SPEECH CODING AT MEDIUM BIT RATES

USING ANALYSIS BY SYNTHESIS TECHNIQUES

by

Nick Gouvianakis

A Doctoral Thesis submitted in partial fulfilment

of the requirements for the award of Doctor of Philosophy

of the Loughborough University of Technology

February 1989

SUPERVISOR : Professor Costas Xydeas

ABSTRACT

The need for more powerful speech compression techniques is becoming greater as more and more applications for digital voice storage and transmission are coming into use. The demands on the digital speech compression systems have been greatly increased, as the synthetic speech quality previously obtained from low-bit-rate coders is inadequate for most current applications. Bringing toll-quality speech capabilities to systems operating at medium transmission bit rates, is an area where major research activity is now focused at.

Hybrid coders which bridge the gap between vocoders and waveform coders, and can produce high quality narrowband speech at bit rates between 4 kbits/sec and 16 kbits/sec, are investigated in this thesis.

The Multipulse Excitation (MPE) model is first considered, which provides for a more detailed description of the excitation in a LPC coder, while recognising the importance of "perceptually shaping" the distortion introduced by the coding process. The excitation signal is modeled as a sequence of irregularly spaced pulses, whose positions and amplitudes are determined by a MPE optimisation algorithm. A general classification of MPE optimisation algorithms is attempted, by considering both high-complexity and low-complexity algorithms. Two classes of algorithms that can be included in real time speech coding implementations, are further examined and their performance/complexity characteristics are compared. Furthermore, a model for the distribution of the pulse amplitudes is developed, that can be used to design "globally" optimum amplitude-quantizers.

A new MPE coder is proposed, which employs a codebook of pulse-position patterns and can achieve a much more efficient coding of the pulse positions than conventional MPE coders. A fast algorithm is developed that can be used to design the pulse-position codebook, by maximising a measure of the coder's performance.

A novel scheme, called Backward Excitation Recovery coder, is finally described, which reconstructs the excitation signal of a LPC coder using a backward adaptation procedure. As a result, only the parameters of the filter need to be transmitted to the decoder (receiver) in order to recover the speech signal. Many BER coding schemes are developed and their performance/complexity characteristics are compared. One BER coder in particular, defines the excitation from the past synthesised samples of the speech signal, and can optimally operate with very small delays of the order of 3 ms. Finally, a procedure is developed that "designs" vector-quantizers for the coefficients of the BER synthesis filter, by optimising the coder's performance.

ACKNOWLEDGEMENTS

I wish to thank my supervisor Professor C.Xydeas for his support, guidance and friendship. It has been a real pleasure working with him.

The financial support of Group RT5224 of British Telecom Research Laboratories, is greatfully acknowledged by the author.

Finally I wish to thank my parents for their encouragement and understanding.

CONTENTS

CHAPTER 4 (MULTI-STAGE AND BLOCK-SEARCH OPTIMISATION METHODS)

CHAPTER 5 (CODEBOOK SEARCH MPE CODING)

CHAPTER 6 (BACKWARD EXCITATION RECOVERY CODING)

CHAPTER 1

1.1 Introduction

Speech communication over long distances was first made possible over 100 years ago, with the invention of the telephone. The telephone network has since expanded massively and has reached the remotest areas of the planet. Modern communication networks carry speech, image and data signals, and may involve various transmission media such as optical fibers, UHF radio and satellites.

In order to use a transmission channel efficiently, the input signal (source) must be first converted into a suitable form. The conversion (coding) is usually performed in two stages. The first stage (source coding) uses the knowledge of the source characteristics to produce a "compact" digital signal representation. The second stage (channel coding) is concerned with the properties of the transmission medium, and converts the outcome of the first stage into a signal which can be reliably recovered, i.e. "decoded" at the other end of the transmission channel [1.1].

Digital source-coding techniques are applied to voice signals after their conversion into a digital format using Analog to Digital Conversion (ADC). Digital Signal Processing (DSP) algorithms are then employed which "compress" the digitised speech signal prior to transmission. At the receiver, an inverse procedure is used to "decompress" and thus recover the speech signal [1.2,1.3,1.4]. The coding process must incur a minimal loss of speech quality.

Recent advances in microelectronics have made possible the implementation of complex DSP algorithms using a small number of VLSI circuits [1.5,1.6]. Compared to analog processing, digital coding of speech offers improved reliability, lower development and implementation costs, and easier application of secure encryption techniques. As a result, new voice application areas have emerged [1.7,1.8] and the process of replacing the analog technology, used in existing systems, with new digital technology, has accelerated considerably.

Digital Mobile Radio Communications is recent and important application area where highly efficient speech coding methods must be employed. These coding methods should offer maximum compression of the speech signal in

order to fully exploit the limited bandwidth available, and should allow substantial protection against the effects of adverse transmission conditions [1.9,1.10]. Furthermore, despite the relatively small transmission bandwidth, the quality of the recovered speech must be high and thus acceptable to the mobile radio user [1.11].

Efficient speech coding methods are also used when optimal "loading" of existing communication networks is required. Modern communication networks carry thousands of speech and image channels, and it is therefore very important to allocate minimum bandwidth to each channel. Optimal "loading" is especially important in satellite communications [1.12], where the cost of transmission per bandwidth unit can be substantial.

Speech synthesis and speech storage are areas directly related to speech coding, and they also rely on efficient speech "compression" techniques. Various models of the speech signal which provide good results when employed to form the front end of voice synthesis systems, have been successfully used in the digital coding and transmission of speech [1.13]. On the other hand, speech coding algorithms have been used in commercially available speech synthesis and speech storage systems [1.7]. Application areas for speech synthesis and speech storage include electronic mail by voice, voice-operated database inquiry systems, voice store-and-forward systems, text-to-speech synthesis, aids to people with impaired speech or hearing, and many others in office automation and multimedia environments.

Transmission of very good quality digitised speech over the Public Switched Telephone Network (PSTN) is now possible but requires the use of expensive modems. The proposed Integrated Services Digital Network (ISDN) will in the future provide a digital end-to-end connectivity between terminals, and a standard Digital Terminal Interface [1.14,1.15]. The system will be flexible enough to accomodate Circuit Switched and Packet Switched transmission, while maintaining a transparency with respect to the user. Speech, video and computer data bit streams will be indistinguishable when transmitted using one or more channels provided by the ISDN. Speech coding algorithms will be used in each terminal to convert the voice signal into a digital bit stream for transmission. Speech coding systems will therefore form an important part of future public or private digital communication networks.

Speech compression techniques are becoming increasingly sophisticated and complex in response to an ever growing demand for cost efficient use of the existing transmission and storage media. Furthermore, it seems that the availability of enormous bandwidth in optical fiber transmission networks and the decreasing price of the computer memory chips have not managed to stop the increasing research effort on new speech coding methods that can operate at very low bit rates with a minimal loss in speech quality.

1.2 Speech Coding

Speech Coding methods are employed to "compress" the speech signal prior to transmission or storage. Compression is achieved by removing any redundancy which is present in the speech waveform, using various mathematical models. Efficient Speech Coding methods can minimise the loss in speech quality which always results during the two-stage conversion process of "compression" and "decompression".

There is currently an increasing demand for higher compression ratios and lower transmission bit rates, in many application fields. Whereas at the high bit rates of 64 kbits/sec and 32 kbits/sec, relatively simple compression techniques (i.e. PCM, ADPCM) can provide toll quality speech, at bit rates below 16 kbits/sec the deterioration in the quality of the recovered speech signal is considerable, and only sophisticated coding methods can minimise the loss in quality. The complexity of these highly efficient coding methods increases substantially at bit rates below 8 kbits/sec.

Speech codecs (coder/decoder) employ mathematical models that are based on our knowledge of the human speech production and auditory perception mechanisms. The selection and optimisation of the model parameters can be performed much more efficiently now than it was possible a few years ago, as the current availability of substantial processing power permits the use of very complex mathematical algorithms. The recent trend in speech coding research has been to develop better and more complex processing/optimisation algorithms which can be applied to existing speech production and hearing models.

Current Speech Coding algorithms can offer excellent speech quality at bit rates approaching 10 kbits/sec. Further development is needed however, to

provide the capability of "natural" speech quality at bit rates close to the theoretical limit of 2 kbits/sec [1.7]. In order to reach this limit, future research must focus on the deeper understanding of the higher levels of auditory perception, and on the development of more detailed and accurate models of the human speech production and hearing organs.

The scope of this thesis is to develop highly efficient speech coding algorithms for processing narrowband speech (0-4 kHz bandwidth) at medium bit rates (4-16 kbits/sec). The common characteristic between these algorithms is the technique applied to determine the speech model parameters, known as Analysis-by-Synthesis (AbS). AbS Speech Coding algorithms attempt to exploit the characteristics of the human auditory perception, and thus aim to minimise the "perceptual" effect of the distortion introduced by the coding process.

Two different types of AbS Speech Coders are examined, the Multipulse Excitation (MPE) coder and the Backward Excitation Recovery (BER) coder. Both coders employ the Source-Filter model of speech production, but rely on different procedures to define the Excitation Source. A number of new model-optimisation algorithms are proposed for each type of coder and the performance/complexity of the resulting systems is compared to that obtained from conventional AbS coders. The comparison can help potential designers of a Speech coding system, to determine which algorithm would better suit the complexity/performance requirements of the particular application.

1.3 Thesis Overview

In Chapter 2, the general requirements for speech coders operating in digital communication networks are considered. Furthermore, a review of the speech coding techniques currently used at bit rates between 4 kbits/sec and 16 kbits/sec is presented. Although these coding techniques employ speech models which are generally different, the use of AbS optimisation algorithms has been widely adopted and has led to an improvement in the accuracy of estimation of the model parameters.

In Chapter 3, the theory of the Multipulse Excitation (MPE) coder is developed, and a general classification of the MPE Optimisation algorithms is presented. Some of these algorithms are highly complex and are therefore only of theoretical interest, since they can provide the upper limit in the

performance of MPE coders. Two new classes of algorithms are also proposed, namely the Block-Search and Codebook-Search algorithms, which are examined in more detail in Chapters 4 and 5.

In Chapter 4, the two filter models used in MPE coders (i.e. Short Term Predictor and Long Term Predictor) are developed, and various filter estimation methods are examined. A detailed presentation of the mathematical theory reveals ways of improving MPE coding, and suggests alternative coder configurations. Furthermore, an algorithm is proposed that can be employed to jointly estimate the parameters of the Long Term Predictor and the Multipulse Excitation.

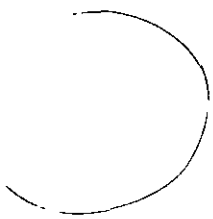
In Chapter 4, a number of Multi-Stage (MS) algorithms are developed, that are employed to estimate the parameters of the MPE. An new Exponential Model is proposed that leads to a simplified MS algorithm, and a computationally efficient version of a popular MS algorithm is developed, using the Gram-Schmidt orthogonalisation procedure. The theory of the proposed Block-Search (BS) algorithms is also developed and the performance of the new coders is compared to that of MS algorithms, in terms of Signal to Noise Ratio (SNR) and complexity (number of operations). Finally the quantization of the MPE parameters is considered, and a new theoretical model for the probability distribution of the excitation values is proposed, which can be used to design globally optimum scalar quantizers.

Chapter 5 examines the proposed Codebook-Search (CS) algorithms. These algorithms are generally more complex than the MS and BS algorithms, but they are also more efficient in coding the Multipulse Excitation Sequence and can therefore achieve better performance at lower bit rates. A method is proposed that can be used to "design" the codebook employed by the MPE coder, according to a "subjective" speech-quality criterion. A fast implementation of the Codebook Design algorithm is also developed. The perfromance of CS-MPE coders with "optimised" codebooks is compared to that obtained when random codebooks are employed.

The introduction of "structure" into the codebook employed by the CS-MPE coder, results a substantial reduction in the complexity of the sustem. Thus a number of possible codebook structures are proposed which lead to computationally efficient ways of searching the MPE codebook.

In Chapter 6, a new type of Speech Coder is proposed. It is the Backward

Excitation Recovery (BER) speech coder which defines the Filter and Excitation parameters using a backward adaptation procedure. This coder can operate at bit rates that are lower than the bit rates at which conventional MPE coders operate. A detailed mathematical description of BER coding is presented, and a number of Filter Optimisation and Excitation Adaptation algorithms are proposed. A large number of possible BER coder configurations are developed and their properties are compared. Two such special cases of BER coding are found to be the Code-Excited LPC (CELP) coder and the Self-Excited (SE) vocoder. Finally, the quantization of the filter coefficients is considered, and a new algorithm is proposed that is used to design and optimise a vector quantizer, according to a "subjective" speech-quality criterion.



REFERENCES

[1.1] J.G.Wade, "Signal Coding and Processing: An Introduction based on Video Systems", Ellis Horwood Limited, John Wiley & Sons, 1987

[1.2] L.R.Rabiner, R.W.Schafer, "Digital Processing of Speech Signals", Prentice-Hall Signal Processing Series, New Jersey

[1.3] N.S.Jayant, P.Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video", Prentice-Hall Signal Processing Series, New Jersey 1984

[1.4] J.L.Flanagan et al., "Speech Coding", IEEE Trans. on Communications, Vol 27, No 4, April 1979, pp 710-736

[1.5] A.Aliphas, J.A.Feldman, "The Versatility of Digital Signal Processing Chips", IEEE Spectrum, Vol 24, No 6, June 1987

[1.6] "Special Issue on Hardware and Software for DSP", IEEE Proceedings, Vol 75, No 9, September 1987

[1.7] B.S.Atal, L.R.Rabiner, "Speech Research Directions", AT&T Technical Journal, Vol 65, Issue 5, Sep/Oct 1986, pp 75-88

[1.8] R.E.Crochiere, J.L.Flanagan, "Speech Processing: An Evolving Technology", AT&T Technical Journal, Vol 65, Issue 5, Sep/Oct 1986, pp 2-11

[1.9] W.C.Lee, "Mobile Communications Engineering", McGraw-Hill Book Company, New York 1982

[1.10] J.G.Gardiner, "Prospects for Sattelite Mobile Radio services for Europe in the 1990s", IERE Journal, Supplement on Mobile Radio, Vol 57, No 6, Nov/Dec 1987, pp S241-S245

[1.11] C.Xydeas et al., "Speech Processing in Mobile Radio Communications", Proc. IEE colloquium on Digitised Speech Communication via Mobile Radio, December 1988

[1.12] K.Feher, "Digital Communications: Satellite/Earth Station Engineering", Prentice-Hall, New Jersey 1983

[1.13] D.O'Shaughnessy, "Speech Communication: Human and Machine", Addison-Wesley Publishing Company, 1987

[1.14] F.T.Andrews, "ISDN 83", IEEE Communication Magazine, Vol 22, No 1, January 1984, pp 6-10

[1.15] R.T.Roca, "ISDN Architecture", AT&T Technical Journal, Vol 65, Issue 1, Jan/Feb 1986, pp 5-17

CHAPTER 2

SPEECH CODING AT MEDIUM BIT RATES

2.1 Introduction

Narrowband Speech Coders operate at bit rates between 0.4 and 64 kbits/sec. Coders operating at high bit rates usually employ very simple coding algorithms, and are capable of producing high quality speech. At low bit rates, highly complex coding algorithms are employed and the quality of the synthesised speech is more "synthetic" than "natural".

At bit rates above 16 kbits/sec, Waveform Coding methods are commonly employed. Widely used Waveform Coding techniques are the Pulse Code Modulation (PCM), the Adaptive Differential PCM (ADPCM), and the Adaptive Delta Modulation (ADM) [2.1]. Waveform Coding methods produce a recovered speech signal which is a faithful reconstruction of the original speech waveform. They are not critically dependent on the characteristics of the input signal and can therefore be applied (with certain modifications) to non-speech signals (i.e. music, video etc.). Such time-varying signals are also generated at "intermediate" stages within the structure of various low-bit-rate speech coders.

PCM offers a high degree of robustness to transmission errors and produces high quality speech at 56 kbits/sec and toll quality at 64 kbits/sec. PCM coding has dominated speech processing applications for almost 25 years. ADPCM speech coders which employ adaptive predictors and adaptive quantizers, can produce the same quality as PCM coders, but at the lower bit rate of 32 kbits/sec. Some ADPCM coders can also "shape" the spectral distribution of the added quantization noise, in order to reduce the "perceptual" level of distortion. The noise shaping increases the level of distortion in the spectral areas where the speech power is high and higher levels of noise can be tolerated [2.2]. The opposite happens in the areas where the speech power is low. ADM methods are similar to ADPCM methods, but employ a two-level quantizer and are simpler to implement. They are very robust in the presence of transmission errors and can therefore be used in "noisy" environments. The performance of Waveform Coding methods deteriorates rapidly as the bit rate is reduced to 16 kbits/sec and below.

The lower end of the transmission-bit-rate "spectrum" is occupied by speech coding methods known as vocoders [2.3,2.4]. Vocoders use a model of the human speech production mechanism to obtain a compact representation of the speech signal. They usually operate at bit rates between 0.4 and 4.8 kbits/ sec. Speech production is modeled by a Source-Filter arrangement. The coding process models the signal in terms of two components corresponding to the action of the vibrating vocal chords (Source) and the shape of the vocal tract (Filter). Usually, two types of speech sounds are distinguished: voiced (periodic) and unvoiced (noise-like). The source signal is composed of a periodic series of pulses when speech is classified as voiced, and it becomes random when speech is classified as unvoiced. The source signal forms the input to the filter, whose output is the synthesised speech signal.

The quality of speech produced by vocoders is "synthetic" and deteriorates rapidly in the presence of acoustic noise. The identification of the speaker is often problematic. The reasons for the poor speech quality are the simplistic model for the source signal and the inaccurate modeling of the time-varying phase characteristics of the speech waveform.

A number of "hybrid" speech coding schemes have been developed in recent years, which bridge the gap between Waveform Coders and Vocoders, and promise to improve the quality of the recovered speech at low bit rates. These coders operate at medium bit rates (between 4 and 16 kbits/sec) and can generally produce speech which is more "natural" than the speech produced by vocoders. The speech models employed are more detailed than the models used by vocoders, and thus more accurate description of the model is derived and transmitted. The optimisation and the quantization of the model parameters is performed so that the synthesised speech signal becomes an accurate reconstruction of the original speech signal, either in the time or in the frequency domain.

Noise shaping can be included in these systems to reduce the level of the "perceived" distortion. The shaping of the noise spectrum is more important in medium-bit-rate coding than it is in Waveform Coding, because the level of quantization noise is higher. At bit rates close to 10 kbits/sec, hybrid coding methods can produce high quality narrowband (0-4 kHz bandwidth) speech, which is only possible to achieve using Waveform Coding techniques

at much higher bit rates. In order to reduce the bit rate, at which high quality speech can be produced, even further, Vector Quantization methods [2.5,2.6,2.7,2.8] are employed to quantize efficiently the model parameters.

The speech coding schemes that have given promising results at medium bit rates will be described in the following sections.

2.2 Requirements for Speech Coders

The selection of a speech coder for a particular application is usually made by testing and comparing a number of candidate speech coders under various conditions. The primary requirement is that the speech coder chosen, produces the "best" possible speech quality at the given bit rate.

Speech quality can be measured using objective measures [2.9,2.10,2.11] like the Segmental Signal-to-Noise Ratio (Seg-SNR) [2.1]. The objective quality measures do not always correspond to the speech quality as judged by a listener, and thus in many applications the speech quality is assessed by performing subjective listening tests. At bit-rates below 4 kbits/sec the speech quality is poor and the tests carried out are concerned with speech intelligibility and speaker recognisability [2.12]. At higher bit rates the speech quality is more natural, and the subjective tests measure the "perceptual" quality [2.13] under various test conditions. Typical test conditions relevant to applications in digital speech transmission are :

1) Coding of speech when no transmisson errors occur.

2) Coding of speech with injected transmission errors with random or bursty arrival statistics. Error protection (channel coding) of the most sensitive coder parameters may be included to improve the performance of the speech coder under

3) Synchronous or asynchronous tandem encodings. A tandem encoding refers to the conversion from the digital format of one coder to that of another speech coder. A tandem encoding is synchronous when it involves digital-to-digital conversion, while it is asynchronous when it involves digital-to-analog followed by analog-to-digital conversion. Transmission through the telephone network may involve several tandem encodings with PCM links, and it is therefore important to measure the resulting loss of speech-quality when a particular speech coder is employed.

4) Asynchronous tandem encodings with injected analog impairments (noise, amplitude and group-delay distortion), these conditions being critical for voiceband (non-speech) data transmission [2.14].

5) Presence of acoustical noise in the input speech signal. This condition is encountered in Land Mobile Radio Communications where the level of noise in the environment can be quite high. Adaptive noise cancellation techniques [2.15] may be applied that can reduce the noise level without causing considerable distortion to the speech signal itself.

Speech quality is assessed under the defined test conditions using various "subjective" quality measures. Typical subjective quality measures are :

1) The Mean Opinion Score (MOS), which requires the listeners to judge the speech quality using a five-point scale [2.16]. The speech quality can be judged as excellent, good, fair, poor or bad. The final score is simply the average judgement.

2) The Subjective SNR [2.17,2.18]. This measure involves adding sufficient speech-modulated white noise to the input speech signal, so that it and the coded speech are equally preferred. The Signal-to-Noise Ratio of the resulting multiplicative-noisy speech is defined as the Subjective SNR of the coded speech. A popular method that measures the Subjective SNR involves the use of the Modulated Noise Reference Unit (MNRU) [2.19].

3) Binary Decision Preference tests [2.1]. These tests require each listener to indicate his or her preference when comparing the speech quality obtained from two different coders. Non-binary preference ranking for a set of speech coders can be derived by using the results of many such pair-comparisons.

Other inportant factors taken under consideration when choosing a speech coder for a particular application are :

1) The coder's complexity, which is usually measured in terms of the numerical processing power and the amount of physical memory required in order to implement the coder.

2) The encoding delay introduced by the speech coder. Small encoding delays are important in applications such as Mobile Radio Communications where the system's overall delay must be kept to a minimum.

3) The ability of the speech coder to handle voiceband (non-speech) data. Coders that operate at medium bit rates are generally not very efficient in handling voiceband data, because they are "tuned" for speech signals. In this case a detection process must be applied that switches to a different coding method when voiceband data are transmitted.

4) Ease of transcoding with PCM in applications that use the digital telephone network.

5) Amenability to Variable Rate coding. This is a desirable property of the coder, when optimal loading of the transmission network is required. Packet-Switced networks for example, may require the speech coders to switch to lower transmission bit-rates when the traffic is very high. Variable Rate speech coders usually have a hierarchical structure, so that the less important parameters can be dropped when the bit-rate is reduced.

2.3 Predictive Coding of Speech

Predictive Coding of speech is an analysis/synthesis modeling process that decomposes the speech signal into two time-varying components with different properties [2.20,2.21]. The two components are separately encoded and are recombined to form synthetic speech. One of the components is an adaptive linear filter which models the slow-changing spectral distribution of the speech signal, and includes contributions from the glottal response, the vocal tract shape and the lip radiation [2.22]. The second component is the excitation signal which is fed to the filter in order to produce synthetic speech. The analysis process (decomposition) is performed at the encoder (transmitter) and the encoded model parameters are transmitted to the decoder (receiver) where synthetic speech is reproduced.

Predictive models are employed by vocoders, which encode the excitation signal with a very small number of parameters (i.e. input gain, pitch period and voiced/unvoiced classification) [2.23]. This crude encoding of the excitation signal results synthetic speech of poor quality. Predictive models are also employed by waveform coders such as ADPCM, which encode the excitation signal very accurately and can produce high quality speech at bit rates above 16 kbits/sec. Medium bit-rate coders allocate a much smaller number of bits in encoding the excitation than most waveform coders, but are

capable of producing high quality speech by using efficient and complex techniques to encode the excitation.

1) LINEAR PREDICTION CODING (LPC)

The model most often used in predictive coding is the Autoregressive (AR) model :

$$s(i) = \sum_{m=1}^{l} a_m \, s(i-m) + e(i) \quad , \quad i=0,1,2,... \qquad (Eq\ 2.3.1)$$

where $s(i)$ is the speech signal, $e(i)$ is the forward prediction error, and l is the number of $\{a_m\}$ coefficients. Equation 2.3.1 can be written in the one-sided z-transform domain as :

$$S(z) = A(z)\ E(z) \qquad (Eq\ 2.3.2)$$

$S(z)$ and $E(z)$ correspond to the speech and prediction error signals, and $A(z)$ is the all-pole filter :

$$A(z) = \frac{1}{1 - \sum_{m=1}^{l} a_m \, z^{-m}} = \frac{1}{1 - P(z)} \qquad (Eq\ 2.3.3)$$

The filter $P(z)$ is a one-sample-ahead predictor. The AR model is shown in Fig 2.3.1(a). The error samples $\{e(i)\}$ are assumed to be statistically independent and therefore $E(z)$ has a flat frequency distribution. As the speech spectral distribution is the product of the frequency distributions corresponding to the filter $A(z)$ and the error sequence $E(z)$ (Eq 2.3.2), most of the spectral shape information is contained in the filter $A(z)$. The all-pole filter $A(z)$ can model accurately the spectral resonances (formants) which are characteristic of most speech sounds, but cannot model well the sounds that contain spectral antiresonances (such as nasal sounds).

The filter parameters are estimated using Linear Prediction Coding (LPC) methods. LPC methods employ Least Squares (LS) algorithms to minimise the energy E of the prediction error $\{e(i)\}$ over a defined time period $(0,n-1)$:

$$E = \sum_{i=0}^{n-1} e(i)^2 \qquad (Eq\ 2.3.4)$$

Using Equations 2.3.2 and 2.3.4 and by applying Parceval's theorem, it can

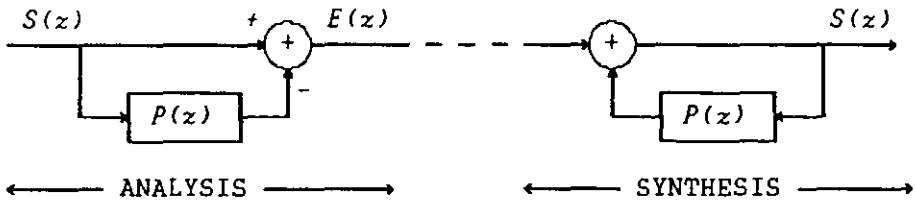


Fig 2.3.1(a)

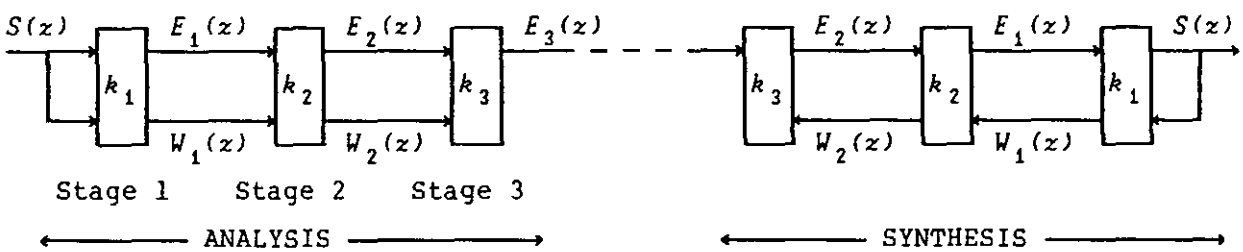


Fig 2.3.1(b)

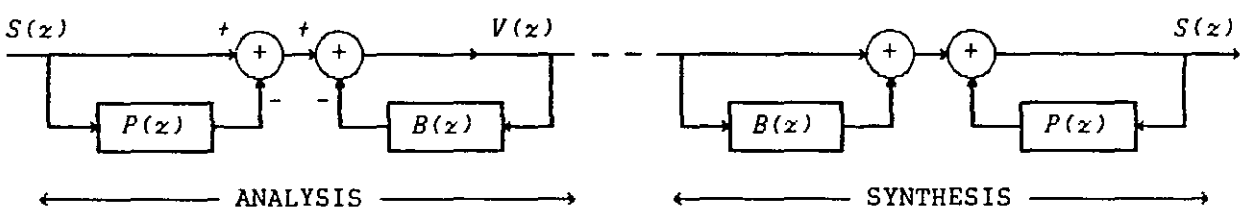


Fig 2.3.1(c)

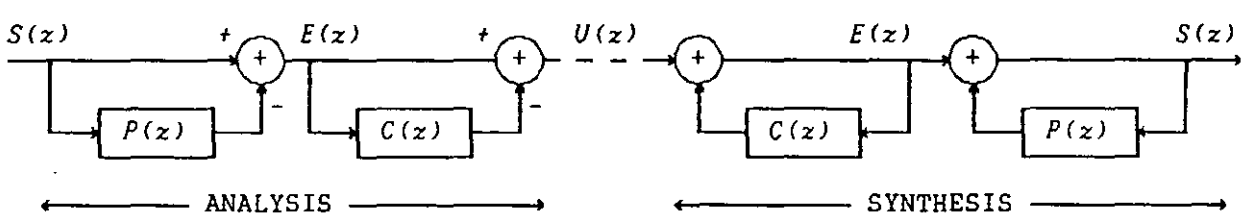


Fig 2.3.1(c)

FIGURE 2.3.1 Analysis/Synthesis models used in Predictive Coding of Speech. (a) Autoregressive model, (b) Lattice Autoregressive model, (c) ARMA model, (d) Cascade of two AR filters.

be shown that minimising the value of the prediction error energy E is equivalent to minimising the value of the expression :

$$E = \sum_{i=o}^{n-1} \frac{|S(f_i)|^2}{|A(f_i)|^2} \qquad (Eq\ 2.3.5)$$

where :

$$f_i = exp\left(j\ \frac{2\pi i}{n}\right) \quad \text{and} \quad j = \sqrt{-1} \qquad (Eq\ 2.3.6)$$

The expression of Eq 2.3.5 is the sum of the ratios of the speech power spectrum to the model power spectrum over the defined frequencies. The frequency points where the speech power $|S(f_i)|^2$ is greater than the model power $|A(f_i)|^2$ contribute more to the value of the error E, than the points where the power ratio is smaller than one. As the LPC-model spectrum has a certain degree of "smoothness", in areas where speech spectral peaks (harmonics) are closely spaced it provides a closer match to the high-power parts of the spectrum than the low-power parts. The LPC-model spectrum thus follows the envelope of the speech spectrum.

2) ESTIMATION OF THE LPC-FILTER COEFFICIENTS

The filter coefficients must be updated frequently in order to follow the variation with time of the speech spectral distribution. The adaptation of the filter coefficients can be performed using Block or Recursive algorithms [2.27, 2.28]. Block algorithms divide the speech signal into blocks of consequtive samples, and each block is processed independently. The optimum parameters of the filter are determined by a series of mathematical operations and the end result is a new set of coefficients for each speech block. Recursive algorithms apply a set of recursive equations so that a new set of filter coefficients is generated at each input speech sample [2.29].

The primary distinction between the two types of algorithms is that Block algorithms have a finite memory, while Recursive algorithms usually have infinite memory. Block algorithms are very popular in speech processing because they allow the LPC model to adapt to the rapid changes of the speech signal statistical properties.

The most widely used Block estimation algorithms are :

i) The Covariance method [2.22], which minimises the energy of the forward prediction error (Eq 2.3.4) without applying any time window to the speech

data. The filter coefficients are estimated by solving the covariance system of linear equations. Computationally efficient solutions are available which exploit the special structure of the covariance matrix [2.30] and reduce the number of operations required to solve the system of equations.

The estimated filter $A(z)$ is not guaranteed to be minimum-phase (i.e. have all its poles inside the unit circle) and this may sometimes create instability problems. The Covariance method can be modified by introducing the concept of "generalised reflection coefficients" [2.31,2.32], which can be combined with the application of Levinson's Recursion [2.33] to produce minimum-phase filters. The Covariance method can also be extended by defining a backward prediction error $\{w(i)\}$ as :

$$s(i) = \sum_{m=1}^{l} a_m \, s(i+m) + w(i) \quad , \quad i=0,1,2,... \qquad (Eq\ 2.3.7)$$

and by minimising the sum of the energies of the forward and backward prediction errors [2.34].

ii) The Autocorrelation method [2.22], which applies a suitable window (e.g. Hamming window) to the speech data, and minimises the energy of the prediction error over a range that extends form $-\infty$ to ∞. The windowing operation distorts the speech signal and reduces the spectral resolution of the method [2.35]. The Autocorrelation method though has certain advantages in that it produces minimum-phase filters and permits the use of a computationally efficient formula (Durbin's solution [2.36]) for the calculation of the filter coefficients.

iii) Lattice methods [2.37], which define the LPC-filter in a series of stages that correspond to the stages of a Lattice Filter. The ith stage of the Lattice filter produces two sequences that correspond to the forward and backward prediction-errors $E_i(z)$ and $W_i(z)$ (see Fig 2.3.1(b)), and defines one reflection coefficient k_i. The reflection coefficients can be converted to the coefficients of the direct-form (transversal) filter by applying Levinson's recursion.

Various Lattice-filter structures have been defined, that are theoretically equivalent but behave differently when finite precision arithmetic is used. There are also many possible ways of calculating the reflection coefficients [2.38]. A popular Lattice method is the Burg algorithm [2.39],

which is also known as the Maximum-Entropy method, because it extrapolates the autocorrelation sequence derived from the speech data in a way that maximises the entropy of the model spectrum [2.40]. The Lattice methods yield minimum-phase filters, they have good numerical properties and produce a set of reflection coefficients that possesses much more desirable quantization properties than the equivalent set of direct-form (transversal) filter coefficients.

Note that the Autocorrelation method can be considered as a Lattice method and can also be implemented using a Lattice filter structure. Notice also that all three types of methods (i.e. the Covariance, Autocorrelation and Lattice methods) give the same results when the chosen speech interval is sufficiently large.

3) IMPROVED LPC MODELS

The AR model of Eq 2.3.1 can be improved by extending to the case of an Autoregressive Moving Average (ARMA) process, as shown in Fig 2.3.1(c). A further set of coefficients $\{b_m\}$ is defined, and the modeling equation is :

$$s(i) = \sum_{m=1}^{l} a_m \, s(i-m) + \sum_{m=1}^{g} b_m \, v(i-m) + v(i) \quad , \quad i=0,1,2,... \qquad (Eq\ 2.3.8)$$

where g is the number of the $\{b_m\}$ coefficients, and $\{v(i)\}$ is the prediction error sequence. The equivalent of Eq 2.3.8 in the z-transform domain is :

$$S(z) = \frac{1 + \sum_{m=1}^{g} b_m \, z^{-m}}{1 - \sum_{m=1}^{l} a_m \, z^{-m}} \, V(z) = \frac{1 + B(z)}{1 - P(z)} \, V(z) \qquad (Eq\ 2.3.9)$$

The linear transfer function is now rational (pole-zero), and the ARMA filter can provide a better model for the resonances and antiresonances of the speech frequency distribution. The estimation of the ARMA filter coefficients is a nonlinear problem which is solved by "linearising" the error minimisation process, or by applying iterative minimisation techniques [2.34,2.41,2.42]. Lattice filter structures can be defined for the case of the ARMA linear filter that guarantee the minimum-phase properties of the filter [2.43,2.44]

A different improvement of the LPC model of speech can be made by adding a second filter that models the harmonic structure of the speech frequency distribution. The model is that of an Autoregressive process :

$$y(i) = \sum_{m=-q}^{q} c_m \, y(i-d-m) + u(i) \quad , \quad i=0,1,2,... \qquad (Eq\ 2.3.10)$$

where $y(i)$ may be the speech signal or the prediction-error sequence defined in Eq 2.3.1, $u(i)$ is a second prediction-error sequence, and d is an estimate of the pitch period of the speech waveform. The filter has $(2q+1)$ coefficients $\{c_m\}$ in total, and its frequency response has very pronounced harmonic peaks at multiples of the fundamental frequency (defined by the value of d). The transfer-function model corresponding to Eq 2.3.10 is:

$$Y(z) = H(z)\, U(z) \qquad (Eq\ 2.3.11)$$

where :

$$H(z) = \frac{1}{1 - \sum_{m=-q}^{q} c_m \, z^{-m}} = \frac{1}{1 - C(z)} \qquad (Eq\ 2.3.12)$$

The two filters defined by Eqs 2.3.3 and 2.3.12 are connected in series (see example in Fig 2.3.1(d)) and their coefficients may be optimised separately or jointly [2.45] using Linear Prediction techniques. The filter $H(z)$ is not guaranteed to be minimum-phase and its stability must be checked and corrected if required [2.46].

4) QUANTIZATION OF THE LPC-FILTER PARAMETERS

Ideally, the quantization of the LPC-filter parameters must have no discernible effect on the quality of the encoded speech, and at the same time the number of bits allocated for the quantization of the filter parameters must be sufficiently small for speech coding applications. In practice however, a small quality degradation is unavoidable at low bit rates, and the quantizers are designed to minimise the perceived loss in quality.

The coefficients of the direct-form (transversal) AR-LPC filter are not suitable for direct quantization, due to their large dynamic range. Alternative sets of filter parameters can be defined that have better quantization properties. Suitable parameter sets are :

i) The Log-Area-Ratios (LARs), which are derived from the filter reflection coefficients and have flat spectral sensitivity characteristics [2.47,2.48]. A set of filter parameters with similar properties as the LARs can be defined by transforming the filter reflection coefficients according to the companding characteristic of the inverse-sine function [2.20,2.49,2.50].

ii) The poles of the LPC filter, which are directly related to the frequencies and the bandwidths of the speech spectral resonances (formants) [2.51].

iii) The Line Spectrum Pairs (LSPs), which are defined by introducing two artificial boundary conditions to the AR-LPC filter. These conditions correspond to a complete opening and a complete closure at the glottis in the acoustic tube model of the vocal tract, and produce one symmetric and one anti-symmetric polynomial respectively. The roots of these two polynomials lie on the unit circle and correspond to the discrete frequencies of two interleaved Line Spectra (LSP) [2.52,2.53]. It has been established experimentally that the LSP parameters have better quantization and interpolation properties than the LAR parameters [2.54].

The different sets of parameters can be quantized using various quantization methods such as :

i) Uniform quantization.
ii) Non-uniform PDF-optimised quantization.
iii) Adaptive quantization with Forward or Backward adaptation of the quantizer step-size.
iv) Vector Quantization (VQ) with Euclidian or Spectral distance-measures such as the Itakura-Saito distortion-measure [2.5,2.55]. Another VQ method uses an adaptive vector-codebook for the LAR parameters [2.56]. The adaptive codebook is formed using a random Gaussian codebook and an estimate of the covariance of the LAR parameters which is updated using backward adaptation.

Greater efficiency in coding the LPC-filter parameters can be achieved by taking into account the interframe correlation of the parameters. Methods that model the interframe variation of the LPC-parameters include :

i) A Vector-Autoregressive model that represents the evolution (in time) of the LAR filter parameters. This model assumes a step-function input to the vector-predictive model [2.57].

ii) Vector Predictive Quantization, which predicts the current set of para-

meters from past parameter sets, using a predictor codebook [2.58]. This method is also known as Switched-Adaptive Interframe Vector Prediction [2.59].

5) ADAPTIVE PREDICTIVE CODING (APC)

The block diagrams of a basic APC encoder and decoder are shown in Figures 2.3.2(a) and (b) respectively. The prediction error (residual) $E(z)$ is quantized on a sample-by-sample basis, and is transmitted to the decoder together with the quantized parameters of the LPC filter. The quantization of the residual is performed inside the prediction loop at the encoder, in order to prevent the quantization noise from being amplified during the speech synthesis process at the decoder. The encoded residual is :

$$T(z) = E(z) + Q(z) \qquad (Eq\ 2.3.13)$$

where $Q(z)$ is the added quantization noise. Also from Fig 2.3.2(a) :

$$E(z) = S(z) - G(z) \qquad (Eq\ 2.3.14)$$

and :

$$G(z) = \left[G(z) + T(z) \right] P(z) \qquad (Eq\ 2.3.15)$$

By substituting Eqs 2.3.14 and 2.3.15 into Eq 2.3.13, the relationship obtained is :

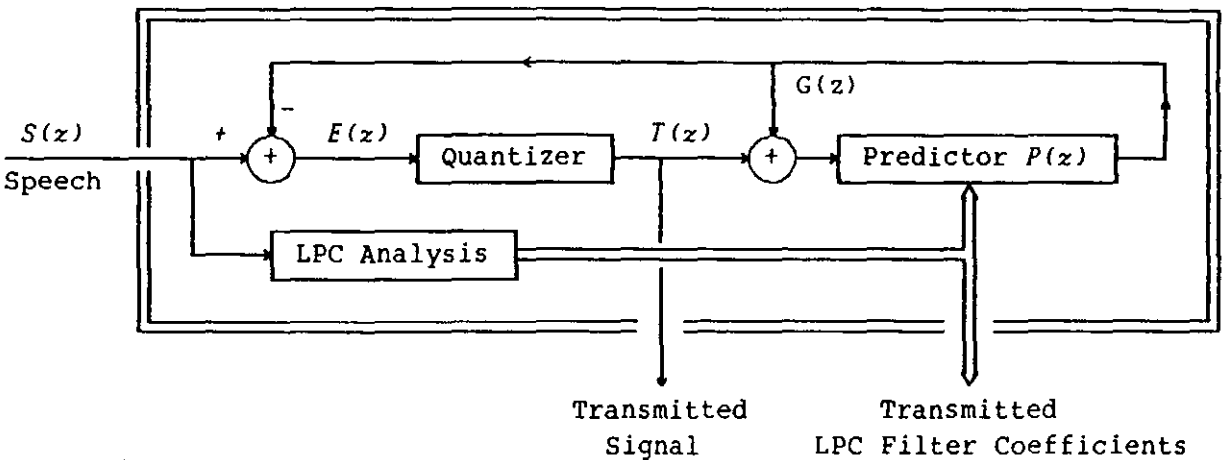
$$T(z) = \left[S(z) + Q(z) \right] \left[1 - P(z) \right] \qquad (Eq\ 2.3.16)$$

The speech signal $D(z)$ synthesised at the receiver is (see Fig 2.3.2(b)) :

$$S(z) = T(z) \frac{1}{1 - P(z)} = S(z) + Q(z) \qquad (Eq\ 2.3.17)$$

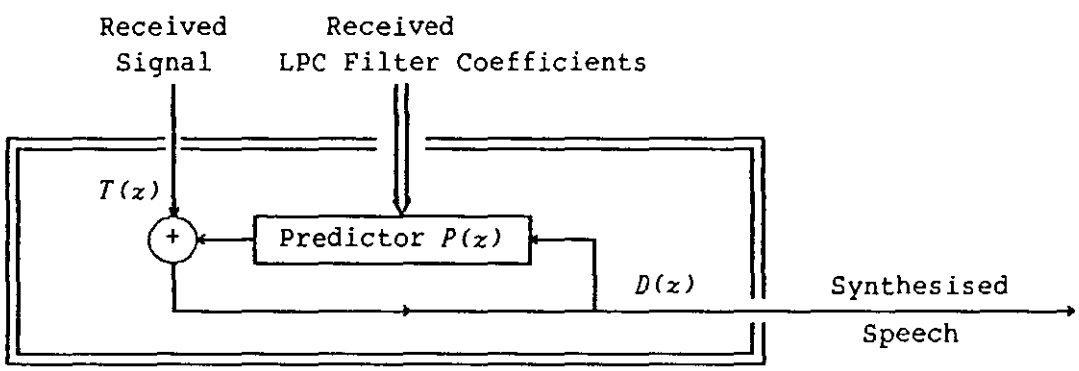
Thus in an APC coder the synthesised signal is identical to the input speech signal with the addition of the noise introduced by the quantizer.

The definition of the APC coder can be extended to include a second filter (pitch predictor) that models the harmonic structure of the speech frequency distribution. The pitch-predictor is connected in series with the LPC-filter and usually precedes the LPC-filter in the synthesis stage. A noise shaping filter can also be added, that reduces the subjective loudness of the quantization noise [2.20]. The noise shaping filter increases the level of quantization noise in the formant regions (where noise is partially masked



ENCODER

Fig 2.3.2(b)



DECODER

Fig 2.3.2(b)

FIGURE 2.3.2 The encoder (a) and decoder (b) of a basic Adaptive Predictive Coder (APC).

by the speech signal) and decreases the noise level in the spectral valleys. The masking of the quantization noise by the speech signal allows lower bit rates to be used, while maintaining high speech quality. The coefficients of the noise shaping filter are derived from the parameters of the LPC filter.

If the residual is quantized on a sample-by-sample basis, then at bit rates below 10 kbits/sec a two-level quantizer must be used, and the resulting coarse quantization becomes a major source of audible distortion in the synthesised signal. Efficient quantization of the residual (at less than one bit per sample) can be achieved by using a center-clipping quantizer that only encodes the largest samples (peaks) of the residual [2.20], or by using adaptive bit-allocation in the time and frequency domain [2.60].

Greater efficiency in quantizing the residual can be achieved by using Delayed Decision Coding techniques combined with Vector Quantization. Tree codes generated by a stochastically (random) populated innovations tree have been proposed for operation at 16 kbits/sec, producing speech of near toll quality (equivalent to 7 bits/sample log-PCM) [2.61]. Bit rates as low as 4.8 kbits/sec can be achieved by vector-quantizing the residual using as a distance measure the RMS value of the signal distortion [2.62] (see section 7 on Analysis-by-Synthesis predictive coding).

6) RESIDUAL-EXCITED LINEAR PREDICTION (RELP) CODING

RELP coders employ the Autoregressive model of speech (Eq 2.3.1) and code the prediction error (residual) using a combination of time and frequency domain techniques. The most commonly used coding-method is Baseband Coding combined with High Frequency Regeneration (HFR). The basic assumption in this method is that the lowest frequencies of the residual spectrum carry the highest perceptual importance, and that the preservation of the residual baseband contributes to the naturalness of the synthesised speech.

The early RELP coding schemes used a simple procedure to code the residual [2.63,2.64]. This process involves the low-pass filtering (up to 800 Hz) and decimation of the residual. The down-sampled signal is encoded and transmitted using waveform coding methods. At the receiver, the transmitted signal is up-sampled and processed to regenerate the high frequency part of the spectrum. The signal is then fed to the LPC synthesis-filter to produce synthetic speech.

Simple HFR techniques include nonlinear processing of the baseband signal (e.g. full-wave rectification), baseband duplication (e.g. spectral folding) or a combination of the two [2.65,2.66]. These methods cannot reconstruct the upper frequency band accurately, and generate audible distortion in the form of "hoarsness" or "tonal noise".

Better models have been developed for the coding of the residual in the time or the frequency domain. These models include :

i) Pitch-aligned HFR methods that duplicate the baseband spectrum in a pitch synchronous manner, thus reducing the harmonic discontinuities and the tonal noise [2.67,2.68]. These methods operate in the frequency domain and require an estimate of the speech fundamental frequency.

ii) Use of a full-band pitch predictor in the time domain, to remove the pitch information from the residual before decimation and restore it after upsampling [2.69].

iii) A generalised decimation process which produces an irregularly down-sampled residual and minimises a perceptual distortion measure [2.70,2.71]. The down-sampled residual is quantized using APCM. Very good quality speech can be produced when this method is employed at 10 kbits/sec.

iv) Vector Quantization of the harmonic frequency components (real and imaginary) of the residual [2.72]. Pitch synchronous replication of the harmonics must be used to regenerate the high part of the frequency spectrum.

v) A dynamic spectral model which defines a set of adaptively selected sub-bands of the residual, rather than a single low-pass sub-band. This model is defined in the frequency domain, and the bits are allocated to the sub-bands according to their significance [2.73].

RELP coders can produce very good quality speech at 16 kbits/sec, but the quality falls off rapidly as the bit rate is reduced below 8 kbits/sec. They are used in applications where low algorithm-complexity is necessary, and in general the quality of the sythesised speech is not as good as that obtained from Analysis by Synthesis Predictive coders.

7) ANALYSIS-BY-SYNTHESIS (AbS) PREDICTIVE CODING

The block diagram of a general AbS predictive coder is shown in Figure 2.3.3(a) and (b). The speech model used is the analysis/synthesis model

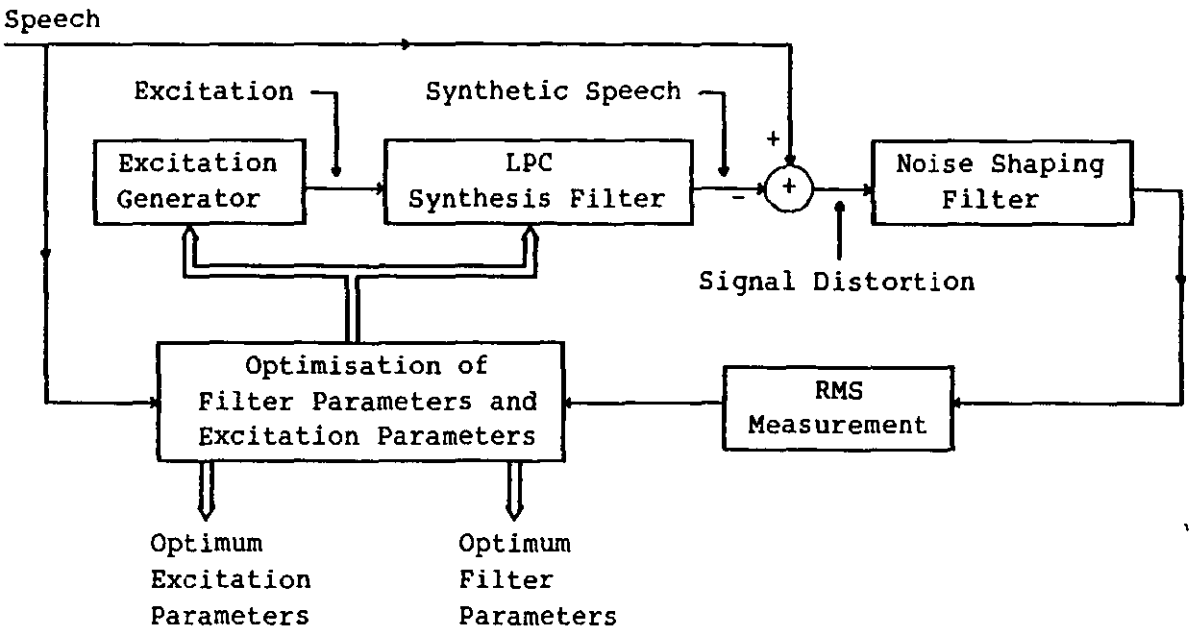
employed by APC and RELP coders, thus synthetic speech is produced by passing the excitation signal through the LPC synthesis filter. In an AbS coder, both the filter and the excitation signal are parametrically represented over a short time interval (5 ms to 20 ms). The set of parameters controls the shape of the excitation signal and determines the frequency response of the LPC filter. Furthermore, the parameter values are adjusted by a closed-loop optimisation process that minimises the value of a weighted distortion measure.

As seen in Fig 2.3.3(a), an error signal is formed by comparing the original and synthetic speech waveforms. The error signal passes through a noise-shaping filter that puts more emphasis (amplifies) on the frequency regions where the speech power is low and the noise (distortion) cannot be masked by the speech signal, and attenuates the error signal in the speech formant regions where higher levels of noise can be tolerated due to the masking effect. The RMS value of the filtered error signal serves as a measure of the "subjective" level of distortion.

AbS-Predictive coders are more effective in minimising the distortion introduced by the coding process and in achieving the desired noise spectrum, when compared to conventional open-loop coders such as APC and RELP. They can produce speech of "excellent" quality (acceptable to most applications) at a bit rate of 10 kbits/sec, while at the lower bit rate of 6 kbits/sec many AbS-Predictive coders can produce speech of very good quality. Typical AbS-Predictive coders are :

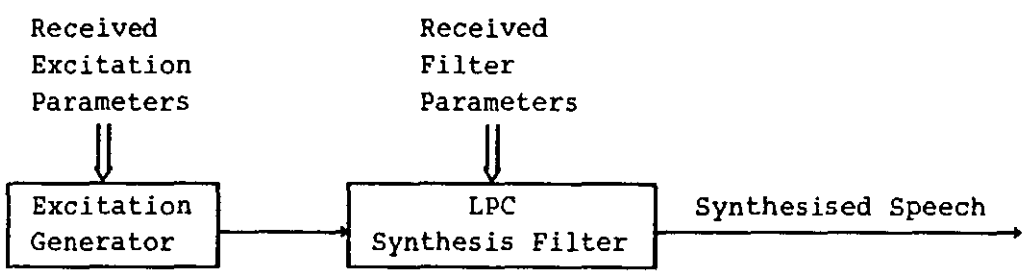
i) The Multipulse Excitation (MPE) coder, which models the excitation signal with a sequence of irregularly spaced pulses [2.74]. The synthesis filter of a MPE coder can be derived from any LPC filter model (see Fig 2.3.1), although the most efficient filter model has been found to be the one that combines two AR filters in series (Fig 2.3.1(d)), the first modeling the smooth spectral envelope and the second (pitch predictor) modeling the harmonic frequency structure of speech [2.75]. The parameters of the excitation (pulse positions and amplitudes) and a part of the synthesis filter (pitch predictor) are determined by the closed-loop optimisation process.

Various algorithms have been proposed for the optimisation of the excitation parameters that are simple enough to permit real-time implementation of the MPE coder [2.76,2.77]. MPE coders produce speech of very good quality at



ENCODER

Fig 2.3.3(a)



DECODER

Fig 2.3.3(b)

FIGURE 2.3.3 The encoder (a) and decoder (b) of a general Analysis-by-Synthesis (AbS) Predictive Coder.

bit rates above 8 kbits/sec, while at 16 kbits/sec they perform better than APC coders [2.78]. Alternative MPE models have been proposed in order to improve the coder's performance at bit rates below 8 kbits/sec [2.79,2.80].

ii) The Regular-Pulse Excitation (RPE) coder, which is very similar to the MPE coder, but models the excitation with a sequence of equally spaced pulses [2.81]. The performance of the RPE coder is very close to that of a MPE coder when they both operate at the same bit rate. Various computationally efficient RPE coding methods have been proposed, and one of them has been chosen as the speech coding standard for the European digital Mobile-Radio system [2.82,2.83].

iii) The Code-Excited LPC (CELP) coder, which selects the excitation signal from a codebook of random Gaussian excitation sequences [2.84]. These random sequences do not exhibit any "pitch" structure (which is necessary during voiced speech), and it is therefore essential to include a pitch predictor in the synthesis filter, in order to "induce" a pitch structure to the excitation waveform. The number of computations required to select the optimum excitation sequence from the codebook can be very large, and various simplified methods have been proposed [2.85,2.86,2.87] that allow the CELP coder to be implemented in real-time. The CELP coder produces speech of very good quality at 8 kbits/sec, while at 4.8 kbits/sec the quality is natural and to a certain degree speaker dependent.

iv) The Backward Excitation Recovery coder, which employs a backward adaptation procedure for the excitation signal, and therefore does not need to transmit any information concerning the excitation to the decoder [2.88]. The parameters of the synthesis filter are determined by the closed-loop optimisation procedure. Various excitation adaptation and filter optimisation algorithms have been proposed, and some of them lead to systems with very small encoding delays (around 3 ms). The performance of BER coders is very close to that of CELP coders at the bit rates of 4.8 and 8 kbits/sec.

The excitation models employed by these four types of AbS coders can be combined to form hybrid systems [2.89,2.90]. This generalisation may also lead to AbS coding systems that decompose the excitation signal into a fixed number of "optimised" excitation waveforms [2.91].

2.4 Sinusoidal Coding

Sinusoidal coders decompose the speech signal into a number of sinusoidal components (sine waves) with time varying amplitudes and frequencies. The speech spectrum is therefore modeled by a time varying Line Spectrum. The model parameters are quantized individually or using a functional representation, and are transmitted to the decoder where the sine-waves are reconstructed and added to form synthetic speech. As most of these sinusoidal coding methods are applied in the frequency domain, special care must be taken to avoid frame boundary discontinuities in the time domain.

The sinusoidal models permit a very accurate representation of speech at high bit rates, but at bit rates lower than 10 kbits/sec model simplifications are used, which affect the coder's performance. An assumption often employed at low bit rates is that the spectral lines are harmonically related, so that coding of the individual frequencies is not required.

Sinusoidal models used in speech coding include :

i) Harmonic Coding, which performs a short-time Fourier analysis of the speech signal and identifies the harmonics of the Line Spectrum model with the aid of a pitch estimator [2.92]. The Line Spectrum is recontructed at the encoder and subtracted from the spectrum of the original speech. The residual spectrum is encoded using Adaptive Transform Coding (ATC) and is transmitted to the decoder together with the amplitudes, phases and pitch estimate. Harmonic coders can produce good communications quality speech at 9.6 kbits/sec. Modifications to the basic model and dynamic quantization strategies have been proposed to improve the performance of harmonic coders at 4.8 and 6 kbits/sec [2.93,2.94,2.95]

ii) Sinusoidal Transform Coding, which determines the time-varying amplitudes and phases of the sine waves from the short-time Fourier analysis of the speech signal [2.96,2.97]. It uses a functional description of the time evolution of the amplitudes and phases of the sinusoidal components. Linear Frequency tracks are constructed in each frame, preserving the continuity between frames and allowing for the "death" of old and "birth" of new frequency tracks. Cubic polynomials are used to provide a "maximally smooth" phase unwrapping and frame boundary continuity. The coder can produce very good quality speech at 8 kbits/sec, and can be modified (by using a harmonic frequency model) to allow operation at 4.8 and 2.4 kbits/sec [2.98,2.99]

iii) Analysis-by-Synthesis Sinusoidal coding, which uses a polynomial representation of the time evolution of the amplitudes and phases of the sinusoidal components, and determines the polynomial coefficients by minimising the energy of an error signal formed by subtracting the synthesised signal from the original speech signal [2.100]. As a closed form solution is not available for all the parameters, an Analysis-by-Synthesis procedure is employed to minimise the energy of the error signal. The minimisation process is constrained to produce "smooth" parameter tracks and preserve the signal continuity at the frame boundaries.

2.5 Sub-band Coding (SBC)

In Sub-band Coding [2.1,2.101,2.102], the speech frequency band is divided into a number of sub-bands (typically between four and sixteen) by a bank of filters. Each sub-band is translated to zero frequency and is sampled at its Nyquist rate. The samples from each sub-band are encoded using APCM or DPCM techniques. At this stage, each sub-band can be encoded using perceptual criteria which are specific to that band. At the receiver, the sub-bands are translated back to their original frequencies and are added to produce the reconstructed speech signal.

By allocating a different number of bits to each band, the variance of the reconsruction error can be separately controlled, and the shape of the overall reconstruction error spectrum can be varied dynamically to reduce the "perceptual" level of distortion. Furthermore, the quantization noise is contained within each band, and leakage from other frequency bands is prevented.

The filter-bank (which is the most complex part of the coder) is implemented using Quadrature Mirror Filters (QMF) [2.103], which eliminate the problem of aliasing. QMF filters have the property that if a full-band signal is passed through the filter-bank and is then decimated to the Nyquist frequency in each sub-band, interpolated back to the original frequency, and resynthesised using the synthesis version of the filter-bank, the resulting signal can be an arbitrarily close replica of the input signal. While the QMF filters cancel aliasing in the absence of quantization, once quantization is introduced this is no longer true.

Sub-Band Coders can produce speech of very good communications quality at

a bit rate of 16 kbits/sec, comparable to the quality obtained from Multipulse Excitation coders at the same bit rate [2.102]. At a bit rate of 9.6 kbits/sec some bands may not be transmitted at all (if their energy content is low) thus affecting the aliasing-cancellation properties of the QMF filters and resulting "whispering" quality, caused by the energy aliased into the spectral gaps. At a bit rate of 4.8 kbits/sec, acceptable quality can still be obtained by dynamically frequency-shifting the speech signal so that the formants align with fixed-frequency bandbass filters [2.104].

2.6 Adaptive Trasform Coding (ATC)

ATC coders transform the speech signal into a spectral representation, and quantize the spectral coefficients using a dynamic bit-allocation strategy [2.105,2.106,2.107]. At the decoder, the quantized coefficients are inverse-transformed back into the time domain.

The transformation most commonly used in speech processing is the Discrete Cosine Trasform (DCT). The DCT transform is closer (in terms of performance) to the "optimal" Karhunen-Loeve Transform (KLT), than the other well known transforms (FFT, WHT, etc.). It is also effective in reducing the frame boundary discontinui- ties, by minimising time aliasing (transfer of energy between the left and right edges of the frame).

The dynamic bit-allocation ensures that the high energy spectral components are quantized as accurately as the low energy components, by distributing the number of bits according to a rough estimate of the spectral envelope. The estimate of the spectral envelope must be transmitted as side information, so that the decoder can determine how the bits were distributed amongst the spectral coefficients at the encoder. Depending on the overall bit rate, some coefficiets may be assigned zero bits, thus creating spectral gaps in the synthesised speech. By adjusting the bit allocation strategy, noise shaping can be achieved and the "subjective" quality of the reconstructed speech can be improved.

The side information is often based on a "smooth-spectrum" estimate that does not include the effect of pitch-induced fine structure in the input spectrum. A consequence of such smoothing is increased zero-bit allocation at high frequencies. By including information on the pitch structure, many low energy components lying between the pitch harmonics at low frequencies

are allocated zero bits, thus releasing bits for the high frequency part of the spectrum. The best results in ATC are obtained using the bit-allocation procedure based on the "unsmoothed-spectrum" estimate.

ATC coders can produce very good communications quality speech at a bit rate of 16 kbits/sec. At a bit rate of 9.6 kbits/sec, the rapid movements of spectral gaps from frame to frame produce a "tonal noise" effect. Highly intelligible and speaker-specific speech quality can be obtained at bit rates between 4 kbits/sec and 8 kbits/sec.

2.7 Conclusions

Speech Coders operating at medium bit rates employ efficient coding techniques and are capable of producing speech of very good quality. Requirements for such coders vary depending on the application. Primary consideration is the speech quality obtained under "transmission" conditions. Such conditions may include tandem encodings and transmission through "noisy" channels. The quality of the recovered speech is measured using various subjective listening tests. Other important properties are the delay and complexity characteristics of the coder.

The coders mensioned in this Chapter can produce very good communications quality speech at 16 kbits/sec. Operation at lower bit rates affect the performance of the coders in different ways. The best results at low bit rates are obtained using Analysis-by-Synthesis Predictive coders.

