencies in the "valleys" of the LPC spectral envelope.

A suitable weighting filter [42] is of the form:

\[
W(z) = \frac{A(z)}{A(z/\gamma)} = \frac{1 - \sum_{i=1}^{p} a_i z^{-i}}{1 - \sum_{i=1}^{p} a_i \gamma^i z^{-i}}
\] (2.19)

The weighting filter has an effect of broadening the frequency regions corresponding to the formants (see Figure 2.7). Suitable values of \( \gamma \) lie between 0.8 and 0.99, which can be fine tuned by listening tests.

In CELP, the weighting filter can be incorporated into the STP to form a modified LPC synthesis filter:

\[
\frac{1}{A'(z)} = \frac{1}{A(z)} W(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i \gamma^i z^{-i}}
\] (2.20)

### 2.3.3 Short-Term Predictor

The functionality of the STP in CELP is basically that of LPC synthesis filtering. In the spectral domain, the STP represents the smoothed shape of the speech spectrum (see Figure 2.7, page 26) using a compact set of values. The parameters of the STP are computed
2.3 The CELP Coder

using, usually, the autocorrelation method. As previously discussed in Section 2.2.2, the actual quantisation and coding of the STP coefficients is commonly performed through a transformed set of LPC parameters, the line spectral frequencies (LSFs). A LPC order of ten provides sufficient transparency in coding the spectral envelope of speech.

There are still research interests in the quantisation of LSFs using the minimum capacity (that is, the number of bits) [32,45] and several other related issues. The prime interest, in this thesis, concerning the STP is effective error control strategies for the LSFs through exploitation of the monotonicity criterion and the correlation properties of LSFs.

Figure 2.7: Original and weighted smoothed speech spectrum \((\gamma = 0.8)\).
2.3.4 Long-Term Predictor

The source-filter model assumes a suitable source for the excitation signal of the vocal tract filter (modelled by the STP). In CELP, this excitation source is obtained via a suitable mix of voiced and unvoiced excitations. The long-term predictor (LTP) in CELP aims to synthesise the voiced part of speech. In the frequency domain, the LTP contributes to the periodic “comb-structure” of the speech signal, with its delay value (otherwise termed as pitch period) determining the spacing between the harmonics. It has been known that a good degree of accuracy is needed in the computation of the delay (hence, pitch) and this makes the LTP analysis/synthesis processes in CELP a major factor in high speech quality reproduction.

Associated with the pitch period for the LTP are scaling factors, termed as LTP coefficients or pitch gains ($\beta_i$), whose number depends on the order of the LTP. In general, the pitch inverse filter is given by:

$$P(z) = 1 - \sum_{i=I}^{J} \beta_i z^{-(D+i)}$$  \hspace{1cm} (2.21)

where $I$ and $J$ are the tap limits, and $D$ is the LTP delay. It should be noted that the LTP in CELP synthesises the long term correlative components in speech, irrespective of whether the signal is voiced or not. As such, the pitch corresponding to the delay value in the LTP may not necessary indicate the presence of voiced speech at all.

The determination of the LTP parameters, $D$ and $\beta_i$, follows a procedure that closely resembles that in the LPC analysis – that of error energy minimisation. Since CELP employs a “closed loop” approach due to the AbS optimisation procedure, the analysis of the desired LTP parameters then involves minimising the error between the original and the processed speech. This process is depicted in Figure 2.8 (page 28).
2.3 The CELP Coder

From Figure 2.8, the synthesised signal is given by:

\[
s(n) = \sum_{i=I}^{J} \beta_i \sum_{k=0}^{n} h(k)r(n - k - D - i)
\]

(2.22)

where \(h(k)\) is the impulse response of the STP. Then, defining the LTP error signal energy, \(E_{LTp}\) as:

\[
E_{LTp} = \sum_{n=0}^{L-1} [s(n) - \hat{s}(n)]^2
\]

(2.23)

where \(L\) is the LTP update length. Then,

\[
\frac{\partial E_{LTp}}{\partial \beta_j} = -2 \left[ \sum_{n=0}^{L-1} s(n) - \sum_{i=I}^{J} \beta_i \sum_{k=0}^{n} h(k)r(n - k - D - j) \right] \\
\times \left( \sum_{k=0}^{n} h(k)r(n - k - D - j) \right) \\
= 0, \quad -I \leq j \leq J
\]

(2.24)

Denoting \(Z_j(n)\) as:

\[
Z_j(n) = \sum_{k=0}^{n} h(k)r(n - k - D - j)
\]

(2.25)
Equation 2.24 becomes:

\[
\sum_{n=0}^{L-1} s(n)Z_j(n) - \sum_{i=-I}^{J} \beta_i \sum_{n=0}^{L-1} Z_i(n)Z_j(n) = 0, \quad -I \leq j \leq I
\]  

(2.26)

For I=J=1, Equation 2.26 reduces to:

\[
\begin{bmatrix}
\beta_{-1} \\
\beta_0 \\
\beta_1 \\
\end{bmatrix} = 
\begin{bmatrix}
A(-1,-1) & A(0,-1) & A(1,-1) \\
A(-1,0) & A(0,0) & A(1,0) \\
A(-1,1) & A(0,1) & A(1,1)
\end{bmatrix}^{-1} 
\begin{bmatrix}
B(-1) \\
B(0) \\
B(1)
\end{bmatrix}
\]  

(2.27)

where,

\[
A(i,j) = \sum_{n=0}^{L-1} Z_i(n)Z_j(n)
\]  

(2.28)

\[
B(i) = \sum_{n=0}^{L-1} s(n)Z_i(n)
\]  

(2.29)

For a single tap LTP (that is, I=J=0),

\[
\beta_0 = \frac{B(0)}{A(0,0)}
\]  

(2.30)

It can be observed that Equations 2.28 and 2.29 are the ubiquitous correlation operations (autocorrelation for Equation 2.28, and cross-correlation for Equation 2.29) as was seen in the LPC formulations (see Section 2.2.2).

Upon determining the value of the LTP gain coefficients, the optimal value of D can then be found by using Equation 2.23 and to minimise \(E_{LTP}\). It must be mentioned that the preceding formulations had assumed that the LTP delay, D, is always greater or equal to L, the LTP update length. For situations otherwise, recursion within the update sub-block results, and a non-linear solution of the LTP coefficients is involved. A discussion of this "problem" is given in [29].
2.3.5 Codebook Excitation

By superposition, after having (totally) removed all periodicity in the reference waveform, the source-filter model (see Figure 2.2, page 12) would only have the random unvoiced constituent of speech left. In CELP, this component is obtained from a codebook composed of (usually) Gaussian samples. However, since the LTP is located after the codebook excitation sub-block, the codebook vectors have to, in addition, provide initialisation samples in the LTP memory.

![Figure 2.9: Codebook vector search in CELP.](image)

The AbS loop optimisation search for the codebook vector is illustrated in Figure 2.9. As with the formulations earlier discussed, the objective is to minimise the error signal energy between $s(n)$ and $s_k(n)$.

$$E_{\text{codebook}} = \sum_{n=0}^{L-1} [s(n) - s_k(n)]^2 \quad (2.31)$$

With $s_k(n)$ given as:

$$s_k(n) = G r_k(n) \otimes h(n) \quad (2.32)$$

where $G$ is the codebook vector scaling factor or gain, $h(k)$ is the impulse response of the STP, $\otimes$ indicates the convolutional sum operation, and $k$ indicates the codebook vector index. Minimising $E_{\text{codebook}}$, that is:

$$\frac{\partial E_{\text{codebook}}}{\partial G} = 0 \quad (2.33)$$
2.4 The Multi-Band LPC Vocoder

gives:

\[ G = \frac{\sum_{n=0}^{L-1} s(n) [r_k(n) \otimes h(n)]}{\sum_{n=0}^{L-1} [r_k(n) \otimes h(n)]^2} \]  

Substituting into Equation 2.31:

\[ \min \{E_{\text{codebook}}\} = \sum_{n=0}^{L-1} s^2(n) - \frac{\left[ \sum_{n=0}^{L-1} s(n) [h(n) \otimes r_k(n)] \right]^2}{\sum_{n=0}^{L-1} [h(n) \otimes r_k(n)]^2}, \quad k = 1 \ldots C \]  

Thus, minimising \( E_{\text{codebook}} \) is equivalent to maximising the last term in the RHS of Equation 2.35. These calculations are the most computationally intensive in CELP since \( C \) is usually large (> 1000). This has spawned much research activity around structuring the codebook for efficient excitation searches.

As depicted in Figure 2.4 (page 20), the CELP decoding process involves scaling the codebook excitation vector with its associated gain, then filtering it through the cascaded LTP and STP synthesis blocks. In effect, the codebook gain, \( G \), represents the energy of the speech signal. Together with the feedback implied by use of memory in the LTP and STP, corruption of \( G \) is subjectively very annoying, and seriously limits CELP’s operation in noisy transmission channels. On the contrary, corruption of the codebook index is known to be less subjectively noticeable.

2.4 The Multi-Band LPC Vocoder

An alternative approach to analysing the excitation signal is to consider it in the spectral domain. Voiced speech waveforms have a characteristic periodic structure due to the harmonics generated by a fundamental frequency, commonly termed as pitch. If this structure were to be matched and removed by a speech coding model, then the remaining spectral components would consist mainly of noise-like energy. This operation can be considered as a frequency-domain analogy to the time-domain filtering processes in CELP.
This technique of modelling the excitation signal is exactly that which is employed in the Multi-Band LPC (MB-LPC) vocoder. Based on the Multi-Band Excitation model [16], the MB-LPC coder offered an option to LBR speech coding at below 4 kb/s. The version of the MB-LPC coder used in this thesis has a capability of producing natural, high quality speech at as low as 2.4 kb/s [52].

The appeal of using the multi-band type of coders is apparent: the INMARSAT Standard M [19] and the APCO Standard [20] are all based on the MBE coder as it has the capacity of producing high quality speech with an inherently high degree of robustness to channel errors. As will be demonstrated in later chapters, the MB-LPC also possesses such desirable features, an understanding of which involves a closer examination of the coding technique. The following sections aim to provide such a description, together with some brief discussions on the parameters' error sensitivities.

2.4.1 General Description

The MB-LPC vocoder’s block operation is depicted in Figure 2.10, and can summarised as follows:

1. Input speech is first divided into 20-milliseconds frames. A high pass filter is used to remove DC and very low frequency components from the speech. A soft window function, such as a Hamming or Kaiser window, is applied to the speech samples in the current frame.

2. An estimation of the pitch in the current frame is then performed. The speech frame is first low pass filtered before an autocorrelation function is used to produce a value of the estimate of the pitch, $P_f$, with a coarse resolution (half sample). Pitch tracking is used to ensure smooth transitions of the initial pitch estimate from the previous frame and to the next frame.
2.4 The Multi-Band LPC Vocoder

Encoder

Input Speech

High Pass Filter

Windowing Function

Compute Initial Pitch

LPC Analysis

Inverse Filter Response

DFT

Candidate Pitches

Construct Synthetic Spectrum

Minimise Error

Voiced Component

V/UV

Unvoiced Component

LPC Synthesis Filter

Post Filter

Speech

Decoder

Voiced Component

LPC Synthesis Filter

V/UV

Unvoiced Component

Figure 2.10: Block diagram of MB-LPC vocoder.
3. The speech frame is then LPC-analysed. The coefficients are then used to form an inverse filter, which is used to remove the spectral envelope in speech. This operation results in a “flattened” excitation spectrum, which is then further characterised to derive the other parameters.

4. Using the initial pitch estimate, $P_I$, a number of refined pitch estimates are then generated (quarter sample resolution). Based on these candidate pitch values, $P_R$, a synthetic speech spectrum is then constructed and compared with the reference spectrum for spectral amplitude matching. The value of $P_R$ which results in the least error in the least squares sense is then considered to be the desired refined pitch estimate. Note that this is an analysis-by-synthesis search for the desired pitch value.

5. With the final pitch value obtained, the corresponding synthetic spectrum is then divided into bands, each consisting of three harmonics. The voicing decision for each band is then computed based on the closeness of fit (in terms of spectral distortion) of each band with the reference spectrum. Hence, a high degree of matching would indicate periodic energy in the given band, whilst the converse would mean that the band can be approximated using noise-like sources.

6. The scaling factor for the excitation spectrum is then approximated by a root mean square (RMS) formulation of the excitation spectral amplitudes taken at harmonic intervals of the final pitch estimate, $P_R$. This factor represents the energy of the excitation spectrum.

7. The transformed LPC parameters (LSFs), final pitch value $P_R$, the voicing decisions (V-UV) and the excitation scaling factor are then quantised and coded for transmission.

8. At the receiving end, the decoded pitch and the voicing decisions are used to synthesise the excitation signal with the appropriate harmonics. This is performed in the time domain.
9. Using the voicing decisions, unvoiced bands are synthesised in the frequency domain prior to a discrete Fourier Transform (DFT). The voiced and unvoiced signals are then scaled by the energy factor before being summed up.

10. The combined voiced and unvoiced excitation signal is then passed through a LPC synthesis filter to produce speech. Post filtering of the synthesised speech may be incorporated to enhance the quality of the reproduced speech.

As can be inferred from the above, the MB-LPC coder produces a very compact set of parameters for the reproduction of speech. In particular, voiced energy in speech is completely modelled using only the pitch and the voicing decisions, whilst the unvoiced energy can be completely derived. Consequently, the MB-LPC coder is able to operate at rates below 4 kb/s with a relatively high degree of ease.

2.4.2 Spectral Envelope Estimation

The MB-LPC vocoder concerns itself primarily with the modelling of the excitation signal in the speech production model. Hence, the excitation spectrum must first be "normalised" such that further analyses performed later are not unnecessarily biased by the spectral tilt. Equivalently, the operation seeks to "flatten" the speech spectrum.

Since the spectral envelope of speech can be modelled using linear prediction, a relatively simple way of flattening the speech spectrum would be to perform LPC inverse filtering. Equally effective, the excitation spectrum can be obtained by multiplying the speech spectrum with the frequency response of the LPC inverse filter. This is illustrated in Figure 2.11 (page 36).

The LPC coefficients needed to set up the LPC inverse filter can be computed using the Autocorrelation Method, as was discussed in Section 2.2.2. A tenth-order LPC analysis is sufficient for coding transparency, with the actual quantisation and transmission performed through the use of LSFs (see Section 2.2.3).
2.4 The Multi-Band LPC Vocoder

2.4.3 Pitch Determination

The periodic "comb" structure of speech signals, particularly in high-energy voiced segments, consists of "pulses" in the spectrum, with local peaks harmonically related to the fundamental frequency, more commonly known as pitch. This can be observed from Figure 2.11. It has been determined that high accuracy in the estimation of this pitch is vital for a good match to the fine structure of the original speech spectrum containing periodic energy [16].

Since traditional pitch detection techniques do not provide sufficient accuracy in the pitch estimation process, the analysis-by-synthesis (AbS) optimisation procedure is adopted
here. Obviously, this is computationally prohibitive and thus, a two-part procedure is employed in both the MBE and the MB-LPC vocoders: a search for an initial pitch \( P_I \), and then a local optimisation of \( P_I \) using a higher resolution to give \( P_R \). This procedure [16,51] enables the pitch estimation algorithm to be real-time implementable whilst achieving the desired accuracy.

### 2.4.3.1 Initial Pitch Estimation

In the first part, the 20-ms speech frame (160 samples) is first low pass filtered and then centred on a 281-point Hamming window, \( w_I(n) \). For a candidate pitch value, \( P \) (in samples), the following (unbiased) error criterion is evaluated:

\[
E(P) = \sum_{n=-140}^{140} [s(n)w_I(n)]^2 - P \sum_{k=-140}^{140} \Phi(kP)
\]  

(2.36)

where

\[
\Phi_P = \begin{cases} 
\sum_{n=-140}^{140} s(n)w_I^2(n)s(n-p)w_I^2(n-p) & \text{for integer } p \\
(1 + \lfloor p \rfloor - p)\Phi(\lfloor p \rfloor) + (p - \lfloor p \rfloor)\Phi(\lfloor p \rfloor + 1) & \text{for non-integer } p 
\end{cases}
\]  

(2.37)

with \( \lfloor x \rfloor \) is equal to the largest integer lesser than or equal to \( x \). \( P \) is constrained to be in the range:

\[
P \in \{21, 21.5, 22, 22.5, \ldots, 121.5, 122\}
\]  

(2.38)

The theoretical justifications for using the above form of autocorrelation for the computation of \( \Phi(p) \), rather than the usual one, are detailed in the references [15,16]. In actual implementations, pitch continuity from previous frames and to future frames is desirable since an abrupt deviation in the pitch estimate can cause degradation in the speech quality. To satisfy this requirement, *pitch tracking* is used to restrict the range of the candidate pitch values as follows:

\[
0.8P(n) \leq P(n+1) \leq 1.2P(n)
\]  

(2.39)
where \( n \) indicates the speech frame index.

To account for the influence on the range of \( P_l \) due to "look-back" pitch tracking with previous frames, a \textit{cumulative error criterion} is employed instead of simply \( E(P) \):

\[
CE(P_l) = E_j(P_l) + E_{j-1}(\hat{P}_{j-1}) + E_{j-2}(\hat{P}_{j-2})
\]

(2.40)

where \( \hat{P}_{j-i} \) denotes the value of the pitch estimate used in the \((j - i)\)th frame. A similar (but not identical) formulation is used for the "look-ahead" pitch tracking consideration. The value of \( P_l \) which finally minimises the cumulative error criterion is then designated as the best pitch estimate, \( \hat{P}_l \).

A complete description of the algorithm is beyond the intended scope of this section. Details of the algorithm are provided in the INMARSAT Standard M manual [19]. It is interesting to note, at the moment, that the pitch tracking which spans over 2 frames, is a very useful source criterion as far as error control is concerned. The integrity of the value of the pitch, when transmitted over noisy channels, is vital since a completely different periodic spectrum is subjectively noticeable.

\subsection{2.4.3.2 Pitch Refinement}

In the pitch refinement procedure, the frame is now centred on a 221-point Kaiser window \((\omega_R)\) before a DFT operation. \( \hat{P}_l \) is now intended to be improved to a value, \( P_R \), with a quarter sample resolution in the range:

\[
\hat{P}_l - \frac{9}{8}, \hat{P}_l - \frac{7}{8}, \ldots, \hat{P}_l + \frac{7}{8}, \hat{P}_l + \frac{9}{8}
\]

With a candidate pitch, \( P_R \), the frequency equivalent, \( \omega_0 = \frac{2\pi}{P_R} \), is calculated and then used to construct a synthetic spectrum, \( S_\omega(n, \omega_0) \). The latter is then compared with the reference spectrum, \( S_\omega(n) \). The number of spectral harmonics needed to synthesise
where $F_s$ is the sampling rate. With the number of harmonics known, the spectral amplitudes of each of the harmonics are then computed:

$$A_l(\omega_0) = \frac{\sum_m S_w(m)W_R(m - l\omega_0)}{\sum_m W_R^2(m - l\omega_0)} \quad \text{for} \quad 1 \leq l \leq L$$

where $W_R(n)$ is the Fourier Transform of the pitch refinement window, $w_R(n)$. The summation range in Equation 2.42 is chosen to be over the range where $W_R(n)$ peaks (see Figure 2.12). In essence, the synthetic spectrum is composed of the sum of the “impulse” frequency responses, using $W_R(n)$, centred on each harmonic, \{lw_0; 1 \leq l \leq L\}, with an amplitude $A_l(\omega_0)$.

With a candidate spectrum constructed for a given $P_R$, the spectral matching error,
2.4 The Multi-Band LPC Vocoder

\( E_R(\omega_0) \) is computed:

\[
E_R(\omega_0) = \sum_n |S_w(n) - S_w(n, \omega_0)|^2
\]

(2.43)

where the summation limits are selected to cover the frequency range from 1.5 to 4 kHz (for a 8 kHz sampling rate). The value of \( P_R \) which minimises \( E_R(\omega_0) \) is then regarded as the best pitch estimate, \( \hat{P}_R \). This value is then used to further determine the voicing decisions (V-UV) and the energy (\( \sigma \)) parameters.

2.4.4 Voicing Determination

The process of discriminating the frequency regions dominated by periodic energy from the regions consisting mainly of noise-like energy is termed voicing determination. The resulting parameters, the voiced/unvoiced (V-UV) decisions, are binary in nature and serve to indicate the degree of closeness of a harmonic or group of harmonics (band) with respect to the excitation spectrum, \( S_w(n) \). Since the V-UV decisions only characterise local spectral regions, the spectrum is first divided into a number of bands, \( \hat{K} \):

\[
\hat{K} = \begin{cases} \left\lceil \frac{\hat{L} + 2}{3} \right\rceil & \text{if } \hat{L} \leq 36, \\ 12 & \text{otherwise} \end{cases}
\]

(2.44)

where \( \hat{L} \) is the number of harmonics as computed in Equation 2.41. Each partial spectrum containing the three harmonics within band \( k \), \( 1 \leq k \leq \hat{K} \), is then used to compute a voicing measure, \( D_k \):

\[
D_k = \frac{\sum_m |S_w(m) - S_w(m, \omega_0)|^2}{\sum_m |S_w(m)|^2}
\]

(2.45)

Basically, the voicing measures \( \{D_k ; 1 \leq k \leq \hat{K}\} \), reflect the lack of periodic energy in band \( k \): if \( D_k \) is less than a given threshold, then the harmonics within \( D_k \) are declared voiced. Otherwise, the same band is considered to be best approximated using a noise-like energy source. The threshold is an energy-dependent function which includes consideration of the V-UV decisions from the previous frame and some heuristics.
Although the voicing measure, $D_k$, is not biased by the value of $k$, human speech energy is known to be concentrated at the lower frequency regions. Thus, the V-UV bits for these bands are subjectively more important as far as accuracy and channel conditions are concerned.

### 2.4.5 Excitation Gain

The final procedure in decomposing a frame of speech samples into a set of MB-LPC model parameters is that of computing the scaling factor or gain of the excitation spectrum. This energy factor, $\sigma$, effectively reflects the "loudness" or signal variation of speech. With the pitch value, $\hat{\theta}_R$, determined, the excitation gain is computed as the mean squared formulation of the harmonic excitation amplitudes:

$$
\sigma = \frac{1}{\hat{L}} \sum_{l=1}^{\hat{L}} [A_l(\omega_0)]^2
$$

(2.46)

where $A_l(\omega_0)$ is calculated as in Equation 2.42, and $\hat{L}$ is given by Equation 2.41. Although the excitation gain is used in an equivalent manner to the scaling factors (codebook and pitch) in CELP, it is much less sensitive to quantisation errors or channel corruptions since no feedback is involved in the synthesis process.