REFERENCES

[2.1] N.S.Jayant, P.Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video", Prentice-Hall Signal Processing Series, New Jersey 1984

[2.2] D.O'Shaughnessy, "Speech Communication: Human and Machine", Addison-Wesley Publishing Company, 1987

[2.3] M.R.Schroeder, "Vocoders: Analysis ad Synthesis of Speech", IEEE Proceedings, Vol 54, No 5, May 1966, pp 720-734

[2.4] J.L.Flanagan et al., "Speech Coding", IEEE Trans. on Communications, Vol 27, No 4, April 1979, pp 710-736

[2.5] A.Buzo et al., "Speech Coding Based Upon Vector Quantization", IEEE Trans. ASSP, Vol 28, No 5, October 1980, pp 562-574

[2.6] R.M.Gray, "Vector Quantization", IEEE ASSP Magazine, Vol 1, No 2, April 1984, pp 4-29

[2.7] J.Makhoul et al., "Vector Quantization in Speech Coding", IEEE Proceedings, Vol 73, No 11, November 1985, pp 1551-1588

[2.8] V.Cuperman, A.Gersho, "Vector Predictive Coding of Speech at 16 kbits/sec", IEEE Trans. on Communications, Vol 33, No 7, July 1985 pp 685-696

[2.9] R.Viswanathan et al., "Objective Speech Quality Evaluation of Narrow-Band LPC Vocoders", IEEE Proc. ICASSP, 1978, pp 591-594

[2.10] T.Barnwell, "Objective Measures for Speech Quality Testing", Journal of Acoust. Soc. of America, Vol 66, 1979, pp 1658-1663

[2.11] J.Koljonen, M.Karjalainen, "Use of Psychoacoustical Models in Speech Processing: Coding and Objective Performance Evaluation", IEEE Proc. ICASSP, 1984, pp 1.9.1

[2.12] M.E.Hawley, "Speech Intelligibility and Speaker Recognition", Dowen Hutchinson Ross, Stroudsburg, 1977

[2.13] W.R.Daumer, "Subjective Evaluation of several efficient Speech Coders", IEEE Trans. on Communications, April 1982, pp 655-662

[2.14] M.Decina, G.Modena, "CCITT Standards on Digital Speech Processing", IEEE Jounal on Selected Areas in Communications, Vol 6, No2, February 1988, pp 227-234

[2.15] B.Widrow et al., "Adaptive Noise Cancelling: Principles and Applications", IEEE Proceedings, Vol 63, No 12, December 1975, pp 1692-1716

[2.16] D.J.Goodman, R.D.Nash, "Subjective Quality of the same Speech Trans-

mission Conditions in seven different countries", IEEE Trans. on Communications, Vol 30, April 1982, pp 642-654

[2.17] CCITT Study Group XII, "Transmission Quality of Digital Systems", Contribution COM XVI, Annex 2, 1976

[2.18] M.Nakatsui, P.Mermelstein, "Subjective Speech-to-Noise Ratio as a Measure of Speech Quality for Digital Waveform Coders", Journal of Acoust. Soc. of America, Vol 72, 1982, pp 1136-1144

[2.19] H.B.Law, R.A.Seymour, "A Reference Distortion System using Modulated Noise", IEE, Paper No 3992E, November 1962

[2.20] B.S.Atal, "Predictive Coding of Speech at Low Bit Rates", IEEE Trans. on Communications, Vol 30, No 4, Apr 1982, pp 600-613

[2.21] J.D.Gibson, "Adaptive Predictive Coding for Speech", IEEE ASSP Magazine, July 1984, pp 12-26

[2.22] L.R.Rabiner, R.W.Schafer, "Digital Processing of Speech Signals", Prentice-Hall Signal Processing Series, New Jersey

[2.23] J.L.Flanagan et al., "Speech Coding", IEEE Trans. on Communications, Vol 27, No 4, April 1979, pp 710-736

[2.24] B.S.Atal, S.L.Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", Journal of Acoust. Soc. of America, Vol 50, 1971, pp 637-655

[2.25] J.Makhoul, "Linear Prediction: A Tutorial Review", IEEE Proceedings, Vol 63, April 1975, pp 561-580

[2.26] J.D.Markel, A.H.Gray, "Linear Prediction of Speech", Springer-Verlag, New York, 1976

[2.27] E.V.Stansfield et al., "Adaptive Filters in Speech Coding", IEE Proceedings, Vol 134, Pt. F, No 3, June 1987

[2.28] M.L.Honig, D.G.Messerschmitt, "Adaptive Filters: Structures, Algorithms and Applications", Kluwer Academic Publishers, 1984

[2.29] J.D.Gibson et al., "Digital Speech Analysis using Sequential Estimation Techniques", IEEE Trans. ASSP, Vol 23, August 1975, pp 362-369

[2.30] L.Marple, "A New Autoregressive Spectrum Analysis Algorithm", IEEE Trans. ASSP, Vol 28, No 4, August 1980, pp 441-453

[2.31] B.W.Dickinson, J.M.Turner, "Reflection Coefficient Estimation using Cholesky Decomposition", IEEE Trans. ASSP, Vol 27, No 2, Apr 1979, pp 146-148

[2.32] B.R.Musicus, "An Iterative Algorithm for Finding Stable Solutions to the Covariance or Modified Covariance Autoregressive Modeling

Methods", Proc ICASSP 1982, pp 244

[2.33] N.Levinson, "The Wiener RMS Error in Filter Design and Prediction", Journal Math. Physics, Vol 25, 1947, pp 261-278

[2.34] S.M.Kay, S.L.Marple, "Spectrum Analysis - A Modern Perspective", IEEE Proceedings, Vol 69, No 11, November 1981, pp 1380-1419

[2.35] S.Chandra, W.C.Lin, "Experimental Comparison between Stationary and Nonstationary Formulations of Linear Prediction Applied to Voiced Speech Analysis", IEEE Trans. ASSP, Vol 22, Dec 1974, pp 403-415

[2.36] J.Durbin, "The fitting of Time-Series Models", Rev. Inst. Int. Statistics, Vol 28, 1960, pp 233-244

[2.37] B.Friedlander, "Lattice Filters for Adaptive Processing", IEEE Proceedings, Vol 70, August 1982, pp 829-867

[2.38] J.Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction", IEEE Trans. ASSP, Vol 25, Oct 1977, pp 423-428

[2.39] A.H.Gray, D.Y.Wong, "The Burg algorithm for LPC Speech Analysis/ Synthesis", IEEE Trans ASSP, Vol 28, No 6, Dec 1980, pp 609-615

[2.40] J.F.Claerbout, "Fundamentals of Geophysical Data Processing", McGraw-Hill, 1976

[2.41] K.Steiglitz, "On the Simultaneous Estimation of Poles and Zeros in Speech Analysis", IEEE Trans. ASSP, Vol 25, June 1977, pp 229-234

[2.42] H.Morikawa, H.Fujisaki, "Adaptive Analysis of Speech Based on a Pole-Zero Representation", IEEE Trans. ASSP, Vol 30, Feb 1982, pp 77-87

[2.43] D.T.Lee, "Recursive Ladder Algorithms for ARMA Modeling", IEEE Trans. on Automatic Control, Vol AC-27, No 4, August 1982, pp 753-764

[2.44] H.Lev-Ari et al, "Least-Squares Adaptive Lattice and Transversal Filters: A Unified Geometric Theory", IEEE Trans. on Information Theory, Vol IT-30, No 2, March 1984, pp 222-236

[2.45] P.Kabal, R.P.Ramachandran, "Joint Solutions for Formant and Pitch Predictors in Speech Processing", Proc ICASSP 1988, pp 315

[2.46] R.P.Ramachandran, P.Kabal, "Stability and Performance Analysis of Pitch Filters in Speech Coders", IEEE Trans. ASSP, Vol 35, No7, Jul 1987, pp 937-946

[2.47] R.Viswanathan, J.Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems", IEEE Trans. ASSP, Vol 23, No 3, June 1975, pp 309-321

[2.48] A.H.Gray, J.D.Markel, "Quantization and Bit Allocation in Speech Processing", IEEE Trans. ASSP, Vol 24, No 6, Dec 1976, pp 459-473

[2.49] A.H.Gray, et al., "Comparison of Optimal Quantizations of Speech Reflection Coefficients", IEEE Trans. ASSP, Vol 25, No 1, February 1977, pp 9-23

[2.50] J.D.Markel, A.H.Gray, "Implementation and Comparison of Two Transformed Reflection Coefficient Scalar Quantization Methods", IEEE Trans. ASSP, Vol 28, No 6, October 1980, pp 575-583

[2.51] A.Erell et al., "Psychoacoustically Based Scalar Quantization of the LPC Poles", IEEE Proc. ICASSP, April 1988, New York, pp 71-74

[2.52] F.K.Soong, B.Juang, "Line Spectrum Pair (LSP) and Speech Data Compression", IEEE Proc. ICASSP, 1984, pp 1.10.1

[2.53] F.K.Soong, B.Juang, "Optimal Quantization of LSP Parameters", IEEE Proc. ICASSP, New York 1988, pp 394-397

[2.54] N.Sugamura, N.Farvardin, "Quantizer Design in LSP Speech Analysis-Synthesis", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 88, pp 432-440

[2.55] Y.Linde et al, "An Algorithm for Vector Quantizer Design", IEEE Trans on Communications, Vol 28, No 1, January 1980, pp 84-95

[2.56] B.S.Atal, "Stochastic Gaussian Model for Low-Bit Rate Coding of LPC Area parameters", Proc ICASSP 1987, pp 2404

[2.57] A.Veiga, Y.Grenier, "A Multi-Step Model for Speech Parameter Trajectories", IEEE Proc. ICASSP, New York 1988, pp 67-70

[2.58] Y.Shoham, "Vector Predictive Quantization of the spectral parameters for Low Bit Rate Speech Coding", Proc ICASSP 1987, pp 2181

[2.59] M.Yong et al., "Encoding of LPC Spectral Parameters using Switched-Adaptive Interframe Vector Prediction", IEEE Proc. ICASSP, New York 1988, pp 402-405

[2.60] M.Honda, F.Itakura, "Bit Allocation in Time and Frequency Domains for Predictive Coding of Speech", IEEE Trans. ASSP, Vol 32, No 3, June 1984, pp 465-473

[2.61] V.Iyengar, P.Kabal, "A low delay 16 kbits/sec Speech Coder", IEEE Proc. ICASSP, New York 1988, pp 243-246

[2.62] J.Chen, A.Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with adaptive Postfiltering", Proc ICASSP 1987, pp 2185

[2.63] C.K.Un, T.Magill, "The Residual-Excited Linear Prediction Vocoder with Transmission Rate below 9.6 kbits/sec", IEEE Trans. on Comms., December 1975, pp 1466-1474

[2.64] R.Viswanathan et al., "Voice-Excited LPC coders for 9.6 kbps Speech

Transmission", IEEE Proc. ICASSP, 1979, pp 558-561

[2.65] J.Makhoul, M.Berouti, "High-Frequency Regeneration in Speech Coding Systems", IEEE Proc. ICASSP, 1979, pp 428-431

[2.66] C.K.Un, J.R.Lee, "On Spectral Flattening Techniques in Residual-Excited Linear Prediction Vocoding", IEEE Proc. ICASSP, 1982, pp 216

[2.67] H.Katterfeldt, "A DFT-Based Residual-Excited Linear Predictive Coder (RELP) for 4.8 and 9.6 kb/s", IEEE Proc. ICASSP, 1981, pp 824-827

[2.68] R.L.Zinser, "An efficient Pitch-Aligned High-Frequency Regeneration technique for RELP vocoders", Proc ICASSP 1985, pp 969

[2.69] R.Sluyter et al., "A 9.6 kb/s Speech Coder for Mobile Radio Applications", IEEE Proc. ICC, May 1984, pp 1159-1162

[2.70] P.Hedelin, "RELP Vocoding with Uniform and Non-Uniform Downsampling", IEEE Proc. ICASSP, 1983, pp 1320-1323

[2.71] J.P.Adoul et al, "Generalization of the Multipulse coding for Low Bit Rate coding purposes: The Generalized Decimation", Proc ICASSP 1985,

[2.72] B.Fette et al, "Experiments with a high quality, low complexity 4800 bps Residual Excited LPC (RELP) Vocoder", Proc ICASSP 1988, pp 263

[2.73] B.Mazor et al., "Adaptive Subbands Excited Transform (ASET) Coding", IEEE Proc. ICASSP, 1986, pp 3075-3078

[2.74] B.S.Atal, J.R.Remde, "A New Model of LPC Excitation for producing natural-sounding speech at Low Bit Rates", Proc ICASSP 1982, pp 614

[2.75] P.Kroon, E.F.Deprettere, "A Class of Analysis-by-Synthesis Predictive Coders for High Quality speech coding at rates between 4.8 and 16 kbits/sec", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 88, pp 353-363

[2.76] N.Gouvianakis, C.Xydeas, "A Comparative Study of Multistage Sequential and Block Sequential Search Multipulse LPC algorithms", Proc Int. Conf. IASTED, Paris, Jun 1985, pp 122-125

[2.77] A.Fukui, K.Shibagaki, "Implementation of a Multi-Pulse speech codec with Pitch Prediction on a Single Chip Floating-Point Signal Processor", Proc ICASSP 1987, pp 968

[2.78] B.S.Atal, L.R.Rabiner, "Speech Research Directions", AT&T Technical Journal, Vol 65, Issue 5, Sep/Oct 1986, pp 75-88

[2.79] V.Savvides, C.Xydeas, "A new approach to Low Bit Rate Speech Coding", Proc Int. Conf. Digital Processing of Signals in Communications, IERE, Loughborough, Sep 1988

[2.80] S.Ono, K.Ozawa, "2.4 kbps Pitch Prediction Multi-Pulse speech coding"

Proc ICASSP 1988, pp 175

[2.81] P.Kroon et al, "Regular-Pulse Excitation: A novel approach to effective and efficient Multipulse Coding of Speech", IEEE Trans ASSP Oct 1986, pp 1054-1063

[2.82] J.E.Natvig, "Evaluation of six Medium Bit Rate Coders for the Pan-European Digital Mobile Radio System", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 1988, pp 324-331

[2.83] P.Vary et al, "Speech Codec for the European Mobile Radio System", Proc ICASSP 1988, pp 227

[2.84] M.R.Schroeder, B.S.Atal, "Code-Excited Linear Prediction (CELP): High quality speech at very Low Bit Rates", Proc ICASSP 1985, pp 937

[2.85] I.M.Trancoso, B.S.Atal, "Efficient Procedures for finding the Optimum Innovation in Stochastic Coders", IEEE Proc. ICASSP, 1986, pp 2375

[2.86] G.Davidson, A.Gersho, "Complexity Reduction Methods for Vector Excitation Coding", IEEE Proc. ICASSP, 1986, pp 3055

[2.87] D.Lin, "Speech Coding Using Efficient Pseudo-Stochastic Block Codes", IEEE Proc. ICASSP, Dallas 1987, pp 1354-1357

[2.88] N.Gouvianakis, C.Xydeas, "Advances in Analysis by Synthesis LPC Speech Coders", IERE Journal, Supplement on Mobile Radio, Vol 57, No 6, Nov/Dec 1987, pp S272-S286

[2.89] R.C.Rose, T.P.Barnwell, "Quality Comparison of Low Complexity 4800 bps Self Excited and Code Excited Vocoders", Proc ICASSP 1987,pp 1637

[2.90] T.V.Sreenivas, "Modelling LPC-Residue by components for good quality speech coding", Proc ICASSP 1988, pp 171

[2.91] G.Davidson, A.Gersho ,"Multiple-Stage Vector Excitation coding of speech waveforms", IEEE Proc. ICASSP, New York 1988, pp 163

[2.92] L.Almeida, J.Tribolet, "Nonstationary Spectral Modeling of Voiced Speech", IEEE Trans. ASSP, Vol 31, 1983, pp 664-678

[2.93] E.C.Bronson et al., "Harmonic Coding of Speech at 4.8 kbits/sec", IEEE Proc. ICASSP, 1987, pp 2213-2216

[2.94] I.M.Trancoso et al, "Quantization issues in Harmonic Coders", IEEE Proc. ICASSP, New York 1988, pp 382

[2.95] D.L.Thomson, "Parametric Models of the Magnitude/Phase Spectrum for Harmonic speech coding", Proc ICASSP 1988, pp 378

[2.96] R.J.McAulay, T.F.Quatieri, "Speech Analysis/Synthesis based on a Sinusoidal Representation", IEEE Trans. ASSP, Aug 1986, pp 744-754

[2.97] T.F.Quatieri, R.J.McAulay, "Speech Transformations based on a Sinu-

soidal Representation", IEEE Trans. ASSP, Dec 1986, pp 1449-1464

[2.98] R.J.McAulay, T.F.Quatieri, "Multirate Sinusoidal Transform Coding at Rates from 2.4 kBps to 8 kBps", IEEE Proc. ICASSP, 1987, pp 1645

[2.99] R.McAulay, T.F.Quatieri, "Computationally efficient Sine-Wave Synthesis and its application to Sinusoidal Transform Coding", IEEE Proc. ICASSP, New York 1988, pp 370

[2.100] E.B.George, M.J.T.Smith, "A new speech coding model based on a Least Squares Sinusoidal representation", Proc ICASSP 1987, pp 1641

[2.101] R.Crochiere, "On the design of Sub-band Coders for Low Bit Rate Speech Communication", Bell Syst. Tech. J., Vol 56, 1977, pp 747-770

[2.102] R.Cox et al., "New Directions in Subband Coding", IEEE Journal on Selected Areas in Comms., Vol 6, No 2, Feb 1988, pp 391-409

[2.103] R.Crochiere, L.Rabiner, "Multirate Digital Signal Processing", Prentice-Hall, Englewood Cliffs, 1983

[2.104] R.Crochiere, M.Sambur, "A variable-band Coding scheme for speech encoding at 4.8 kb/s", Bell Syst. Tech J, Vol 56, 1977, pp 771-780

[2.105] R.Zelinski, P.Noll, "Adaptive Transform Coding of Speech Signals", IEEE Trans. ASSP, Vol 25, No 4, August 1977, pp 299-309

[2.106] J.Tribolet, R.Crochiere, "Frequency Domain Coding of Speech", IEEE Trans. ASSP, Vol 27, 1979, pp 512-530

[2.107] R.Crochiere et al., "Real-Time Speech Coding", IEEE Trans. on Comms. April 1982, pp 621-634

CHAPTER 3

MULTIPULSE EXCITATION SPEECH CODING

3.1 Introduction

The design of a Multipulse Excitation (MPE) codec and its application to speech compression, was first introduced by B.S.Atal in 1982 [3.1]. A number of concepts were combined to create a powerful coding technique, which has since been applied to the development of many diverse speech coding schemes, such as the Code Excited LPC [3.2] and Backwards Excitation Recovery [3.3] techniques

The encoder section of a MPE codec, controls the output of a LPC speech synthesiser by systematically adjusting its internal parameters in order to produce a close match between the synthesised and original speech waveforms. The internal parameters of the LPC synthesiser are transmitted to the decoder, which repeats the speech synthesis process and recovers the speech waveform.

In order to achieve an efficient coding operation, information from both the past and the future of the speech signal is needed at every instant. The encoder has to delay decisions concerning the adjustmemt of the LPC synthesiser, until enough speech samples have been received. The same delay characteristics are common amongst speech coding schemes that achieve a high compression of the input speech data. The term Analysis by Synthesis is used to describe both the Delayed Decision Coding attribute, and the operation which optimises the coder's internal parameters in order to produce a close approximation of the original speech waveform.

Another concept used in the design of the MPE codec is that of noise masking [3.4]. The distortion introduced by the coder should ideally have a power spectral distribution that would minimise its perceptibility in the presence of speech. In practice this is hard to achieve, because of the dynamic behaviour of speech and the difficulty in determining the ideal noise spectrum. It has been established though, that a noise-like signal can be masked by a high power correlated source over the same spectral region. As a result, a higher level of noise can be tolerated in the spectral regions where there is concentration of speech energy (i.e. in the formants).

In a MPE coder, it is possible to have a certain amount of control over the spectral content of the distortion introduced, by formulating an appropriate error measure between the original and synthesised speech waveforms. This allows the continuous and adaptive adjustment of the noise spectrum and reduces the perceptible distortion.

The ability of the MPE system to adjust and optimise its performance according to a predefined speech quality criterion, allows a great deal of control over the design and implementation aspects of the MPE coder. The MPE coder can be quite robust in various acoustic environments, because it does not depend on voicing decisions (classification of signals as periodic or noise-like). Furthermore, it can operate, if necessary, without a long-term predictor, thus limiting the error accumulation in the presence of transmission errors.

The complexity of MPE codecs, or other Analysis by Synthesis speech coding schemes, is not prohibitive by today's standards in VLSI design. The tremendous increase in power of the monolithic Digital Signal Processors has decimated the development and implementation costs of complex speech codecs and has facilitated their widespread use [3.5,3.6,3.7,3.8]. The ever broadening applications field has also played an important role in the diversification of the research aims and the setting of new standards and goals [3.9]. Commercial interest in high quality speech transmission and the need to exploit the available bandwidth in applications such as mobile radio and satellite communications [3.10], has been one of the driving forces in the design of new and efficient speech codecs.

MPE codecs are capable of toll quality speech at 16 kbits/sec, while at 9.6 kbits/sec the obtained speech quality compares favourably with that obtained from conventional codecs like RELP or subband [3.11]. MPE coding has also been applied to wideband speech transmission at 32 and 16 kbits/sec [3.8], and a quite respectable coded speech quality has been demonstrated at a transmission rate of 2.4 kbits/sec [3.12].

In this chapter, the basic principles behind the MPE coding schemes will be first described. A review of the existing techniques of pulse amplitude estimation and pulse position optimisation will then be presented. Many of these techniques have been borrowed from the mathematical field of numerical optimisation, while others have been designed specifically for the solution

of the MPE optimisation problem. The latter techniques are usually simpler, and will be examined in more detail in the next two chapters.

3.2 Definition of the Multipulse Excitation

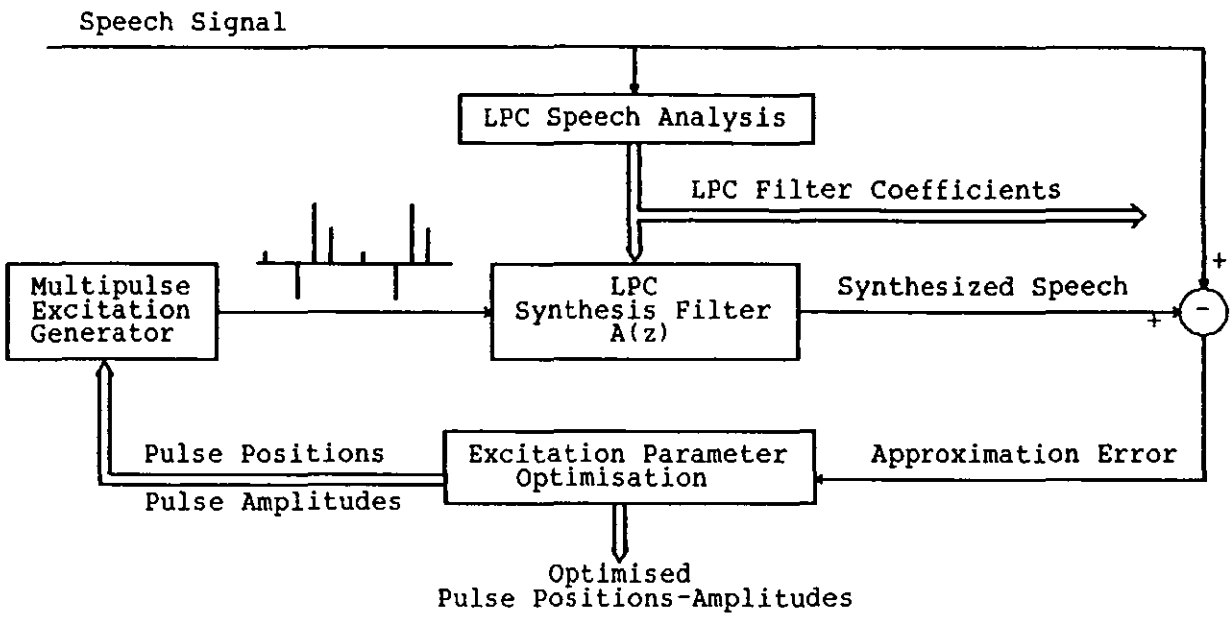
Figure 3.2.1 shows an operational diagram of the Multipulse Excitation Encoder-Decoder. The optimisation loop of the encoder section indicates that a Delayed Decision Coding process is taking place and that the input to the LPC filter is not defined sequentially in time, as the signal flow might suggest, but in a batch mode. It also indicates the repetitive nature of the excitation optimisation process.

The speech waveform is first divided into consequtive frames, each containing n speech samples. The LPC filter is then derived directly from the speech data (using one of the Linear Prediction techniques usually employed by LPC speech coders), and the Multipulse Excitation sequence is optimised so that the output of the LPC synthesis filter closely approximates the original speech waveform. The spectral distribution of the distortion introduced by the coder can be forced to approximate a given distribution, by properly adjusting the excitation optimisation process.

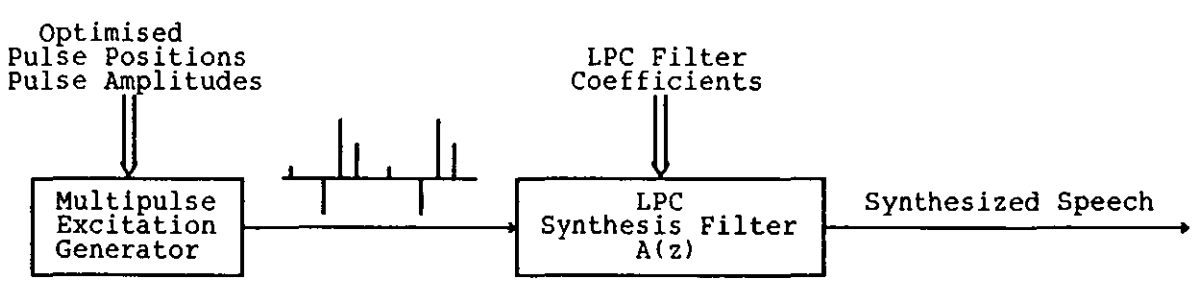
The input excitation sequence (MPE) is formed using a small number of impulses, whose amplitudes and positions within each frame, are determined by the Excitation Parameter Optimisation algorithm.

The process of defining the MPE sequence can be described as follows. Consider a frame of n speech samples, represented by the n-dimensional vector $\mathbf{s}^T = [\, s(0), s(1), s(2), \ldots, s(n-1) \,]$ and the excitation vector corresponding to the same frame $\mathbf{x}^T = [\, x(0), x(1), \ldots, x(n-1) \,]$. Assuming a fixed number of pulses q for each frame, the pulses are located at sample instances $p_1, p_2, \ldots, p_q$, and have amplitudes $b_1, b_2, \ldots, b_q$ respectively. For example, if $n=8, q=3, p_1=3, p_2=2$ and $p_3=6$ then $\mathbf{x}^T = [\, 0, 0, b_2, b_1, 0, 0, b_3, 0 \,]$.

The filter used by the coder can be a general linear filter, but usually it takes the form of a single Autoregressive (AR) filter which models the combined effect of the glottal shape, vocal tract response and lip radiation. A second AR filter can be introduced (in series with the first), which models the quasi-periodic nature of the speech waveform [3.13], or a more complex ARMA filter model can be used instead of the standard AR model [3.14,3.15]. When a single AR filter model is considered, LPC methods like



MULTIPULSE EXCITATION ENCODER



MULTIPULSE EXCITATION DECODER

FIGURE 3.2.1 The Multipulse Excitation (MPE) coder

the stabilised covariance [3.16,3.17] or the maximum entropy method [3.18], can be used to estimate the filter coefficients.

The response of the LPC synthesis filter to an input sequence $\{x(i)\}$ is $\{x(i)\}*\{h(i)\}$,where $\{h(i)\}$ is the filter's impulse response. The convolution of the two time series can be considered as a matrix multiplication operation. The filter's response s_y is expressed as :

$$\mathbf{s}_y = \mathbf{A}\,\mathbf{x} \qquad (Eq\ 3.2.1)$$

where $\mathbf{A}$ is the nxn lower triangular convolution matrix :

$$\mathbf{A} = \begin{bmatrix} h(o) & o & o & \cdots & o \\ h(1) & h(o) & o & & o \\ h(2) & h(1) & h(o) & & o \\ \vdots & \vdots & \vdots & & \vdots \\ h(n-1) & h(n-2) & h(n-3) & \cdots & h(o) \end{bmatrix} \qquad (Eq\ 3.2.2)$$

The input vector $\mathbf{x}$ contains both zero and non-zero elements, and Eq 3.2.1 can be simplified if a new nxq matrix $\mathbf{A}[q]$ is formed using only those q columns of matrix $\mathbf{A}$ which correspond to the non-zero elements of $\mathbf{x}$. These columns are arranged in the order given by the position indexes $p_1,p_2,..,p_q$:

$$\mathbf{A}[q] = \begin{bmatrix} o & h(o) & \cdots & o \\ o & h(1) & & o \\ \vdots & h(2) & & \vdots \\ & & & h(o) \\ h(o) & & & h(1) \\ h(1) & \vdots & & h(2) \\ \vdots & & & \vdots \\ h(n-p_1-1) & h(n-p_2-1) & \cdots & h(n-p_q-1) \end{bmatrix} \qquad (Eq\ 3.2.3)$$

Eq 3.2.1 is then transformed to :

$$\mathbf{s}_y = \mathbf{A}[q]\,\mathbf{b} \qquad (Eq\ 3.2.4)$$

where $\mathbf{b}^T = [\, b_1,b_2,\ldots,b_q \,]$ is the q-dimensional vector of the pulse amplitudes.

The pulse positions $p_1,p_2,\ldots,p_q$ and pulse amplitudes $b_1,b_2,\ldots,b_q$ should be determined so that the response s_y of the LPC filter to the generated multipulse input signal closely approximates the original speech sequence $\mathbf{s}$.

A parameter estimation problem can be formed by assuming an additive noise

model for the speech signal :

$$\mathbf{s} = \mathbf{A}[q]\,\mathbf{b} + \mathbf{e}_a \qquad (Eq\ 3.2.5)$$

$\mathbf{A}[q]\,\mathbf{b}$ is a function of the $2q$ random variables that correspond to the pulse positions and amplitudes, and $\mathbf{e}_a$ is a sample from a random noise process whose statistical properties are known. These two random parts of the model are assumed to be uncorrelated.

A good estimate of the parameter values (pulse positions-amplitudes) would be the one that maximises the a posteriori probability of the set of the parameter values, given the observed sample waveform s (Bayesian estimate) [3.19]. The continuous-discrete nature of this conditional probability density function means that a numerical approach would be necessary in order to locate its maximum. A different treatment of the two sets of variables can therefore be advantageous, since a semi-analytical solution for the pulse amplitudes and a numerical solution for the pulse positions can be sought.

For a fixed set of pulse positions, by disregarding any knowledge about the pulse amplitudes' joint-probability-distribution and by assuming a zero-mean Gaussian noise process, an analytical solution in the form of the Maximum-Likelihood estimate, can be found for the pulse amplitudes. The same estimate can be obtained from the theory of Least Squares (LS) estimation, since the noise statistics are assumed to be Gaussian.

Having obtained an analytical solution for the amplitudes of a set of pulses with fixed positions, the task of defining the optimum multipulse excitation sequence is converted into an error function minimisation problem. The function variables (pulse positions) can only take discrete values, therefore integer programming [3.20] or other simpler iterative optimisation techniques can be used to define the pulse positions.

3.3 Estimation of the pulse amplitudes

When the pulse positions $p_1, p_2, \ldots, p_q$ are fixed, the matrix $\mathbf{A}[q]$ is also fixed and the values of the pulse amplitudes $b_1, b_2, \ldots, b_q$ can be determined by minimising the power of the noise (approximation error) $\mathbf{e}_a$ in Eq 3.2.5. In the general case where the noise samples are correlated, a linear transformation can be applied to produce an equivalent estimation problem

where the noise samples are uncorrelated. The noise covariance matrix C_a is known in advance, and is assumed to be positive definite :

$$C_a = E\left[e_a \; e_a^T \right] \qquad (Eq\ 3.3.1)$$

It can therefore be factorised as :

$$C_a = L \; L^T \qquad (Eq\ 3.3.2)$$

where L is a lower triangular square root of C_a. Application of the transformation σL^{-1} (where σ is a constant) to Eq 3.2.5, generates a new set of equations :

$$s_w = A_w[q] \; b + e_w \qquad (Eq\ 3.3.3)$$

where : $s_w = \sigma L^{-1} s$, $A_w[q] = \sigma L^{-1} A[q]$ and $e_w = \sigma L^{-1} e_a$

The noise samples are now uncorrelated and the new noise covariance is :

$$C_w = E\left[e_w \; e_w^T \right] = \sigma^2 \; L^{-1} E\left[e_a \; e_a^T \right] L^{-T} = \sigma^2 \; I_n \qquad (Eq\ 3.3.4)$$

where I_n is the nxn identity matrix.
The noise energy over a specified frame is minimised when :

$$\nabla_b \left[e_w^T \; e_w \right] = [\, o, o, \ldots, o \,]^T \qquad (Eq\ 3.3.5)$$

or

$$\nabla_b \left[\left\| s_w - A_w[q] \; b \right\|^2 \right] = [\, o, o, \ldots, o \,]^T \qquad (Eq\ 3.3.6)$$

Eq 3.3.6 defines the LS problem and as long as the matrix $A_w[q]$ has a full rank, a unique solution exists. A number of methods with good numerical properties can be employed to find the solution. The Gram-Schmidt orthogonalization procedure or the Cholesky matrix decomposition algorithm [3.21] are commonly used. The latter is used to solve the system of normal equations derived from Eq 3.3.6 :

$$\left(A_w[q]^T \; A_w[q] \right) b = A_w[q]^T \; s_w \qquad (Eq\ 3.3.7)$$

A geometrical interpretation of this minimisation problem is possible by visualizing the signal components as vectors in an n-dimensional space. The columns of the matrix $A_w[q]$ define a q-dimensional subspace and $A_w[q]$ b

belongs to the same subspace. In Eq 3.3.6 therefore, the function which is minimised is equal to the square of the distance between the speech vector $\mathbf{s}_w$ and the vector $\mathbf{A}_w[q]\,\mathbf{b}$. This distance is minimum when $\mathbf{A}_w[q]\,\mathbf{b}$ coincides with the projection of $\mathbf{s}_w$ on the subspace defined by $\mathbf{A}_w[q]$.

Equation 3.3.7 can be solved for the component values $b_1, b_2, .., b_q$ that ensure the orthogonality between the error vector $\mathbf{e}_w$ and the signal vector $\mathbf{A}_w[q]\,\mathbf{b}$ (hence the name normal equations). It can be shown that the projection operator takes the form :

$$\mathbf{P}_r = \mathbf{A}_w[q] \left(\mathbf{A}_w[q]^T \mathbf{A}_w[q]\right)^{-1} \mathbf{A}_w[q]^T \qquad (Eq\ 3.3.8)$$

and that the minimum error energy is :

$$\left[\ \mathbf{e}_w^T\ \mathbf{e}_w\ \right]_{min} = \left(\mathbf{s}_w - \mathbf{P}_r \mathbf{s}_w\right)^T \left(\mathbf{s}_w - \mathbf{P}_r \mathbf{s}_w\right) = \mathbf{s}_w^T\ \mathbf{s}_w - \mathbf{b}^T\ \mathbf{A}_w[q]^T\ \mathbf{s}_w \qquad (Eq\ 3.3.9)$$

As the pulse positions $P_1, P_2, \ldots P_q$ change, so does the subspace defined by $\mathbf{A}_w[q]$. Ultimately the subspace closest to the speech vector $\mathbf{s}_w$ will identify the best set of values for the pulse positions.

3.4 Optimising the pulse positions

The estimation of an optimum Multipulse Excitation sequence has been transformed to an equivalent problem of minimising the distance between two n-dimensional vectors corresponding to the original and LPC-synthesised waveforms (Eq 3.3.6), subject to the coordinate transformation $\sigma\mathbf{L}^{-1}$ that produces a new set of axes corresponding to the column vectors of matrix $\mathbf{A}_w$:

$$\mathbf{A}_w = \sigma\mathbf{L}^{-1}\,\mathbf{A} \qquad (Eq\ 3.4.1)$$

The minimum distance ε (error RMS) for a given set of pulse positions $P_1, P_2, \ldots, P_q$ (corresponding to a particular subspace), can be found using the analytical formula of Eq 3.3.9. This minimum distance changes when a different set of pulse positions is considered and can be expressed as a function of the integer variables $P_1, P_2, \ldots, P_q$:

$$\varepsilon(P_1, P_2, \ldots, P_q) = \left\|(\mathbf{I}_n - \mathbf{P}_r)\mathbf{s}_w\right\| \geq 0 \quad , [P_1, P_2, \ldots, P_q] \in \Psi \quad , \Psi \subset \mathbb{Z}^q \qquad (Eq\ 3.4.2)$$

The projection operator $\mathbf{P}_r$ refers to the subspace defined by the columns

of matrix $\mathbf{A}_w[q]$ which in turn is dependent on the values of $p_1, p_2, \ldots, p_q$, Ψ is the set of permissible combinations of values that $p_1, p_2, \ldots, p_q$ may take (the order is not important), and Z^q is the set of all q-dimensional integer vectors.

Enumeration of all the vectors $[p_1, p_2, \ldots, p_q] \in \Psi$ is only feasible when Ψ is relatively small, as is the case when a Codebook-Search algorithm is used to find the pulse positions (see Chapter 5), or when a Regular-Pulse Excitation sequence is postulated [3.22]. If there is complete freedom in the choice of pulse positions, then Ψ has $\binom{n}{q}$ elements and complete enumeration is only possible when the frame size n and the number of pulses per frame q are small. Since Multipulse Excitation coders usually benefit from the use of large frames, other methods are employed to find the pulse positions that minimise the distance between the original signal and the responce of the LPC filter.

Two broad classes of pulse position estimation methods can be defined, to highlight the differences between Successive Elimination and Multivariate Optimization techniques. These differences can sometimes be subtle but in general, Successive Elimination techniques are more deterministic in their approach than Multivariate Optimisation techniques, which may adopt a probabilistic search strategy for the pulse positions.

3.4.1 Successive Elimination Techniques

These methods progressively decompose the parameter "space" Ψ into increasingly smaller subsets. The minimum of the approximation error function $\varepsilon(p_1, p_2, \ldots, p_q)$ is bracketed by the subset boundaries and becomes more and more localised as the subsets shrink and finally reduce to single vectors. This is done in a systematic way by restricting the range of values over which each of the position variables $p_1, p_2, \ldots, p_q$ is allowed to vary. A repetitive evaluation of the approximation error function is performed and decisions concerning the subdivision of the subsets are taken based on the past history of computed function values.

The complexity of the process is determined by the number of iterations necessary to converge to a solution, and by the effort needed to compute the approximation error function. In the following examples, a general definition of this function is implied, to accomodate the various simplified

amplitude-estimation algorithms developed in the next chapter. Temporarily the function $\epsilon(p_1,p_2,\ldots)$ will be assumed to provide a measure of the approximation error when a number of pulse positions are specified, without necessary implying the use of Eq 3.3.9.

Examples of Successive Elimination techniques applied to the estimation of the pulse positions are :

1) COMBINATORIAL SEARCH

Combinatorial search methods perform a search through an imagined tree of pulse position combinations. The tree is set up so that each path along its branches corresponds to a single set of pulse positions (Fig 3.4.1).

The search for the optimum set of pulse positions starts at the first level where n branches diffuse from a single parental node. These branches correspond to the possible locations of the first pulse. Different criteria can be used to select the most "promising" paths through the tree. For example a threshold ρ_1 can be used to control the selection of the branches which satisfy the requirement :

$$\epsilon(p_1) < \rho_1 \qquad , \rho_1 \in R \qquad \textit{(Eq 3.4.1.1)}$$

or alternatively, n_1 branch candidates can be chosen that correspond to the n_1 lowest values of the approximation error function $\epsilon(p_1)$, $o \leq p_1 \leq n-1$.

The tree expands at the second level by appending $n-1$ branches at the end of every branch that was selected at the first level, to accomodate the $n-1$ possible locations of the second pulse. A new threshold ρ_2 or a new integer constant n_2 can be used to control the search at this level, where the function $\epsilon(p_1,p_2)$ is evaluated. The process continues until finally at the qth-level, the path associated to the lowest error value is chosen. Care must be taken to recognise the paths that represent the same set of pulse positions and reject all but one.

Careful choice of the values $\rho_1,\rho_2,\ldots,\rho_q$ or $n_1,n_2,\ldots,n_q$ can improve the efficiency of the method and reduce the number of required error-function evaluations. Note that the complexity of the algorithm depends on the kind of approximation error function that is chosen to be minimised at each stage, and is therefore dependent on the implied difficulty in obtaining estimates of the pulse amplitudes.

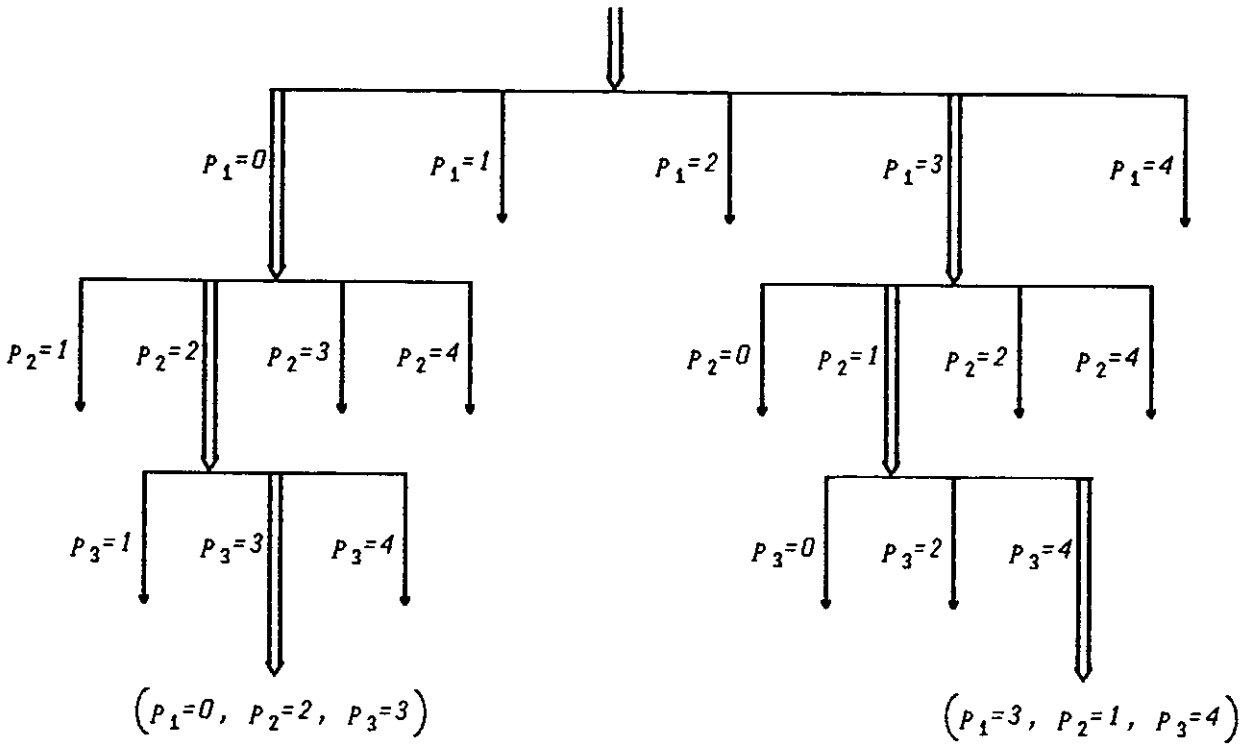


FIGURE 3.4.1 Combinatorial-Search MPE Optimisation Method. A Tree Search for the optimum set of pulse positions in a MPE frame of 5 samples is attempted. Each branch corresponds to a possible set of pulse positions, and two branches are chosen at each level of the tree.

The number of paths selected at each level will in general be greater than at the previous level and can increase substantially by the time the last stage is reached. To keep this number low, a process can be employed at the end of each stage to reject the paths associated to high approximation error values. A variant of this method, only keeps a fixed number of paths per stage (Fig 3.4.1 shows 2 paths) and these paths are the ones with the lowest error values [3.23]. If this number is kept equal to one, then the position found at each stage cannot be altered by further stages (this property is common to all the algorithms of Example-3).

2) BRANCH AND BOUND OPTIMISATION

The Branch and Bound technique is a nonlinear programming method [3.24, 3.25] and is one of the possible combinatorial optimisation methods that can be applied to the MPE optimisation problem [3.26].

The Branch and Bound technique can be used to perform a recursive binary subdivision of the parameter space (of the pulse-position variables). This subdivision can be explained using a binary decision tree whose branches are inclusively related. The parameter set Ψ is broken down into subsets in such a way that the subset corresponding to each branch contains the subsets of the descending branches (Fig 3.4.2). Each decision (branching) splits the interval, over which a single position variable is defined, into two.

A lower and an upper bound of the error-function are estimated for each "active" path along the tree. Each path associated with a lower bound which is greater than an established upper bound elsewhere within the tree, is not worth pursuing and is therefore "deactivated" to facilitate a faster search through the rest of the tree.

In a function minimisation problem, accurate estimation of the lower bound is crucial because it indicates the potential gain obtained by following a certain (search) path. Unfortunately in this case, a lower bound of the MPE approximation error function is difficult to calculate, and a guess has to be made based on the value of the corresponding upper bound (which is easier to find).

Since each binary decision concerns a single pulse, a simple enumeration of the permissible pulse positions can provide the minimum approximation error value. This value is also an upper bound of the error function because

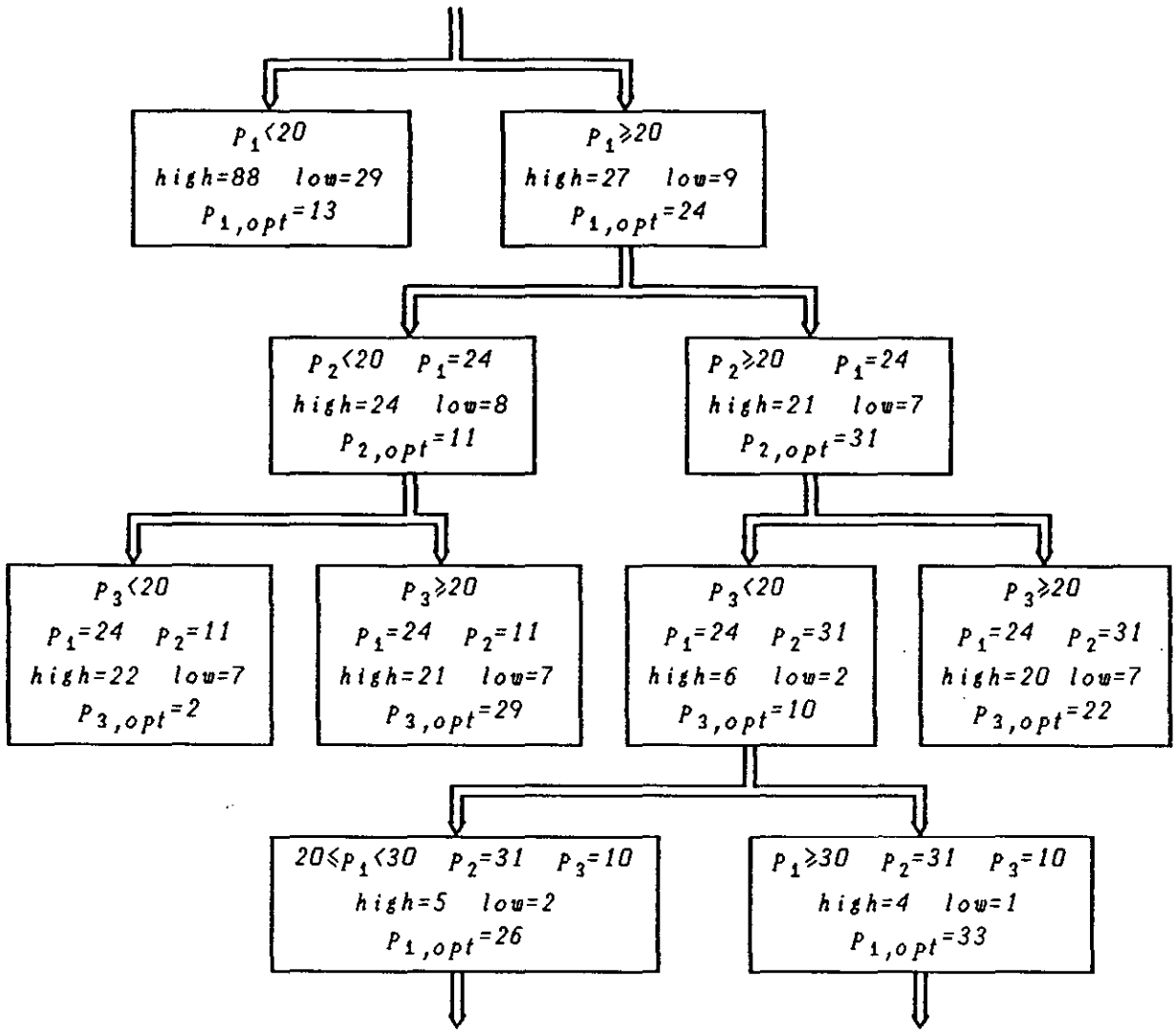


FIGURE 3.4.2 Branch-and-Bound MPE Optimisation Method. Upper error bounds are established by varying the position of a single pulse at each level of the tree and by selecting the position where the error is minimum. The lower bounds are set to one-third the value of the upper bound. The binary sub-division continues until all the pulse positions are fixed.

subsequent steps along the same path can only reduce the error. In Fig 3.4.2 the value of the upper bound (high) for each path corresponds to the minimum of the error-function and is calculated using an exhaustive enumeration of the permissible pulse positions. In this example, the lower bound (low) is set equal to one third of the upper bound. Before the next decision (branching) is taken, the pulse is placed at the position where the minimum error occured. The process shrinks the subsets into single vectors, and terminates by choosing the vector of pulse positions that results the minimum approximation error.

The method can be considerably simplified by reducing the number of active paths and by using the simplified procedures described in the next chapter, to estimate the pulse amplitudes and the approximation error.

3) MULTI-STAGE (MS) OPTIMISATION

Multi-Stage optimisation algorithms include some of the most popular and simple MPE optimisation methods [3.27,3.28]. Each method starts by optimising the position of the first pulse and continues by adding further pulses in the next stages, until the required number of q pulses per frame is reached. The position of each new pulse is optimised in a separate stage and cannot be redefined in the next stages, even though the pulse's amplitude may be corrected.

Differences between the various MS algorithms are related to the degree of involvement of the pulse amplitude estimation process. The computational effort involved in the calculation of the pulse amplitudes and the estimation of the approximation error determines the complexity of the MS-algorithm, since the particular pulse position optimisation strategy only allows a limited number of pulse position combinations to be considered. Various MS algorithms will be examined in Chapter 4.

A different MS optimisation procedure would initially consider a complete set of n pulses and would then reduce them to the required number using a thinning process [3.29,3.30]. The same search logic as before can be applied in this case, if the position of a "vacant" pulse (hole) is optimised at each stage, instead of an actual pulse.

3.4.2 Multivariate Optimisation Techniques

As the name implies, these techniques adopt a more probabilistic approach to find an optimum set of pulse positions. The parameter "space" Ψ is irregularly sampled, and information obtained from the sampling operation is used to direct the search for the minimum of the approxiamtion error function. Successive samples create paths in Ψ, which can be progressively constrained, not in an absolute manner, but in the sense that the probability of sampling a point outside a confinement subset (or subsets) in Ψ becomes increasingly small.

Examples of Multivariate Optimisation techniques are :

1) SIMULATED ANNEALING

This technique simulates the cooling process of a liquid substance, at the molecular level. The physical cooling process reduces the total (kinetic and potential) energy of the molecules. The total energy can be considered as a function of the distribution of the molecular quantum energies. A temperature drop results a reduction of the total energy, brought by the general tendency of the molecules to drop to lower energy levels. These energy transitions are random and can lead to higher, as well as lower energy levels (for each molecule). A slow cooling process reduces the total energy until a global energy-minimum is reached at the crystalline state. If the cooling process is accelerated, crystal deformations will appear and the energy minimum will only correspond to a local minimum.

This "slow" cooling process can be imitated by a function minimisation algorithm which attempts to locate the global minimum of a multivariate error-function (which corresponds to the total energy of the physical model) [3.31]. The algorithm constructs a search path in the parameter space of the error-function, in a series of optimisation steps. At each optimisation step, random deviations from the last point of the search path are generated, and these deviations are considered as possible "transitions" in the values of the function variables. When a "successful" transition occurs, a new point is added to the search path, and the process continues by considering random deviations from the new point.

A simple model is used to measure the probability of a "successful" transition. The model assumes that this probability is related to the amount

by which the function value changes when the transition occurs, and depends on the value of a parameter T (which corresponds to the temperature of the physical substance). Consider an error-function $E(\mathbf{p})$, where $\mathbf{p}$ is the set of function variables. The conditional probability of a transition being successful depends on the corresponding change in the function value $\delta E(\mathbf{p})$, and is modeled by the exponential function :

$$P\left[\begin{matrix}Successful\\Transition\end{matrix} / \delta E(\mathbf{p}), T\right] = \begin{bmatrix} exp(-\delta E(\mathbf{p})/kT) & , \delta E(\mathbf{p})>0 \\ 1 & , \delta E(\mathbf{p})<0 \end{bmatrix} \quad (Eq\ 3.4.2.1)$$

This model assumes that a transition is always successful if it is accompanied by a reduction of the function value ($\delta E(\mathbf{p})<0$). A transition that increases the value of the error-function ($\delta E(\mathbf{p})>0$) is considered as successful with a probability derived from the model of Eq 3.4.2.1. A high value of T results an equal probability of "accepting" an increase or a decrease of the function value, but a small value of T favours the transitions which result lower function values. The optimisation strategy is therefore to keep the value of T high initially, and then gradually reduce it to a value close to zero. As the value of T is reduced, the probabilty of "accepting" an increase in the error-function's value becomes smaller, and the function values become more and more localised around the local minima. Finally, when the value of T is close to zero (corresponding to a temperature of absolute zero), the global function minimum is reached and no transitions occur.

Results from a two-dimensional optimisation problem are shown in Figure 3.4.3(a). The error-function considered, has 5 local minima at points (1,1), (1,-1), (-1,1), (-1,-1) and (0,0). The function value at these points is 1,1,1,1 and .999 respectively. Thus, the point (0,0) is where the global function minimum occurs. The distance of each point added to the search path during the optimisation process, from the point (0,0), is plotted while the value of the parameter T is lowered. Sudden transitions can be obsereved even when the value of T is small, and this indicates the existence of other local minima. Finally only small perturbations around the point of global minimum (0,0) are observed.

The Simulated Annealing algorithm is simple but may involve a large number of function computations. It has been successfully applied to a number of

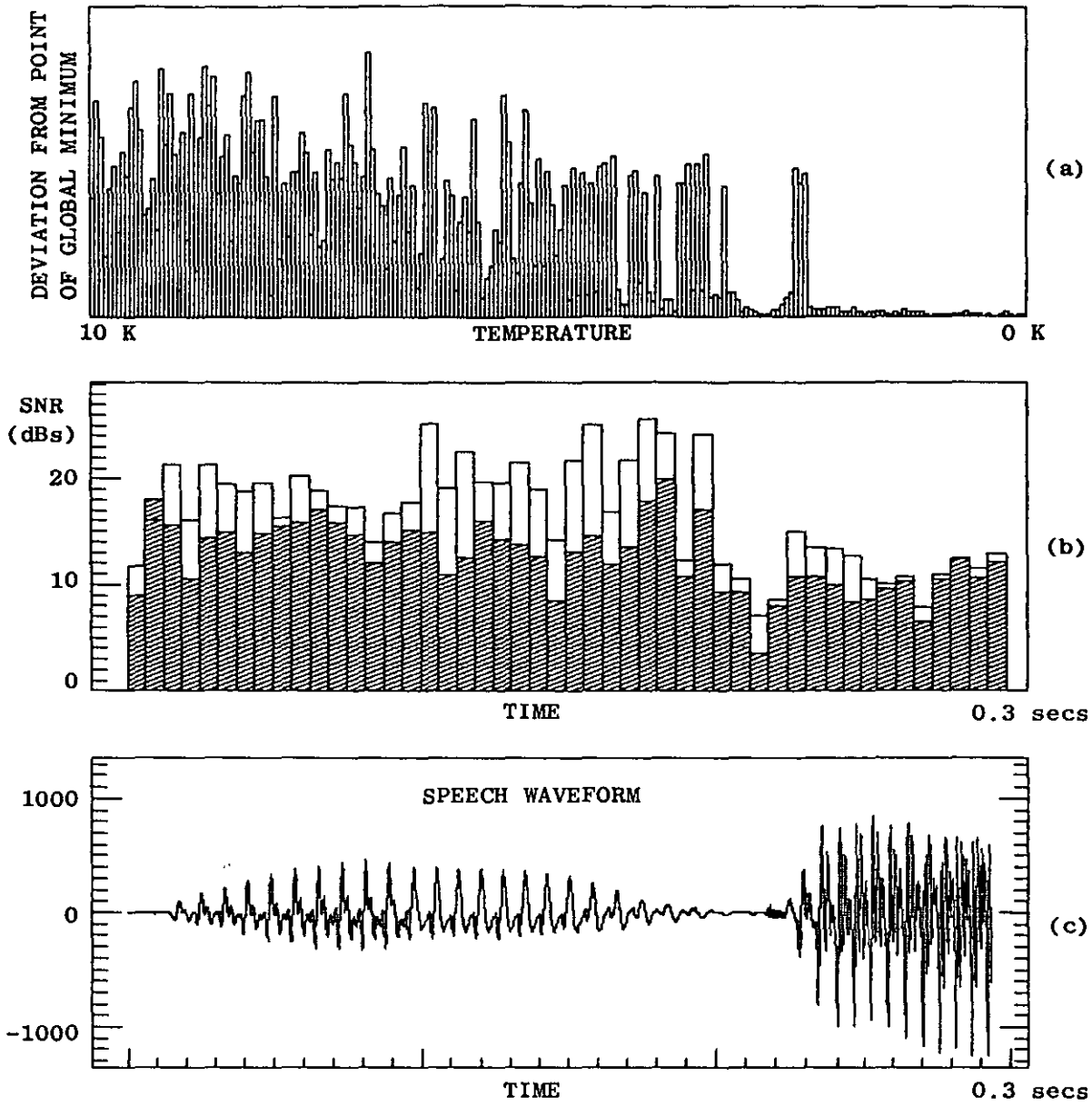


FIGURE 3.4.3 Simulated-Annealing MPE Optimisation Method.
(a) Two-dimensional Function Minimisation. The deviation from the point (0,0) where the function minimum occurs, is plotted as the temperature variable decreases.
(b) SNR versus time, for the speech waveform shown in (c). Black squares correspond to the Maximum-Crosscorrelation MPE algorithm, and white squares correspond to the Simulated-Annealing MPE algorithm.

combinatorial optimisation problems [3.32] and can give good results when used to optimise the pulse positions in a MPE coder [3.33]. In the case of the MPE system, the error-function corresponds to the energy of the approximation error (difference between the original and synthesised speech waveforms), and the transitions correspond to pulse movements.

The SNR obtained when a Simulated Annealing MPE algorithm is applied to a short voiced speech segment of 48 frames, is shown in Fig 3.4.3(b) (white squares). For comparison, the SNR obtained in each frame by applying the Maximum Cross-correlation MPE algorithm (described as method MS1 in Chapter 4) [3.28], is also shown (hatched squares). A MPE frame of 50 samples is used and 5 pulses are defined in each frame. The 12th order LPC filter $A(z)$ is defined over a larger interval of 200 samples. The speech waveform is shown in Figure 3.4.3(c).

The Simulated Annealing MPE algorithm works by considering transitions which change the positions of the excitation pulses (given an initial set of q pulse positions). At each optimisation step, a pulse is randomly selected and it is moved to an adjacent position either to the left or to the right (random choice). The probability model of Eq 3.4.2.1 is employed to decide the fate of each transition. When a transition is judged to be "successful", the whole process continues by selecting another pulse and moving it to an adjacent position either to the left or to the right. When a transition is not "successful", the selected pulse must be returned to its original position (from where it was displaced), before the process is allowed to continue by selecting another pulse.

Initially the pulses are placed at random positions within each speech frame and then the algorithm generates random one-sample displacements of the position of a randomly selected pulse. The error is calculated using the formula of Eq 3.3.9 and the new pulse position is rejected or accepted according to the probability model of Eq 3.4.2.1. The value of the parameter T is gradually decreased and the approximation error is monitored until no more transitions are considered to be "successful". At this point, the value of the approximation error is accepted as the minimum. As seen in Fig 3.4.3(b), the SNR improvement can be as much as 10 dBs in some frames.

2) RANDOM SEARCH

Random Search optimisation algorithms sample the parameter set Ψ in a random manner. A search route is usually constructed and random fluctuations are produced in order to explore the possibility of further advancing the route. Different methods employ different search strategies to locate the minimum [3.34,3.35]. They are computationally inefficient but are well suited for solving combinatorial optimisation problems and minimise the risk of accepting a local minimum as the global one.

The parameter set Ψ can be sampled more efficiently if the elements that are unlikely to be selected as the optimum solution can be identified in advance. This can be done experimentally using an existing MPE coder and a training process to determine the pulse position combinations which are unlikely to be encountered in practice. The corresponding position vectors can then be removed from Ψ. In the extreme case, a limited vocabulary of pulse position vectors can be constructed to replace the parameter set Ψ itself. In such a case, the search for the minimum approximation error can be done using an exhaustive enumeration technique. MPE coders based on this Codebook Search (CS) strategy will be described in Chapter 5.

3) BLOCK SEARCH (BS) OPTIMISATION

These methods start with an initial estimate of the pulse positions and search for the optimum set of pulse positions by perturbing the position vector components in a random or systematic way. When a vector is located that results a minimum approximation error, the search resumes by choosing the new vector as the initial point. These methods are less complex than the Simulated Annealing and Random Search methods, and give results comparable to the Multi-Stage optimisation algorithms [3.36,3.37]. The BS optimisation algorithms will be examined in the next chapter.

3.5 Conclusions

A brief description of the Multipulse Excitation optimisation process has been presented. It was shown that the excitation pulse amplitudes can be estimated, for a fixed set of pulse positions, by minimising the distance between two vectors, corresponding to the input speech waveform and the synthesized output of the MPE coder. The estimation is done assuming an

additive noise model and a known noise covariance.

The problem of optimising the excitation pulse positions has been converted to a function minimisation problem. The function itself represents a measure of the distortion introduced by the coder, and the function variables are the integer positions of the pulses.

Two broad classes of optimisation algorithms, the Successive Elimination and the Multivariate Optimisation techniques, have been described along with typical examples from each class. The most complex amongst these algorithms have a lot in common with well known Integer Programming methods. These algorithms can be employed by a MPE coder to obtain exceptionally good results, but they are usually unsuitable for use in real-time environments and speech transmission applications where the algorithm complexity must be weighed against implementation costs.

The Multi-Stage (MS), Codebook Search (CS) and Block-Search (BS) optimisation methods can be efficiently implemented using the existing technology and can be easily modified to suit the performance/complexity requirements of a particular application. These methods will be examined in more detail in the next chapters.

REFERENCES

[3.1] B.S.Atal, J.R.Remde, "A New Model of LPC Excitation for producing natural-sounding speech at Low Bit Rates", Proc ICASSP 1982, pp 614

[3.2] M.R.Schroeder, B.S.Atal, "Code-Excited Linear Prediction (CELP): High quality speech at very Low Bit Rates", Proc ICASSP 1985, pp 937

[3.3] N.Gouvianakis, C.Xydeas, "Advances in Analysis by Synthesis LPC Speech Coders", IERE Journal, Supplement on Mobile Radio, Vol 57, No 6, Nov/Dec 1987, pp S272-S286

[3.4] M.R.Schroeder et al, "Optimizing Digital Speech Coders by exploiting Masking Properties of the Human Ear", J. Acoustical Society of America, Vol 66, No 6, Dec 1979, pp 1647-1652

[3.5] R.V.Cox, "Recent Trends in Digital Speech Coding", Proc GLOBECOM 1983 IEEE, pp 784-788

[3.6] M.Taka et al, "DSP Implementations of Sophisticated Speech Coders", IEEE J. Selected Areas in Communications, Vol 6, No 2, Feb 1988, pp 274-282

[3.7] H.Alrutz, "Implementation of a Multi-Pulse coder on a Single Chip Floating-Point Signal Processor", Proc ICASSP 1986, pp 2367

[3.8] C.Horne et al, " VLSI Implementable algorithm for transparent coding of Wide Band speech below 32 kb/s", Proc Int. Conf. IASTED, Paris, Jun 1985

[3.9] R.E.Crochiere, J.L.Flanagan, "Speech Processing: An Evolving Technology", AT&T Technical Journal, Vol 65, Issue 5, Sep/Oct 1986, pp 2-11

[3.10] J.G.Gardiner, "Prospects for Sattelite Mobile Radio services for Europe in the 1990s", IERE Journal, Supplement on Mobile Radio, Vol 57, No 6, Nov/Dec 1987, pp S241-S245

[3.11] J.E.Natvig, "Evaluation of six Medium Bit Rate Coders for the Pan-European Digital Mobile Radio System", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 1988, pp 324-331

[3.12] S.Ono, K.Ozawa, "2.4 kbps Pitch Prediction Multi-Pulse speech coding" Proc ICASSP 1988, pp 175

[3.13] S.Singhal, B.S.Atal, "Improving Performance of Multi-Pulse LPC coders at Low Bit Rates", Proc ICASSP 1984, pp 1.3.1

[3.14] I.M.Trancoso et al, "A study on Short-Time Phase and Multipulse LPC" Proc ICASSP 1984, pp 10.3.1

[3.15] I.M.Trancoso et al, "Pole-Zero Multipulse Speech Representation using Harmonic Modelling in the Frequency domain", Proc ICASSP 1985, pp 260

[3.16] B.W.Dickinson, J.M.Turner, "Reflection Coefficient Estimation using Cholesky Decomposition", IEEE Trans. ASSP, Vol 27, No 2, Apr 1979, pp 146-148

[3.17] B.R.Musicus, "An Iterative Algorithm for Finding Stable Solutions to the Covariance or Modified Covariance Autoregressive Modeling Methods", Proc ICASSP 1982, pp 244

[3.18] A.H.Gray, D.Y.Wong, "The Burg algorithm for LPC Speech Analysis/ Synthesis", IEEE Trans ASSP, Vol 28, No 6, Dec 1980, pp 609-615

[3.19] G.A.Mack, V.K.Jain, "Bayesian Deconvolution of Speech containing Pulsed Excitation", Proc ICASSP 1985, pp 1137

[3.20] R.S.Garfinkel, G.L.Nemhauser, "Integer Programming", John Wiley & Sons, 1972

[3.21] H.R.Schwarz, "Numerical Analysis of Symmetric Matrices", Prentice-Hall, 1973

[3.22] P.Kroon et al, "Regular-Pulse Excitation: A novel approach to effective and efficient Multipulse Coding of Speech", IEEE Trans ASSP Oct 1986, pp 1054-1063

[3.23] A.Dembo, D.Malah, "A new approach to Multipulse LPC coder design", Proc ICASSP 1985, pp 949

[3.24] S.W.Lang, "Solving a class of Nonlinear Least Squares problems", Proc ICASSP 1985, pp 312

[3.25] L.R.Foulds, "Optimization Techniques: An Introduction", Springer-Verlag, New York, 1981

[3.26] T.F.Thorpe, "Performance Bounds for Digital Coding of Speech", PhD dissertation, Trinity College, University of Cambridge, March 1987

[3.27] M.Berouti et al, "Efficient Computation and Encoding of the Multi-pulse Excitation for LPC", Proc ICASSP 1984, pp 10.1.1

[3.28] T.Araseki et al, "Multi-Pulse Excited Speech coder based on Maximum Crosscorrelation Search algorithm", IEEE Proc. Global Telecommun. Conference, 1983, pp 794-798

[3.29] S.Singhal, "Reducing computation in Optimal Amplitude Multipulse coders", Proc ICASSP 1986, pp 2363

[3.30] G.M.Furnival, R.W.Wilson, "Regressions by Leaps and Bounds", Technometrics, Vol 16, No 4, Nov 1974, pp 499-511

[3.31] W.H.Press, B.P.Flannery, "Numerical Recipes: The Art of Scientific Computing", Cambridge University Press, 1986

[3.32] J.Vaisey, A.Gersho, "Simulated Annealing and Codebook Design", IEEE Proc ICASSP, New York 1988, pp 1176

[3.33] T.F.Thorpe, N.G.Kingsbury, "A monotonic descent algorithm to choose pulse locations in the excitation sequence of a multipulse speech coder", IEE Electronic Letters, Vol 21, No 21, Oct 1986, pp 972-973

[3.34] G.N.Vanderplaats, "Numerical Optimization Techniques for Engineering Design", McGraw-Hill, 1984

[3.35] G.Schrack, N.Borowski, "An Experimental Comparison of three Random Searches", Conf. on Numerical Methods for Non-linear Optimization, Academic Press, 1972, pp 137-148

[3.36] N.Gouvianakis, C.Xydeas, "A Multipulse Excited LPC coder implementation based on a Block Solution approach", Proc Int. Conf. Digital Processing of Signals in Communications, IERE, Loughborough 1985, pp 85-92

[3.37] N.Gouvianakis, C.Xydeas, "A Comparative Study of Multistage Sequential and Block Sequential Search Multipulse LPC algorithms", Proc Int. Conf. IASTED, Paris, Jun 1985, pp 122-125

CHAPTER 4

MULTI-STAGE AND BLOCK-SEARCH OPTIMISATION METHODS

4.1 Introduction

The powerful concept of Analysis by Synthesis optimisation makes the design of efficient Multipulse Excitation (MPE) coders, over a wide range of transmission bit rates, a straight forward task. MPE coders are especially successful at bit rates between 8 and 16 kbits/sec, and therefore compete with Residual Excited Linear Prediction coders (RELP) and Subband coders.

MPE coders can benefit from the substantial amount of research that has been carried out in order to improve the performance of predictive coders, especially on the issues of LPC parameter quantization [4.1,4.2,4.3]. RELP coders attempt to preserve the subjectively important properties of the LPC residual but, because they design the filter excitation in an "open loop" manner, they have to rely on inreasingly sophisticated and efficient excitation coding techniques [4.4,4.5].

The flexibility of the Multipulse Excitation model and the efficiency of the "closed loop" MPE optimisation process contribute to the naturalness of the coded speech [4.6]. It is therefore not unexpected to find traces of the same coder optimisation principles embedded in a number of low to medium bit-rate coding techniques that have been developed [4.7,4.8,4.9,4.10,4.11]. MPE coders are also quite robust in the presence of acoustic noise or transmission errors even at low bit rates [4.12], where traditional vocoding techniques become highly sensitive to wrong voicing decisions.

The performance of a typical MPE coder progressively deteriorates as the transmission bit-rate is brought below 8 kbits/sec. This happens because the number of pulses that are available to reconstruct each pitch period during voiced speech, becomes increasingly smaller. This "pulse starvation" effect is more noticeable for high-pitched voices.

A number of modifications have been suggested to improve the performance of MPE coders at lower bit-rates. The inclusion of a long-term predictor in the form of an all pole filter, can help in preserving the periodicity of the recovered speech, by reducing the number of excitation pulses that are necessary to reconstruct the speech waveform in an interval equivalent to many pitch periods [4.13,4.14]. Vector Quantization of the excitation and

LPC parameters can be used to bring the bit-rate down to 4.8 kbits/sec [4.15,4.16,4.17], and a pitch synchronous Multipulse Excitation optimisation and interpolation can reduce the bit rate even further to 2.4 kbits/sec [4.18,4.19]. In addition, new and efficient techniques for the quantization of the LPC parameters are becoming increasingly important when operating at bit rates less than 6 kbits/sec, while preserving the naturalness of the coded speech [4.12,4.20,4.21,4.22].

The coded speech quality obtained from MPE coders at 16 kbits/sec is very close to toll-quality (equivalent to more than 7 bits PCM coded speech), at 9.6 and 8 kbits/sec good communications-quality speech is achieved and in the 2.4-4.8 kbits/sec range, coded speech sounds more "natural" and "full" when compared to the output of typical LPC or channel vocoders.

Efforts to improve the performance of MPE coders concentrate on developing better Multipulse Excitation optimisation algorithms and on improving the basic MPE model. Better estimation algorithms for the LPC filter [4.23,4.24, 4.25,4.26,4.27] and use of adaptive post-filtering techniques [4.28] can produce a small improvement of the coded speech quality. More effective control over the noise frequency distribution can be achieved using a split-band design [4.29,4.30,4.31], by independently adjusting the noise level in each of the frequency bands.

The complexity of the MPE optimisation algorithms is not prohibitive, and increasingly complex implementations have been demonstrated [4.14,4.32,4.33] following the considerable advances in the technology of the general purpose Digital Signal Processing devices.

A detailed examination of the pulse amplitude estimation process and a more complete description of the LPC sythesis filter model (which may include a Short Term Predictor and a Long Term predictor), will be presented in this chapter. The effectiveness of the MPE noise shaping process in changing the spectral distribution of the added distortion and improving the perceptual quality of the coded speech, will also be examined.

A number of MPE optimisation algorithms, drawn from the two categories of Multi-Stage and Block-Search optimisation techniques, will be presented. These algorithms can normally be incorporated in any MPE coding system, regardless of modifications to the basic MPE-LPC model. The performance of a MPE coder can improve substantially when changing from a simple to a complex

MPE optimisation algorithm, and the effect is more pronounced at higher transmission bit-rates. A choice can usually be made among a few candidate optimisation algorithms that would suit the particular requirements of a speech coding application and it is therefore important to know the merits of each available algorithm.

Finally results on the algorithms' SNR performance and complexity, and a method to design PDF optimised quantizers for the amplitudes of the excitation pulses, will be presented.

4.2 Noise shaping in a MPE coder

An analytical method to estimate the pulse amplitudes based on an additive noise model was described in Section 3.3. The noise covariance was assumed to be known in advance, but in practice there is very little knowledge of the underlying noise process. It is desirable though to enforce certain spectral properties on the distortion introduced by the coder, in order to reduce the perceptible distortion. The desirable noise spectral shape should ideally follow the spectral variation of the speech waveform and should allow for most of the noise energy to be concentrated within the speech formant frequency regions, where higher noise levels can be tolerated.

Figure 4.2.1 shows the power spectra of the speech and the estimated noise (added distortion) when a unit noise covariance matrix C_a is chosen for the MPE amplitude estimation process. These spectra correspond to a 64 ms voiced speech segment, and an iterative MPE optimisation algorithm is employed to bring the Signal to Distortion Ratio (SDR) to the specified values of 12 dBs (Fig 4.2.1(a)) and 24 dBs (Fig 4.2.1(b)). The resulting noise spectrum is certainly not flat and indicates that the MPE model favours the high power speech spectral regions with a considerably higher SDR than that of the low power regions.

This effect is more obvious at high SDRs and is caused by the inherent bias of the Linear Prediction filter estimation method, in modeling the high power speech spectral components more accurately than the low power components. As a consequence, when the order of the LPC filter is substantially increased, a better spectral model of speech and a flatter noise spectrum result.