### 2.4.6 Speech Synthesis

In CELP, the process by which speech is synthesised from the model parameters at the receiving end is basically a "mirrored" setup of the encoding procedure. However, speech is recovered in the MB-LPC and MBE coders in a different fashion. Additional care has to be taken since, unlike CELP, the current speech samples are constructed without reliance on synthesised samples from the previous frame. In particular, phase continuity has to be explicitly ensured at frame boundaries, which is conveniently provided through the use of memory in CELP.
To do this, the synthesis of the voiced components of speech is performed in the time domain and involves some phase and amplitude interpolations. This is in contrast to the analysis procedure since the latter is implemented spectrally. This synthesis process is depicted in Figure 2.13. The exact procedure in which voiced speech is synthesised is discussed in detail in [19, 51].

The unvoiced speech component is synthesised in the time domain as well. The noise spectrum is then weighted using the excitation gain and the V-UV decisions, since voiced spectral bands are completely recovered through the voiced speech synthesis procedure above. To avoid unwanted spectral components that could be generated by amplitude discontinuities in the unvoiced speech samples, an overlap-add method is used (see Figure 2.14).

The voiced and unvoiced speech components of the excitation signal (in time domain) is then LPC-synthesis filtered to re-introduce the spectral envelope. The synthesised speech signal is then post-filtered to enhance its subjective quality.

2.5 Remarks

In this chapter, an overview of some characteristics of low bit rate speech coding techniques, their impact on available resources and the speech quality provided have been discussed. The theory of linear prediction was presented and its relationship to the speech production
process and modern speech coding techniques was outlined. Two LPC-based coders, the CELP and the MB-LPC, were examined in terms of their analysis and coding techniques. These provided some insight into their resilience (or lack thereof) to induced errors at the decoder, and the corresponding behaviour that would result. These allow an estimation of the degree of error protection and/or control strategies needed for deployment in real voice communication systems.

Figure 2.14: Synthesis of unvoiced component of speech.
Chapter Three

Error Control In Speech Coders

The primary advantage which contemporary digital speech coders, such as CELP, provide is the high perceptual quality of the reproduced speech at low bit rates. Consequently, the requirements of both bandwidth and transmission power in systems using these coders can be reduced. These measures trade off an increase in the capacity of the transmission system for a higher implementation complexity. In effect, the demand on the "external" resources has been transferred to the realisation platform (for example, faster DSPs), a pre-requisite assumed satisfied by rapid technological advancements.

The reduced requirements on bandwidth and power have improved many important aspects (such as capacity) in digital communications at the expense of the transmission conditions in the radio channel. In traditional communication systems, the reliability of the received information stream can often be improved by ensuring a sufficiently high signal-to-noise ratio (SNR), or by deploying error control strategies such as error correction [12] or re-transmission schemes. A high SNR implies a high level of transmitted power, whilst the use of error control entails some degree of bandwidth expansion (assuming a constant information rate). In mobile communication systems, these resources are placed at premium, thus leaving the received information stream susceptible to errors. For LBR speech coders, these appear as errors in the model parameters, resulting in decreased or even unacceptable speech quality.

To ensure an acceptable end-to-end quality of voice service, error control strategies
will have to be incorporated in LBR speech coders since power is limited in portable units. To complicate matters, such strategies require overhead in the system’s spectral resource, and thus will have to be carefully designed. It has been recognised that an integrated approach [4,34] is necessary for an optimal and efficient speech and channel codec, resulting in the synergy of speech and channel coding techniques. These techniques are the subjects of discussion in this chapter. First, a review of the error control problem and its difficulties pertaining to mobile radio communications is presented. Next, classical forward error correction (FEC) coding is described in limited detail. The role of FEC is then examined in the context of the error protection requirements of the speech coder parameters. In addition, the treatment of some error control/concealment methods developed for LBR speech coders is discussed. The chapter is then concluded with some remarks on the performance limitations of the techniques considered in this chapter and how advanced strategies can be developed to overcome these short-comings.

3.1 The Quality-Robustness Dilemma

In digital speech communication systems, the quality of voice service to be provided corresponds to a transmission bit rate at which the speech coder should operate. For mobile radio applications, the reliable transmission/reception of the bit stream must also be catered for. Thus, a speech coding application with an allocated bit rate is expected to provide both the maximum speech quality and channel robustness possible. For example, in GSM, the output gross bit rate is specified at 22.8 kb/s, which the implementation of the combined speech and channel codec (coder-decoder) must not exceed. High speech quality can be ensured by operating the speech coder at high source rates, as depicted previously in Figure 2.1 (page 9). However, for integrity of the coder’s bit stream in noisy channels, the output gross rate must accommodate some FEC redundancy, and this reduces the maximum possible source coding rate. These are conflicting interests which a practical speech coding implementation has to balance.
The next question then arises: what amount of FEC redundancy is required in order to counteract errors, whilst ensuring an adequate rate for the source coder to maintain the speech quality? A general rule of thumb would be to allow, for block FEC redundancy, a number at least equal to twice the number of erroneous bits (or symbols) expected. This gives rise to contradiction since, for the best performance of an error-protected speech coding scheme, a substantial fraction of the codec output stream would consist of FEC check bits (symbols). This is unfortunate since only FEC schemes are viable in real-time operations such as speech coding.

To alleviate this problem, a compromise in the error protection needed has to be made. This is accomplished by designing the FEC code to counteract a certain degree of error corruption. To make up for the apparent loss in the FEC capability during “unusual” channel conditions – attributed to system constraints, co-channel interference, multiple signal paths and fading effects – some other methods of ensuring an acceptable level of quality are used. These schemes are generally referred to as zero-redundancy techniques, so termed since they provide some degree of error protection without the use of redundancy. As a result, the amount of FEC in a speech and channel coding implementation can be reduced whilst still providing an acceptable subjective quality.

Zero-redundancy techniques rely on what is termed as source criteria in this thesis. Specifically, source criteria are mathematical or statistical constraints on the coder parameters and characteristics, which can indicate the quality of the synthesised speech with a good degree of certainty. The inclusion of source criteria in error control represents an integral approach in optimising the performance of the speech and channel codec. This synergy of source and channel coding will help solve the quality-against-robustness issue,
3.2 Forward Error Correction Coding

The problem of reliable communications in digital speech systems, as in many other transmission systems, can be abstracted and classified under the theory of communications. In his classic paper, Shannon [43] stated that: “The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.” A branch of applied mathematics, called information and/or coding theory thus emerged. This section is devoted to the discussion and implementation of the techniques involved in coding theory.

To illustrate the central idea in coding (or channel coding) theory, two mathematical models, the binary symmetric source (or just source), and binary symmetric channel (BSC) need to be defined. The source is capable of producing one of the two symbols (or just bits since they are binary), “0” and “1” at a rate $S$ bits per unit time. The BSC is capable of transferring $C$ bits per unit time between two points. The BSC is deemed as being not completely reliable, and thus has a probability, $p$, $0 \leq p \leq \frac{1}{2}$, that the output bit will not be the same as the input bit. $p$ is often known as the raw bit error probability or sometimes
as channel bit error rate (BER).

\[
\text{Prob(bit in error)} = \text{Prob}(1|0) = \text{Prob}(0|1) = p
\]  (3.1)

\[
\begin{align*}
0 & \quad 1-p \\
p & \quad p \\
1-p & \quad 0 \\
1 & \quad 1-p
\end{align*}
\]

Figure 3.3: Binary symmetric channel with bit error rate \( p \).

Obviously, the condition, \( C \geq S \) must be satisfied. Suppose \( C = 3S \), which means that the channel can deliver three times the number of bits the source can produce. Thus, the source output can be encoded by repeating each bit three times before transmission. At the information sink, three versions of each source bit will be received, but which may not be identical due to the channel "noise" corrupting each version. To decode a received bit, the best strategy here would be to take a majority vote of the three corrupted versions of the bit. Thus, if \( P_{\text{ICD}} \) denotes the probability of received bit still in error after decoding, then:

\[
P_{\text{ICD}} = \text{Prob}(2 \text{ bits in error}) + \text{Prob}(3 \text{ bits in error}) \\
= \binom{3}{2} p^2(1-p) + \binom{3}{3} p^3 \\
= 3p^2 - 2p^3
\]  (3.2)

If \( p < \frac{1}{2} \), then \( P_{\text{ICD}} \) will be less than the channel BER. If \( p \) is small, then the above coding scheme will have improved the channel's reliability dramatically. This demonstrates the essence of channel coding objective: to improve a given transmission channel's reliability through suitable coding of the source information. Shannon's work had shown that a given channel can convey an information message, with a rate not exceeding the channel
3.2 Forward Error Correction Coding

capacity, at error rates which can be arbitrarily diminished through the use of coding. **Forward error correction** (FEC) coding is concerned with the use of suitable coding schemes which attempt to achieve Shannon's predicted performance.

FEC codes can be broadly divided into two fundamentally different types: block and tree codes. In block coding, a sequence of information symbols is first segmented into \( k \)-symbol blocks. Each block is treated as a separate entity, and an encoding operation associates it with an \( n \)-tuple sequence of channel symbols, where \( n > k \). The resulting codeword now possesses properties which can resist a designed degree of channel corruption.

In tree codes, the information sequence is processed continuously and is then associated with a code sequence containing more symbols. The encoding operation involves breaking the information input into \( k_0 \)-symbol blocks and then, on the basis of each and preceding blocks, emitting an \( n_0 \)-symbol section of the code sequence.

As the descriptions imply, both block and tree codes themselves consist of a large number of different codes. Thus, the discussion of FEC codes in this thesis will be restricted to BCH and convolutional codes. In both of these two classes, a common measure of their system requirements is the code rate, \( R \):

\[
R = \frac{k}{n} \quad \text{for block codes}
\]
\[
= \frac{k_0}{n_0} \quad \text{for convolutional codes}
\]

**3.2.1 BCH Codes**

A large proportion of important random error correcting (REC) block codes are categorised as BCH (Bose, Chaudhuri and Hocquenghem) codes. These codes allow a varying degree of error correction capabilities, including the Hamming codes and the Golay (23,12) code. BCH codes are based on the theory of finite field (also known as Galois field) arithmetic, which itself is a sub-class of a branch of mathematics known as group theory. Not surpris-
ing, several texts on error control coding have termed BCH codes as group codes.

The theory and algebraic structure of BCH codes is very well developed and enriched, as can be seen from the various approaches in the encoding and decoding procedures possible. The following discussions, however, will concentrate on common aspects of these material. A full presentation of BCH codes would be prohibitively long, and is thus left to the relevant texts (see Bibliography, page 167).

### 3.2.1.1 Galois Field Basics

A Galois field, \( \text{GF}(q) \), consists of \( q \) different values with some defined arithmetic operations \( (q \) being a prime number or an integer power of a prime number). These operations, when performed on two (or more) values in the field, always produce a result which itself is a member of the field. This is known as the enclosure property. The properties of a Galois field are:

1. There are two defined operations: addition \((+\)\) and multiplication \((\times\)\).
2. If \( a, b \) are elements in \( \text{GF}(q) \), then \( a \times b \) is also an element in \( \text{GF}(q) \), where \( \times \) can be \( a + \) or \( \times \) operation.
3. There is a zero element \((0)\) in the field, such that \( a + 0 = a \).
4. There is a unity element \((1)\) in the field, such that \( a \times 1 = a \).
5. Every element in \( \text{GF}(q) \) has an additive inverse, such that their addition produces the element \( 0 \). For example, if \( a, -a \) are elements in \( \text{GF}(q) \), then \( a + (-a) = 0 \).
6. Every nonzero element in \( \text{GF}(q) \) has a multiplicative inverse, such that their multiplication produces the element \( 1 \). Thus, \( a \times a^{-1} = 1 \).
7. The associative, commutative and distributive laws apply in \( \text{GF}(q) \).

If \( q \) is a prime number, then \( \text{GF}(q) \) is known as a prime field. The multiplication and addition operations are simply ordinary multiplication and addition modulo \( q \). As in ordinary arithmetic, it is possible to perform a \( \text{GF}(q) \) multiplication by using logarithms.
3.2 Forward Error Correction Coding

This is done by treating the primitive element of the field, \( \alpha \) (whose powers constitute all the nonzero elements), as the base.

If \( q \) is an integer power of a prime number, \( p \), then \( \text{GF}(q = p^m) \) is termed as an extension field. Addition and multiplication will be performed modulo a special degree-\( m \) polynomial, \( p(x) \), which is known as a primitive polynomial. The coefficients of \( p(x) \) come from the field \( \text{GF}(p) \). In most cases, either the binary field (that is, \( \text{GF}(2) \)) or its extension field (that is, \( \text{GF}(2^m) \)) is used, and is assumed henceforth.

3.2.1.2 Description of BCH Codes

For any non-zero positive integers, \( m \) and \( t \) (\( t < 2^{m-1} \)), a binary BCH code exists with the following parameters:

\[
\begin{align*}
\text{Block length:} & \quad n = 2^m - 1 \\
\text{Number of redundancy bits:} & \quad n - k \leq mt \\
\text{Minimum distance:} & \quad d_{\text{min}} \geq 2t + 1
\end{align*}
\]  

Such a code can correct any \( t \) or fewer errors in a block of \( n \) bits. It is also common to term this code as a \( t \)-error-correcting code. The formulation of a specified BCH code begins with its generator polynomial, \( g(x) \) with roots from the Galois field \( \text{GF}(2^m) \). Thus, a \( t \)-error-correcting BCH code may, among other possibilities, have:

\[ \alpha, \alpha^2, \ldots, \alpha^{2t-1}, \alpha^{2t} \]

as roots, where \( \alpha \) is a primitive element of \( \text{GF}(2^m) \). These roots satisfy the following:

\[ g(\alpha^i) = 0, \quad \text{for } 1 \leq i \leq 2t \]
3.2 Forward Error Correction Coding

The generator polynomial for the BCH code is then:

\[ g(x) = \text{LCM} \{ \phi_1(x), \phi_2(x), \ldots, \phi_{2t}(x) \} \tag{3.7} \]

where \( \phi_1(x) \) is the minimal polynomial of \( \alpha^j \). It follows that any \( t \)-error-correcting BCH code can be constructed once the corresponding generator polynomial is obtained. Indeed, as complex as it may seem, the derivation of the generator polynomial is actually highly algorithmic in nature. A computer program can be used to produce a generator polynomial with arbitrary parameters.

3.2.1.3 Encoding

The encoding of a given information bit sequence, of length (or dimension) \( k \), can be accomplished in two ways: using generator matrices, or treating the code as cyclic.

For a BCH code with the generator polynomial:

\[ g(x) = g_0 + g_1X + \ldots + g_{n-k-1}X^{n-k-1} + g_{n-k}X^{n-k} \tag{3.8} \]

the encoding process for the first method involves multiplying the \( k \)-tuple information sequence by the following \( k \)-by-\( n \) matrix:

\[
G = \begin{bmatrix}
g_0 & g_1 & g_2 & \ldots & g_{n-k} & 0 & 0 & 0 & \ldots & 0 \\
0 & g_0 & g_1 & g_2 & \ldots & g_{n-k} & 0 & 0 & \ldots & 0 \\
0 & 0 & g_0 & g_1 & g_2 & \ldots & g_{n-k} & 0 & \ldots & 0 \\
& & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \ldots & 0 & g_0 & g_1 & g_2 & \ldots & g_{n-k}
\end{bmatrix} \tag{3.9}
\]
Hence, if the input vector is denoted as \( u \) and its corresponding codeword as \( v \), then:

\[
v = u \cdot G
\]  

(3.10)

In encoding using \( G \) in the above form, the input sequence is not preserved in the codeword, \( v \). This is known as non-systematic encoding. It is possible to re-arrange \( G \) such that \( u \) is duplicated in either the high or low order positions of \( v \), i.e., systematic encoding of \( u \).

A more common method of systematic encoding in BCH codes would be to treat it as a cyclic code. Mathematically, the encoding process in a cyclic form is:

\[
v(x) = u(x) \cdot x^{n-k} + \left[ u(x) \cdot x^{n-k} \mod g(x) \right]
\]  

(3.11)

This encoding operation can be implemented using shift registers, as shown in Figure 3.4, which generates the last term on the RHS of Equation 3.11.

Figure 3.4: Encoding circuit used in cyclic codes.

Alternative arrangements of this shift register encoding circuit exist for more efficiency and speed.

### 3.2.1.4 Decoding

While it appears that the channel encoding process is relatively straightforward, decoding a received codeword is a more difficult task. In actuality, much research effort on FEC codes is directed at the search for efficient decoding methods. Again, the texts (for
3.2 Forward Error Correction Coding

example, [8]) should be sought for more in-depth discussion of the various approaches in channel decoding.

The first step in decoding a received, possibly corrupted, sequence \( r(x) \) is to compute its syndrome, \( S \):

\[
S_i = r(\alpha^i), \quad \text{for } 1 \leq i \leq 2t
\]  

(3.12)

Since BCH codes are linear, then:

\[
r(x) = v(x) + e(x)
\]  

(3.13)

where \( e(x) \) represents the error pattern and \( v(x) \) is the intended codeword being conveyed. Thus, from Equation 3.12:

\[
S_i = v(\alpha^i) + e(\alpha^i)
\]

\[
= e(\alpha^i), \quad \text{since } v(\alpha^i) = 0
\]  

(3.14)

This shows that the syndrome of a received sequence is only dependent on the error pattern. Subsequent steps in the process of recovering the information sequence are involved with determining what this error pattern is, without regard of what the actual information sequence is.

If \( e(x) \) were to be defined as:

\[
e(x) = x^{j_1} + x^{j_2} + \ldots + x^{j_v} \quad \text{for } 0 \leq j_1 < j_2 < \ldots < j_v \leq n - 1
\]  

(3.15)

where \( v \leq t \), then Equation 3.14 can be expanded and re-written as:
3.2 Forward Error Correction Coding

\[ S_1 = \alpha^{j_1} + \alpha^{j_2} + \ldots + \alpha^{j_v} \]
\[ S_2 = (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \ldots + (\alpha^{j_v})^2 \]
\[ \vdots \]
\[ S_{2t} = (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \ldots + (\alpha^{j_v})^{2t} \]  

(3.16)

The unknowns in the above equations, \{\alpha^{j_i} ; 1 \leq i \leq v\}, are required to be solved by a given decoding algorithm, whether directly (for example, Peterson’s method) or indirectly. Decoding algorithms, which attempt to solve Equation 3.16 indirectly, are used for BCH codes with multiple correction capabilities, since they are more efficient. These algorithms usually adhere to the following steps:

1. Compute the syndromes.
2. Compute the error-location polynomial, using the Berlekamp-Massey algorithm.
3. Obtain the roots of the error-location polynomial using Chien’s search.
4. Perform the error correction by inverting the bit indicated by the inverse of the roots obtained earlier.

A different approach in decoding a received codeword is to use error trapping. This method exploits the fact that all BCH codes are cyclic, and thus, if the positions where the errors occur can be constrained to some known positions or “windows” through cyclic shifting, the errors are “trapped”, and decoding results. Error trapping is a form of information set decoding, as with others such as Minimum Weight Decoding (MWD) [11].

Error Trap Decoding

Let the received codeword be denoted as:

\[ r(x) = v(x) + e(x) \]  

(3.17)
Then, the syndrome, $s(x)$, is obtained through the operation:

$$s(x) = r(x) \mod g(x)$$

$$= e(x) \mod g(x) \quad (3.18)$$

since $g(x)$ is a factor of $v(x)$. Suppose in a $t$-error-correcting BCH code, the errors are confined to the least significant $(n - k)$ positions of the received codeword, that is:

$$e(x) = x^{j_1} + x^{j_2} + \ldots + x^{j_i}, \quad 0 \leq j_i \leq n - k - 1, \quad 1 \leq i \leq t \quad (3.19)$$

Then, the syndrome, $s(x)$ will be equal to $e(x)$ since $e(x)$ has a degree less than $(n - k)$. Hence, when such a situation occurs, the transmitted sequence, $v(x)$, can be simply obtained as:

$$v(x) = r(x) - s(x) \quad (3.20)$$

For other error patterns, $e'(x)$, a similar decoding process can be used if:

$$e'(x) = x^r e(x) (\mod g(x)) \quad (3.21)$$

where $r$ is some integer and the error span is no more than $(n - k)$. Thus, what is needed are just cyclic shifts of the error pattern, $e'(x)$ such that it equals $e(x)$ in the above form. In terms of implementation, the shift register circuit, after the syndrome has been computed, is simply clocked with no input. The cyclic property exploited for error trapping is illustrated in Figure 3.5.

In many cases, including the Golay $(23,12)$ triple-error-correcting code, it is possible that a decodable error pattern has a span greater than $(n - k)$. In such instances, decoding using this technique will never succeed, since a section of the error pattern will always be outside the syndrome window. The problem in these cases can be stated as thus: if it is possible to “see” all possible permutations of an error pattern of no more than $t$ errors, then the code is decodable using error trapping; otherwise, additional windows are necessary to
cover the "blind" positions. In the systematic search decoder [30], these additional windows are created by toggling each bit in a systematic way. Error trapping then proceeds in the usual fashion. Thus, this is no different from a trial and error search procedure for the blind window. Kasami [26] provided an analysis in which he identified the additional windows for the Golay code as only being at positions 16 and 17. The basis for the choice of the blind windows involves a mathematical treatment of the minimum covering problem. In the following, an alternative approach will be described.

**Location of the Blind Windows [18]**

Consider the Golay (23,12) code. All triple-weight error patterns have two errors which do not span more than \((n - k)\) positions. Therefore, the syndrome window can always "see" at least two errors. Thus, for the third "blind" error, external windows, in addition and outside of the syndrome window, are required. Error patterns of weight less than three can be considered as special instances of the above consideration.

Before proceeding, it is convenient to define a quantity, termed displacement, which relates the "distance" of the two given positions, taking into account the cyclic property. Thus, for any weight-three error pattern, a maximum displacement value of \(\left\lfloor \frac{n-r}{t} \right\rfloor = 6\).
3.2 Forward Error Correction Coding

is possible. (\( \lfloor x \rfloor \) means taking the largest integer smaller or equal to \( x \).) Displacement values which are larger than 6 need not be considered by virtue of the cyclic property. For example, a displacement of 8 is equivalent to the situation where the displacement value ranges from 1, 2, \ldots, 6. Hence, by analysing combinations of errors with displacements from 1 to 6 and deriving the external windows required, a full error-trap decoding algorithm can be formulated.

(1) Consider all errors being displaced from each other by 6 positions, as shown in Figure 3.6. The blind spots will be at positions 14, 15 and 16, but since the syndrome window can further shift by at least 3 positions before it loses track of the two errors, an external window at position 16 can shift into positions 14 and 15. Thus, the only required blind window would just be at position 16.

(2) For a displacement of 5, the blind spots are from 12 to 17. Using the external window at position 16 obtained earlier, positions 12 to 15 are cyclically covered. Since position 17 remains blind, an additional window at this position is required.

(3) For a displacement of 4, positions 11 to 18 are blind; again, positions 16 and 17 are sufficient since a maximum of 5 cyclic shifts are permissible.

Figure 3.6: Blind window(s) determination with errors displaced by 6 in the syndrome window.
(4) For displacements of 3 or less, the situation is similar to the preceding one, i.e., the blind spots are covered using either the syndrome window with cyclic shifts, or by the two positions determined earlier.

In conclusion, the minimum number of blind windows required for a complete error trapping algorithm for the Golay \((23,12)\) code would be just 2, at positions 16 and 17.

This method of determining the blind windows is also applicable for other cyclic codes, for example, Reed Solomon codes. For triple-error-correcting codes with syndrome windows which can always "see" two errors, the above procedure is directly applicable since the condition:

\[ n - k \geq \frac{n}{t} \]  

is always satisfied.

### 3.2.1.5 Reed Solomon Codes

Although the codes discussed so far are binary codes, BCH codes also include sub-classes of non-binary codes, the most important of which are Reed Solomon (RS) codes. RS codes work with symbols from \(GF(q)\), where \(q\) is any power of a prime number (2 in the case...
A $t$-error-correcting RS code has the following parameters:

- **Block length**: $n = q - 1$
- **Number of parity symbols**: $n - k = 2t$
- **Minimum distance**: $d_{\text{min}} = 2t + 1$

It should be noted that in RS codes, the minimum distance, $d_{\text{min}} = n - k - 1$, is the maximum possible value for any code with the same $n$ and $k$ (see Bibliography). Thus, RS codes are also known as **maximum distance separable (MDS)** codes. The generator polynomial for RS codes follow a similar structure to that of binary BCH codes; for example:

$$g(x) = (x + \alpha)(x + \alpha^2)\ldots(x + \alpha^{2t})$$

$$= \prod_{i=1}^{2t}(x + \alpha^i) \quad (3.23)$$

In our applications, we will use a systematic implementation. RS codes treat a $\text{GF}(2^m)$ symbol as a single entity, and thus, can correct error bursts with relative ease. They are suitable in diffused environments, such as mobile radio channels, where errors occur together instead of randomly. The encoding and decoding procedures in RS codes are varied and enriched by transform techniques which offer computational and implementational advantages. A thorough treatment of both time and frequency domain techniques for encoding and decoding of RS codes can be found in texts on error control coding (for example, [5]), and hence will be omitted here.

### 3.2.1.6 Erasure Decoding

In real digital communication systems, there are instances, when the received signal is corrupted to a degree such that any output symbol is possible, or there is no information as to what the symbol was. Such a symbol is known as an erasure, and it effectively inserts another symbol in the BSC output symbol set. This is illustrated in Figure 3.8.
A number of erasures, $e$, can be incorporated into decoders for BCH codes subject to the following constraint:

$$d_{\text{min}} > 2t + e$$  \hspace{1cm} (3.24)

The decoding can exploit this extra degree of freedom to provide more error correction power, sometimes as much as soft decision decoding (see next section). While this is relatively trivial in binary codes (by decoding twice, the first with erasures replaced by 1's, and the next with 0's), algebraic decoding for RS codes requires some modifications for an errors-and-erasures algorithm. This is treated in considerable detail in [8].

### 3.2.1.7 Soft Decision Decoding

The inclusion of erasures can be generalised to that of a channel output having $Q$ symbols, $Q > 2$. This is depicted in Figure 3.9. Effectively, the received analogue signal from the channel is mapped into a symbol (out of $Q$ possibilities) which relates the degree of similarity of the received signal to those possible at the receiver. Utilising these soft symbols in a FEC decoder is, therefore, akin to a matched filtering operation.

To quantify the "closeness" of a codeword, with soft decisions, to that hypothesised by the FEC decoder, an appropriate distance measure or cost function is required. For additive white Gaussian noise (AWGN) channels, the optimum distance cost function is to
3.2 Forward Error Correction Coding

Figure 3.9: Channel output with soft outputs ($Q = 8$).

use the Euclidean distance:

$$d_i^2 = \sum_{i=0}^{n-1} [r(i) - S_i(i)]^2, \quad 0 \leq r(i), S_i(i) \leq Q - 1$$  \hspace{1cm} (3.25)

where $r(i)$ is the $i$-th symbol in a received sequence and $S_i$ is the hypothesised codeword. For $Q = 2$ (that is, a BSC) the distance cost function is equivalent to just summing of symbol differences, or Hamming distance.

To date, there is no decoding algorithm for block codes which incorporates soft decisions in an integrated manner. Chase [7], Forney [23] and many other researchers have described techniques, which make use of hard decision decoding (errors or errors-and-erasures) algorithms, for soft decision decoding implementations. Essentially, these techniques use soft decisions to generate a set of test error patterns which are then used to perturb the received codeword prior to hard decision decoding. The decoded codeword is then the one which minimises the distance or cost function, $d_i$. Chase's algorithm is illustrated in Figure 3.10 on page 63.
3.2 Forward Error Correction Coding

The primary issue in soft decision decoding is the computation complexity that can realistically be tolerated, which is an exponential function of the number of error patterns used. However, soft decision decoding can provide a considerable increase in the code’s performance. Coding gain, which quantifies the relative effectiveness of a coding scheme in a noisy channel, in soft decision decoding is often about a few decibels better (typically around 2 dB at BERs of 10^{-5}) than in hard decision decoders.

Figure 3.10: Flow diagram of Chase’s Algorithm [7] for soft decision decoding of linear block codes.
3.2.2 Convolutional Codes

Convolutional codes form the other widely used class of FEC codes. Unlike BCH codes, which depend on algebraic properties for constructing good classes of codes and developing decoding algorithms, good convolutional codes are found by computerised searches. The decoding procedure of a convolutional code is based on the sequential state machine nature of the encoder itself. In addition, the implementation simplicity and ease of utilising soft decisions has resulted in, not surprisingly, more widespread use of convolutional codes in practice (as evident in Europe's GSM, USA's IS54 and the Japanese JDC mobile communication systems).

In contrast to BCH codes, the convolutional encoding operation involves memory: the $n_0$ outputs of an encoding operation at any given time are dependent not only on the current block of $k_0$ inputs, but also on $m$ previous input blocks. Such a code is termed as a $(n_0, k_0, m)$ convolutional code. In general, larger values of $m$ are necessary to achieve low error rates. In this section, only rate $1/n_0$ convolutional codes (that is, $k_0 = 1$) are discussed. (A general in-depth treatment of convolutional codes is avoided since the texts like [8] do a better job.)

3.2.2.1 Encoding

An encoder for a $(2,1,2)$ code (rate $= \frac{1}{2}$) is depicted as in Figure 3.11. With the encoder initially cleared, the register stages are shifted with every input bit, and the output bits obtained using modulo-2 addition. The encoding circuit will have to be cleared upon coding of the last input bit by simply appending $m$ zeroes to the input sequence.

There are many ways of representing the behaviour of the encoding process, and these include state diagrams, generator polynomials and code trellises. For decoding convolutional codes, the code trellis is often the most appropriate.
3.2.2.2 Decoding: The Viterbi Algorithm

At the receiving end, a convolutional decoder has to reproduce, as accurately as possible, the input sequence from the observed and possibly corrupted, coded sequence. Because a block of \(n_0\) bits is generated with contributions from previous inputs, the mapping of the output bits to the corresponding input bit has to take this memory effect into account. Consequently, all possible permutations of the previous \(n\) input bits must be considered. This can be compactly described by a code trellis, as illustrated in Figure 3.12. Not sur-

![Code trellis for a (2,1,2) convolutional code.](image-url)
3.2 Forward Error Correction Coding

prisingly, the code trellis is often used as an aid in decoding a convolutional code sequence. Viterbi [30] introduced a decoding algorithm for convolutional codes, best illustrated using a trellis, which is efficient and provides a maximum likelihood performance. This algorithm has come to be known as the Viterbi Algorithm (VA), and will be the main subject of discussion in this section. Other decoding methods have also been devised, but will not be considered here.

The basic operation of the VA in decoding convolutional codes is as shown in Figure 3.13. For every trellis node that could be arrived at at a given time, paths emanating from a total of $2^k_0$ previous nodes have to be considered.

![Figure 3.13: Example of path selection in the Viterbi algorithm.](image)

Hence, the selection criterion for the optimal path is to choose one which minimises the $j$-th node's distance cost function (or metric), $d_{\text{node}}(j, n)$:

$$d_{\text{node}}(j, n) = \min_i \left\{ d_{\text{node}}(i, n-1) + d_{\text{path}}(i, j) \right\}$$  \hspace{1cm} (3.26)

where $n$ indicates the frame index. The path metric is computed as follows:

$$d_{\text{path}}(i, j) = \sum_{p=0}^{2^k_0-1} \left[ s_{i,j}(p) - r(p) \right]^2$$  \hspace{1cm} (3.27)

$s_{i,j}(p)$ denotes the $p$-th bit of the output sequence from node $i$ to node $j$, and $r(p)$ is the $p$-th bit of the received frame. For a complete decoding procedure of a received code
sequence, the node metrics are initialised as follows:

\[
d_{\text{node}}(i, 0) = \begin{cases} 
0 & \text{for } i = 0, \\
+\infty & \text{for } 1 \leq i \leq 2^{m_{k_0}} - 1
\end{cases}
\] (3.28)

Since the encoder's memory is cleared at the end of the code sequence (which, in practice, is related to the speech frame), the optimal node for decoding the last bit will be the zero-th node.

This highly modular accumulate-compare-and-select "building block" behind the VA's operation is the key behind the simplicity of the decoder for convolutional codes. Consequently, integrated circuits implementing the decoder can operate at very high data rates. The basic limitation of the VA is that the storage of \(2^{m_{k_0}}\) nodes for every single iteration is required, precluding the use of large memory orders. At the same time, computations increase at an exponential rate, thus making the implementation of powerful convolutional codes both memory and computationally intensive.

### 3.2.2.3 Soft Decision Decoding

In addition to the relatively high speed a convolutional decoder can operate at, the adaptation of soft decisions for the VA is relatively trivial, which is certainly a significant advantage over algebraic decoding BCH codes. All that is required of the VA is to extend the range of values from which the path metric, \(d_{\text{path}}(i, j)\), is derived (see Equation 3.27). That is:

\[
0 \leq s_{i,j}(k), r(k) \leq Q - 1, \text{ for } 0 \leq k \leq 2^{m_{k_0}} - 1
\] (3.29)

where \(Q\) is usually a value of 8. Thus, soft decision decoding of convolutional codes, which can improve the coding gain by a few decibels (dB) or more, is often regarded as a standard implementation. Since the VA always attempts to match the received bit stream with the best sequence, the probability of the chosen sequence is maximised. Thus, decoding using the VA is also known as Maximum Likelihood (ML) decoding.
3.2.2.4 Punctured Codes

If one or more bits from each output $n_0$ bits were not to be transmitted on a regular basis, then the resulting convolutional code is said to be punctured. For instance, in a $(2,1,2)$ code, if the first output bit (see Figure 3.11, page 65) were not to be relayed to the receiving end on alternate frames, then for every two input bits, three output bits will be produced. Thus, the rate of the code has been changed from $1/2$ to $2/3$.

Punctured convolutional codes offer the communications engineer the ability to adapt the FEC coding scheme to the system requirements without excessively modifying either the encoding or decoding algorithm. At the same time, puncturing a binary code to produce the same rate as that obtained by using an optimum code (which encodes a 2-bit input using 3 bits) means considerable computation reductions at the cost of little degradation in performance.

Decoding a punctured code is carried out in very much the same way, using the Viterbi algorithm. Bits that are not transmitted are treated as erasures at the receiving end, that is, the path metric is computed without considering the punctured bit. It is worthwhile noting that the IS54 and JDC mobile systems use punctured codes to obtain the rates desired.

3.3 Key Considerations In Coding Synergy

Having established that error control in speech coders should rely on a combination of zero-redundancy techniques and FEC coding schemes for the desired performance, the next step in designing a combined source and channel codec will be to consider other aspects such as error sensitivities of the speech model parameters, the quantisation and coding schemes used, the effects of errors on the synthesis process and on the synthesised speech quality. In general, LBR speech coders can tolerate a limited degree of error corruption in their parameters, that is, the synthesised speech is of sufficiently good quality to maintain an
acceptable level of service. Intelligibility is also maintained, but only up to a certain residual BER. Hence, this can be taken advantage of, in the design of FEC schemes, to increase the operational range of the speech coder in noisy channels. To optimise the deployment of FEC, it is necessary to identify characteristics of the speech coder parameters (see Figure 3.14) in terms of the degree of error protection. These are discussed in the following, using the MB-LPC model as an example LBR speech coder.

![Diagram](image)

Figure 3.14: Considerations in the determining a LBR speech coder’s need for error control.

### 3.3.1 Error Sensitivity

The lack of resilience of a speech model parameter to channel corruptions is called error sensitivity. This is often related to the rate at which the synthesised speech quality degrades at increasing BERs. Thus, the more sensitive a given parameter is to errors, the more noticeable are its effects on the subjective quality.

The degree of error sensitivity of a LBR coder parameter can be deduced from its role in the speech synthesis process. For example, the Line Spectral Frequencies (LSFs) carry information about the spectral envelope of speech. Hence, discrepancies between
the received and the intended LSFs will be reflected in the speech spectral characteristics, resulting in artefacts such as whistles and short grunts (explosions of low frequency energy). Similarly, errors in the pitch parameter in the MB-LPC vocoder will produce a very different synthetic spectrum, thereby losing intelligibility and sometimes, speaker identifiability. All these can happen at relatively low BERs. At the other extreme, the voicing decisions (V-UV) can tolerate a much higher BER whilst still maintaining good intelligibility. Subjectively, a loss in quality is observed, but not on a pronounced scale as for the LSFs or pitch.

It would appear that the pitch and LSFs would need to be error-protected heavily. However, it would be premature to say that these should rely solely on FEC without considering other factors.

3.3.2 Confidence of Source Criteria

As mentioned earlier, source criteria represent ways by which degradations in the speech quality can be predicted. Source criteria can generally be divided into two classes: hard and soft. For example, the monotonicity criterion of the LSFs (as discussed in Section 2.2.3) is required to be observed regardless of the synthesis conditions. It is, thus, a hard source criterion since violation will definitely cause degradations. As a result, errors in a LSF vector can be inferred, when violation of a criterion occurs, to some degree without using FEC at all.

The pitch parameter, during highly voiced regions, has considerable correlations with values from previous frames. Thus, a limited deviation of the current pitch value from the previous one will be expected in such situations, constituting a form of intrinsic error control. However, the synthesis of the speech itself might be erroneous, and this creates uncertainties in the detection of such speech regions. Thus, the source criterion for the pitch parameter is considered to be soft since it cannot be enforced with definite confidence.

The degree of enforcing source criteria on a speech coder parameter can directly affect
the error control requirements. If channel errors can be detected with a good degree of confidence (as with hard source criteria), then the parameter need not necessarily be heavily protected using FEC. Also, indications by source criteria can furnish the channel decoder with additional information about the received symbols' states to improve the FEC's coding performance to beyond that of Maximum Likelihood Decoding (MLD). This is discussed in Chapter 5.

3.3.3 Responsiveness to Error Concealment

Error concealment refers to a general class of techniques which achieve a degree of error correction in a subjective sense. Corrupted parameters, which are not recoverable due to lack of FEC coding or otherwise, can be "corrected" with a certain degree of accuracy while minimising the speech quality degradations. For instance, smoothing or substitution techniques operate by using observed statistical characteristics to detect and modify a corrupted parameter. This can be considered as a combination of error concealment with source criteria. Parameters which respond readily to error concealment algorithms could avoid the use of FEC altogether, while still providing an acceptable level of service quality even when corrupted. Such a form of error control is very attractive since it means protection from corruptions in an efficient and optimal way.

In the MB-LPC vocoder, the excitation scaling factor, $\sigma$, is particular responsive to zero redundancy techniques which attempt to smooth out errors in the received parameter. Thus, given the knowledge that the received value has a high probability of being corrupted, subjectively good performance can be achieved by simply copying and reducing the scaling factor from the previous frame. This is discussed further in Chapter 6.
3.3 Key Considerations In Coding Synergy

3.3.4 Error Propagation

The speech synthesis process may not simply depend on the coder parameters of a given frame, but also on contributions from previous frames. This is true of any speech coder which relies on feedback prediction, that is, signal samples are analysed or synthesised using knowledge of previous output samples. Evidently, these coders use memory for the prediction process.

In the presence of errors, the synthesis process is no longer optimal in the sense that the speech samples contain degradations. These affect incoming frames since these speech samples are re-used, thereby spreading the degradations. This is known as error propagation. Fortunately, the MBE and the MB-LPC coders do not depend significantly on feedback prediction. In contrast, the CELP coder's LTP filter uses memory, and thus has the potential of subjectively amplifying the effects of errors.

The MBE coder was, in 1990, chosen to be the INMARSAT Standard M [19] coding scheme partly due to its high robustness to channel errors. This observation was also noted from the behaviour of the MB-LPC vocoder under similar channel conditions. Hence, the reliance of a given speech coding scheme on FEC to combat errors can be significantly reduced if the problems of error propagation are considered at the design stages.

3.3.5 Quantisation Schemes

The type of quantisation schemes used in speech coders is another area for scrutiny where combined source and channel coding considerations are concerned. Schemes, differing from scalar quantisation (SQ), attempt to reduce the source bit rate at the expense of increasing the parameter's error sensitivity. Examples of such quantisation schemes are vector quantisation (VQ), adaptive quantisation (AQ) and differential quantisation (DQ).

In VQ, the bitmaps are rendered less robust since only one bit in error is sufficient to completely modify the coded parameters. The issue is then how different the received
3.4 Error Control Configurations

Having discussed the various factors influencing the degree of error control using FEC, it would appear that the best solution would be to use unequal error protection. The simplest way to implement this would be to identify and FEC-protect crucial parameters while leaving the rest to zero-redundancy techniques. In doing so, the use of redundancy for a speech and channel coding system is minimised while still realising the speech quality desired. Varying degrees of FEC protection can be used, but this may make the scheme possibly too complex to be an attractive option, particularly in block FEC codes. Convolutional codes provide a convenient way of implementing “different” error correction capabilities in the same code: symbols at the beginning and end of a code sequence have a lower residual error rate compared to those in the middle [4].

3.4.1 Speech Quality Assessment

How, then, can a parameter be determined as crucial? Researchers [9] have used a bit-by-bit systematic approach to obtain indications of parameters’ characteristics contributing to the resultant speech quality, by using an objective measure such as segmental signal to noise ratio (segSNR) or log spectral distance (SD). However, in these approaches, the assumption
that these objective measures correlate well with the perceived subjective quality does not hold, particularly for vocoders such as the MB-LPC. In addition, no consideration as to the preference for the type of artefacts (for example, clicks, grunts, bangs or squeaks) is accommodated. This can, potentially, cause a lot of research efforts to be mis-directed or even wasted in attempts to produce robust and high quality speech communication systems.

Hence, as the search for a perfect objective measure continues, the only way to test and quantify the effectiveness of a source and channel coding technique is through subjective listening tests using the Mean Opinion Score (MOS) [34] (see Table 3.1). This approach of testing is, to date, the most reliable form of obtaining the human perceptual preferences. The MOS is the most popular way of specifying the quality of the synthesised speech perceived and is very often used as a benchmark in assessing speech coding implementations. It is this approach that is preferred in the work throughout in this thesis.

<table>
<thead>
<tr>
<th>MOS</th>
<th>Quality</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Excellent</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>4</td>
<td>Good</td>
<td>Perceptible, but not annoying</td>
</tr>
<tr>
<td>3</td>
<td>Fair</td>
<td>Slightly annoying</td>
</tr>
<tr>
<td>2</td>
<td>Poor</td>
<td>Annoying</td>
</tr>
<tr>
<td>1</td>
<td>Unsatisfactory</td>
<td>Very annoying</td>
</tr>
</tbody>
</table>

Table 3.1: 5-point MOS grading scales [34].

### 3.4.2 Types of Configuration

In the process of selecting optimal and efficient error control strategies for LBR speech coders, it is constructive to classify the parameters in terms of the need for the type of error protection. Many forms of such techniques can be categorised into three types, as depicted in Figure 3.15. The problem of matching the speech quality and robustness requirements is thus transformed to that of selecting the best error control setup, from the illustrated possibilities, for a given parameter. This process is, in fact, closely related to
3.4 Error Control Configurations

Figure 3.15: Types of error control schemes in LBR speech coders.

the issues on the coding synergy discussed earlier, and is best depicted in Figure 3.16. To illustrate how the choice of the error control setup can be matched to a LBR speech coder parameter, some examples are given in the discussions.

Type 0 – no error protection

The human auditory perception is known to be able to withstand some amount of degradations without losing the contents of the intended message. Thus, the indices for the stochastic codebook excitation in the CELP coder could employ a Type 0 error control scheme since the degradations ascribed to erroneous vectors are subjectively less noticeable.
3.4 Error Control Configurations

Similarly, the higher order V-UV decisions in the MB-LPC vocoder need not be protected as they can tolerate a high degree of channel corruptions.

**Type 1 – protection using ZR techniques**

Parameters which demonstrate a high degree of reliable source criteria and responsiveness to smoothing or substitution techniques, as that in LSFs, can achieve some degree of error protection using zero redundancy algorithms. In essence, errors in these parameters can be *concealed* such that they are subjectively not annoying. Hence, the degree of error protection needed can be satisfied using a Type 1 setup.
Type 2 – protection using FEC schemes and ZR techniques

The next category is for parameters which contribute to the absolute quality of the synthesised speech, or are very sensitive to errors. For the MB-LPC vocoder, the pitch parameter falls into this category, thus requiring the error protection capabilities provided for by Type 2 schemes. Alternatively viewed, the pitch parameter is being protected such that the occurrences of errors are reduced by the FEC coding scheme. In bad channel conditions, zero-redundancy schemes then ensure that the speech synthesis process degrades in a graceful manner.

Type Hybrid – combined protection

It is well known that the binary representation of a speech coder parameter has, in itself, an uneven distribution of error sensitivities. For example, the excitation scaling factor, \( \sigma \), in the MB-LPC vocoder produces speech degradations which are more noticeable when the most significant bit (MSB) is corrupted as compared to the least significant bit (LSB). Undoubtedly, the MSBs will require a higher level of error protection than the LSBs under the same channel conditions.