The unfavourable distribution of the noise power can be corrected by

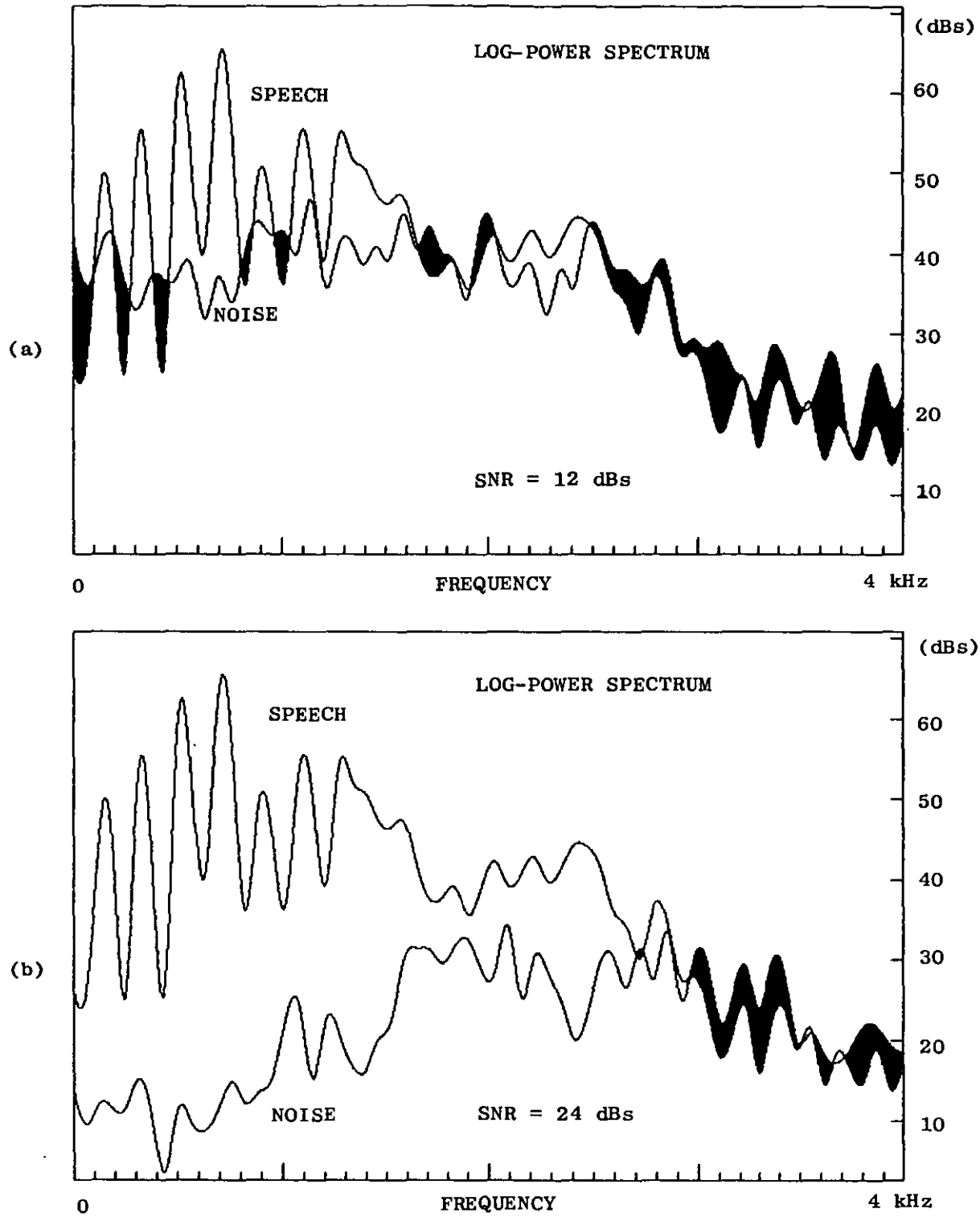


FIGURE 4.2.1 Power Spectra of Speech and Distortion (noise) introduced by the MPE coding process, when no noise shaping is employed. The duration of the time window is 64 ms, and the SNR is (a) 12 dBs and (b) 24 dBs.

setting the noise covariance matrix C_a to correspond to a desirable spectral shape, thus counteracting the uneven frequency distribution of the noise and reversing the tendency of the MPE optimisation process to model more accurately the high power speech spectral regions. This only allows limited control over the spectral distribution of the distortion introduced by the coder but can still produce a noticeable improvement in the quality of coded speech at high SDRs.

A model of the desirable noise spectral shape, that has been used extensively is :

$$W^{-1}(z) = \frac{A(z)}{A_w(z)} \qquad (Eq\ 4.2.1)$$

The LPC filter $A(z)$ models the speech spectral envelope and the filter $A_w(z)$ can be constant [4.34] or adaptive [4.35]. An adaptive filter will be considered here, of the form :

$$A_w(z) = A(z/\gamma) = \frac{1}{1 - \sum_{m=1}^{l} a_m \gamma^m z^{-m}} = \frac{1}{1 - \sum_{m=1}^{l} g_m z^{-m}} \quad , 0 \le \gamma \le 1 \qquad (Eq\ 4.2.2)$$

The desirable noise spectral shape is therefore related to the model :

$$W^{-1}(z) = \frac{1 - \sum_{m=1}^{l} g_m z^{-m}}{1 - \sum_{m=1}^{l} a_m z^{-m}} \qquad (Eq\ 4.2.3)$$

The poles of the ARMA filter $W^{-1}(z)$ coincide with the poles of the LPC filter $A(z)$ and its zeros are along the same radial axes as its poles but shrunk by a factor γ. By varying the value of γ between 0 and 1, the shape of the frequency response of $W^{-1}(z)$ changes from being identical to the frequency response of the LPC filter $A(z)$ to being completely flat. The "desired" noise covariance matrix related to the frequency response of $W^{-1}(z)$ can be formulated as :

$$C_a = L_w L_w^T \qquad (Eq\ 4.2.4)$$

where L_w is the lower triangular convolution matrix which is formed using the impulse response of the filter $W^{-1}(z)$.

The pulse amplitude estimation process is affected by this choice of noise covariance, because the linear transformation defined in Eq 3.3.3 is changed to $\sigma \mathbf{L}_w^{-1}$. If $\mathbf{W}$ is the lower triangular convolution matrix that corresponds to the filter $W(z)$ then :

$$\mathbf{W} = \mathbf{L}_w^{-1} \qquad (Eq\ 4.2.5)$$

Since the scaling constant σ does not affect the solution for the pulse amplitudes (Eq 3.3.7), the linear transformation is set equal to $\mathbf{W}$. This transformation is equivalent to a filtering operation using the filter $W(z)$.

The power attenuation characteristics of $W(z)$ are plotted in Figure 4.2.2 for two different values of the constant γ. The frequency response of the LPC filter $A(z)$ is also shown for comparison. When $\gamma=1$ then $W(z)=1$ and no noise shaping is applied. When $\gamma=0$ then $W(z)$ becomes equal to the LPC inverse filter :

$$W(z) = 1 - \sum_{m=1}^{l} a_m z^{-m} \qquad (Eq\ 4.2.6)$$

and maximum noise shaping is applied.

A different interpretation can now be given to Eq 3.3.3. The signal s_w corresponds to a filtered version of the speech waveform s, which suffered an increased attenuation of its high power frequency components by the noise shaping filter $W(z)$. Also, $\mathbf{A}_w[q]\mathbf{b}$ is the output of the combined filters $A(z)$ and $W(z)$ in response to an input of excitation pulses of amplitudes $b_1, b_2, \ldots, b_q$ and positions $p_1, p_2, \ldots, p_q$ (see Figure 4.4.1(a)).

If the noise shaping filter is chosen as in Eq 4.2.3 then the combined filter is reduced to $A_w(z)$, called Modified Synthesis Filter (MSF), and the MPE optimisation process is simplified. Further simplifications can be made by exploiting the properties of the MSF. The impulse response of the MSF is:

$$A_w(z) = h(0) + \gamma h(1) + \gamma^2 h(2) + \gamma^3 h(3) + \ldots \qquad (Eq\ 4.2.7)$$

By lowering the value of γ, the effective duration of the impulse response is increasingly restricted. If the minimum distance between the excitation pulses is kept greater than the impulse response duration, then simple non-iterative algorithms for the definition of the MPE sequence can be derived [4.36,4.37]. Furthermore, by setting the value of γ equal to zero, a very

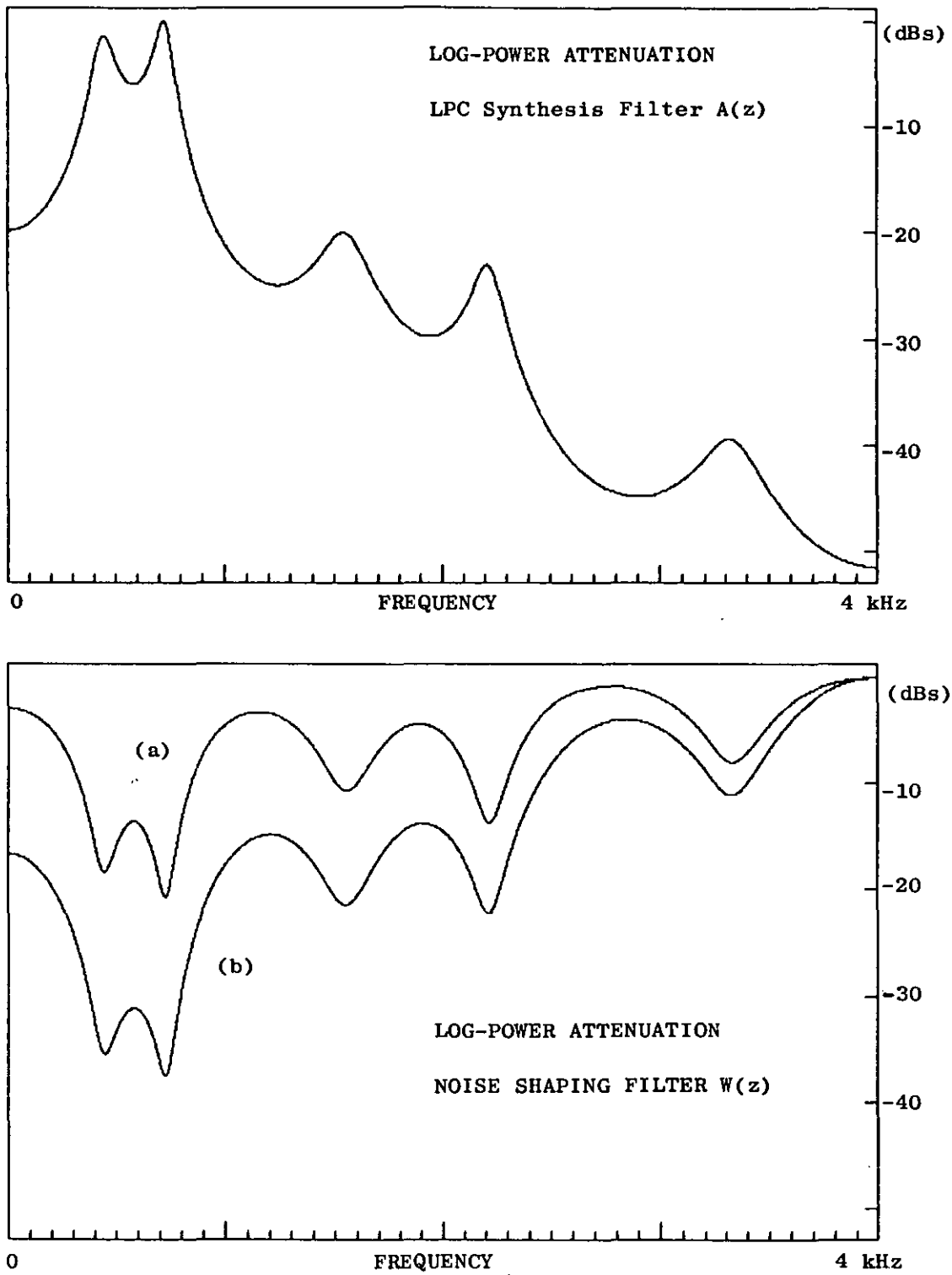


FIGURE 4.2.2 Log-Power Attenuation characteristics of the LPC Synthesis Filter $A(z)$, and the noise shaping filter $W(z)$ which is derived from $A(z)$. The constant γ takes the values (a) 0.8 and (b) 0.4.

simple non-iterative MPE estimation process can be formulated [4.25,4.38, 4.39,4.40], but the quality of the coded speech will be compromised.

4.3 Short Term Linear Predictor Model

The all pole LPC filter $A(z)$ models the short-term spectral envelope of speech, placing particular emphasis on the high power frequency components. The filter is usually constrained to be minimum phase and is estimated using Short-Term Linear-Prediction Error-Minimisation methods. The filter coefficients should be updated frequently enough to allow adequate sampling of the dynamically changing frequency distribution of the speech waveform, especially during sound transitions. The interval over which the LPC filter is defined should include the MPE definition interval, but could be allowed to precede and overlap the MPE interval if the overall coder delay is to be kept to a minimum [4.41].

Speech can be modeled as an Autoregressive (AR) process :

$$s(i) = \sum_{m=1}^{l} a_m s(i-m) + e_p(i) \quad , \quad i=0,1,2,... \qquad (Eq\ 4.3.1)$$

where e_p is an uncorrelated innovation source called forward prediction error. A backward prediction error can be similarly defined and the filter coefficients $\{a_m\}$ can be estimated by minimising a function of the two errors (as when a lattice LPC predictor is defined).

Better models have been proposed to remove the bias of the estimated coefficients during voiced speech when large prediction errors occur periodically, thus violating the hypothesis of an uncorrelated innovation source [4.23,4.42]. The assumption is made that a Multipulse Excitation sequence has been predetermined and can be included in the AR model:

$$s(i) = \sum_{m=1}^{l} a_m s(i-m) + \sum_{k=1}^{q} b_k \delta(i-p_k) + e_m(i) \quad , \quad i=0,1,2,... \qquad (Eq\ 4.3.2)$$

where $\delta(i)$ is the Kronecker function. The speech quality improvement is very small [4.27] because the error which is minimised in Eq 4.3.2 is substantially different from the error minimised during the MPE optimisation. This can be verified by forming the one-sided z-transform equivalent of Eq 4.3.2:

$$S(z) = \sum_{m=1}^{l} a_m \left[S(z)z^{-m} + \sum_{i=o}^{m-1} s(i-m)z^{-i} \right] + \sum_{k=1}^{q} b_k z^{-P_k} + E_m(z) \qquad (Eq\ 4.3.3)$$

or by rearranging terms :

$$S(z)\left[1 - \sum_{m=1}^{l} a_m z^{-m}\right] = \sum_{i=o}^{l-1} \left[\sum_{m=1}^{l-i} a_{i+m} s(-m) \right] z^{-i} + \sum_{k=1}^{q} b_k z^{-P_k} + E_m(z) \qquad (Eq\ 4.3.4)$$

The error minimised by the MPE optimisation algorithms is a function of the difference between the original and synthesized speech waveforms. From Eqs 3.2.5 and 3.3.3, the modeling error whose energy is minimised becomes :

$$E_w(z) = W(z)\left[S(z) - S_y(z)\right] \qquad (Eq\ 4.3.5)$$

To calculate the synthesized waveform $S_y(z)$, the difference equation :

$$s_y(i) = \sum_{m=1}^{l} a_m s_y(i-m) + \sum_{k=1}^{q} b_k \delta(i-p_k) \qquad (Eq\ 4.3.6)$$

must be solved.. Working in the same way as for Eq 4.3.4, we obtain :

$$S_y(z) = M_y(z) + A(z) \sum_{k=1}^{q} b_k z^{-P_k} \qquad (Eq\ 4.3.7)$$

where

$$M_y(z) = A(z) \sum_{i=o}^{l-1} \left[\sum_{m=1}^{l-i} a_{i+m} s_y(-m) \right] z^{-i} \qquad (Eq\ 4.3.8)$$

is the transient response of the LPC synthesis filter and $s_y(-m)$, $m=1,2,..,l$ are the last l synthesized speech samples of the previous frame. Assuming for the moment that :

$$s_y(-m) = s(-m) \ , \ m=1,2,\ldots,l \qquad (Eq\ 4.3.9)$$

and substituting Eqs 4.3.7 and 4.3.4 into Eq 4.3.5, the relationship between the two modeling errors can be established :

$$E_w(z) = W(z)A(z)E_m(z) = A_w(z)E_m(z) \qquad (Eq\ 4.3.10)$$

The power spectra of the two modeling errors are therefore quite different. The coefficients of the LPC filter could be estimated by minimising the energy of $E_w(z)$ instead of $E_m(z)$, but that would involve an iterative

non-linear optimisation process which could be complex and might not guarantee the stability of the estimated filter. A better way to remove the effect of the error surge during voiced speech, is to include a second AR filter that models the fine structure of the speech frequency distribution (long-term predictor).

4.4 Pulse Amplitude Optimisation (Flow Diagrams - Simplifications)

The transient response of the LPC filter, defined by Eq 4.3.8, was not taken into acount when the amplitude estimation problem was examined in Section 3.3. To inlude this extra term, the additive noise model of Eq 3.2.5 must be modified to :

$$\mathbf{s} = \mathbf{s}_y + \mathbf{e}_a = \mathbf{m}_y + \mathbf{A}[q]\,\mathbf{b} + \mathbf{e}_a \qquad (Eq\ 4.4.1)$$

where $\mathbf{m}_y$ is the transient response and $\mathbf{A}[q]\,\mathbf{b}$ is the forced response of the LPC filter. The normal equations for the pulse amplitudes also change to :

$$\mathbf{D}[q]\,\mathbf{b} = \mathbf{A}_w[q]^T\,\mathbf{W}(\mathbf{s}-\mathbf{m}_y) \qquad (Eq\ 4.4.2)$$

where :

$$\mathbf{D}[q] = \mathbf{A}[q]^T\,\mathbf{A}[q] \qquad (Eq\ 4.4.3)$$

The minimum error energy for a given set of pulse positions becomes :

$$\left[\ \mathbf{e}_w^T\,\mathbf{e}_w\ \right]_{min} = (\mathbf{s}-\mathbf{m}_y)^T\mathbf{W}^T\mathbf{W}\,(\mathbf{s}-\mathbf{m}_y) - \mathbf{b}^T\mathbf{A}_w[q]^T\,\mathbf{W}(\mathbf{s}-\mathbf{m}_y) \qquad (Eq\ 4.4.4)$$

This alteration is reflected in the flow diagram of the MPE optimisation process. As shown in Fig 4.4.1(a), the subtraction of the LPC transient from the input speech samples is followed by a filtering through the noise shaping filter $W(z)$. The result is compared to the response of the Modified Synthesis filter (MSF), when excited by the MPE sequence. The difference between the two waveforms forms the error $E_w(z)$ whose energy must be minimised. Note that the initial state of the MSF is set to zero.

The MPE optimisation process can also be interpreted in another way by considering the relationship between the speech waveform and the residual $E_p(z)$. The one sided z-transform equivalent of Eq 4.3.1 is :

$$S(z) = M(z) + A(z)E_p(z) \qquad (Eq\ 4.4.5)$$

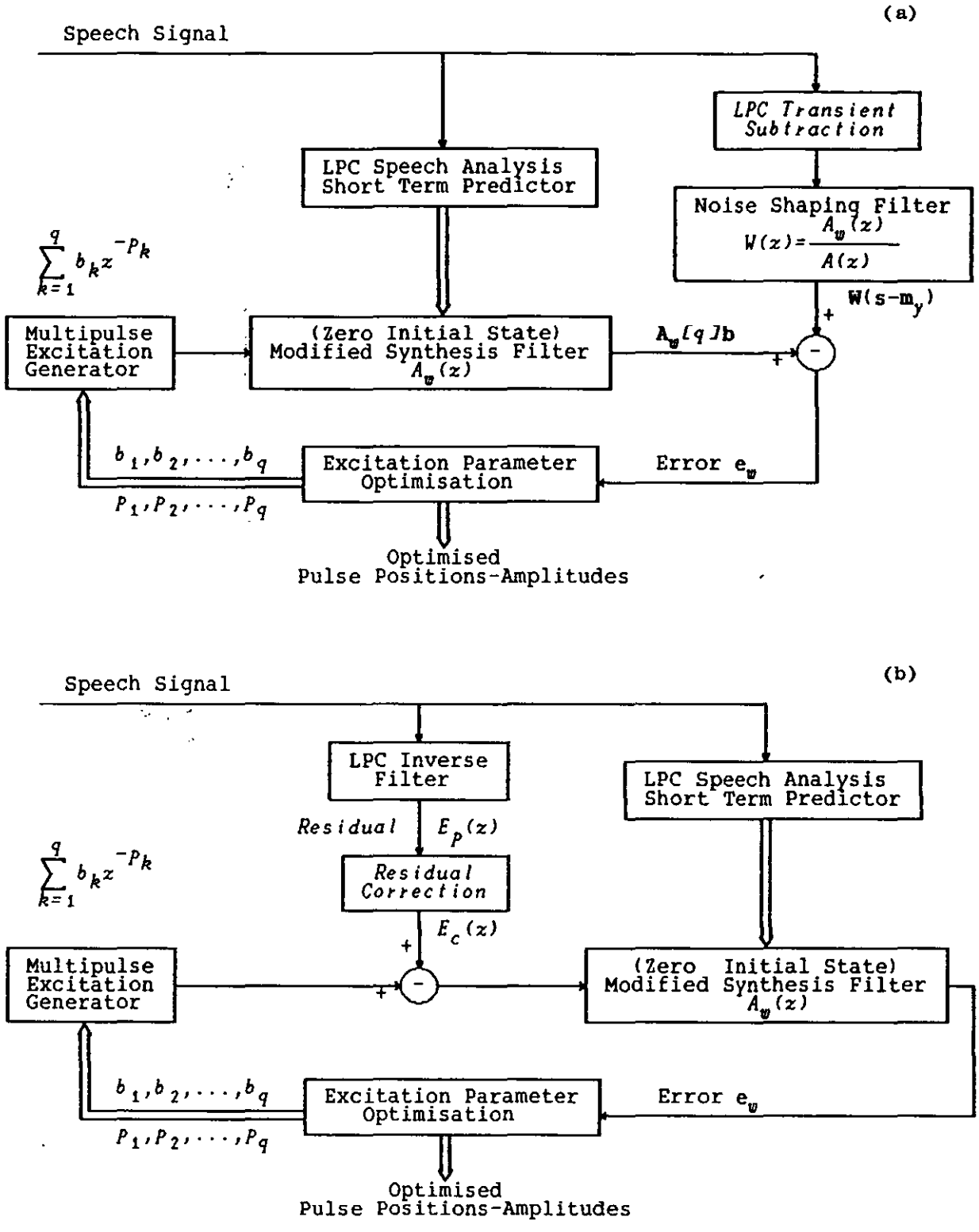


FIGURE 4.4.1 (a) and (b) Two equivalent diagrams for the MPE coding process

where :

$$M(z) = A(z) \sum_{i=0}^{l-1} \Big[\sum_{m=1}^{l-i} a_{i+m} s(-m) \Big] z^{-i} \qquad (Eq\ 4.4.6)$$

Substitution of Eqs 4.4.5 and 4.3.7 into Eq 4.3.5 leads to :

$$E_w(z) = A_w(z) E_c(z) - A_w(z) \sum_{k=1}^{q} b_k z^{-p_k} \qquad (Eq\ 4.4.7)$$

where :

$$E_c(z) = E_p(z) + \sum_{i=0}^{l-1} \Big[\sum_{m=1}^{l-i} a_{i+m} \big(s(-m) - s_y(-m) \big) \Big] z^{-i} \qquad (Eq\ 4.4.8)$$

The error $E_w(z)$ in Eq 4.4.7 is now formed in a different way. The two waveforms compared in the optimisation process, correspond to the response of the zero-initial-state MSF to the excitation inputs, the MPE and the new excitation waveform $E_c(z)$. This new waveform differs from the prediction error $E_p(z)$ in only the first l samples in each frame. This adjustment is made to account for the fact that the last l samples of the previous speech frame are not the same as the last l synthesized samples of the same frame. This "corrected" residual will now be used in place of the original LPC residual.

In Figure 4.4.1(b), the two excitation waveforms are directly compared before passing through the zero-initial-state MSF to form the error $E_w(z)$. The idea of a "corrected" residual can be transfered to the amplitude estimation equations. Using the matrix equivalent of Eq 4.4.5 ($\mathbf{s} = \mathbf{m} + \mathbf{A}\,\mathbf{e}_p$) and by substituting into Eq 4.4.2, the normal equations for the pulse amplitudes become :

$$\mathbf{D}[q]\,\mathbf{b} = \mathbf{A}_w[q]^T \cdot \mathbf{A}_w\,\mathbf{e}_c \qquad (Eq\ 4.4.9)$$

where :

$$\mathbf{e}_c = \mathbf{e}_p + \mathbf{A}^{-1}(\mathbf{m} - \mathbf{m}_y) \qquad (Eq\ 4.4.10)$$

The elements of the qxq matrix $\mathbf{D}[q]$ are :

$$D[q](k,m) = \Phi(p_k, p_m) = \sum_{i=0}^{n-max(p_k,p_m)-1} h_w(i) h_w(i + |p_k - p_m|) \quad , \quad 1 \le k,m \le q \qquad (Eq\ 4.4.11)$$

When the minimum distance between the excitation pulses is kept greater

than the effective duration of the MSF impulse response, the matrix $D[q]$ becomes diagonal and the solution of the normal equations 4.4.2 or 4.4.10 is considerably simplified [4.36]. The reduced complexity of the amplitude estimation process allows the search algorithm for the optimum pulse positions, to consider a larger set of possible position combinations.

An extreme simplification results when the constant γ in the noise shaping filter $W(z)$ is set equal to zero. In this case the impulse response of $A_w(z)$ is a unit impulse and the amplitudes, estimated using Eq 4.4.9, are :

$$b_k = e_c(p_k) \qquad , k=1,2,\ldots,q \qquad \textit{(Eq 4.4.12)}$$

This simple solution for the pulse amplitudes is accompanied by a simple solution for the pulse positions. The minimum error energy for a given set of pulse positions is :

$$\left[\mathbf{e}_w^T \mathbf{e}_w \right]_{min} = \mathbf{e}_c^T \mathbf{e}_c - \sum_{k=1}^{q} e_c^2(p_k) \qquad \textit{(Eq 4.4.13)}$$

The optimum set of pulse positions can therefore be determined by locating the maxima of the "corrected" residual $E_c(z)$ [4.40]. As expected, results obtained from this algorithm are not as good as the results obtained from a full implementation of a MPE optimisation algorithm.

Depending on the search strategy adopted by a MPE optimisation algorithm, the system of normal equations 4.4.2 may need to be solved for each of the pulse position combinations considered. To avoid the considerable amount of computation involved in calculating the matrix $D[q]$ for each set of pulse positions, a larger $n \times n$ matrix can be calculated only once :

$$\mathbf{D} = \mathbf{A}_w^T \mathbf{A}_w$$

or : *(Eq 4.4.14)*

$$D(i+1,j+1) = \Phi(i,j) \qquad , 0 \le i,j \le n-1$$

and the elements of $D[q]$ can be chosen from the matrix $\mathbf{D}$ as :

$$D[q](k,m) = D(p_k+1, p_m+1) \qquad , 1 \le k,m \le q \qquad \textit{(Eq 4.4.15)}$$

Computational savings result because the elements along each diagonal of $\mathbf{D}$ can be recursively computed from its first row or column, using the formula:

$$D(i+1,j+1) = D(i,j) - h_w(n-i)h_w(n-j) \qquad \textit{(Eq 4.4.16)}$$

Provided there is sufficient storage available, the full matrix **D** can be calculated from its first row, by performing $n(n-1)/2$ multiplications and additions.

To reduce the computational load and the storage requirements, the matrix **D** may be assumed to have a Toeplitz structure, in which case its elements are only a function of the distance between the pulse positions :

$$D(i+1,j+1) = \Phi_a(i,j) = \sum_{k=0}^{n-|i-j|-1} h_w(k)h_w(k+|i-j|) \quad , \; 0 \le i,j \le n-1 \qquad (Eq\ 4.4.17)$$

This is a similar approximation to the one done in LPC analysis when the prediction error summation limits are extended to infinity to form the autocorrelation LPC estimation method. The approximation of Eq 4.4.17 will therefore be referred to as the autocorrelation approximation and will be used extensively to simplify a number of MPE optimisation algorithms.

4.5 Long Term Linear Predictor Model

A second all-pole filter can be combined with the LPC filter defined in Section 4.3, to form a composite AR model. This second filter models the "fine" spectral structure of the speech signal and is defined as :

$$C(z) = \frac{1}{1 - \sum_{j=0}^{g} c_j z^{-d-j}} \quad , \; d>0 \qquad (Eq\ 4.5.1)$$

where d is an integer delay coefficient. The summation term of the denominator in Eq 4.5.1, forms a d-steps ahead prediction, hence the filter $C(z)$ will be referred to as the Long-Term Predictor (LTP).

The use of a LTP has been adopted by a number of predictive coding schemes [4.2,4.43], and results a greater efficiency in coding the excitation signal. In the case of the MPE coder, the use of a LTP induces periodicity in the synthesized speech waveform with less effort (i.e. smaller number of pulses), and can therefore achieve an overall reduction in the number of excitation pulses necessary to fulfil the quality requirements of a particular application [4.44]. The effect of the LTP in improving the quality of coded speech is epecially noticeable for high pitched voices.

A change in the value of the delay coefficient d has a highly nonlinear

effect on the state of the LTP. That is why the LTP is usually placed before the LPC filter when forming a combined synthesis filter. For the same reason extra protection from transmission errors must be allowed for the LTP coefficients, to safeguard against the accumulation and propagation of errors that may be caused by the long duration of the LTP's impulse response [4.13,4.45].

The value of the LTP delay coefficient is not an estimate of the pitch period, because the LTP is optimised to minimise a modeling error and is not designed to estimate the speech fundamental frequency. The role of the LTP in a MPE coder is also different from its role in other predictive coders. When the LTP coefficients are updated very frequently, the LTP cannot operate for a sufficiently long time with the same set of coefficients, and its behaviour is more tightly controlled by the MPE optimisation algorithm.

Avoiding for the momemt, the problem of estimating the LTP coefficients, the effect of introducing the LTP in the MPE optimisation process can be studied as follows :

The response of the LTP filter to the Multipulse Excitation can be recursively calculated using the difference equation :

$$x(i) = \sum_{j=o}^{g} c_j x(i-d-j) + \sum_{k=1}^{q} b_k \delta(i-P_k) \quad , \quad i=o,1,2,... \qquad (Eq\ 4.5.2)$$

The z-transform equivalent of Eq 4.5.2 is :

$$X(z) = \sum_{j=o}^{g} c_j \left[X(z) z^{-d-j} + \sum_{i=o}^{d+j-1} x(i-d-j) z^{-i} \right] + \sum_{k=1}^{q} b_k z^{-P_k} \qquad (Eq\ 4.5.3)$$

or by rearranging terms :

$$X(z) = M_l(z) + C(z) \sum_{k=1}^{q} b_k z^{-P_k} \qquad (Eq\ 4.5.4)$$

where :

$$M_l(z) = C(z) \sum_{j=o}^{g} \left[c_j \sum_{i=o}^{d+j-1} x(i-d-j) z^{-i} \right] \qquad (Eq\ 4.5.5)$$

is the transient response of the LTP filter. If the output of the LTP filter is used as the input to the LPC filter in Eq 4.3.7, instead of the MPE input, then the response of the LPC filter becomes :

$$S_y(z) = M_y(z) + A(z)M_l(z) + A(z)C(z) \sum_{k=1}^{q} b_k z^{-P_k} \qquad (Eq\ 4.5.6)$$

Substitution of Eq 4.5.6 into Eq 4.3.5 results :

$$E_w(z) = W(z)\left[S(z)-M_y(z)-A(z)M_l(z)\right] - A_w(z)C(z) \sum_{k=1}^{q} b_k z^{-P_k} \qquad (Eq\ 4.5.7)$$

The modeling error $E_w(z)$ is formed in the same way as before, when the LTP filter was not present. The difference is that a new "combined" transient response is subtracted from the speech signal and the combined synthesis filter $A_w(z)C(z)$ replaces the MSF $A_w(z)$ in the amplitude estimation equations (Fig 4.5.1(a)). Note that the search algorithms that optimise the pulse positions remain the same.

The MPE optimisation process can be represented in three different ways, corresponding to the diagrams in Figures 4.5.1(a),(b) and (c). To derive the second approach (the first was defined by Eq 4.5.7), Eqs 4.4.5 and 4.4.8 can be used to transform Eq 4.5.7 into:

$$E_w(z) = A_w(z)\left[E_c(z)-M_l(z)\right] - A_w(z)C(z) \sum_{k=1}^{q} b_k z^{-P_k} \qquad (Eq\ 4.5.8)$$

Eq 4.5.8 shows that the effect of the "combined" transient response can be taken into account by correcting the LPC residual for a second time, to remove the effect of the LTP transient. The LPC and LTP filters may then be assumed to start from a zero state (Fig 4.5.1(b)).

An AR model can be established in the same way that the Short-Term Predictor (STP) model was defined by Eq 4.3.1. The AR model for the Long Term Predictor is based on the residual $E_p(z)$ and assumes a different innovation source $E_s(z)$, called second residual :

$$e_p(i) = \sum_{j=o}^{g} c_j e_p(i-d-j) + e_s(i) \quad , \quad i=o,1,2,\ldots \qquad (Eq\ 4.5.9)$$

By transforming Eq 4.5.9 into the z-domain and combining it with Eq 4.4.5, a new model for the speech signal can be derived :

$$S(z) = M(z) + A(z)M_s(z) + A(z)C(z)E_s(z) \qquad (Eq\ 4.5.10)$$

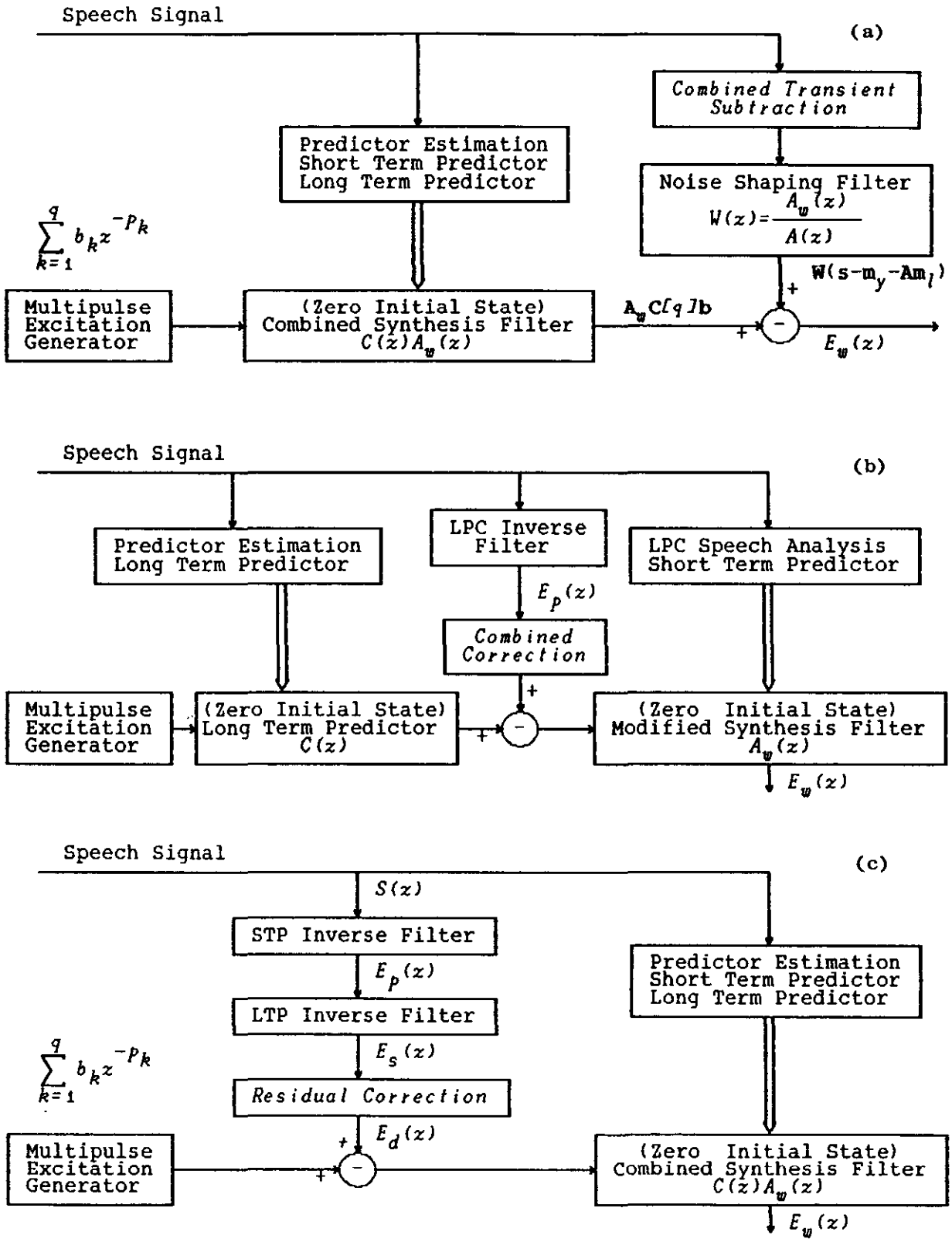


FIGURE 4.5.1 (a), (b) and (c) Three equivalent diagrams for the MPE coding process, when a Long Term Predictor model is employed. The matrix $C[q]$ is a nxq convolution matrix derived from the impulse response of the Long Term Predictor.

where :

$$M_s(z) = C(z) \sum_{j=o}^{g} \left[c_j \sum_{i=o}^{d+j-1} e_p(i-d-j) z^{-i} \right] \qquad (Eq\ 4.5.11)$$

Eqs 4.5.6 and 4.5.10 can be substituted into Eq 4.3.5 to form the error :

$$E_w(z) = A_w(z) C(z) \left[E_d(z) - \sum_{k=1}^{q} b_k z^{-P_k} \right] \qquad (Eq\ 4.5.12)$$

where :

$$E_d(z) = E_s(z) + \left[1 - \sum_{j=o}^{g} c_j z^{-d-j} \right] \left[\sum_{i=o}^{l-1} \left[\sum_{m=1}^{l-i} a_{i+m} \left(s(-m) - s_y(-m) \right) \right] z^{-i} \right] +$$

$$+ \sum_{j=o}^{g} \left[c_j \sum_{i=o}^{d+j-1} \left(e_s(i-d-j) - x(i-d-j) \right) z^{-i} \right] \qquad (Eq\ 4.5.13)$$

The "corrected" second residual, defined by Eq 4.5.13, is subtracted from the MPE sequence (Eq 4.5.12) and the result passes through the zero-initial-state combined synthesis filter, to form the error signal (Fig 4.5.1(c)).

Two different models have been proposed [4.13,4.24] for the estimation of the LTP filter coefficients. The first minimises a prediction error and the second minimises the same modeling error that is minimised by the MPE optimisation algorithms :

1) OBTAINING THE LTP PARAMETERS BY MINIMISING THE PREDICTION ERROR

The LPC predictor model, defined by Eq 4.3.1, removes the short-term correlation from the speech signal. A second predictor (Long-Term-Predictor) can be used to minimise any long-term correlation that is left over from the first modeling stage (Short-Term-Predictor). The coefficients of the LTP filter are estimated using Linear Prediction methods, similar to the covariance and autocorrelation methods used in LPC analysis. The covariance method can be derived by rewriting Eq 4.5.9 as :

$$\mathbf{e}_p = \mathbf{E}_p[d]\ \mathbf{c} + \mathbf{e}_s \qquad (Eq\ 4.5.14)$$

where $\mathbf{e}_p$ and $\mathbf{e}_s$ are the n_c-dimensional vectors that contain the samples of the first and second residual in the current frame of n_c samples, $\mathbf{c}$ contains the LTP coefficients and :

$$\mathbf{E}_p[d] = \begin{bmatrix} e_p(-d) & e_p(-d-1) & \dots & e_p(-d-g) \\ e_p(-d+1) & e_p(-d) & & e_p(-d-g+1) \\ \vdots & \vdots & & \\ e_p(-d+n_c-1) & e_p(-d+n_c-2) & \dots & e_p(-d-g+n_c-1) \end{bmatrix} \qquad (Eq\ 4.5.15)$$

The Least Squares estimate of the LTP coefficients is :

$$\mathbf{c} = \left(\mathbf{E}_p[d]^T\ \mathbf{E}_p[d]\right)^{-1}\ \mathbf{E}_p[d]^T\ \mathbf{e}_p \qquad (Eq\ 4.5.16)$$

and the minimum prediction error energy becomes :

$$\left[\ \mathbf{e}_s^T\ \mathbf{e}_s\ \right]_{min} = \mathbf{e}_p^T\ \mathbf{e}_p - \mathbf{e}_p^T\ \mathbf{E}_p[d]\left(\mathbf{E}_p[d]^T\ \mathbf{E}_p[d]\right)^{-1}\ \mathbf{E}_p[d]^T\ \mathbf{e}_p \qquad (Eq\ 4.5.17)$$

To determine the optimum value of the delay coefficient d, the expression of Eq 4.5.17 must be evaluated for every permissible value of d (usually a range of integer values between 20 and 160) in order to find the global error minimum.

The LTP filter defined by Eq 4.5.16 can become unstable and that might cause a deterioration in the quality of the coded speech due to the presence of annoying clicks and pops in the recovered speech. Simple testing procedures are available to detect the instability of the LTP filter and correct it without reducing the prediction gain of the LTP [4.46].

The coefficients of the STP and LTP can be estimated simultaneously in a single optimisation stage. An analytic process to achieve the optimisation is only possible when $d \geq n_c$. The second non-zero sample of the LTP impulse response is then outside the current frame of n_c samples and it can be safely assumed that $C(z)=1$. Substituting Eqs 4.4.6 and 4.5.11 into Eq 4.5.10 the new combined model of speech is formed :

$$S(z) = \sum_{m=1}^{l}\left[a_m \sum_{i=o}^{n_c} s(i-m)z^{-i}\right] + \sum_{j=o}^{g}\left[c_j \sum_{i=o}^{n_c} e_p(i-d-j)z^{-i}\right] + E_s(z) \qquad (Eq\ 4.5.18)$$

Eq 4.5.18 does not include any prediction error samples $e_p(i)$ from the current analysis frame $(o, n-1)$, and can be rewritten in a matrix form as :

$$\mathbf{s} = \mathbf{S}\ \mathbf{a} + \mathbf{E}_p[d]\ \mathbf{c} + \mathbf{e}_s \qquad (Eq\ 4.5.19)$$

where $\mathbf{e}_s$ is the modeling error (second residual), $\mathbf{a}$ is the l-dimensional

vector that contains the coefficients of the STP, and the nxl matrix **S** is defined as :

$$\mathbf{S} = \begin{bmatrix} s(-1) & s(-2) & \dots & s(-l) \\ s(o) & s(-1) & & s(-l+1) \\ \vdots & & & \vdots \\ s(n-2) & s(n-3) & \dots & s(-l+n-1) \end{bmatrix} \qquad (Eq\ 4.5.20)$$

The coefficients of the STP and LTP can be jointly optimised by minimising the energy of the modeling error in Eq 4.5.19. Both sets of coefficients are now dependent on the value of the parameter d, and their optimum values can be found (for a given value of d) using Least Squares methods. These optimum coefficients are :

$$\begin{bmatrix} \mathbf{a} \\ \hline \mathbf{c} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{S}^T\mathbf{S} & \mathbf{S}^T\,\mathbf{E}_p[d] \\ \hline \mathbf{E}_p[d]\,\mathbf{S}^T & \mathbf{E}_p[d]^T\,\mathbf{E}_p[d] \end{array}\right]^{-1} \left[\begin{array}{c} \mathbf{S}^T\mathbf{s} \\ \hline \mathbf{E}_p[d]^T\,\mathbf{s} \end{array}\right] \qquad (Eq\ 4.5.21)$$

When the filter coefficients are calculated using Eq 4.5.21, the stability of both the STP and LTP filters must be checked and "correction" procedures must be applied when one of the filters becomes unstable.

Equation 4.5.21 can only be used when the specified value of the delay parameter d does not necessitate the use of current prediction error samples $e_p(i)$ $(d \geq n_c)$. This constraint can sometimes be lifted, and iterative minimisation techniques have been proposed [4.47], that jointly optimise the STP and LTP coefficients when the value of the delay coefficient is smaller than the frame size $(d < n_c)$.

The considerable complexity of the LTP estimation process can be reduced by employing the autocorrelation method to determine the filter coefficients and by using a simpler search method to determine the optimum value of the LTP delay coefficient d. The number of LTP coefficients $\{c_j\}$ is usually set between one and three $(o \leq g \leq 2)$ with best results obtained for a three-tap predictor $(g=2)$. The optimum definition interval (n_c) depends on the bit rate at which the MPE coder operates [4.14], and should in general be greater than 15 ms to allow the LTP to be effective in tracking the periodicity of the speech waveform.

2) OPTIMISING THE LTP COEFFICIENTS BY MINIMISING THE SIGNAL DISTORTION

The modeling error minimised by the MPE optimisation process can be expressed as a fuction of the LTP coefficients $\{c_j\}$ and the pulse amplitudes $\{b_k\}$ by combining Eqs 4.5.7 and 4.5.5 to form :

$$E_w(z) = W(z)\left[S(z)-M_y(z)\right] - A_w(z)C(z)\sum_{j=o}^{g}\left[c_j \sum_{i=o}^{d+j-1} x(i-d-j)z^{-i}\right] - A_w(z)C(z)\sum_{k=1}^{q} b_k z^{-P_k}$$

(Eq 4.5.22)

Since the impulse response of the LTP filter is :

$$C(z) = 1 + \sum_{j=o}^{g} c_j z^{-d-j} + \left[\sum_{j=o}^{g} c_j z^{-d-j}\right]^2 + \ldots \qquad \textit{(Eq 4.5.23)}$$

the error $E_w(z)$ is linearly dependent on the LTP coefficients $\{c_j\}$ only when $d \geq n$ (that is when the second non-zero sample of the LTP impulse response lies outside the MPE frame). Under this restriction, the LTP coefficients and the pulse amplitudes can be jointly optimised by minimising the energy of the distortion $\mathbf{e}_w$. This distortion is defined as :

$$\mathbf{e}_w = \mathbf{W}\left(\mathbf{s}-\mathbf{m}_y\right) - \left[\begin{array}{c|c} \mathbf{A}_w\mathbf{X}[d] & \mathbf{A}_w[q] \end{array}\right]\left[\begin{array}{c} \mathbf{c} \\ \hline \mathbf{b} \end{array}\right] \qquad \textit{(Eq 4.5.24)}$$

where :

$$\mathbf{X}[d] = \begin{bmatrix} x(-d) & x(-d-1) & \ldots & x(-d-g) \\ x(-d+1) & x(-d) & & x(-d-g+1) \\ \vdots & & & \vdots \\ x(-d+n-1) & x(-d+n-2) & \ldots & x(-d-g+n-1) \end{bmatrix} \qquad \textit{(Eq 4.5.25)}$$

The jointly (LS) optimised LTP coefficients and pulse amplitudes are :

$$\left[\begin{array}{c} \mathbf{c} \\ \hline \mathbf{b} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{X}[d]^T\mathbf{A}_w^T\mathbf{A}_w\mathbf{X}[d] & \mathbf{X}[d]^T\mathbf{A}_w^T\mathbf{A}_w[q] \\ \hline \mathbf{A}_w[q]^T\mathbf{A}_w\mathbf{X}[d] & \mathbf{A}_w[q]^T\mathbf{A}_w[q] \end{array}\right]^{-1} \left[\begin{array}{c} \mathbf{X}[d]^T\mathbf{A}_w^T\mathbf{W}\left(\mathbf{s}-\mathbf{m}_y\right) \\ \hline \mathbf{A}_w[q]^T\mathbf{W}\left(\mathbf{s}-\mathbf{m}_y\right) \end{array}\right] \qquad \textit{(Eq 4.5.26)}$$

and the minimum distortion energy is :

$$\left[\mathbf{e}_w^T\,\mathbf{e}_w\right]_{min} = \left(\mathbf{s}-\mathbf{m}_y\right)^T\mathbf{W}^T\mathbf{W}\left(\mathbf{s}-\mathbf{m}_y\right) - \left[\mathbf{c}^T\mathbf{X}[d]^T\mathbf{A}_w^T + \mathbf{b}^T\mathbf{A}_w[q]^T\right]\mathbf{W}\left(\mathbf{s}-\mathbf{m}_y\right)$$

(Eq 4.5.27)

To determine the optimum value of the LTP delay coefficient, the pulse amplitudes are initially assumed to be zero (since the MPE sequence has not been defined yet) and the expression of Eq 4.5.24 is evaluated for the optimum LTP coefficients $\{c_i\}$ (Eq 4.5.23), when d is varied within a predefined interval of integer values.

The optimum value of the LTP delay coefficient remains fixed during the ensuing MPE optimisation process, but the LTP coefficients $\{c_i\}$ can be reoptimised, this time jointly with pulse amplitudes. Eq 4.5.23 can easily be included in the search for the optimum pulse positions and can improve the performance of various MPE coding schemes without significantly increasing their complexity.

Alternatively, the LTP coefficients may remain constant during the MPE optimisation process. In this case, the contribution of the LTP is removed from the speech signal (Eq 4.5.7) before the optimisation process begins, and the pulse positions and amplitudes are estimated by completely disregarding the presence of the LTP.

The complete algorithm of LTP estimation can be considerably simplified by exploiting the similarity between the matrix elements of Eq 4.5.23 for adjacent values of the delay coefficient d [4.41].

The restriction on the minimum value of the delay coefficient ($d \geq n$) can be eased by artificially reconstructing a version of the waveform $X(z)$ for the current MPE frame using its past values, and then proceeding as before using Eqs 4.5.21 and 4.5.22 [4.48,4.49]. The same restriction can also be lifted by using nonlinear optimisation methods to determine the LTP coefficients. The energy of the modeling error (Eq 4.5.22) is then considered as a function of the LTP coefficients, and it is minimised using nonlinear programming techniques.

The LTP filter performes better (in terms of the perceptual quality of the coded speech) when its coefficients are determined by minimising the energy of the distortion instead of the prediction error (mentioned as the previous method). Unfortunately, the distortion minimisation method poses a restriction on the minimum value of the delay coefficient (which should be larger than the size of the MPE frame), and this means that the size of the MPE frame cannot be increased without impeding the performance of the LTP filter. This requirement contrasts with the observed improved performance of

most MPE optimisation algorithms, obtained when larger speech frames are used. A compromise is therefore sought which usually limits the duration of the MPE analysis frame between 5 ms and 10 ms.

The frequent updating of the LTP filter parameters necessitates the use of efficient quantizers for the LTP coefficients. Vector quantizers can be designed based on Euclidian error measures or, for improved efficiency, on the minimisation of the average distortion introduced by the coder.

4.6 Multi-Stage (MS) Optimisation Algorithms

The Multi-Stage algorithms, mentioned in Chapter 3, begin by optimising the position of a single excitation pulse in the first stage, and then add more pulses, optimising their positions in separate stages. The number of pulses is steadily increased, until the required total is reached.

The position of each pulse, once defined cannot be changed in the next optimisation stages. The pulse amplitudes though, can be redefined in later stages either by jointly optimising the amplitudes of a group of pulses, or by allowing additional pulses to be placed at locations already occupied by pulses defined in previous stages. When pulse coincidences are allowed, the pulses that occupy the same position are added together to form a single pulse.

The pulse position optimisation at each stage is done by examining all the possible pulse locations within a speech frame, and choosing the location that results the minimum approximation error (Eq 3.4.2). The complexity of the MS algorithm is determined by the computational effort required to calculate approximation error. The complex MS algorithms are very efficient at high pulse rates (number of pulses per second), and can therefore improve noticeably the performance of a MPE coder operating at a high transmission bit rate. On the other hand, when ease of implementation and cost are a primary consideration, the choice may be restricted to simpler and less efficient algorithms.

Five MS algorithms will be described, starting with the simpler algorithms. The first three allow additional pulses to reinforce or weaken existing pulses, by permitting pulse coincidences, while the last two exclude this possibility. The autocorrelation approximation defined in Eq 4.4.17 can by employed to simplify all five methods. The performance of these algorithms, in terms of Signal to Noise Ratios (SNR), and their complexity, in terms of multiplications/additions, will be compared to the results obtained from Block Search algorithms. It will also be demonstrated how the use of the noise shaping filter $W(z)$, as defined in Eq 4.2.1, can substantially improve the efficiency of simple MS algorithms at high pulse rates, by reducing the dependency of the approximation error estimation on the relative pulse proximity.

A general linear filter model will be assumed during the presentation of

the MPE optimisation algorithms, but the simplifications that result from the use of a single LPC synthesis filter will be pointed out.

1) METHOD MS1

This method is relatively easy to implement and can give results comparable to those obtained from the more complicated MS methods, when the number of pulses per frame is small. Improvements of this method will be presented as methods MS2 and MS3.

The excitation in the first stage of the algorithm is a single pulse which can be placed in any of the n sampling points within a MPE analysis frame. The error, as defined by Eqs 4.3.5 and 4.4.1, is a function of the pulse position and can be expressed as :

$$\mathbf{e}_w[i] = \mathbf{y}_o - b_1(i)\ \mathbf{f}[i] \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.1)$$

where :

$$\mathbf{y}_o = \mathbf{W}\left(\mathbf{s}-\mathbf{m}_y\right) \qquad (Eq\ 4.6.2)$$

is the "target"signal which is compared to the output of the zero-initial-state MSF, and :

$$\mathbf{f}[i] = \left[\ o,o,\ldots,h_w(1),h_w(2),\ldots,h_w(n-i-1)\ \right] \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.3)$$

is the impulse response of the MSF $A_w(z)$, shifted by i samples. The solution to the normal equations 4.4.2 is simply :

$$b_1(i) = \frac{c_o(i)}{\Phi(i,i)} \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.4)$$

where :

$$c_o(l) = \mathbf{y}_o^T\ \mathbf{f}[l] \quad , \quad o \le l \le n-1 \qquad (Eq\ 4.6.5)$$

is the cross-correlation between the signal $\mathbf{y}_o$ and the shifted impulse response of the MSF, and $\Phi(i,j)$ is the auto-covariance matrix defined in Eq 4.4.11. The error energy can be found by combining Eqs 4.4.4 and 4.6.3 to form the approximation error function for the first pulse :

$$\varepsilon_1(i) = \left[\ \mathbf{e}_w[i]^T\ \mathbf{e}_w[i]\ \right]_{min} = \mathbf{y}_o^T\ \mathbf{y}_o - \frac{c_o(i)^2}{\Phi(i,i)} \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.6)$$

The expression of Eq 4.6.6 has to be calculated for each of the n possible

pulse positions in order to find the position where the minimum error occurs. Since the first term of the right hand side of Eq 4.6.6 is constant, the optimum pulse position must be that one which maximises the second term. Once the optimum position has been determined, the pulse amplitude is calculated using Eq 4.6.4.

In the second stage, the position p_1 and the amplitude b_1 of the first pulse are kept constant, and the error becomes a function of the position and amplitude of the second pulse :

$$\mathbf{e}_w[i] = \mathbf{y}_1 - b_2(i)\ \mathbf{f}[i] \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.7)$$

where :

$$\mathbf{y}_1 = \mathbf{y}_o - b_1\ \mathbf{f}[p_1] \qquad (Eq\ 4.6.8)$$

The optimum amplitude of the second pulse at position i can be found by minimising the error energy, assuming that the position and amplitude of the first pulse are fixed. This optimum amplitude is :

$$b_2(i) = \frac{c_1(i)}{\Phi(i,i)} \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.9)$$

where :

$$c_1(i) = \mathbf{y}_1^T\ \mathbf{f}[i] = c_o(i) - b_1\ \Phi(p_1, i) \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.10)$$

and the approximation error function for the second pulse is :

$$\varepsilon_1(i) = \left[\ \mathbf{e}_w[i]^T\ \mathbf{e}_w[i]\ \right]_{min} = \mathbf{y}_1^T\ \mathbf{y}_1 - \frac{c_1(i)^2}{\Phi(i,i)} \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.11)$$

The optimum position of the second pulse is found by locating the minimum of the approximation error function defined in Eq 4.6.11, or equivalently, by maximising the second term of the right hand side of Eq 4.6.11.

In a similar way, assuming that q pulses have already been defined and that their positions and amplitudes remain constant, the error can be expressed as a function of the position and amplitude of the $q+1$ pulse :

$$\mathbf{e}_w[i] = \mathbf{y}_q - b_{q+1}(i)\ \mathbf{f}[i] \quad , \quad o \le i \le n-1 \qquad (Eq\ 4.6.12)$$

where :

$$\mathbf{y}_q = \mathbf{y}_{q-1} - b_q\ \mathbf{f}[p_q] = \mathbf{y}_o - \mathbf{A}_w[q]\ \mathbf{b} \qquad (Eq\ 4.6.13)$$

The optimum amplitude of the $q+1$ pulse is :

$$b_{q+1}(i) = \frac{c_q(i)}{\Phi(i,i)} \quad , \quad 0 \leq i \leq n-1 \qquad \textit{(Eq 4.6.14)}$$

where :

$$c_q(i) = \mathbf{y}_q^T \mathbf{f}[i] = c_{q-1}(i) - b_q \Phi(p_q, i) \quad , \quad 0 \leq i \leq n-1 \qquad \textit{(Eq 4.6.15)}$$

and the approximation error function for the same pulse is :

$$\varepsilon_{q+1}(i) = \mathbf{y}_q^T \mathbf{y}_q - \frac{c_q(i)^2}{\Phi(i,i)} \quad , \quad 0 \leq i \leq n-1 \qquad \textit{(Eq 4.6.16)}$$

The iterative relationships of Eqs 4.6.13-4.6.16 form the basis of the MSl algorithm and are employed once in every optimisation stage. This does not necessarily imply that a new pulse is added at each stage, as it is possible to determine an optimum pulse position that is the same as the position of an existing pulse. When a pulse coincidence occurs, the amplitudes of the two pulses are added and the total number of pulses does not change.

Pulse coincidences are more likely to happen when the number of pulses defined in each analysis frame is increased. As a consequence, the number of stages required to construct an excitation of q pulses is disproportionately inreased when q becomes large.

A flow diagram of the complete algorithm is presented in Fig 4.6.1. The subscripts denoting the stage number have been dropped because the updated array values can replace the values calculated in the previous stage. An n-sample excitation sequence $\{x(i)\}$ is defined and initially zeroed. The pulses defined in each stage are added to the excitation sequence, thus when two pulses coincide their amplitudes are added together. The auto-covariance matrix $\Phi(i,j)$ can be precalculated and stored using the efficient procedure described in Section 4.4 or, if storage is limited, the required elements can be calculated in every stage.

The cross-correlation values are stored in the array $\mathbf{c} = [c(o), \ldots, c(n-1)]$ and are updated in every stage. As seen in Fig 4.6.1, there are two equivalent ways of updating the cross-correlation array. In the first method, the scaled auto-covariance values are subtracted from the cross-correlation $c_{q-1}(i)$ (Eq 4.6.15). The second method first updates the signal component

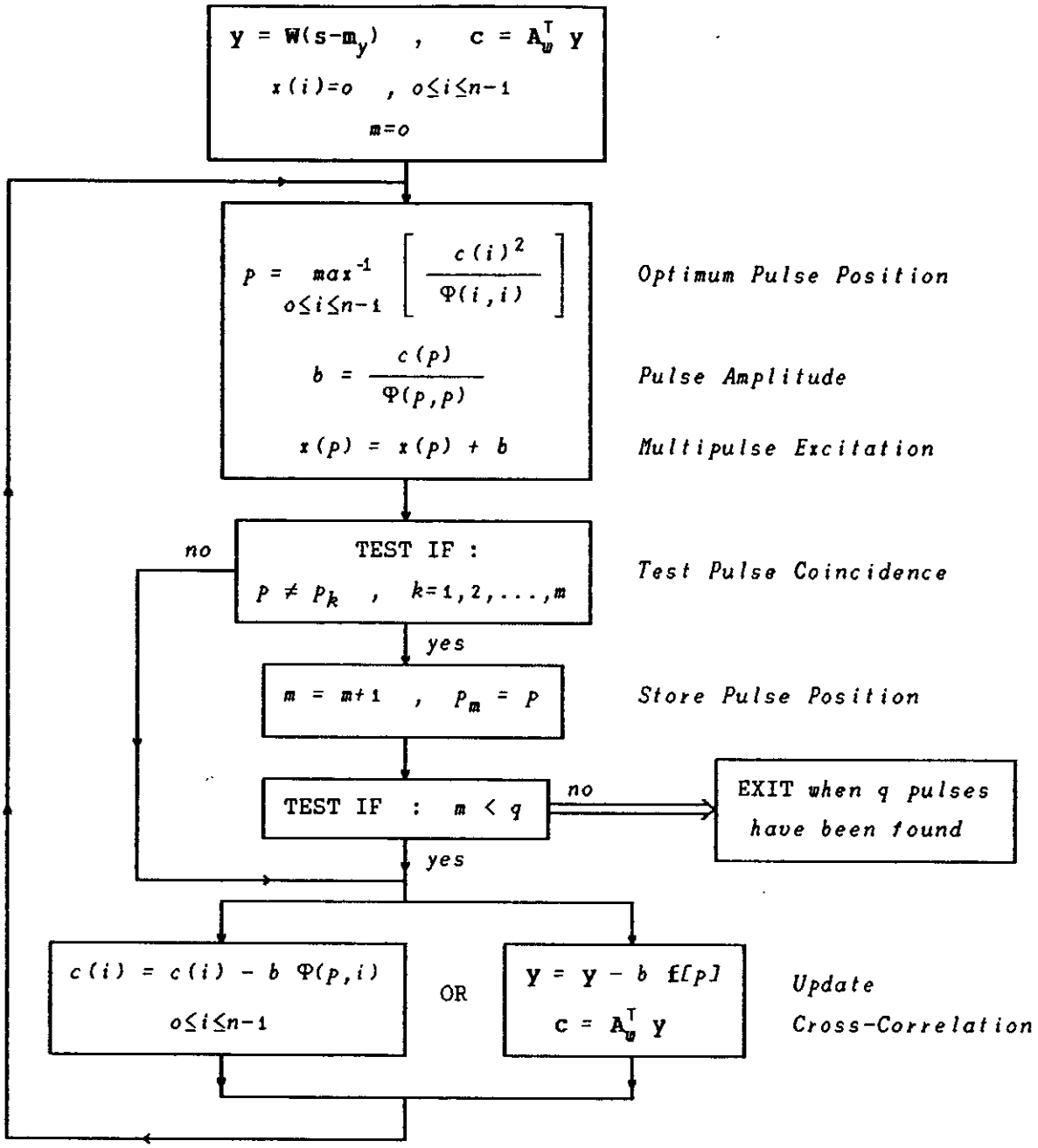


FIGURE 4.6.1 The MPE Optimisation Method MS1.

$y_{q-1}(i)$ (Eq 4.6.13) and then computes its cross-correlation with the shifted impulse response of the MSF (Eq 4.6.15). Which of the two methods is more efficient computationally, depends on the coder parameters chosen for a particular application.

It is interesting to note that the value of the cross-correlation $c_q(i)$ is zeroed at the position where the pulse is placed. For example, using Eqs 4.6.14 and 4.6.15, the value of the cross-correlation at the optimum position of the q pulse is :

$$c_q(P_q) = c_{q-1}(P_q) - \frac{c_{q-1}(P_q)}{\Phi(P_q,P_q)} \Phi(P_q,P_q) = 0 \qquad \textit{(Eq 4.6.17)}$$

It is therefore certain that the $q+1$ pulse will not be placed at the same position as the pulse defined in the previous stage. It cannot be guaranteed though that the optimum position of the $q+1$ pulse will not be the same as the position of another previously defined pulse, because in general :

$$c_q(i) \neq 0 \quad , \quad i \neq P_q \qquad \textit{(Eq 4.6.18)}$$

To eliminate all pulse coincidences the pulse amplitudes must be adjusted so that the corresponding cross-correlation values become zero :

$$c_q(i) = 0 \quad , \quad i \in \left[P_1, P_2, \ldots, P_q \right] \qquad \textit{(Eq 4.6.19)}$$

This can be done by solving the system of normal equations 4.4.2 to obtain the jointly optimised pulse amplitudes. This technique will be described as method MS4.

The effect of increasing the number of pulses per frame on the number of pulse coincidences can be measured by finding how the average number of stages per frame changes when the required number of pulses per frame is increased. The results given below were taken from a long speech data training set of eight male and seven female speakers. The speech signal is low-passed to 3.4 kHz and sampled at 8 kHz. A MPE analysis frame of 100 samples and a non-overlapping LPC analysis frame of 200 samples are used (at 8 kHz sampling rate). The order of the single LPC filter is 12 and the constant γ of the noise shaping filter $W(z)$ is set equal to 0.9 :

Pulses per Frame	*5*	*10*	*15*	*20*	*25*	*30*
Average Number of Stages per Frame	*5.0*	*10.3*	*16.5*	*24.1*	*33.4*	*44.8*

A disproportionate increase in the number of stages per frame can be observed at high pulse rates. The same effect is observed when the average number of stages required to bring the SNR in every frame to a preset level, is plotted over a wide range of SNRs (Fig 4.6.2(a)). The logarithm (to the base of 10) of the average number of stages is plotted in Fig 4.6.2(a) and the almost linear relationship indicates that the number of stages increases exponentially as the SNR level is increased. Consequently this is translated to an exponential increase in the computational complexity of the algorithm.

Both the LPC analysis and MPE analysis frames in Fig 4.6.2(a) are of 128 samples, and the constant γ is set equal to one. In Figures 4.6.2(b),(c) and (d), the value of γ is set to γ=0.9, γ=0.8 and γ=0.6 respectively. To provide a more meaningful comparison and to compress the large range of values, it is the logarithm of the ratio of two values that is plotted. The reference value is the value of the average number of stages when γ=1, and the values of the average number of stages when γ=0.9, γ=0.8 and γ=0.6 are divided by the reference value and the logarithm (to the base of 10) of the ratio is plotted.

It is evident that at high SNRs, the performance of the MS1 algorithm can be dramatically improved by lowering the value of the constant γ. That happens because by reducing the value of γ, the effective duration of the impulse response of the Modified Synthesis Filter $A_m(z)$ is also reduced. The assumption that the pulse amplitudes once determined remain fixed, becomes very unrealistic when two pulses approach each other, because their interdependence is not properly taken into account (by jointly optimising their amplitudes). By reducing the duration of the MSF impulse response, their interdependence is minimised and the efficiency of the pulse position optimisation algorithm is improved. This improved efficiency is especially noticeable when the number of pulses defined in each frame is large (high SNR), because it is quite likely that during the search for the optimum pulse position, locations close to existing pulses will be examined. In the extreme case when γ=0, a closed form solution for the pulse positions exists (Eq 4.4.13).

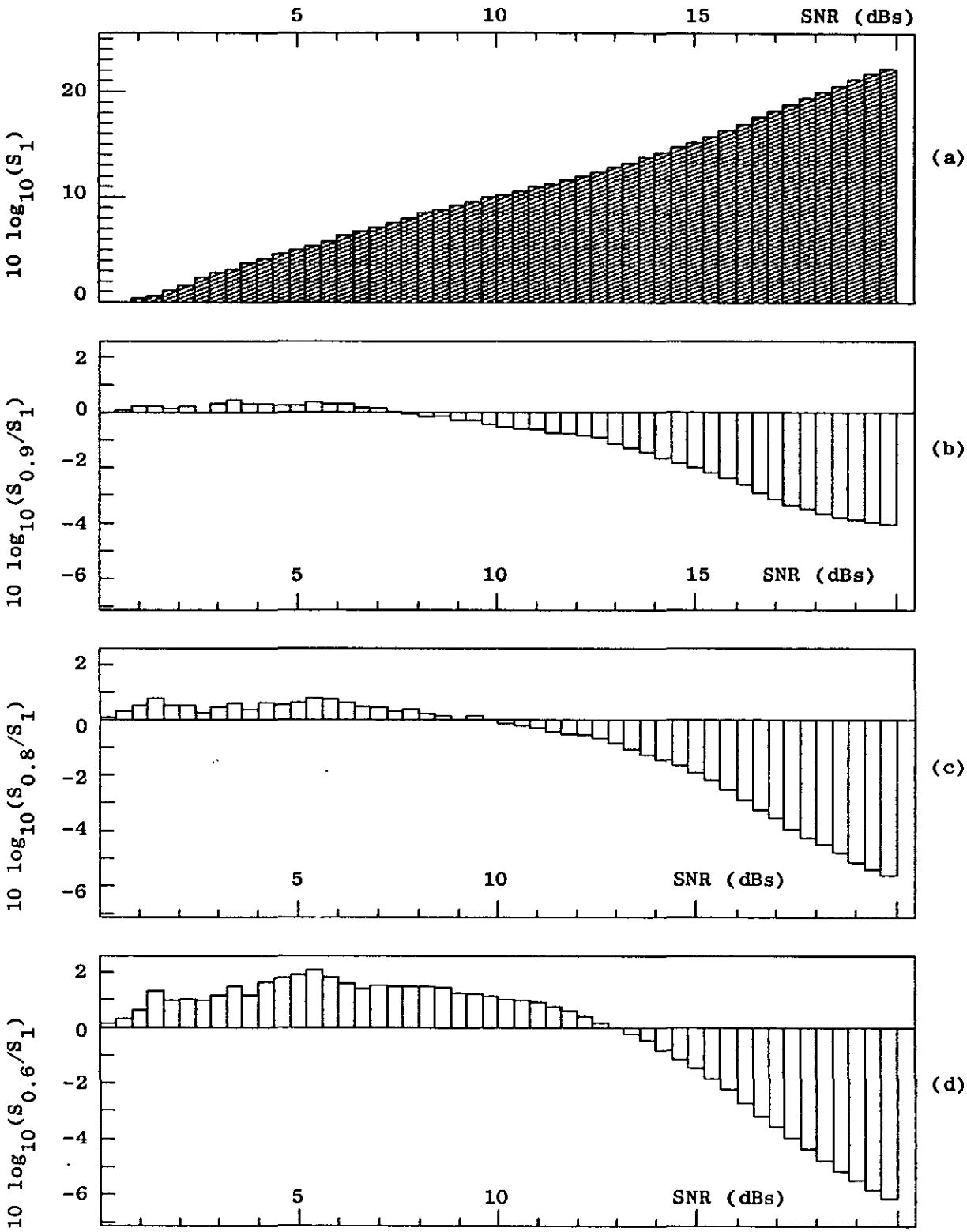


FIGURE 4.6.2 Plots of the average number of stages S_γ required by method MS1 to reach the level of SNR given on the horizontal axis. S_1 corresponds to $\gamma=1$, $S_{0.9}$ to $\gamma=0.9$, $S_{0.8}$ to $\gamma=0.8$ and $S_{0.6}$ to $\gamma=0.6$

The improved performance of the pulse position optimisation process is opposed by the reduction in SNR caused due to the fact that when the value of γ is small, the error-measure minimised is not closely related to the actual distortion introduced by the coder. Therefore when the value of γ is reduced below a certain level, the performance of the algorithm starts to deteriorate, instead of improving. This can be observed by comparing Figures 4.6.2(a), (b) and (c). If the optimum value of the constant γ for a certain SNR level, is the one that minimises the number of stages required to reach that level, then it is clear that the optimum value of γ decreases as the SNR level (or equivalently the number of pulses) inreases. A small value of γ only becomes advantageous when a relatively high SNR level is required.