It should be noted that, in general, many speech coder parameters will exhibit such a need for unequal error protection. In terms of the error control types suitable for such circumstances, a combination of one or more of the previously mentioned types will suffice. For instance, error protection of the MB-LPC’s pitch parameter using Type 2 for the five MSBs and Type 0 for the remaining bits is able to provide a subjectively good performance whilst not seriously compromising the objectives of spectral efficiency.

3.4.3 Example System – The MB-LPC Vocoder

The various error control configurations detailed earlier provide the basic capabilities for varying degrees of protection of the LBR speech coder parameters from channel errors.
3.4 Error Control Configurations

To illustrate how these configurations can be applied in real applications, the MB-LPC vocoder is used here as an example.

The characteristics of the MB-LPC coder parameters, taking into account the considerations for coding synergy, is detailed in Table 3.2. It should be noted that in terms of the difficulties encountered from the effects of error propagation, the MB-LPC coder is a good example of a robust LBR speech coder.

<table>
<thead>
<tr>
<th>Error sensitivity</th>
<th>Pitch</th>
<th>LSFs</th>
<th>V-UV</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Moderate</td>
</tr>
<tr>
<td>Confidence of source criteria</td>
<td>High/Moderate</td>
<td>High</td>
<td>Low</td>
<td>Moderate</td>
</tr>
<tr>
<td>Responsiveness to error concealment</td>
<td>High/Moderate</td>
<td>Moderate</td>
<td>High</td>
<td>Moderate</td>
</tr>
<tr>
<td>Error propagation</td>
<td>None</td>
<td>Moderate</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Number of bits</td>
<td>8</td>
<td>37</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3.2: Factors in coding synergy for the MB-LPC vocoder.

From Table 3.2, it can be seen that the pitch and the LSFs will need considerable FEC-protection if high speech quality is desirable. However, for minimum demands on coding redundancy in reasonable channel conditions, the LSFs can rely on zero redundancy.
3.5 Zero Redundancy Schemes

Techniques, while the MSBs of the pitch, V-UV and energy parameters can be protected using channel coding. Thus, a combination of all the various error control types is the best compromise in terms of quality and robustness of the speech coding scheme. This is depicted in Table 3.3

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Error control type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitch</td>
<td>Hybrid</td>
<td>5 MSBs - type 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 LSBs - type 0</td>
</tr>
<tr>
<td>LSFs</td>
<td>Type 1</td>
<td></td>
</tr>
<tr>
<td>V-UV</td>
<td>Hybrid</td>
<td>4 MSBs - type 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 LSBs - type 0</td>
</tr>
<tr>
<td>Energy</td>
<td>Hybrid</td>
<td>3 MSBs - type 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 LSBs - type 0</td>
</tr>
</tbody>
</table>

Table 3.3: Error protection for the MB-LPC vocoder.

From the above exercise, it can be seen that the reliance on channel coding is kept to a minimum – a total of 12 (= 5 + 4 + 3) bits only need to be protected. The only other concern, in terms of error control, is that of the LSFs, whereby it is assumed that a parameter smoothing or substitution algorithm (see Chapter 4) is able to cope with transmission corruptions. In the event where a higher transmission rate is possible, channel coding of the LSFs’ bit stream should be deployed so as to maximise the synthesised speech quality. Again, either a Type 2 or Hybrid configuration should be used.

3.5 Zero Redundancy Schemes

The primary function of zero-redundancy algorithms is to “filter” away residual errors present in the parametric information which may be directly due to the transmission channel itself, or to the channel decoder. Preceding discussions have, so far, referred to zero-
redundancy algorithms in a non-descriptive way. In this section, the various forms in which zero-redundancy algorithms can be implemented are discussed, particularly for the CELP and MB-LPC speech coders.

### 3.5.1 Pseudo-Gray Coding

The bitmaps representing the speech coder parameters also have a degree of sensitivity of errors. For instance, in a natural bit assignment scheme, the received parameters can differ considerably from the intended (quantised) version with only one bit change. This renders the coder parameters highly sensitive to corruptions. Figure 3.18 gives a visualisation of the error sensitivities of two bitmap assignment schemes. In pseudo-Gray [4]

![Figure 3.18: Different bit assignment schemes (a) natural binary, (b) pseudo-Gray.](image)

coding, the bitmaps of a parameter are assigned in such a way that certain adjacent values' binary codes differ by one bit. Consequently, single-bit errors are rendered subjectively unimportant, which on average has the effect of improving the parameters' resilience to errors. Although the use of source criteria is not directly implied, pseudo-Gray coding can help ease the implementation of zero-redundancy algorithms by reducing the
complexity of bitmap searches using bit-toggling.

The above discussions on pseudo-Gray coding can be extended to that of index assignment for vectors instead of scalars [53]. Interests in such techniques, particularly for vector quantisation has recently been renewed [17,27].

3.5.2 Parameter Substitution/Smoothing

Error control on parameters using smoothing or substitution methods form a large class of ZR techniques. These techniques of achieving error control in the speech coder rely on the intrinsic source criteria of the parameters. The source criteria are, in turn, based on observed characteristics which constrain the parameters in a predictable way. Good examples of coder parameters which respond well to smoothing/substitution techniques are the LSFs, codebook excitation gains and pitch. In general, smoothing techniques used in zero-redundancy algorithms are varied and parameter-specific and thus, such discussions are meaningful in the context of the parameter and speech coder type.

The forms of zero-redundancy schemes for LSFs are discussed in detail in Chapter 4. In the following, smoothing/substitution techniques for the codebook gains and the pitch are discussed since they are the most sensitive parameters in the CELP and MB-LPC coders respectively.

Codebook Gains

For the codebook gains in the CELP coder, the rarely abrupt changes in the speech signal energy are reflected in the standard deviations of the sub-block gains in each frame. This is depicted in Figure 3.19.

Thus, indications of a corrupted gain, \( g(I) \), from a CELP coder with \( N_s \) sub-blocks
3.5 Zero Redundancy Schemes

Figure 3.19: Distribution of excitation gain deviations.

can be derived as follows:

\[ g(I) = |g(i)| \quad \text{if} \quad \min_i \left\{ \frac{1}{N_s - 1} \sum_{j=0, j \neq i}^{N_s-1} [\overline{g} - |g(j)|]^2 \right\}, \quad 0 \leq i \leq N_s - 1 \quad (3.30) \]

where \( I = i \) when Equation 3.30 is satisfied, and

\[ \overline{g} = \frac{1}{N_s} \sum_{i=0}^{N_s-1} |g(i)| \quad (3.31) \]

Having obtained the “corrupted” gain, \( g(I) \) could then be replaced by a suitable value. Hence, for low average channel BERs where the corruption of more than one gain term is unlikely, this technique works adequately without introducing inadvertent distortion.

To cater for situations where two gain terms are corrupted, an improved adaptive gain smoothing algorithm has been formulated in which an additional gain term from the previous frame is used. In addition, use of the “smoothness” of the gains’ magnitudes is exploited by modelling the deviations through the use of an autoregressive (AR) averaging
process. Here, only the magnitudes are of interest since the signs of the gain terms do not significantly degrade the speech quality.

**Improved Excitation Gain Control**

1. Initialisation: set the standard deviation of the differences, $\sigma_d$ and $g(0)$ (the last gain term in the previous frame), to some pre-set values.

2. Compute the average of the codebook gains:

\[
\bar{g} = \frac{1}{N_s + 1} \sum_{i=0}^{N_s} |g(i)|
\]  

(3.32)

where $N_s$ is the number of sub-blocks in the speech analysis frame.

3. Using $\bar{g}$, determine the absolute difference for each gain term.

\[
\delta(i) = |g(i)| - \bar{g} \quad i = 0..N_s
\]  

(3.33)

4. Treating the two largest $\delta(i)$ as possibly due to corrupted gains, flag the corresponding gain terms as suspect values, $g(I_1)$ and $g(I_2)$. Remove the contribution of these two values from $\bar{g}$:

\[
\bar{g} \leftarrow \frac{1}{N_s - 1} \left[ (N_s + 1)\bar{g} - |g(I_1)| - |g(I_2)| \right]
\]  

(3.34)

5. Recompute all the difference values, $\delta(i)$, using the new value of $\bar{g}$ and obtain the average, $\bar{\delta}$ and standard deviation ($\sigma_\delta$) of the difference values of the unsuspected gain terms.

\[
\bar{\delta} = \frac{1}{N_s - 1} \sum_{i=0, i \neq I_1, I_2}^{N_s} \delta(i)
\]  

(3.35)

\[
\sigma_\delta^2 = \frac{1}{N_s - 1} \sum_{i=0, i \neq I_1, I_2}^{N_s} [\delta(i) - \bar{\delta}]^2
\]  

(3.36)
6. Update the long term AR-estimated standard deviation, $\sigma_d$.

$$
\sigma_d = \gamma \sigma_d + (1 - \gamma) \sigma_b
$$

(3.37)

where $\gamma$ is a constant between 0 and 1.

7. The two suspect values are then tested using the following criterion:

$$
|\delta(j) - \bar{d}| < \lambda \sigma_d, \ j = I_1, I_2
$$

(3.38)

where $\lambda$ is an experimentally optimised constant.

8. If the above test fails for either gain term, then the offending gain(s) is replaced by the average value, $\bar{g}$, or a bit toggling procedure can be used to search for a suitable value. Otherwise, it is assumed to have been received correctly.

The robustness of the codebook gains was observed, in subjective listening tests during simulations using a 4300 bps CELP coder, to be improved from BERs of less than 0.5% to about 1.5%. As further enhancements to the above algorithm, decoding confidences can also be used to isolate suspectedly corrupted gain terms. An adapted version of this algorithm used for the excitation gain control in the AUDETEL speech coder [46] is listed in Appendix B.

Pitch

In the MB-LPC vocoder, the pitch parameter responds readily to substitution techniques, as a consequence of the pitch tracking requirements. Therefore, in highly voiced speech segments, pitch values have high inter-frame correlations, as illustrated in Figure 3.20.

Thus, what is needed in the zero-redundancy algorithm at the decoding end is to detect for such speech regions, and exploit this correlation as a source criterion for the received pitch parameter. This can be done by declaring a speech segment voiced if its
3.5 Zero Redundancy Schemes

Figure 3.20: Correlation of the pitch parameter in voiced speech regions (speech frame size of 160 samples).

Energy content exceeds a threshold:

$$ E(n) \geq \alpha \bar{E}(n - 1) $$  \hspace{1cm} (3.39)

where

$$ E(n) = \sum_{i=0}^{N-1} s^2(n, i) $$  \hspace{1cm} (3.40)

$$ \bar{E}(n) = \beta \bar{E}(n - 1) + (1 - \beta) E(n) $$  \hspace{1cm} (3.41)

and $\alpha, \beta$ are experimentally optimised constants, $N$ is the size of the speech frame, and $n$ is the speech frame index. When such a condition is detected, the received pitch value
should be bounded by Equation 2.39 (page 37):

\[ 0.8P(n) \leq P(n-1) \leq 1.2P(n) \]

If this criterion is not observed, it is then highly possible that the speech synthesised will suffer from noticeable artefacts.

### 3.5.3 Parameter De-sensitisation

While the essence in LBR speech coding is to transform speech into a set of seemingly unrelated components, the inter-dependencies of the model parameters can provide useful indications as to the importance of each parameter in the decoding process. From the analysis of the CELP and MB-LPC speech coding schemes (discussed in Chapter 2), it can be observed that some sub-processes in the speech synthesis are more significant in terms of contributions to the subjective quality. Thus, if a sub-process is known to contribute inactively to a known state of the output speech signal (voiced or non-voiced), then the associated parameter can be de-sensitised such that the actual value becomes inconsequential. To do this, the degree of influence of the parameter's associated sub-process and the interaction with other sub-processes should be determined. This can be done by correlating the presence (or absence) of the characteristic parameter value(s) with the state the synthesised speech would be in.

In the MB-LPC vocoder, voiced speech synthesis during non-voiced regions can result in artefacts, such as breathing noises, which are perceivable in the output signal. This is depicted in Figure 3.21. Thus, if uncertainties in the V-UV flags are detected in the received bit stream, such degradations can be prevented by setting the V-UV flags as unvoiced during non-voiced speech, the presence of which can be detected through the incidence of low energy scaling factors coinciding with low output energy. The use of such a technique helps to maintain the coded speech quality during periods of high channel corruptions.
3.5 Zero Redundancy Schemes

For CELP, parameter de-sensitisation can play a similar role during speech segments containing voiced energy. In these circumstances, the majority of the LPC excitation energy is derived from the LTP synthesis process. The stochastic contribution from the codebook parameters can be muted without significant loss of the subjective speech quality. In contrast, as with the above mentioned example, speech synthesis with corrupted excitation gains can produce annoying artefacts, such as clicks and loud bangs.

### 3.5.4 Lost Frame Recovery

In occasional instances where the received data stream is corrupted to an extent such that errors cannot be corrected reliably, it is preferable to flag the frame as *lost* as far as the speech decoder is concerned. Such incidences can happen in the mobile propagation channel where impairments such as co-channel interference, multiple signal paths and fading are encountered. In these situations where the speech frame’s parameters are effectively erased, highly suitable parametric values would have to be substituted in. This is particularly crucial during voiced speech regions since deviations of the replaced values from the optimal ones are potentially noticeable to a large extent.

The best estimates for the contents of an erased speech frame would be that of the previous frame. This is feasible since speech exhibits high inter-frame correlation which
is reflected in the coder parameters. In addition, perceptual enhancements can be ac-
commodated by adapting the previous frame's parameters to match to the lost frame's 
values better. Lost frame recovery techniques vary from coder to coder, but in general, the 
following procedures are usually effective:

1. Scaling factors are repeated but with reduction in magnitudes since decaying signal 
amplitudes are less noticeable when corrupted.

2. Highly correlated values, such as pitch and the LSFs, are repeated from the previous 
frame.

3. Use of parameter de-sensitisation (discussed in the preceding section) where possible.

As an integral and important part of the solutions to the error control problem, the 
treatment of lost speech frames in real-time speech coding applications warrants significant 
attention. This is evident in both Europe's GSM [6] and the INMARSAT Standard M [19] 
mobile phone systems, to name a few. For the MB-LPC vocoder, optimal lost frame recov-
ery techniques and perceptual enhancement methods are discussed in detail in Chapter 6.

3.6 Remarks

The problem of error control in low bit rate speech coders is closely examined in this 
chapter. With the prospect of system demands and constraints on the rise, as expected 
in third generation mobile systems, the predicaments faced in ensuring an acceptable level 
of voice service quality necessitate the treatment of error control in speech coders in its 
own right. In the light of such considerations, this chapter has presented a systematic 
description of the various concerns of coder parameters. Techniques which allow LBR 
speech coders, with corrupted parametric information, to operate with more graceful speech 
degradations have also been reviewed.

The approaches described in the chapter will, to some extent, result in the design
of robust speech coding schemes, responding to channel corruptions and their effects by *lowering* the speech quality. However, for improved performance and operating capabilities, new source-reliant error control methodologies are required, allowing a higher level of speech quality to be achieved with lesser or even no FEC redundancy. These advanced techniques are the subjects of discussions in the following chapters.
Chapter Four

Error Recovery Of Line Spectral Frequencies

In Chapter 2, the concept of linear prediction and its role in low bit rate speech coders, such as CELP and MB-LPC, was described. The model parameters, the LPC coefficients, common in both speech coders were also presented. In addition, the preferred form of the LPC parameters, the Line Spectral Frequencies (LSFs), was also derived. For a complete speech coding system destined for any application, the LSFs will have to be quantised and coded into a bit stream for transmission or storage.

The issues in this chapter are primarily concerned with the error control of LSFs, particularly, without using channel coding. As will be evident in the discussions, LSFs possess excellent and palpable properties which contribute to their high robustness to channel corruptions. This is very important in real-time speech coding systems, such as the UMTS, since savings in FEC redundancy bits can be significant for low bit coding rates. For instance, in the MB-LPC vocoder, the LSFs represent 62% of the speech coder output bit stream; to protect the LSF vector effectively would mean significant demands on the FEC coding scheme. Thus, either a partially FEC-protected scheme, or an effective LSF zero-redundancy technique would have to be used.

In Section 4.1, the various characteristics of LSFs are discussed in some detail. Some
zero-redundancy techniques for error control of LSFs transmitted over noisy channels are then presented before a more effective scheme is described in Section 4.3.

4.1 Properties of LSFs

Linear prediction is an effective method of modelling the quasi-stationary characteristics of speech. Such an observation, exploited in the LPC analysis, is generally limited in many speech coders to a fixed duration or segment of time. Although these characteristics generally differ in any two time segments, the rate at which they change is often quite low, particularly for consecutive speech frames. Additionally, since real human speech is the result of a stable and linear production system, the mathematical model should, at least, be able to reflect this. These are the subjects of discussions in this section. The final topic in this section is concerned with the spectral sensitivity of each component in a LSF vector, which can be exploited for perceptually better performance.

4.1.1 Monotonicity Criterion

For stability and proper operation of the LPC synthesis filter (used in both the CELP and MB-LPC coders), a received and possibly corrupted LSF vector must observe the monotonicity criterion stated earlier in Section 2.2.3:

\[ 0 < \text{LSF}(0) < \text{LSF}(1) < \ldots < \text{LSF}(p-1) < 4000 \]  

where \( p \) is the LPC analysis order. A typical plot of the LPC spectral envelope with the corresponding LSFs is illustrated in Figure 4.1.

In the event whereby a received LSF element, \( \text{LSF}(i) \), has a value such that:

\[ \text{LSF}(i) + K > \text{LSF}(k), \quad \text{for} \quad 0 \leq i < k \leq p - 1 \]  

(4.2)
4.1 Properties of LSFs

Figure 4.1: A 10th order LPC spectral envelope with LSFs in a typical voiced speech segment.

then, a crossover is said to have occurred. $K$ is a constant which is used to ensure that no two LSF elements are too closely spaced to each other. Otherwise, tonal artefacts, such as whistles, can result in significant speech degradations. When crossovers are detected at the decoding end, a zero-redundancy algorithm must be deployed to remove them. The LSF monotonicity criterion is thus viewed, from the speech coder's requirements, as a source criterion in the speech synthesis process which must be observed at all times. Appropriate measures will then have to be initiated to correct or replace the offending LSF element(s). The issue of how these are performed are later presented in this chapter.

4.1.2 Inter-frame Correlations

The rate of change in the spectral envelope of speech is known to be relatively low as compared to the update rate of the speech encoding/decoding processes. Since most LBR speech coders produce a set of model parameters based on a fixed frame size, a speech segment of a length twice (or more) the frame size may have similar, if not identical, stationary characteristics. This, in turn, could result in consecutive speech parameters having highly correlated values, the most evident of which is manifested in the LPC spectral envelope. Consequently, smooth transitions of the spectral envelope, when plotted over
4.1 Properties of LSFs

time, result. This is illustrated in Figure 4.2

![Figure 4.2: Transitions of LPC spectral envelope and LSFs over consecutive frames.](image)

The intimate relationship of the speech spectral envelope, particularly in the regions around the formants, with its LSFs [45, 49] has been known for some time. Effective exploitation of the high inter-frame correlation of LSFs has been utilised in interpolation techniques for the LPC spectral envelope in speech synthesis. In terms of error control requisites for LSFs, it is feasible for a received vector to be reconstructed from previous frames if the current set is known to contain irrecoverable errors. The correlation of LSFs in consecutive frames can be statistically modelled using their differences:

\[ \Delta f(j, n) = \text{LSF}(j, n) - \text{LSF}(j, n - 1), \quad 0 \leq j \leq p - 1 \]  

(4.3)
where \( j \) is the LSF element order, and \( n \) is the speech frame index. The distribution of \( \Delta f \) is illustrated in Figure 4.3, with the deviations of \( \Delta f \) given in Table 4.1.

These observations enable not only the error control of LSFs to be relatively simple to implement, but also provide a good degree of transparency in the parameter recovery process. Perceptual enhancements can also be incorporated in the reconstructed LSF vector through simple bandwidth expansion, such as that used in the perceptual weighting process discussed in Section 2.3.2.3 (see page 24).
4.1 Properties of LSFs

<table>
<thead>
<tr>
<th>LSF</th>
<th>$\sigma_{\Delta f}$ (Hz)</th>
<th>LSF</th>
<th>$\sigma_{\Delta f}$ (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>44.2</td>
<td>5</td>
<td>137.3</td>
</tr>
<tr>
<td>1</td>
<td>101.9</td>
<td>6</td>
<td>114.2</td>
</tr>
<tr>
<td>2</td>
<td>148.4</td>
<td>7</td>
<td>115.0</td>
</tr>
<tr>
<td>3</td>
<td>147.6</td>
<td>8</td>
<td>109.2</td>
</tr>
<tr>
<td>4</td>
<td>135.9</td>
<td>9</td>
<td>93.6</td>
</tr>
</tbody>
</table>

Table 4.1: Deviations of inter-frame LSF differences.

4.1.3 Spectral Sensitivity

The quantisation and coding characteristics of the speech spectral envelope through the use of LSFs result in differing degrees of distortion ascribed to each element in the LSF vector. As depicted earlier in Figure 4.2 (on page 93), a shift in one LSF element can modify the spectral envelope more significantly than another LSF element which is shifted by the same amount. Thus, during quantisation, the different spectral sensitivities of each LSF element should be taken into account for better coding efficiency and transparency.

In selecting the optimal representation of the LSF vector, a cost function or distance measure is required. The weighted Euclidean distance measure is found to suit this purpose well:

$$d^2(f, \hat{f}) = \sum_{i=0}^{p-1} w_i (f_i - \hat{f}_i)^2$$  \hspace{1cm} (4.4)

where $f$ and $\hat{f}$ are the reference and the reproduction vectors respectively, and $w_i$ are the weighting factors which account for the spectral sensitivities of the LSF elements. The values of the weighting factors, $\{w_i \mid i = 0, 1, \ldots, p - 1\}$, can be obtained through various means. Many different approaches have been proposed, but only two of the most effective methods will be discussed here.

4.1.3.1 Group Delay

The phase response of the LPC filter can cause distortion of the synthesised speech signal if care is not taken to ensure that the group delays of each frequency component are
controlled. Consequently, group delays can be used to account for the specific spectral sensitivity of each frequency since they are known to be well related to the formant structure of the speech spectrum, that is, group delays are larger at formant frequencies [25]. The weighting factors can be computed as follows:

\[
w_i = \begin{cases} 
  u(f_i) \sqrt{\frac{D_i}{D_{\text{max}}}}, & 1.375 \leq D_i \leq D_{\text{max}} \\
  u(f_i) \frac{D_i}{\sqrt{1.375D_{\text{max}}}}, & D_i < 1.375
\end{cases}
\]  
(4.5)

where \(D_{\text{max}}\) is about 20 milliseconds, and

\[
u(f_i) = \begin{cases} 
  1, & f_i < 1000 \text{ (Hz)} \\
  1 - \frac{0.5}{1000} (f_i - 1000), & 1000 \leq f_i \leq 4000 \text{ (Hz)}
\end{cases}
\]  
(4.6)

are factors which account for the human perceptual insensitivity to high frequency quantisation inaccuracies. The group delays are given by:

\[
D(\omega) = (p + 1)T_s + \frac{2(VD - UC)}{V^2 + U^2}
\]  
(4.7)

\[
U = \sum_{j=0}^{p-1} \beta_j \sin(j\omega T_s)
\]  
(4.8)

\[
C = \sum_{j=0}^{p-1} T_s \beta_j \sin(j\omega T_s)
\]  
(4.9)

\[
D = \sum_{j=0}^{p-1} T_s \beta_j \cos(j\omega T_s)
\]  
(4.10)

\[
V = \sum_{j=0}^{p-1} 1 - \beta_j \cos(j\omega T_s)
\]  
(4.11)

with \(T_s\) being the sampling period, and \(\beta_j = -a_j\), the LPC coefficients.
4.2 Interpolation Methods for Corrupted LSFs

4.1.3.2 Power Spectrum

The tilt in the LPC spectral envelope means that lower frequency LSFs are associated with high energy sections of the power spectrum of speech, and vice versa. Thus, errors in quantising lower frequency LSFs are perceptually more noticeable than those in the higher frequency ranges. To take this phenomenon into account, the weighting factors, \( \{ w_i \mid i = 0, 1, \ldots, p - 1 \} \), are computed as follows [40]:

\[
w_i = [P(f_i)]^r
\]  

(4.13)

where \( r \) is an empirical constant obtained experimentally, and \( P(f) \) is the LPC power spectrum. In addition, since the human ear cannot resolve differences at high frequencies as accurately as at low frequencies, more weight is given to lower frequencies:

\[
c_i = \begin{cases} 
1.0, & \text{for } 0 \leq i \leq 7 \\
0.8, & \text{for } i = 8 \\
0.4, & \text{for } i = 9 
\end{cases}
\]  

(4.14)

for a LPC filter order of ten. The resulting weighted Euclidean distance measure thus becomes:

\[
d^2(\hat{f} - \hat{f}) = \sum_{i=0}^{p-1} [c_i w_i (f_i - \hat{f}_i)]^2
\]  

(4.15)

This formulation of distance measure has been reported in a 24-bit vector quantiser [40] which is capable of transparent coding of the LSFs.

4.2 Interpolation Methods for Corrupted LSFs

The stability of the LPC synthesis filter in both CELP and MB-LPC coders depends on the compliance of the minimum phase property, discussed previously in Sections 2.2.2 and 4.1.1. In circumstances whereby the received LSF vectors are corrupted to an extent such that
the monotonicity criterion is violated, parameter smoothing and/or replacement is deemed compulsory. In general, the corrupted LSFs can be stabilised by interpolating\(^1\) values from previous frame(s). The reliability of such simple techniques is quite high, owing to the high inter-frame correlations of LSFs. In the following, variants of such interpolation techniques are presented together with some remarks on the effectiveness of each technique.

4.2.1 Vector Interpolation

In this method, the frame to frame correlations of the LSF vectors are utilised in a very simplistic way. Given the knowledge that a received vector is unstable, the LSF vector is then simply replaced with a previous stable vector. In terms of implementation, the LPC coefficients are simply retained from the previous set, although they may be modified by broadening the speech formant bandwidths:

\[
a_i(n) = a_i(n - 1) \gamma^i, \quad 1 \leq i \leq p
\]  

(4.16)

where \(a_i\) are the LPC coefficients, \(\gamma\) is a constant (between 0 and 1, typically around 0.9) and \(n\) is the speech frame index.

The performance of such a technique is fairly acceptable during segments of voiced speech where the LPC spectral envelope does not change appreciably, or during unvoiced speech where the spectral envelope has no characteristic shape. However, if the corruption of the LSF vector coincides with transitional regions, that is, from unvoiced to unvoiced speech or vice versa, degradations resulting from mismatches between the interpolated vector and intended one can be significant and perceptually noticeable. Thus, the deployment of such an interpolation technique as a parameter replacement scheme is generally limited to lost frame recovery strategies, where the LSF vector has a high incidence of channel corruptions.

\(^1\)The intended action is extrapolation, but the term is left unchanged to avoid confusion about the origin of the methods.
4.2 Interpolation Methods for Corrupted LSFs

4.2.2 Pair-wise Interpolation

This method exploits the fact that when a LSF crossover is detected during non-drastic channel corruptions, the probability that only one LSF element is responsible for the instability is quite high. In such instances, it is feasible to interpolate the offending pair of LSFs from the corresponding values in the previous frame. The resulting algorithm thus becomes that of progressively checking of pairs of LSFs for monotonicity and replacing them if necessary. As implied in this method, the search for the actual corrupted LSF(s) is not catered for, and thus, is easy to implement.

![Diagram of paired-wise interpolation scheme for corrupted LSF(s).](image)

It should be noted that in situations where there is insufficient inter-frame correlation between the received and the previous LSF vector, this algorithm could reduce to that of vector interpolation. Hence, its effectiveness is also limited. In addition, since this method is initiated only when crossover(s) are detected, LSF corruptions which do not violate the criterion are treated as being received correctly. Consequently, degradations in the synthesised speech can still occur without this algorithm being invoked.

4.2.3 Single Element Interpolation

In the methods discussed so far, replacing the offending LSFs have the undesired but possible effect of changing other values which are not corrupted. This means that degradations in the output speech quality can still be perceived even though a corrupted LSF vector
has been stabilised. In the single LSF interpolation technique, the aim is to isolate the offending LSF in a crossover, and then only replace it. Thus, the remaining LSFs remain unaltered, thereby preserving as closely as possible the intended LPC spectral envelope.

The problem of stabilising LSF crossovers in single element interpolation techniques then becomes primarily that of determining which LSF is actually corrupted. Stabilisation of the culprit LSF can be effected by replacing it with a corresponding value in the previous frame(s). To determine which LSF is likely to be the corrupted one, the inter-frame and intra-frame correlations of LSFs are exploited, as in the following two methods.

**Formant Tracking**

In this method [50], reliance on inter-frame correlations is used. The premise in the method is that the LPC spectral envelope changes slowly such that the corresponding shifts in the LSFs are also small. Since errors incurred in the channel are independent of the LSF value being transmitted, uncharacteristic changes can therefore be used to isolate the offending LSF in a crossover.

The formant tracking method of locating the corrupted LSF element in a crossover can be stated as thus: given a received and previous LSF, \( \text{LSF}(i, n) \) and \( \text{LSF}(i, n - 1) \) respectively, the element most likely in error is that which satisfies:

\[
\max_j \left\{ \left| \text{LSF}(j, n) - \text{LSF}(j, n - 1) \right| \right\}, \quad \text{for } j = i, i - 1
\]

given \( \text{LSF}(i - 1, n) > \text{LSF}(i, n) - K \).

This method works quite well if the LPC spectral envelope does not change significantly in consecutive frames. Hence, its performance will degrade when LSFs are corrupted in frames during transitional regions of speech. In addition, the assumption that only one LSF is corrupted means that the formant tracking method will only operate well where a low level of corruption is expected.
4.2 Interpolation Methods for Corrupted LSFs

Frequency Spacing

This method [4] relies on the premise that the differences between adjacent LSF elements follow a statistical distribution which can be modelled by the algorithm, with \( \bar{d}(i) \) and \( \sigma(i) \) as the average and standard deviation of \( \text{LSF}(i + 1) - \text{LSF}(i) \) respectively.

The process of locating the destabilising LSF element is a series of test conditions as follows:

1. Define the following:

\[
d(i - 2) = |\text{LSF}(i - 1) - \text{LSF}(i - 2)|
\]

\[
d(i) = |\text{LSF}(i + 1) - \text{LSF}(i)|
\]

(4.18)

(4.19)

2. \( \text{LSF}(i - 1) \) is the offending LSF if:

\[
\Delta_l = |d(i - 2) - \bar{d}(i - 2)| > \sigma(i - 2)
\]

(4.20)

3. Else, \( \text{LSF}(i) \) is the destabilising element if:

\[
\Delta_h = |d(i) - \bar{d}(i)| > \sigma(i)
\]

(4.21)

4. If both the above tests fail, then label \( \text{LSF}(i - 2) \) as the corrupted element if:

\[
\Delta_l - \bar{d}(i - 2) > \Delta_h - \bar{d}(i)
\]

(4.22)

failing which, \( \text{LSF}(i) \) is then considered to be the corrupted element.

With the aid of the statistical properties of the LSFs, this method can, obviously,
perform better than the previous methods mentioned. However, the assumption that only one LSF is corrupted is still relied upon, thereby limiting the use of the frequency spacing method to channel conditions to that of relatively low BERs. In spite of this disadvantage, the frequency spacing method has demonstrated that source statistics can be used as an effective tool in assisting the detection and correction of corrupted LSFs. This is a very important observation which should be taken into consideration for effective advanced error control algorithms of LBR speech coder parameters.

### 4.3 Error Recovery Algorithm

In the methods discussed so far, the detection and “correction” of corrupted LSFs depended on one assumption – the channel BER is low (in the order of $10^{-3}$ or less), and therefore, the occurrence of two or more corrupted LSF elements in a received vector is negligible. Moreover, the effectiveness of these methods relied on the indication of one or more LSF crossovers, a situation which does not necessarily exist even if the received LSF vector is significantly corrupted. In the presence of high transmission errors, the quality of the synthesised speech will continue to degrade in spite of the utilisation of these ZR algorithms.

In circumstances where it is desirable to solely rely on a ZR scheme as a means of ensuring an acceptable level of speech quality, the interpolation techniques for LSFs discussed in the earlier sections may not provide an adequate performance. A more effective method, which can exploit the correlation properties of LSFs to a higher degree, is required. Evidently, source statistics for the LSFs will have to be used in conjunction with the inter-frame correlations observed in speech.

Scalar quantisers are used for coding the LSF elements for the LBR speech coders discussed in this thesis. This quantiser is organised as in Table 4.2. A total of 37 bits are required to code the LSFs for a high degree of transparency. Paliwal and Atal [40], however, have demonstrated that by using vector quantisation, only 24 bits are required for the coding of the LPC spectral envelope with high reproduction fidelity. Since this rate is
4.3 Error Recovery Algorithm

<table>
<thead>
<tr>
<th>Index</th>
<th>LSF Element Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>178 210 420 752 1041 1438 2005 2286 2775 3150</td>
</tr>
<tr>
<td>1</td>
<td>218 235 450 844 1174 1583 2115 2410 2908 3272</td>
</tr>
<tr>
<td>2</td>
<td>236 265 500 910 1274 1671 2176 2480 3000 3354</td>
</tr>
<tr>
<td>3</td>
<td>267 295 540 968 1340 1740 2222 2528 3086 3415</td>
</tr>
<tr>
<td>4</td>
<td>293 325 585 1016 1407 1804 2260 2574 3159 3473</td>
</tr>
<tr>
<td>5</td>
<td>332 360 640 1064 1466 1855 2297 2613 3234 3531</td>
</tr>
<tr>
<td>6</td>
<td>378 400 705 1110 1514 1905 2333 2650 3331 3580</td>
</tr>
<tr>
<td>7</td>
<td>420 440 775 1155 1559 1947 2365 2689 3453 3676</td>
</tr>
<tr>
<td>8</td>
<td>- 480 850 1202 1611 1988 2394 2723 - -</td>
</tr>
<tr>
<td>9</td>
<td>- 520 950 1249 1658 2034 2427 2758 - -</td>
</tr>
<tr>
<td>10</td>
<td>- 560 1050 1295 1714 2081 2463 2790 - -</td>
</tr>
<tr>
<td>11</td>
<td>- 610 1150 1349 1773 2135 2501 2830 - -</td>
</tr>
<tr>
<td>12</td>
<td>- 670 1250 1409 1834 2193 2551 2879 - -</td>
</tr>
<tr>
<td>13</td>
<td>- 740 1350 1498 1906 2267 2625 2957 - -</td>
</tr>
<tr>
<td>14</td>
<td>- 810 1450 1615 2008 2369 2728 3049 - -</td>
</tr>
<tr>
<td>15</td>
<td>- 880 1550 1808 2166 2476 2851 3197 - -</td>
</tr>
</tbody>
</table>

Table 4.2: Scalar quantiser for tenth-order LSF.

only a fraction of that in the scalar quantiser used here, it is obvious that some redundancy still exists in the quantiser used here, that is, not all combinations of the quantiser are actually used. In the improved error recovery algorithm for the LSFs described in the following, extensive use of this observed redundancy is exploited.

A suitably sized codebook, comprising LSF vectors trained using the LBG algorithm [31], is used as a list of “valid” code vectors. A received LSF vector can, thus, be matched with this codebook for validity, and hence, erroneous LSFs can be indicated. Conceptually, this is analogous to the decoding procedures in channel coding schemes, although the search process involved in this algorithm is exhaustive in nature. Viewed from another point, the redundancy resulting from the non-optimal usage of capacity in the scalar quantiser is being treated as a form of intrinsic “FEC”, which when fully exploited, can achieve robust transmission and reception of the LSFs without the use of extra transmission bandwidth.

As in channel coding schemes, a distance measure is required for the algorithm during the codebook search process. To quantify the spectral correlation in two LSF vectors, the algorithm uses the root-mean-squared (RMS) spectral distortion, $SD_{RMS}$ [28], as the
4.3 Error Recovery Algorithm

distance measure:

\[
SD_{RMS} \text{ (dB)} = \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} \left[ 10 \log|S(\omega)|^2 - 10 \log|\hat{S}(\omega)|^2 \right]^2 d\omega}
\]  (4.23)

where \( S(\omega) \) and \( \hat{S}(\omega) \) are the smoothed LPC spectra of the received and candidate LSF vectors respectively. In practice, obtaining \( SD_{RMS} \) through Equation 4.23 in its direct form is computationally demanding. An equivalent formulation is to use Cepstral Distance [10] (see Appendix C).

Error Recovery Algorithm [35]

(1) A first-in first-out (FIFO) buffer is used to store \( D \) previous LSF vectors, \( LSF(j,n-i), i = 1 \ldots D, j = 0 \ldots p-1 \), where \( p \) is the LPC order and \( D \) is the depth of the FIFO. The list of vectors in the FIFO will be used as a selection criterion for the codebook entry, and to flag suspected LSF elements.

(2) The received LSF vector is labelled as \( LSF(j,n) \). Excessive deviations of \( LSF(j,n) \), on an element by element basis, is then tested against those in the FIFO. That is, with \( d(j), j = 0 \ldots p-1 \), initially set to zero, the following is performed:

\[
d(j) \leftarrow d(j) + 1 \text{ if } |LSF(j,n) - LSF(j,n-i)| \geq \alpha \sigma(j)
\]  (4.24)

for \( \alpha = 3,4, i = 1 \ldots D, j = 0 \ldots p-1 \). \( \{\sigma(j) \mid j = 0 \ldots p-1\} \) is the standard deviation of LSF element \( j \) obtained from a large data set.

(3) Since the presence of LSF crossover(s) indicate the definite occurrence of error(s), subsequent computations must ignore the suspect LSF(s). This is performed by initialising a set of flags

\[
c(j) = 1, \ j = 0 \ldots p-1
\]  (4.25)

and then performing the following:
4.3 Error Recovery Algorithm

- if no crossover exists, that is \( \text{LSF}(j, n) > \text{LSF}(j - 1, n) \), \( j = 1 \ldots p - 1 \), then

\[
\begin{align*}
c(j) &= 1 \\
\end{align*}
\]

(4.26)

- otherwise,

\[
\begin{align*}
c(j) &= 0 \text{ if } d(j) \geq D \\
c(j - 1) &= 0 \text{ if } d(j - 1) \geq D \\
c(j) &= c(j - 1) = 0 \text{ otherwise}
\end{align*}
\]

(4.27) (4.28) (4.29)

(4) The algorithm then proceeds to associate the received LSF vector with a sub-set of codebook entries. This is done by shortlisting vectors whose weighted Euclidean distances are the lowest values, sorted in ascending order:

\[
PD_j = \sum_{i=0}^{p-1} c(i) w(i) \left[ C_j(i) - \text{LSF}(i, n) \right]^2 \quad j = 0 \ldots K - 1
\]

(4.30)

where \( \{C_j \mid j = 0 \ldots K - 1\} \) is the codebook of size \( K \), and \( w(i) \) are weighting factors obtained as in Section 4.1.3.1.

(5) For the final selection of the codebook vector from the shortlisted set, the spectral distortion \( SD_{RMS} \) is computed with all the vectors in the FIFO. Beginning with the first vector in the short-listed sub-set obtained in Step 4, a codebook vector, \( C_{best} \), is used to correct the received vector if:

\[
\left| \rho_{\text{codebook}}(k) - SD_{RMS}(C_{best}, \text{LSF}(n - i)) \right| \leq 2 \sigma_{\text{codebook}}(k), \quad i = 1 \ldots D
\]

(4.31)

where \( \rho_{\text{codebook}}(k) \) and \( \sigma_{\text{codebook}}(k) \) are the average and the standard deviation of the spectral distortion values associated with codebook vector, \( C_k \). If Equation 4.31 fails, the next code vector in the short-listed set is used and the step is repeated.

(6) Finally, the received LSF values, \( \text{LSF}(j, n), j = 0 \ldots p - 1 \), is replaced with \( C_{best}(j) \).
4.3 Error Recovery Algorithm

if \( c(j) = 0, j = 0 \ldots p - 1 \), that is, LSFs that were suspected to be corrupted. Bit toggling can be used to search for the desired value.

(7) Update the FIFO with LSF\((j, n)\) before proceeding to the next received LSF vector, LSF\((j, n + 1)\).

The algorithm is depicted in Figure 4.5.

Figure 4.5: Block diagram of the LSF Error Recovery Algorithm.

As implied in the procedures described, the algorithm neither assumes the presence of crossovers in order to flag a LSF element as possibly in error, nor does it attempt to isolate only one element. Hence, this algorithm is able to perform better than the simple LSF interpolation schemes, as will be shown later.
4.3 Error Recovery Algorithm

Performance

To see the effects the Error Recovery Algorithm has on corrupted LSFs, both objective and subjective tests were performed. The simulation conditions for obtaining the objective performance is illustrated in Figure 4.6. Simulations with a LSF database corrupted at different BERs were conducted, and the average spectral distortion, $SD_{RMS}$, was used to determine the degree of corruption experienced in the reconstructed LPC spectrum when compared to the original. $SD_{RMS}$ is computed as:

$$SD_{RMS} \ (dB) = \frac{1}{N} \sum_{n=0}^{N-1} SD_{RMS}(n)$$

(4.32)

where $N$ is the total number of frames evaluated. Table 4.3 shows the performance of the Error Recovery Algorithm (ERA), with that of the Formant Tracking (FT) technique (as discussed in Section 4.2.3) for comparisons.

In all cases with errors incurred in the LSFs, the Error Recovery Algorithm outperformed the Formant Tracking method quantitatively. Subjective listening tests, using an in-house 4300 bps CELP coder, have also correlated with these results, and this is illustrated in Figure 4.7. Although the objective performance of the Error Recovery Algorithm is not
Table 4.3: Performance of FT and ERA algorithms ($D = 2$, $K = 2048$).

<table>
<thead>
<tr>
<th>BER (%)</th>
<th>FT</th>
<th>ERA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.93</td>
<td>0.97</td>
</tr>
<tr>
<td>1</td>
<td>1.36</td>
<td>1.19</td>
</tr>
<tr>
<td>2</td>
<td>1.76</td>
<td>1.39</td>
</tr>
<tr>
<td>3</td>
<td>2.04</td>
<td>1.62</td>
</tr>
<tr>
<td>5</td>
<td>2.82</td>
<td>2.12</td>
</tr>
<tr>
<td>7</td>
<td>3.33</td>
<td>2.61</td>
</tr>
<tr>
<td>10</td>
<td>3.98</td>
<td>3.29</td>
</tr>
<tr>
<td>15</td>
<td>5.14</td>
<td>4.45</td>
</tr>
</tbody>
</table>

Significantly superior, listening tests have shown that the subjective quality is noticeably higher.

Figure 4.7: Pair-wise comparisons of subjective speech quality using FT and ERA techniques.
4.4 Remarks

In this chapter, non-FEC error control strategies for the linear prediction parameters in LPC-based speech coders have been presented. Being one of the most sensitive parameters to errors, the Line Spectral Frequencies require intervention, using these techniques, in order to maintain high quality speech reproduction in system applications. This is no less true in situations where protection from channel corruptions using FEC is not feasible or is costly in terms of system resources.

It can be seen that the LSFs possess many properties which can be exploited in zero-redundancy schemes. Various techniques for stabilising a corrupted LSF vector, and hence making them more robust to errors, have been outlined. Simple LSF interpolation techniques can be employed in channels with relatively low bit error rates. However, due to the simplistic assumptions made, inadequacy of these schemes is evident in noisy channels, resulting in speech artefacts which are perceptually noticeable.

To achieve an improved degree of error protection and a higher synthesised speech quality, a more complex but effective technique has been developed. The LSF Error Recovery Algorithm exploits correlations in the parameters as well as the coding redundancy in the scalar quantiser as a form of an intrinsic error control, but unfortunately demands some degree of implementation considerations. In spite of this, the technique helps to realise a more robust transmission environment for the spectral information in speech, and serves as an example of how the synergy of source and channel coding can contribute to the realisation of future, spectrally efficient speech coding systems.
Chapter Five

Source Aided Channel Coding

In digital voice communications utilising low bit rate speech coding algorithms, the output speech quality attainable is a function of, among other factors, the transmission bit rate and the degree of channel impairments in the transmission medium. While advances in speech coding techniques have been very successful in providing high quality speech reproduction at (ever) decreasing bit rates, efforts in combating channel noise in mobile radio environments have faced an uphill struggle. The task of transforming these transmission environments, using channel coding, interleaving, equalisation, etc., to that of desirable conditions is still, unfortunately, a highly elusive one.