Methods that jointly estimate the pulse amplitudes while optimising the pulse positions, do not usually benefit from the use of low γ values and perform best when γ is close to one. This will be demonstrated when method MS5 is examined.

The autocorrelation approximation introduced in Section 4.4 can readily be used to simplify the MS1 algorithm. The symmetric auto-covariance matrix $\Phi(i,j)$ in Fig 4.6.1 can be replaced by the Toeplitz matrix $\Phi_a(i,j)$ defined in Eq 4.4.17. The optimum pulse position can then be determined at each stage by finding where the absolute maximum of the cross-correlation occurs (Eq 4.6.16), that is why this method has been described as the maximum cross-correlation MPE algorithm [4.50]. The n auto-covariance values have to be calculated only once at the start of the algorithm, and the cross-correlation array can be efficiently updated by subtracting the scaled auto-covariance values from the cross-correlation values calculated in the previous stage. This simplified method will be referred to as method MS1a.

2) EFFICIENT CALCULATION OF THE CROSS-CORRELATION AND AUTO-COVARIANCE

The cross-correlation $\mathbf{c}$ between a given signal $\mathbf{y}$ and the shifted impulse response of the MSF can be performed efficiently by filtering the time reversed signal sequence through the filter $A_w(z)$. Since the convolution matrix $\mathbf{A}_w$ is symmetrical about its cross-diagonal (persymmetric), it can be represented as :

$$\mathbf{A}_w^T = \mathbf{P}\,\mathbf{A}_w\,\mathbf{P} \qquad \textit{(Eq 4.6.20)}$$

where **P** is a unit matrix with ones along its cross-diagonal. The cross-correlation array in Fig 4.6.1 can then be calculated as :

$$\mathbf{c} = \mathbf{A}_w^T \mathbf{y} = \mathbf{P} \left[\mathbf{A}_w \left(\mathbf{P}\, \mathbf{y} \right) \right] \qquad \textit{(Eq 4.6.21)}$$

which, starting from the inner brackets, is equivalent to a time reversal of the signal **y**, followed by a convolution with the MSF and a second time reversal. If the order of the filter $A_w(z)$ is much smaller than the size of the MPE analysis frame, then the convolution operation (found in Eqs 4.6.5 and 4.6.10) requires less computations than a direct calculation of the cross-correlation.

Similar computational savings can be achieved in the calculation of the auto-covariance of the MSF impulse response. The signal used in the computation is now the MSF impulse response itself :

$$[\ \Phi(i,o),\Phi(i,1),\ldots,\Phi(i,n-1)\ ]^T = \mathbf{P} \left[\mathbf{A}_w \left(\mathbf{P}\ \mathbf{f}[i] \right) \right] \quad , \quad o \leq i \leq n-1 \qquad \textit{(Eq 4.6.22)}$$

Eq 4.6.22 can be used to calculate the first row of the matrix $\Phi(i,j)$ and the efficient recursive formula of Eq 4.4.16 can be used to derive the rest of the matrix elements. When the autocorrelation approximation is applied, the matrix $\Phi_a(i,j)$ is Toeplitz and is therefore defined from its first row.

3) METHODS MS2 AND MS3

Both MS2 and MS3 methods are based on method MS1 but try to avoid some of its shortcomings. The reason method MS1 becomes inefficient at high pulse rates, is that when the excitation pulses approach each other their interaction is not accurately compensated for, and the approximation error measured is larger than it would be if the pulse amplitudes were jointly optimised. As a consequence, some of the pulses may need to be readjusted in later optimisation stages by bringing additional pulses to the same locations.

To avoid the pulse coincidences, the pulses amplitudes may, at some stage during the MPE optimisation, be readjusted. This form of delayed amplitude correction allows the pulses derived during the first stages, to take into account the presence of pulses defined in later stages.

Various pulse correction schemes have been proposed that involve a "joint" or "repeated" amplitude reoptimisation at the end of each stage of the algorithm. These schemes reoptimise a small group of pulses [4.51,4.14] or all the excitation pulses [4.13,4.52]. A method that jointly reoptimises all the pulse amplitudes at the end of each stage, will be later presented as method MS4.

A simpler approach has been adopted in methods MS2 and MS3. A repeated pulse amplitude reoptimisation can be applied in such a way as to maximise the reduction in the approximation error. To avoid solving a system of equations, a single pulse is identified which causes the maximum drop in the value of the approximation error, when its amplitude is corrected. The same process can be repeated a number of times to identify more pulses . Notice that the pulse amplitudes, as defined by this process, will eventually converge to the set of values that would have been obtained if all the pulses were jointly reoptimised by solving the system of normal equations in Eq 4.4.2.

The sequential selection of the pulses to be reoptimised can be achieved using Eq 4.6.16, which provides the approximation error as a function of the $q+1$ pulse position, and takes into account that q pulses have already been defined. However, instead of searching over the whole range of possible pulse positions to find where the error minimum occurs, the search is now restricted to the positions already occupied by pulses. Clearly the pulse whose amplitude correction causes the maximum error reduction is :

$$k = \max_{1 \le i \le q}^{-1} \left[\frac{c_q(p_i)^2}{\Phi(p_i, p_i)} \right] \qquad \textit{(Eq 4.6.23)}$$

The corrected pulse amplitude is formed from its initial amplitude plus the amplitude of an additional pulse placed in the same position :

$$b_k' = b_k + b_{cor} \qquad \textit{(Eq 4.6.24)}$$

where :

$$b_{cor} = \frac{c_q(p_k)}{\Phi(p_k, p_k)} \qquad \textit{(Eq 4.6.25)}$$

The updated cross-correlation sequence is :

$$c'_q(i) = \mathbf{A}_w^T \left[\mathbf{y}_q - b_{cor}\, \mathbf{f}[p_k] \right] = c_q(i) - b_{cor}\Phi(p_k,i) \quad , \; o \le i \le n-1 \qquad (Eq\ 4.6.26)$$

which is zero at p_k but not at p_q (where $c_q(i)$ is zero). Another pulse is identified by applying Eq 4.6.23 to the updated cross-correlation sequence $\{c'_q(i)\}$. If a series of adjustments is performed, the value of the cross-correlation will eventually become zero at every pulse location p_k, and the pulse amplitudes will converge towards a set of "optimum" values. This convergence is possible because the values of the autocovariance $\Phi(i,j)$ are decaying as the pulses move away from each other, and this causes the repeated corrections to become increasingly smaller in size.

As mentioned earlier, the set of "optimum" pulse amplitudes obtained (at the limit) by repeating the pulse selection procedure many times, is the same as the set of amplitudes derived by solving the system of normal equations in Eq 4.4.2. This happens because the satisfaction of the conditions set by Eq 4.4.2 ensures the orthogonality of the error vector $\mathbf{y}_q$ to the subspace defined by the signal components $\mathbf{f}[p_m]$, and hence :

$$\mathbf{y}_q^T\, \mathbf{f}[p_m] = c_q(p_m) = o \quad , \quad m = 1,2,\ldots,q \qquad (Eq\ 4.6.27)$$

The advantage gained by using the repeated pulse reoptimisation method is that a direct solution of the system of normal equations is avoided and the performance/complexity of the algorithm can be varied by altering the number of pulse-amplitude corrections, i.e. the of times the expressions in equations 4.6.23-4.6.26 are evaluated.

Method MS2 first defines all the q pulse positions using method MS1, and then applies Eqs 4.6.23-4.6.26 to reoptimise the pulse amplitudes. Since all the pulse corrections are made after the last stage of the MS1 algorithm, no more than q cross-correlation values need to be updated. Eq 4.6.26 therefore changes to :

$$c'_q(p_m) = c_q(p_m) - b_{cor}\Phi(p_k,p_m) \quad , \quad 1 \le m \le q \qquad (Eq\ 4.6.28)$$

and as a result, the computational complexity of the algorithm is greatly reduced. A reasonable compromise between low algorithm complexity and good performance can be achieved by setting the number of corrective iterations (application of Eqs 4.6.23-4.6.25 and Eq 4.6.28) equal to twice the number

of pulses per MPE analysis frame. In this case, the performance of the algorithm is comparable to that obtained when the amplitudes are reoptimised by solving the system of normal equations. However, the complexity of the amplitude reoptimisation stage of method MS2, increases proportionately to the square of the number of pulses per MPE frame, while the solution of the normal equations would require a number of operations proportional to the third power of the number of pulses.

Method MS3 applies the pulse correction equations 4.6.23-4.6.26 after the position of a new pulse has been determined. As seen in Fig 4.6.3, where the flow diagram of method MS3 is presented, the pulse positions are optimised in exactly the same way as in method MS1. When a new pulse is located, Eqs 4.6.23-4.6.26 are used to iteratively reoptimise the amplitudes of all the pulses that have been defined up to that stage. The number of corrective iterations $n(m)$, is a function of the number of pulses m that have already been defined, and should in general increase as more pulses are added to the excitation during the MPE optimisation. Finally after the last optimisation stage, the cross-correlation can be updated as efficiently as in method MS2, since the positions of all q pulses have been determined.

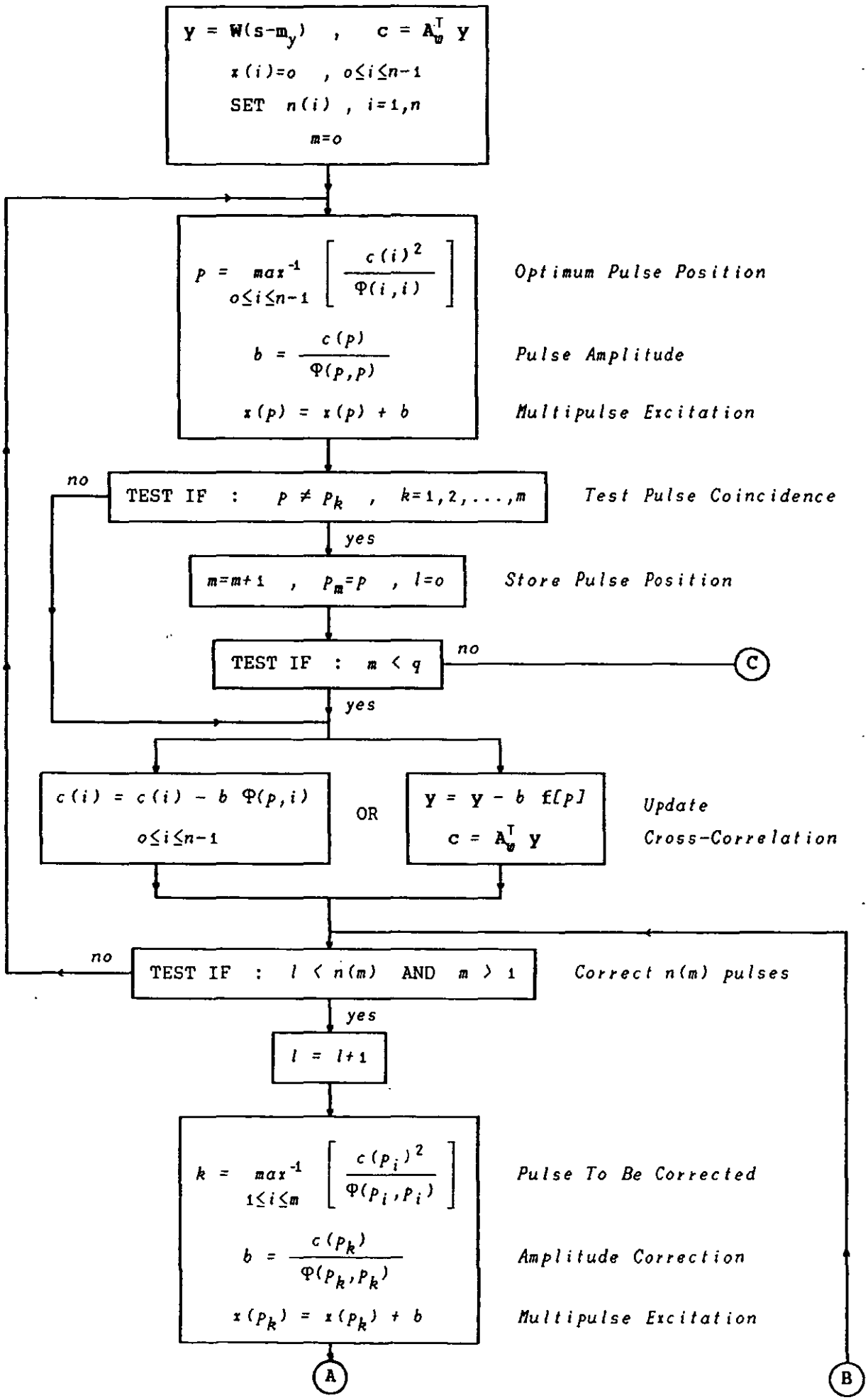
An exponential model is used to derive the required number of corrective iterations, applied when a new pulse is found. This model has been chosen in order to compensate for the exponentially increasing inefficiency of the amplitude estimation process at high pulse rates, which was observed when method MS1 was examined. The number of iterations is given by :

$$n(m) = \text{int}\left[\rho \left(\exp(\partial_1 m) - 1 \right) \right] \quad , \quad m = 1, 2, \ldots, q \qquad \text{(Eq 4.6.29)}$$

where $\text{int}[\ldots]$ gives the integer part of a real number, and ρ is a constant adjusted so that the total number of iterations is :

$$\sum_{m=1}^{q} n(m) = \partial_2 \qquad \text{(Eq 4.6.30)}$$

The value of ∂_1 can be determined experimentally, by maximising the average segmental SNR (Seg-SNR) over a speech data training set, for a fixed value of ∂_2. Values for ∂_1 and ∂_2 were chosen to optimise the performance of the algorithm when a MPE frame of 100 samples is used. When a different


$y = W(s-m_y)$, $c = A_w^T y$
$x(i)=o$, $o \le i \le n-1$
SET $n(i)$, $i=1,n$
$m=o$
$p = \max_{o \le i \le n-1}^{-1} \left[\frac{c(i)^2}{\Phi(i,i)} \right]$
Optimum Pulse Position
$b = \frac{c(p)}{\Phi(p,p)}$
Pulse Amplitude
$x(p) = x(p) + b$
Multipulse Excitation
no
TEST IF : $p \neq p_k$, $k=1,2,\ldots,m$
Test Pulse Coincidence
yes
$m=m+1$, $p_m=p$, $l=o$
Store Pulse Position
TEST IF : $m < q$
no
C
yes
$c(i) = c(i) - b\ \Phi(p,i)$
$o \le i \le n-1$
OR
$y = y - b\ f[p]$
$c = A_w^T y$
Update
Cross-Correlation
no
TEST IF : $l < n(m)$ AND $m > 1$
Correct n(m) pulses
yes
$l = l+1$
$k = \max_{1 \le i \le m}^{-1} \left[\frac{c(p_i)^2}{\Phi(p_i,p_i)} \right]$
Pulse To Be Corrected
$b = \frac{c(p_k)}{\Phi(p_k,p_k)}$
Amplitude Correction
$x(p_k) = x(p_k) + b$
Multipulse Excitation
A
B

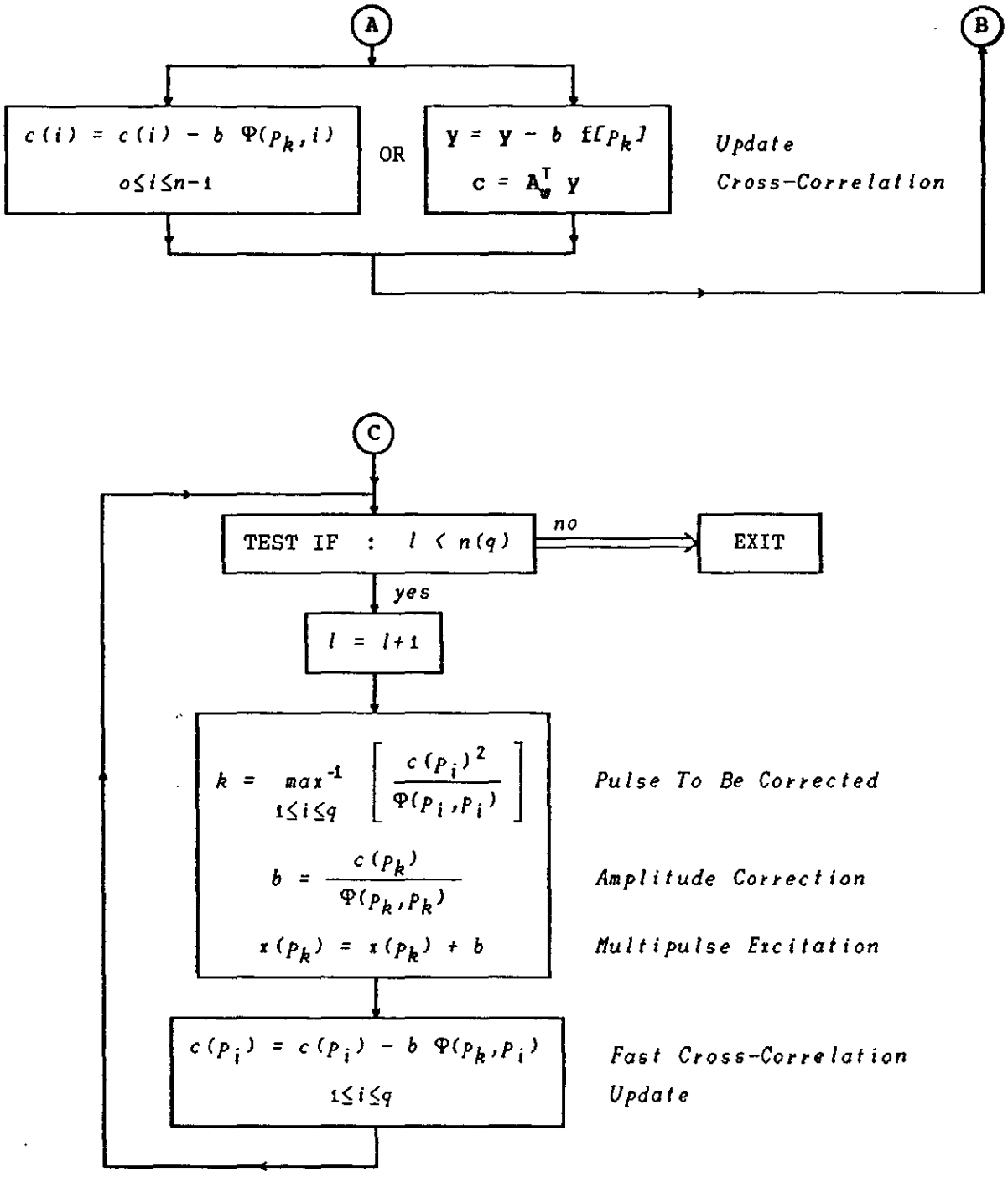


FIGURE 4.6.3 The MPE Optimisation Method MS3.

frame size is used, ∂_2 must be scaled up and ∂_1 must be scaled down by a factor equal to the ratio of the frame size to a frame of 100 samples. The values chosen for ∂_1 and ∂_2 (for a MPE frame of 100 samples and a sampling rate of 8 kHz), can be expressed as functions of the number of pulses per frame :

$$\partial_1 = \frac{5}{q} \qquad (Eq\ 4.6.31)$$

and

$$\partial_2 = \text{int}\left[6 \left(\exp(0.16\ q)-1\right) \right] \qquad (Eq\ 4.6.32)$$

The value chosen for ∂_2 results a low computational complexity without compromising the performance of the algorithm. The algorithm performance is close to the performance that would be achieved if a joint amplitude reoptimisation was performed at the end of each stage (method MS4), by solving the system of normal equations. The overall complexity though is still relatively low because of the relatively smaller number of corrective iterations involved. For example, when 5,10,15 or 20 pulses are defined in each MPE frame, the number of iterations corresponding to each new excitation pulse is, according to Eqs 4.6.29-4.6.32 :

Pulse Number	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Pulses/Frame 5	*0*	*0*	*0*	*2*	*5*	← *CORRECTIVE ITERATIONS*														
10	*0*	*0*	*0*	*0*	*0*	*1*	*2*	*3*	*6*	*11*										
15	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*2*	*3*	*5*	*7*	*9*	*13*	*19*					
20	*0*	*0*	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*2*	*3*	*4*	*5*	*7*	*9*	*12*	*16*	*20*	*26*	*34*

In the initial stages, no amplitude reoptimisation is required because the excitation pulses are usually spaced far apart. Most of the corrective iterations are concentrated in the final stages of the algorithm, where pulse coincidences are more likely to happen and the pulse amplitude estimation process is most inefficient. The fact that a large number of corrective iterations is used in the final stage where the cross-correlation updating is very efficient (Eq 4.6.28), indicates that method MS3 is a relatively low-complexity high-performance MPE optimisation method.

The use of the amplitude reoptimisation technique by method MS3, reduces the values of the cross-correlation at the pulse locations and as a result,

the number of pulse coincidences is reduced. Compared to the numbers given for method MS1, the average number of optimisation stages in each MPE frame is (for method MS3) :

Pulses per Frame	*5*	*10*	*15*	*20*	*25*	*30*
Average Number of Stages per Frame	*5.0*	*10.0*	*15.1*	*20.2*	*25.3*	*30.3*

These figures show a significant reduction in the number of pulse coincidences which contributes to the improved performance of method MS3.

Both MS2 and MS3 methods can be modified to take advantage of the autocorrelation approximation (Eq 4.4.17), in the same way as method MS1 was. The two modified methods will be referred to as methods MS2a and MS3a.

4) GEOMETRICAL INTERPRETATION OF METHODS MS2 AND MS3

A geometrical visualisation of the repeated pulse adjustment performed by methods MS2 and MS3, is shown in Fig 4.6.4 for the case of a 3-dimensional vector space. The axes correspond to the vectors $f[o]$, $f[1]$ and $f[2]$, derived from the impulse response of the MSF. The unit vector along each axis is :

$$u[i] = \frac{f[i]}{\sqrt{\Phi(i,i)}} \quad , \quad i=o,1,2 \qquad (Eq\ 4.6.33)$$

and the magnitude of the projection of the signal vector $\mathbf{y}_o=[\, y_o(1), y_o(2), y_o(3)\,]$ onto each axis is:

$$t_o(i) = \frac{\mathbf{y}_o^T\, f[i]}{\sqrt{\Phi(i,i)}} \quad , \quad i=o,1,2 \qquad (Eq\ 4.6.34)$$

Finding the projection with the largest absolute magnitude is equivalent to minimising the approximation error in Eq 4.6.6, with respect to the position of the first pulse. If the first pulse is chosen along the $f[o]$ axis, then the error signal $\mathbf{y}_1$ is formed by subtracting the projection of $\mathbf{y}_o$ onto $f[o]$, from $\mathbf{y}_o$:

$$\mathbf{y}_1 = \mathbf{y}_o - t_o(o)\, u[o] = \mathbf{y}_o - \frac{\mathbf{y}_o^T\, f[o]}{\Phi(o,o)}\, f[o] = \mathbf{y}_o - b_1\, f[o] \qquad (Eq\ 4.6.35)$$

The projection of the new vector $\mathbf{y}_1$ onto the axis $f[o]$ is now zero, and

this corresponds to the fact that the cross-correlation between the signals $\mathbf{y}_1$ and $\mathbf{f}[o]$ is also zero. The second pulse is similarly chosen along $\mathbf{f}[1]$ and the error signal is formed by subtracting the projection of $\mathbf{y}_1$ onto $\mathbf{f}[1]$, from $\mathbf{y}_1$:

$$\mathbf{y}_2 = \mathbf{y}_1 - t_1(1)\ \mathbf{u}[1] = \mathbf{y}_1 - \frac{\mathbf{y}_1^T\ \mathbf{f}[1]}{\Phi(1,1)}\ \mathbf{f}[1] = \mathbf{y}_1 - b_2\ \mathbf{f}[1] \qquad (Eq\ 4.6.35)$$

The projection of $\mathbf{y}_2$ onto axis $\mathbf{f}[1]$ is zero, but its projection onto axis $\mathbf{f}[o]$ has a magnitude of :

$$t_2(o) = \frac{\mathbf{y}_2^T\ \mathbf{f}[o]}{\sqrt{\Phi(o,o)}} = \frac{\mathbf{y}_o^T\ \left[\ \Phi(o,1)^2\ \mathbf{f}[o] - \Phi(o,o)\Phi(o,1)\ \mathbf{f}[1]\ \right]}{\Phi(o,1)\Phi(o,o)\sqrt{\Phi(o,o)}} \qquad (Eq\ 4.6.36)$$

which cannot be zero because the two vectors $\mathbf{f}[o]$ and $\mathbf{f}[1]$ are not parallel. A correction can therefore be applied by subtracting this residual error component from $\mathbf{y}_2$, to form a new error signal :

$$\mathbf{y}_2' = \mathbf{y}_2 - t_2(o)\ \mathbf{u}[o] \qquad (Eq\ 4.6.37)$$

This causes the amplitude of the first pulse to be modified to :

$$b_1' = b_1 + \frac{\mathbf{y}_2^T\ \mathbf{f}[o]}{\Phi(o,o)} \qquad (Eq\ 4.6.38)$$

The process can be repeated by alternatively correcting the pulse amplitudes b_1 and b_2, forcing the projection of the error signal $\mathbf{y}_2'$ onto each of the axes $\mathbf{f}[o]$ and $\mathbf{f}[1]$, to become increasingly smaller. The error signal will eventually converge to $\mathbf{y}_o - \mathbf{P}_r\mathbf{y}_o$ which is the error vector orthogonal to the subspace defined by $\mathbf{f}[o]$ and $\mathbf{f}[1]$. The sum of the projections along each axis will converge to the optimum values :

$$opt(m) = \frac{\mathbf{f}[m-1]^T}{\sqrt{\Phi(m-1,m-1)}} \left[\ \sum_{i=o}^{\infty} \mathbf{y}_{2i+m-1}\ \right] \quad , \quad m=1,2 \qquad (Eq\ 4.6.39)$$

which correspond to the optimum amplitudes of the first two pulses. The values $opt(1)$ and $opt(2)$, are coordinates of the point $\mathbf{P}_r\mathbf{y}_o$, which is the orthogonal projection of the point $\mathbf{y}_o$ onto the 2-dimensional subspace defined by the vectors $\mathbf{f}[o]$ and $\mathbf{f}[1]$. These two values correspond to the set of pulse amplitudes that could have been directly obtained by solving the

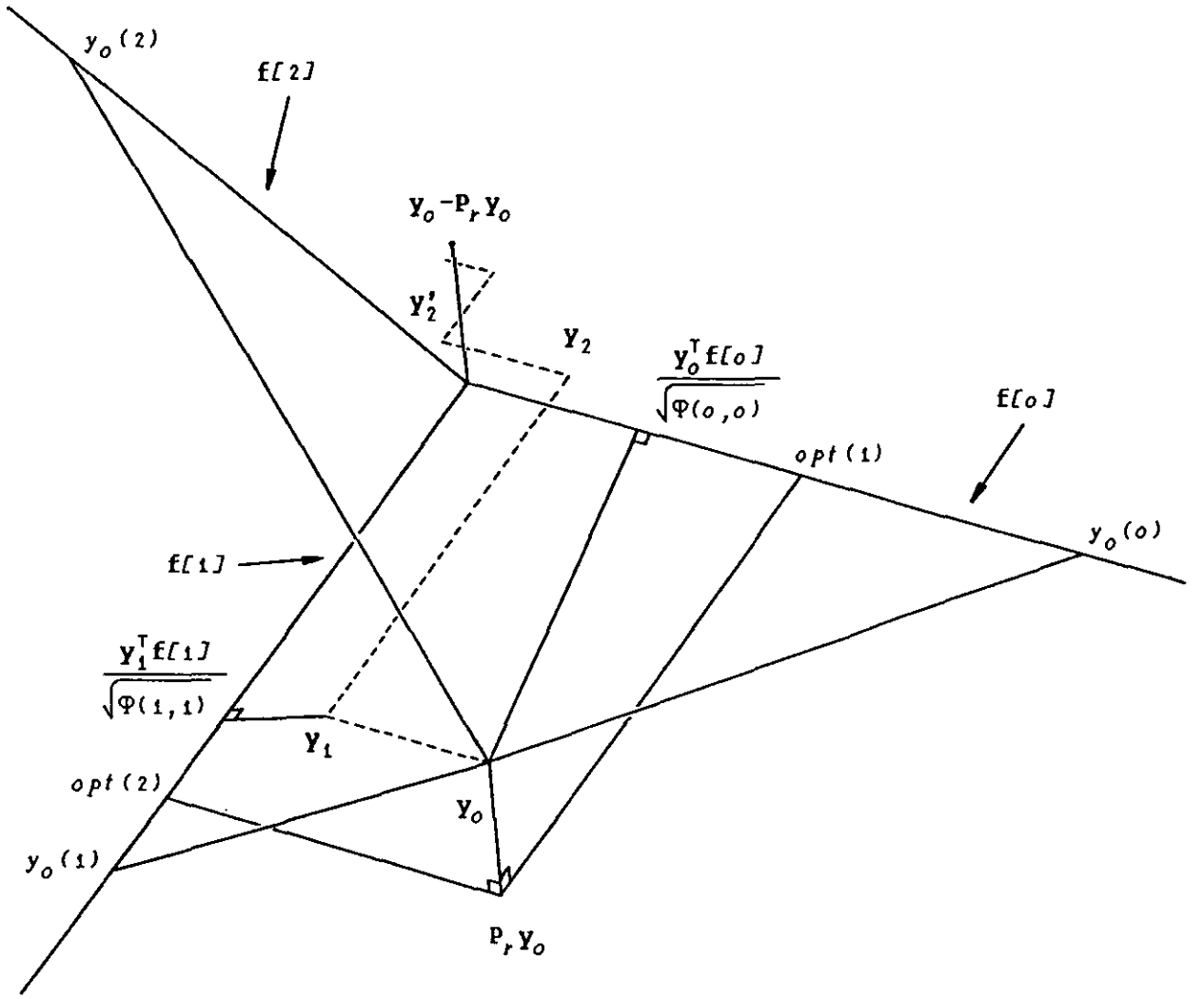


FIGURE 4.6.4 A 3-dimensional interpretation of methods MS2 and MS3.

system of normal equations (Eq 4.4.2).

A generalization of these results into higher dimensions can be carried out in a straight forward manner.

5) METHODS MS4 AND MS5

These two Multi-Stage optimisation methods differ from the previous three, in that they solve the system of normal equations (Eq 4.4.2) in order to optimise the pulse amplitudes. Method MS4 defines the position and amplitude of a single pulse in each stage, and then reoptimises the amplitudes of all the pulses found up to that stage, by solving the system of normal equations. Method MS4 however, still suffers from the drawbacks of the previous optimisation methods, in assuming that the pulse amplitudes remain fixed during the search procedure used at each stage to define the optimum position of a new pulse. Method MS5 lifts this constraint, by jointly optimising the amplitudes of all the existing pulses, during the search procedure which defines the optimum position of each new pulse.

Computationally efficient implementations of these two methods have been presented, based on the Cholesky matrix factorisation algorithm [4.52,4.53]. A different approach will be followed here, which derives simplified solutions for both algorithms, using the Gram-Schmidt orthogonalization process. It will be shown that method MS5 is equivalent to a MPE optimisation method that decomposes the speech signal into a set of orthogonal components corresponding to the excitation pulses [4.13,4.33].

The Gram-Schmidt procedure has been applied to linear data fitting problems and works by constructing an orthonormal set of basis vectors, which span the same subspace as a given set of input vectors. In the case of a MPE coder, the input vectors are derived from the impulse response of the Modified Synthesis Filter, and form the columns of the nxq convolution matrix $\mathbf{A}_w[q]$. This matrix can be factorised as the product of two matrices :

$$\mathbf{A}_w[q] = \mathbf{U}\,\mathbf{V} \qquad (Eq\ 4.6.40)$$

The orthonormal set of basis vectors is formed from the columns of the nxq matrix $\mathbf{U}$, and $\mathbf{V}$ is a qxq upper triangular matrix. This factorisation is always possible, as long as $\mathbf{A}_w[q]$ has a full rank, and it can be done in various ways (Householder transformations, Modified Gram-Schmidt, etc.). The

Gram-Schmidt algorithm will be used here because it results a simple MPE optimisation algorithm.

Once the factorisation is done, the normal equations (Eq 4.4.2) are transformed to the equivalent set of equations :

$$\mathbf{V}^T\mathbf{U}^T\mathbf{U}\,\mathbf{V}\,\mathbf{b} = \mathbf{V}^T\mathbf{U}^T\,\mathbf{W}(\mathbf{s}-\mathbf{m}_y) \qquad \textit{(Eq 4.6.41)}$$

As $\mathbf{U}^T\mathbf{U}=1$, Eq 4.6.41 is equivalent to :

$$\mathbf{V}\,\mathbf{b} = \mathbf{U}^T\,\mathbf{W}(\mathbf{s}-\mathbf{m}_y) = \mathbf{U}^T\,\mathbf{y}_o \qquad \textit{(Eq 4.6.41a)}$$

If the right hand side of Eq 4.6.41a, which is composed of the coordinates of the input speech vector $\mathbf{y}_o$ with respect to the orthonormal set of basis vectors, is computed, then the optimum pulse amplitudes can be found in a simple way by using backward substitution. It will later become clear that the backward substitution need only be performed once, when all the pulse positions have been optimised.

The algorithms used to implement methods MS4 and MS5 are quite similar, and will be developed by considering only the first three stages of the multi-stage MPE optimisation process. The general case of the two algorithms (for any number of stages) is presented in Fig 4.6.5.

(Stage 1)

The first basis vector $\mathbf{u}_1$ should be in the same direction as one of the $\mathbf{f}[i]$ vectors and can therefore be considered as a function of the position of the first pulse :

$$\mathbf{u}_1[i] = \frac{\mathbf{f}[i]}{\|\mathbf{f}[i]\|} \quad , \quad o \le i \le n-1 \qquad \textit{(Eq 4.6.42)}$$

The minimum-error vector at position i is orthogonal to $\mathbf{f}[i]$, and is equal to :

$$\mathbf{e}_{w,min}[i] = \mathbf{y}_o - \left(\mathbf{y}_o^T\,\mathbf{u}_1[i]\right)\mathbf{u}_1[i] \quad , \quad o \le i \le n-1 \qquad \textit{(Eq 4.6.43)}$$

The energy of the minimum-error component at position i is :

$$\left\|\mathbf{e}_{w,min}[i]\right\|^2 = \mathbf{y}_o^T\mathbf{y}_o - \frac{c_o(i)^2}{\Phi(i,i)} \quad , \quad o \le i \le n-1 \qquad \textit{(Eq 4.6.44)}$$

where :

$$c_o(i) = \mathbf{y}_o^T \, \mathbf{f}[i] \quad , \quad o \leq i \leq n-1 \qquad (Eq\ 4.6.45)$$

is the initial cross-correlation sequence and $\Phi(i,j)$ is the auto-covariance defined in Eq 4.4.11. The optimum position of the first pulse is the one that minimises the error energy, or equivalently :

$$p_1 = \max_{o \leq i \leq n-1}^{-1} \left[\frac{c_o(i)^2}{\Phi(i,i)} \right] \qquad (Eq\ 4.6.46)$$

A few additional variables can now be introduced that will be used in the next stages :

$$\mathbf{u}_1 = \mathbf{u}_1[p_1] \qquad (Eq\ 4.6.47)$$

$$v_o(i) = \Phi(i;i) \quad , \quad o \leq i \leq n-1 \qquad (Eq\ 4.6.48)$$

$$w_1 = \sqrt{v_o(p_1)} \qquad (Eq\ 4.6.49)$$

$$t_1 = \mathbf{y}_o^T \, \mathbf{u}_1 = \frac{c_o(p_1)}{w_1} \qquad (Eq\ 4.6.50)$$

Combining Eqs 4.6.42, 4.6.48 and 4.6.49, the relationship :

$$\mathbf{f}[p_1] = w_1 \, \mathbf{u}_1 \qquad (Eq\ 4.6.51)$$

is established, which is the first equation of the matrix factorisation, and includes the first basis vector $\mathbf{u}_1$ and the upper left element w_1 of matrix $\mathbf{V}$ The value t_1 is the first coordinate of the signal vector $\mathbf{y}_o$ with respect to the first coordinate axis, and the system of equations in Eq 4.6.41a for the first pulse amplitude b_1 becomes :

$$w_1 \, b_1 = t_1 \qquad (Eq\ 4.6.51a)$$

(Stage 2)

Following the Gram-Schmidt procedure, the second basis vector of unit magnitude is defined as a function of the position of the second pulse :

$$\mathbf{u}_2[i] = \frac{\mathbf{f}[i] - \left(\mathbf{u}_1^T \, \mathbf{f}[i]\right) \mathbf{u}_1}{\left\| \mathbf{f}[i] - \left(\mathbf{u}_1^T \, \mathbf{f}[i]\right) \mathbf{u}_1 \right\|} = \frac{\mathbf{f}[i] - \left(\mathbf{u}_1^T \, \mathbf{f}[i]\right) \mathbf{u}_1}{\sqrt{\Phi(i,i) - \left(\mathbf{u}_1^T \, \mathbf{f}[i]\right)^2}} \quad , \quad i \neq p_1$$

(Eq 4.6.52)

Two new sequences can be defined in order to simplify the equations :

$$m_1(i) = \mathbf{u}_1^T \mathbf{f}[i] = \frac{\Phi(p_1, i)}{w_1} \quad , \quad 0 \le i \le n-1 \qquad (Eq\ 4.6.53)$$

and

$$v_1(i) = \Phi(i,i) - \left(\mathbf{u}_1^T \mathbf{f}[i]\right)^2 = v_0(i) - m_1(i)^2 \quad , \quad 0 \le i \le n-1 \qquad (Eq\ 4.6.54)$$

The second basis vector can then be expressed as :

$$\mathbf{u}_2[i] = \frac{\mathbf{f}[i] - m_1(i)\ \mathbf{u}_1}{\sqrt{v_1(i)}} \quad , \quad i \neq p_1 \qquad (Eq\ 4.6.55)$$

Method MS4 forms the error assuming the amplitude of the first pulse to be fixed to the value found in the first stage. This value does not have to be calculated explicitly because the error after the first stage can be obtained from Eq 4.6.43. The new error vector is a function of the position and amplitude of the second pulse :

$$\mathbf{e}_w[i] = \mathbf{y}_0 - \left(\mathbf{y}_0^T \mathbf{u}_1\right) \mathbf{u}_1 - b_2(i)\ \mathbf{f}[i] \quad , \quad 0 \le i \le n-1 \qquad (Eq\ 4.6.56)$$

The energy of the new error can be minimised with respect to the amplitude of the second pulse, if the new error is made orthogonal to the error vector obtained from the first stage. The minimum error energy will then be :

$$\left[\ \mathbf{e}_w[i]^T\ \mathbf{e}_w[i]\ \right]_{min} = \mathbf{y}_0^T \mathbf{y}_0 - \frac{c_1(i)^2}{\Phi(i,i)} \quad , \quad 0 \le i \le n-1 \qquad (Eq\ 4.6.57)$$

where :

$$c_1(i) = c_0(i) - c_0(p_1)\ \frac{\Phi(p_1, i)}{\Phi(p_1, p_1)} = c_0(i) - t_1\ m_1(i) \quad , \quad 0 \le i \le n-1 \qquad (Eq\ 4.6.58)$$

The optimum position of the second pulse for method MS4 is therefore :

$$p_2 = \max_{0 \le i \le n-1}^{-1} \left[\frac{c_1(i)^2}{\Phi(i,i)} \right] \qquad (Eq\ 4.6.59)$$

As in methods MS1-MS3, the second pulse cannot be placed at the same location as the first pulse, because the value of the new cross-correlation sequence $\{c_1(i)\}$ is zero at p_1. After the optimisation of the position of the second pulse, the amplitudes of both pulses are reoptimised. How this is done will be explained at the end of step 2. Note that up to now, the pulse

amplitudes have not been explicitly calculated.

Method MS5 minimises the error with respect to the amplitudes of both pulses, while optimising the position of the second pulse. The new error vector is orthogonal to the subspace defined by the two input vectors $f[p_1]$ and $f[i]$ (or u_1 and $u_2[i]$), and can be expressed as a function of the position of the second pulse :

$$e_{w,min}[i] = y_o - \left(y_o^T u_1\right) u_1 - \left(y_o^T u_2[i]\right) u_2[i] \quad , \quad i \neq p_1 \qquad (Eq\ 4.6.60)$$

The minimum energy of the new error signal is :

$$\left\|e_{w,min}[i]\right\|^2 = y_o^T y_o - \left(y_o^T u_1\right)^2 - \left(y_o^T u_2[i]\right)^2 = y_o^T y_o - t_1^2 - \frac{c_1(i)^2}{v_1(i)} \quad , \quad i \neq p_1$$

(Eq 4.6.61)

The optimum position of the second pulse, according to method MS5, is :

$$p_2 = \max_{i \neq p_1}^{-1} \left[\frac{c_1(i)^2}{v_1(i)}\right] \qquad (Eq\ 4.6.62)$$

The similarity of Eqs 4.6.59 and 4.6.62 indicates that the two optimisation methods MS4 and MS5, are quite close in complexity. Method MS5 performs better when it can take advantage of the optimality of the amplitude estimation process that it uses, and as it will be shown, that happens when the number of pulses in each frame is increased and their interaction becomes substantial.

As in stage 1, a few additional variables to be used in the next stages are defined :

$$u_2 = u_2[p_2] \qquad (Eq\ 4.6.63)$$

$$w_2 = \sqrt{v_1(p_2)} \qquad (Eq\ 4.6.64)$$

$$t_2 = y_o^T u_2 = \frac{c_1(p_2)}{w_2} \qquad (Eq\ 4.6.65)$$

From Eqs 4.6.52-4.6.54 and 4.6.64, the second relationship of the matrix factorisation algorithm is derived :

$$f[p_2] = m_1(p_2)\, u_1 + w_2\, u_2 \qquad (Eq\ 4.6.66)$$

If at this stage the pulse amplitudes are needed, their optimum values can

be calculated by solving the system of equations :

$$\begin{bmatrix} w_1 & m_1(p_2) \\ o & w_2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \qquad (Eq\ 4.6.67)$$

Backward substitution can be used to derive first the value of b_2 and then the value of b_1.

(Stage 3)

The basis vector attached to the third pulse is :

$$\mathbf{u}_2[i] = \frac{\mathbf{f}[i] - m_1(i)\ \mathbf{u}_1 - m_2(i)\ \mathbf{u}_2}{\sqrt{v_2(i)}} \quad , \quad i \neq p_1, p_2 \qquad (Eq\ 4.6.68)$$

where :

$$m_2(i) = \mathbf{u}_2^T\ \mathbf{f}[i] = \frac{\Phi(p_2, i) - m_1(p_2)\ m_1(i)}{w_2} \quad , \quad o \leq i \leq n-1 \qquad (Eq\ 4.6.69)$$

and

$$v_2(i) = \Phi(i,i) - m_1(i)^2 - m_2(i)^2 = v_1(i) - m_2(i)^2 \quad , \quad o \leq i \leq n-1 \qquad (Eq\ 4.6.70)$$

Method MS4 assumes that the pulse amplitudes are reoptimised after the second stage, and defines the error as a function of the position and amplitude of the third pulse :

$$\mathbf{e}_w[i] = \mathbf{y}_o - \left(\mathbf{y}_o^T\ \mathbf{u}_1\right) \mathbf{u}_1 - \left(\mathbf{y}_o^T\ \mathbf{u}_2\right) \mathbf{u}_2 - b_3(i)\ \mathbf{f}[i] \quad , \quad o \leq i \leq n-1 \qquad (Eq\ 4.6.71)$$

The new error vector must be orthogonal to the error signal left over from the second stage. The minimum error energy therefore occurs at position :

$$p_3 = \max_{o \leq i \leq n-1}^{-1} \left[\frac{c_2(i)^2}{\Phi(i,i)} \right] \qquad (Eq\ 4.6.72)$$

where :

$$c_2(i) = c_1(i) - t_2\ m_2(i) \quad , \quad o \leq i \leq n-1 \qquad (Eq\ 4.6.73)$$

It is interesting to note that the values of the new cross-correlation sequence $\{c_2(i)\}$ are now zero at the locations of both previous pulses :

$$c_2(p_i) = o \quad , \quad i = 1, 2 \qquad (Eq\ 4.6.74)$$

This sets this method apart from the previous three optimisation methods

examined, where the cross-correlation was not guaranteed to have a zero value at each pulse location and pulse coincidences could occur. The condition of Eq 4.6.74 makes certain that the number of optimisation stages is the same as the number of excitation pulses defined in each frame.

Method MS5 similarly defines the optimum position of the third pulse as:

$$p_3 = \max_{i \neq p_1, p_2}^{-1} \left[\frac{c_2(i)^2}{v_2(i)} \right] \qquad (Eq\ 4.6.75)$$

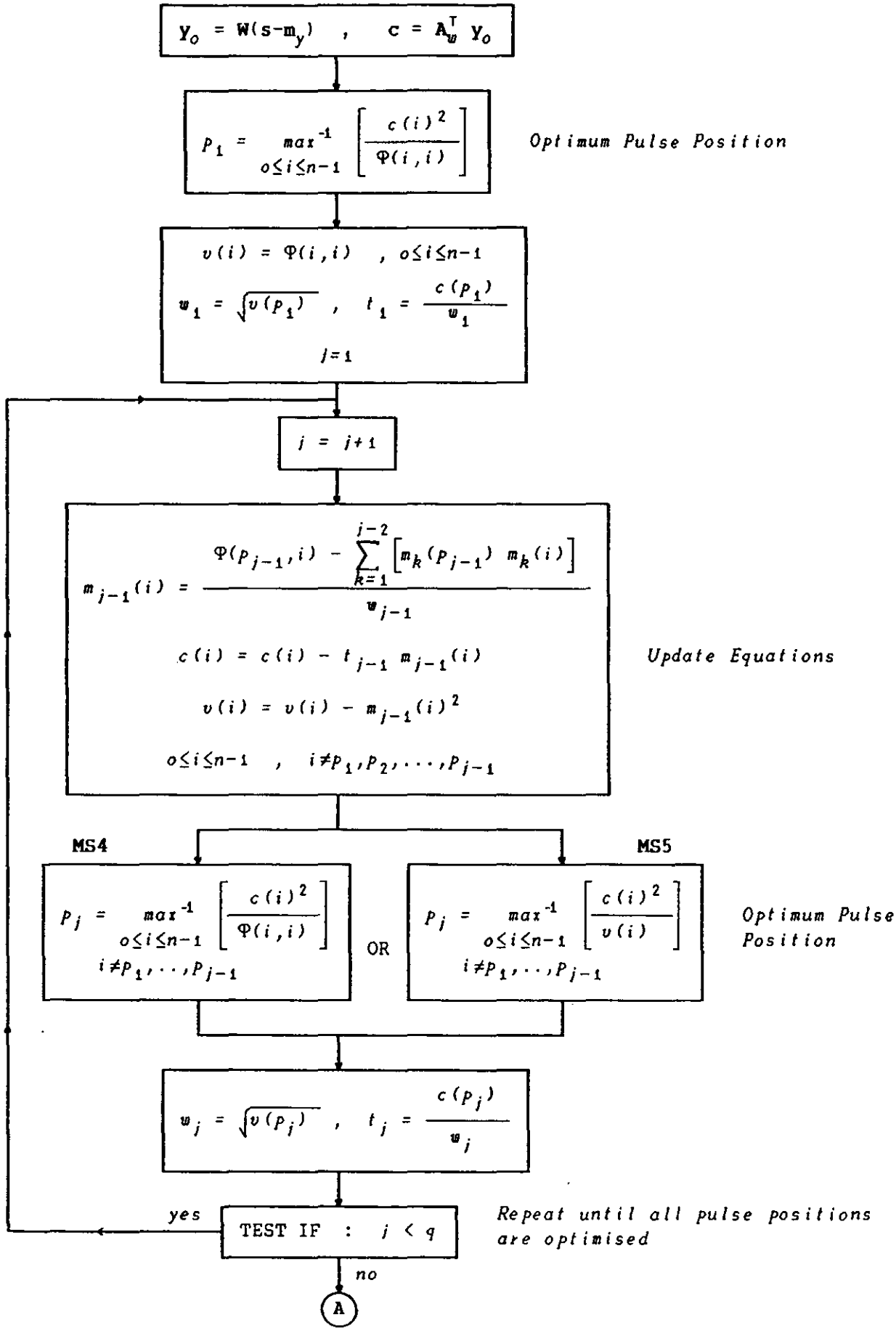
The generalisation of the steps of both algorithms is shown in Fig 4.6.5, together with the final backward substitution, required to find the pulse amplitudes. The matrix factorisation when completed, results the matrices :

$$\mathbf{A}_w[q] = \mathbf{U}\,\mathbf{V} = \left[\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q \right] \begin{bmatrix} w_1 & m_1(p_2) & m_1(p_3) & \ldots & m_1(p_q) \\ o & w_2 & m_2(p_3) & \ldots & m_2(p_q) \\ o & o & w_3 & \ldots & m_3(p_q) \\ \vdots & \vdots & \vdots & & \vdots \\ o & o & o & \ldots & w_q \end{bmatrix} \qquad (Eq\ 4.6.76)$$

The basis vectors are not required by the algorithms but the upper triangular matrix $\mathbf{V}$ can be substituted in Eq 4.6.41 to find the optimum pulse amplitudes after all the pulse positions have been determined. The right hand side of Eq 4.6.41 contains the q coordinates $t_1, t_2, \ldots, t_q$ of the signal vector $\mathbf{y}_o$, with respect to the orthonormal set of basis vectors $\mathbf{u}_1$, $\mathbf{u}_2, \ldots, \mathbf{u}_q$. The pulse amplitudes are calculated by solving the system of equations 4.6.41, using the backward substitution formula as shown in Figure 4.6.5

Method MS5 ensures that the error vector associated to each pulse, remains orthogonal to the subspace defined by all previous pulses, during the pulse position optimisation. It is therefore equivalent to the orthogonalising MPE optimisation method presented in [4.13] and [4.33], but considerably simpler.

Methods MS4 and MS5 require far fewer stages than methods MS1-3 to reach a given level of SNR in every speech frame. The top diagrams of Figs 4.6.6 and 4.6.7 show how the average number of stages varies for each method, when the required level of SNR for each frame is increased (no noise shaping is


$y_o = W(s-m_y)$, $c = A_w^T y_o$
$P_1 = \max_{o\le i\le n-1}^{-1} \left[\frac{c(i)^2}{\Phi(i,i)}\right]$
Optimum Pulse Position
$v(i) = \Phi(i,i)$, $o\le i\le n-1$
$w_1 = \sqrt{v(P_1)}$, $t_1 = \frac{c(P_1)}{w_1}$
$j=1$
$j = j+1$
$m_{j-1}(i) = \frac{\Phi(P_{j-1},i) - \sum_{k=1}^{j-2}\left[m_k(P_{j-1})\ m_k(i)\right]}{w_{j-1}}$
$c(i) = c(i) - t_{j-1}\ m_{j-1}(i)$
$v(i) = v(i) - m_{j-1}(i)^2$
$o\le i\le n-1$, $i\neq P_1, P_2, \cdots, P_{j-1}$
Update Equations
MS4
MS5
$P_j = \max_{\substack{o\le i\le n-1 \\ i\neq P_1,\cdots,P_{j-1}}}^{-1} \left[\frac{c(i)^2}{\Phi(i,i)}\right]$
OR
$P_j = \max_{\substack{o\le i\le n-1 \\ i\neq P_1,\cdots,P_{j-1}}}^{-1} \left[\frac{c(i)^2}{v(i)}\right]$
Optimum Pulse Position
$w_j = \sqrt{v(P_j)}$, $t_j = \frac{c(P_j)}{w_j}$
yes
TEST IF : $j < q$
Repeat until all pulse positions are optimised
no
A

(continued ...)

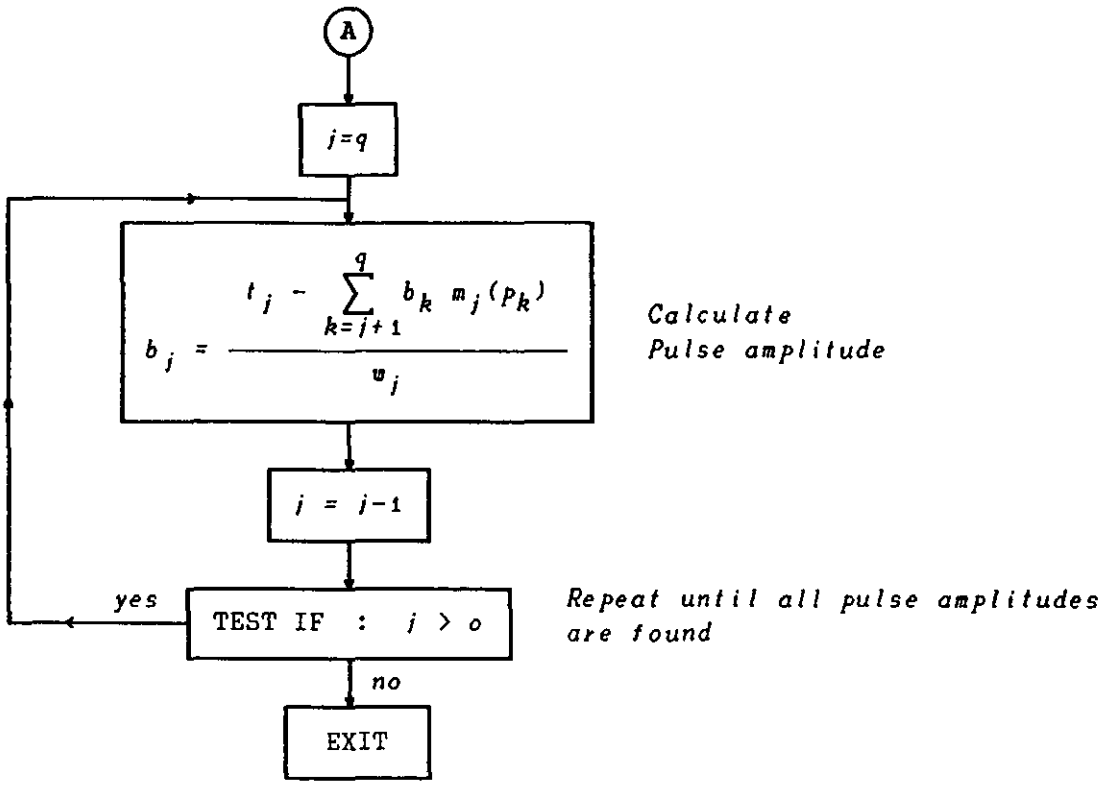


FIGURE 4.6.5 The MPE Optimisation Methods MS4 and MS5.

included). Compared to Fig 4.6.2(a) of method MS1, the relationship is now linear instead of exponential.

The diagrams (b),(c) and (d) of Figures 4.6.6 and 4.6.7, show the relative variation in the average number of stages, with reference to the values of diagram (a), when the value of the noise shaping filter constant γ is lowered (increasing the effect of the noise shaping process). It is clear that method MS4 benefits from a lower value of γ, especially at high SNRs (large number of pulses), because the interaction between the pulses is reduced when the value of γ is lowered. The same effect is not observed when method MS5 is used, because the pulse interaction is taken into account during the pulse position optimisation.

Method MS5 gives the best results over most of the SNR range, when γ is close to one. A comparison of Figures 4.6.6 and 4.6.7 reveals that when γ becomes smaller than one, the required number of stages for a given SNR value tends to increase when method MS5 is used, but the opposite effect is observed when method MS4 is used. If the required SNR level is above 10 dBs, the efficiency of method MS4 is increased when a value of γ between 0.6 and 0.9 is used. This suggests that the performance of method MS5 approaches that of MS4, when the value of γ is reduced. In order to maximise the efficiency of method MS5 and contrast its performance with that of the other optimisation methods (which use a lower value of γ to their advantage), the value of γ is kept equal to one whenever method MS5 is used.

Both MS4 and MS5 methods can be modified to take advantage of the auto-correlation approximation (Eq 4.4.17). The symmetric auto-covariance matrix $\Phi(i,j)$ can be replaced by the Toeplitz matrix $\Phi_a(i,j)$ which is easier to calculate and requires less storage than $\Phi(i,j)$. The same approximation has been used in methods MS1a-MS3a. The two modified methods will be referred to as methods MS4a and MS5a.

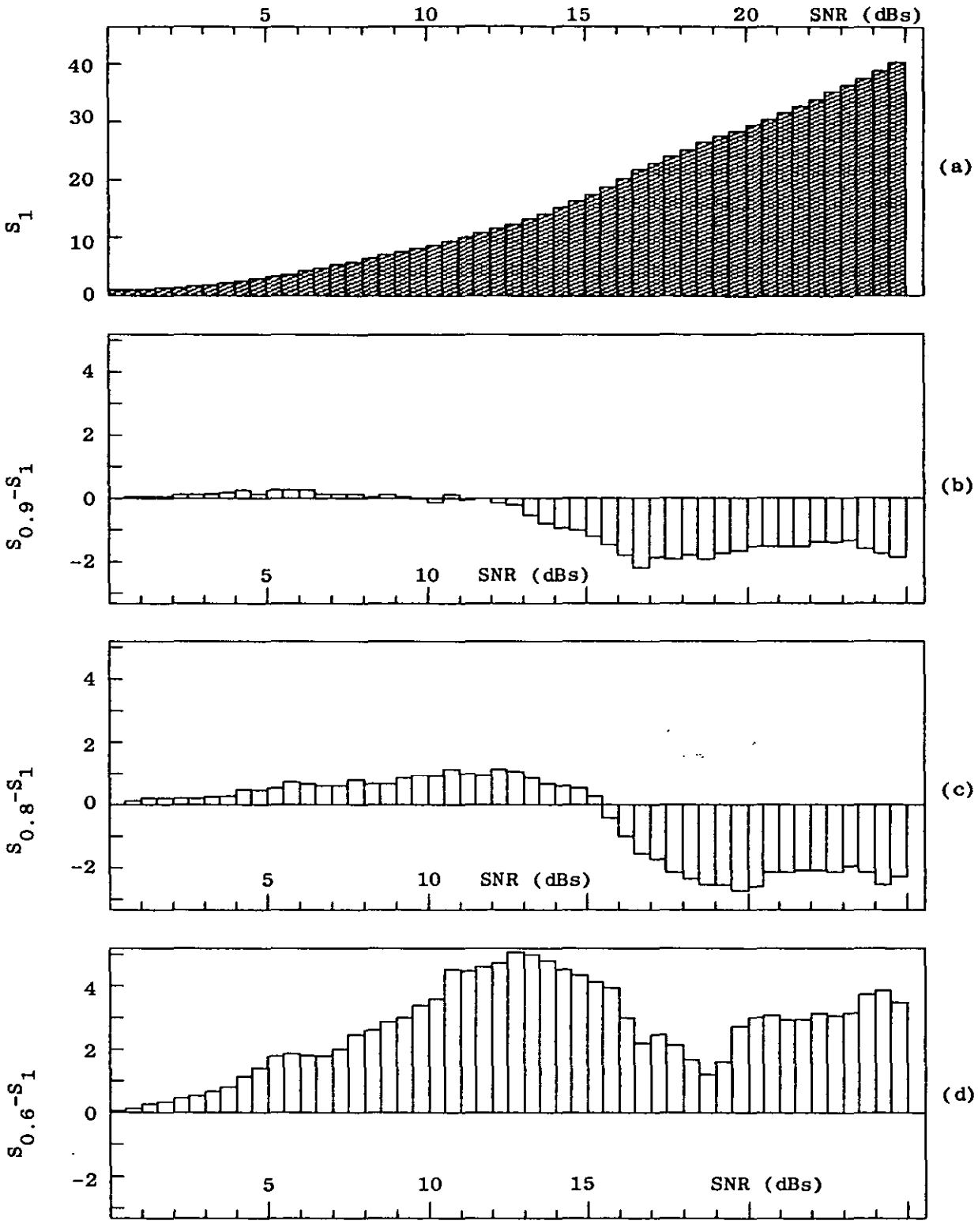


FIGURE 4.6.6 Plots of the average number of stages S_γ required by method MS4 to reach the level of SNR given on the horizontal axis. S_1 corresponds to $\gamma=1$, $S_{0.9}$ to $\gamma=0.9$, $S_{0.8}$ to $\gamma=0.8$ and $S_{0.6}$ to $\gamma=0.6$

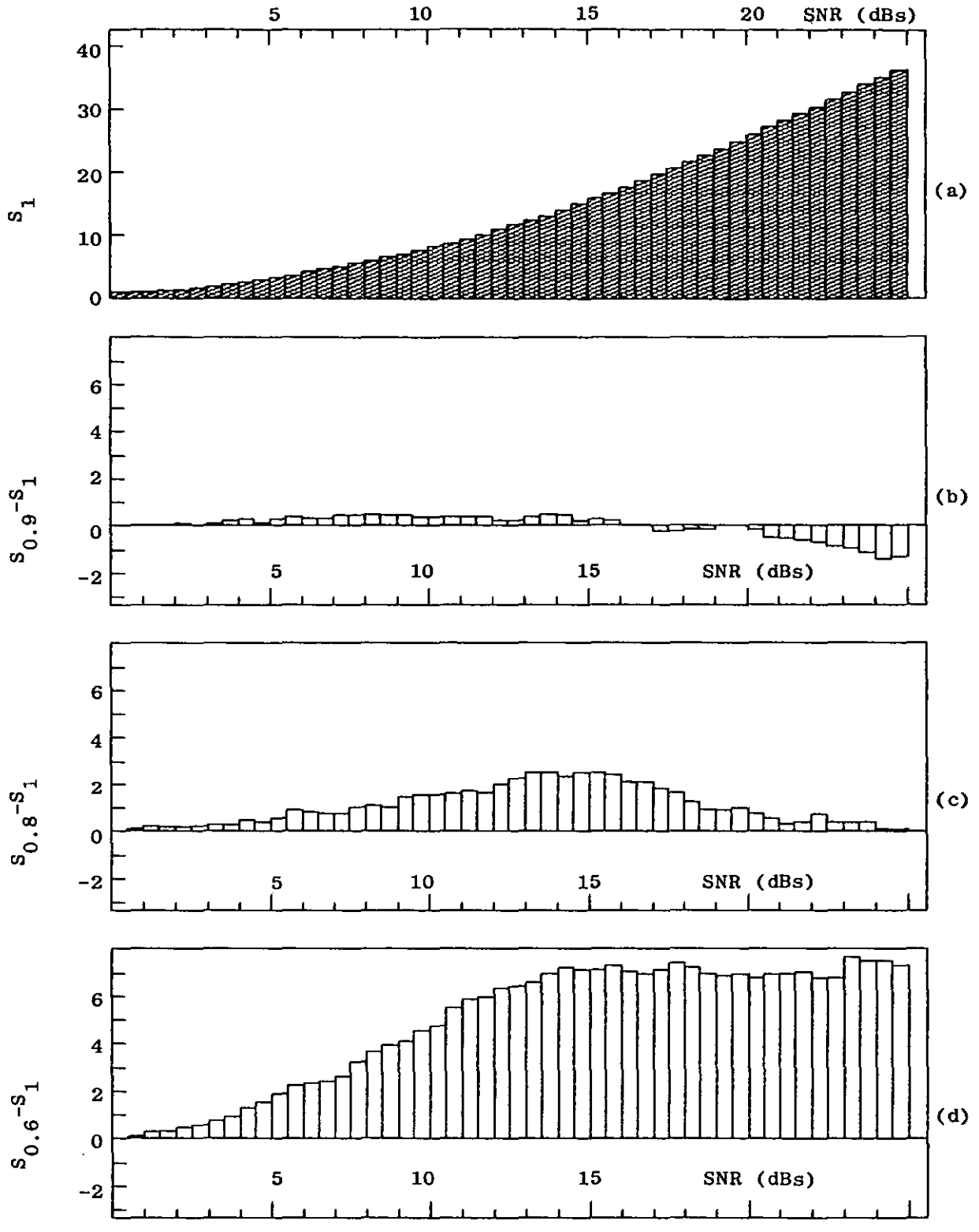


FIGURE 4.6.7 Plots of the average number of stages S_γ required by method MS5 to reach the level of SNR given on the horizontal axis. S_1 corresponds to γ=1, $S_{0.9}$ to γ=0.9, $S_{0.8}$ to γ=0.8 and $S_{0.6}$ to γ=0.6

2.7 Block Search (BS) Optimisation Algorithms

Block Search optimisation methods start with an estimate of all the q pulse positions and then iteratively improve the accuracy of this estimate. To define the positions of q pulses, the algorithm constructs a search route by finding a sequence of q-dimensional position vectors $[p_1, p_2, \ldots, p_q]$ which progressively minimise the approximation error introduced by the MPE coder. Each vector in this sequence is defined by examining a number of different pulse arrangements in every MPE frame and monitoring the corresponding changes of the approximation error. The pulse arrangement which results the minimum approximation error is selected.

The pulse arrangements are generated using simple rules and are variations of a single pattern of pulse positions (source pattern). The source pattern is periodically updated (replaced by the set of pulse positions that has produced the minimum approximation error) and forms a sequence of position vectors associated with a monotonically decreasing approximation error.

The repeated optimisation causes unpredictable changes to the pulse locations, so that the final set of pulse positions may be quite different from the initial estimate of the positions. In that aspect, the outcome of a BS optimisation method is less constrained than that obtained from a MS method, which does not permit pulses defined in the first optimisation stages to be relocated in later stages.

The approximation error is calculated by fully taking into account the pulse interaction, thus a joint amplitude reoptimisation is implicitly performed. This increases the complexity of the BS algorithms but also ensures that good optimisation results are obtained.

The pulse arrangements are generated from the source pattern by choosing a single pulse and altering its position within a set of allowed pulse locations. The size of the set determines the complexity of the BS method, since a large set requires a large number of pulse arrangements to be examined. The pulses chosen to be relocated are selected sequentially and may be chosen more than once.

The initial estimate of the pulse positions does not have to be very accurate, therefore a simple method can be used to find the pulse positions. The more complex the BS optimisation algorithm is, the less it relies on the accuracy of the initial position estimate. The performance of simple BS

schemes though can be adversely affected when the initial set of pulse positions is grossly inaccurate.

Two BS algorithms will be examined. The first one considers a large set of allowed pulse locations and is therefore the most complex. The second method limits the number of alternative pulse locations by only considering positions in the vicinity of existing pulses. As for the MS algorithms, both BS algorithms will be described assuming a general linear synthesis filter.

1) METHOD BS1

Given an initial set of excitation pulses at positions $p_1, p_2, \ldots, p_q$, the BS1 algorithm optimises the position of each pulse individually, assuming the rest of the pulses to be at fixed locations. The pulses are selected in a cyclic order so that the pulse at position p_1 is optimised first, then the pulse at p_2 and so on. The first iteration is completed when the optimum position p_q has been determined, and the next iteration starts by optimising p_1 again. The number of iterations n_r is fixed and is usually quite small.

The optimum position of each pulse is found by calculating the approximation error for every possible location within the MPE frame. The pulse is then placed at the location that resulted the minimum error, and the process continues by optimising the position of the next pulse. The approximation error is evaluated using a joint amplitude estimation process, which effectively compensates for the interaction between the pulses.

The Gram-Schmidt orthogonalisation procedure is used to solve the system of normal equations (Eq 4.4.2). The order with which the pulses are considered is important because a change in the pulse order results a different set of orthogonal axes. By carefully rearranging the order of the pulses, it is possible to avoid the complete reconstruction of the set of orthogonal axes for every pulse position optimised, thus reducing the complexity of the BS1 algorithm.

Consider the first step where the pulse at position p_1 is optimised. An orthogonal set of basis vectors is constructed from the vectors $f[p_i]$ (shifted versions of the MSF impulse response) corresponding to the fixed positions $p_2, \ldots, p_q$. The last axis corresponding to the variable position p_1 should be orthogonal to the $(q-1)$ dimensional subspace defined by the remaining pulses. To achieve that, the same recursive equations that were

used in method MS5 can be applied, with the only difference that the pulse positions $p_2,\ldots,p_q$ are now known in advance.

In order to simplify the formulation of the equations, the pulses are resuffled by interchanging the values of p_1 and p_q. The pulse whose position is optimised is now last and the recursive orthogonalisation equations are :

$$w_j = \sqrt{v_{j-1}(p_j)} \quad , \quad t_j = \frac{c_{j-1}(p_j)}{w_j} \qquad (Eq\ 4.7.1)$$

$$m_j(i) = \frac{\Phi(p_j,i) - \sum_{k=1}^{j-1} \left[m_k(p_j)\ m_k(i)\right]}{w_j} \quad , \quad i \neq p_1,p_2,\ldots,p_{q-1} \qquad (Eq\ 4.7.2)$$

$$c_j(i) = c_{j-1}(i) - t_j\ m_j(i) \quad , \quad i \neq p_1,p_2,\ldots,p_{q-1} \qquad (Eq\ 4.7.3)$$

$$v_j(i) = v_{j-1}(i) - m_j(i)^2 \quad , \quad i \neq p_1,p_2,\ldots,p_{q-1} \qquad (Eq\ 4.7.4)$$

The index j varies from 1 up to $q-1$ and the initial arrays $c_o(i)$ and $v_o(i)$ take the values :

$$\mathbf{c}_o = \mathbf{A}_w^T\ \mathbf{W}\left(\mathbf{s} - \mathbf{m}_y\right) = \mathbf{A}_w^T\ \mathbf{y}_o \qquad (Eq\ 4.7.5)$$

and :

$$v_o(i) = \Phi(i,i) \quad , \quad o \leq i \leq n-1 \qquad (Eq\ 4.7.6)$$

The approximation error can be expressed as a function of the position of the last pulse as :

$$\left\|\mathbf{e}_{w,min}[i]\right\|^2 = \mathbf{y}_o^T\ \mathbf{y}_o - \sum_{j=1}^{q-1} t_j^2 - \frac{c_{q-1}(i)^2}{v_{q-1}(i)} \quad , \quad i \neq p_1,\ldots,p_{q-1} \qquad (Eq\ 4.7.7)$$

The optimum position of the last pulse is therefore :

$$p_q = \max_{\substack{o \leq i \leq n-1 \\ i \neq p_1,\ldots,p_{q-1}}}^{-1} \left[\frac{c_{q-1}(i)^2}{v_{q-1}(i)}\right] \qquad (Eq\ 4.7.8)$$

The same process can be repeated for the second pulse, by interchanging the values of p_2 and p_q. The second pulse is now last and its position can be optimised in the same way as for the first pulse.

Notice that since p_1 remains unchanged when p_2 is optimised, the values of w_1, t_1, $m_1(i)$, $c_1(i)$ and $v_1(i)$ do not have to be recalculated and can be obtained from the previous optimisation step. In the same way, when the values of p_3 and p_q are interchanged, the values of p_1 and p_2 remain unchanged and the recursive equations 4.7.1-4.7.4 need only be applied for $j=3,\ldots,q-1$.

The flow diagram of method BS1 in Fig 4.7.1 shows the double optimisation loop for each iteration and each pulse. The subscript j has been dropped from the arrays $c_j(i)$ and $v_j(i)$ because they can be replaced by their updated values. The intermediate values of $c(i)$ and $v(i)$ that can be used to initialise the recursive updating when the next pulse position is optimised, are stored in the arrays $c_s(i)$ and $v_s(i)$. An interchange variable l, points to the position variable p_l whose value is interchanged with the value of p_q

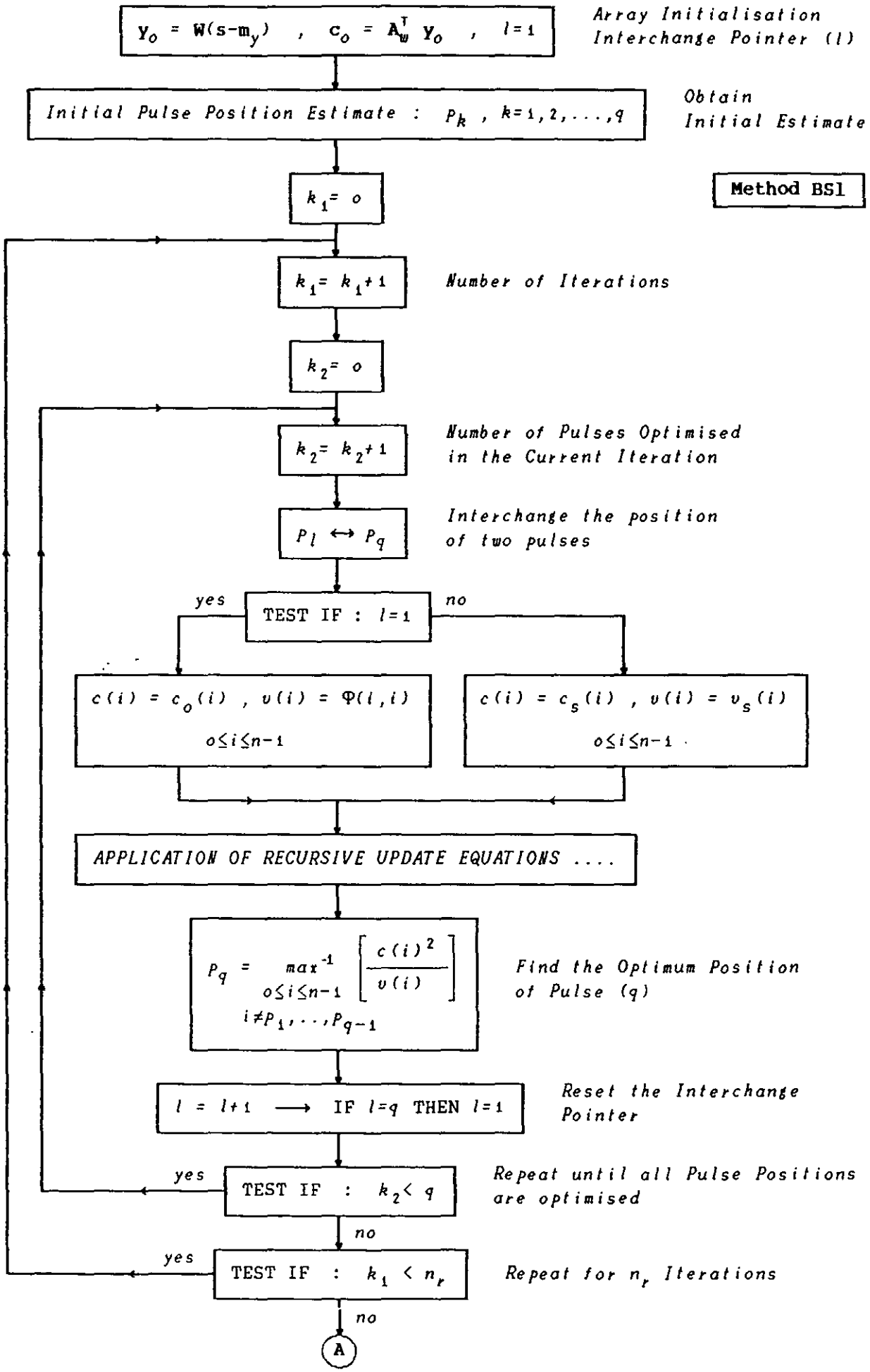
To see the effect of the pulse resuffling, a 3-pulse excitation can be arranged with initial positions p_1=10, p_2=20 and p_3=30. The order with which the pulses enter the orthogonalisation procedure in this case is :

ITERATIONS →		k_1= 1				k_1= 2					
k_2	l	P_1	P_2	P_3	$P_{3,opt}$	k_2	l	P_1	P_2	P_3	$P_{3,opt}$
1	1	30	20	10	11	1	2	21	31	11	12
2	2	30	11	20	21	2	1	12	31	21	22
3	1	21	11	30	31	3	2	12	22	31	32

It is clear from this example that all the pulses are sequentially optimised in every iteration and with the same order. The order of the first $(q-1)$ pulses changes cotinuously but this does not affect the solution for the last pulse, because the orthogonal set of basis vectors is rebuilt every time, starting from the pulse whose order has changed.

When the last iteration has finished, the elements of the upper triangular matrix factor **V** (Eq 4.6.76) and the coordinates $t_1, t_2, \ldots, t_q$ of the signal vector $\mathbf{y}_o$ are known, and can be substituted in Eq 4.6.41 to calculate the pulse amplitudes, using backward substitution as shown in Figure 4.6.5.

The complexity of the BS1 algorithm is relatively high but its performance approaches that obtained from the highly complex Successive-Elimination and


$y_o = W(s-m_y)$, $c_o = A_w^T y_o$, $l=1$
Array Initialisation
Interchange Pointer (l)
Initial Pulse Position Estimate : P_k , $k=1,2,\ldots,q$
Obtain
Initial Estimate
Method BSl
$k_1 = o$
$k_1 = k_1 + 1$
Number of Iterations
$k_2 = o$
$k_2 = k_2 + 1$
Number of Pulses Optimised
in the Current Iteration
$P_l \leftrightarrow P_q$
Interchange the position
of two pulses
yes
TEST IF : $l=1$
no
$c(i) = c_o(i)$, $v(i) = \Phi(l,i)$
$o \le i \le n-1$
$c(i) = c_s(i)$, $v(i) = v_s(i)$
$o \le i \le n-1$
APPLICATION OF RECURSIVE UPDATE EQUATIONS ....
$P_q = \max_{\substack{o \le i \le n-1 \\ i \ne P_1,\ldots,P_{q-1}}}^{-1} \left[\frac{c(i)^2}{v(i)} \right]$
Find the Optimum Position
of Pulse (q)
$l = l+1 \longrightarrow$ IF $l=q$ THEN $l=1$
Reset the Interchange
Pointer
yes
TEST IF : $k_2 < q$
Repeat until all Pulse Positions
are optimised
no
yes
TEST IF : $k_1 < n_r$
Repeat for n_r Iterations
no
A

(continued ...)

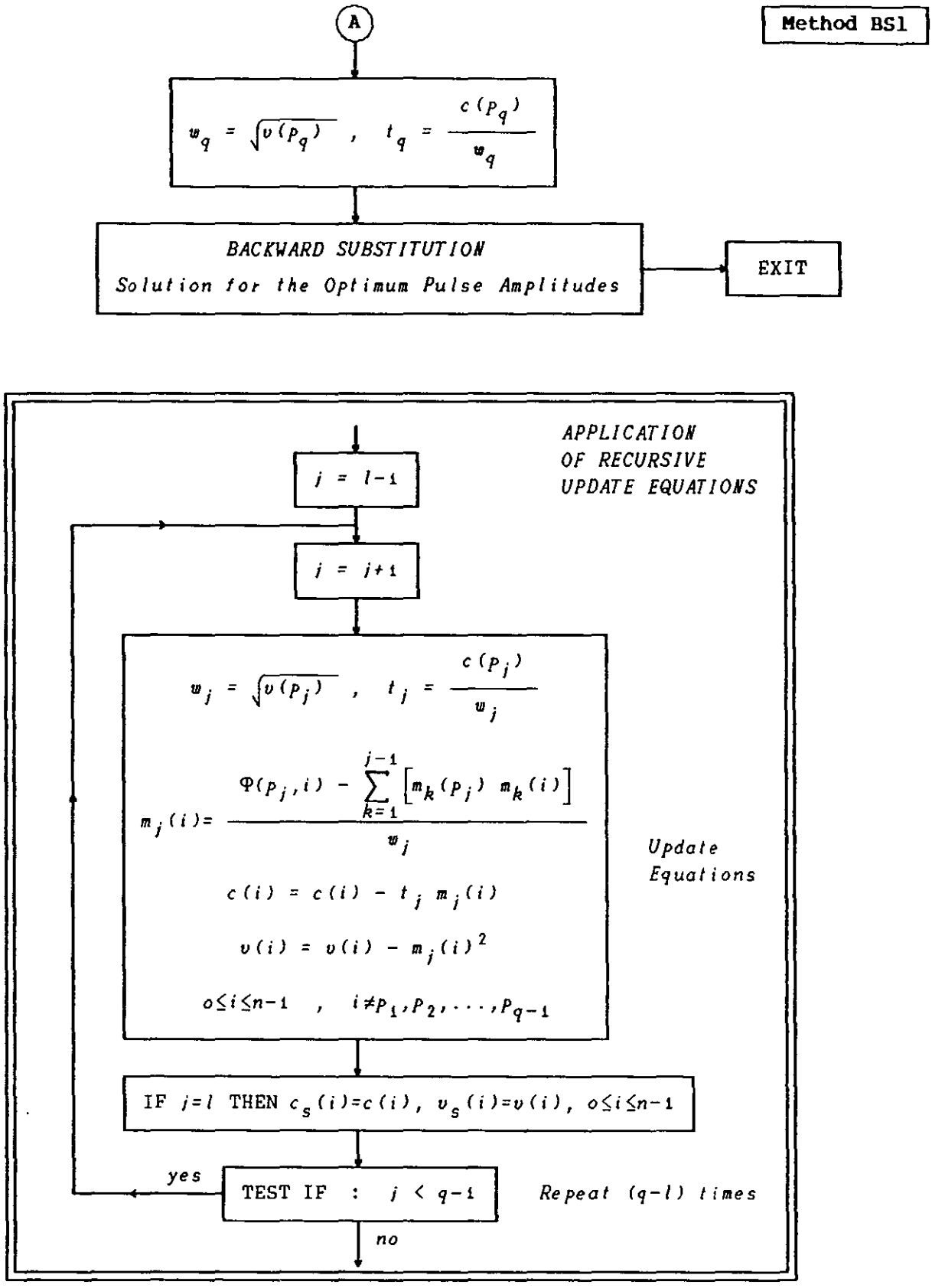


FIGURE 4.7.1 Flow diagram of the MPE Optimisation Method BS1.

Multivariate-Optimisation methods discussed in the previous chapter.

The outcome of the optimisation in method BS1 is not heavily dependent on the accuracy of the initial position estimate. This can be confirmed by comparing the SNR obtained when the initial estimate is provided by method MS1a and when the pulses are initially arranged in a regular grid formation (equally spaced) covering the entire MPE frame. The figures given below show the variation of the average Segmental-SNR as the pulse rate increases. These results were taken using a speech data training set of eight male and seven female speakers. The speech signal was low-passed to 3.4 kHz and sampled at 8 kHz. A MPE frame of 100 samples and a non-overlapping LPC frame of 200 samples were used (at 8 kHz sampling rate). The order of the AR-LPC filter, the number of iterations n_r and the noise shaping filter constant γ were set equal to *12*, *1* and *1* respectively.

Pulses/sec →	*400*	*800*	*1200*	*1600*	*2000*	*2400*
INITIAL ESTIMATE	*SNR (dBs)*					
Method MS1a	*10.2*	*15.2*	*18.8*	*21.4*	*23.7*	*26.0*
Regular Grid	*9.8*	*14.9*	*18.5*	*21.2*	*23.5*	*25.9*

The differences observed in the SNR are small, but since the complexity of method MS1a is also very small compared to that of BS1, method MS1a will be used in the future to provide the initial position estimate. The noise shaping filter constant γ will also be set equal to 1 because, as for method MS5, the performance of the algorithm in terms of the attainable SNR deteriorates as the value of γ is reduced below 1.

Method BS1a is formed when the Toeplitz matrix $\Phi_a(i,j)$ is used instead of $\Phi(i,j)$ in the BS1 algorithm (see Eq 4.4.17).

2) METHOD BS2

The performance of the BS2 algorithm is more dependent on the accuracy of the initial position estimate since it allows only a restricted pulse movement within each MPE frame. The pulses are selected in the same way as in method BS1, but the search for the optimum pulse position that minimises the approximation error is done in a small interval around the established position of each pulse.

It has been found experimentally that when method MS1a is used to provide

the initial position estimate, the pulses rarely need to be moved further than two samples away from their initial positions, in order to reach their locally optimum position. Method BS2 therefore adopts a one-dimensional steepest descent approach, which sequentially optimises each of the position variables $p_1, p_2, \ldots, p_q$. The steepest descent algorithm guides the approximation error to a local minimum by considering a limited number of pulse arrangements.

Assuming that an interval of 5 samples is sufficient in order to find the optimum position of each pulse, a simple minimisation procedure can be followed to locate the error minimum. The direction of decreasing error is first established by moving the pulse one position to the right (Fig 4.7.2). If the approximation error is reduced then one further position to the right is examined and the minimisation process ends. If the error increases then the pulse is moved one position to the left of its original location. If the error is still larger than it was in the original position then there is no need to continue the search because the local minimum occurs where the pulse was originally placed. Otherwise another position to the left is examined before the one-dimensional search stops, to be repeated again for the next pulse. The whole process is repeated until all the pulse positions have been optimised, in which case the first iteration is completed. In general, the algorithm allows for n_r iterations to be performed.

A joint amplitude estimation is implicitly performed, but the approximation error is calculated differently from method BS1, resulting a lower algorithmic complexity. The error equations will be formed assuming that the position p_q of the last pulse is being optimised. The normal equations are :

$$\mathbf{D}\,\mathbf{b} = \mathbf{A}_w[q]^T\,\mathbf{y} \qquad \textit{(Eq 4.7.9)}$$

where :

$$\mathbf{D} = \mathbf{A}_w[q]^T\,\mathbf{A}_w[q] \qquad \textit{(Eq 4.7.10)}$$

and :

$$\mathbf{y} = \mathbf{W}\left(\mathbf{s} - \mathbf{m}_y\right) \qquad \textit{(Eq 4.7.11)}$$

These equations can be rewritten in a composite matrix form to isolate the contribution of the last pulse at position i :

$$\left[\begin{array}{c|c} \mathbf{A}_w[q-1]^T\,\mathbf{A}_w[q-1] & \mathbf{A}_w[q-1]^T\,\mathbf{f}[i] \\ \hline \mathbf{f}[i]^T\,\mathbf{A}_w[q-1] & \mathbf{f}[i]^T\,\mathbf{f}[i] \end{array}\right] \mathbf{b} = \left[\begin{array}{c} \mathbf{A}_w[q-1]^T\,\mathbf{y} \\ \hline \mathbf{f}[i]^T\,\mathbf{y} \end{array}\right] \qquad (Eq\ 4.7.12)$$

where $\mathbf{A}_w[q-1]$ is the $nx(q-1)$ convolution matrix :

$$\mathbf{A}_w[q-1] = \left[\, \mathbf{f}[p_1], \mathbf{f}[p_2], \ldots, \mathbf{f}[p_{q-1}] \,\right] \qquad (Eq\ 4.7.13)$$

Eq 4.7.12 can be rewritten in a more compact form as :

$$\left[\begin{array}{c|c} \mathbf{X} & \mathbf{x}[i] \\ \hline \mathbf{x}[i]^T & \Phi(i,i) \end{array}\right] \mathbf{b} = \left[\begin{array}{c} \mathbf{d} \\ \hline c(i) \end{array}\right] \qquad (Eq\ 4.7.14)$$

where $c(i)$ is the ith element of the vector :

$$\mathbf{c} = \mathbf{A}_w\,\mathbf{y} \qquad (Eq\ 4.7.15)$$

If the iverse of the $(q-1)x(q-1)$ minor matrix $\mathbf{X}$ is known, then the inverse of matrix $\mathbf{D}$ can be efficiently calculated. Assuming that $\mathbf{D}$ is strongly non-singular [4.54,4.55], its inverse can be expressed as :

$$\mathbf{D}^{-1} = \left[\begin{array}{c|c} \mathbf{X}^{-1} + \dfrac{\mathbf{X}^{-1}\,\mathbf{x}[i]\,\mathbf{x}[i]^T\,\mathbf{X}^{-1}}{\Phi(i,i) - \mathbf{x}[i]^T\,\mathbf{X}^{-1}\,\mathbf{x}[i]} & \dfrac{-\,\mathbf{X}^{-1}\,\mathbf{x}[i]}{\Phi(i,i) - \mathbf{x}[i]^T\,\mathbf{X}^{-1}\,\mathbf{x}[i]} \\ \hline \dfrac{-\,\mathbf{x}[i]^T\,\mathbf{X}^{-1}}{\Phi(i,i) - \mathbf{x}[i]^T\,\mathbf{X}^{-1}\,\mathbf{x}[i]} & \dfrac{1}{\Phi(i,i) - \mathbf{x}[i]^T\,\mathbf{X}^{-1}\,\mathbf{x}[i]} \end{array}\right] \qquad (Eq\ 4.7.16)$$

or in a more compact form :

$$\mathbf{D}^{-1} = \left[\begin{array}{c|c} \mathbf{U}[i] & \mathbf{u}[i] \\ \hline \mathbf{u}[i]^T & u(i) \end{array}\right] \qquad (Eq\ 4.7.17)$$

By combining Eqs 4.7.14 and 4.7.16, the solution of the system of normal equations is found to be :

$$\mathbf{b} = \left[\begin{array}{c} \mathbf{X}^{-1}\,\mathbf{d} \\ \hline o \end{array}\right] - \frac{\mathbf{x}[i]^T\,\mathbf{X}^{-1}\,\mathbf{d} - c(i)}{\Phi(i,i) - \mathbf{x}[i]^T\,\mathbf{X}^{-1}\,\mathbf{x}[i]} \left[\begin{array}{c} \mathbf{X}^{-1}\,\mathbf{x}[i] \\ \hline -1 \end{array}\right] \qquad (Eq\ 4.7.18)$$

or equivalently :

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_s \\ \hline o \end{bmatrix} + \left(\mathbf{x}[i]^T \, \mathbf{X}^{-1} \, \mathbf{d} - c(i) \right) \begin{bmatrix} \mathbf{u}[i] \\ \hline u(i) \end{bmatrix} \qquad (Eq\ 4.7.19)$$

where :

$$\mathbf{b}_s = \mathbf{X}^{-1} \, \mathbf{d} \qquad (Eq\ 4.7.19a)$$

is the solution of the minor system of normal equations that only includes the first $(q-1)$ pulses. The approximation error can be expressed as a function of the position of the last pulse as :

$$\varepsilon(i) = \left\| \mathbf{e}_{w,min}[i] \right\|^2 = \mathbf{y}_o^T \, \mathbf{y}_o - \mathbf{b}^T \begin{bmatrix} \mathbf{d} \\ \hline c(i) \end{bmatrix} \qquad (Eq\ 4.7.20)$$

By substituting Eq 4.7.18 into Eq 4.7.20, the error expression becomes :

$$\varepsilon(i) = \left\| \mathbf{e}_{w,min}[i] \right\|^2 = \mathbf{y}_o^T \, \mathbf{y}_o - \mathbf{b}_s^T \, \mathbf{d} - \frac{\left[\mathbf{x}[i]^T \, \mathbf{X}^{-1} \, \mathbf{d} - c(i) \right]^2}{\Phi(i,i) - \mathbf{x}[i]^T \, \mathbf{X}^{-1} \, \mathbf{x}[i]} \qquad (Eq\ 4.7.20a)$$

or equivalently (see vector definitions in Eq 4.7.17) :

$$\varepsilon(i) = \mathbf{y}_o^T \, \mathbf{y}_o - \mathbf{b}_s^T \, \mathbf{d} - \frac{\left[\mathbf{u}[i]^T \, \mathbf{d} + u(i) \, c(i) \right]^2}{u(i)} \qquad (Eq\ 4.7.21)$$

The locally optimum position of the last pulse is therefore :

$$p_q = \underset{(local)}{max^{-1}} \left[\frac{\left[\mathbf{x}[i]^T \, \mathbf{X}^{-1} \, \mathbf{d} - c(i) \right]^2}{\Phi(i,i) - \mathbf{x}[i]^T \, \mathbf{X}^{-1} \, \mathbf{x}[i]} \right] \qquad (Eq\ 4.7.22)$$

Eq 4.7.22 indicates that if the inverse of the minor matrix $\mathbf{X}$ is known, the approximation error can be calculated for each new pulse position, with $O(n^2+2n)$ multiplications/additions. The matrix $\mathbf{X}$ remains fixed while the positions of the first $(q-1)$ pulses are not changed, thus its inverse need only be calculated once during the search for the optimum position of each pulse.

The inverse $\mathbf{X}^{-1}$ is initially calculated using the Cholesky factorisation method, and is updated each time a different pulse is chosen to be

optimised. This iterative updating operation is performed in three steps, as shown in Fig 4.7.2. The complete inverse matrix $\mathbf{D}_0^{-1}$ is first calculated using Eq 4.7.16. Then a rearrangement of two of its columns and rows is made, to reflect the change in the order of the two pulses at positions p_l and p_q. This in effect interchanges the variables corresponding to the amplitudes of the two pulses. In matrix form, this variable interchange is performed by premultiplying and postmultiplying the original inverse $\mathbf{D}_0^{-1}$ (corresponding to the previous values of the variables p_l and p_q), with a qxq permutation matrix $\mathbf{P}_{l,q}$, which rearranges the order of the two variables l and q, thus reflecting the new values of p_l and p_q :

$$\mathbf{D}^{-1} = \mathbf{P}_{l,q}\ \mathbf{D}_0^{-1}\ \mathbf{P}_{l,q} \qquad (Eq\ 4.7.23)$$

In the third step, the inverse of the new minor matrix $\mathbf{X}$ is updated using the formula :

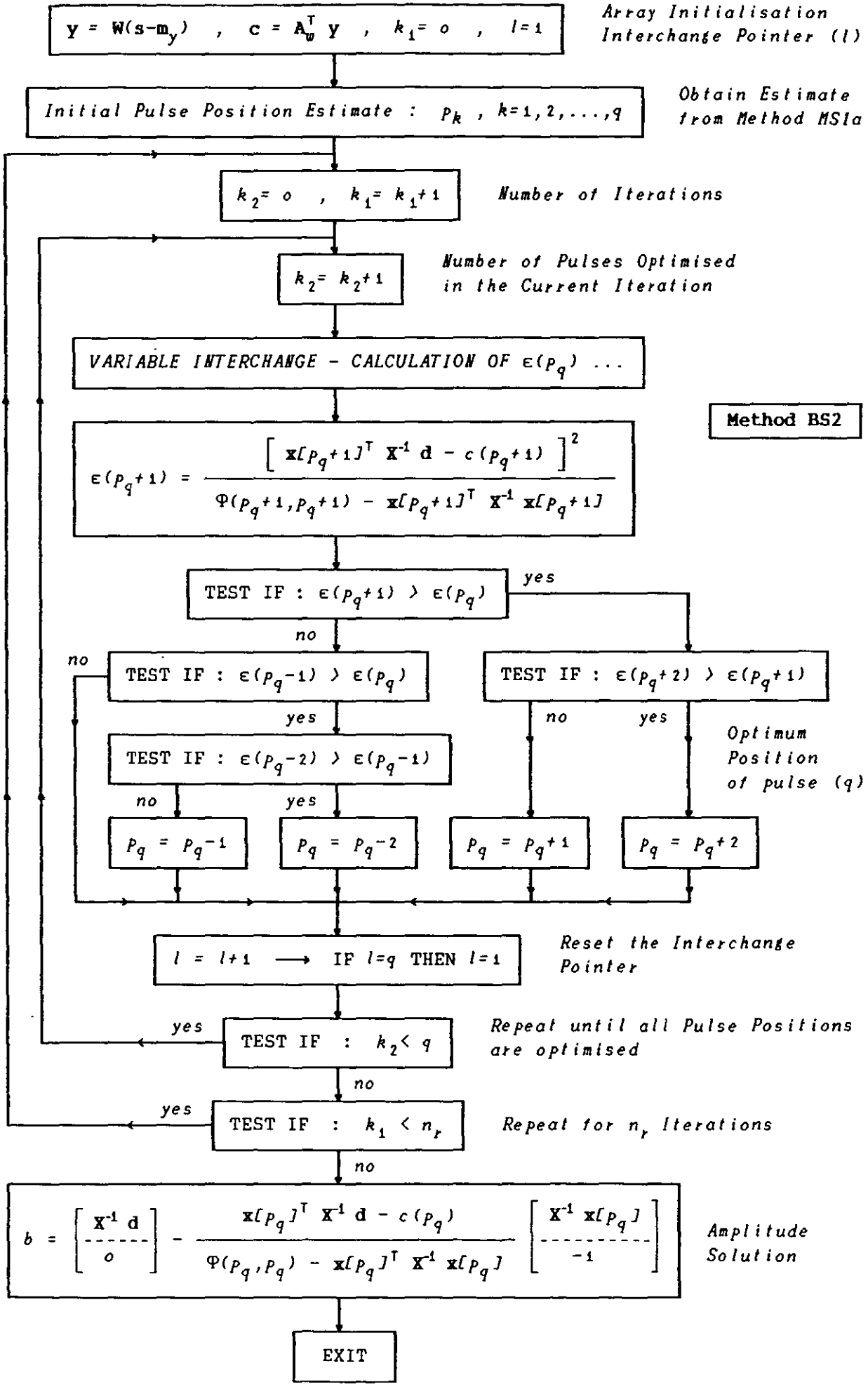
$$\mathbf{X}^{-1} = \mathbf{U}[p_q] - \frac{\mathbf{u}[p_q]\ \mathbf{u}[p_q]^T}{u(p_q)} \qquad (Eq\ 4.7.24)$$

The first iteration is completed when all the pulses have been considered. After n_r complete iterations, the pulse amplitudes are calculated using Eq 4.7.18.

The assumption of strong nonsingularity of the matrix D has not created any problems in practice, even though single precision arithmetic was used in the simulation of the algorithm.

A simplified version of the BS2 algorithm has been proposed [4.56], which limits the number of pulses chosen to be optimised, based on a pulse energy criterion. The results obtained from this method are dependent on the number of pulses optimised in each MPE frame.

Method BS2a is formed if the autocovariance matrix $\Phi_a(i,j)$ is used instead of $\Phi(i,j)$ in the BS2 algorithm.


$y = W(s-m_y)$, $c = A_w^T y$, $k_1 = o$, $l = 1$
Array Initialisation
Interchange Pointer (l)
Initial Pulse Position Estimate : p_k , $k = 1, 2, \ldots, q$
Obtain Estimate from Method MSIa
$k_2 = o$, $k_1 = k_1 + 1$
Number of Iterations
$k_2 = k_2 + 1$
Number of Pulses Optimised in the Current Iteration
VARIABLE INTERCHANGE - CALCULATION OF $\epsilon(p_q)$ ...
Method BS2
$\epsilon(p_q+1) = \dfrac{\left[x[p_q+1]^T X^{-1} d - c(p_q+1) \right]^2}{\Phi(p_q+1, p_q+1) - x[p_q+1]^T X^{-1} x[p_q+1]}$
TEST IF : $\epsilon(p_q+1) > \epsilon(p_q)$
yes
no
TEST IF : $\epsilon(p_q-1) > \epsilon(p_q)$
no
yes
TEST IF : $\epsilon(p_q-2) > \epsilon(p_q-1)$
TEST IF : $\epsilon(p_q+2) > \epsilon(p_q+1)$
no
yes
no
yes
Optimum Position of pulse (q)
$p_q = p_q - 1$
$p_q = p_q - 2$
$p_q = p_q + 1$
$p_q = p_q + 2$
$l = l + 1 \longrightarrow$ IF $l = q$ THEN $l = 1$
Reset the Interchange Pointer
yes
TEST IF : $k_2 < q$
Repeat until all Pulse Positions are optimised
no
yes
TEST IF : $k_1 < n_r$
Repeat for n_r Iterations
no
$b = \begin{bmatrix} X^{-1} d \\ \hline o \end{bmatrix} - \dfrac{x[p_q]^T X^{-1} d - c(p_q)}{\Phi(p_q, p_q) - x[p_q]^T X^{-1} x[p_q]} \begin{bmatrix} X^{-1} x[p_q] \\ \hline -1 \end{bmatrix}$
Amplitude Solution
EXIT

(continued ...)

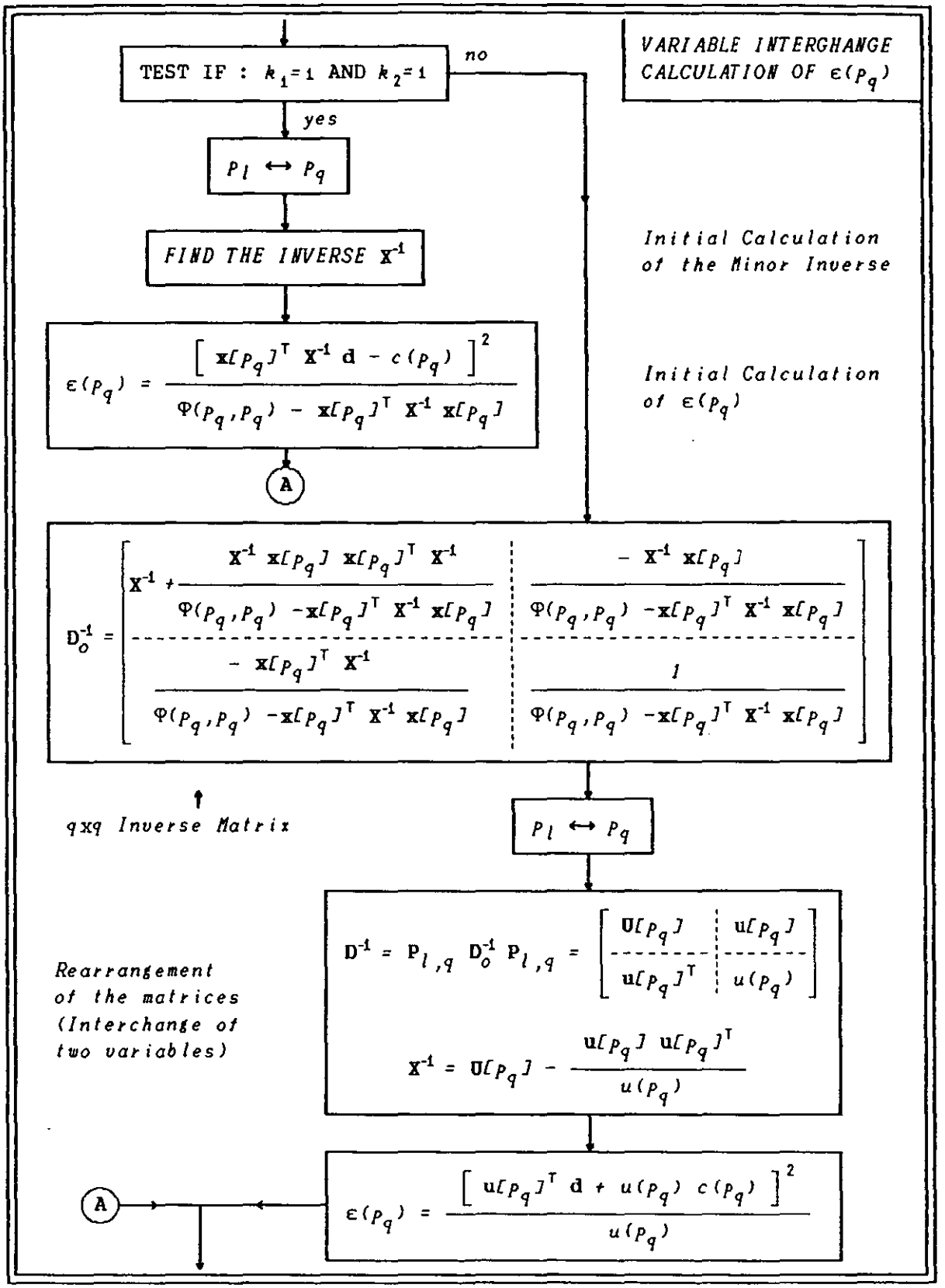


FIGURE 4.7.2 Flow diagram of the MPE Optimisation Method BS2.

4.8 Comparison of the MPE Optimisation Methods

The five MS and two BS optimisation methods are compared in Figures 4.8.1, 4.8.2 and 4.8.3, in terms of their complexity (operation per sample) and SNR performance (average Segmental-SNR). The comparison indicates the qualitative differences between the optimisation algorithms, and can be used as a guideline when a choice has to be made between the different algorithms for a particular speech processing application.

A measure of the complexity of each algorithm is provided in terms of the number of multiplications and divisions required during the MPE optimisation process . As the complexity is often proportional to the MPE frame size, and in order to permit a direct comparison with the numerical capabilities of the latest DSP chips, the complexity figures given in Fig 4.8.1 represent the number of multiplications and divisions per sampling interval. The number of additions is usually slightly smaller than the number of multiplications, and is not included in the complexity calculations.

Note that methods MS1, MS2, MS3 and BS2 vary their computational requirements from one MPE frame to another, and thus the complexity figures for these methods represent time averages. Methods MS4, MS5 and BS1 on the other hand, perform the same number of multiplications and divisions in every MPE frame. Notice that the number inside the parenthesis next to the BS1 and BS2 methods (in the "method" column) is the number of optimisation iterations n_r.

The formulas given in the Search Complexity column of Fig 4.8.1, indicate the dependency of the algorithmic complexity on the pulse rate (q is the number of pulses per MPE frame), and the size n of the MPE frame. These expressions do not include the computational effort required to determine the values of the auto-covariance (Eq 4.4.17) and cross-correlation (Eq 4.6.5), because it is independent of the pulse rate. In particular, given the input speech samples and the coefficients of the LPC sythesis filter, the number of multiplications required to calculate the auto-covariance and the autocorrelation is $4ln+n(n-1)/2$ (where l is the order of the AR-LPC filter). This figure comes to 98 operations per sample when a 12th order LPC filter and a MPE frame of 100 samples are used, and is included in all the individual complexity figures for the various pulse rates.

The average Segmental-SNR figures of Fig 4.8.2 and 4.8.3 were obtained using a 70 sec long (at 8 kHz sampling rate) speech data file, containing sentences spoken by 8 male and 7 female speakers. A MPE frame of 100 samples and a non-overlapping LPC frame of 200 samples were used. The order of the AR-LPC filter was 12 and the noise shaping filter constant γ was set to 0.9, except for methods MS5 and BS1 where it was set to 1 for the reasons explained in the relevant sections. The parameters of the coder were left unquantized, in order to remove the dependency of the SNR on the efficiency of the various parameter quantization schemes.

The SNR figures and the subjective quality obtained from each optimisation method are directly related, because the methods are very similar to each other. A comparison of the SNR figures can therefore reveal the relative improvement obtained when some methods are used instead of the others.

As seen in Fig 4.8.2, the gap between the performance of the simple and complex optimisation methods widens as the pulse rate is increased. At high pulse rates (greater than 1200 pulses/sec), the improvement in the quality of the encoded speech is especially noticeable when the complex MPE optimisation algorithms are used. At the pulse rate of 1600 pulses/sec (often used in 16 kbits/sec systems), the SNR difference between the simplest method MS1 and the most complex method BS1 is 4.5 dBs, which is perceptually significant. In contrast, the difference in SNR obtained at 400 pulses/sec is small

Method BS1 which gives the best SNR results is also the most complex. A comparison of methods MS3 and MS4 shows that their SNR values are very close, but the complexity of method MS3 can be quite smaller than that of MS4. This suggests that method MS3 can be used instead of method MS4 and provide the same results at a smaller computational cost. This is even more obvious when methods MS3a and MS4a are compared.

Another interesting comparison can be made between methods MS4 and MS5. Method MS5 gives higher SNRs although its complexity is almost the same as that of method MS4.

The BS methods are usually more complex than the MS methods and are favoured (when compared to the MS methods) by the use of small MPE frames and the application of the autocorrelation approximation (methods BS1a and BS2a). Their complexity inreases with the third power of the number of pulses q, and therefore are easier to implement when the MPE frame size is

small and the number of pulses per frame is also small. Interestingly method BS2a(i) is less complex and gives better SNR results than method MS5a at 800 pulses/sec, but this situation is reversed at higher pulse rates.

Methods BS1 and BS1a perform well under all conditions (pulse rate, frame size etc.), and give the best SNR results out of all the examined MPE optimisation methods. Their performance approaches that of the more complex Successive-Elimination and Multivariate-Optimisation methods described in the previous chapter. Unfortunately their complexity even at moderate pulse rates is already quite high.

The effect of increasing the number of iterations (from 1 to 2) in the BS algorithms is very small, even though the complexity is almost doubled. This enforces our earlier conclusion that the optimisation results of methods BS1 and BS2, are close to a local minimum of the approximation error. As method BS1 examines every avilable pulse positiom within the MPE frame, it is reasonable to assume that the results of method BS1 are close to the global error minimum.

In Fig 4.8.3(a) the variation of the SNR at 800 pulses/sec is shown, when the noise shaping filter constant γ takes different values. A definite peak is observed for most methods around the value of 0.9. As it was shown in Figures 4.6.2, 4.6.6 and 4.6.7, this peak is transferred to a lower value of γ when the pulse rate is increased (resulting greater pulse congestion). The peak is less obvious and sometimes does not occur when methods MS5 and BS1 are used, which is why γ is set to 1 for these two methods. Interestingly the SNR values of all the optimisation methods converge to the same value as γ is reduced below 0.6. It is also clear that the simpler optimisation algorithms can benefit more (in terms of SNR) from a carefully chosen value of γ.

In Fig 4.8.3(b) the dependence of the SNR on the size of the MPE frame is shown. It is evident that the use of a smaller frame reduces the relative freedom in arranging the pulses over a time period, and as a result the the SNR results are not as good as when larger frames are used. The drop in the SNR is even higher when the autocorrelation approximation is applied in small MPE frames, because the summation range for each term of the autocorrelation sequence may be comparable to the duration of the LPC filter's impulse response, and the difference between the symmetric

auto-covariance matrix $\Phi(i,j)$ and its Toeplitz approximation $\Phi_a(i,j)$ may be large.

The use of BS optimisation methods becomes more attractive when smaller frame sizes are employed. This becomes apparent when the SNR results of the BS algorithms for small frames, are compared with those obtained from the MS algorithms in Fig 4.8.3(b). It is the combination of high SNR performance and moderate complexity (when small MPE frames are used) which gives the low to intermediate bit-rate BS algorithms an advantage over the MS schemes.

In Figures 4.8.4 and 4.8.5, the power spectral distribution of the speech signal and of the noise introduced by the MPE coder, are shown for a 64 ms voiced speech segment. Method MS5 was used to define the MPE signal in such a way as to keep the SNR at a fixed level, by appropriately adjusting the number of pulses defined in each MPE frame. The SNR level was set to 12 dBs for Fig 4.8.4 and 24 dBs for Fig 4.8.5. In Figures 4.8.4(a) and 4.8.5(a) the noise shaping filter constant γ was set equal to 1 and in Figures 4.8.4(b) and 4.8.5(b) it was set equal to 0.8. The black areas occur in the spectral regions where the power of the noise is higher than the power of the speech signal.

The effect of the noise shaping is very small when the SNR is low, but it is noticeable when the SNR is high. Unfortunately the improvement brought by the use of a noise shaping filter is mostly needed when the SNR is low (at low transmission bit rates). In practice, the subjective quality of the encoded speech is slightly improved when noise shaping is used at high pulse rates (corresponding to high SNR levels).

Pulses/sec		400	800	1200	1600	2000	2400
Method	Search Complexity	Algorithm Complexity (average number of multiplications per sample)					
MS1	$O(3q)$	115	130	150	170	200	235
MS2	$O\left(3q + \frac{6}{n} q^2\right)$	115	135	160	195	235	290
MS3	$O(3q+\partial_2)$	115	145	195	285	480	900
MS4	$O\left(\frac{1}{2} q^2+\frac{7}{2} q\right)$	130	180	260	370	500	660
MS5	$O\left(\frac{1}{2} q^2+\frac{5}{2} q\right)$	125	170	245	350	475	625
BS1(1)	$O\left(\frac{1}{3} q^3+q^2-\frac{1}{3n} q^4\right)$	175	515	1295	2635	4600	7215
BS1(2)	$O\left(\frac{2}{3} q^3+2q^2-\frac{2}{3n} q^4\right)$	240	920	2475	5145	9070	14285
BS2(1)	$O\left(\frac{6}{n} q^3-\frac{5}{n} q^2\right)$	110	165	300	560	970	1560
BS2(2)	$O\left(\frac{11}{n} q^3-\frac{4}{n} q^2\right)$	115	215	475	960	1730	2830

Pulses/sec		400	800	1200	1600	2000	2400
Method	Search Complexity	Algorithm Complexity (average number of divisions per sample)					
MS1-4	$O(1)$	1	1	1	1	1	1
MS5	$O(q)$	5	10	15	20	25	30
BS1(1)	$O(q)$	5	10	15	20	25	30
BS1(2)	$O(2q)$	10	20	30	40	50	60
BS2(1)	$O\left(\frac{5}{n} q\right)$	0	1	1	1	1	2
BS2(2)	$O\left(\frac{10}{n} q\right)$	1	1	2	2	3	3

FIGURE 4.8.1 Complexity (number of multiplications and divisions per sample) of the MPE optimisation algorithms at various pulse rates. The search complexity is given as a function of the number of pulses q and the number of samples n in the MPE analysis frame.

Pulses/sec	400	800	1200	1600	2000	2400
Method	SNR (dBs)					
MS1	9.2	13.1	15.4	17.0	18.3	19.5
MS2	9.2	13.6	16.2	18.2	19.9	21.3
MS3	9.2	13.8	16.8	19.3	21.5	23.6
MS4	9.3	14.0	17.1	19.6	21.9	23.8
MS5	9.8	14.4	17.8	20.5	22.7	25.0
BS1(1)	10.2	15.2	18.8	21.4	23.7	26.0
BS1(2)	10.3	15.4	18.9	21.6	23.8	26.2
BS2(1)	9.5	14.3	17.5	20.0	22.1	24.1
BS2(2)	9.5	14.5	17.7	20.2	22.3	24.3

Pulses/sec	400	800	1200	1600	2000	2400
Method	SNR (dBs)					
MS1a	9.1	12.8	14.8	16.3	17.6	18.7
MS2a	9.2	13.4	15.9	17.7	19.2	20.4
MS3a	9.2	13.5	16.3	18.6	20.5	22.3
MS4a	9.2	13.6	16.4	18.8	20.8	22.5
MS5a	9.2	13.8	17.0	19.6	21.7	23.6
BS1a(1)	9.5	14.8	18.5	21.4	23.7	26.0
BS2a(1)	9.4	14.1	17.0	19.3	21.2	22.8

FIGURE 4.8.2 SNR performance of the MPE optimisation algorithms at various pulse rates.

Constant γ	1.00	0.95	0.90	0.85	0.80	0.70	0.60
Method	SNR (dBs) at 800 Pulses/sec						
MS1	12.0	13.0	13.1	13.0	12.7	11.6	10.4
MS4	14.0	14.1	14.0	13.6	13.1	11.9	10.6
MS5	14.4	14.6	14.4	13.6	13.0	11.7	10.3
BS1(1)	15.2	15.2	14.8	14.1	13.4	11.9	10.4
BS2(1)	13.7	14.4	14.3	13.9	13.4	12.1	10.6
MS1a	10.4	12.4	12.8	12.8	12.5	11.5	10.4
MS4a	11.8	13.5	13.6	13.4	13.0	11.8	10.5
MS5a	12.3	13.9	13.8	13.4	12.9	11.6	10.2
BS1a(1)	12.8	14.5	14.8	13.9	13.3	11.8	10.4
BS2a(1)	12.1	13.9	14.1	13.8	13.3	12.0	10.6

(a)

Frame (samples)	20	40	50	100	200
Method	SNR (dBs) at 800 Pulses/sec				
MS1	12.5	12.9	13.0	13.1	13.4
MS4	12.8	13.5	13.6	14.0	14.4
MS5	12.4	13.5	13.8	14.4	15.0
BS1(1)	13.6	14.5	14.7	15.2	15.8
BS2(1)	13.2	13.9	14.0	14.3	14.7
MS1a	11.3	12.0	12.3	12.8	13.2
MS4a	11.4	12.5	12.8	13.6	14.2
MS5a	11.5	12.7	13.1	13.8	14.4
BS1a(1)	12.3	13.5	13.8	14.8	15.0
BS2a(1)	12.2	13.3	13.5	14.1	14.6

(b)

FIGURE 4.8.3 SNR performance of the MPE optimisation algorithms at a pulse rate of 800 pulses/sec. (a) The value of the noise-shaping-filter constant γ is varied (b) The size of the MPE analysis frame is varied.

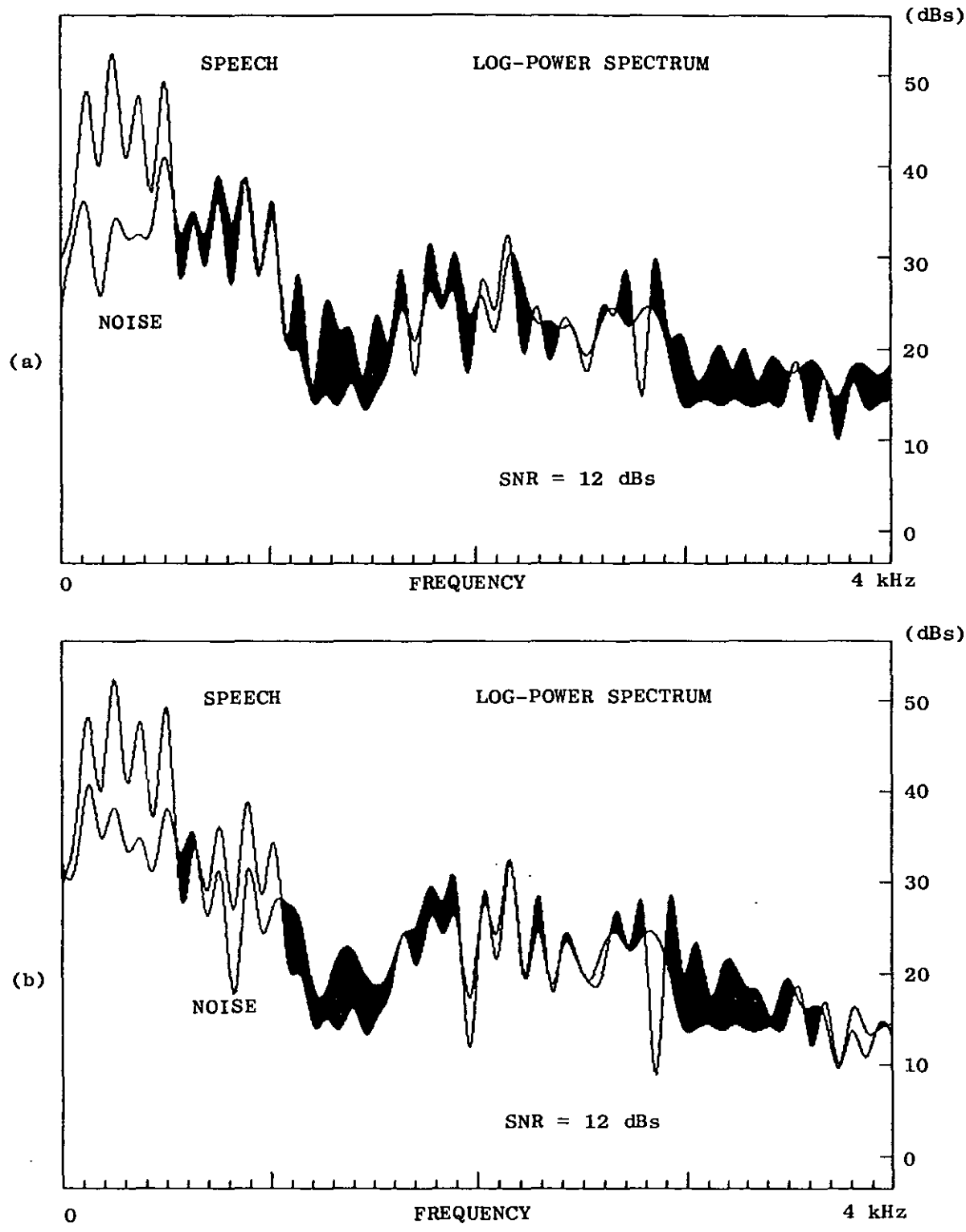


FIGURE 4.8.4 Power Spectra of Speech and Distortion (noise) introduced by the MPE coding process for (a) γ=1 and (b) γ=0.8. The duration of the time window is 64 ms and the SNR is 12 dBs. The black areas occur in the spectral regions where the power of the noise is higher than the power of speech.

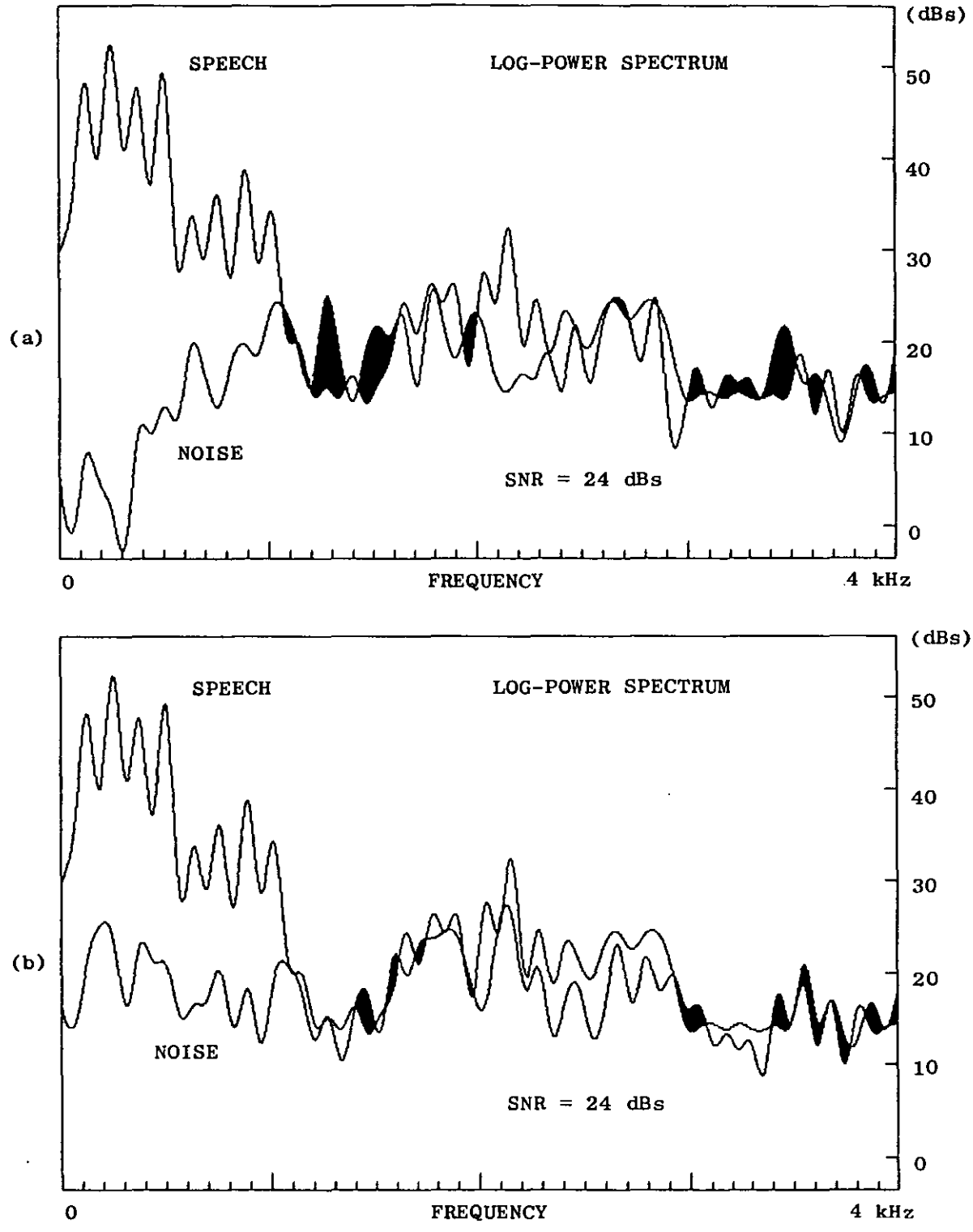


FIGURE 4.8.5 Power Spectra of Speech and Distortion (noise) introduced by the MPE coding process for (a) γ=1 and (b) γ=0.8. The duration of the time window is 64 ms and the SNR is 24 dBs.

4.9 Quantization of the MPE parameters (Pulse Amplitudes)

The efficient quantization of the LPC filter parameters [4.57,4.58,4.21] and the MPE pulse positions [4.52,4.34,4.48], are still subjects of ongoing research, even though a number of efficient quantization methods already exist. In the next chapter it will be described how the inclusion of an efficient encoding process (for the pulse positions) in the MPE optimisation algorithm, can cause a significant reduction in the number of bits allocated to the quantization of the pulse positions. For the moment though, it will be assumed that the excitation pulses are unconstrained and that every combination of pulse positions is equally likely to be chosen by the MPE optimisation algorithm. In this case, an enumerative coding algorithm which maps each possible combination of pulse positions to a different integer value [4.59,4.60], is optimum and requires $log_2\binom{n}{q}$ bits for the exact quantization of the pulse positions.

The design of optimum (MMSE) quantizers for the pulse amplitudes will now be addressed. At low transmission bit rates, the number of bits allocated to the quantization of each pulse amplitude is limited and the use of optimised amplitude quantizers becomes necessary. The pulse amplitudes are quantized using the PCM-AQF method (with forward adaptive estimation of the input variance), to allow for the wide fluctuations of the power of the speech signal

The Max-Lloyd quantizer [4.61,4.62] can be designed using an iterative optimisation process, based on the experimentally derived Probability Density Function (PDF) of the input [4.63,4.64]. The input samples are the pulse amplitudes normalised by their periodically updated standard deviation (in practice the standard deviation is quantized separately and then is used to normalise the pulse amplitudes). The iterative optimisation process is not guaranteed to converge to the global error minimum, except for a few well known theoretical PDFs (Gaussian, Laplacian, etc.), which satisfy the log-concavity sufficient condition [4.65,4.66]. For this reason, a different approach will be followed here, which involves the modeling of the experimental PDF with a theoretical PDF that guarantees a convergence to the global error minimum.

The experimental PDFs of the normalised amplitudes are shown in Fig 4.9.1 (as histograms), for 4 different pulse rates (of 480, 800, 1200 and 1600

pulses/sec). The PDFs are obviously symmetric and therefore only the right half is shown. The shape of the PDF depends on the pulse rate but is not affected by the choice of the MPE optimisation algorithm or the MPE frame update rate. The experimental PDFs were obtained using method MS5, under the conditions (speech data, MPE frame etc.) described in the previous section.

The model PDF chosen is that of the gamma distribution (the Gaussian and lognormal distributions were also examined), and the dependency of the model on the pulse rate is controlled by a single parameter. This parameter is optimised so that the theoretical model accurately fits the experimental distribution.

The general symmetric gamma PDF is :

$$P(x) = \frac{\lambda^a}{2\,\Gamma(a)}\,|x|^{a-1}\,e^{-\lambda|x|} \quad , \quad -\infty < x < \infty \quad , \quad a \geq 0 \qquad (Eq\ 4.9.1)$$

where x is the normalised pulse amplitude, and the gamma function is defined as :

$$\Gamma(a) = \int_0^{+\infty} t^{a-1}\,e^{-t}\,dt \qquad (Eq\ 4.9.2)$$

The variance of the general gamma PDF is :

$$\sigma^2 = \frac{a(a+1)}{\lambda^2} \qquad (Eq\ 4.9.3)$$

Also the log-concavity test for the general gamma PDF gives :

$$\frac{\partial^2 log\ P(x)}{\partial x^2} = \frac{-2(a-1)}{x^2} \quad , \quad x \neq 0 \qquad (Eq\ 4.9.4)$$

which is negative when $a \geq 1$. Under this condition the general gamma PDF will result a guaranteed convergence to the globally optimum quantizer levels. Since the normalised amplitude variable x has a unit variance, the model PDF becomes (using Eqs 4.9.1 and 4.9.3) :

$$P(x) = \frac{\left(\sqrt{a(a+1)}\right)^a}{2\,\Gamma(a)}\,|x|^{a-1}\,exp\left(-\sqrt{a(a+1)}\,|x|\right) \qquad (Eq\ 4.9.5)$$

The model PDF can now be adjusted to fit the experimental distribution, by optimising the value of the parameter a. The goodness of fit is measured by

the deviation of the experimental amplitude histogram from the theoretically predicted histogram. The experimental histogram is constructed by arranging the input data into k adjacent "bins", so that for example M_i normalised amplitude values are present in the interval between two thresholds y_{i-1} and y_i. The predicted number for the same interval is :

$$M_i(a) = N \int_{y_{i-1}}^{y_i} P(x)\, dx \qquad (Eq\ 4.9.6)$$

where N is the total number of amplitude values :

$$N = \sum_{i=1}^{k} M_i \qquad (Eq\ 4.9.7)$$

The value of k is chosen to be close to 50 for the experimental histograms.

The difference between the experimental and predicted value forms another random variable which is commonly assumed to have a variance of $\sigma_i^2 = M_i(a)$ [4.67]. If another simlifying assumption is made, by postulating that these random variables are normally distributed and independent, then the likelihood of the parameter a is maximised when the quantity :

$$d = \sum_{i=1}^{k} \left[\frac{M_i - M_i(a)}{\sqrt{M_i(a)}} \right]^2 \qquad (Eq\ 4.9.8)$$

is minimised. This means that the Maximum Likelihood and the weighted MMSE estimates of a are the same, when normally and independently distributed deviations are assumed [4.67,4.68]. The value of d can be minimised using standard non-linear programming methods, and the Polak-Ribiere conjugate gradient optimisation method [4.69,4.70,4.71] was chosen for that purpose.

It can be shown that the probability distribution of d near its minimum, can be approximated by the χ^2 distribution with $k-2$ degrees of freedom [4.68]. The probability that the χ^2 variable is greater than or equal to the measured minimum value of d (d_{min}) is :

$$P(\chi^2 \geq d_{min}) = 1 - \frac{\Gamma\left(\frac{k-2}{2}, \frac{d_{min}}{2}\right)}{\Gamma\left(\frac{k-2}{2}\right)} \qquad (Eq\ 4.9.9)$$

where the incomplete gamma function is defined as :

$$\Gamma(w,u) = \int_0^{+u} t^{w-1} e^{-t} \, dt \qquad \text{(Eq 4.9.10)}$$

The probability of Eq 4.9.9 gives a measure of how likely it is that the gamma distribution chosen is indeed the underlying PDF model. The value of this probability has been calculated for a number of different pulse rates, and is displayed in Fig 4.9.2. The values are generally quite small and this may be caused by the assumption that the deviations from the theoretically predicted amplitude histogram, are normally distributed. This assumption renders as extremely unlikely the experimental histogram values which differ from the predicted values by more than twice the local standard deviation, so when a few experimental values differ considerably from the predicted values, the probability given by Eq 4.9.9 becomes very small. It is also unreasonable to expect a very good fit, because the amplitude PDF changes slightly from one speaker to another.

The best results are obtained for pulse rates approaching 800 pulses/sec, and this can also be observed in Fig 4.9.1, where the theoretic (continuous line) and experimental (histogram) PDFs are superimposed. The rising part of the PDF is well approximated at low pulse rates, while the tail of the PDF is well approximated at high pulse rates.

Note that the values of the parameter a are all greater than 1, and this guarantees a convergence to the globally optimum quantizer levels. In practice, quantizers based on the gamma PDF model give very good results and perform better than optimised logarithmic or uniform quantizers.

The value of a can be expressed as an power function of the pulse rate r. A model that has been derived using the χ^2 goodness of fit measure on the values of a shown in Fig 4.9.2, is :

$$a(r) = 0.875 + 702 \, r^{-0.826} \qquad \text{(Eq 4.9.11)}$$

The predicted values of a are also shown in Fig 4.6.2, and these are very close to the original values of a. In Fig 4.9.3 the model PDFs derived from Eqs 4.9.5 and 4.9.11 are shown, for pulse rates of 400, 800, 1600 and 3200 pulses/sec. Based on these PDFs, the optimum (MMSE) scalar quantizers can be designed using the Max-Lloyd iterative optimisation process [4.63,4.64].

A comparison of the SNR values obtained at 8000, 9600 and 16000 bits/sec,

is shown below. The MPE optimisation method MS5 is used to code a 4 sec speech interval, of one sentence spoken by a male speaker and another one by a female speaker. The MPE frame and LPC frame contain 100 and 200 speech samples respectively. The log area coefficients of the LPC filter are uniformly quantized, and the estimate of the pulse amplitudes' standard deviation is updated in every MPE frame and quantized using a logarithmic quantizer (with a total of 10 bits per LPC frame). The number of bits allocated to the quantization of the LPC filter parameters and the individual pulse amplitudes, is specified in the respective columns. Three SNR figures are given for each bit rate. The first is obtained when all the parameters are left unquantized, the second is based on the optimum quantizers designed from the gamma PDF model (Eqs 4.9.5 and 4.9.11), and the third SNR value is obtained from an optimised uniform quantizer.

Bit Rate (bits/sec)	*Pulses q*	*LPC (bits)*	*Amplit. (bits)*	*No Quant. (dBs)*	*Gamma PDF (dBs)*	*Opt.Uniform (dBs)*
8000	*8*	*50*	*4*	*12.1*	*11.5*	*11.5*
8000	*9*	*54*	*3*	*12.9*	*11.3*	*10.8*
9600	*10*	*62*	*4*	*13.6*	*12.8*	*12.6*
9600	*12*	*58*	*3*	*15.0*	*12.2*	*11.5*
16000	*20*	*52*	*5*	*19.4*	*18.1*	*17.8*
16000	*23*	*56*	*4*	*20.7*	*17.0*	*16.4*

The advantage of the gamma PDF based quantizer over the optimised uniform quantizer is more pronounced when the number of bits allocated to the quantization of the pulse amplitudes is small. It is clear that at higher bit rates, more quantizer levels are necessary in order to avoid a large drop in the SNR. At 16000 bits/sec for example, 5 bits per pulse are more than adequate, that is why both the gamma and the uniform quantizer give approximately the same results. The placement of the quantizer levels becomes more critical when the number of bits per pulse is reduced to 3 (at 8000 and 96000 kbits/sec) or 4 (at 16000 bits/sec), in which case, a 0.6 dB SNR improvement can be gained from the use of the gamma PDF based quantizer. It is also clear that at a given bit rate, it is preferable to use a smaller number of pulses per frame and quantize them more accurately, than to increase the number of pulses and reduce the number of quantizer levels.

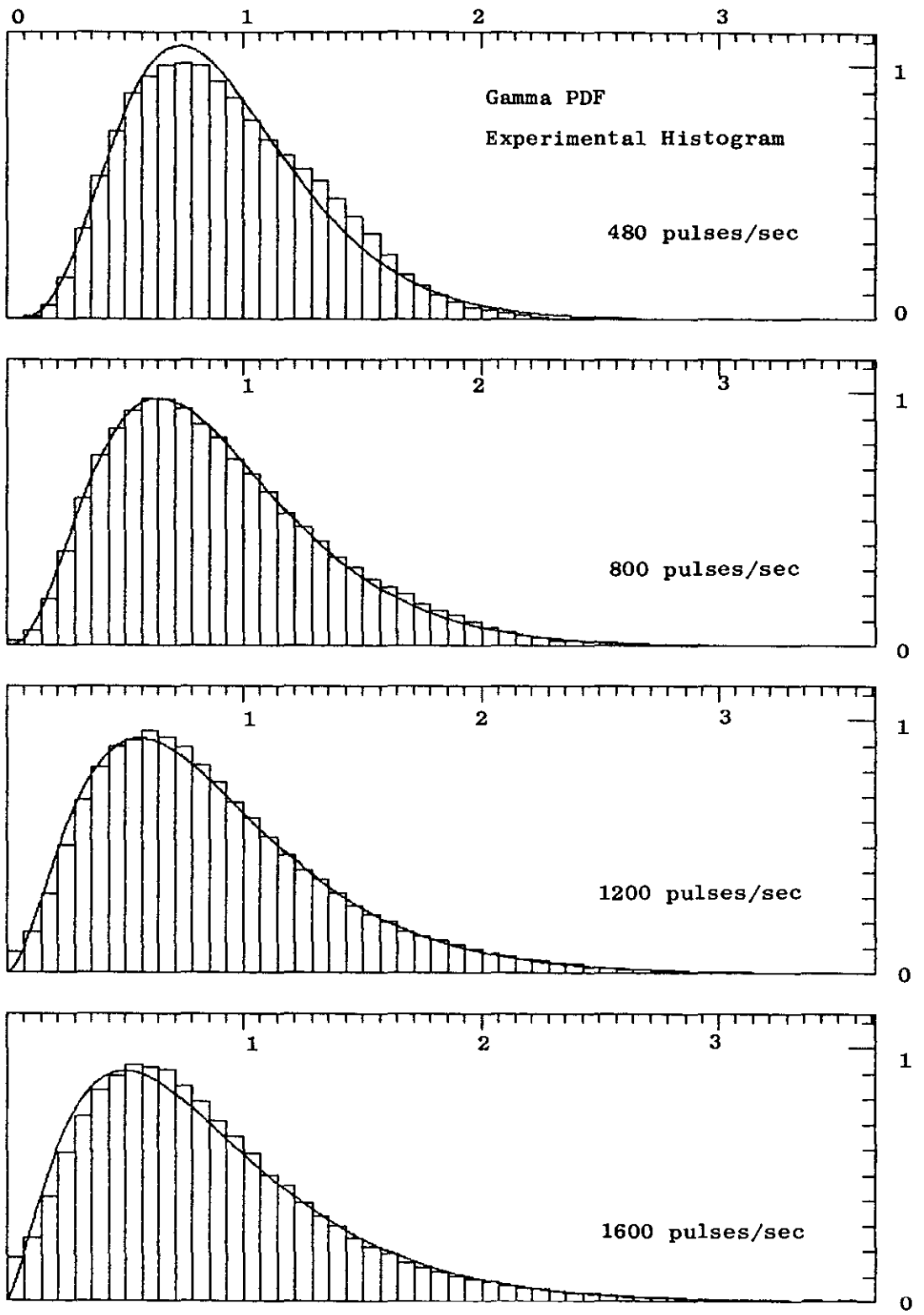


FIGURE 4.9.1 Experimental histogram and Gamma PDF model for the pulse amplitudes at four different pulse rates.