To safeguard against the resulting uncertainties incurred in the received bit stream, techniques discussed in Chapter 3 are deemed necessary as means of ensuring an acceptable level of service quality. These techniques are not, however, without their shortcomings. In this chapter, a different approach to the problem is adopted. Here, it is argued that to provide the maximum level of service quality possible, the integrity of the coder parameters should be preserved using all means possible. To realise this objective, it might appear that such an aim would necessitate the use of more powerful FEC schemes, along with the associated bandwidth penalties. However, this need not be always the case, at least in the context of LBR speech coding applications (otherwise, the argument would have come full circle!). **Source Aided Channel Coding (SACC)** [36–38] is a technique developed to alleviate the demands on FEC in LBR speech coders, and aims to provide extra error...
correction power *without* the obligatory increase in coding redundancy.

A discussion on the limitations of zero redundancy schemes and their effectiveness is first presented in Section 5.1. A theoretical view of SACC is described next, following which block and trellis decoding methods, in conjunction with SACC, are outlined in Section 5.3. Some quantitative results, demonstrating the performance improvements achievable through application of this technique using the MB-LPC coder, are also given. Finally, some discussions on the applicability of SACC for CELP coders are presented before the chapter is concluded.

### 5.1 Limitations of Zero Redundancy Techniques

Zero redundancy (ZR) techniques for LBR speech coding algorithms have generally been accepted as accompanying measures to FEC coding schemes in curtailing the effects of channel errors. These techniques, discussed earlier in Section 3.5, can be classified under the following types:

1. pseudo-gray coding,
2. parameter substitution/Smoothing,
3. parameter de-sensitisation, and
4. lost frame recovery.

In all of the above-mentioned methods, the basic idea is common – to sustain the subjective speech quality without introducing annoying speech degradations or artefacts. In the operation of these ZR schemes, the choice of the parametric values selected for the speech synthesis process is often based on some loosely defined quality criteria, such as "subjectively annoying" or "perceptually noticeable". These criteria are difficult to qualify precisely in terms of the aural pleasantness (or rather, the lack of it), and tend to vary from person to person and speech conditions. This is further compounded by the fact that no
objective measure is capable of accurately quantifying these observations. As a reflection of this lack of consensus in the preference for the varying degrees of speech degradations, ZR schemes become inexact in their operations and choice of suitable parametric values.

It then follows that the synthesis process, with ZR schemes invoked at the decoder, will no longer be optimal since the reconstructed speech samples deviate from the actual ones. However, in doing so, the ZR algorithms ensure that the speech quality is not compromised to a degree such that it is subjectively unacceptable. In effect, the speech quality has been deliberately lowered in response to adverse channel conditions. This, ultimately, compromises the level of service quality.

In speech coding systems which are error-protected using a combination of FEC and ZR schemes, the rate of decrease in the voice quality is dependent on the effectiveness of the ZR algorithms since the FEC error correction capabilities are largely dictated by system specifications. It should be noted, however, that if the parameter “prediction” capabilities of ZR algorithms were to be enhanced considerably by some means, the quality of the synthesised speech would not degrade as rapidly.

### 5.2 Conceptual View

The lack of accuracy in choosing “good” parametric values for ZR algorithms can be resolved by using information from the FEC decoders. FEC decoding algorithms use a cost function or selection criterion in the codeword searching process. This criterion takes the form of a “generalised” Euclidean distance (Equation 3.25), for equiprobable codewords and no channel memory:

\[
d^2_i = \sum_{i=0}^{n-1} [r(i) - S_i(i)]^2, \quad 0 \leq r(i), S_i(i) \leq Q - 1
\]

For hard decision FEC decoding, that is, \( Q = 2 \), the above formulation becomes the Hamming distance, which is simply the number of bits which are different in the received
code sequence and the hypothesised codeword.

Therefore, all that remains is for the ZR algorithms to simply base the choice of a parametric value on a process which seeks to minimise the above criterion. In doing so, an agreement as to the appropriate choice of the value between the ZR algorithm and the FEC decoder would have been reached since the value would have been the "closest" both in terms of the FEC’s selection criterion and the LBR speech decoder’s source criteria. This procedure is depicted in Figure 5.1:

An equivalent but better approach would be to influence the FEC decoding algorithm with source criteria at the same time as it performs the searching process for the best codeword. This arrangement is more advantageous since codewords corresponding to parametric values which do not observe the LBR speech coder’s source criteria are eliminated by the FEC decoder, including those which are considered to be the best estimates of the received code sequence. Consequently, the FEC scheme’s error correction capabilities can be extended beyond that normally considered possible. In these circumstances, the performance of soft decision decoding in FEC coding schemes would have been enhanced further, thereby resulting in fewer decoding failures and lower residual BERs.
**5.2 Conceptual View**

### 5.2.1 Error Control Configuration

It was shown in Section 3.3 that the types of error control schemes for LBR speech coders rely on the intrinsic robustness of the coder parameters, ZR techniques, FEC coding schemes or a combination of all these. In the error control configuration designated Type 2 in this thesis, the speech coder’s source criteria influence only the ZR algorithm. The FEC decoder proceeds to obtain a code sequence without consideration of the source criteria’s constraints on the parametric values.

In SACC, this basic concept of letting the FEC decoder to decide “what is best” for the speech decoder independently is modified. Instead, the FEC decoding algorithm is forced to take into account and satisfy the appropriate source criteria of the speech coder parameters. This is illustrated in Figure 5.2. SACC, thus, represents an integral consideration of the speech and channel decoding processes in interpreting the received bit stream. In contrast, error control in conventional schemes, including those which have been standardised, are designed, implemented and optimised separately.

To qualify and support the claim that SACC can improve a FEC coding scheme’s error performance, an alternative view of how the use of soft decisions can enhance a block FEC coding scheme is necessary. Chase [7] has conceptualised the channel state information (CSI) from the channel demodulator as a set of confidence values accompanying the bits of a received code sequence. To obtain the most probable estimate of the received codeword, the channel decoding algorithm seeks to minimise a score which is equal to the sum of the confidence values in the error positions. Thus, the channel decoding algorithm is said to have an extra degree of freedom, in addition to that in the decoder, for selecting the best estimate of the received bit stream.

The use of channel state information in soft decision decoding accounts for the improvement of an FEC code’s capability to correct more errors than would usually be possible via hard decision decoding. In SACC, the LBR speech coder’s source criteria constitute another degree of freedom which the FEC decoder can further exploit (as depicted in
Figure 5.2: Type 2 error control configurations: (a) conventional, (b) SACC.

Figure 5.3. This has the effect of limiting the vector space which the FEC decoder has to search to obtain the best codeword and also include other vectors, which are beyond the soft decision decoding bounds but conform with the constraints of the source criteria. Consequently, the error correction power of the FEC code can be expected to be increased further.

5.2.2 Improved Maximum Likelihood Decoding

In searching for the most probable code sequence from an observed bit stream, FEC decoding algorithms seek to maximise the \textit{a posteriori} probability of the codeword hypothesised. In mathematical terms, the chosen codeword, \( u_j \), is given by:

\[
    u_j = u_k \quad \text{if} \quad \max_k \{ p(u_k \mid r) \} \quad \forall k
\]
where $u_k$ is a hypothesised vector, and $r$ is the observed code vector. Using Bayes' rule, the maximisation process in Equation 5.1 becomes:

$$
\max_k \left\{ p(u_k \mid r) \right\} \equiv \max_k \left\{ p(r \mid u_k) Pr(u_k) \right\}
$$

(5.2)

In typical FEC decoding techniques, an assumption is made for the a priori probabilities, $Pr(u_k)$, of the hypothesised code sequences – they are considered to be equally likely. This reduces Equation 5.1 to simply:

$$
u_j = u_k \quad \text{if} \quad \max_k \left\{ p(r \mid u_k) \right\}
$$

(5.3)

A FEC decoding algorithm which produces a code sequence, $u_j$, satisfying Equation 5.3 is said to be a Maximum Likelihood (ML) decoder.

The above formulation of obtaining the best estimate of the received code sequence has, at least in the context of LBR speech coding, a significant short-coming. It is well known that the coder parameters are seldom, if ever, equiprobable, as have been clearly depicted...
in the inter-frame correlations of the LSF vectors (see Figure 4.3, page 94). Additionally, a speech coder's source criteria have an intrinsic capability of predicting, in perceptually important speech conditions, the transmitted bit sequences. From this argument, it is hence apparent that the FEC decoding process should include the a priori distribution of the code vectors, \( Pr(u_k) \), as non-uniform for a true ML decoding performance.

In practice, obtaining the distribution \( Pr(u_k) \) is not straight forward since human speech, which the bit stream is ultimately dependent upon, is difficult to model precisely on a global basis, and is likely to vary with different speech databases and conditions. The approach taken here is to rely on the LBR speech coder's source criteria for indication of the validity of a hypothesised code vector, such as the monotonicity criterion of LSFs. To do this, it is necessary to map a candidate vector's bits to the appropriate coder parameter first, then referring to its relevant source criterion. This process is equivalent to equipping the FEC decoding algorithm with confidence flags which are indicative of the source criteria's decisions.

5.3 SACC in the MB-LPC Coder

The preceding sections in this chapter have, so far, referred to SACC without any mention of a specific LBR speech coding algorithm or FEC coding scheme. This concept is generic and system-independent, and therefore, is highly appealing and applicable to a large number of speech coding implementations. Both linear block and convolutional codes can be empowered with a LBR speech coder's source criteria for improved decoding performance. This section is devoted to explaining and illustrating how source criteria can be incorporated in the FEC decoding process.

Before these discussions proceed, it is necessary to first focus on the type and nature of the source criteria pivotal in SACC. Since this is coder- and implementation-specific, the Multi-Band Linear Predictive Coding (MB-LPC) vocoder will first be used as an example of a LBR speech coding system with which the advantages of SACC can be demonstrated.
Both qualitative and quantitative results will be used to consolidate the claims made for SACC. Discussions on the deployment of source criteria in SACC for the CELP coder are deferred to the following section.

5.3.1 Source Criteria in the MB-LPC Coder

The MB-LPC coder [52] produces, from a speech segment, four different parameters used for the reconstruction of high quality speech: LSFs, pitch, voicing decisions (V-UV) and excitation scaling factor. The speech analysis and synthesis details of this coder were given in Section 2.4.1. Subjective tests, conducted in order to assess the coder’s need of error protection strategies in noisy channel conditions, have indicated the relative degrees of error sensitivities. These are shown in Table 5.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Degree of subjective degradations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitch</td>
<td>High</td>
</tr>
<tr>
<td>LSFs</td>
<td>Noticeable</td>
</tr>
<tr>
<td>Energy</td>
<td>Moderate</td>
</tr>
<tr>
<td>V-UV</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 5.1: Sensitivities of MB-LPC parameters to corruptions.

It can be observed from the table that the most sensitive parameters are the LSFs and the pitch. On the contrary, the V-UV and the energy parameters have demonstrated remarkable robustness to channel errors. Therefore, only the LSFs and the pitch parameters’ source criteria will be of interests here.

LSFs

The monotonicity criterion of the LSFs, which guarantees the stability of the LPC synthesis filter, is a source criterion used in the MB-LPC coder. This has been extensively described in Sections 2.2.2 and 4.1.1. The primary concern with the LSFs is the compliance with the
5.3 SACC in the MB-LPC Coder

following relationship:

\[ \text{LSF}(i) + K < \text{LSF}(j), \quad 0 \leq i < j \leq p - 1 \]

where \( p \), the LPC analysis order, is ten in this implementation of the MB-LPC.

Pitch

The source criterion for the pitch parameter is similar to that described in Section 3.5.2. Essentially, this criterion ensures that the pitch tracking, which preserves pitch continuity in voiced speech regions, is observed:

\[ 0.8P(n - 1) \leq P(n) \leq 1.2P(n - 1) \]

Uncharacteristic and abrupt changes in pitch values have been known to be perceptually noticeable to a significant degree.

5.3.2 Block-based Decoding

The performance of a linear block code is fundamentally limited by its error correction capability:

\[ t = \left\lfloor \frac{d_{\text{min}} - 1}{2} \right\rfloor \]  \quad (5.4)

where \( t \) is the maximum number of bits (symbols) that are correctable by the code, and \( d_{\text{min}} \) is the minimum distance of the code. Soft decisions [7] can enhance the error correction capability by attempting to correct \( 2t \) errors, thereby reducing the code's output BER to a much lower value.

In digital mobile communication systems, the use of soft decisions in FEC decoding is often considered highly desirable, if not standard. Thus, in the design of an efficient speech and channel coding scheme, it is often assumed that the channel decoder is able to
cope with the majority of the channel corruptions incurred, that is, the maximum error correction capability of the code is often utilised. However, the code will not be able to cope with conditions where the number of errors in the received bit stream is more than $2t$. The ensuing consequence is that the errors received at the input of the FEC decoder will remain uncorrected and inevitably appear in the speech coder parameters, perhaps introducing even more erroneous bits. The use of source criteria should, in such circumstances, be able to detect such decoding failures and initiate appropriate action before the corrupted parameters are transferred to the speech decoder. This process is illustrated in Figure 5.4.

![Flow diagram of modified Chase algorithm for SACC.](image-url)

Figure 5.4: Flow diagram of modified Chase algorithm for SACC.

The primary difference between a soft decision channel decoding algorithm and that of SACC is that the criterion for selecting a potential codeword is preceded by manda-
5.3 SACC in the MB-LPC Coder

Tory satisfaction of the LBR speech coder's source criteria. The latter ensures that code sequences which are unlikely to be transmitted in the first place, are not considered as estimates of the received sequence. This creates the possibility of a scenario whereby no potential code sequence could be allowed as an estimate of the input sequence. For example, in the Golay (23,12,7) code, Chase's second algorithm [7] uses \( \left\lceil \frac{7}{2} \right\rceil = 3 \) least reliable bits for creating the list of error patterns. This list of 8 \( (= 2^3) \) test error patterns, for perturbing the input sequence, may produce decoding results which are all discarded by the modified decoding algorithm when more than 6 errors are present. Thus, the number of bits, which are considered most unreliable for generating the error patterns in the search list, can be increased to 4 or more. In SACC, the number of errors, \( t' \), that could possibly be corrected is then:

\[
2t < d_{\text{min}} \leq t'
\]  

(5.5)

With a larger list of test patterns produced from permuting more unreliable bits, the FEC decoding algorithm will be able to cope with a higher degree of channel impairments whilst conforming to the constraints imposed by the source criteria. Unfortunately, this further compounds the computation complexities already inherent in soft decision decoding of block codes, particularly non-binary ones such as Reed-Solomon codes, thereby putting an implementation penalty on the increased performance achievable. Thus, it is not unusual to find decoding algorithms based on table look-ups implemented in preference to the more computationally demanding techniques. This approach is actually more appealing as memories are cheaper and require less power than would a faster DSP to perform the same job.

Simulation Conditions

A shortened single-error-correcting Reed-Solomon (RS) (6,4) code over the finite field GF\( (2^3) \) was used to demonstrate the decoding advantage in SACC. The generator polyno-
5.3 SACC in the MB-LPC Coder

The polynomial for this RS code is:

\[ g(x) = \prod_{i=1}^{2} (x + \alpha^i) \]
\[ = (x + \alpha)(x + \alpha^2) \]
\[ = x^2 + \alpha^4x + \alpha^3 \]  

(5.6)

The encoding circuit for this code is as shown in Figure 5.5.

![Circuit for generating the parity checks in the RS(6,4) shortened code.](image)

The MB-LPC coder operates at a bit rate of 2950 bits per second (bps): of the 59 source bits, 48 are protected using four RS codewords, while the remaining bits are not protected. This results in a gross bit rate of the combined speech and channel coding system of 4150 bps. Tables 5.2 to 5.4 show the bit allocations of the MB-LPC coder bit stream.

Decoding Performance

An additive white Gaussian noise (AWGN) channel was used as a model for the propagation channel's characteristic, with the bits relayed using binary antipodal signalling. For operation in a mobile radio environment, interleaving of the RS codeword symbols would
be employed to break up bursts of errors. Simulations of various decoding algorithms are performed, as in the following:

(1) conventional hard decision (HD) decoding,
(2) SACC with hard decision (SACC-HD) decoding,
(3) conventional soft decision (SD) decoding, using Chase’s Algorithm 1, and
(4) SACC with soft decision (SACC-SD) decoding.

The various decoding performances are shown in Figure 5.6.
As depicted, the performances of the channel decoder under SACC have consistently outperformed that of the conventional schemes. At a post-decoding or residual BER of $10^{-3}$, an improvement in the coding gain of SACC-SD over that of the soft decision RS decoder was approximately 0.6 dB. It is also evident that the use of source criteria in the RS decoder has resulted in lower incorrect decoding rates. This, in turn, means that fewer speech frames are corrupted, thereby maintaining the synthesised speech quality instead of degrading it as in conventional schemes using ZR algorithms.

To assess the effectiveness of SACC on the subjective quality, in-house listening tests were conducted with speech corrupted at selected rates. Figure 5.7 shows that the speech synthesised with SACC is subjectively superior to that of conventional schemes. In fact, the SACC scheme was deemed to be qualitatively much better than indicated as much fewer speech artefacts have been perceived to be produced.
5.3 SACC in the MB-LPC Coder

5.3.3 Trellis-based Decoding

The implementation of block FEC coding schemes in SACC is, as observed in the preceding section, achieved in an indirect fashion – the bit sequences are decoded before being restricted by the source criteria. This results in a large number of decodings before the desired codeword is obtained, potentially exhausting the computation power available. To avoid this, use of block codes in FEC schemes are generally limited to those of either relatively short lengths and/or low error correction capabilities, such as Hamming codes. On the contrary, the implementation of SACC with trellis decoding algorithms is a much simpler and direct procedure than that for algebraic decoding methods. As a matter of fact, with the LBR speech coder’s source criteria imposing restrictions on the possible permutations of the parameters’ bit sequences, a reduced amount of searching is needed as
5.3 SACC in the MB-LPC Coder

compared to the “blind” approach taken in the conventional Viterbi algorithm (VA). This observation will be illustrated using examples later in this section.

The fundamental difference between the conventional and SACC-influenced trellis decoding algorithms is the number of nodes, in the code trellis, evaluated during the search process. While the sole aim is still to minimise the accumulated metrics of each trellis path arriving at a given node, SACC will precede the selection procedure of the trellis paths by interpreting and subjecting the associated input conditions to the appropriate source criteria. To illustrate this, examples are described in the following, using the source criteria of the LSFs and the pitch parameters in the MB-LPC coder.

LSFs

Consider the decoding branches from node 1112 (binary) in a channel coding scheme using a (2,1,3) convolutional code (see Figure 5.8). Assume that the input bits needed to arrive at this node corresponds to the bitmap sequence for LSF element 8 (which is 3453 Hz, see Table 4.2, page 103). Since the next LSF element (LSF 9) is coded using three bits, attempting to deduce the next 3-bit sequence using the VA in a conventional manner will mean that all possible terminating nodes for LSF element 9 will be searched. In the SACC procedure, the only values of LSF element 9 which observes the LSF monotonicity criterion are 3531 (index 5), 3580 (index 6) and 3676 Hz (index 7). Thus the decoding algorithm need only search three terminating nodes. An acceleration of the computation speed of the decoding algorithm, as a side-effect, may also be achieved. The profile of the trellis searches of these two methods are depicted in Figure 5.8.

It should be noted that the criterion enforced for the LSFs in the above example is a “global” one, that is, it is to be observed irrespective of the speech conditions. Consequently, if the trellis path profiles for the entire LSF vector in SACC are to be compared with that of the conventional VA, a substantial reduction in the computation complexity is apparent.
If the current synthesis process is in the midst of voiced speech regions, then the value of the current pitch parameter is bounded by that received previously (as discussed in Section 5.3.1). As an example, assume a value of 35.5 samples for the pitch in the previous frame. Then the bitmap of the current pitch value is limited to between to the following two values, inclusively:

\[
P_{\text{high}} = \left\lfloor 2 \times (1.2 \times 35.5) - 29 \right\rfloor
\]
\[
= 56
\]
\[
= 00111000_2
\]

\[
P_{\text{low}} = \left\lfloor 2 \times (0.8 \times 35.5) - 29 \right\rfloor
\]
\[
= 27
\]
\[
= 00011011_2
\]

Figure 5.9 shows the paths searched in the code trellis, enumerated for bit permutations between the two bitmaps. Although the number of paths evaluated is significantly reduced,
5.3 SACC in the MB-LPC Coder

Figure 5.9: Reduced trellis search for pitch parameter bitmap (voiced speech detected, previous pitch value = 35.5).

the decoding procedure is still optimal in the context of SACC. The source criterion for the pitch parameter is, however, dependent on the state of the output speech (voiced signals with high energy, or unvoiced with low energy). This, unlike the LSFs, does not guarantee a reduction in the computations for all speech frames.

In using the source criterion for either the LSFs or the pitch, the trellis decoding algorithm in SACC never searches nor evaluates bit streams that cannot possibly occur at the transmitting speech coder. In essence, this means that the number of codewords has been reduced to reflect the non-uniformity of the \textit{a priori} probabilities, $P_r(u_k)$ (see Equation 5.3). Hence, the channel decoding can result in improved decoding performance, as has been expected and rigorously explained in Section 5.2.2.

Simulation Conditions

To illustrate the enhanced performance of a convolutional code implemented under SACC, the MB-LPC speech coder was used at a rate of 2850 bits per second (bps). The allocation and order of transmission for the speech coder's bits are shown in Table 5.5.

A punctured convolutional code was used to error-protect the MB-LPC coder bit
5.3 SACC in the MB-LPC Coder

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Number of Bits</th>
<th>Transmission Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitch</td>
<td>8</td>
<td>First</td>
</tr>
<tr>
<td>LSFs</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>V-UV</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Energy</td>
<td>5</td>
<td>Last</td>
</tr>
<tr>
<td>Total</td>
<td>57</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Bit allocations and order of transmission of MB-LPC coder parameters.

Stream. The generators for the (2,1,7) code, punctured to a rate of 2/3, are:

\[
g_1(X) = 1 + X + X^3 + X^4 + X^5 + X^6 + X^7
\] (5.7)

\[
g_2(X) = 1 + X^2 + X^4 + X^7
\] (5.8)

(commonly specified as (337, 251, 337) in octal).

![Encoder circuit for the (2,1,7) convolutional code, punctured to a rate of 2/3](image)

Figure 5.10: Encoder circuit for the (2,1,7) convolutional code, punctured to a rate of 2/3 (Clark and Cain, page 403).