Pulse Rate (pulses/sec)	*Gamma PDF Model Parameter (a)*	*Value of (a) from Eq 4.9.11*	$P(\chi^2 \geq d_{min})$
160	10.68	11.49	0
320	6.76	6.86	3.6 E-41
480	5.21	5.16	1.1 E-15
640	4.31	4.25	3.1 E-6
800	3.69	3.68	1.2 E-4
1200	2.87	2.88	5.9 E-19
1600	2.43	2.46	4.9 E-66
2000	2.19	2.19	0
2400	2.02	2.01	0

FIGURE 4.9.2 The parameter (a) of the Gamma PDF model (2nd column) is obtained by fitting the model to the experimental data at each pulse rate. The goodness of fit measure (4th column) is based on the assumption that the error between the experimental and model PDFs has a Gaussian distribution. The predicted values of the parameter (a), obtained using Eq 4.9.11, are shown in the 3rd column.

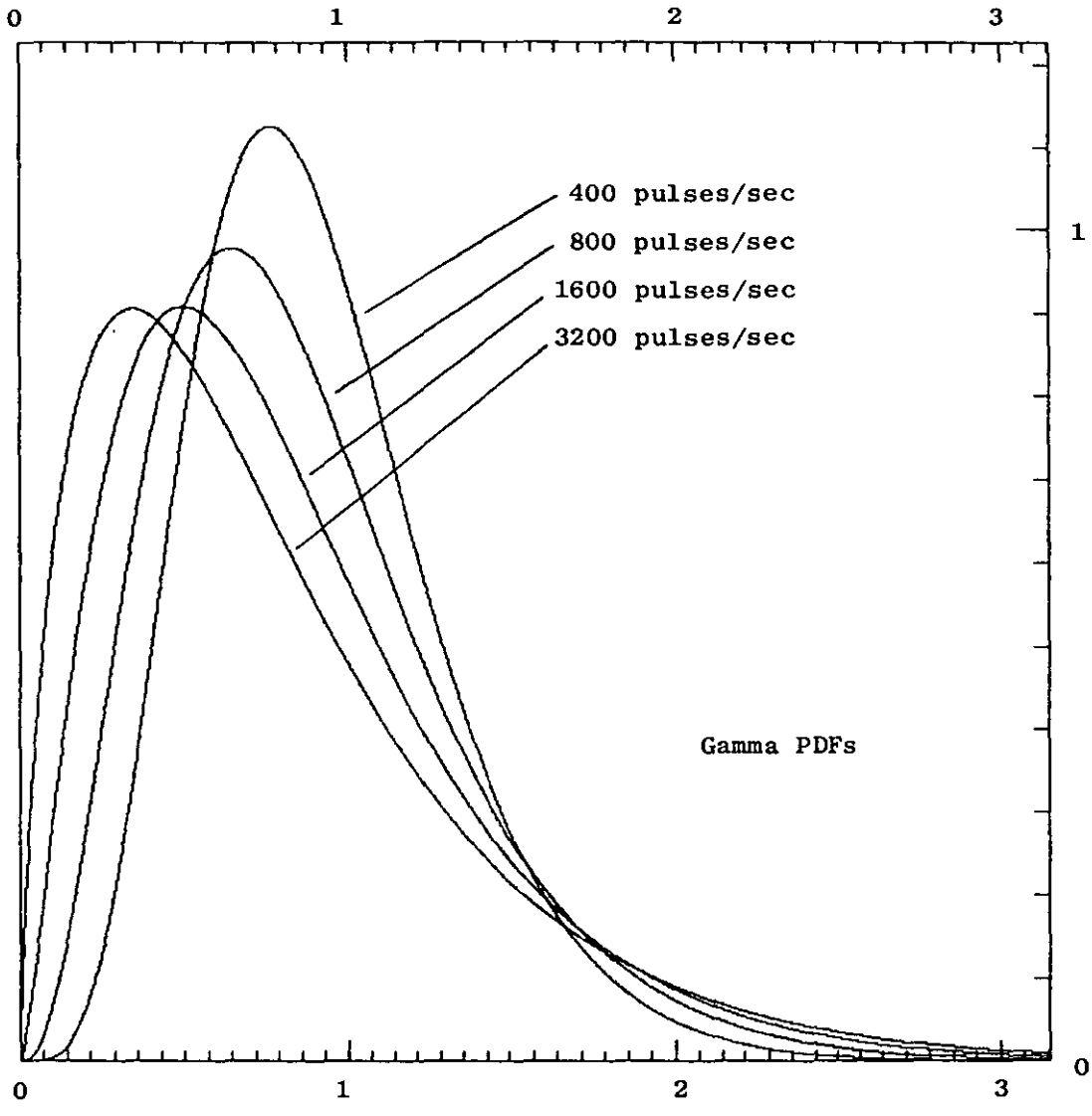


FIGURE 4.9.3 The Gamma PDF model for the pulse amplitudes, at four different pulse rates. The values of the model parameter (a) were obtained from Eq 4.9.11.

4.10 Conclusions

A detailed description of the Linear Prediction models used in MPE coders has been presented. Two cases were examined, introducing the Short Term Predictor (STP) and Long Term Predictor (LTP) models. In the first case, the synthesis filter of the MPE coder is based on an AR model and its coefficients are calculated using Linear Prediction methods. The reason for the failure of the attempted improvement (which removed the effect of the excitation signal from the estimated filter coefficients) was discovered.

The sought improvement can be obtained by changing the basic model and combining the STP model with that of the LTP. The effect of the LTP in improving the quality of the coded speech is especially noticeable for high pitched voices. Two methods were presented for the estimation of the LTP coefficients. The second method minimises the distortion introduced by the coder and leads to a joint optimisation of the LTP coefficients and the pulse amplitudes. The first method is a Linear Prediction method and is not as effective as the second, but it is usually preferred at lower bit rates because it is not affected by the infrequent updating of the coefficients.

The MPE optimisation algorithms presented, can be used in conjunction with any of the two linear filter models. They can also be used in conventional and non-conventional MPE coding schemes. The speech quality obtained from the MPE coders that employ these optimisation algorithms, compares favourably with the quality obtained from subband and RELP speech coders at 9.6 and 16 kbits/sec. Good communications quality is obtained at 9.6 kbits/sec, and near toll quality at 16 kbits/sec. To lower the bit rate, the basic structure of the MPE coder must be changed [4.19,4.30].

The complexity of the MPE optimisation methods examined, vary with the pulse rate and the size of the MPE frame. The Multi-Stage (MS) algorithms are well suited for variable bit rate applications and are generally simpler than the Block Search (BS) algorithms. Method MS1 is the simplest and can easily be implemented on a single DSP chip. For better results, methods MS2 and MS3 can be used instead. Method MS3 uses an exponential correction model and can be simpler than method MS4, even though the SNR results (and therefore the subjective quality) obtained from these two methods are very close.

Methods MS5 and BS1 produce the best optimisation results, but method BS1 can be much more complex than MS5. It may be possible to reduce the comple-

xity of method BS1 to twice the complexity of method MS5 by solving the system of equations in the MPE optimisation in a different way, but this has not been proved yet. Method BS2 performs very well at low and medium pulse rates, and can effectively replace many of the MS methods, when a small MPE frame size is used. It is also possible to simplify all the examined optimisation methods by employing the autocorrelation approximation (Eq 4.4.17).

The effect of the noise shaping filter is mainly concentrated on increasing the efficiency of the MPE optimisation algorithms, and seems to offer little advantage concerning the exploitation of the noise masking effect to improve the subjective speech quality, since the spectral redistribution of the distortion is not very effective. The tuning of the noise shaping filter though, can be used to improve the performance of the simpler MPE optimisation algorithms, and as such it can be usefull to MPE coders in general.

PDF optimised quantizers are essential when the number of bits allocated to the quantization of the pulse amplitudes is relatively small. A model based on the general symmetric gamma PDF has been developed, that guarantees the convergence of the quantizer optimisation process to the global optimum. The PDF changes as the pulse rate is increased, and a power relationship has been developed that can be used to predict the shape of the PDF at any given pulse rate.

REFERENCES

[4.1] B.Fette et al, "Experiments with a high quality, low complexity 4800 bps Residual Excited LPC (RELP) Vocoder", Proc ICASSP 1988, pp 263

[4.2] B.S.Atal, "Predictive Coding of Speech at Low Bit Rates", IEEE Trans. on Communications, Vol 30, No 4, Apr 1982, pp 600-613

[4.3] B.S.Atal, "Stochastic Gaussian Model for Low-Bit Rate Coding of LPC Area parameters", Proc ICASSP 1987, pp 2404

[4.4] R.L.Zinser, "An efficient Pitch-Aligned High-Frequency Regeneration technique for RELP vocoders", Proc ICASSP 1985, pp 969

[4.5] D.Lin, "A novel LPC synthesis model using a Binary Pulse Source Excitation", Proc ICASSP 1986, pp 461

[4.6] B.Caspers, B.S.Atal, "Role of Multi-Pulse Excitation in Synthesis of Natural-Sounding Voiced Speech", Proc ICASSP 1987, pp 2388

[4.7] J.P.Adoul et al, "Generalization of the Multipulse coding for Low Bit Rate coding purposes: The Generalized Decimation", Proc ICASSP 1985, pp 256

[4.8] D.L.Thomson, D.P.Prezas, "Selective modeling of the LPC residual during Unvoiced frames: White Noise or Pulse Excitation", Proc ICASSP 1986, pp 3087

[4.9] C.Galand et al, "High-Frequency regeneration of Base-Band Vocoders by Multi-Pulse Excitation", Proc ICASSP 1987, pp 1934

[4.10] R.C.Rose, T.P.Barnwell, "Quality Comparison of Low Complexity 4800 bps Self Excited and Code Excited Vocoders", Proc ICASSP 1987,pp 1637

[4.11] T.V.Sreenivas, "Modelling LPC-Residue by components for good quality speech coding", Proc ICASSP 1988, pp 171

[4.12] I.Lecomte et al, "Medium Band speech coding for Mobile Radio Communications", Proc ICASSP 1988, pp 231

[4.13] P.Kroon, E.F.Deprettere, "Experimental Evaluation of different approaches to the Multi-Pulse coder", Proc ICASSP 1984, pp 10.4.1

[4.14] A.Fukui, K.Shibagaki, "Implementation of a Multi-Pulse speech codec with Pitch Prediction on a Single Chip Floating-Point Signal Processor", Proc ICASSP 1987, pp 968

[4.15] H.Koyama, A.Gersho, "Fully Vector-Quantized Multipulse LPC at 4800 BPS", Proc ICASSP 1986, pp 445

[4.16] R.Garcia-Gomez et al, "Vector Quantized Multipulse LPC", Proc ICASSP 1987, pp 2197

[4.17] S.Singhal, "On encoding Filter Parameters for Stochastic coders", Proc ICASSP 1987, pp 1633

[4.18] K.Ozawa, T.Araseki, "Low Bit Rate Multi-Pulse speech coder with natural speech quality", Proc ICASSP 1986, pp 457

[4.19] S.Ono, K.Ozawa, "2.4 kbps Pitch Prediction Multi-Pulse speech coding" Proc ICASSP 1988, pp 175

[4.20] G.Davidson et al, "Real-Time Vector Excitation coding of speech at 4800 bps", Proc ICASSP 1987, pp 2189

[4.21] P.Kroon, B.S.Atal, "Quantization Procedures for the Excitation in CELP coders", Proc ICASSP 1987, pp 1649

[4.22] Y.Shoham, "Vector Predictive Quantization of the spectral parameters for Low Bit Rate Speech Coding", Proc ICASSP 1987, pp 2181

[4.23] S.Singhal, B.S.Atal, "Optimizing LPC parameters for Multi-Pulse Excitation", Proc ICASSP 1983, pp 781

[4.24] S.Singhal, B.S.Atal, "Improving Performance of Multi-Pulse LPC coders at Low Bit Rates", Proc ICASSP 1984, pp 1.3.1

[4.25] A.Parker et al, "Low Bit Rate Speech Enhancement using a new method of Multiple Impulse Excitation", Proc ICASSP 1984, pp 1.5.1

[4.26] J.Lefevre, O.Passien, "Efficient algorithms for obtaining Multipulse Excitation for LPC coders", Proc ICASSP 1985, pp 957

[4.27] J.Picone et al, "Joint Estimation of the LPC parameters and the Multi-Pulse Excitation", Speech Communications, Vol 5 (1986), pp 253

[4.28] J.Chen, A.Gersho, "Real-Time Vector APC Speech Coding at 4800 bps with adaptive Postfiltering", Proc ICASSP 1987, pp 2185

[4.29] I.M.Trancoso et al, "A study on Short-Time Phase and Multipulse LPC" Proc ICASSP 1984, pp 10.3.1

[4.30] A.H.Crossman, F.Fallside, "Multipulse-Excited Channel Vocoder", Proc ICASSP 1987, pp 1926

[4.31] V.Savvides, C.Xydeas, "A new approach to Low Bit Rate Speech Coding", Proc Int. Conf. Digital Processing of Signals in Communications, IERE, Loughborough, Sep 1988

[4.32] H.Alrutz, "Implementation of a Multi-Pulse coder on a Single Chip Floating-Point Signal Processor", Proc ICASSP 1986, pp 2367

[4.33] J.M.Santos-Suarez, "Single Chip Multi-Pulse coder with Pitch Prediction", Proc ICASSP 1988, pp 639

[4.34] P.Kroon, "Time-domain coding of (near) toll quality speech at rates below 16 KB/s", PhD Thesis, March 1985, Delft University of Technology, Netherlands

[4.35] B.S.Atal, J.R.Remde, "A New Model of LPC Excitation for producing natural-sounding speech at Low Bit Rates", Proc ICASSP 1982, pp 614

[4.36] G.A.Senensieb et al, "A non-iterative algorithm for obtaining Multi-Pulse Excitation for Linear-Predictive Speech coders", Proc ICASSP 1984, pp 10.2.1

[4.37] E.F.Deprettere, P.Kroon, "Regular Excitation reduction for effective and efficient LP-Coding of speech", Proc ICASSP 1985, pp 965

[4.38] V.K.Jain, "Simplified algorithm for Multi-Pulse LPC analysis of Speech", Proc ICASSP 1983, pp 789

[4.39] A.Ichikawa et al, "A speech coding method using Thinned-Out Residual" Proc ICASSP 1985, pp 961

[4.40] S.T.Alexander, "A simple Noniterative Speech Excitation algorithm using the LPC Residual", IEEE Trans ASSP, Apr 1985, pp 432-434

[4.41] M.Berouti et al, "Reducing signal delay in Multipulse coding at 16 kb/s", Proc ICASSP 1986, pp 3043

[4.42] V.K.Jain, "Efficient algorithm for Multi-Pulse LPC analysis of Speech", Proc ICASSP 1984, pp 1.4.1

[4.43] E.V.Stansfield et al, "Adaptive Filters in Speech Coding", Proc. IEE, Vol 134, Pt F, No 3, Jun 1987

[4.44] P.Vary et al, "Speech Codec for the European Mobile Radio System", Proc ICASSP 1988, pp 227

[4.45] J.E.Natvig, "Evaluation of six Medium Bit Rate Coders for the Pan-European Digital Mobile Radio System", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 1988, pp 324-331

[4.46] R.P.Ramachandran, P.Kabal, "Stability and Performance Analysis of Pitch Filters in Speech Coders", IEEE Trans. ASSP, Vol 35, No7, Jul 1987, pp 937-946

[4.47] P.Kabal, R.P.Ramachandran, "Joint Solutions for Formant and Pitch Predictors in Speech Processing", Proc ICASSP 1988, pp 315

[4.48] P.Kroon, E.F.Deprettere, "A Class of Analysis-by-Synthesis Predictive Coders for High Quality speech coding at rates between 4.8 and 16 kbits/sec", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 88, pp 353-363

[4.49] W.B.Kleijn et al, "Improved Speech Quality and Efficient Vector Quantization in SELP", Proc ICASSP 1988, pp 155

[4.50] T.Araseki et al, "Multi-Pulse Excited Speech coder based on Maximum Crosscorrelation Search algorithm", IEEE Proc. Global Telecommun. Conference, 1983, pp 794-798

[4.51] K.Ozawa et al, "A study on Pulse Search algorithms for Multipulse Excited speech coder realization", IEEE Journal on Selected Areas in Communications, Vol SAC-4, No 1, Jan 1986, pp 133-141

[4.52] M.Berouti et al, "Efficient Computation and Encoding of the Multipulse Excitation for LPC", Proc ICASSP 1984, pp 10.1.1

[4.53] S.Singhal, "Reducing computation in Optimal Amplitude Multipulse coders", Proc ICASSP 1986, pp 2363

[4.54] G.Carayannis et al, "Fast Recursive Algorithms for a class of Linear equations", IEEE Trans. ASSP, Vol 30, No 2, Apr 1982, pp 227-239

[4.55] B.Noble, J.W.Daniel, "Applied Linear Algebra", Prentice-Hall, 1977

[4.56] N.Gouvianakis, C.Xydeas, "A Comparative Study of Multistage Sequential and Block Sequential Search Multipulse LPC algorithms", Proc Int. Conf. IASTED, Paris, Jun 1985, pp 122-125

[4.57] A.H.Gray, J.D.Markel, "Quantization and Bit Allocation in Speech Processing", IEEE Trans., Vol ASSP-24, No 6, Dec 1976, pp 459-473

[4.58] N.Sugamura, N.Farvardin, "Quantizer Design in LSP Speech Analysis-Synthesis", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 88, pp 432-440

[4.59] J.P.Schalkwijk, "An Algorithm for Source Coding", IEEE Trans. on Information Theory, Vol 18, May 1972, pp 395-399

[4.60] T.M.Cover, "Enumerative Source Coding", IEEE Trans. on Information Theory, Vol 19, Jan 1973, pp 73-77

[4.61] J.Max, "Quantizing for Minimum Distortion", IEEE Trans. on Information Theory, Vol 6, No 1, 1960, pp 7-12

[4.62] S.P.Lloyd, "Least Squares Quantization in PCM", IEEE Trans on Information Theory, Vol 28, No 2, 1982, pp 129-137

[4.63] N.S.Jayant, P.Noll, "Digital Coding of Waveforms", Prentice-Hall, New Jersey, 1984

[4.64] P.Kabal, "Quantizers for the Gamma Distribution and Other Symmetrical Distributions", IEEE Trans, Vol ASSP-32, No 4, Aug 1984, pp 836-841

[4.65] P.E.Fleischer, "Sufficient Conditions for achieving Minimum Distortion in a Quantizer", IEEE Int. Conv. Rec., Part 1, 1964, pp 104-111

[4.66] A.V.Trushkin, "Sufficient Conditions for Uniqueness of a Locally Optimal Quantizer for a Class of Convex Error Weighting Functions", IEEE Trans on Information Theory, Vol 28, Mar 1982, pp 187-198

[4.67] W.H.Press, B.P.Flannery, "Numerical Recipes: The Art of Scientific Computing", Cambridge University Press, 1986

[4.68] T.T.Soong, "Probabilistic Modeling and Analysis in Science and Engineering", John Wiley & Sons, New York, 1981

[4.69] R.Fletcher, C.M.Reeves, "Function Minimization by Conjugate Gradients", The Computer Journal, Vol 7, 1964, pp 149-154

[4.70] E.Polak, "Computational Methods in Optimisation", Academic Press, New York, 1971

[4.71] G.N.Vanderplaats, "Numerical Optimization Techniques for Engineering Design", McGraw-Hill, 1984

CHAPTER 5

CODEBOOK SEARCH MPE CODING

5.1 Introduction

The concepts of the Multipulse Excitation model and the Analysis by Synthesis optimisation approach, have been successfully combined to produce efficient and robust MPE coders at the medium bit rates of 8-16 kbits/sec. MPE coders though suffer from an unacceptable degradation of the coded speech quality when the bit rate is reduced below 7 kbits/sec.

At 4-6 kbits/sec, a number of alternative Analysis by Synthesis coding algorithms have been developed and applied successfully in coders like CELP [5.1] and the Self-Excited Vocoder [5.2]. These coders, although capable of producing quite good quality speech at low bit rates, exhibit an almost speaker dependent behaviour. MPE coders on the other hand are not affected by the variations in the signal characteristics due to different speakers, and in addition their performance is robust in a wide range of acoustic environments

Attempts have been made to bridge the gap between the MPE and the low bit-rate Analysis-by-Synthesis coders, by combining elements from each type of coder (MPE and CELP for example). Vector Quantization [5.3] and Multi-Band operation [5.4] have been considered in order to reduce the bit rate at which MPE coders can produce acceptable results. It is still not very clear though whether MPE coders will be able to operate successfully at low transmission bit rates (2.4 to 4.8 kbits/sec), without considering major changes in the basic MPE model.

A recently proposed MPE coder operating at 2.4 kbits/sec [5.5], combines the use of a Long Term Predictor with increased constraints on the values the pulse amplitudes may take at integer multiples of the fundamental period. CELP coders on the other hand, are now efficiently implemented using sparse codebooks (a form of Multipulse Excitation), and there is a tendency to relax the severe constraints (i.e. limited number of excitation vectors) that up to now have been imposed on the structure of the excitation signal. The excitation codebook in CELP for example, has been split into one codebook for the pulse positions and another for the pulse amplitudes

[5.6,5.7,5.8]. The problem of properly designing these codebooks has not been properly addressed yet though, and in practice they are randomly generated. Another CELP scheme relaxes the constraints on the pulse positions in a sparce excitation codebook (by increasing the size of the codebook), but restricts the freedom with which the relative values of the pulse amplitudes may be chosen [5.9].

A different approach will be considered in this chapter, which contributes to the merging of the two types of Analysis by Synthesis coders, and can reduce the bit rate at which MPE coders operate, without compromising the quality of the coded speech. This reduction in the bit rate is achieved by imposing constraints on the positions of the excitation pulses, while retaining the amplitude optimisation methods used in standard MPE coders. The constraints take the form of a fixed codebook for the pulse positions, which is searched using an Analysis by Synthesis optimisation loop. Furthermore, a systematic method is proposed that optimises the position codebook for a particular coder configuration, using a training process.

Simulation results indicate that this Codebook Search (CS) MPE method is capable of more than halving the number of bits required for the coding of the pulse positions, which in a typical MPE coder account for approximately one third of the total bit rate.

The Codebook Search approach can be combined with further constraints inposed on the relative values of the pulse amplitudes, to bring the transmission bit rate to even lower values. The CS approach was originally proposed and used to encode a frequency representation of the excitation signal in a LPC coder [5.10], and can also be used as an alternative method in speech coding systems that attempt to decompose the speech signal into a limited number of time or frequency components [5.11,5.12,5.13,5.14,5.15].

5.2 Operation of the CS-MPE coder

The diagram of the CS-MPE coder is shown in Fig 5.2.1. A codebook of pulse position patterns is available to both the encoder and the decoder. In the encoder the codebook entries are translated to sets of pulse positions and an Analysis by Synthesis (AbS) procedure is applied in order to determine the optimum codebook entry. The AbS optimisation loop, in effect searches through the codebook using an appropriate search strategy. For every

codebook entry considered, the corresponding pulse amplitudes are calculated and quantized, and the approximation error is compared to previous values obtained during the search.

The approximation error is a measure of the minimum distance between the original and synthesised speech waveforms, and can be estimated in the same way as in the MPE coders already examined, causing a redistribution of the distortion energy in the frequency domain (noise shaping) and thus minimising the perceptual distortion. The index of the codebook entry that minimises the approximation error is transmitted to the decoder, together with the quantized pulse amplitudes and LPC parameters. In the decoder, the Multipulse Excitation sequence is reconstructed and the speech signal is recovered at the output of the LPC synthesis filter.

The pulse amplitudes can be estimated by solving the system of normal equations (Eq 4.4.2) to find the values :

$$\mathbf{b}_{opt} = \left[\mathbf{A}_w[q]^T \mathbf{A}_w[q] \right]^{-1} \mathbf{A}_w[q]^T \mathbf{W} (\mathbf{s}-\mathbf{m}_y) \qquad (Eq\ 5.2.1)$$

where $\mathbf{b}_{opt}$ is the q-dimensional vector containing the optimum pulse amplitudes, $\mathbf{A}_w[q]$ is the nxq convolution matrix corresponding to the Modified Synthesis Filter (MSF) $A_w(z)$, and $\mathbf{W}(\mathbf{s}-\mathbf{m}_y)$ is the n-dimensional vector containing the desired response of the MSF. The minimum energy of the distortion (approximation error) corresponding to the optimum values of the pulse amplitudes is :

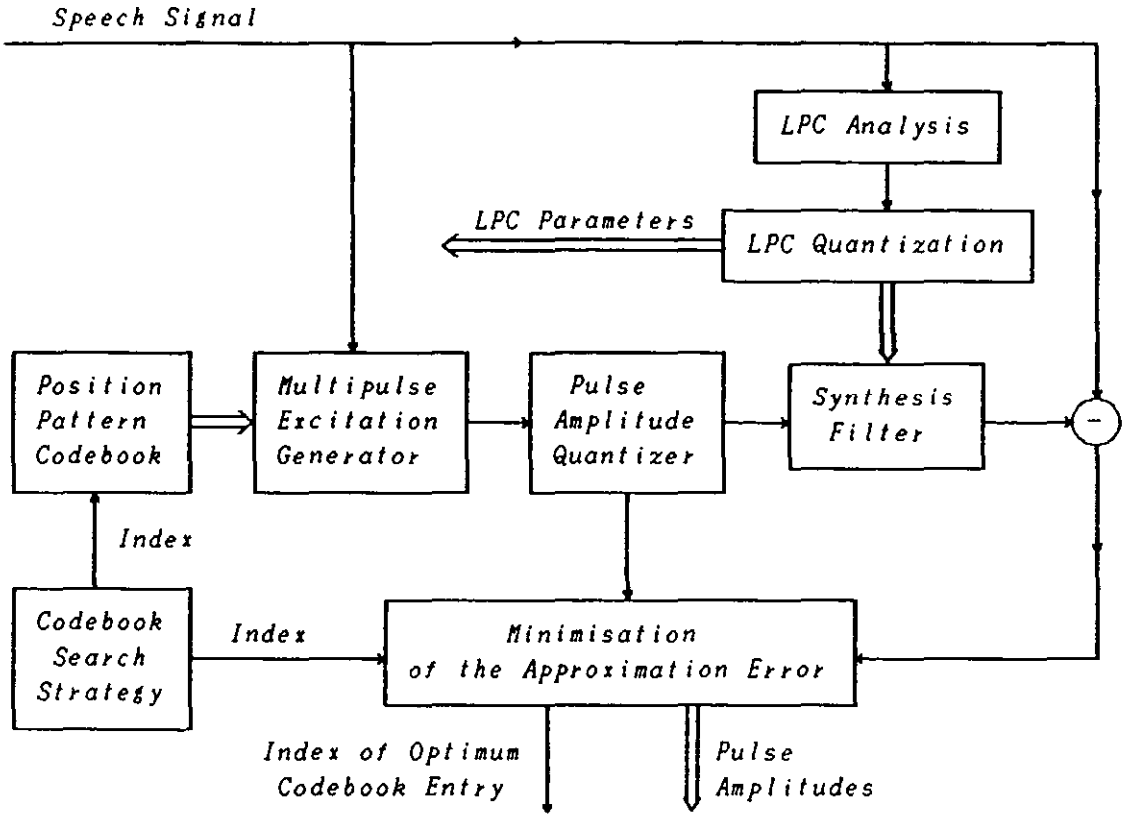
$$\varepsilon = \left[\mathbf{e}_w^T \mathbf{e}_w \right]_{min} = (\mathbf{s}-\mathbf{m}_y)^T \mathbf{W}^T \mathbf{W} (\mathbf{s}-\mathbf{m}_y) - \mathbf{b}_{opt}^T \mathbf{A}_w[q]^T \mathbf{W} (\mathbf{s}-\mathbf{m}_y) \qquad (Eq\ 5.2.2)$$

If the quantized amplitudes values $\mathbf{b}_{quant}$ are used instead, then the value of the approximation error changes to :

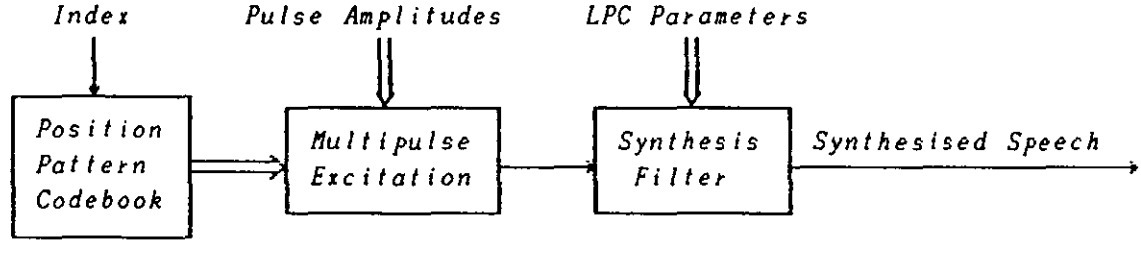
$$\mathbf{e}_w^T \mathbf{e}_w = (\mathbf{s}-\mathbf{m}_y)^T \mathbf{W}^T \mathbf{W} (\mathbf{s}-\mathbf{m}_y) - 2\, \mathbf{b}_{quant}^T \mathbf{A}_w[q]^T \mathbf{W} (\mathbf{s}-\mathbf{m}_y) + \qquad (Eq\ 5.2.3)$$

$$+ \mathbf{b}_{quant}^T \mathbf{A}_w[q]^T \mathbf{A}_w[q]\, \mathbf{b}_{quant}$$

The qxq auto-covariance matrix $\left(\mathbf{A}_w[q]^T \mathbf{A}_w[q]\right)$ can be efficiently calculated using the procedures described in Section 4.6.2, or the autocorrelation approximation (Eq 4.4.17) can be used instead. The q-dimensional cross-correlation array $\left(\mathbf{A}_w[q]^T \mathbf{W} (\mathbf{s}-\mathbf{m}_y)\right)$ can also be calculated efficiently as



MPE-CS Encoder



MPE-CS Decoder

FIGURE 5.2.1 The Codebook-Search (CS) MPE coder.

described in Section 4.6.2.

The use of Eqs 5.2.1 and 5.2.3 involves $O\left(\frac{2}{3}\,q^3+3q^2\right)$ operations (multiplications and additions) and the complexity of the Codebook Search algorithm is determined by the size of the position codebook. Compared to the figures given for the MPE algorithms in Chapter 4, the complexity of the CS-MPE optimisation (number of multiplications per sample) is $O\left(\frac{2m}{3n}\,q^3+\frac{3m}{n}\,q^2\right)$, where m is the size of the codebook. The assumption is made that every entry in the codebook is examined but, as it will be shown it the next section, this need not always be the case. Tree search algorithms can considerably simplify the CS-MPE optimisation.

The amplitude estimation process can be simplified, if consequtive codebook entries are allowed to have $q-1$ common elements (pulse positions). In such a case, the computationally efficient algorithm described when the BS2 MPE optimisation method was examined, can be used to calculate the approximation error for each codebook entry, based on results obtained for the previous entries. If in addition to this simplification, the amplitude quantizer is placed outside the optimisation loop (to be used only once for the optimum codebook entry), then the complexity of the CS optimisation is reduced to $O\left(\frac{3m}{n}\,q^2-\frac{m}{n}\,q\right)$ multiplications per sample. The performance of the CS-MPE coder is only slightly affected when codebooks with overlapping entries are employed.

5.3 Codebook Search Strategies

The complexity of the CS-MPE coder very much depends on the size and the structure of the position codebook. Small size codebooks can be computationally efficient but cannot reduce the overall transmission bit rate, because the severe restrictions imposed on the pulse positions cause a substantial loss of speech quality, which can only be compensated by an equivalent increase in the pulse rate. Such is the case for the Regular Pulse MPE coder [5.16], which uses a codebook of pulse positions with only 3 or 4 entries.

Larger codebooks can be used to sample more efficiently the parameter "space" Ψ of the pulse positions (see Section 3.4). This sampling process can be optimised by minimising the average distortion introduced by a hypothetical CS-MPE coder, using a different approach than the approach adopted by conventional Vector Quantizers operating on continuous vector

processes (LPC parameter quantization, waveform quantization, etc.). This new method of designing the position codebook in an "optimal" way, will be described in the next section.

The complete enumeration of all the codebook entries is not the only possible search strategy that a CS-MPE coder can employ. More efficient search strategies can be constructed, using a tree-structured codebook or a multiple codebook.

A tree-codebook can be arranged so that entries along the descendent paths share a number of common elements with the entries at the parental nodes. Even though this overlapping is not necessary in a tree-structured search, it can be used to simplify the error calculations, as was explained in the previous section. The tree-codebook can also be arranged so that at each level, the positions of a small group of pulses (or enven a single pulse) are defined. The pulse positions (and sometimes even the pulse amplitudes) would then remain fixed when entries at the next level are examined. This latter tree-search algorithm leads to a generalisation of the Multi-Stage MPE optimisation algorithms, mentioned in Chapter 4.

A multiple codebook search algorithm, uses the first codebook to broadly define the pulse positions, and additional codebooks to improve and refine the estimate of the pulse positions. Each codebook used, effectively causes small displacements to the optimum position values, obtained when the previous codebook was searched. This search algorithm is a special case of the Random Search MPE optimisation method, mentioned in Chapter 3.

There are many possible strategies that can be used by the hypothetical CS-MPE coder, but not every one of them lends itself to a straight forward optimisation and design of the position codebook. An unstructured codebook can be desiged and optimised more easily, because there are no constraints that need to be adhered to. This is the reason why an unstructured codebook is adopted here (which requires the enumeration of all its entries in order to find the optimum set of pulse positions), even though this increases the complexity of the CS-MPE optimisation algorithm.

The codebook optimisation method described in the next section, is applied to the unstructured position codebook, but it could be modified to take into account the constraints present in structured codebooks.

In Fig 5.3.1, the SNR (black squares) is plotted for 48 consequtive speech

frames, when 4 different random position codebooks are used. The codebooks are constructed using a uniform distribution for the pulse positions. The first two codebooks are realisations of a random codebook with 1024 entries. The third codebook also has 1024 entries, but they are arranged so that two consequtive codebook entries differ only in the position of one pulse. The fourth codebook has a 2-level tree structure with 64 branching paths at each level. Each position pattern at the second level is closely related to the parental pattern, by limiting the difference between the corresponding pulse positions to no more than 2. The MPE frame contains 50 samples and 5 pulses are defined in each frame. For comparison, the SNR obtained when method MS5 is applied to the same speech data, is also plotted (white squares and joined lines).

It is clear that the first two random codebooks give SNR results very close to the results obtained from method MS5, even though the number of bits allocated to the coding of the pulse positions by the MS5-MPE system, is more than twice the number of bits allocated by the CS-MPE system (22 bits compared to 10 bits). It is also interesting to observe that for some frames, the CS-MPE coder can achieve a higher SNR level than the MS5-MPE coder, even though the pulse positions of each codebook entry have been chosen at random.

The SNR results obtained from the third random codebook, are surprisingly only slightly inferior (by an average of 0.5 dBs), even though the freedom in choosing the pulse positions for each of the codebook entries, has been severely hampered. The SNR values obtained when the fourth codebook is used, are also smaller than for the first two codebooks by an average of 0.5 dBs, and the number of bits allocated to the coding of the pulse positions is slightly higher (12 bits instead of 10). The advantage of using the tree structured codebook is that the computational complexity of the codebook search algorithm is much smaller, because only 128 entries need to be examined for each speech frame, instead of 1024 entries previously required.

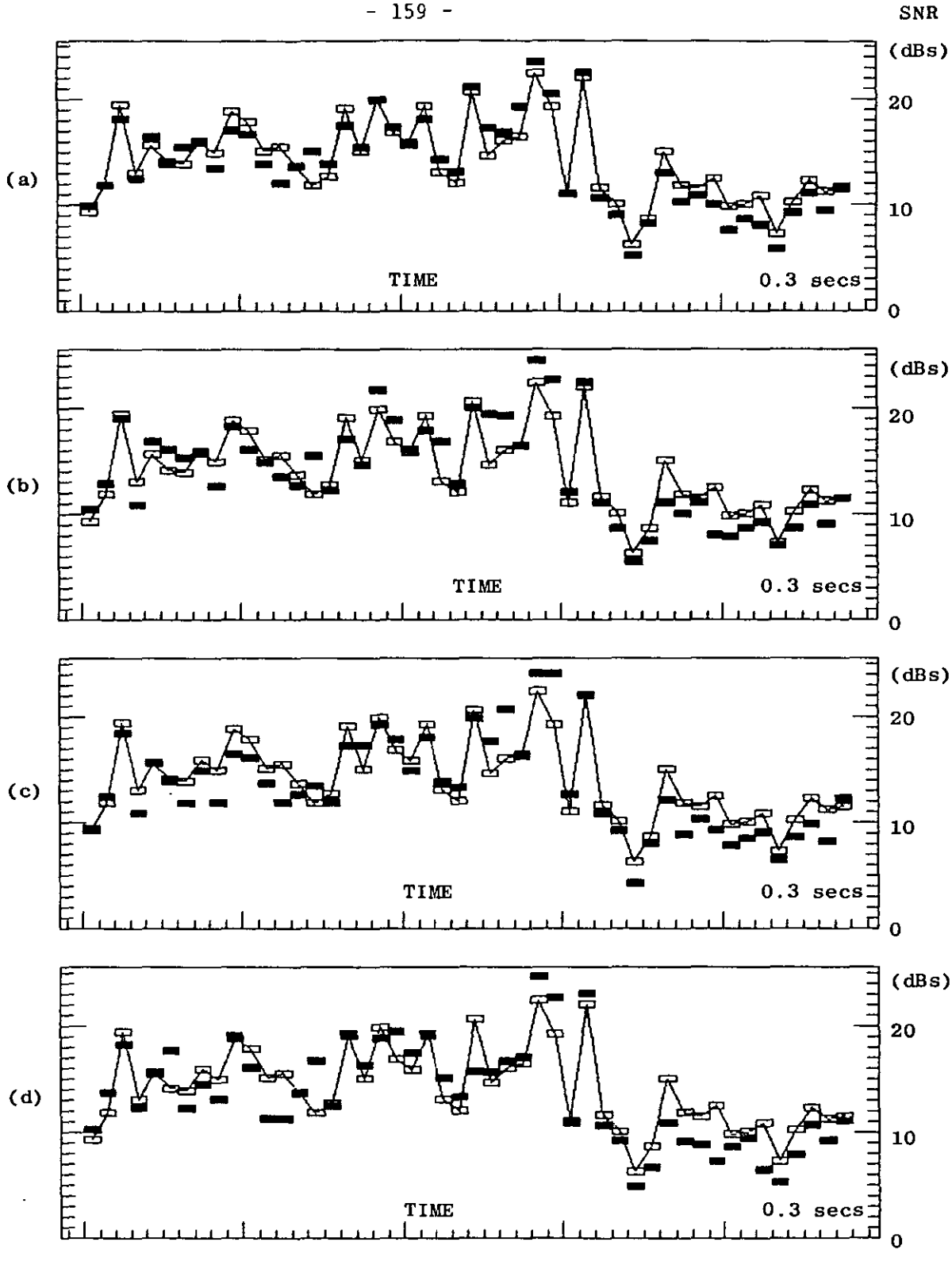


FIGURE 5.3.1 Variation of SNR (black squares) with time, when the CS-MPE coder employs four different position-codebooks : (a) Random Codebook 1 (b) Random Codebook 2 (c) Random Codebook with overlapping entries (d) Tree Codebook. The SNR values (white squares) obtained from a conventional MPE algorithm (method MS5) are also shown.

5.4 Design and Optimisation of the Position Codebook

A position codebook of a given size, can be designed to maximise the performance of the CS-MPE coder shown in Fig 5.2.1. The average distortion added to the speech signal or the average SNR achieved during the coder's operation, can be used to measure the coder's performance, and a training process can be used to determine an "optimal" pulse position codebook. The average SNR has been chosen for this purpose because it weights the distortion measurements according to the power of the speech signal. This property is closely related to the perceivable distortion, since more noise can be tolerated at higher signal levels.

If the approximation error (Eq 5.2.3) and the signal energy corresponding to frame i are ε_i and u_i respectively, then the average SNR achieved by the CS-MPE coder when a codebook C^m of size m is employed, is defined as :

$$S(C^m) = \frac{1}{N} \sum_{i=1}^{N} log\left(\frac{u_i}{\varepsilon_i}\right) \qquad (Eq\ 5.4.1)$$

where N is a large number of speech frames. The optimum codebook C^m_{opt} should therefore maximise the value of $S(C^m)$. Since the speech energy u_i is not affected by the codebook choice, the expression of Eq 5.4.1 is maximised when the expression :

$$E(C^m) = - \sum_{i=1}^{N} log(\varepsilon_i) \qquad (Eq\ 5.4.2)$$

is maximised. The optimum codebook of size m is then :

$$C^m_{opt} = \max_{C^m \subset \Psi}{}^{-1} \left[E(C^m) \right] \qquad (Eq\ 5.4.3)$$

The codebook C^m is a subset of the parameter "space" Ψ which contains all the possible combinations of pulse positions. The optimum codebook C^m_{opt} must therefore be assembled using elements of Ψ. Since the size of the parameter space Ψ is usually very large, the number of ways m of its elements can be combined to form a subset C^m is also prohibitively large. That is why a manageable (but still very large) subset Ω^w of Ψ, is used instead of Ψ :

$$C^m \subset \Omega^w \quad , \quad \Omega^w \subset \Psi \qquad (Eq\ 5.4.4)$$

where w is the number of elements (position patterns) contained in Ω^w. The assumption is therefore made that the optimum codebook C^m_{opt} can be found by thorougly examining the possible subsets of a smaller parameter "space" Ω^w

The set Ω^w is generated by a training process, whereby a conventional MPE optimisation algorithm (method MS5) is used to find the position pattern for each frame of a long speech training sequence. These patterns sample the parameter "space" Ψ in a way which is based on actual speech data. It is therefore reasonable to assume that the set Ω^w is closely related to the average speech characteristics and in a way, contains the best candidates out of which the optimum codebook C^m_{opt} should be constructed. It is obvious though that the higher the number of patterns w is, the better the optimum codebook will be.

The problem of finding the optimum codebook has been transformed to the equivalent problem of choosing m out of w position patterns, so that the coder's performance measure is maximised :

$$C^m_{opt} = \max_{C^m \subset \Omega^w}{}^{-1} \left[E(C^m) \right] \qquad (Eq\ 5.4.5)$$

This is a nonlinear combinatorial optimisation problem which can be solved using integer programming methods [5.17,5.18]. A simpler algorithm will be used here, in order to reduce the enormous complexity involved. The optimum codebook C^m_{opt} will be obtained from the set Ω^w using a thinning process which progressively reduces the number of elements of Ω^w in the least destructive way.

The codebook optimisation algorithm starts by considering the set Ω^w as a codebook of size w. The performance of the CS-MPE coder using this codebook should be better than the performance obtained when any other codebook of smaller size is used instead. The value of $E(\Omega^w)$ is therefore an upper limit for the performance of a codebook of size m :

$$E(C^m) \leq E(\Omega^w) \quad , \quad C^m \subset \Omega^w \qquad (Eq\ 5.4.6)$$

The optimisation algorithm then searches for an element of Ω^w that could be removed from the codebook and cause a minimum drop in the performance. The codebook left after the removal of the ith element of Ω^w is Ω^{w-1}_i, which

now contains only $w-1$ elements. To find the optimum codebook of size $w-1$, the set Ω_i^{w-1} which maximises the performance measure must be identified. The optimum codebook of size $w-1$ will also minimise the drop in the performance of the CS-MPE coder, when moving from a codebook of size w to a smaller one of size $w-1$:

$$C_{opt}^{w-1} = \max_{1 \le i \le w}^{-1} \left[E(\Omega_i^{w-1}) \right] = \min_{1 \le i \le w}^{-1} \left[E(\Omega^w) - E(\Omega_i^{w-1}) \right] \qquad (Eq\ 5.4.7)$$

The element of Ω^w that must be removed, is identified by calculating the value of $E(\Omega_i^{w-1})$ for $i=1,2,\ldots,w$. The process continues by removing another element and forming the optimum codebook of size $w-2$:

$$C_{opt}^{w-2} = \max_{1 \le i \le w-1}^{-1} \left[E(\Omega_i^{w-2}) \right] = \min_{1 \le i \le w-1}^{-1} \left[E(C_{opt}^{w-1}) - E(\Omega_i^{w-2}) \right] \qquad (Eq\ 5.4.8)$$

After $w-m$ such optimisation steps, the optimum codebook of size m will be:

$$C_{opt}^{m} = \max_{1 \le i \le m+1}^{-1} \left[E(\Omega_i^{m}) \right] \qquad (Eq\ 5.4.9)$$

The codebook optimisation problem has been solved in a series of steps, whereby a sequence of "optimum" codebooks is defined, whose size is continuously decreasing. The smaller codebooks are subsets of the preceding codebooks and they are bound by the performance measurements of the larger codebooks. A monotonically decreasing sequence of performance measurements is formed, which progressively reduces the upper limit of the performance that can be achieved using a codebook of size m :

$$E(\Omega^w) \ge E(C_{opt}^{w-1}) \ge \ldots \ge E(C_{opt}^{m+1}) \ge E(C_{opt}^{m}) \qquad (Eq\ 5.4.10)$$

Since the difference between two consequtive elements of this series is minimised (Eqs 5.4.7 and 5.4.8), the value of $E(C_{opt}^{m})$ is maximised and thus, the codebook C_{opt}^{m} itself is optimised. This optimisation algorithm is constrained by the inclusive relationships between the optimised codebooks and can be described using the general set of equations :

$$C_{opt}^{w-l} = \max_{1 \le i \le w-l+1}^{-1} \left[E(\Omega_i^{w-l}) \right] \quad , \quad 1 \le l \le w-m \qquad (Eq\ 5.4.11)$$

An optimisation algorithm similar to the one just described can also be developed, which will build up the optimum codebook of size m, starting from

a codebook of size one. Using the same reasoning that led to Eq 5.4.11, this algorithm can be defined by the set of equations :

$$C_{opt}^{l} = \max_{1 \le i \le w-l+1}{}^{-1} \left[E(\Omega_i^l) \right] \quad , \quad 1 \le l \le m \qquad (Eq\ 5.4.12)$$

where now Ω_i^1 is the single-element subset of Ω^w which is formed by removing all the elements of Ω^w except the element i. Similarly, Ω_i^2 is formed by adding another element of Ω^w to the previously optimised (single-element) codebook C_{opt}^1.

Usually the size of the codebook is chosen to be much smaller than the number of training vectors ($m \ll w$), and when that happens, the optimisation algorithm defined by Eq 5.4.12 becomes computationally more efficient than the algorithm defined by Eq 5.4.11. In the next section however, a fast implementation of the codebook optimisation algorithm will be presented, which renders the two methods of Eq 5.4.11 and Eq 5.4.12 computationally equivalent. For this reason and bacause of its better optimisation properties, the algorithm defined by Eq 5.4.11 was selected and used in the proposed CS-MPE schemes.

5.5 Fast Implmentation of the Codebook Optimisation Algorithm

The application of the codebook optimisation Eqs 5.4.11 requires $\left(\frac{1}{3} N w^3\right)$ computations of the approximation error (Eq 5.2.3), where N is the number of frames into which the speech training sequence has been partitioned, and w is the number of position patterns contained in the set Ω^w. Since each error computation requires the solution of a system of equations (Eq 5.2.1), the computational complexity of the optimisation algorithm becomes prohibitively large even for a small number of training vectors. In order to reduce the compuational load, two modifications of the codebook optimisation algorithm will be considered.

The first modification involves the calculation of the LPC filter's transient response (vector $\mathbf{m}_y$ in Eq 5.2.2 and Eq 5.2.3). The transient response of the LPC filter in every frame, is normally dependent on the output of the CS-MPE coder in the previous frames. This dependency is now broken, and it is assumed that the transient response can remain fixed, at least for part of the optimisation process.

Originally the transient response is obtained by running a conventional MPE coding algorithm (method MS5) on the speech training data. The codebook optimisation algorithm then considers the samples of the transient response constant while the codebook is being optimised, but periodically a decision may be taken to recalculate and update the values of the transient response, by running the CS-MPE coder on the speech training data, using the smallest optimum codebook that has been defined by the optimisation process. This modification hardly affects the results obtained from the optimisation, but is the element that mostly contributes to the reduction in the complexity of the codebook optimisation algorithm.

The second modification removes the pulse amplitude quantizer from the error estimation process, so that the simpler Eq 5.2.2 can be used instead of Eq 5.2.3, to calculate the approximation error. An added advantage of this simplification is that the results of the optimisation (optimum position codebook) will not depend on the characteristics and the accuracy of the pulse amplitude quantizer.

In the first step of the codebook optimisation algorithm, the optimum position codebook C_{opt}^{w-1} is constructed. Eq 5.2.2 is used to calculate the value of the approximation error ε_j^k, for every frame j of the speech training sequence and for every entry k of the codebook Ω^w. The results are stored in a Nxw matrix $E(j,k)$, by taking the logarithms first :

$$E(j,k) = log\left(\varepsilon_j^k\right) \quad , \quad j=1,2,\ldots,N \quad , \quad k=1,2,\ldots,w \qquad (Eq\ 5.5.1)$$

The hypothetical CS-MPE coder employing the Ω^w codebook, would choose the codebook entry (position pattern) which would minimise the approximation error in each frame. The elements of matrix $E(j,k)$ should therefore be rearranged so that its first column would contain the logarithm of the minimum error value for each frame, the second column would contain the second smallest value, and so on. This operation involves the sorting of the elements in each row of $E(j,k)$, in order of magnitude :

$$E(j,k) < E(j,l) \quad , \quad k<l \quad , \quad j=1,2,\ldots,N \qquad (Eq\ 5.5.2)$$

A second Nxw matrix $P(j,k)$ stores the indexes of the position patterns that correspond to the elements of $E(j,k)$. The two matrices are shown in Fig 5.5.1(a) in a simulation of the optimisation algorithm for 12 speech

frames and a codebook of 6 position patterns.

To find the optimum codebook C_{opt}^{w-1}, the position pattern which will cause the minimum drop in the performance of the CS-MPE coder, if it is removed from Ω^w, must be identified. The drop in performance is measured by the difference between two performance measurements :

$$v(i) = E(\Omega^w) - E(\Omega_i^{w-1}) = -\sum_{j=1}^{N} E(j,1) - \left[-\sum_{\substack{j=1 \\ P(j,1)\neq p_i}}^{N} E(j,1) - \sum_{\substack{j=1 \\ P(j,1)=p_i}}^{N} E(j,2) \right] ,$$

$$, \quad p_i \in \Omega^w \quad , \quad i=1,2,\ldots,w \qquad (Eq\ 5.5.3)$$

By rearranging the terms of Eq 5.5.3, a simpler formula is obtained :

$$v(i) = \sum_{\substack{j=1 \\ P(j,1)=p_i}}^{N} \left[E(j,2) - E(j,1) \right] \quad , \quad p_i \in \Omega^w \quad , \quad i=1,2,\ldots,w \qquad (Eq\ 5.5.4)$$

The index of the pattern that will minimise the drop in performance, if it is removed from Ω^w, is therefore :

$$p = \min_{1\leq i\leq w}^{-1} \left[v(i) \right] \qquad (Eq\ 5.5.5)$$

and the optimum position codebook of size $w-1$ is :

$$C_{opt}^{w-1} = \Omega_p^{w-1} \qquad (Eq\ 5.5.6)$$

In the second step, the algorithm must determine the optimum codebook C_{opt}^{w-2} of size $w-2$. To do that, the approximation error matrix $E(j,k)$ and the pattern matrix $P(j,k)$ do not have to be recalculated, but can be obtained from the matrices that were formed in the previous step. This is done by removing from the two matrices formed in the first step, all the elements associated with the rejected position pattern p_p. The rest of the matrix elements are then shifted to occupy the empty matrix cells, so that two new $(N)x(w-1)$ matrices are formed, which can be used in the second optimisation step. Equations 5.5.4 and 5.5.5 are again emloyed to identify the second pattern to be removed from Ω^w, and the optimum codebook of size $w-2$ becomes:

$$C_{opt}^{w-2} = \Omega_p^{w-2} \qquad (Eq\ 5.5.7)$$

The optimum codebook of size m is found after $w-m$ such steps. Simulation results from the first 3 steps of the optimisation algorithm are shown in Figures 5.5.1(a),(b) and (c). All the references to the pattern which is removed at each step, are enclosed inside brackets. Patterns 1, 2 and 6 are removed in the first second and third step respectively. Notice how the number of columns is reduced as the optimisation progresses.

The computational effort involved in the application of Eqs 5.5.4 and 5.5.5 is very small compared to the calculation of the approximation error matrix. This matrix only needs to be calculated once (initially and then every time the transient response of the LPC filter is updated) and the number of operations involved determines the complexity of the codebook optimisation algorithm. The number of computations of the approximation error has dropped to (Nw), which is considerably lower than the figure given at the beginning of this section.

When the number of speech frames in the training data set and the number of patterns in Ω^w are large, the matrices $E(j,k)$ and $P(j,k)$ become large and computer storage problems may arise. These problems can be solved by reducing the number of columns of the two matrices to a much smaller number than originally considered. Initially, the approximation error will be calculated for every frame and every pattern in Ω^w, but only the d smallest values of the error (for each frame) will be stored in the error matrix $E(j,k)$. Similarly, only d index values will be stored in every row of the index matrix $P(j,k)$.

When the matrix elements corresponding to the codebook entry rejected at each step, are removed, some of the matrix rows will be unaffected (because the particular pattern was not included), and some will be reduced in length (by removing the corresponding elements and shifting the rest).

Since the full matrices will not be available during the codebook optimisation, care must be taken to determine any rows where all but one of the matrix elements have been removed. When this happens, each "empty" row of the error matrix must be updated, by calculating the approximation error for the corresponding frame and for every entry in the optimum codebook defined up to this stage of the process. Again only the d smallest error

values and the corresponding pattern indexes will be stored.

The added computation increases the complexity of the codebook optimisation algorithm, but it has been found experimentally that the computational complexity only doubles when :

$$d \approx \frac{w}{100} \qquad (Eq\ 5.5.8)$$

This value reduces the storage requirememnts by 100 times.

The flow diagram of the algorithm is shown in Fig 5.5.2. Notice that the transient response of the LPC filter (and therefore the approximation error and index matrices also) is updated when the size of the optimum position codebook reaches a value which is a power of 2 (corresponding to an integer number of bits required for the encoding of the pattern index).

(a)

APPROXIMATION ERROR MATRIX

23	(58)	59	65	74	92
12	20	(21)	40	52	76
(21)	21	23	45	58	94
25	59	77	77	(88)	90
13	14	(27)	62	81	94
24	46	61	(65)	95	98
(16)	25	34	53	58	99
14	21	46	49	(50)	70
36	(44)	47	62	64	94
26	65	(78)	78	96	99
19	31	(39)	49	51	53
(10)	16	30	60	63	86

PATTERN INDEX MATRIX

5	(1)	2	3	4	6
2	5	(1)	3	4	6
(1)	2	5	3	6	4
4	5	3	6	(1)	2
3	5	(1)	4	2	6
4	5	3	(1)	6	2
(1)	2	5	4	3	6
3	2	4	6	(1)	5
2	(1)	6	3	4	5
3	4	(1)	5	6	2
3	4	(1)	2	5	6
(1)	3	5	6	2	4

(b)

APPROXIMATION ERROR MATRIX

23	(59)	65	74	92
(12)	20	40	52	76
(21)	23	45	58	94
25	59	77	77	(90)
13	14	62	(81)	94
24	46	61	95	(98)
(25)	34	53	58	99
14	(21)	46	49	70
(36)	47	62	64	94
26	65	78	96	(99)
19	31	(49)	51	53
16	30	60	(63)	86

PATTERN INDEX MATRIX

5	(2)	3	4	6
(2)	5	3	4	6
(2)	5	3	6	4
4	5	3	6	(2)
3	5	4	(2)	6
4	5	3	6	(2)
(2)	5	4	3	6
3	(2)	4	6	5
(2)	6	3	4	5
3	4	5	6	(2)
3	4	(2)	5	6
3	5	6	(2)	4

(c)

APPROXIMATION ERROR MATRIX

23	65	74	(92)
20	40	52	(76)
23	45	(58)	94
25	59	77	(77)
13	14	62	(94)
24	46	61	(95)
34	53	58	(99)
14	46	(49)	70
(47)	62	64	94
26	65	78	(96)
19	31	51	(53)
16	30	(60)	86

PATTERN INDEX MATRIX

5	3	4	(6)
5	3	4	(6)
5	3	(6)	4
4	5	3	(6)
3	5	4	(6)
4	5	3	(6)
5	4	3	(6)
3	4	(6)	5
(6)	3	4	5
3	4	5	(6)
3	4	5	(6)
3	5	(6)	4

FIGURE 5.5.1 Results from the first three stages (a), (b) and (c) of the Codebook Optimisation process. The matrix rows correspond to individual speech frames, and the elements of each row are the values of the approximation error arranged in order of magnitude (Approximation Error Matrix), and the corresponding indexes of the Position-Codebook entries (Pattern Index Matrix).

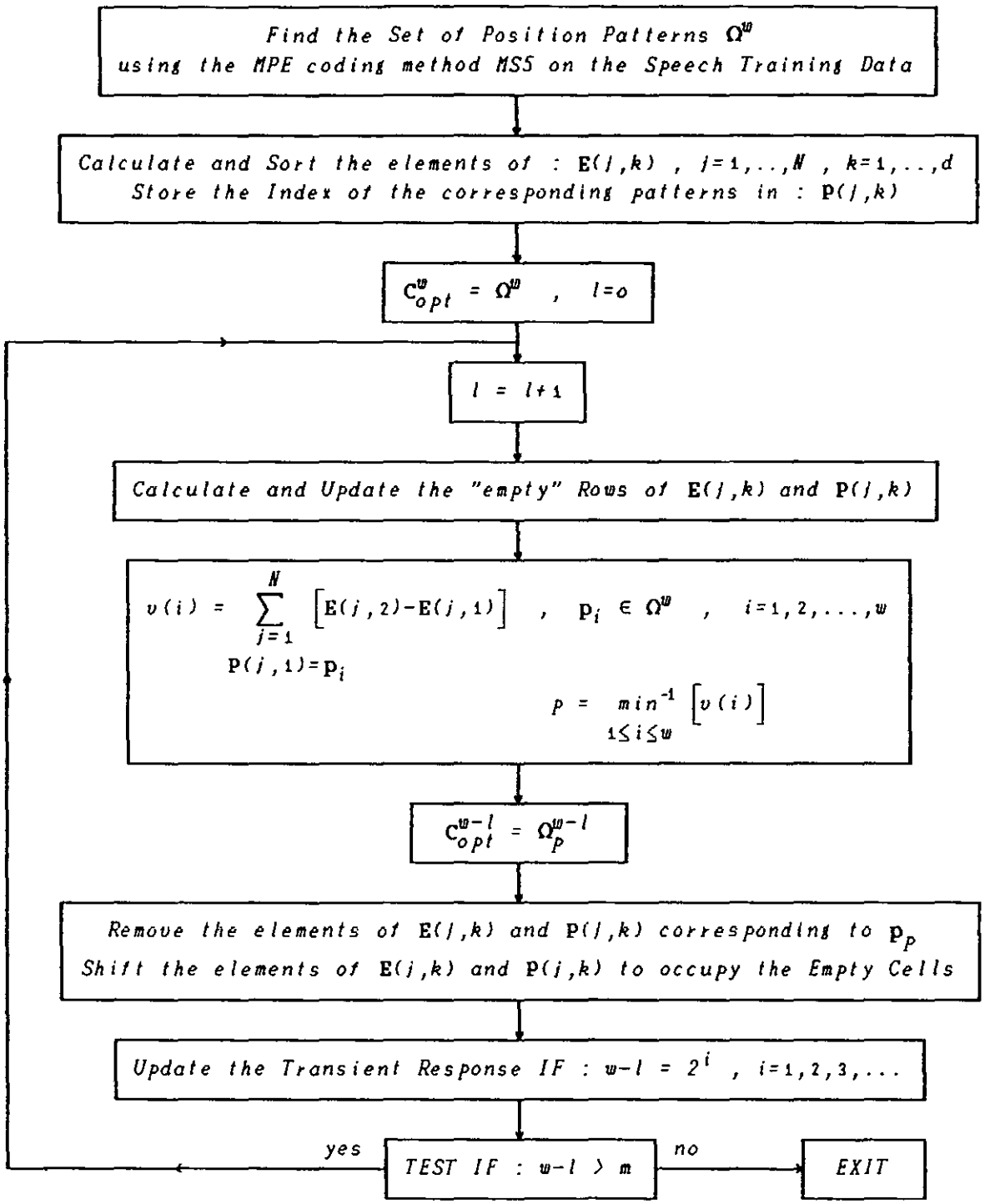


FIGURE 5.5.2 Flow diagram of the Fast Algorithm for the optimisation of the Pulse-Position-Codebook.

5.6 Results

CS-MPE coders are more efficient than conventional MPE systems (when operating under the same conditions) in coding the pulse positions. This is because in conventional MPE schemes, any combination of pulse positions is considered possible, and the description of the pulse positions to the decoder has to be very accurate. In contrast, CS-MPE coders employ a form of Vector Quatization for the pulse positions (codebook of position patterns), and use a distance measure based on the energy of the distortion introduced by the coding process.

The Codebook Search optimisation, samples the parameter space of the position variables in a random or structured way and as seen in Section 5.3, the performance of the coder's error minimisation process can be as good as that obtained from the best optimisation techiques described in Chapter 4, and sometimes even better.

The number of bits required for the coding of the pulse positions can be less than half the number of bits required by the conventional MPE algorithms, while the performance and the speech quality obtained from the two types of coders are very similar. CS-MPE coders can therefore operate at lower bit rates than convetional MPE coders, while retaining the same standards in the quality of the encoded speech.

The CS-MPE system can achieve this coding efficiency even when it employs a random position codebook. Undoubtedly though, a properly designed codebook can give better results. This happens because the ideal distribution of the pulses within the bounds of the MPE frame, is not uniform. This can be seen by plotting the long-term discrete PDF for the pulse positions when a conventional MPE coding system is used (see Fig 5.6.1(a)).

The PDF shown in Fig 5.6.1(a) was obtained using method MS5 to optimise the positions of 9 pulses in every frame of 50 speech samples. To find the average distribution, 70 secs of speech from 8 male and 7 female speakers was used. As seen in Fig 5.6.1(a), the first and last sections of the PDF deviate considerably from the uniform PDF. Fig 5.6.1(b) shows the PDF of the distance (in samples) between consequtive pulses obtained when method MS5 is used. This can be compared to the corresponding PDF of Fig 5.6.1(c) which is obtained when the pulses are trully uniformly distributed. It is evident from these two PDFs, that two consequtive pulses are more likely to be

positioned at adjacent locations when uniformly (randomly) distributed, than when defined by an efficient MPE optimisation algorithm. A codebook optimisation process can take advantage of these differences and adjust the properties of the position codebook to take into account the average speech characteristics.

In Fig 5.6.2, the average Segmental-SNR obtained from a 4 sec speech interval (containing two sentences by one male and one female speaker), is given for a conventional MPE coder (employing method MS5) and for a CS-MPE coder employing first an optimised (trained) and then a random codebook. Two pulse rates are considered i.e. 1120 and 1440 pulses/sec. The MPE frame contains 50 samples, while a 12th order AR-LPC synthesis filter is employed and the LPC frame is set to 200 samples. The value of the noise shaping filter constant γ is equal to 1. Notice that the pulse amplitudes and the LPC parameters are left unquantized.

The optimisation of the codebook was performed as described in Section 5.5, and a different set of speech training data (than the set used in the coding experiments of Fig 5.6.2) was used to derive first a Ω^m set of 4096 position patterns and then C^m_{opt} codebooks of various sizes. The same speech training data were used by the codebook optimisation process to calculate the performance measurements (average SNR). The variation of the average SNR during the optimisation process, is shown in Fig 5.6.3(a) for the two pulse rates of 1120 and 1440 pulses/sec. The random codebook contains uniformly distributed pulses, and both codebooks are unstructured (complete enumeration of all the codebook entries is required). A separate column in Fig 5.6.2 gives the number of bits required for the coding of the pulse positions in each frame (which in the case of the CS-MPE coder indicates the size of the codebook).

As seen in Fig 5.6.2, when the CS-MPE coder employs a 10-bit optimised codebook or a 13-bit random codebook, it gives SNR results which are very close to the results obtained from the MS5-MPE coder. The number of bits required for the coding of the pulse positions is therefore between one half and one third the number of bits required by the MS5-MPE coder in order to achieve the same SNR performance as the CS-MPE coder. When less efficient (than method MS5) MPE optimisation methods are employed, smaller codebooks are required by the the CS-MPE coder to achieve the same performance as the MPE coder.

The efficiency in coding the pulse positions is translated to a reduction of the transmission bit rate by approximately 3000 bits/sec. This figure is typical of the savings that can be achieved at a bit rate higher than 12 kbits/sec. When the transmission rate is close to 9600 bits/sec, the savings are reduced to approximately 2000 bits/sec. It is clear (from Fig 5.6.2) that the optimised codebook is more efficient (by 3 bits) than the random codebook in achieving the same SNR performance. The steady decrease of the SNR as the size of the codebook is reduced, is evident from both Figures 5.6.2 and 5.6.3(a), and is equivalent to the SNR drop observed in common scalar and vector quantizers when the size of the codebook is reduced.

When the size of the MPE frame or the pulse rate is increased, the size of the codebook must also be increased, to cope with the increased number and mobility of the pulses. This means that a certain ratio must be kept between the number of bits allocated to the coding of the pulse positions by a conventional MPE coder and by a CS-MPE coder. A small codebook leads to a loss in speech quality and can only be compensated by a significant rise in the pulse rate, which eventually consumes the potential benefits that could be gained by using the Codebook Search technique.

A large unstructured codebook would be cumbersome to use, due to the complexity of the codebook search process. Tree codebooks and multiple codebooks can be used to reduce the complexity of the CS-MPE coding process, at a cost of a slightly increased transmission bit rate.

At a given bit rate, the parameters chosen for the CS-MPE coder are usually different than in conventional MPE coders. A smaller frame is preferred and the pulse rate is usually higher. The quantizers used for the normalised amplitude values are Gaussian-optimised quantizers. As seen in Fig 5.6.3(b), the long-term PDF of the normalised amplitudes (for a pulse rate of 1440 pulses/sec), is much closer to the Gaussian unit variance distribution (superimposed curve) than it is to the gamma distribution used in Chapter 4 for the conventional MPE coders.

The results given below show the average Segmental-SNR obtained from the 4 sec speech segment which was also used to provide the coding results obtained from conventional MPE coders, in Chpater 4. A 12th order AR-LPC filter is employed and the LPC frame contains 200 samples. The value of the noise shaping filter constant γ is set equal to 1. The RMS value of the

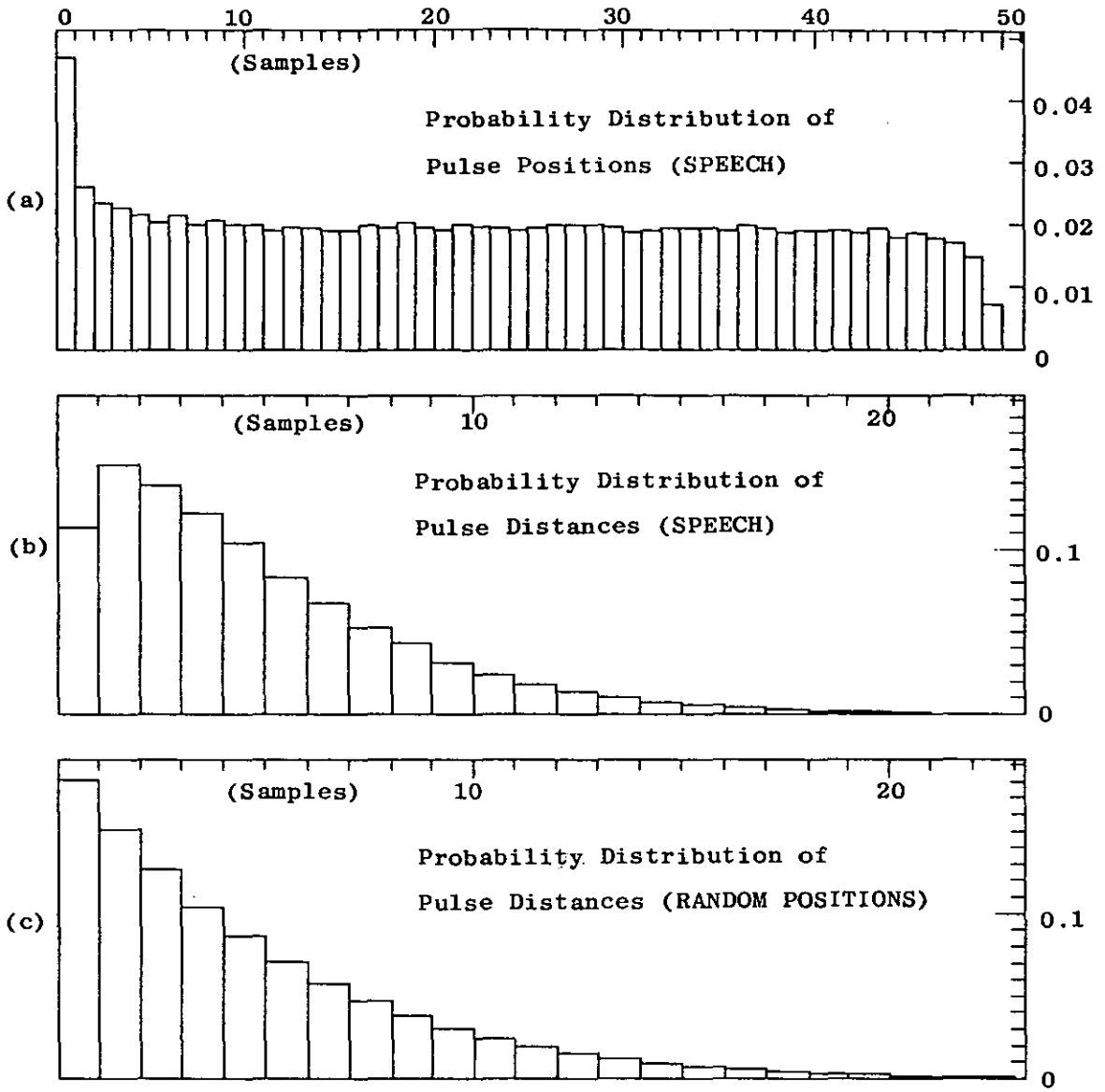


FIGURE 5.6.1 Probability ditributions for : (a) Pulse Positions obtained from coding of speech (b) Distance (in samples) between consecutive pulses, obtained from coding of speech (c) Distance between consecutive pulses when their positions are randomly distributed. The size of the MPE frame is 50.