The encoding circuit for this code is shown in Figure 5.10. The output bits totalled 96 (= (3/2) \times (57 + 7)) for each speech frame of duration 20 milliseconds. This results in a combined speech and channel codec operating at 4800 bps, a bit rate which some LBR speech codecs have been standardised at (such as the DoD CELP [24] in 1989), or are in the process of standardisation (for example, the recently announced call for an improved INMARSAT Standard M speech and channel codec, dubbed "mini-M"). Simulations have been performed using four decoding approaches:
(1) conventional VA with hard decisions (HD),
(2) SACC with hard decisions (SACC-HD),
(3) conventional VA with eight-level quantised soft decisions (SD), and
(4) SACC with eight-level quantised soft decisions (SACC-SD).

Decoding Performance

The signalling scheme and the propagation channel were similar to that used earlier. The decoding performances of the four schemes simulated are shown in Figure 5.11.

![Graph showing performance comparisons of conventional VA decoding schemes and SACC schemes.]

Figure 5.11: Performance comparisons of conventional VA decoding schemes and SACC schemes.

Figure 5.11 clearly shows the performance improvements of the reduced-search decoding algorithm under SACC over that of conventional VA methods. At residual BERs above $10^{-3}$, SACC has reduced the error rates to about half that of the conventional VA schemes.
The higher error correction capability of the convolutional code is also evident from the rate of decoding failures, shown in Figure 5.12. As was also noted in the reduced BERs achieved by SACC, the lower number of speech frames with decoding failures means that the speech synthesis process is able to proceed with the correct coder parameters, thereby sustaining speech at the highest quality possible.

5.4 SACC in the CELP Coder

The CELP coder [42] and its variants are low bit rate speech coding techniques that are widely used. Many mobile communication systems, such as USA’s IS54 and Japan’s JDC
use coders which are derived from the basic structure of CELP. Source aided channel coding, discussed so far with reference to the MB-LPC coder, can also be incorporated into these CELP-based speech and channel coding systems. In these section, discussions on the source criteria of these LBR speech coders are presented. In particular, emphasis will only be placed on identifying the constraints that can be used to impose on the channel decoding algorithm for improved performance. No specific channel coding scheme will be used since the implementation can be used or easily modified from those mentioned in the earlier sections.

**LSFs**

Both the CELP and the MB-LPC speech coding algorithms use linear prediction as part of their speech modelling processes. The LPC filter analysis and synthesis procedures are identical, although vector quantisation (VQ) may be used in place of scalar quantisation, the latter more commonly used in view of implementation simplicity and speed. Thus, the focus will be on scalar quantisation of the transmission parameters of the LPC filter, the LSFs. It may seem that SACC is not applicable to VQ-ed LSFs. But for practical and realistic implementations, split VQ [40] is often preferable as the computation complexity for a full VQ process can be too high to be cost-effective. This, in turn, means that some source redundancy will still exist for SACC to take advantage of.

The nature and implementation of the source criteria of the LSFs into the FEC decoding process is very much a similar process as was discussed in Section 5.3. Hence, this will not be further mentioned.

**Codebook Gains**

As noted previously in Section 3.5.2, the codebook excitations (also termed as stochastic or vector excitations) have gain values which are highly correlated. This property is depicted in Figure 3.19 (see page 82), and has been exploited as a source criterion for use in the
adaptive gain smoothing algorithm in the AUDETEL speech codec [46].

An alternative way of viewing this property is to consider the occurrences of code sequences whose components are the bitmaps of the quantised codebook gains. For instance, the codebook gains are coded using a 5-bit quantiser [48]: 1 bit for the sign (positive or negative scaling), and the remaining 4 bits for the magnitude. Ignoring the sign bit, a CELP coder with four sub-blocks will produce a 16-bit code sequence (per frame) as one of the parameters. If the quantiser for each codebook gain is organised as that in Table 5.6, then compliance with the source criterion will mean that the simultaneous occurrence of the binary bitmaps 0000 (value = 1.0) and 1111 (value = 998.0), is a highly unlikely or even non-existent event. Observations made on the utilisation of codebook indices, on a frame by frame basis, revealed that a significant fraction of the total number of possible combinations is not used. This is depicted in Figure 5.13.

It, thus, becomes a simple matter for the source criterion in SACC to bias the channel decoding algorithm to reflect this observed characteristic. For block-based decoding algorithms, all that is needed is to perform checks on the validity of the decoded codewords against a list of “legal” sequences. In trellis-based decoding, the same list can be used or modified to search the code trellis of the convolutional (or block) codes.

<table>
<thead>
<tr>
<th>Magnitude</th>
<th>Bitmap</th>
<th>Magnitude</th>
<th>Bitmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0000</td>
<td>168.0</td>
<td>1000</td>
</tr>
<tr>
<td>3.0</td>
<td>0001</td>
<td>209.0</td>
<td>1001</td>
</tr>
<tr>
<td>10.0</td>
<td>0010</td>
<td>255.0</td>
<td>1010</td>
</tr>
<tr>
<td>27.0</td>
<td>0011</td>
<td>314.0</td>
<td>1011</td>
</tr>
<tr>
<td>48.0</td>
<td>0100</td>
<td>390.0</td>
<td>1100</td>
</tr>
<tr>
<td>74.0</td>
<td>0101</td>
<td>495.0</td>
<td>1101</td>
</tr>
<tr>
<td>102.0</td>
<td>0110</td>
<td>653.0</td>
<td>1110</td>
</tr>
<tr>
<td>134.0</td>
<td>0111</td>
<td>998.0</td>
<td>1111</td>
</tr>
</tbody>
</table>

Table 5.6: Quantiser table for codebook magnitudes in CELP [48].
Pitch Gains

The pitch gains of the long term predictor (LTP) in CELP indicates the degree of similarities of the speech samples between two given sub-blocks. In highly voiced speech regions, the speech characteristics (periodicity, amplitude) in adjacent sub-blocks may correlate to such a degree that their corresponding pitch gains tend to be of magnitudes near unity. That is, they also exhibit a clustering characteristic similar to that observed in the codebook gains in CELP. This is exemplified in Figure 5.14.

To exploit the correlations in the pitch gains as a source criterion, the presence of voiced speech regions must first be detected. This may be provided by using an energy threshold similar to that used for the MB-LPC vocoder (see Section 3.5.2). With indications about the state of the output synthesised speech in the last frame, the possibilities in the pitch gain, for the sub-block immediately following, can be constrained to a pre-determined range. However, the same cannot be said of the other pitch gain values, since they depend on each other in a successive manner, the only exception being the presence of significantly voiced speech. The region marked “A” in Figure 5.14 is an example of such situation. Thus, the scope of the criterion for the pitch gains is rather limited, but may still help to
Figure 5.14: Relationship of pitch gains with highly voiced speech regions.

Pitch Lag Values

As with the pitch gains, correlations in the pitch lags in the CELP coder are emphasised in the presence of voiced speech segments (as in Figure 5.14). In these subjectively prominent regions, the values of the pitch lags, $p(n)$, are often related to each other as follows:

$$p(n) = \frac{a}{b} p(n + 1) \pm p_d \text{ for } n = 0 \ldots N_s - 1$$  \hspace{1cm} (5.9)

where $N_s$ is the number of sub-blocks in the CELP coder, $a$ and $b$ are constants in the range \{1, 2, 3\}, and $p_d$ is a constant. Essentially, Equation 5.9 simply states that two pitch lag values are more or less harmonically related (multiple, sub-multiple, and multiple of sub-multiple) during speech regions containing periodic energy. The search for the pitch lag values can be quite extensive since the number of possibilities implied by Equation 5.9
is numerous. Thus, the implementation of the source criterion for the pitch lag values in CELP can be considerably expensive in terms of computation complexity. However, the successful use of this criterion can potentially reduce “roughness” in the speech synthesised with corrupted values.

5.5 Remarks

Source aided channel coding represents an effort in integrating the channel coding’s selection criteria with that of the speech coder’s source constraints in order to attain a higher level of error performance. This approach is distinctively different from that adopted in conventional schemes where errors in the speech coder parameters are concealed through zero redundancy schemes. In extending the FEC coding schemes’ error capabilities to beyond what they can normally offer, SACC can help, at least in part, to realise the implementation of high quality, spectrally efficient and robust low bit rate speech coders.

The conceptual view of SACC has been presented in two different but related directions:

1. As an additional degree of freedom presented to the channel decoding algorithm to isolate and disregard impossible or unlikely code sequences hypothesised, and

2. The use of parameter prediction capabilities in LBR speech coders’ source criteria as forms of a priori probabilities for the hypothesised code sequences, which are unexploited in Maximum Likelihood decoding methods.

The advantages of SACC have been demonstrated using both linear block and convolutional channel coding schemes with a speech coder. Quantitative results, obtained for a (6,4) shortened Reed-Solomon code and a (2,1,7) punctured convolutional code as FEC schemes for the MB-LPC coder, have shown that the error correction capabilities have exceeded the usual bounds experienced in both hard and soft decision decoding methods. The post-decoding bit error rates of the channel decoder obtained have been consistently reduced,
resulting in improved speech quality synthesised under noisy channel conditions.

The most significant advantage, which SACC contributes to the synergy of source and channel coding techniques, is its adaptability for potentially widespread applications in low bit rate speech coding systems. Many mobile radio communication systems can exploit the increased efficiency and quality of voice service possible, and the adoption of SACC does not necessitate any modification to the systems' specifications. These implementations include Europe's half-rate GSM candidates, USA's IS54, Japan's JDC, and the INMARSAT Standard M speech and channel coding algorithms, to name a few.
Chapter Six

Lost Frame Recovery

Mobile speech communications for the masses, using digital radio techniques, have ushered in many difficulties in the implementation issues not commonly encountered in digital transmission systems. One such problem that has to be addressed is the diverse characteristics of the radio propagation channel, which have sparked off a large amount of research interests in advanced modulation schemes, channel equalisation methods and access techniques (such as frequency hopping) [47]. The aim in these techniques is identical with that posed in error control for speech coders - to ensure a minimum acceptable level of service quality within the system constraints of transmission power (operating life of batteries in mobile units and co-channel interference), and spectral efficiency (bits per second per Hertz).

It is obvious that to cater for all scenarios possible in the propagation channel, one would require a significant degree of redundancy in the system. This is not feasible in terms of application economies and resources, and thus some engineering compromise is inevitable. This results in undesirable operating aspects, for instance, temporary loss of synchronisations or frames during heavy signal fadings or attenuations. For LBR speech coders, circumstances like this must be expected of since the end-to-end voice quality is determined largely by the speech synthesised. This aspect of error control is thus deemed crucial in the implementation of mobile communication systems, where voice services are expected to be the dominant offerings.
In this chapter, discussions on techniques for ensuring an acceptable subjective quality of the synthesised speech, during such irrecoverable error conditions, are presented. Section 6.1 explains the need of Lost Frame Recovery (LFR) schemes, while Section 6.2 delves into the underlying principles of such approaches. The following sections describe the various techniques used in LFR schemes before the chapter is concluded.

6.1 Motivation

Forward error correction (FEC) coding of the digital representations of speech signals is capable of ensuring a low output error rate (BER) with which a high level of voice quality can be attained. Effectively, the transmission channel has been transformed to one in which information can be relayed accurately and confidently within the given limits. The performance of an error-protected speech coding scheme is, however, heavily dependent on the actual mobile radio propagation conditions, the latter being capable of deteriorating to levels of corruptions which are beyond the expectations of FEC schemes. Tables 6.1 and 6.2 give some illustrations of what such error conditions can be expected in real systems.

Given such constraints in which a practical speech and channel codec for a mobile communication system is to operate, it can be seen that no reasonable degree of FEC protection can counteract the stipulated channel conditions in both GSM's and the Standard M’s specifications. More importantly, the speech synthesised from corrupted parameters during such circumstances is completely unintelligible and could potentially contain artefacts that are subjectively annoying to a significant extent. It is, thus, obvious that a approach different from those presented in the previous chapters is necessary in order to ensure an overall minimum acceptable speech quality.

The solution in these circumstances is to use LFR strategies to augment forward error correction coding and other zero redundancy techniques. The combination of these approaches, as depicted in Figure 6.1 (page 141), will be able to form a complete error control sub-system in a speech and channel codec destined for digital mobile radio applications.
6.2 Residual Redundancy in Source Parameters

<table>
<thead>
<tr>
<th>Average BER (%)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP0 0.0</td>
<td>Clear channel</td>
</tr>
<tr>
<td>EP1 4.5</td>
<td>C/I = 10 dB</td>
</tr>
<tr>
<td>EP2 8.3</td>
<td>C/I = 7 dB</td>
</tr>
<tr>
<td>EP3 13.4</td>
<td>C/I = 4 dB</td>
</tr>
</tbody>
</table>

*C/I* = Carrier to co-channel interference ratio

Table 6.1: Channel error conditions for GSM’s half rate speech and channel codec.

<table>
<thead>
<tr>
<th>Channel conditions</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>$10^{-3}$ to $4 \times 10^{-2}$</td>
</tr>
<tr>
<td>Bursty</td>
<td>504 bits error-free</td>
</tr>
<tr>
<td></td>
<td>72 bits corrupted</td>
</tr>
<tr>
<td>4-state Markov model</td>
<td>$9.29 \times 10^{-4}$ to $1.69 \times 10^{-1}$</td>
</tr>
</tbody>
</table>

Table 6.2: Channel error conditions for the new INMARSAT Standard M codec (version 3).

It should be noted that the quality of the synthesised speech using LFR algorithms is intrinsically lower than that achieved normally. The key concerns in the design and implementation of LFR algorithms are:

1. the effectiveness of the speech recovery process in LFR techniques, and
2. the impact on the speech quality with LFR techniques invoked at different rates.

These issues will be the subjects of discussions in the following sections in this chapter, with reference to the MB-LPC vocoder.

6.2 Residual Redundancy in Source Parameters

In circumstances where speech frames are erased, the error control sub-system is presented with the problem of reconstructing parametric data with which the LBR speech coder
6.2 Residual Redundancy in Source Parameters

Figure 6.1: System block of a unit in mobile communications.

can use to synthesise speech. In general, replacing such corrupted data can be difficult since the human perceptual characteristics and information content, implied by the model parameters and their combinations, are very important. Fortunately, LFR schemes can exploit the residual redundancy and correlations in the coder parameters, and speech characteristics to assist the speech recovery process. In the following, some insight as to how such observations can provide such aid are discussed.

6.2.1 Parameter Repetition

In the majority of model-based LBR speech coding algorithms, a fixed duration of the input speech signal to be analysed and coded is used. For example, the CELP and its variants use speech frame lengths which are between 20 and 30 milliseconds. The MB-LPC coder, used extensively in this thesis, operates on speech using 160 samples (20 milliseconds, sampling rate of 8kHz) at any single instance, but can easily extend to 240 samples if necessary, provided that the constraints on the buffering delay are not compromised.
A speech waveform, corresponding to a single utterance of a word or phoneme, is typically longer in length than the frame size (irrespective of the language used). In highly voiced speech regions, the waveform samples are repetitive in nature. Segmentation of these samples into fixed-size frames for analysis will inevitably mean that the modelling processes in LBR speech coders are likely to be very similar, if not identical. This observation is best illustrated in Figure 6.2.

![Figure 6.2: Voiced speech region (partial) for the word “landing” segmented into MB-LPC coder frames.](image)

Having observed that the processing nature of LBR speech coding techniques does not totally eliminate human speech redundancies, a straightforward way of predicting a set of coder parameters for an erased frame is to repeat them from those in the adjacent frames. For example, for the MBE vocoder and its variants, an erased pitch value is reconstructed as:

\[ P(n) = P(n - 1) \] (6.1)
In fact, during regions of highly voiced speech as illustrated earlier in Figure 6.2, this approach is very effective and is often preferred over other methods for the reconstruction of the speech waveform, such as muting.

### 6.2.2 Parameter Dynamics

In addition to the highly repetitive nature observed in speech waveforms, LBR speech coder parameters also inherit characteristics that are the result of the gradual speech production processes in humans - clustering of parameter values, as previously detailed in Section 5.4. Coder parameters reflecting the “loudness” or magnitude of the speech waveforms have tendencies to exhibit such a phenomena. Figure 6.3 shows the clustering characteristic of the excitation gain factors, \( \sigma(n) \), in the MB-LPC vocoder, plotted using their inter-frame differences, \( \delta(n) \):

\[
\delta(n) = \sigma(n - 1) - \sigma(n)
\]

Figure 6.3: Inter-frame correlation reflected in differences, \( \delta(n) \), of the MB-LPC’s excitation gain values.

Thus, the value of \( \sigma(n) \) of a given frame is often equal or close to that in the previ-
ous frame. For such parameters, the simple technique of copying values from the previous frame(s) has to be modified in order to account for the dynamics of speech. In addition, accommodations for subjective enhancements of the reconstructed speech can also be catered for. For the MBE vocoder (and its variants), the reconstruction of the excitation scaling parameter, $\sigma(n)$, is found to be very effective by using the following formulation:

$$
\sigma(n) = \beta \sigma(n - 1)
$$

(6.3)

where $\beta$ is a constant between 0 and 0.9, typically about 0.6. The same approach can also be used for the codebook gain magnitudes in CELP coders. The upper limit of $\beta$ is deliberately kept below unity such that occurrences of excessive surges in the synthesised speech energy is kept to a minimum, which otherwise will cause significant discomfort to the listener. This is especially important in LBR speech coding algorithms which rely significantly on feedback prediction, as in CELP coders.

### 6.3 Frame Recovery in the MB-LPC Coder

In the preceding sections, it was demonstrated how LFR strategies are essential for actual LBR speech coder applications, such as mobile radio telephony. In general, given the circumstances whereby system constraints have already dictated the type of the speech coding algorithm and FEC protection scheme to be used, it is still possible to design and implement a corresponding LFR scheme optimised for the given system. Therefore, different approaches for treating such conditions in different LBR speech coders are inevitable, although similarities may exist.

In this section, discussions on LFR techniques are restricted to the MB-LPC vocoder only. Table 6.3 illustrates the relative sensitivities of each coder parameters (identical to the one listed earlier). It should be noted that LFR strategies also utilise the speech coder's source criteria where possible.
Table 6.3: MB-LPC parameter sensitivities.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Subjective Degradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSFs</td>
<td>Noticeable</td>
</tr>
<tr>
<td>Pitch</td>
<td>High</td>
</tr>
<tr>
<td>Energy</td>
<td>Moderate</td>
</tr>
<tr>
<td>V-UV</td>
<td>Low</td>
</tr>
</tbody>
</table>

LSFs

The reconstruction of a LSF vector in an erased speech frame is fundamentally similar to that of Vector Interpolation, as was already discussed in Section 4.2.1. This method of recovering the coded speech’s spectral envelope is only fairly effective as the subjective quality of the recovered speech is relatively sensitive to discrepancies in the reconstructed and the original LPC spectral envelope. Bandwidth broadening should also be employed since any changes in the formants can be incorporated by the recovered weighted spectrum. Figure 6.4 shows an instance of the LPC spectrum recovery process.

Figure 6.4: LPC spectrum reconstruction in LFR process.
6.3 Frame Recovery in the MB-LPC Coder

Pitch

The pitch in MBE-based vocoders may be the most perceptually important parameter in the speech analysis synthesis process, but replacing an erased frame’s pitch value is relatively straightforward. Except in instances where frame erasures coincide with the built-up transitions of speech energy, the simple process of using the previous frame’s pitch value for use in the erased frame does not significantly degrade the synthesised speech quality. This is primarily due to the fact that periodicity in voiced speech waveforms, where the degree of perception and intelligibility is high, follows a smooth and unabrupt contour (see Figure 3.20, page 85). This has already been observed and exploited in the MBE’s pitch estimation process. Therefore, by simply repeating the pitch parameter value from the previous frame, that is, \( P(n) = P(n-1) \), little degradations in the quality can be perceived. At this stage, it is worthwhile noting that although a recovered pitch value may be perceptually but not numerically identical to the intended value, objective measures can easily fail to reflect this.

In the cases where the encoded speech mainly consists of either unvoiced energy or silence, the pitch does not contribute significantly to the synthesis process since it is effectively disregarded by the states in the voicing decisions (see Section 2.4.6). Hence, in these regions, the actual pitch value is irrelevant in LFR strategies in MBE-type vocoders.

Table 6.4 lists down the degree of degradations perceived in the synthesised speech due to discrepancies in the recovered pitch parameter in relation to the speech signal regions.

<table>
<thead>
<tr>
<th>Speech regions</th>
<th>Degradation in quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silence</td>
<td>None</td>
</tr>
<tr>
<td>Unvoiced</td>
<td>Low</td>
</tr>
<tr>
<td>Voiced</td>
<td>Moderate to low</td>
</tr>
<tr>
<td>Unvoiced → Voiced</td>
<td>High</td>
</tr>
<tr>
<td>Voiced → Unvoiced</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 6.4: Effects of recovered pitch in erased frames on speech quality.
6.3 Frame Recovery in the MB-LPC Coder

Excitation Factor

The volume or "loudness" of the speech being encoded is approximated by the excitation scaling factor (or gain) in the MB-LPC vocoder. Consequently, it can be expected that grossly incorrect excitation values used at the decoding end can result in potentially annoying outbursts of the synthesised speech. Although this may mean that the integrity of the excitation gain is significant, subjective listening experiments have indicated otherwise, particularly when only the three least significant bits are corrupted (a total of five bits are used to quantise the excitation gain). This is primarily due to the lack of dependence on feedback prediction in the speech analysis and synthesis process, and thus minimising the problem and effects of error propagation. This results in a comparatively more robust coder than CELP-type algorithms (the latter has been known to produce loud clicks and annoying bangs, an important aspect since the artefacts are potentially very annoying).

In addition, large deviations in the values of adjacent excitation gains are uncharacteristic of real speech, as was earlier discussed in Section 6.2.2. Thus, an erased frame's excitation gain, \( \sigma(n) \), in the MB-LPC coder can be obtained as in Equation 6.2.2:

\[
\sigma(n) = \beta \sigma(n-1)
\]

where \( \beta \) is set to 0.6. With this value, consecutively erased frames will result in rapidly decaying speech signal amplitudes. The decreasing values of \( \sigma(n) \) ensure that, for consecutively erased speech frames, the output synthesised energy decays gradually.

Voicing Decisions

The V-UV parameter is also very insensitive to channel corruptions, as has been indicated by the degradations in the output speech quality at high errors rates. In the LFR process, hence, accurate recovery of the voicing decisions is not critically important, and the simple repetition technique, used for the pitch parameter, can be employed as well. However, parameter de-sensitisation (see Section 3.5.3) should be used in order to prevent excessive
periodic energy, due to differences in the recovered and actual V-UV decisions, being inadvertently introduced in the output speech. Otherwise, speech artefacts such as breathing noises and buzziness may be perceived (potentially making the speakers sound asthmatic).

The recovery procedure for the MB-LPC coder's V-UV parameter consists of two stages - copying the decisions from the previous frame, and setting those corresponding to higher frequencies to unvoiced:

\[
VUV(n, i) = \begin{cases} 
VUV(n - 1, i) & \text{for } 0 \leq i \leq 4, \\
0 & \text{for } 5 \leq i \leq 8.
\end{cases}
\]  

(6.4)

where \(n\) is the speech frame index.

A modification to the above process, for speech regions which are just unvoiced or silence, is possible. In these circumstances where the excitation gain, \(u(n)\), is known to be unreliable as well, the entire V-UV parameter can be de-sensitised, that is, the V-UV decisions are all set to unvoiced, irrespective of the recovered value. This process requires the detection of such speech segments, and the latter is easily provided for by using a threshold on the output synthesised speech energy, \(E(n)\):

\[
E(n) < \varepsilon \bar{E}(n - 1)
\]

(6.5)

where \(\bar{E}(n)\) is a running average of the output speech energy, and \(\varepsilon\) is a constant which is experimentally optimised (around or less than 0.1) for best results. This technique is similar to that used for the source criterion for the pitch parameter, as discussed in Section 3.5.2 (see page 84).

### 6.4 Quality Performance

To ascertain the effectiveness of the LFR strategy for the MB-LPC vocoder, subjective listening tests were performed. The quality of the synthesised speech, at various lost
frames rates, was assessed using the MOS grading system since objective forms of quality measures fail to correlate sufficiently with the perceived quality. The MOS impairment scale (see Table 6.5) was used instead of the quality grading scheme since impairments were expected in the first place, thus making the study more informative by analysing the degree of annoyance. However, the quality of the recovered speech, as indicated by the MOS scoring system, is relative to the clear channel speech quality of the MB-LPC vocoder.

<table>
<thead>
<tr>
<th>MOS Grade</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Not noticeable</td>
</tr>
<tr>
<td>4</td>
<td>Noticeable, but not annoying</td>
</tr>
<tr>
<td>3</td>
<td>Slightly annoying</td>
</tr>
<tr>
<td>2</td>
<td>Annoying</td>
</tr>
<tr>
<td>1</td>
<td>Very annoying</td>
</tr>
</tbody>
</table>

Table 6.5: The MOS impairment grading system used in the subjective tests.

![Figure 6.5: Subjective quality of speech synthesised by the MB-LPC vocoder at various lost frame rates.]

As can be observed in Figure 6.5, the MB-LPC vocoder was able to synthesise speech
with an acceptable level of speech quality even at random lost frames rate of 14.2%. At
24.8%, the amount of speech artefacts began to degrade the speech quality by a significant
degree. At the same time, no loud explosions of the synthesised speech energy was heard,
as might be expected in CELP-based coders if particular care is not taken. Therefore, it
is reasonable to say that the performance of the MB-LPC vocoder is still acceptable even
at lost rates of up to 14%. This consolidates the observations that the MBE-based speech
coding algorithms are extremely robust under harsh propagation conditions (where more
than 10% loss rate is considered harsh).

The maximum lost frame rate of 14% deduced can be used as a useful design guide-
line in LBR speech communications – the amount of error protection using FEC can be
determined, and the level of service quality can be estimated. It is the author’s opinion
that the inclusion of LFR strategies for all LBR speech coders, particularly those which
are operating in a radio environment, be considered as crucial and a key role in the error
control problem.

6.5 Remarks

Lost frame recovery (LFR) strategies for LBR speech coding algorithms represent the last
“defence line” against the effects of corruptions incurred in the transmission channel. The
aim of these techniques is to ensure a predictable, yet acceptable level of service quality in
real applications operating in a wide range of propagation conditions.

In this chapter, the MB-LPC vocoder was used as the base system with which LFR
strategies were discussed. The use of human perceptual characteristics and residual re-
dundancies, which are ever present in the speech coder parameters, are exploited in the
analysis and formulation of LFR techniques. The effectiveness of the recovery scheme
for each individual speech coder parameter was also described in detail, with subjective
enhancements (for example, parameter de-sentisation) introduced in the recovery process
wherever feasible.
The effects of the parameter recovery processes and the rates at which such techniques are invoked are characterised through subjective listening tests. The MOS impairments scale was used to quantify the level of synthesised speech quality. The tests conducted on the MB-LPC vocoder indicated an acceptable level of speech quality at lost frame rates as high as 14%, with degradations rapidly becoming annoying when a higher incidence of frame recovery processes was invoked. Such indications of the speech coder's behaviour are very useful in the design stages of a LBR speech communication system, since the factors which affect the service quality, including channel coding requirements, can thus be deduced before the implementation stages. The same observation also serves to underline the need for an integrated approach in the conceptual stages of a LBR speech coding algorithm.
Chapter Seven

Conclusions

Innovations, technological advances and the availability of spectral resources have enabled mobile communications to become reality. With high expectations on the service quality, efficiency and cost-effectiveness, it is no wonder that researchers have been very successful at devising and improving techniques used in speech coding sciences. However, the impacts of channel corruptions on these LBR speech coders have only aroused some interests in the past, even when the threat to service degradations is indisputable. Only in recent times have the error control aspects and requirements of LBR speech coders have begun to receive more attention and spawn dedicated research activities. The subjects discussed in this thesis attest to the growing significance and importance of advanced error control techniques for LBR speech coders.

Throughout this thesis, emphasis has been directed at real and practical research in preference over more theoretical approaches. The adoption of such an "engineering" attitude, in the investigation and formulation of error control methodologies, originates from the desire to involve real speech coding systems and issues in their real-life applications. Consequently, in the presentation for most part of the research work detailed in this thesis, there is a disposition to the implementation and application of error control techniques and algorithms in LBR speech coding systems.

In general, this thesis had concentrated on related but different aspects of error control in LBR speech coding systems — substitution techniques for corrupted coder parameters;
enhancing the decoding capabilities of FEC codes by exploiting source criteria, and recovery schemes for erased or lost speech frames. All these methods' capabilities are based on residual redundancy in the speech coder's parametric information and the constraints imposed on by the speech modelling process.

The following section gives a chapter by chapter summary of the key views and findings presented in this thesis. Finally, some ideas and recommendations for continued involvement in error control for low bit rate speech coders are given, with the hope that these, together with existing and newly proposed techniques, will represent pivotal concepts in the management of speech quality in contemporary and future digital communication systems.

7.1 Concluding Overview

Digital speech coding techniques have come a long way since the widespread adoption of PCM. In Chapter 2, the characteristics of digital communication systems and their constraints on the design of modern speech coding methodologies were presented. A fundamental concept in model-based speech coding, linear prediction, was detailed. Together with descriptions of two low bit rate speech coders, the CELP and the MB-LPC, an insight as to how more effective error control strategies can be formulated was derived. The author contends that such knowledge is not just necessary, but vital in devising both efficient and effective error control methods.

In Chapter 3, the problems faced in mobile communication systems were outlined. In particular, constraints on bandwidth and transmission power, coupled with harsh propagation conditions, present a dilemma where the quality of service has to be balanced against speech robustness. A detailed list of considerations, crucial in the design of combined speech and channel coding strategies, was drawn up and are as follows:

(1) error sensitivities of coder parameters,
(2) degree of confidence of the source criteria of a parameter,
(3) parameter’s responsiveness to error concealment,
(4) error propagation and its effects on the speech synthesis process, and
(5) parameter quantisation schemes.

In addition, schemes for abating the effects of channel corruptions on the synthesised speech, using both system and parameter orientated approaches, were also discussed.

The problem of ensuring subjectively robust transmission of the Line Spectral Frequencies was considered in Chapter 4. This is especially important as this parameter is used to encode one of human speech’s most perceptually sensitive features, the LPC spectral envelope. This is in addition to the fact that many low bit rate speech coders, including the two used in this thesis, are LPC-based. An algorithm, designed to exploit both quantisation and source redundancies, was detailed. Subjectively better performance over other approaches was obtained, giving proof that source statistics should be utilised for additional performance gains when designing parameter substitution techniques.

The conventional way of exploiting source criteria in the speech coder parameters was modified significantly through Source Aided Channel Coding. Chapter 5 was devoted to the concept, design, implementation and performance quantification of SACC. This technique is generic and can potentially benefit many LBR speech and channel coding systems. SACC was shown to enhance the FEC’s decoding performance, thereby improving the quality of the synthesised speech. As much as 50% of the decoding failures in conventional schemes could have been avoided through this technique. Both block- and trellis-based FEC codes have been used in demonstrating the performance enhancements, which is especially valuable considering the fact that no additional redundancy is needed. SACC represents, in the author’s view, a central concept in the synergy of source and channel coding for low bit rate speech communications, since the criteria from both the channel (soft decisions) and the speech coder (source criteria) are taken advantage of simultaneously.

Chapter 6 investigates the issues encountered by the speech coder in attempting to recover speech during conditions when the entire speech frame is unreliable or lost. Lost
frame recovery techniques for the MB-LPC coder were described, along with the observations which make these possible in the first place. These are deemed highly necessary in mobile communications as the average bit error rates can rise to levels beyond the correction capabilities of practical FEC schemes.

In summing up the above, the following statements can be made:

(a) With the restrictions placed on spectrum and transmission power, error control in LBR speech and channel codecs need a combined approach in ensuring a high level of service quality at low transmission rates. Although there are on-going research in attempting to improve the level of service via the radio interface, the author believes that the real control over the speech quality lies at the speech and channel codec.

(b) Optimal performance of error control schemes require in-depth understanding of the speech coding algorithm used, and the parameters’ characteristics and interrelationships. This ensures the use of minimum amount of resources for maximum throughput.

(c) The use of source statistics is vital in the design of zero-redundancy algorithms for best subjective performance. Reliance on error protection using FEC codes can thus be alleviated, possibly by a significant extent. This represents the ideal way for mitigating channel errors without the use of FEC coding schemes.

(d) There is scope for higher levels of optimisation in the exploitation of source criteria in LBR speech coders. This thesis has shown that it is possible, in conjunction with soft decisions from the channel demodulator, to achieve higher error correction rates with source criteria taking an active part in the FEC decoding algorithm.

(e) Lost frame recovery strategies form an integral part of error control in LBR speech coders. This ensures that a large range of transmission conditions is catered for, and that the speech coder is behaving in a more controlled manner.

(f) Robustness to channel corruptions represents an essential feature in the design and
implementation of LBR speech coders. Effective error control, consequently, now constitutes one of the priorities in the codec specifications, as evident in the standardisation processes for the GSM and INMARSAT Standard M systems.

(g) There will be continued interests in integrated strategies for speech and channel coding applications, such as the proposed communication networks for the masses, for example, Personal Communication Systems (PCS) and Future Public Land Mobile Telecommunication Systems (FPLMTS), where high service quality, efficiency and low resource requirements are emphasised.

7.2 Future Work

The achievements of the research work in this thesis have the potential of making more contributions to the error control problem in low bit rate speech communications. Additionally, it is foreseeable that various other approaches are capable of providing similar breakthroughs. In general, these improvements and areas for further investigations are:

(a) The implementation of SACC here is restricted to one specific speech coding algorithm, namely the MB-LPC vocoder. Although the concept is generic, the application of SACC is susceptible to variations in the specifics of the coding techniques in processing and coding speech. Therefore, there is scope for investigating the benefits SACC can bring to "popular" speech coders such as CELP and VSELP.

(b) SACC exploits the speech coder parameter redundancies and constraints through the use of source criteria. The latter do not include the probabilistic modelling of speech and the corresponding model parameters, though this presents an additional dimension which could be capitalised on. Thus, SACC could be extended to incorporate, for instance, Hidden Markov models (HMM), for increased decoding performance.

(c) The use of powerful FEC codes during periods of low channel corruptions represent a loss of coding efficiency, whereas those with lower correction capabilities are unable
to cope with more adverse propagation conditions. It is envisaged that for an optimal balance between the two extreme scenarios, adaptive channel coding should be employed. In particular, less critical or sensitive bits from the speech coder parameters could be used to provide the signalling (if necessary) and redundancy needed for switching between two (or more) FEC codes with different coding rates.

(d) The coding redundancy in scalar quantisers for LSFs could be optimised for error control. Code representations of valid combinations of this parameter can be organised to provide maximum “distances” between them, similar to the way FEC codes are constructed. Also, the same concept could be applied to other parameters. In this way, the reliance on FEC protection could be substantially reduced since the coded parameters have an intrinsic capability to detect and/or correct errors.
List Of Appendices

A  Publications

B  Vector Gain Control In The AUDETEL Speech Coder

C  Log Spectral Distance Computation
Appendix A

Publications


Appendix B

Vector Gain Control In The AUDETEL Speech Coder

The Pulsed Residual Excited Linear Prediction (PRELP) speech coder is inherently robust under low bit error rates of up to $10^{-3}$. However, very disturbing clicks or bangs in the synthesised speech may result if the error rate is increased to cause some of the excitation vector gains to be decoded with very large magnitude errors. In order to reduce the effect of large excitation vector gain errors, the following built-in excitation vector gain control algorithm is used.

For the very first time, initialise magnitude of last vector gain for the previous frame $p_v^4$ and a long-term average of the standard deviation of the differences $\tilde{\sigma}_d(n-1)$ as below:

$$ p_v^4 = \mu, \quad \tilde{\sigma}_d(n-1) = \nu $$

(B.1)

where $\mu$ and $\nu$ are constants. Then, for every subsequent speech frame, follow the steps given below:

1. Hamming decode all the four vectors gains in the current speech frame, while keeping a count of the corrupted gains, that is, error correction(s) is(are) needed.
2. Taking $pv_4$ into consideration, compute the average of the five vector gains:

$$
\bar{g} = \frac{1}{5} \sum_{i=0}^{4} |g(i)|
$$

(B.2)

where $|g(0)| = pv_4$.

3. Compute the absolute deviation $\delta(i)$ for each gain magnitude

$$
\delta(i) = \left| |g(i)| - \bar{g} \right| \text{ for } i = 0, 1, ..., 4
$$

(B.3)

4. Compute the average of the deviations $\bar{\delta}$,

$$
\bar{\delta} = \frac{1}{5} \sum_{i=0}^{4} \delta(i)
$$

(B.4)

5. Compute the standard deviation (SD) of the deviations:

$$
\sigma_d = \sqrt{\frac{\sum_{i=0}^{4} \delta^2(i)}{5} - \bar{\delta}^2}
$$

(B.5)

6. Compute a long-term running average of $\sigma_d$ as

$$
\bar{\sigma}_d(n) = \gamma \sigma_d + (1 - \gamma)\bar{\sigma}_d(n - 1)
$$

(B.6)

where $\gamma$ takes on a typical value of 0.1.

7. During Hamming decoding, check if there are any errors incurred in the channel. If no errors are detected then

$$
\bar{\sigma}_d(n - 1) = \bar{\sigma}_d(n)
$$

(B.7)

$$
pv_4 = |g(4)|
$$

(B.8)

and EXIT the vector gain control process. Else continue to the next step.
8. Isolate the vector gain, $g(j)$, which gives the largest deviation, $\delta(j)$. Perform the following test:

$$|\delta(j) - \bar{\delta}| < \beta \bar{\sigma}_d(n)$$

(B.9)

where $\beta$ is a constant, and $j$ is an integer ranging between between 1 and 4. If the above test succeeds, then

$$\bar{\sigma}_d(n-1) = \bar{\sigma}_d(n) \text{ and } pv_4 = |g(4)|$$

(B.10)

and EXIT. Else, continue to the next step.

9. Toggle bit, $b_1$, of the encoded value of the suspected gain $g(j)$. Then decode the resulting value, $g_m(j)$, and obtain its magnitude.

10. Then compute the following:

$$g_m = \frac{1}{4}(5\bar{g} - |g_m(j)|)$$

(B.11)

$$\delta(i) = |g(i) - \bar{g}_m| \text{ for all } i \neq j$$

(B.12)

$$\bar{\delta} = \frac{1}{4} \sum_{i \neq j} \delta(i)$$

(B.13)

$$\delta^2(i)$$

$$\sigma^2_d = \frac{1}{4} \sum_{i \neq j} \delta^2(i) - \bar{\delta}^2$$

(B.14)

$$\bar{\sigma}_d(n) = \gamma \sigma_d + (1 - \gamma)\bar{\sigma}_d(n-1)$$

(B.15)

11. If $|\delta(i) - \bar{\delta}| < \beta \bar{\sigma}_d(n)$, then

$$g(j) = g_m(j) \text{ and } \bar{\sigma}_d(n-1) = \bar{\sigma}_d(n)$$

(B.16)

and go to Step (14), else continue.

12. Toggle the next more significant bit of encoded value of $g(j)$ and repeat Steps (10) and (11).
13. If toggling of all the four bits has been tried and yet the criterion given in Step (11) is not satisfied, then

\[ |g(j)| = \frac{1}{2} |g(j - 1)| + \frac{1}{2} |g(j + 1)| \]

(B.17)

for \( j = 1 \ldots 3 \), and for \( j = 4, |g(j)| = \bar{g}_m \). The sign of \( g(j) \) is preserved.

\[ \bar{\sigma}_d(n - 1) = \bar{\sigma}_d(n) \]

(B.18)

14. Now compute the following,

\[ \bar{g}_m = \frac{1}{5} \sum_{i=0}^{4} |g(i)| \]

(B.19)

\[ \delta(i) = | |g(i)| - \bar{g}_m| \quad \text{for } i = 0, 1 \ldots 4 \]

(B.20)

\[ \bar{\delta}(n) = \frac{1}{5} \sum_{i=0}^{4} \delta(i) \]

(B.21)

\[ \sigma_d = \sqrt{\frac{\sum_{i=0}^{4} \delta^2(i)}{5} - \bar{\delta}^2} \]

(B.22)

\[ \bar{\sigma}_d(n) = \gamma \sigma_d + (1 - \gamma) \bar{\sigma}_d(n - 1) \]

(B.23)

\[ \bar{\sigma}_d(n - 1) = \bar{\sigma}_d(n) \]

(B.25)

\[ p_{w4} = |g(4)| \]

(B.26)

15. Isolate the vector gain, \( g(k) \), now resulting in the largest deviation, \( \delta(k) \). If \( |\delta(k) - \bar{\delta}| < \beta \bar{\sigma}_d(n) \), then,

and EXIT. Else continue to the next step.

16. Toggle bit \( b_1 \) of the encoded value of the suspected gain, \( g(k) \), and decode the resulting value, \( g_m(k) \).

17. Compute new value of \( \bar{g}_m, \delta(i), \bar{\delta}, \sigma_d \) and \( \bar{\sigma}_d(n) \) using Equations (B.11) to (B.15).
18. If $|\delta(k) - \bar{\delta}| < \beta \bar{\sigma}_d(n)$, then

\[ g(k) = g_m(k) \quad (B.27) \]
\[ \bar{\sigma}_d(n - 1) = \bar{\sigma}_d(n) \quad (B.28) \]
\[ pv_4 = |g(4)| \quad (B.29) \]

and EXIT, else continue.

19. Toggle the next most significant bit of the encoded value, and recompute $g_m$, $\delta_i$, $\bar{\delta}$, $\sigma_d$ and $\bar{\sigma}_d(n)$, and repeat Step (18).

20. If toggling of all the four bits has been tried and yet the criterion given in Step (18) is no still satisfied, then

\[ |g(k)| = \frac{1}{2}|g(k - 1)| + \frac{1}{2}|g(k + 1)| \quad (B.30) \]

for all $k$ between 1 and 3, and for $k = 4$, $|g(k)| = g_m$ and,

\[ \bar{\sigma}_d(n - 1) = \bar{\sigma}_d(n) \quad (B.31) \]
\[ pv_4 = |g(4)| \quad (B.32) \]

and EXIT.
Appendix C

Log Spectral Distance Computation

The cepstrum, \(\{c_k^x\}\), is the inverse Fourier transform of the logarithm power spectrum of a discrete time series, \(x(n)\) [10]:

\[
c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |X(\omega)|^2 e^{j\omega k} \, d\omega
\]  

(C.1)

where \(c_k^x\) denotes the \(k\)th cepstral coefficient of the signal, \(x(n)\). Equivalently, by duality:

\[
\ln |X(\omega)|^2 = \sum_{k=-\infty}^{\infty} c_k e^{-j\omega k}
\]  

(C.2)

\[
= c_0 + 2 \sum_{k=1}^{\infty} c_k \cos(\omega k)
\]  

(C.3)

The log spectral distance between two signals, \(x(n)\) and \(y(n)\), is defined as:

\[
SD = \left[ \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| 10 \log |X(\omega)|^2 - 10 \log |Y(\omega)|^2 \right|^2 \right]^{1/2}
\]  

(C.4)

\[
= \left[ \frac{K^2}{2\pi} \int_{-\pi}^{\pi} \left| \ln |X(\omega)|^2 - \ln |Y(\omega)|^2 \right|^2 \right]^{1/2}
\]  

(C.5)

where \(K = 10/\ln(10) \approx 4.343\).
Substituting Equation C.3 into Equation C.5:

\[
SD = K \left[ \sum_{k=-\infty}^{\infty} (c_k^x - c_k^y)^2 \right]^{\frac{1}{2}}
\]

\[
= K \left[ (c_0^x - c_0^y)^2 + 2 \sum_{k=1}^{\infty} (c_k^x - c_k^y)^2 \right]^{\frac{1}{2}}
\]

(C.6)

(C.7)

The upper limit of the summation in Equation C.7 need not be infinity, but instead can replaced by a value large enough to give satisfactory results. Hence, the log spectral distance can be reduced to:

\[
SD = K \left[ (c_0^x - c_0^y)^2 + 2 \sum_{k=1}^{N} (c_k^x - c_k^y)^2 \right]^{\frac{1}{2}}
\]

where \( N \) is usually greater than or equal to 50.

The cepstral coefficients, \( c_k^x \), are computed from the LPC filter coefficients, \( a_i \), by the following equations:

\[
c_k^x = -a_1
\]

\[
c_k^x = -a_k - \sum_{i=1}^{k-1} \frac{k-i}{k} c_{k-i} a_i \quad \text{for } k = 2, \ldots, p
\]

\[
c_k^x = -\sum_{i=1}^{p} \frac{k-i}{k} c_{k-i} a_i \quad \text{for } k = p+1, \ldots, N
\]

where \( p \) is the LPC filter order. \( c_0^x \) is equal to the energy of the signal \( x(n) \) (given by zero-lag autocorrelation),
Bibliography

- P. Papamichalis, “Practical Approaches to Speech Coding”, Prentice-Hall, 1987


References