Pulses/sec	1120		1440	
METHOD	BITS*	SNR (dBs)	BITS*	SNR (dBs)
MPE - Method MS5	27	15.6	32	17.8
MPE - CS Trained	8	14.5	8	16.4
MPE - CS Trained	9	14.9	9	16.8
MPE - CS Trained	10	15.3	10	17.2
MPE - CS Random	8	13.5	8	15.2
MPE - CS Random	9	13.9	9	15.6
MPE - CS Random	10	14.3	10	16.0
MPE - CS Random	11	14.7	11	16.4
MPE - CS Random	12	15.0	12	16.8
MPE - CS Random	13	15.4	13	17.2

FIGURE 5.6.2 Average Seg-SNR obtained from a conventional MPE coder (Method MS5), and a CS-MPE coder that employs an Optimised (Trained) or a Random Position-Codebook. The number of pulses defined in each MPE frame is the same for all the coders, and the number of bits (*) required for the coding of the pulse positions in each case is shown.

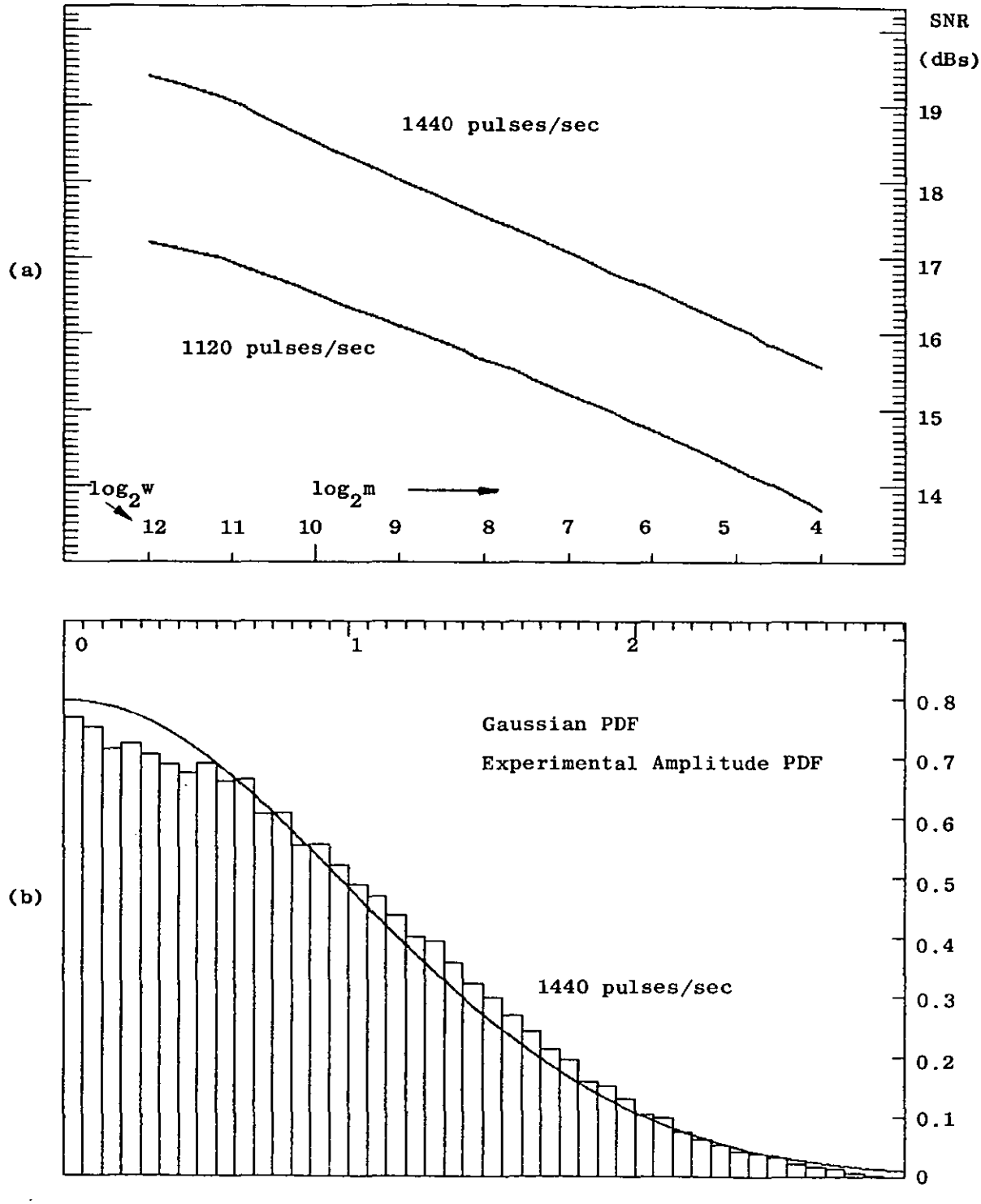


FIGURE 5.6.3 (a) Reduction of the average Seg-SNR during the Codebook Optimisation process, as the size *m* of the Codebook is reduced. Two pulse rates are considered i.e. 1120 and 1440 pulses/sec.
(b) Experimental PDF for the (absolute) pulse amplitudes (histogram) at a pulse rate of 1440 pulses/sec, and the Gaussian PDF with a second moment of one.

pulse amplitudes is calculated every two (at 8 and 9.6 kbits/sec) or three frames (at 16 kbits/sec), and quantized using a 32 level logarithmic quantizer. The pulse amplitudes are first normalised by the RMS value and then quantized with a Gaussian PDF optimised quantizer.

In the respective columns, the values of the transmission bit rate, the MPE frame size, the number of pulses per frame, the number of bits required for the quantization of the LPC parameters (log area-ratios), the number of bits allocated to each quantized amplitude value and the size of the position codebook, are given.

Bit Rate bits/sec	*Frame (samples)*	*Pulses (q)*	*LPC (bits)*	*Amplit. (bits)*	*Codebook (bits)*	*SNR (dBs)*
8000	*50*	*6*	*50*	*4*	*11*	*12.7*
8000	*50*	*6*	*46*	*4*	*12*	*13.0*
9600	*50*	*8*	*50*	*4*	*13*	*14.5*
16000	*40*	*11*	*50*	*5*	*13*	*19.8*

The position codebooks used by the CS-MPE coder were random, because it is difficult to generate optimised codebooks of the same size. Even so, the SNR figures are higher than the corrsponding figures given in Chapter 4 for the conventional MPE coders, by approximately 1.5 dBs. The SNR values are expected to increase, when larger frames and larger codebooks are employed by the CS-MPE coder. This increase can be observed at 8000 bits/sec, when a 12-bit position codebook is used instead of an 11-bit codebook. Obviously, optimised codebooks will further improve the performance of the CS-MPE coder.

5.7 Conclusions

A MPE-LPC scheme which employs a codebook for the pulse positions, has been proposed. The coder operates as a vector quantizer, using an Analysis by Synthesis optimisation loop to search through the parameter space of the pulse positions and minimise the distortion intro- duced by the coding process. The codebook can be unstructured or it can be arranged in the form of a tree codebook or a multiple codebook. A compu- tationally efficient optimisation algorithm has been described, which can be used to design an optimised unstructured codebook, based on a training process which takes

into account the average speech characteristics. The codebook optimisation algorithm can be modified and applied to structured codebooks.

The CS-MPE coder can operate at lower bit rates than conventional MPE coders, because it allocates fewer bits to the coding of the pulse positions without causing any loss in speech quality. Savings of 2-3 kbits/sec can be achieved with an optimised or even a random pulse position codebook.

Alternatively, The CS-MPE coder performs better than conventional MPE coders, when operating at the same transmission bit rate. This is true even when random codebooks are employed. The complexity of the MPE optimisation algorithm (when an unstructured position codebook is employed) though, may be higher than in conventional MPE schemes. The performance of the CS-MPE coder improves when larger MPE frames and codebooks are used.

Larger position codebooks must be used when the size of the MPE frame is increased, in order to cope with the increasing number of possible pulse position combinations. To avoid the considerable increase of the algorithm's computational complexity, tree or multiple codebooks must be used. Sometimes though, use of a small frame size by a MPE coder can be advantageous, as it is for example when a Long Term Predictor (LTP) is employed and optimised by minimising the distortion introduced by the coder. In this case, small frames improve the performance of the LTP and therefore the MPE coder can take advantage of the efficiency of the Codebook Search optimisation process and reduce the number of bits required for the quantization of the pulse positions.

The codebook optimisation algorithm described is quite general and does not depend on the availability of convenient quadratic error measures. It can therefore be employed in a number of general speech coding applications, where the speech waveform is decomposed into a number of discrete components [5.19,5.20,5.21,5.22], or in other related discrete optimisation problems.

REFERENCES

[5.1] M.R.Schroeder, B.S.Atal, "Code-Excited Linear Prediction (CELP): High quality speech at very Low Bit Rates", Proc ICASSP 1985, pp 937

[5.2] R.C.Rose, T.P.Barnwell, "The Self Excited Vocoder - An alternative approach to Toll Quality at 4800 bps",IEEE Proc ICASSP, Tokyo 1986, pp 453-456

[5.3] H.Koyama, A.Gersho, "Fully Vector-Quantized Multipulse LPC at 4800 BPS", Proc ICASSP 1986, pp 445

[5.4] V.Savvides, C.Xydeas, "A new approach to Low Bit Rate Speech Coding", Proc Int. Conf. Digital Processing of Signals in Communications, IERE, Loughborough, Sep 1988

[5.5] S.Ono, K.Ozawa, "2.4 kbps Pitch Prediction Multi-Pulse speech coding" Proc ICASSP 1988, pp 175

[5.6] R.Garcia-Gomez et al, "Vector Quantized Multipulse LPC", Proc ICASSP 1987, pp 2197

[5.7] P.Kroon, B.S.Atal, "Quantization Procedures for the Excitation in CELP coders", Proc ICASSP 1987, pp 1649

[5.8] P.Kroon, E.F.Deprettere, "A Class of Analysis-by-Synthesis Predictive Coders for High Quality speech coding at rates between 4.8 and 16 kbits/sec", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 88, pp 353-363

[5.9] M.A.Ireton, C.S.Xydeas, "On improving Vector Excitation Coders through the use of Spherical Lattice Codebooks (SLC's)", to be presented at IEEE Int. Conf. ICASSP, May 1989

[5.10] N.Gouvianakis, "Report on Low Bit Rate Speech Coding", British Telecom Progress Report, April 1988

[5.11] T.F.Quatieri, R.J.McAulay, "Speech Transformations based on a Sinusoidal Representation", IEEE Trans ASSP, Dec 1986, pp 1449-1464

[5.12] E.B.George, M.J.T.Smith, "A new speech coding model based on a Least-Squares Sinusoidal representation", Proc ICASSP 1987, pp 1641

[5.13] R.McAulay, T.F.Quatieri, "Computationally efficient Sine-Wave Synthesis and its application to Sinusoidal Transform Coding", Proc ICASSP 1988, pp 370

[5.14] Y.Lee, H.F.Silverman, "On a General Time-Varying model for speech signals", Proc ICASSP 1988, pp 95

[5.15] D.L.Thomson, "Parametric Models of the Magnitude/Phase Spectrum for Harmonic speech coding", Proc ICASSP 1988, pp 378

[5.16] P.Kroon et al, "Regular-Pulse Excitation: A novel approach to effective and efficient Multipulse Coding of Speech", IEEE Trans ASSP Oct 1986, pp 1054-1063

[5.17] R.S.Garfinkel, G.L.Nemhauser, "Integer Programming", John Wiley & Sons, New York, 1972

[5.18] L.R.Foulds, "Optimization Techniques: An Introduction", Springer-Verlag, New York, 1981

[5.19] S.Adlersberg, V.Cuperman, "Transform domain Vector Quantization for speech signals", Proc ICASSP 1987, pp 1938

[5.20] T.V.Sreenivas, "Modelling LPC-Residue by components for good quality speech coding", IEEE Proc. ICASSP, New York 1988, pp 171

[5.21] C.d'Alessandro, J.Lienard, "Decomposition of the speech signal into short-time waveforms using spectral segmentation", IEEE Proc. ICASSP, New York 1988, pp 351

[5.22] K.Tamaribuchi, S.Saito, "A New Analysis Method for Acoustic Signals composed of Sine Waves", IEEE Proc. ICASSP, New York 1988, pp 2440

CHAPTER 6

BACKWARD EXCITATION RECOVERY CODING

6.1 Introduction

Speech coding systems which combine the source-filter model of speech production and the concept of Analysis by Synthesis optimisation, have proved to be capable of providing near toll-quality speech at transmission rates well below 16 kbits/sec. There are numerous applications for medium and low bit-rate speech coders which require high quality speech, for example in satellite and mobile communication networks, voice storage and mail, integrated services networks, etc. These applications tend to multiply, as easily implementable coding methods, capable of producing very good quality speech at even lower transmission bit-rates, become available.

The Multipulse Excited (MPE) and Codebook Excited LPC (CELP) [6.1,6.2] coders have partially fulfilled the expectations of very good quality speech at medium and low bit rates. Both schemes employ a synthesis filter which usually consists of two linear autoregressive filters in cascade. The first filter takes the form of a Long Term Predictor (LTP) and models the fine spectral structure of speech, while the second filter is based on a Short Term Predictor (STP) and models the short term spectral envelope of speech. The parameters of both filters are defined periodically from the input speech samples. This is ususally achieved outside the main Analysis by Synthesis optimisation loop, by a two-stage minimisation of the energy of the residual signal produced by filtering the input speech through the inverse of the synthesis filter. Both systems transmit information, which is used by the decoder to reconstruct the excitation and the adaptive filter "components" of the coder.

An alternative Analysis by Sythesis coding algorithm can be obtained by defining the excitation signal from past information which is available at both the encoder and the decoder, and by determining the filter parameters from an Analysis by Sythesis optimisation process which minimises an appropriately chosen error measure. This general coding approach will be referred to as Backward Excitation Recovery (BER) coding [6.3,6.4].

BER coders operate at bit rates between 4 and 8 kbits/sec, and the quality

of speech production varies from good, at 4.8 kbits/sec, to very good communications quality at 8 kbits/sec. Furthermore, a BER coder may employ optimisation techniques which require speech frames of the order of 2-3 ms, thus minimising the encoding delay (compared to conventional LPC coders operating at low bits rates). This low delay property can be an advantage in transmission applications like satellite mobile radio, where the overall transmission delay must be minimised.

The theory of the BER systems will be presented for the general case of a multi-input linear synthesis filter. Two different models will be used for the estimation of the filter coefficients, one of them relies on Linear Prediction theory whereas the other one is based on the estimation and minimisation of the distortion introduced by the coding process.

A number of recursive adaptation methods will be presented, which can be employed to define the excitation signals, based on the past information available at both the encoder and the decoder. Finally, vector quantization of the synthesis filter coefficients will be considered, and a quantizer optimisation method will be described which is based on the minimisation of the average distortion introduced by the coding process.

6.2 Operation of the BER coder

The flow diagram of the BER encoder is shown in Fig 6.2.1. The encoder forms the excitation signals which drive the synthesis filter, using stored information which was generated during the operation of the BER coder in the past. The same information is available at the decoder, and thus to recover the synthesised speech waveform, only the parameters of the synthesis filter need to be transmitted.

The excitation sequences drive a multi-input synthesis filter. An example of a 3-input synthesis filter driven by 3 different excitation sequences, is shown in Fig 6.2.2(b). The first section of the synthesis filter consists of a series of FIR filters $B_j(z)$ whose outputs are combined and fed to the IIR filter $A(z)$. Each FIR filter takes the form :

$$B_j(z) = \sum_{k=0}^{q_j} b_k^j \, z^{-k-n-d_j} \quad , \quad j=1,2,\ldots,n_b \quad , \quad d_j \geq 0 \qquad (Eq\ 6.2.1)$$

where n is the number of samples in each speech frame, n_b is the number of FIR filters included in the synthesis filter, b_k^j are the coefficients of the jth FIR filter (q_j+1 coefficients are allocated to each filter), and d_j is a positive number which specifies a time delay (for each FIR filter) and will be referred to as a delay parameter.

The second part of the synthesis filter is the autoregressive (all pole) filter :

$$A(z) = \frac{1}{1 - \sum_{m=1}^{l} a_m z^{-m}} \qquad (Eq\ 6.2.2)$$

where l is the order of the filter.

The past samples of the excitation sequences $\{u_j(i)\}$ are assumed to be available at both the encoder and the decoder. These sequences are therefore specified only in the negative time direction :

$$\{u_j(i)\} \quad , \quad i<0 \quad , \quad j=1,2,\ldots,n_b \qquad (Eq\ 6.2.3)$$

Notice that some of these sequences may be identical, if the same excitation sequence is used at more than one inputs of the synthesis filter.

An Analysis by Synthesis optimisation procedure is used by the encoder to choose n-sample intervals from the past history of the excitation sequences, that could be used by the synthesis process to produce a synthesised version of the speech signal, which would be as close to the original signal as possible. The position of each interval (in time) is specified by the delay parameter d_j included in the definition of the synthesis filter. The Analysis by Synthesis procedure optimises the value of each delay parameter and thus chooses the "best" intervals from the stored samples of the excitation sequences.

The Analysis by Synthesis optimisation loop minimises the approximation error and chooses the optimum filter coefficients, which are then quantized and transmitted to the decoder. The optimum quantized filter coefficients are also employed by a local decoder which calculates the synthesised version of the speech signal and updates the history of the excitation sequences, before the next speech frame is processed.

The updating of the excitation sequences is performed identically by both

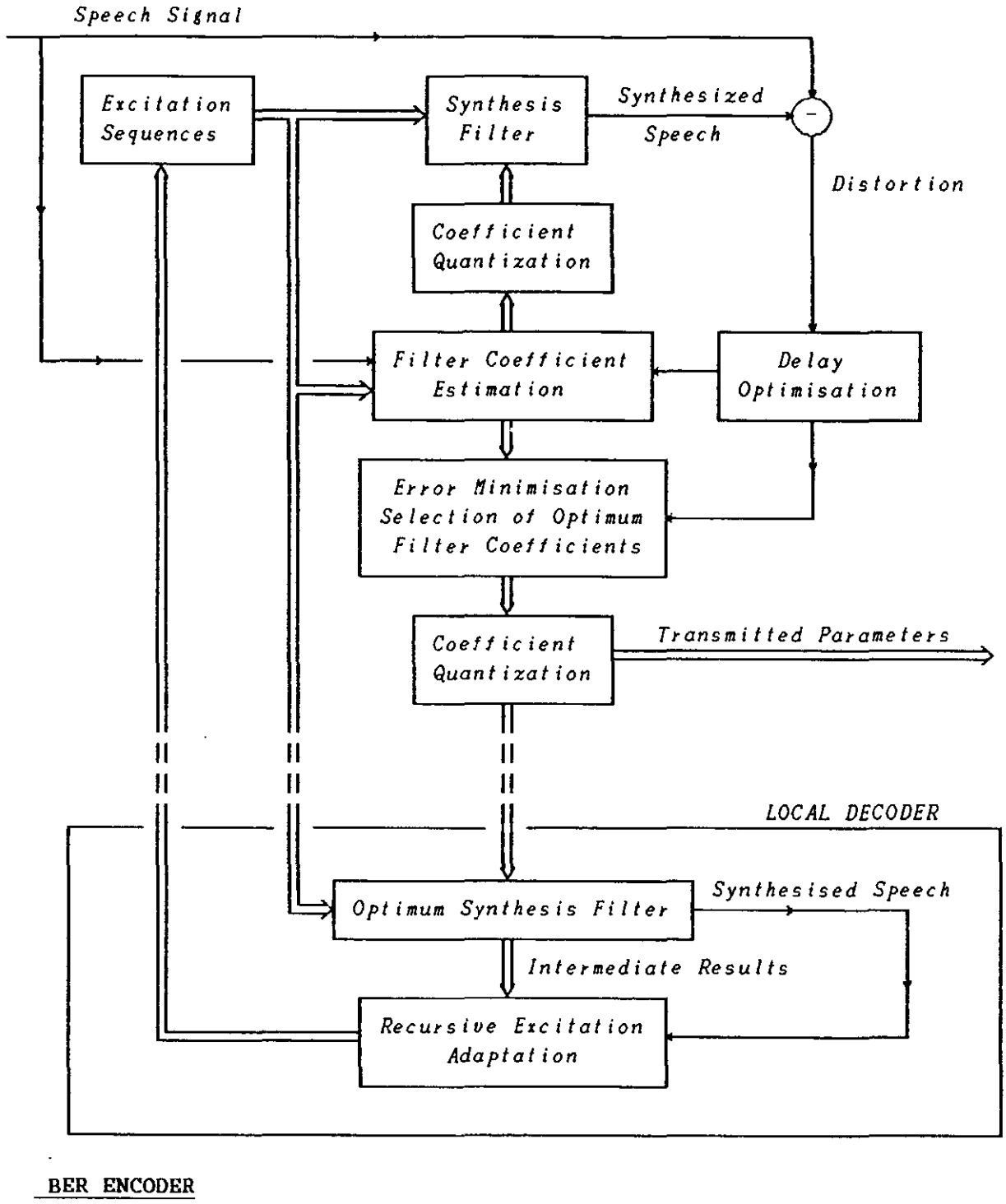
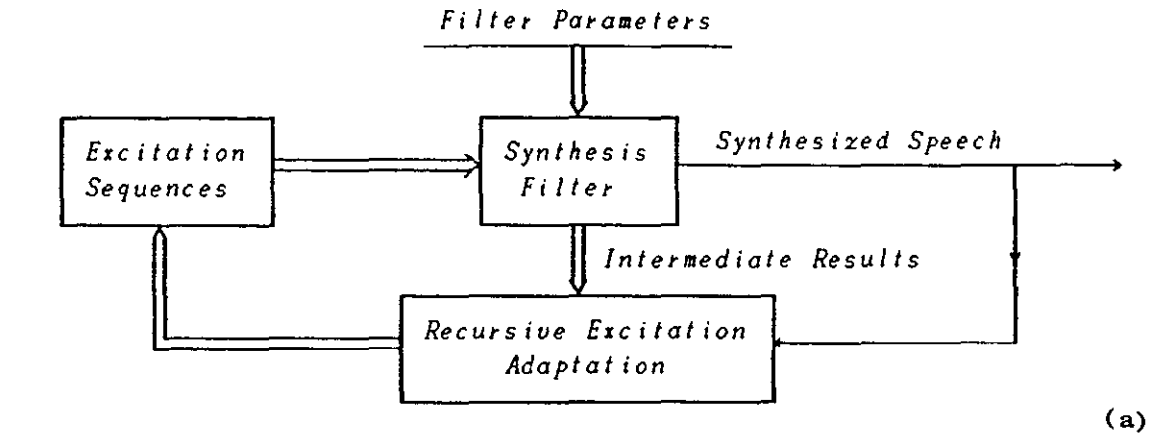
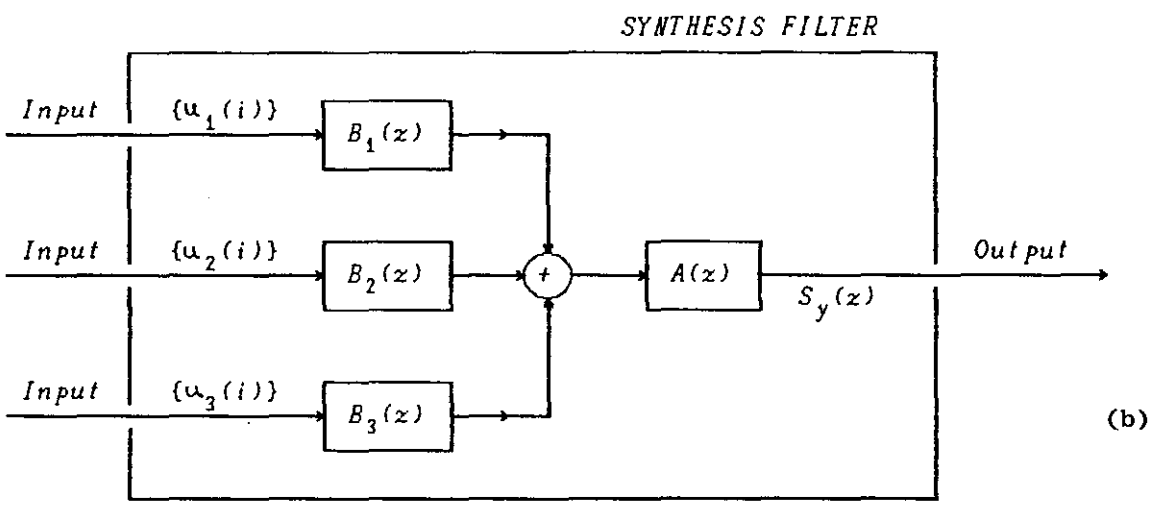


FIGURE 6.2.1 The Encoder section of a BER coder.



(a)

BER DECODER



(b)

FIGURE 6.2.2 (a) The Decoder section of a BER coder.
(b) The general form of the Synthesis Filter employed by the BER coder.

the encoder and the decoder, without requiring the transmission of side information. It must therefore be done using any "intermediate results" that have been produced during the coders' operation. These results include the various signals at the outputs of the FIR filters and the synthesised speech signal itself. This recursive excitation adaptation process can be detected in Fig 6.2.1 by tracing the double excitation lines. The recursive adaptation algorithm employed by the local decoder, provides the link between the past and the present values of the excitation signals, and closes the excitation adaptation loop.

The decoder, shown in Fig 6.2.2(a), operates in the same way as the local decoder of the encoder (Fig 6.2.1) in forming the synthesised version of the speech signal, provided of course that the filter parameters are received free of channel errors. In general, the system has a tendency to propagate transmission errors, as indeed is the case with any backward adaptive speech coding system. Nevertheless, standard "initialisation" techniques can be easily incorporated into the system to minimise the effect of channel errors on the quality of the recovered speech signal.

The estimation of the synthesis filter coefficients (not including the delay parameters) will be examined first. Two methods will be considered, the first one employing the ARX Linear Prediction model, which enables the estimation of all the synthesis filter coefficients by minimising the energy of the prediction error. The second method optimises the coefficients of the $B_j(z)$ FIR filters, by minimising the energy of the distortion introduced by the coding process. Different strategies will be explored, for the optimisation of the delay parameters, and a number of recursive excitation adaptation schemes will also be considered.

6.3 Filter Coefficient Estimation using Linear Prediction

The synthesis filter model used by the BER coder (shown in Fig 6.2.2(b)) differs from the synthesis models employed by conventional LPC speech coders in that it can accept many inputs and both the input sequences (past excitation samples) and the desired response of the filter (speech signal) are known in advance. The estimation of the filter coefficients can be treated as a System Identification problem and a suitable Linear Prediction model that can be used for the BER synthesis filter, is the Autoregressive with Exogenous inputs (ARX) [6.5,6.6,6.7,6.8]. The same model (with a single

input) has been employed by MPE coding schemes that redefine the AR-LPC filter, once an estimate of the Multipulse Excitation has been formed [6.9,6.10]. The ARX model for the BER coder can be written as :

$$s(i) = \sum_{m=1}^{l} a_m\, s(i-m) + \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j\, u_j(i-n-d_j-k) + e_p(i) \quad , \quad o \le i \le n-1 \qquad (Eq\ 6.3.1)$$

where $\{s(i)\}$ is the sequence of n speech samples contained in the speech frame examined, $\{e_p(i)\}$ is the innovation sequence (prediction error) whose samples are assumed to be uncorrelated and distributed as zero-mean Gaussian variables, and the delay parameters d_j are assumed to be fixed. Under these conditions, the Maximum Likelihood (ML) estimate and the Least Squares (LS) estimate of the filter coefficients are equivalent. The LS estimate of the coefficients is obtained by minimising the energy of the prediction error over the n-sample speech interval (Eq 6.3.1). In matrix notation, Eq 6.3.1 is transformed to :

$$\mathbf{s} = \mathbf{S}\,\mathbf{a} + \sum_{j=1}^{n_b} \mathbf{U}_j\, \mathbf{b}_j + \mathbf{e}_p \qquad (Eq\ 6.3.2)$$

where $\mathbf{s}$ and $\mathbf{e}_p$ are the n-dimensional vectors containing the speech and prediction error samples, the vector $\mathbf{a}=[a_1,a_2,\ldots,a_l]$ contains the coefficients of the filter $A(z)$, $\mathbf{b}_j=[b_o^j,b_1^j,\ldots,b_{q_j}^j]$ contains the coefficients of the filter $B_j(z)$, $\mathbf{S}$ is the $n \times l$ speech matrix :

$$\mathbf{S} = \begin{bmatrix} s(-1) & s(-2) & \ldots & s(-l) \\ s(o) & s(-1) & \ldots & s(1-l) \\ \vdots & \vdots & & \vdots \\ s(n-2) & s(n-3) & \ldots & s(n-1-l) \end{bmatrix} \qquad (Eq\ 6.3.3)$$

and $\mathbf{U}_j$ are the $(n) \times (q_j+1)$ excitation matrices :

$$\mathbf{U}_j = \begin{bmatrix} u_j(-n-d_j) & u_j(-n-d_j-1) & \ldots & u_j(-n-d_j-q_j) \\ u_j(-n-d_j+1) & u_j(-n-d_j) & \ldots & u_j(-n-d_j-q_j+1) \\ \vdots & \vdots & & \vdots \\ u_j(-d_j-1) & u_j(-d_j-2) & \ldots & u_j(-d_j-q_j-1) \end{bmatrix} \quad , \; j=1,\ldots,n_b \qquad (Eq\ 6.3.4)$$

The coefficients of the BER synthesis filter can be estimated by expressing the variance of the prediction error as a quadratic function of the

required coefficients and by locating the function minimum. This minimum generally exists and is unique. Notice that either all or some of the filter coefficients may be estimated in this way. Assuming for the moment that all the filter coefficients are required, the LS estimate of their values can be found by solving a system of $(1+q_1+\ldots+q_{n_b})$ equations with the same number of unknowns. The estimated coefficient values are then :

$$\begin{bmatrix} \mathbf{a} \\ \hline \mathbf{b}_1 \\ \hline \vdots \\ \hline \mathbf{b}_{n_b} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^T\mathbf{S} & \mathbf{S}^T\mathbf{X}_1 & \ldots & \mathbf{S}^T\mathbf{X}_{n_b} \\ \hline \mathbf{X}_1^T\mathbf{S} & \mathbf{X}_1^T\mathbf{X}_1 & \ldots & \mathbf{X}_1^T\mathbf{X}_{n_b} \\ \hline \vdots & \vdots & & \vdots \\ \hline \mathbf{X}_{n_b}^T\mathbf{S} & \mathbf{X}_{n_b}^T\mathbf{X}_1 & \ldots & \mathbf{X}_{n_b}^T\mathbf{X}_{n_b} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{S}^T \\ \hline \mathbf{X}_1^T \\ \hline \vdots \\ \hline \mathbf{X}_{n_b}^T \end{bmatrix} \mathbf{s} \qquad \textit{(Eq 6.3.5)}$$

The coefficient matrix of Eq 6.3.5 is symmetric and contains auto-correlation and cross-correlation terms from the speech and excitation sequences. In most cases, these terms are related and can be calculated using computationally efficient algorithms (see Section 4.4).

If only some of the filter coefficients need to be estimated, Eq 6.3.2 can be rearranged so that the left hand side contains the coefficients whose values have been obtained beforehand and are assumed to be constant, and the right hand side contains the coefficients whose values are required. For example, if only the last (n_q-k+1) coefficient sets $\mathbf{b}_k,\mathbf{b}_{k+1},\ldots,\mathbf{b}_{n_b}$ are required and $\mathbf{a},\mathbf{b}_1,\mathbf{b}_2,\ldots,\mathbf{b}_{k-1}$ are known, Eq 6.3.2 is rearranged as :

$$\mathbf{s} - \mathbf{S}\,\mathbf{a} - \sum_{j=1}^{k-1} \mathbf{U}_j\,\mathbf{b}_j = \sum_{j=k}^{n_b} \mathbf{U}_j\,\mathbf{b}_j + \mathbf{e}_p \qquad \textit{(Eq 6.3.2a)}$$

The same procedure as before is followed to minimise the variance of the prediction error $\mathbf{e}_p$ in Eq 6.3.2a, with respect to the required coefficients. The optimum values of the coefficients are now :

$$\begin{bmatrix} \mathbf{b}_k \\ \hline \vdots \\ \hline \mathbf{b}_{n_b} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_k^T\mathbf{X}_k & \ldots & \mathbf{X}_k^T\mathbf{X}_{n_b} \\ \hline \vdots & & \vdots \\ \hline \mathbf{X}_{n_b}^T\mathbf{X}_k & \ldots & \mathbf{X}_{n_b}^T\mathbf{X}_{n_b} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}_k^T \\ \hline \vdots \\ \hline \mathbf{X}_{n_b}^T \end{bmatrix} \left(\mathbf{s} - \mathbf{S}\mathbf{a} - \sum_{j=1}^{k-1} \mathbf{U}_j\mathbf{b}_j\right) \qquad \textit{(Eq 6.3.5a)}$$

The process of estimating part of the synthesis filter at a time, is used to simplify the BER filter optimisation algorithm, so that one section of the synthesis filter can be optimised first, then another section, and so on. Notice that if only the coefficients of $A(z)$ are required, and the coefficiennts of the $B_j(z)$ filters have zero values, then the estimate given by Eq 6.3.5 will be the same as the one obtained from the Covariance LPC estimation method.

The optimisation of the sythesis filter at the encoder, is performed in a series of steps. At each step, a different "setup" of the synthesis filter is tested. The filter "setup" changes by increasing the value of the delay parameters, or by adding more FIR filters. As the maximum number n_b of FIR filter-sections is predefined, and the range of the delay variables is known, only a limited number of possible filter "setups" can be defined. All the different filter "setups" are generated and examined at the encoder, by the Analysis by Synthesis optimisation process (Fig 6.2.1). For each "setup", a set of filter coefficients is calculated by solving the corresponding system of linear equations (Eq 6.3.5). The speech signal is synthesised for every set of (quantized) filter coefficients, using the recursive formula :

$$s_y(i) = \sum_{m=1}^{l} a_m \, s_y(i-m) + \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \, u_j(i-n-d_j-k) \quad , \quad o \le i \le n-1 \qquad (Eq\ 6.3.6)$$

where $\{s_y(i)\}$ is the synthesised speech signal and $\{u_j(i)\}$, $j=1,2,\ldots,n_b$ are the excitation sequences. The error signal (distortion) is measured by the difference between the original and synthesised speech signals, and its energy is calculated for each set of filter coefficients, produced by the optimisation process. The set of coefficients which minimise the error energy are chosen by the error minimisation algorithm (Fig 6.2.1) and transmitted to the decoder.

The all pole filter $A(z)$ estimated from Eq 6.3.5, is not guaranteed to be minimum phase (have all its poles inside the unit circle), and may cause the sythesis process (Eq 6.3.6) to become unstable. This happens very rarely during the optimisation of the synthesis filter, and can be simply detected by noticing the diverging values of the synthesised signal.

An effective remedy would be to disregard the filter "setups" which result

unstable synthesis filters. This would only cause a small disturbance to the filter optimisation process. Alternatively, a different filter estimation procedure can be used, which rearranges the synthesis filter of the BER coder into an ARMA-Lattice structure, by embedding both the output and input signals into a conventional Autoregressive-Lattice model [6.11,6.12,6.13, 6.14,6.15,6.16]. The error minimised would then be a function of the forward and backward prediction errors, and the filter $A(z)$ would be guaranteed to be minimum phase. The former approach will be used here, because of its simplicity and because it can give results which are as good as the results obtained using the latter approach.

6.4 Estimating the Filter Coefficients by Minimising the Signal Distortion

The synthesis filter estimation method described in Section 6.3, defines the filter coefficients by minimising the prediction error, and selects the best filter "setup" by minimising the distortion introduced by the coding process. Alternatively, for a given filter "setup" it is possible to estimate the coefficients of the synthesis filter, by minimising the signal distortion instead of the prediction error.

The energy of the error signal (distortion) can be expressed as a quadratic function of the filter coefficients b_k^j. The minimum of this function, and therefore the optimum values of the b_k^j coefficients, can be found using standard Least-Squares methods. It is not possible however, to use the same technique to estimate the coefficients of the all-pole filter $A(z)$, because the error energy function would be a higher order polynomial and numerical methods would need to be employed in order to locate the function minimum.

The error signal is the difference between the original and synthesised speech waveforms :

$$E_a(z) = S(z) - S_y(z) \qquad (Eq\ 6.4.1)$$

and the synthesised signal can be calculated using the difference equation 6.3.6. By taking the one-sided z-transform [6.17] of both sides of Eq 6.3.6 and rearranging the terms (see Appendix A), the synthesised signal becomes :

$$S_y(z) = A(z) \sum_{j=1}^{n_b} U_j(z)\ B_j(z) + A(z) \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n+d_j+k-1} u_j(i-n-d_j-k)\ z^{-i} + M_y(z)$$

(Eq 6.4.2)

where :

$$M_y(z) = A(z) \sum_{k=o}^{l-1} z^{-k} \sum_{m=1}^{l-k} a_{k+m}\ s_y(-m) \qquad (Eq\ 6.4.3)$$

The first term of the right hand side of Eq 6.4.2 corresponds to the forced response of the synthesis filter, while the second and third terms correspond to its transient response. The term $M_y(z)$ is the transient response of the all-pole filter $A(z)$. Using Eq 6.2.1, the forced response can be written as :

$$FORCED\ RESPONSE = A(z) \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j\ U_j(z)\ z^{-n-d_j-k} \qquad (Eq\ 6.4.4)$$

The samples of the sequence given in Eq 6.4.4, all lie outside the $[o,n-1]$ interval of the speech frame and can therefore be ignored. Thus Eq 6.4.2 can be simplified to :

$$S_y(z) = A(z) \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n-1} u_j(i-n-d_j-k)\ z^{-i} + M_y(z) \qquad (Eq\ 6.4.5)$$

Using matrix notation, Eq 6.4.5 can be converted to :

$$\mathbf{s}_y = \mathbf{Q} \sum_{j=1}^{n_b} \mathbf{U}_j \mathbf{b}_j + \mathbf{m}_y \qquad (Eq\ 6.4.6)$$

where $\mathbf{Q}$ is the nxn lower triangular convolution matrix :

$$\mathbf{Q} = \begin{bmatrix} h(o) & o & o & \dots & o \\ h(1) & h(o) & o & \dots & o \\ h(2) & h(1) & h(o) & \dots & o \\ \vdots & \vdots & & & \vdots \\ h(n-1) & h(n-2) & h(n-3) & \dots & h(o) \end{bmatrix} \qquad (Eq\ 6.4.7)$$

$\{h(i)\}$ is the impulse response of the all-pole filter $A(z)$ and the matrices

$\mathbf{U}_j$ were defined by Eq 6.3.4. Using Eqs 6.4.1 and 6.4.6, the error signal (distortion) can be expressed as a function of the b_k^j coefficients :

$$\mathbf{e}_a = \mathbf{s} - \mathbf{m}_y - \sum_{j=1}^{n_b} \mathbf{Q}\,\mathbf{U}_j \mathbf{b}_j \qquad \textit{(Eq 6.4.8)}$$

Assuming that the values of the filter delay parameters are fixed and that an estimate of the coefficients of the filter $A(z)$ is already available, the filter coefficients b_k^j can be calculated using standard Least-Squares methods. If, at a certain stage during the synthesis filter optimisation process, all the b_k^j ,$k=0,1,\ldots,q_j$,$j=1,2,\ldots,n_b$ coefficients were required, their optimum values would be :

$$\begin{bmatrix} \mathbf{b}_1 \\ \hline \mathbf{b}_2 \\ \hline \vdots \\ \hline \mathbf{b}_{n_b} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_1 & \mathbf{X}_1^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_2 & \cdots & \mathbf{X}_1^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_{n_b} \\ \hline \mathbf{X}_2^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_1 & \mathbf{X}_2^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_2 & \cdots & \mathbf{X}_2^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_{n_b} \\ \hline \vdots & \vdots & & \vdots \\ \hline \mathbf{X}_{n_b}^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_1 & \mathbf{X}_{n_b}^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_2 & \cdots & \mathbf{X}_{n_b}^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_{n_b} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}_1^T\mathbf{Q}^T \\ \hline \mathbf{X}_2^T\mathbf{Q}^T \\ \hline \vdots \\ \hline \mathbf{X}_{n_b}^T\mathbf{Q}^T \end{bmatrix} (\mathbf{s}-\mathbf{m}_y)$$

(Eq 6.4.9)

As in the case of the prediction error minimisation equations (Eq 6.3.5), the coefficient matrix of Eq 6.4.9 contains autocorrelation and cross-correlation terms which are generally related. These relationships can be exploited to simplify considerably the computation of the coefficient matrix and subsequently the BER optimisation algorithm.

The energy of the error signal (Eq 6.4.8) is calculated (for the quantized filter coefficients) using an augmented coefficient matrix, and the computationally efficient formula (see Appendix B) :

$$\mathbf{e}_a^T\,\mathbf{e}_a = \begin{bmatrix} 1 \\ \hline -\mathbf{b}_1 \\ \hline \vdots \\ \hline -\mathbf{b}_{n_b} \end{bmatrix}^T \begin{bmatrix} (\mathbf{s}-\mathbf{m}_y)^T(\mathbf{s}-\mathbf{m}_y) & (\mathbf{s}-\mathbf{m}_y)^T\mathbf{Q}\,\mathbf{X}_1 & \cdots & (\mathbf{s}-\mathbf{m}_y)^T\mathbf{Q}\,\mathbf{X}_{n_b} \\ \hline \mathbf{X}_1^T\mathbf{Q}^T(\mathbf{s}-\mathbf{m}_y) & \mathbf{X}_1^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_1 & \cdots & \mathbf{X}_1^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_{n_b} \\ \hline \vdots & \vdots & & \vdots \\ \hline \mathbf{X}_{n_b}^T\mathbf{Q}^T(\mathbf{s}-\mathbf{m}_y) & \mathbf{X}_{n_b}^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_1 & \cdots & \mathbf{X}_{n_b}^T\mathbf{Q}^T\mathbf{Q}\,\mathbf{X}_{n_b} \end{bmatrix} \begin{bmatrix} 1 \\ \hline -\mathbf{b}_1 \\ \hline \vdots \\ \hline -\mathbf{b}_{n_b} \end{bmatrix}$$

(Eq 6.4.10)

The error energy is calculated for every possible "setup" of the synthesis filter examined by the Analysis-by-Synthesis optimisation procedure (see Fig 6.2.1). The error minimisation algorithm finally chooses the set of coefficients that correspond to the smallest value of the error energy. These coefficients are then transmitted to the decoder.

As in the case when the filter coefficients are estimated using Linear Prediction methods (Section 6.3), only some of the filter coefficients may be required at some stage during the synthesis filter optimisation process. Equation 6.4.6 can be rearranged so that its left hand side contains all the coefficients whose values have been determined at a previous optimisation stage, and its right hand side contains the coefficients which need to be estimated. The error energy would then be minimised with respect to the required coefficients, by solving the corresponding system of linear equations (Eq 6.4.9).

The Distortion-Minimisation (DM) Filter-Estimation algorithm described in this Section, can be combined with the Linear-Prediction (LP) estimation algorithm of Section 6.3, if the latter algorithm is employed to provide an estimate of the coefficients of the all-pole filter $A(z)$. Alternatively, an estimate of the filter $A(z)$ can be obtained (and subsequently used by the DM algorithm) using conventional LPC methods (eg. the Covariance or the Maximum Entropy filter estimation methods)

The DM algorithm can be modified to permit negative values for the delay parameters $(d_j(o)$. In this case the forced response of the synthesis filter (Eq 6.4.4) cannot be ignored, and an estimate of the excitation sequences in the positive time direction must be formed by artificially extending the excitation sequences using the past excitation samples [6.18,6.19]. The same modification can be applied to the LP estimation algorithm of Section 6.3.

The DM algorithm can be simplified by limiting the effective duration of the impulse response $h_o, h_1, h_2, \ldots$ [6.18], or by choosing the "setup" of the synthesis filter in some other way (using for example the LP method) and then using the DM estimation method only in the final filter optimisation stage, to obtain all the b_k^j filter coefficients.

An interesting relationship between the error signal (distortion) and the prediction error (Eq 6.3.1), can be formed by taking the z-transform of both

sides of Eq 6.3.1. By rearranging the terms, an equation similar to Eq 6.4.2 can be obtained :

$$S(z) = A(z) \sum U_j(z)\ B_j(z) + A(z) \sum_{j=1}^{n_b} \sum_{k=0}^{q_j} b_k^j \sum_{i=0}^{n+d_j+k-1} u_j(i-n-d_j-k)\ z^{-i} + M(z) + E_p(z) \qquad (Eq\ 6.4.11)$$

where $E_p(z)$ is the prediction error, and :

$$M(z) = A(z) \sum_{k=0}^{l-1} z^{-k} \sum_{m=1}^{l-k} a_{k+m}\ s(-m) \qquad (Eq\ 6.4.12)$$

If the assumption is made that the last l samples of the synthesised and original speech waveform in the previous analysis frame were equal :

$$s(-m) = s_y(-m) \quad , \quad m=1,2,\ldots,l \qquad (Eq\ 6.4.13)$$

then, using equations 6.4.1, 6.4.2 and 6.4.11, a relationship between the two error signals can be established :

$$E_a(z) = E_p(z)\ A(z) \qquad (Eq\ 6.4.14)$$

It is clear that the two error sequences have quite different spectral distributions, and that the minimisation of the energy of one of them does not necessarily cause the minimisation of the energy of the other. The two sequences (and therefore the DM and LP estimation algorithms) become equivalent only when the all-pole filter $A(z)$ is removed from the synthesis filter (that is when $A(z)=1$).

6.5 Optimisation of the Filter Delay Parameters

The Analysis by Synthesis optimisation process employed by the BER encoder seeks to determine n-sample intervals from the stored sequences of past excitation samples, that can be used to drive a multi-input synthesis filter whose output is the reconstructed speech signal. The relative position of each interval (in time) is specified of a delay parameter d_j. Each delay parameter corresponds to a separate FIR section of the synthesis filter (Fig 6.2.2(b)) and may take any integer value in the interval $[o, n_d-1]$.

The values of the delay parameters are determined so that the energy of the distortion introduced by the coding process is minimised (Fig 6.2.1). For each set of delay values tested by the Analysis by Synthesis optimisation process, a set of filter coefficients $\mathbf{a}, \mathbf{b}_j$ can be found, using either the Linear Prediction (LP) or the Distortion Minimisation (DM) methods, described in Sections 6.3 and 6.4.

The synthesis filter coefficients is therefore optimised using analytical methods, whereas the delay parameters are optimised using numerical methods. As the number of FIR sections of the synthesis filter is usually not large, a sequential optimisation approach is followed to determine the values of the delay parameters.

The synthesis filter optimisation process determines the value of d_1 first, allowing only $A(z)$ and $B_1(z)$ to have non-zero coefficients. In the second stage of the optimisation process, the value of d_2 is determined assuming that the value of d_1 remains fixed and allowing only $A(z)$, $B_1(z)$ and $B_2(z)$ to have non-zero coefficients. The process continues until all the delay parameters, and therefore all the FIR filter sections, are determined.

At each stage a delay parameter is defined by first evaluating the energy of the signal distortion for every possible value of the delay parameter, and then choosing the delay value which results the minimum distortion energy. This simple optimisation strategy provides the common ground for the development of various filter optimisation algorithms. These algorithms are formed by choosing a different set of filter coefficients to be optimised at each stage ("setup"), or by changing the filter estimation algorithm (either the LP or the DM methods).

Possible optimisation algorithms for the Single-FIR-Section synthesis

filter are given in Fig 6.5.1(a). The first two algorithms $M_1(1)$ and $M_1(2)$ optimise the coefficients **a** and $\mathbf{b}_i$ jointly, using either the LP estimation method (Eq 6.3.5) or a combination of the LP and DM methods. When the two estimation methods are combined (to form method LP-DM), Eq 6.3.5 is employed to estimate the **a** and $\mathbf{b}_i$ coefficients, and Eqs 6.4.9 and 6.4.10 are used to reoptimise the $\mathbf{b}_i$ coefficients and measure the distortion energy (Eq 6.4.10) for every possible value of d_1 $(o \leq d_1 \leq n_d - 1)$. Notice that once the optimum value of the parameter d_1 is determined by method $M_1(1)$, it is possible to reoptimise the $\mathbf{b}_i$ coefficients using Eq 6.4.9. This possibility is indicated by the presence of the (R) symbol next to the coefficient estimation method.

Methods $M_1(3)$ and $M_1(4)$ assume that the coefficients of the filter $A(z)$ have been estimated separately using a conventional LPC method (Covariance or Maximum Entropy), and that they are kept constant during the optimisation of the parameter d_1. As a consequence, the frame over which $A(z)$ is defined can be different (and usually larger) than the frame over which the $\mathbf{b}_i$ coefficients are defined. Method $M_1(3)$ uses the LP method to estimate the set of coefficients $\mathbf{b}_i$, but may be modified to allow the reoptimisation (R) of the $\mathbf{b}_i$ coefficients from Eq 6.4.9, after the "optimum" value of the parameter d_1 has been determined.

As more FIR sections are added to the synthesis filter, the number of possible optimisation algorithms rapidly increases. In Fig 6.5.1(b), algorithms that can be employed to optimise a synthesis filter with two FIR sections, are described. The value of the parameter d_1 is optimised in the first stage, and d_2 is optimised in the second stage. As in the case of the Single-FIR-Section synthesis filter, three filter estimation algorithms are used i.e. the LP (Eq 6.3.5), the DM (Eq 6.4.9) and the LP-DM (Eqs 6.3.5 and 6.3.9) methods. Different sets of filter coefficients are chosen to be optimised at each stage, and some filter coefficients remain constant during the optimisation of the parameters d_1 or d_2.

In methods $M_2(1)$-$M_2(8)$ the coefficients of the filter $A(z)$ are estimated during the optimisation of the parameters d_1 or d_2, using the LP estimation method (Eq 6.3.5). In contrast, methods $M_2(9)$-$M_2(12)$ estimate the coefficients of the filter $A(z)$ separately, using a conventional LPC method and possibly a larger speech interval than the interval over which the $\mathbf{b}_1$ and $\mathbf{b}_2$ coefficients are defined. The **a** coefficients are then kept constant during

		OPTIMISATION OF d_1	
OPTIMISATION METHOD	COEFFICIENT ESTIMATION	CONSTANT COEFFICIENTS	VARIABLE COEFFICIENTS
$M_1(1)$	LP (R)		a b_1
$M_1(2)$	LP-DM		a b_1
$M_1(3)$	LP (R)	a	b_1
$M_1(4)$	DM	a	b_1

(a)

		d_1 OPTIMISATION		d_2 OPTIMISATION	
OPTIM. METHOD	COEFFICIENT ESTIMATION	CONSTANT COEFF.	VARIABLE COEFF.	CONSTANT COEFF.	VARIABLE COEFF.
$M_2(1)$	LP (R)		a b_1	a b_1	b_2
$M_2(2)$	LP (R)		a b_1	a	b_1 b_2
$M_2(3)$	LP (R)		a b_1	b_1	a b_2
$M_2(4)$	LP (R)		a b_1		a b_1 b_2
$M_2(5)$	LP-DM (R)		a b_1	a b_1	b_2
$M_2(6)$	LP-DM		a b_1	a	b_1 b_2
$M_2(7)$	LP-DM (R)		a b_1	b_1	a b_2
$M_2(8)$	LP-DM (R)		a b_1		a b_1 b_2
$M_2(9)$	LP (R)	a	b_1	a b_1	b_2
$M_2(10)$	LP (R)	a	b_1	a	b_1 b_2
$M_2(11)$	DM (R)	a	b_1	a b_1	b_2
$M_2(12)$	DM	a	b_1	a	b_1 b_2

(b)

FIGURE 6.5.1 Algorithms used to define the Filter Delay-Parameters when :
(a) the BER synthesis filter contains one FIR section
(b) the BER synthesis filter contains two FIR sections.

the optimisation of the parameters d_1 and d_2.

Notice that a further optimisation stage can be added to most methods, that reoptimises (R) the $\mathbf{b}_1$ and $\mathbf{b}_2$ coefficients (using Eq 6.4.9), after the "optimum" values of the parameters d_1 and d_2 have been determined.

When three FIR sections are included in the synthesis filter, the number of possible filter optimisation algorithms is even greater. In general, these filter optimisation algorithms can be applied to any BER coding scheme, irrespective of the type and number of excitation sequences used.

The filter optimisation algorithms can be simplified by using a subset of the synthesis filter coefficients (one or two coefficients for each FIR filter section) in order to determine the optimum values of the delay parameters, and finally estimating all the filter coefficients (using either the LP or the DM algorithms) for the optimised delay parameters.

The performance of the BER filter-optimisation algorithms can be improved by considering a group of consequtive speech frames, and by minimising the distortion over the entire speech interval, with respect to the filter parameters corresponding to each of the frames in the group.

6.6 Algorithms for the Recursive Adaptation of the Excitation Sequences

As the BER coder processes the speech signal, it goes through two different phases which can be considered separately. In the first phase, described in Section 6.5, the input to the synthesis filter is obtained from the stored sequences of past excitation samples $\{u_j(i)\}$, and the parameters of the synthesis filter are optimised in an Analysis by Synthesis optimisation loop. This operation is only performed by the encoder, and the estimated parameters (filter coefficients and delay parameters) are transmitted to the decoder.

Both the encoder and the decoder then change to an operational mode which is responsible for the adaptation of the excitation sequences and the recovery of the synthesised speech signal. In this second phase the encoder operates as a local decoder (Fig 6.2.1).

The adaptation of the excitation sequences is performed by extending them in the positive time direction, using as components the "time sequences" produced during the filter optimisation (intermediate results) which are

available to both the encoder and the decoder. The extension of the excitation sequences covers a single frame of n samples, and is added to the already stored excitation samples by effectively shifting the stored excitation sequences by n samples and appending the new samples.

There are many possible ways in which the different "time sequences" can be combined to form a suitable continuation of the excitation sequences in the positive time direction, and it is surprising how different the characteristics of the excitation signals can be, depending on the choice of the adaptation algorithm.

The synthesis filter of the BER coder may have many inputs (corresponding to the separate FIR sections), but the number of excitation sequences may be smaller, if two or more inputs are obtained from the same excitation sequence. Consider first the case of a multi-input synthesis filter and a single excitation sequence. A possible excitation adaptation algorithm can be constructed using the output of the "optimised" synthesis filter (see Eq 6.4.5) :

$$U_1(z) = A(z) \sum_{j=1}^{n_b} U_1(\mathbf{b}_j, d_j) + M_y(z) \qquad (Eq\ 6.6.1)$$

where $U_1(z)$ is the continuation of the sequence $\{u_1(i)\}$ in the positive time direction (one-sided z-transform), and for notation convenience the short description of the output of the jth FIR filter section, when the input is taken from the $\{u_m(i)\}$ excitation sequence is :

$$U_m(\mathbf{b}_j, d_j) = \sum_{k=0}^{q_j} b_k^j \sum_{i=0}^{n-1} u_m(i-n-d_j-k)\, z^{-i} \quad , \quad 1 \le m \le n_b \quad , \quad 1 \le j \le n_b \qquad (Eq\ 6.6.2)$$

It is understood that the filter parameters $\mathbf{b}_j, d_j$ used, were determined by the filter optimisation process during the first phase of the coder's operation. Since :

$$U_1(z) = S_y(z) \qquad (Eq\ 6.6.3)$$

the stored excitation samples are taken from the output of the synthesis filter and the excitation sequence $\{u_1(i)\}$ has speech-like characteristics.

In Fig 6.6.1(a), an expanded view of the synthesis filter is shown, just

before the speech synthesis and excitation adaptation operations take place. The parameters of the synthesis filter have been determined by the filter optimisation process. Two FIR filter sections are employed (corresponding to the delay parameters d_1 and d_2) and the excitation samples are stored in the filter delay line. By clocking the filter n times, n synthesised speech samples are recovered at the output. These samples enter the delay line and therefore update the values of the stored excitation sequence. The length of the delay line (number of past excitation samples kept) is equal to $n+n_d+max[q_j]$, where n_d-1 is the maximum value of the delay parameter and q_j+1 is the number of coefficients of the jth FIR filter section. Note that initially (when the BER coder starts operating) the excitation sequence held in the delay line of the encoder and the decoder is set to a preconstructed sequence of zero-mean Gaussian-distributed random samples.

A different excitation adaptation algorithm can be constructed using the combined outputs of the FIR sections of the synthesis filter. If only a single excitation source is used, then the adaptation algorithm is defined by the equation :

$$U_1(z) = \sum_{j=1}^{n_b} U_1(\mathbf{b}_j, d_j) \qquad (Eq\ 6.6.4)$$

If more than one excitation sequences are used, then the order with which the updating equations are applied is important. This is because the filter optimisation process determines the parameters of the $B_1(z)$ filter first, then those of $B_2(z)$, and so on. This order of preference puts different emphasis on each excitation sequence and thus the adaptation order also becomes significant.

When n_b excitation sequences are used, a possible adaptation strategy can be formed by combining the outputs of the FIR filter sections in the following way :

$$U_m(z) = \sum_{j=1}^{n_b-m+1} U_j(\mathbf{b}_j, d_j) \quad , \quad 1 \le m \le n_b \qquad (Eq\ 6.6.5)$$

But equally well, the adaptation algorithm can be changed to :

$$U_m(z) = \sum_{j=1}^{m} U_j(\mathbf{b}_j, d_j) \quad , \quad 1 \leq m \leq n_b \qquad (Eq\ 6.6.6)$$

The adaptation algorithms defined by Eqs 6.6.4, 6.6.5 and 6.6.6 can be combined to accomodate any number of excitation sequences (not necessarily as many as the FIR filter sections). An example is shown in Fig 6.6.1(b), which depicts the "optimised" synthesis filter of a BER coder, prior to the initiation of the excitation adaptation and speech synthesis operations. Three FIR filter sections and two excitation sequences are used. The excitation adapatation algorithm derived from Fig 6.6.1(b) is :

$$\begin{aligned} U_1(z) &= U_1(\mathbf{b}_1, d_1) + U_2(\mathbf{b}_2, d_2) + U_2(\mathbf{b}_3, d_3) \\ U_2(z) &= U_2(\mathbf{b}_2, d_2) + U_2(\mathbf{b}_3, d_3) \end{aligned} \qquad (Eq\ 6.6.7)$$

This algorithm has been formed by combining the algorithms of Eqs 6.6.4 and 6.6.5. By clocking the filter n times, both excitation sequences are updated and the synthesised speech signal is recovered at the output of the synthesis filter. As in the previous example, the sequences held in the delay lines of the encoder and decoder are initially set to preconstructed sequences of random samples (common to both the encoder and decoder).

An extended excitation adaptation algorithm can combine the adaptation strategies defined by Eqs 6.6.1, 6.6.4, 6.6.5 and 6.6.6. An example is shown in Fig 6.6.2(a). Two FIR filter sections are defined, but the two excitation sequences are updated using two different adaptation algorithms. The adaptation algorithm derived from Fig 6.6.2(a) is :

$$\begin{aligned} U_1(z) &= \left(U_1(\mathbf{b}_1, d_1) + U_2(\mathbf{b}_2, d_2)\right) A(z) + M_y(z) \\ U_2(z) &= U_2(\mathbf{b}_2, d_2) \end{aligned} \qquad (Eq\ 6.6.8)$$

The excitation sequence $U_1(z)$ is updated from the output of the synthesis filter and has therefore speech-like characteristics.

The versatility of the excitation definition algorithms can be extended by introducing a non-adaptive element. This can be a "fixed" sequence of zero-mean Gaussian-distributed random samples, which can be used to substitute one or more of the excitation sources. If these "fixed" excitation sequences were exclusively employed by the BER coder, the input-output relationship

corresponding to the operation of the synthesis filter could be expressed in a similar way as in the case of the adaptive components (Eq 6.4.5) :

$$S_y(z) = A(z) \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n-1} c_j(i-n-d_j-k)\ z^{-i} + M_y(z) \qquad (Eq\ 6.6.9)$$

The optimised filter parameters are employed in Eq 6.6.9, and the "fixed" random sequences are only specified in the negative time direction :

$$\{c_j(i)\} \quad , \quad i<o \quad , \quad j=1,2,\ldots,n_b \qquad (Eq\ 6.6.10)$$

Eq 6.6.9 can be rewritten as :

$$S_y(z) = A(z) \sum_{j=1}^{n_b} C_j(\mathbf{b}_j, d_j) + M_y(z) \qquad (Eq\ 6.6.11)$$

where as before, the convenient notation used is :

$$C_m(\mathbf{b}_j, d_j) = \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n-1} c_m(i-n-d_j-k)\ z^{-i} \quad , \quad 1 \le m \le n_b \quad , \quad 1 \le j \le n_b \qquad (Eq\ 6.6.12)$$

The "fixed" random sequences can substitute any excitation sequence in the adaptation algorithms defined by Eqs 6.6.1, 6.6.4, 6.6.5 and 6.6.6. Two such examples are shown in Figures 6.6.2(b) and 6.6.3(a). In both cases, the synthesis filter contains two FIR filter sections, and two excitation sequences are employed, one of whom is a "fixed" random sequence. In the first example, the adaptation algorithm of Eq 6.6.1 has been modified to accept one "fixed" sequence, and the adaptation equation is :

$$U_1(z) = \left(U_1(\mathbf{b}_1, d_1) + C_1(\mathbf{b}_2, d_2)\right) A(z) + M_y(z) \qquad (Eq\ 6.6.13)$$

In this case, the synthesised speech signal is :

$$S_y(z) = A(z) \left[\sum_{k=o}^{q_1} b_k^1 \sum_{i=o}^{n-1} s_y(i-n-d_1-k)\ z^{-i} + \sum_{k=o}^{q_2} b_k^2 \sum_{i=o}^{n-1} c_1(i-n-d_2-k)\ z^{-i} \right] + M_y(z)$$

$$(Eq\ 6.6.14)$$

In the second example (Fig 6.6.3(a)) the algorithm of Eq 6.6.5 has been modified to include one "fixed" excitation source. The adaptation equation is :

$$U_1(z) = U_1(\mathbf{b}_1, d_1) + C_1(\mathbf{b}_2, d_2) \qquad (Eq\ 6.6.15)$$

and the synthesised speech signal is :

$$S_y(z) = A(z) \left[\sum_{k=0}^{q_1} b_k^1 \sum_{i=0}^{n-1} u_1(i-n-d_1-k)\ z^{-i} + \sum_{k=0}^{q_2} b_k^2 \sum_{i=0}^{n-1} c_1(i-n-d_2-k)\ z^{-i} \right] + M_y(z)$$

$$(Eq\ 6.6.16)$$

This last excitation adaptation algorithm makes the BER coder equivalent to a Code Excited LPC (CELP) coder that uses overlapping excitation codewords [6.20,6.21]. The adaptation of the excitation sequence performed by the BER coder corresponds to the operation of the Long Term Predictor (LTP) in a CELP coder, and the "fixed" random sequence employed by the BER coder corresponds to an excitation codebook whose consequtive entries overlap by $n-1$ samples.

The "fixed" random sequences can substitute all the excitation sequences of the adaptation algorithms defined by Eqs 6.6.4, 6.6.5 and 6.6.6. In such a case, the synthesised speech is recovered using Eq 6.6.9.

As the number of excitation sequences increases, the adaptive and non-adaptive elements of the excitation can be combined in a greater number of ways, resulting an increasing number of possible excitation adaptation algorithms. When the numerous excitation adaptation algorithms are combined with the possible filter optimisation algorithms (described in Section 6.5), the resulting number of BER coding schemes can be remarkably large.

In Fig 6.6.4, a summary of the possible excitation adaptation algorithms is given, for the case of a two-input synthesis filter. The first three algorithms $P_2(1)$, $P_2(2)$ and $P_2(3)$ require a single excitation source, while the rest require two excitation sources. The $P_2(3)$ and $P_2(12)$ algorithms involve exclusively "fixed" random sequences. The $P_2(9)$ and $P_2(11)$ adaptation algorithms are similar to the $P_2(8)$ and $P_2(10)$ algorithms but during the optimisation of the synthesis filter, the FIR filter section corresponding to to the "fixed" random excitation sequence is optimised first.

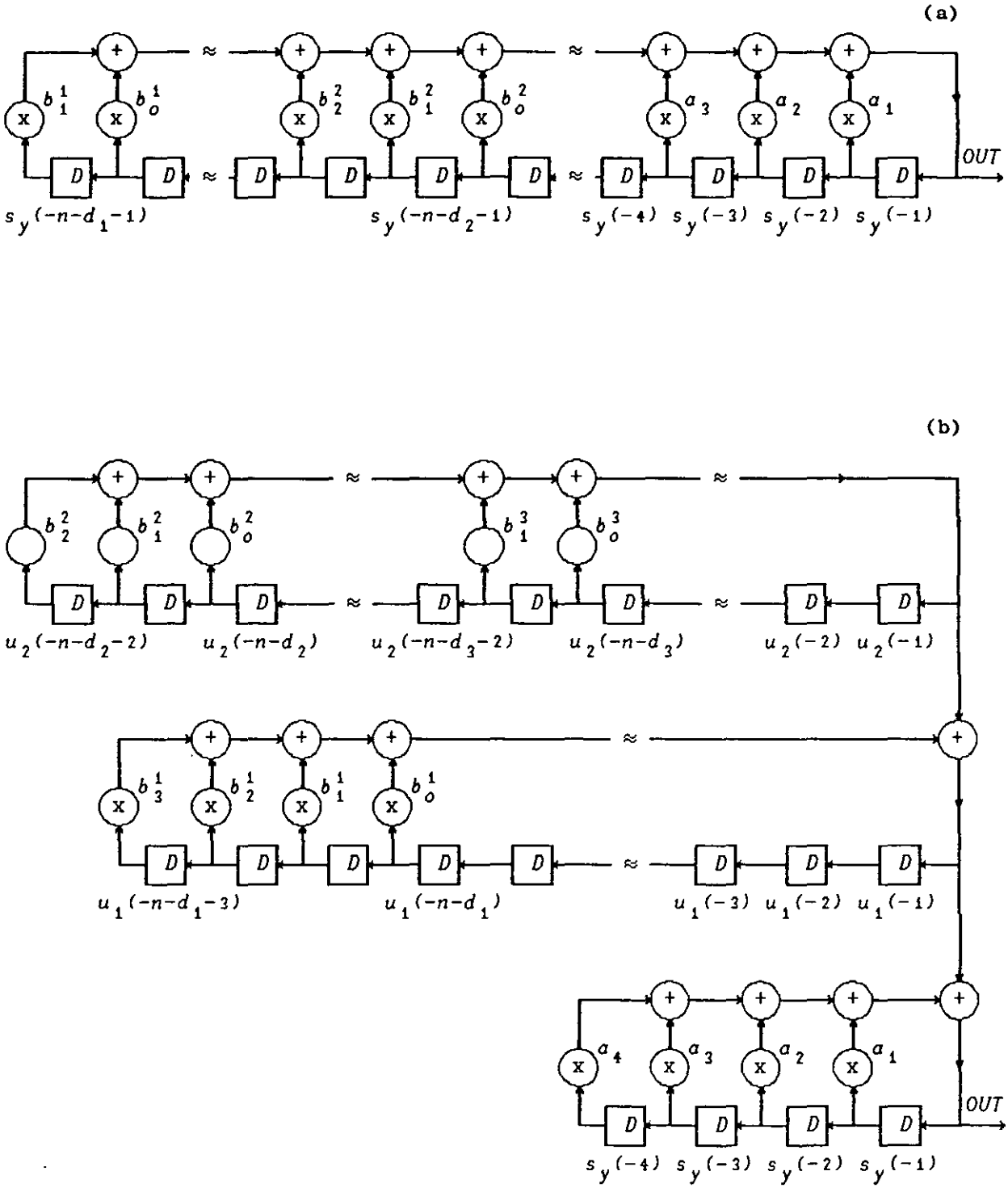


FIGURE 6.6.1 Two different structures of the BER Synthesis Filter :

(a) The Excitation is defined from the past synthesised speech samples. See Eq 6.6.3 or Excitation Adaptation Method $P_2(1)$.

(b) Two separate Adaptive Excitation sequences are defined. See Eq 6.6.5 or Excitation Adaptation Method $P_2(5)$.

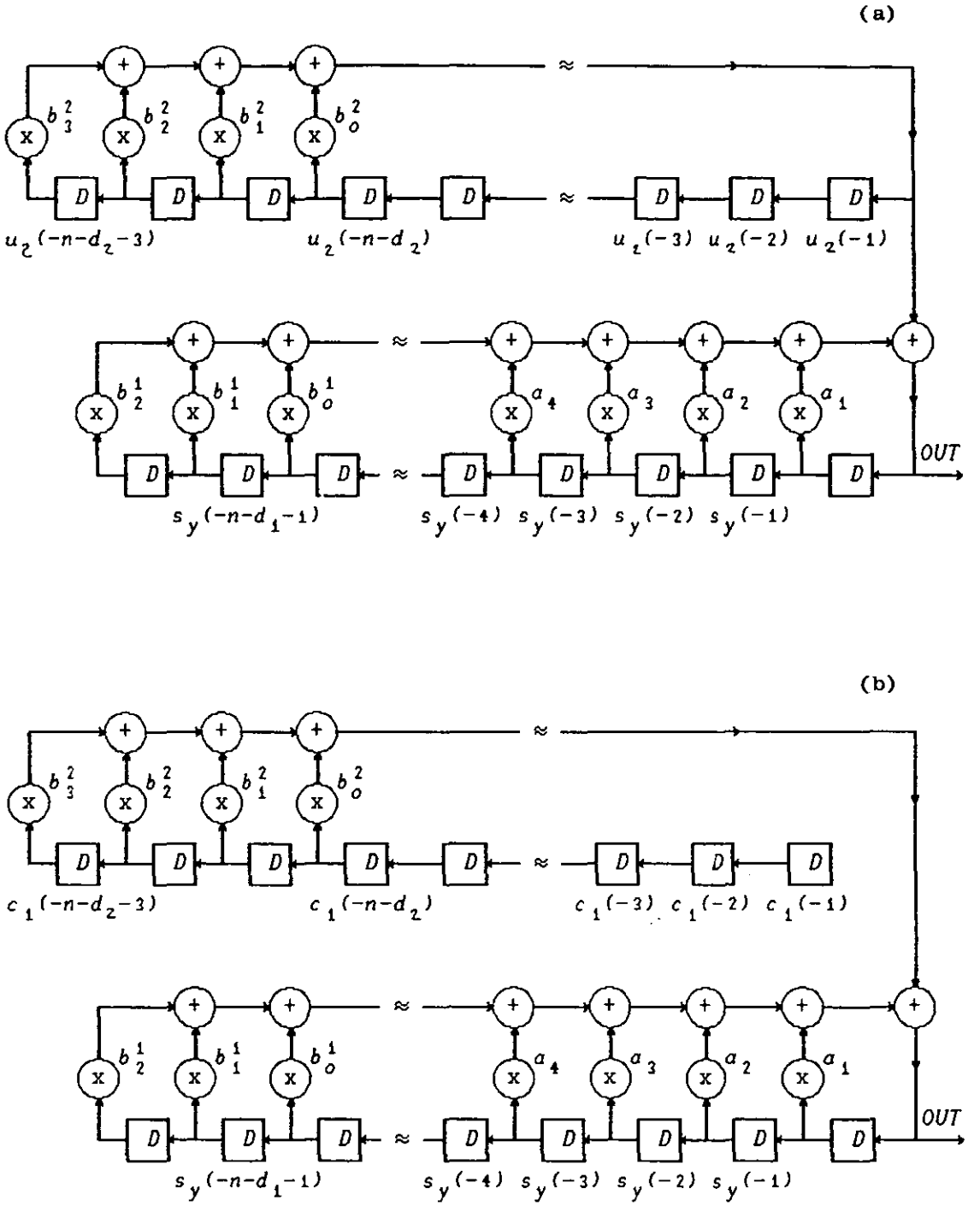
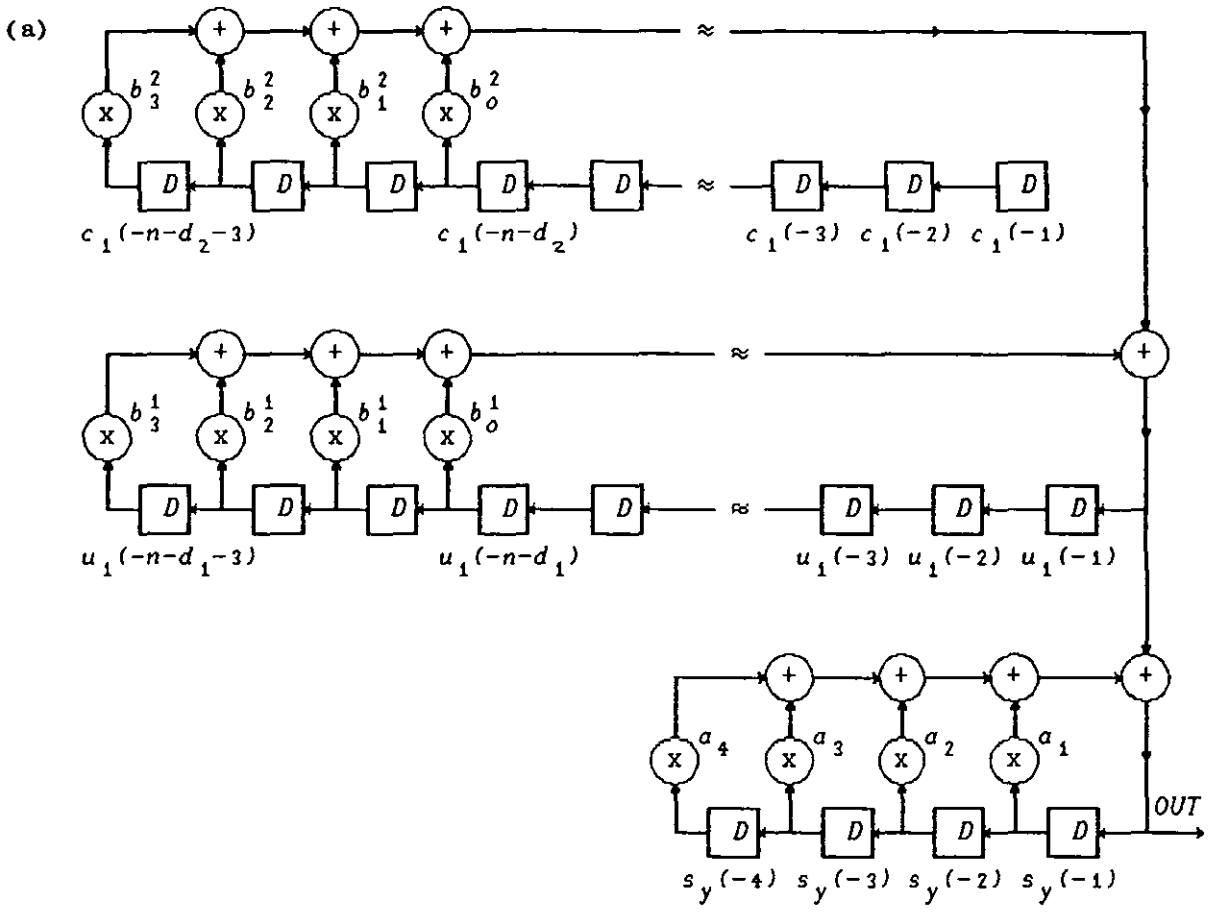


FIGURE 6.6.2 Two more different structures of the BER Synthesis Filter :

(a) The Excitation is defined from a combination of past Synthesised Speech and an Adaptive Excitation Source.
See Eq 6.6.8 or Excitation Adaptation Method $P_2(4)$.

(b) The Excitation is defined from a combination of past Synthesised Speech and a Fixed Excitation Source.
See Eq 6.6.13 or Excitation Adaptation Method $P_2(8)$.



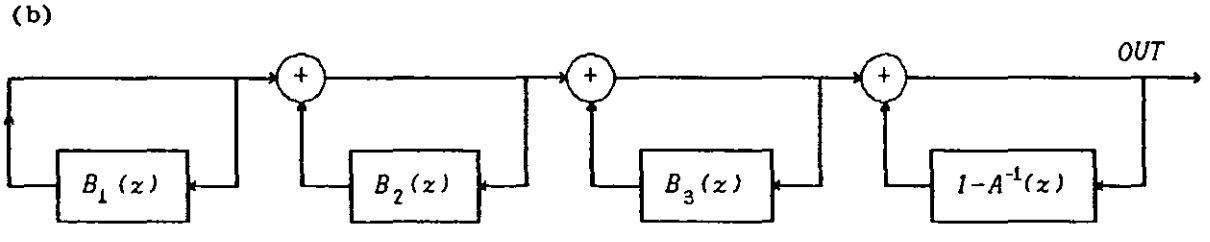


FIGURE 6.6.3 (a) Another possible structure of the BER Synthesis Filter : One Fixed and one Adaptive Excitation sequences are defined. See Eq 6.6.15 or Excitation Adaptation Method $P_2(10)$.

(b) The Self-Excited Vocoder. Three Long-Term Predictors are employed. The first Long-Term Predictor is operating without excitation.

METHOD	RECURSIVE EXCITATION ADAPTATION ALGORITHM
$P_2(1)$	$U_1(z) = \left(U_1(\mathbf{b}_1,d_1) + U_1(\mathbf{b}_2,d_2)\right) A(z) + M_y(z)$
$P_2(2)$	$U_1(z) = U_1(\mathbf{b}_1,d_1) + U_1(\mathbf{b}_2,d_2)$
$P_2(3)$	EXCITATION FROM RANDOM SEQUENCE $\{c_1(i)\}$, $i<o$
$P_2(4)$	$U_1(z) = \left(U_1(\mathbf{b}_1,d_1) + U_2(\mathbf{b}_2,d_2)\right) A(z) + M_y(z)$ $U_2(z) = U_2(\mathbf{b}_2,d_2)$
$P_2(5)$	$U_1(z) = U_1(\mathbf{b}_1,d_1) + U_2(\mathbf{b}_2,d_2)$ $U_2(z) = U_2(\mathbf{b}_2,d_2)$
$P_2(6)$	$U_1(z) = U_1(\mathbf{b}_1,d_1)$ $U_2(z) = \left(U_1(\mathbf{b}_1,d_1) + U_2(\mathbf{b}_2,d_2)\right) A(z) + M_y(z)$
$P_2(7)$	$U_1(z) = U_1(\mathbf{b}_1,d_1)$ $U_2(z) = U_1(\mathbf{b}_1,d_1) + U_2(\mathbf{b}_2,d_2)$
$P_2(8)$	$U_1(z) = \left(U_1(\mathbf{b}_1,d_1) + C_1(\mathbf{b}_2,d_2)\right) A(z) + M_y(z)$
$P_2(9)$	AS $P_2(8)$ BUT OPTIMISING FIRST WITH RESPECT TO $\{c_1(i)\}$
$P_2(10)$	$U_1(z) = U_1(\mathbf{b}_1,d_1) + C_1(\mathbf{b}_2,d_2)$
$P_2(11)$	AS $P_2(10)$ BUT OPTIMISING FIRST WITH RESPECT TO $\{c_1(i)\}$
$P_2(12)$	EXCITATION FROM SEQUENCES $\{c_1(i)\}$ and $\{c_2(i)\}$, $i<o$

FIGURE 6.6.4 Possible Excitation Adaptation algorithms when the BER Synthesis Filter contains two FIR sections.

6.6.1 The Self Excited Vocoder

The Self Excited Vocoder [6.22,6.23,6.24] is a speech coding scheme that shares a number of common elements with the MPE and CELP speech coders. It employs a synthesis filter which consists of a number of Long Term Predictors (LTPs) and an all-pole LPC filter in cascade. It relies solely on the adaptation of the filter parameters to recover the synthesised speech signal and does not transmit any information related to the excitation signal.

In Fig 6.6.3(b), the synthesis filter of a Self Excited Vocoder is shown, when 3 LTPs are employed. The first LTP $B_1(z)$ (defined by Eq 6.2.1), has no input at all and its output can be calculated using the recursive formula :

$$u_1(i) = \sum_{k=o}^{q_1} b_k^1 \, u_1(i-n-d_1-k) \quad , \quad o \le i \le n-1 \qquad (Eq\ 6.6.1.1)$$

In general, the output of the jth LTP can be calculated using the formula:

$$u_j(i) = \sum_{k=o}^{q_j} b_k^j \, u_j(i-n-d_j-k) + u_{j-1}(i) \quad , \quad o \le i \le n-1 \quad , \quad 2 \le j \le n_b \qquad (Eq\ 6.6.1.2)$$

where n_b is the number of LTPs used. The one-sided z-transform of Eq 6.6.1.2 is (see Appendix C) :

$$U_j(z) = \frac{\sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n+d_j+k-1} u_j(i-n-d_j-k)\, z^{-i}}{1 - B_j(z)} + \frac{U_{j-1}(z)}{1 - B_j(z)} \qquad (Eq\ 6.6.1.3)$$

By considering only the samples of $U_j(z)$ that lie inside the interval $[o,n-1]$, Eq 6.6.1.3 can be transformed to :

$$U_j(z) = \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n-1} u_j(i-n-d_j-k)\, z^{-i} + U_{j-1}(z) \qquad (Eq\ 6.6.1.4)$$

Eq 6.6.1.4 can be expanded by performing the recursions, and the result is the non-recursive formula :

$$U_j(z) = \sum_{m=1}^{j} \sum_{k=o}^{q_m} b_k^m \sum_{i=o}^{n-1} u_m(i-n-d_m-k)\, z^{-i} \quad , \quad 1 \le j \le n_b \qquad (Eq\ 6.6.1.5)$$

The output of the last LTP is fed to the all-pole filter $A(z)$ whose output

is the synthesised speech signal :

$$S_y(z) = A(z) \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n-1} u_j(i-n-d_j-k)\, z^{-i} + M_y(z) \qquad (Eq\ 6.6.1.6)$$

Eq 6.6.1.6 is identical to Eq 6.4.5 and shows how closely related the BER coding schemes and the Self Excited Vocoder are.

The Self Excited Vocoder performs the optimisation of the synthesis filter in stages. In the first stage, only the $B_1(z)$ LTP is allowed to have non-zero coefficients, and its parameters are optimised using an Analysis by Synthesis procedure which minimises the distortion introduced by the coding process (Eqs 6.4.9 and 6.4.10). In the second stage the parameters of the first LTP are fixed, and the second LTP $B_2(z)$ is optimised in an Analysis by Synthesis loop that minimises the distortion. Using the same process, the rest of the LTPs are optimised in the following stages. When a single LTP is employed, this filter optimisation algorithm is equivalent to the $M_1(4)$ algorithm described in Section 6.5. When two LTPs are used, the process is equivalent to the $M_2(11)$ optimisation algorithm.

The operation of the LTPs automatically updates the excitation sequences stored in the corresponding delay lines, and this adaptation operation is defined by Eq 6.6.1.5. This equation is identical to Eq 6.6.6, which is one of the possible excitation adaptation strategies that can be adopted by the BER coder. When the Self Excited Vocoder employs two LTPs, the excitation adaptation process is equivalent to the $P_2(7)$ adaptation algorithm.

The Self Excited Vocoder uses two of the possible filter optimisation and excitation adaptation algorithms available to a general BER coder, and is therefore a special case of BER speech coding.

6.7 Comparison of the BER Algorithms

In Sections 6.5 and 6.6, various synthesis filter optimisation and excitation adaptation algorithms were defined. These algorithms form the basis for the development of numerous BER coding schemes. The behaviour of three such coding schemes will now be examined. The first scheme (CODER-1) employs a single-input synthesis filter and defines a single sequence of past excitation samples. The excitation adaptation algorithm utilises the output of the synthesis filter and is defined by the equation :

$$U_1(z) = A(z) \sum_{k=o}^{2} b_k^1 \sum_{i=o}^{n-1} u_1(i-n-d_1-k)\ z^{-i} + M_y(z) \qquad (Eq\ 6.7.1)$$

where :

$$A(z) = \frac{1}{1-a_1 z^{-1}-a_2 z^{-2}} \qquad (Eq\ 6.7.2)$$

The FIR section of the synthesis filter therefore includes 3 coefficients (b_o^1, b_1^1, b_2^1) and one delay parameter (d_1), and the filter $A(z)$ is a second-order all-pole filter. The filter optimisation algorithm chosen is the $M_1(2)$ which optimises the a and $\mathbf{b}_1$ coefficients using the Linear Prediction method (Section 6.3), and then reoptimises the $\mathbf{b}_1$ coefficients to minimise the energy of the signal distortion (for each value of the delay parameter). The speech frame contains 24 samples $(n=24)$ and the delay parameter may take 256 different values $(o \leq d_1 \leq 255)$.

The second scheme (CODER-2) employs a two-input synthesis filter and defines two excitation sequences. The $P_2(2)$ excitation adaptation algorithm and the $M_2(11)$ filter optimisation method are used. Each FIR section of the synthesis filter includes one coefficient (b_o^j) and one delay parameter d_j. The excitation adaptation algorithm is therefore defined by the equation :

$$U_1(z) = \sum_{j=1}^{2} b_o^j \sum_{i=o}^{n-1} u_j(i-n-d_j)\ z^{-i}$$

$$U_2(z) = b_o^2 \sum_{i=o}^{n-1} u_2(i-n-d_2)\ z^{-i} \qquad (Eq\ 6.7.3)$$

The filter $A(z)$ is a 12-th order all pole filter :

$$A(z) = \frac{1}{1 - \sum_{m=1}^{12} a_m z^{-m}} \qquad (Eq\ 6.7.4)$$

which is defined over an interval of 144 speech samples, using the Maximum-Entropy LPC method. The analysis frame size and the range of the delay parameters are set as in CODER-1 ($n=24$, $n_d=256$). Note that if the total number of parameters transmitted per second is measured, then both CODER-1 and CODER-2 require the transmission of 2000 parameters (filter coefficients and delay parameters) per second.

CODER-3 is equivalent to CODER-2, but employs two "fixed" random excitation sequences and uses the $P_2(12)$ excitation definition algorithm.

The transient response of the three coding schemes is first examined, and in particular the response of the coders in the first 24 ms of their operation can be observed in Figures 6.7.1, 6.7.2 and 6.7.3. In Fig 6.7.1(a), the 24 speech samples (3 ms at 8 kHz sampling rate) of the current analysis frame are plotted to the right of the vertical axis (positive time direction), while the past 168 samples (21 ms) are in the negative time direction. In Fig 6.7.1(c), the 280 past excitation samples stored in the delay line of CODER-1, are plotted in the negative time direction. The first 112 samples of this sequence are random and correspond to the random excitation sequence which was initially installed in both the encoder and the decoder, before they started operating. The rest 168 samples have been produced by the excitation adaptation algorithm and are exactly the same as those of the synthesised speech signal in Fig 6.7.1(b).

The time sequences are observed after the completion of the filter optimisation and excitation adaptation operations, so both the continuation of the excitation sequence in the positive time direction, and the synthesised speech waveform in the current frame are shown in Figures 6.7.1(c) and (b).

In Fig 6.7.1(d), the evolution of the filter optimisation process is shown by plotting the changing value of the SNR measured as the value of the delay parameter is varied ($-255 \leq -d_1 \leq o$). The optimum delay value corresponds to the position of the maximum SNR. It is evident that the highest SNR values occur in the section where the excitation signal is speech-like, and that the SNR curve itself follows the "periodic variation" of the speech waveform. The

locations of the SNR peaks correspond to the segments of the excitation waveform which are most similar to the speech signal in the current analysis frame.

The remarkable ability of the BER coder to respond to sudden changes in the speech waveform is demonstrated in Fig 6.7.1. The synthesised speech signal in Fig 6.7.1(b) follows the original speech waveform in Fig 6.7.1(a) closely, even in the first processed frame. This happens in spite of the fact that the BER coder reconstructs the speech signal and the filter excitation using a backward adaptation algorithm.

The response of CODER-2 in the same time interval is shown in Fig 6.7.2(a). The two excitation sequences $\{u_1(i)\}$ and $\{u_2(i)\}$ are also shown in Figures 6.7.2(b) and (c). The excitation sequences exhibit noise-like characteristics and differ in the range of sample values, as a consequence of the particular filter optimisation and excitation adaptation strategies chosen.

Figure 6.7.2(d) shows the two SNR curves corresponding to the separate optimisation of the two delay variables d_1 (lower curve) and d_2 (upper curve). The SNR peaks are much more sharp than in the case of CODER-1, and the lower SNR curve has a periodic component similar to that of the speech waveform. The highest SNR attained by CODER-2 is, in this case, approximately 5 dBs lower than the highest SNR attained by CODER-1.

The response of CODER-3 is shown in Fig 6.7.3(a). The "fixed" random excitation sequences $\{c_1(i)\}$ and $\{c_2(i)\}$ are shown in Figures 6.7.3(b) and (c). The results obtained from CODER-3 are comparable to those of CODER-2, even though the excitation sequences are not allowed to adapt to the changing speech characteristics. The two SNR curves of Fig 6.7.3(d) are similar to the corresponding curves of CODER-2, but the periodic component of the lower curve is now missing for obvious reasons.

In Figures 6.7.4, 6.7.5 and 6.7.6, the steady-state behaviour of the three coding schemes can be studied, by observing their response during a 38 ms interval of voiced speech. The SNR curve corresponding to the filter optimisation in CODER-1, shown in Fig 6.7.4(c), is definitely showing the same periodic structure as the speech waveform, but the height of the SNR peaks is progressively declining as the value of the delay parameter is increased. This reflects the growing dissimilarity between two segments of the speech waveform as their relative distance is increased. It also suggests that the

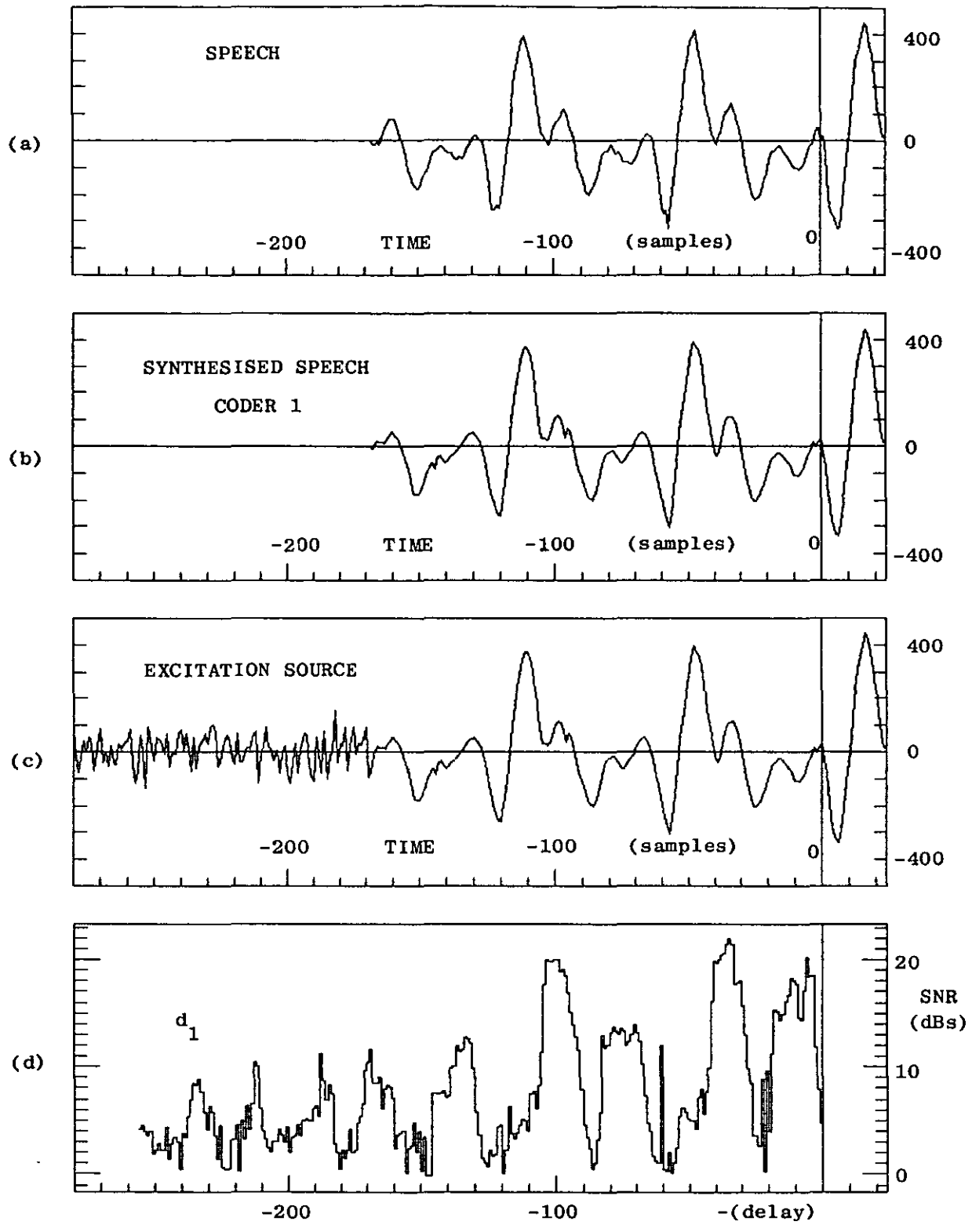


FIGURE 6.7.1 Signals obtained from coding speech with CODER-1. The first 24 ms of the coder's operation are shown. (a) Original Speech (b) Synthesised Speech (c) The Excitation Sequence (d) The SNR obtained for the last 3 ms (24 samples) of the speech waveform, while varying the value of the delay parameter d_1.

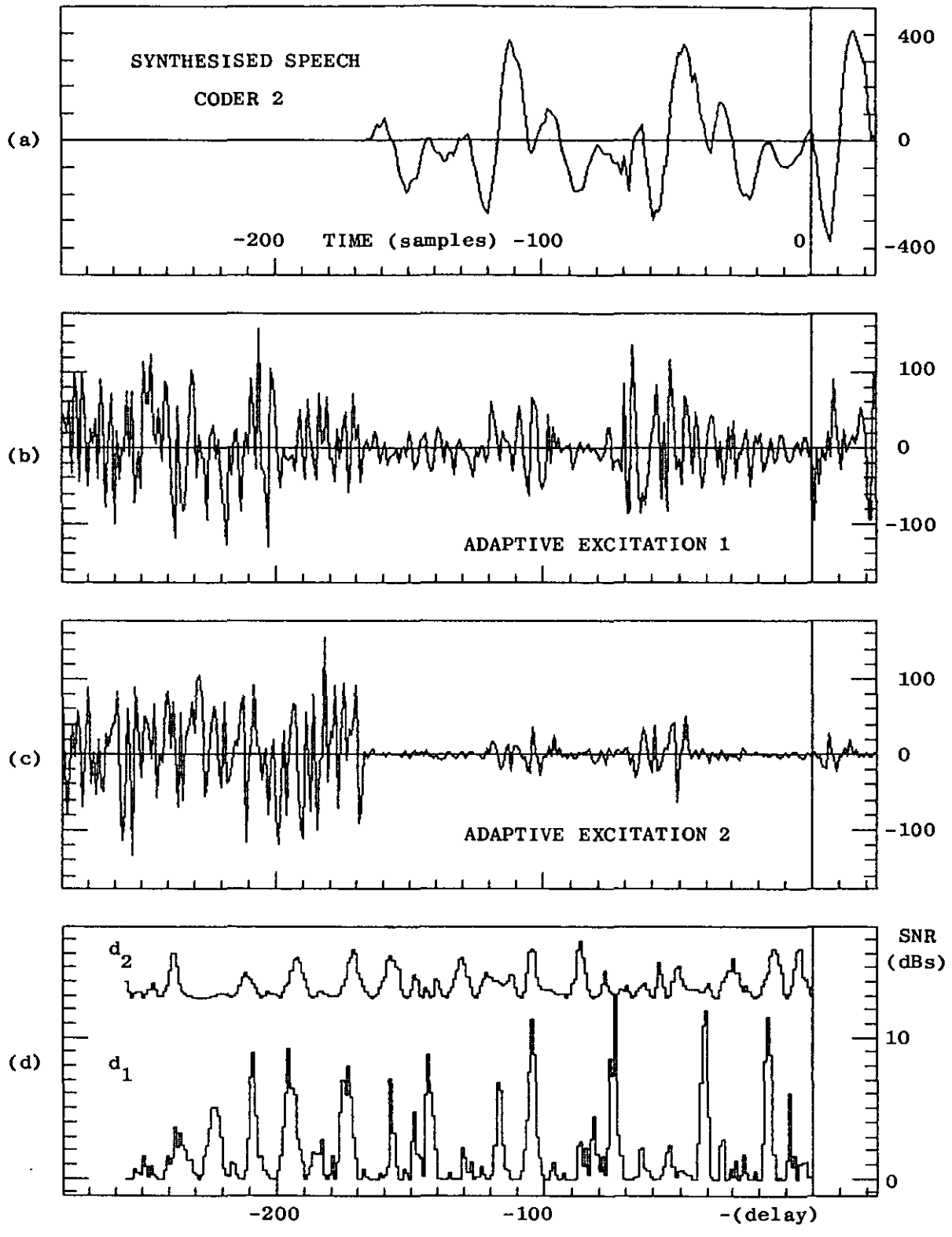


FIGURE 6.7.2 Signals obtained from coding speech with CODER-2. The first 24 ms of the coder's operation are shown. (a) Synthesised Speech (b) First Adaptive Excitation Sequence (c) Second Adaptive Excitation Sequence (d) The SNR obtained for the last 3 ms (24 samples) of the speech waveform, while varying the value of the delay parameters d_1 and d_2.

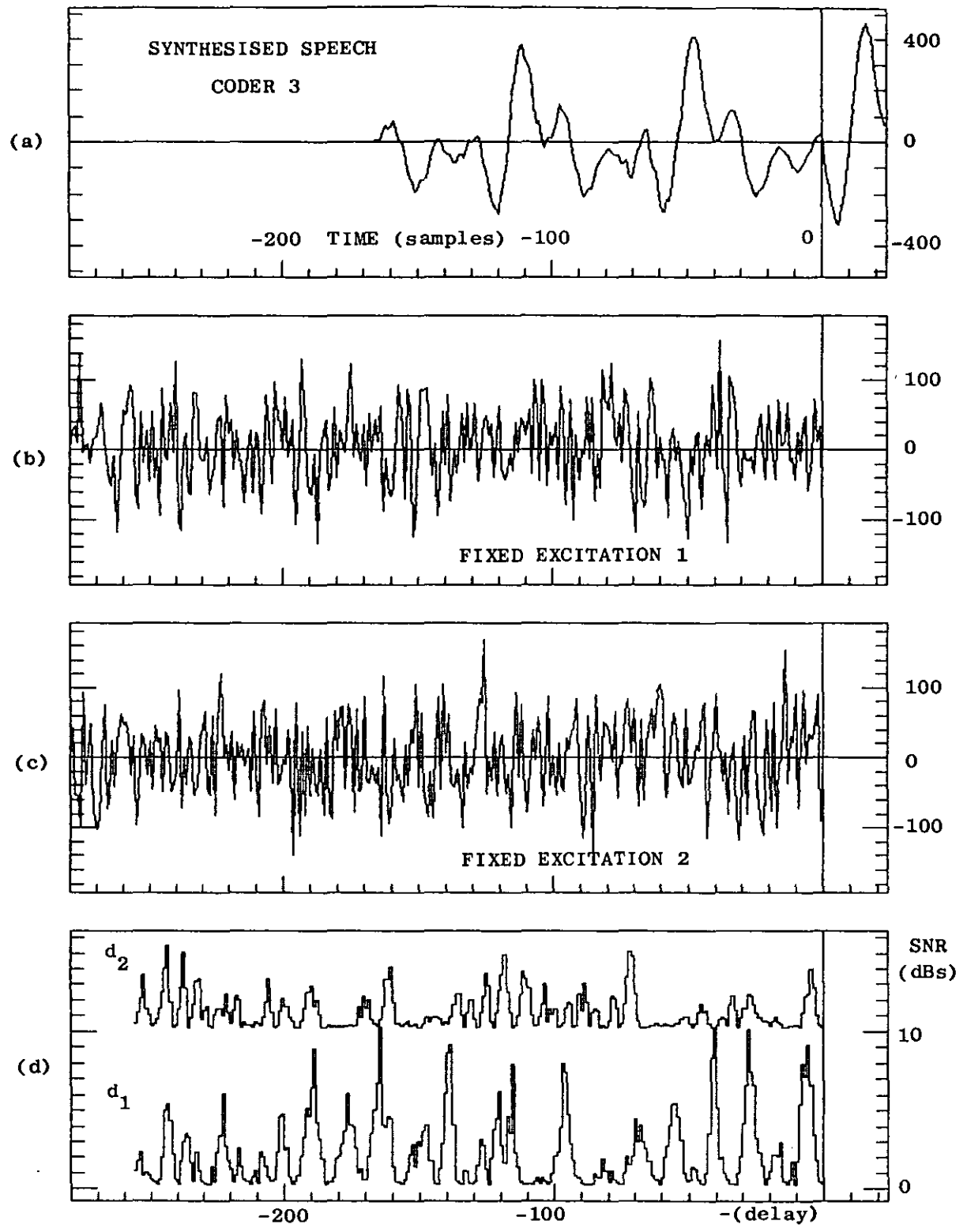


FIGURE 6.7.3 Signals obtained from coding speech with CODER-2. The first 24 ms of the coder's operation are shown. (a) Synthesised Speech (b) First Fixed Excitation Sequence (c) Second Fixed Excitation Sequence (d) The SNR obtained for the last 3 ms (24 samples) of the speech waveform, while varying the value of the delay parameters d_1 and d_2.

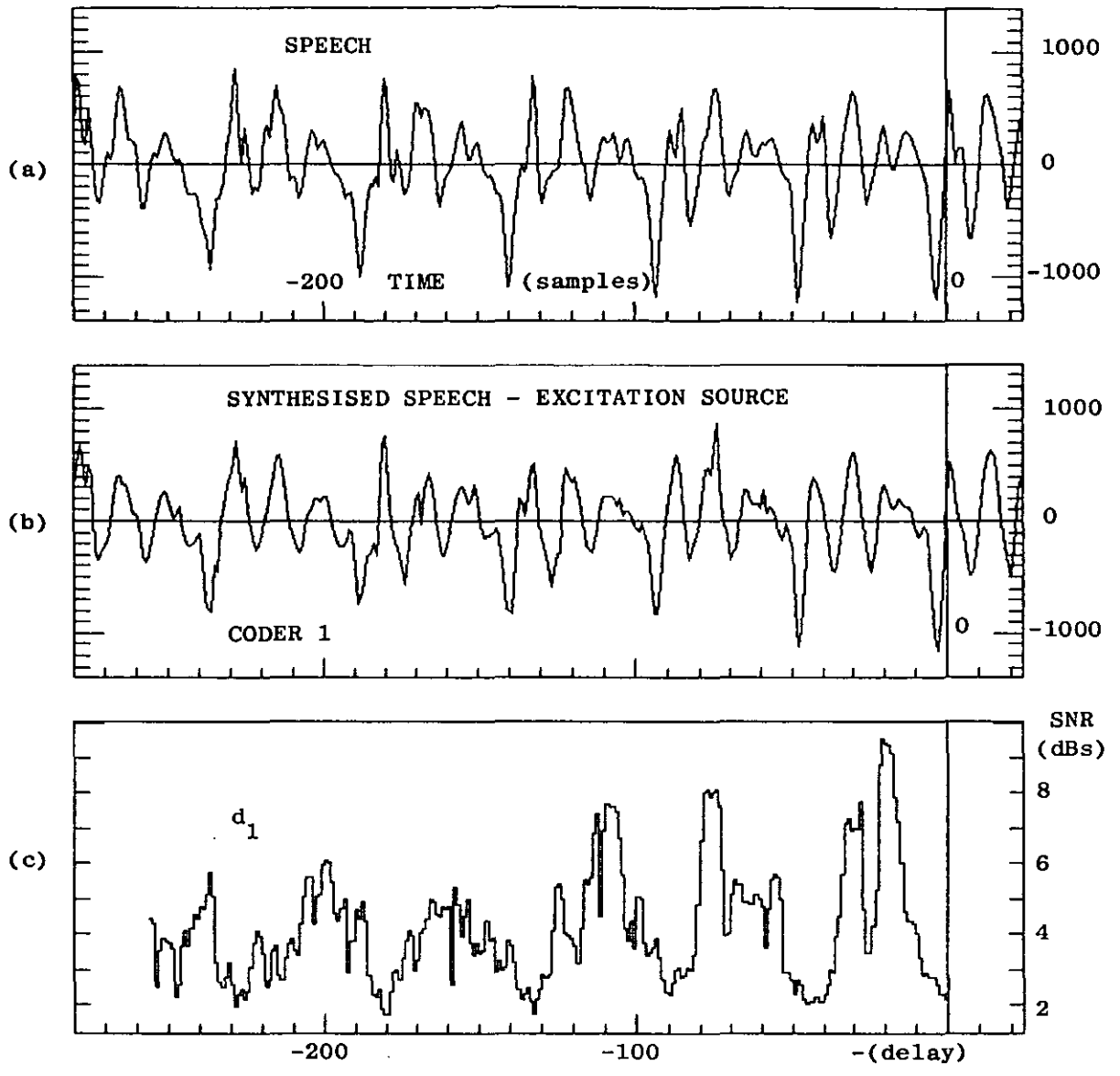


FIGURE 6.7.4 Signals obtained from coding speech with CODER-1.
(a) Original Speech (b) Synthesised Speech, which also forms the Excitation Sequence (c) The SNR obtained for the last 3 ms (24 samples) of the speech waveform, while varying the value of the delay parameter d_1.

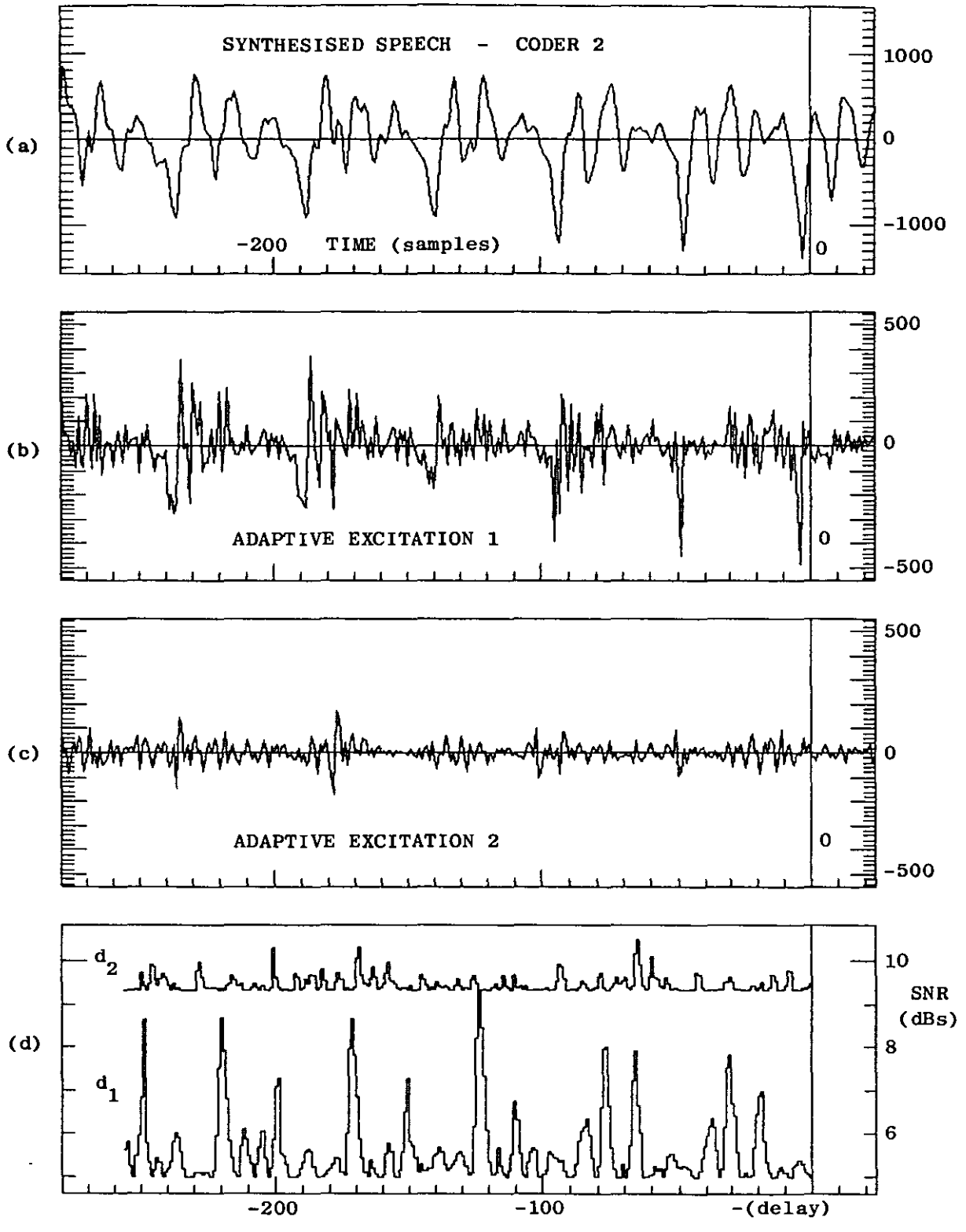


FIGURE 6.7.5 Signals obtained from coding speech with CODER-2.
(a) Synthesised Speech (b) First Adaptive Excitation Sequence (c) Second Adaptive Excitation Sequence (d) The SNR obtained for the last 3 ms (24 samples) of the speech waveform, while varying the value of the delay parameters d_1 and d_2.

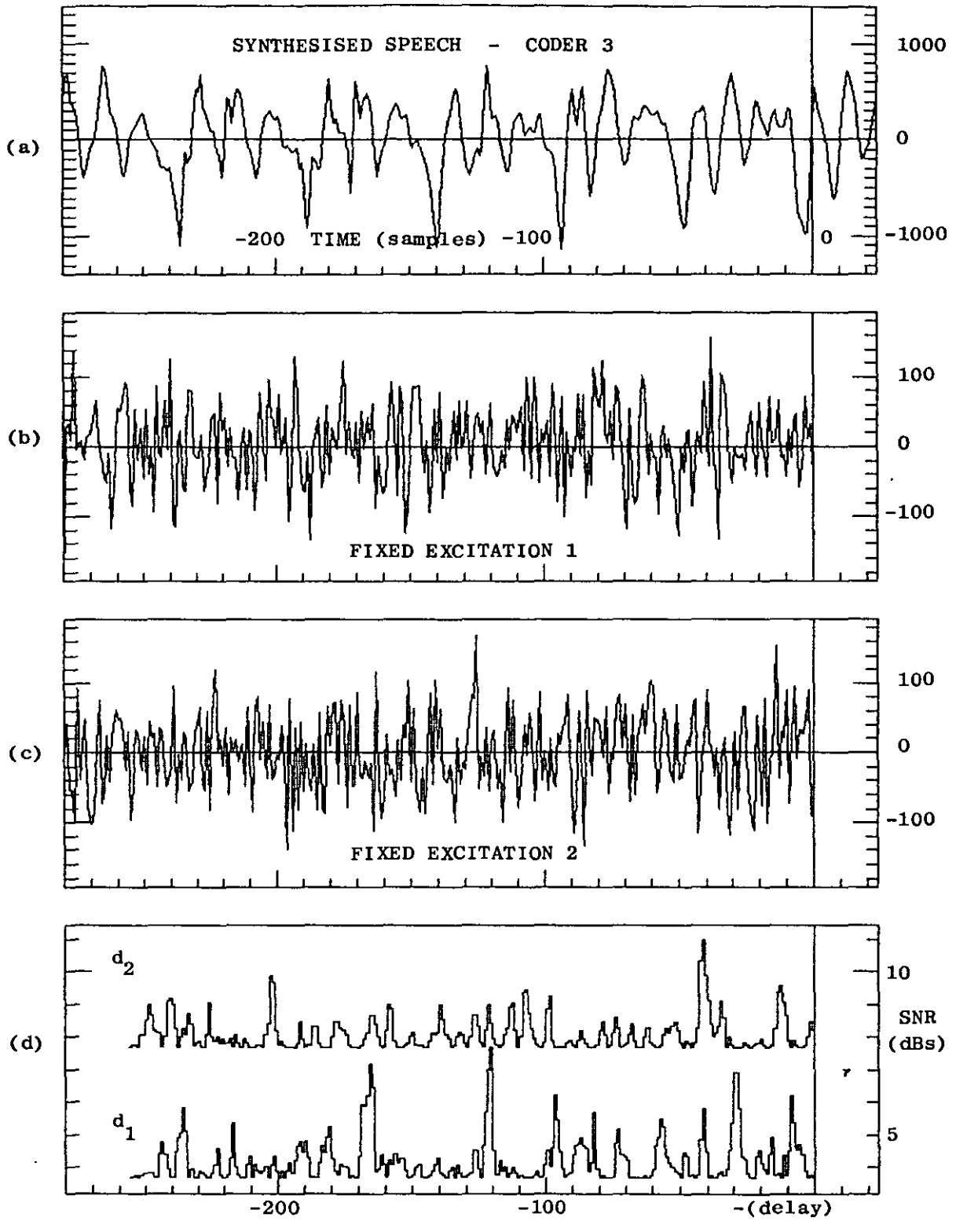


FIGURE 6.7.6 Signals obtained from coding speech with CODER-3. (a) Synthesised Speech (b) First Fixed Excitation Sequence (c) Second Fixed Excitation Sequence (d) The SNR obtained for the last 3 ms (24 samples) of the speech waveform, while varying the value of the delay parameters d_1 and d_2.

maximum permissible value of the delay parameter of CODER-1 can be reduced without affecting the coder's performance.

The excitation sequences of CODER-2, shown in Figures 6.7.5(b) and (c), are clearly different. The $\{u_1(i)\}$ sequence has a definite pitch structure corresponding to the pitch structure of the speech waveform, while $\{u_2(i)\}$ is more random and has a smaller dynamic range. A comparison of the SNR curves corresponding to CODER-2 and CODER-3, shown in Figures 6.7.5(d) and 6.7.6(d), reveals a subtle difference between the two schemes. The first filter optimisation stage of CODER-2 (optimisation of d_1) gives better results than the equivalent stage of CODER-3, because CODER-2 relies more on the $\{u_1(i)\}$ excitation sequence, which has clearly adapted to the speech waveform characteristics. The results obtained from the second optimisation stage of the two schemes are similar, because the two excitation sequences $\{u_2(i)\}$ and $\{c_2(i)\}$ have very similar properties. CODER-2 is therefore expected to give better results than CODER-3, when only one excitation sequence is used.

In Fig 6.7.7, a comparison of the different filter optimisation methods is presented, for two different excitation adaptation strategies. The figures correspond to the average Segmental-SNR attained when these algorithms are employed by a BER coder to encode a 4 sec male/female speech interval. The filter coefficients are left unquantized, and the total number of parameters (filter coefficients and delay parameters) transmitted per second is set equal to 2000. A second column has also been added to each excitation adaptation algorithm (Reopt.), to include the SNR results obtained when the filter coefficients are reoptimised in the final filter-definition stage (for the optimum values of the delay parameters) by minimising the energy of the signal distortion.

The first 6 rows of Fig 6.7.7 correspond to a single-input synthesis filter, and the excitation adaptation strategies are derived from the $P_2(1)$ and $P_2(2)$ algorithms by limiting the number of FIR sections from 2 to 1. The $B_1(z)$ FIR filter has 3 coefficients, and the all-pole filter $A(z)$ is either 2nd order and defined every 24 samples (A_{24}), or it is a 12th order filter and is defined every 144 samples (A_{144}). When the $A(z)$ filter is estimated separately from the $B_1(z)$ filter (as in methods $M_1(3)$ and $M_1(4)$), the Maximum-Entropy LPC method is employed. The speech analysis frame contains 24 samples ($n=24$), and the maximum value of the delay parameter is 63 ($0 \leq d_1 \leq 63$)

The lower 16 rows of Fig 6.7.7 correspond to a two-input synthesis filter and either the $P_2(1)$ or the $P_2(5)$ excitation adaptation strategies are used. Each of the two FIR filter sections contains a single coefficient (as in CODER-2), and the rest of the system variables (update rate of $A(z)$, frame size and maximum delay value) are set as in the case of the single-input synthesis filter.

From the results in Fig 6.7.7 it is clear that when the $P_2(1)$ excitation adaptation algorithm is employed, the performance of the BER coder is maximised when the **a** and $\mathbf{b}_j$ filter coefficients are jointly optimised (as in methods $M_1(1\text{-}2)$ and $M_2(1\text{-}8)$). In this case, both single-input and two-input synthesis filters can give good results, with the highest SNR obtained from method $M_2(8)$. When the **a** and $\mathbf{b}_j$ coefficients are optimised separately (as in methods $M_1(3\text{-}4)$ and $M_2(9\text{-}12)$) the $P_2(1)$ excitation adaptation algorithm favours the single-input synthesis filter and a high update rate for the filter $A(z)$ (A_{24}).

The $P_2(1)$ excitation adaptation algorithm therefore favours the operation of the BER coder with small frames, and minimises the encoding delay for which the BER encoder is responsible (in this case the delay is only 3 ms). The Linear Prediction (LP) and Distortion Minimisation (DM) filter-estimation algorithms perform equally well when the $P_2(1)$ algorithm is employed, and the final reoptimisation of the filter coefficients seems to offer little adavantage.

A two-input synthesis filter gives the best SNR results when the $P_2(2)$ and $P_2(5)$ excitation adaptation algorithms are employed. When the **a** and $\mathbf{b}_j$ filter coefficients are optimised separately (as in methods $M_2(9\text{-}12)$), low update rates (A_{144}) and a higher number of coefficients for the filter $A(z)$ improve the performance of the BER coder. The low update rate of the filter $A(z)$ results a higher encoding delay (18 ms in this case). Equally good results though can be achieved by jointly optimising the **a** and $\mathbf{b}_j$ coefficients (as in methods $M_2(1\text{-}8)$), in which case the encoding delay can be as small as when the $P_2(1)$ excitation adaptation algorithm is used. The DM filter estimation method performs better than the LP method, when the $P_2(2)$ or the $P_2(5)$ excitation adaptation algorithms are used. The performance of the LP method though can be improved and can approach that of the DM method, by using a final filter reoptimisation stage.

In Fig 6.7.8, 12 excitation adaptation algorithms are compared, under the same conditions as in Fig 6.7.7. It is evident that the $P_2(1)$ algorithm performs better when the **a** and $\mathbf{b}_j$ coefficients are jointly optimised (as in methods $M_2(8)$ and $M_1(2)$), irrespective of whether a single-input or a two-input synthesis filter is employed. Algorithm $P_2(2)$ on the other hand, gives best results when a two-input synthesis filter is employed (as in methods $M_2(8)$ and $M_2(12)$), whether the **a** and $\mathbf{b}_j$ coefficients are jointly optimised or not. The $P_2(3)$ algorithm employs one "fixed" random excitation sequence but does not perform as well as the $P_2(2)$ algorithm. By using two "fixed" excitation sequences though, the difference in performance between the adaptive excitation (algorithms $P_2(5)$ and $P_2(7)$) and the "fixed" excitation algorithms is minimised.

The algorithms that involve an excitation sequence with speech-like characteristics ($P_2(1)$,$P_2(4)$,$P_2(6)$,$P_2(8)$ and $P_2(9)$) perform better when the $M_2(8)$ filter optimisation method is used, and the best results are obtained from the $P_2(4)$ excitation adaptation algorithm. By comparing the SNR results of the $P_2(8)$ and $P_2(10)$ algorithms with those of $P_2(9)$ and $P_2(11)$, it becomes clear that the order of application of the excitation adaptation equations is important and can have an effect on the BER coder's performance.

Amongst the excitation adaptation algorithms that involve a noise-like adaptive excitation element ($P_2(2)$,$P_2(5)$,$P_2(7)$,$P_2(10)$ and $P_2(11)$), the best results are obtained from algorithm $P_2(5)$ which defines two adaptive excitation sequences. The BER coder which is equivalent to a CELP coder ($P_2(10)$) performs better than the Self-Excited Vocoder ($P_2(7)$). The best overall results are obtained from the excitation adaptation algorithms that define a speech-like excitation source.

EXCITATION ADAPTATION	$P_2(1)$		$P_2(2)$ or $P_2(5)$	
FILTER OPTIMISATION	SNR (dBs)		SNR (dBs)	
		(Reopt.)		(Reopt.)
$M_1(1)$	11.8	12.3	8.3	9.5
$M_1(2)$	12.7	12.7	9.9	9.9
$M_1(3)$ A_{24}	9.4	10.5	6.9	7.9
$M_1(3)$ A_{144}	9.0	10.0	7.8	8.4
$M_1(4)$ A_{24}	11.3	11.3	9.2	9.2
$M_1(4)$ A_{144}	10.4	10.4	10.1	10.1
$M_2(1)$	11.9	12.4	10.1	10.9
$M_2(2)$	12.5	12.8	10.4	11.1
$M_2(3)$	12.8	12.9	9.6	11.2
$M_2(4)$	12.6	12.6	7.1	8.9
$M_2(5)$	12.2	12.6	10.6	10.7
$M_2(6)$	12.8	12.8	11.3	11.3
$M_2(7)$	12.9	12.8	10.8	11.3
$M_2(8)$	13.7	13.7	11.8	11.8
$M_2(9)$ A_{24}	8.3	8.8	8.8	10.2
$M_2(9)$ A_{144}	8.4	8.8	9.5	10.9
$M_2(10)$ A_{24}	9.1	9.4	8.9	10.1
$M_2(10)$ A_{144}	9.1	9.8	9.6	10.7
$M_2(11)$ A_{24}	8.6	9.2	10.2	10.5
$M_2(11)$ A_{144}	8.7	9.2	10.9	11.5
$M_2(12)$ A_{24}	9.9	9.9	11.0	11.0
$M_2(12)$ A_{144}	10.0	10.0	11.8	11.8

FIGURE 6.7.7 Dependency of the BER-Coder's SNR-performance on the method chosen to optimise the Synthesis Filter. 16 Filter-Optimisation algorithms are considered, and the SNR results are presented for two different Excitation-Adaptation methods.

FILTER OPTIMISATION	$M_2(8)$	$M_2(12)$ A_{144}	$M_1(2)$	$M_1(4)$ A_{144}
EXCITATION ADAPTATION	SNR (dBs)			
$P_2(1)$	13.7	10.0	12.7	10.4
$P_2(2)$	12.1	11.5	9.9	10.1
$P_2(3)$	9.1	9.7	7.8	8.4
$P_2(4)$	14.0	11.0		
$P_2(5)$	11.8	11.8		
$P_2(6)$	13.0	11.4		
$P_2(7)$	11.0	11.0		
$P_2(8)$	13.3	10.6		
$P_2(9)$	11.5	10.2		
$P_2(10)$	11.4	11.4		
$P_2(11)$	10.4	10.7		
$P_2(12)$	10.0	10.7		

FIGURE 6.7.8 Dependency of the BER-Coder's SNR-performance on the method chosen for the Excitation Adaptation. 12 Excitation-Adaptation methods are considered, and the SNR results are presented for four different Filter-Optimisation algorithms.

6.8 Vector Quantization of the Synthesis Filter Coefficients

The quantization of the synthesis filter coefficients is a critical function of the BER coder, because the backward adaptive operation of the synthesis filter sustains a propagation of the quantization errors. Furthermore, the BER coding process (and consequently the quality of the synthesised speech) is more sensitive to the quantization of the coefficients of the FIR filter sections, because the FIR filter coefficients are usually updated more frequently and have a larger dynamic range than the coefficients of the all-pole filter $A(z)$.

A memoryless vector-quantizer (for the FIR filter coefficients) can be designed using various error measures. The criterion chosen here is the "performance" of the BER coder itself. The performance of a BER coder employing a vector codebook (for the quantization of the FIR filter coefficients), is measured by the difference between the original and synthesised speech waveforms. Thus the vector-quantizer is designed to maximise the average performance of the BER coder.

The quantizer optimisation method can be applied to any BER coding system, but it is more suitable for the BER schemes which define an adaptive excitation source. Such schemes allow the excitation source to follow the changing speech characteristics, thus limiting the dynamic range of the filter coefficients during normal (steady-state) operation. If only "fixed" excitation sources are used, then adaptive quantization should be considered and the quantizer optimisation method would need to be modified accordingly.

Assuming that a vector codebook $C=[c_1, c_2, \ldots, c_L]$ of L coefficient sets is available, the BER coder employing that codebook would need to determine the "best" codebook entry (for each frame), by calculating the energy of the signal distortion (difference between the original and synthesised waveforms) for every codebook entry. This calculation can be done either during the optimisation of the synthesis filter (for every possible value of the delay parameters), or after the optimisation of the filter delay parameters. The latter method does not need to quantize the FIR filter coefficients during the optimisation of the synthesis filter, and it is this method that will be used here because it is considerably simpler.

Considering only one FIR section (the process can easily be generalised to include more sections), Eq 6.4.10 can be simplified by dropping the

subscripts of the vectors and matrices involved, and by introducing a new subscript i to denote the index number of the analysis frame. The distortion energy corresponding to the ith frame when the mth codebook entry is used, can then be calculated using the formula :

$$E_i^m = \begin{bmatrix} 1 \\ --- \\ -\mathbf{c}_m \end{bmatrix}^T \left[\begin{array}{c|c} (\mathbf{s}_i-\mathbf{m}_i)^T(\mathbf{s}_i-\mathbf{m}_i) & (\mathbf{s}_i-\mathbf{m}_i)^T\mathbf{Q}_i\mathbf{X}_i \\ \hline \mathbf{X}_i^T\mathbf{Q}_i^T(\mathbf{s}_i-\mathbf{m}_i) & \mathbf{X}_i^T\mathbf{Q}_i^T\mathbf{Q}_i\mathbf{X}_i \end{array}\right] \begin{bmatrix} 1 \\ --- \\ -\mathbf{c}_m \end{bmatrix} \qquad \textit{(Eq 6.8.1)}$$

$$\ldots \text{ for } i=1,2,\ldots,N \text{ and } m=1,2,\ldots,L$$

The vectors $\mathbf{s}_i$ and $\mathbf{m}_i$ correspond to the speech waveform and the transient response of the filter $A(z)$ in the ith frame, $\mathbf{Q}_i$ is the convolution matrix related to the filter $A(z)$ of the same frame, and $\mathbf{X}_i$ is the excitation matrix corresponding to the "optimum" value of the delay parameter in the ith frame. Even though the expression of Eq 6.8.1 seems to concern only the ith frame, it is actually dependent on the outcome of the filter optimisation and excitation adaptation operations in the previous frames. This propagation effect is caused by the backward adaptive operation of the synthesis filter, and it influences the values of the transient response $\mathbf{m}_i$, the excitation matrix $\mathbf{X}_i$ and possibly the convolution matrix $\mathbf{Q}_i$.

The optimum codebook entry for the ith frame is determined by calculating E_i^m for $m=1,2,\ldots,L$ and choosing the entry that minimises the distortion energy. If E_i is the minimum distortion energy corresponding to the ith frame, then a measure of the coder's "performance" can be defined as :

$$P_{\mathbf{C}} = -\sum_{i=1}^{N} g_i E_i \qquad \textit{(Eq 6.8.2)}$$

where g_i are weighting factors, which will later be used to steer the design process towards "subjectively" optimum quantizers.

If C was an initial estimate of the quantizer's codebook, a better codebook $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_L]$ can be designed, that will result a higher performance measurement $P_{\mathbf{V}}$. This can be done by "freezing" (or storing) the various signals obtained during the operation of a BER coder employing the codebook C, and by using these stored signals (in Eq 6.8.1) to calculate the distortion corresponding to a different codebook V. This method circumvents the problem of having to measure the performance of a BER coder employing

the new codebook **V**, and allows us to use a simple algorithm to maximise the value of the performance measure $P_{\mathbf{V}}$. The signal components can be "unfrozen" and updated using the new optimised codebook **V**, but in practice, this updating strategy complicates the optimisation process and offers very little advantage.

The maximisation of the performance measure $P_{\mathbf{V}}$ is achieved by partitioning the summation of Eq 6.8.2 and forming groups corresponding to the individual entries of the former codebook **C** :

$$P_{\mathbf{V}} = \sum_{m=1}^{L} P_{\mathbf{V}}(\mathbf{c}_m, \mathbf{v}_m) \qquad (Eq\ 6.8.3)$$

Each $P_{\mathbf{V}}(\mathbf{c}_m, \mathbf{v}_m)$ term is calculated by grouping together all the frames for which the codebook entry $\mathbf{c}_m$ was originally chosen (during the operation of the BER coder), and by measuring the signal distortion using the new codeword $\mathbf{v}_m$. The value of $P_{\mathbf{V}}$ can now be maximised by maximising each individual term of the summation with respect to the elements of the corresponding codebook entry. Each term of the summation in Eq 6.8.3 can be expressed as a function of the corresponding codeword $\mathbf{v}_m$ as :

$$P_V(\mathbf{c}_m, \mathbf{v}_m) = \begin{bmatrix} 1 \\ \hline -\mathbf{v}_m \end{bmatrix}^T \left[\begin{array}{c|c} \displaystyle\sum_{\substack{i=1 \\ \mathbf{b}_{opt}=\mathbf{c}_m}}^{N} \varepsilon_i (\mathbf{s}_i-\mathbf{m}_i)^T(\mathbf{s}_i-\mathbf{m}_i) & \displaystyle\sum_{\substack{i=1 \\ \mathbf{b}_{opt}=\mathbf{c}_m}}^{N} \varepsilon_i (\mathbf{s}_i-\mathbf{m}_i)^T \mathbf{Q}_i \mathbf{X}_i \\ \hline \displaystyle\sum_{\substack{i=1 \\ \mathbf{b}_{opt}=\mathbf{c}_m}}^{N} \varepsilon_i \mathbf{X}_i^T \mathbf{Q}_i^T (\mathbf{s}_i-\mathbf{m}_i) & \displaystyle\sum_{\substack{i=1 \\ \mathbf{b}_{opt}=\mathbf{c}_m}}^{N} \varepsilon_i \mathbf{X}_i^T \mathbf{Q}_i^T \mathbf{Q}_i \mathbf{X}_i \end{array}\right] \begin{bmatrix} -1 \\ \hline \mathbf{v}_m \end{bmatrix}$$

$$(Eq\ 6.8.4)$$

The summations in Eq 6.8.4 only include the frames for which the optimum set of coefficients ($\mathbf{b}_{opt}$) was initially set equal to the codebook entry $\mathbf{c}_m$. The expression in Eq 6.8.4 is a quadratic function of the elements of the codeword $\mathbf{v}_m$ and is maximised when :

$$\mathbf{v}_m = \left[\sum_{\substack{i=1 \\ \mathbf{b}_{opt}=\mathbf{c}_m}}^{N} \varepsilon_i \mathbf{X}_i^T \mathbf{Q}_i^T \mathbf{Q}_i \mathbf{X}_i \right]^{-1} \left[\sum_{\substack{i=1 \\ \mathbf{b}_{opt}=\mathbf{c}_m}}^{N} \varepsilon_i (\mathbf{s}_i-\mathbf{m}_i)^T \mathbf{Q}_i \mathbf{X}_i \right], \quad m=1,2,\ldots,L$$

$$(Eq\ 6.8.5)$$

By defining the new codebook entries according to Eq 6.8.5, the value of the performance measure P_V is guaranteed to be greater than P_C.

The whole process can be repeated, by using V as the initial codebook estimate. The frames in Eq 6.8.3 can be repartitioned, using Eq 6.8.1 to find the optimum codebook entry $\mathbf{v}_m$ for each frame, and another codebook can be designed using Eq 6.8.5. This iterative process will eventually converge to the optimum codebook of size L. The monotonic increase of the performance measurements is only achieved when the "frozen" signals, obtained from the first codebook C, are used throughout the optimisation process. If these signals are updated periodically (by employing the new codebook and letting the BER system encode the N speech frames), the performance measurements will fluctuate showing an upward trend.

The design of an optimised vector-quantizer for the coefficients of the FIR filter section of a BER coder, is done using a training process arranged in a series of optimisation steps. Initially, the BER coder considered is employed to encode a large set of speech training data. The FIR filter coefficients are left unquantized while the BER coder operates on the speech signal, and the various signals produced (transient response of $A(z)$, excitation sequences, etc.) are stored to be used in the following optimisation of the vector-quantizer codebook. The coefficients of $A(z)$ can be quantized using vector or scalar quantizers.

Eq 6.8.5 is employed in the first optimisation stage, to define a single-entry codebook for the quantizer. In the second step, a second codebook entry is artificially created by adding a small displacement to the only entry of the codebook found in the first step. A succession of two-entry codebooks is then produced by alternately applying Eq 6.8.3 and Eq 6.8.5. The application of Eq 6.8.3 involves the use of Eq 6.8.1, and the partitioning of the frames into groups corresponding to the individual codebook entries.

The series of performance measurements (Eq 6.8.2), corresponding to the series of two-entry codebooks, is monotonically increasing and converges towards an upper limit value. These measurements are based on the stored values of the various signals, that were initially produced from the unquantized FIR filter coefficients. If the stored values of these signals were allowed to be modified at this early stage (by employing the optimised two-

entry codebook), instability problems would occur. That is because a two-entry codebook would introduce large quantization errors during the operation of the BER coder, which could prove to be difficult to correct, taking into cosideration that the BER synthesis filter operates in a backward adaptive mode. For this reason, the updating of the stored sequences should be preferably done when a reasonable sized codebook is available.

In the third optimisation step, two more entries are added to form a four-entry (2-bit) codebook. Again, successive application of equations 6.8.3 and 6.8.5 is used to optimise the four-entry codebook. Further optimisation steps are taken to increase the size of the codebook until it reaches the required size. The same quantizer optimisation method has been extensively used to design waveform quantizers and to reduce the bit rate of vocoders employing the LPC filter model, and is known as the LBG algorithm [6.25]. The designed vector-codebook is unstructured and must be fully searched in order to locate the optimum codebook entry, during coding. The algorithm can be modified to permit the design of structured codebooks (tree, trellis, etc.), which require much less computational effort for the determination of the optimum codebook entry.

The weighting factors ε_i in Eq 6.8.5, are assigned values that give more emphasis to the low-power sections of the speech waveform, in order to reduce the noise level in these sections. This stops the high-power speech intervals from dominating the quantizer optimisation process, and produces vector-quantizers which are "subjectively" more optimum. Extra emphasis is also given to those regions where the quantization error is large, in order to make sure that no error surges occur. The weighting factors are set equal to :

$$\varepsilon_i = \frac{D_i}{T_i} \qquad (Eq\ 6.8.6)$$

where T_i is the energy of the ith speech frame, and D_i is a measure of the distortion energy corresponding to the same frame. The value of D_i is obtained from Eq 6.8.1 during the partitioning of the frames in Eq 6.8.3, and it is subsequently used in Eq 6.8.5 to derive the new codebook entries.

The companding characteristic of a scalar quantizer, designed using the proposed method, is shown in Fig 6.8.1 (thick line). The BER coding system considered, has a synthesis filter with a single FIR filter section that

includes one coefficient and one delay parameter. It uses the $M_1(4)$ filter optimisation algorithm and the $P_2(2)$ excitation adaptation method (modified for the case of a single-input synthesis filter). The speech analysis frame includes 40 samples and the maximum value of the delay parameter is 255 ($0 \leq d_1 \leq 255$). The filter $A(z)$ has 10 coefficients and is estimated using the Maximum-Entropy LPC method over a frame of 240 samples.

The companding characteristic shown in Fig 6.8.1 corresponds to a 7-bit quantizer, but the optimum 4-bit, 5-bit and 6-bit quantizers produce exactly the same curve. The average Segmental-SNR (obtained from a 70 sec long speech training set) corresponding to the different codebook sizes (and bit rates) is given below. The LPC filter $A(z)$ is quantized using scalar quantization of the Log-Area-Ratios with a total of 45 bits per coefficient set.

CODEBOOK SIZE (bits)	*3*	*4*	*5*	*6*	*7*	*8*
SNR (dBs)	*7.19*	*7.76*	*8.22*	*8.37*	*8.43*	*8.43*
BIT RATE (bits/sec)	*3700*	*3900*	*4100*	*4300*	*4500*	*4700*

The SNR values saturate when quantizers with more than 64 levels (6 bits) are used. A 5-bit quantizer is also quite adequate. When 5-bit or smaller sized quantizers are used, the optimised quantizers perform better than uniform or logarithmic quantizers. In Fig 6.8.1, the companding characteristic of a μ-law logarithmic quantizer is also shown (thin line). The parameters of this quantizer were adjusted to approximate the companding curve of the optimised quantizer, and their values were set to μ=10 and v_{max}=6. The two curves in Fig 6.8.1 are similar, especially for small values of the input, but in practice, when fewer than 6 bits are used for the quantization of the FIR filter coefficient, the optimised quantizer improves the quality of the synthesised speech, when compared with the quality obained from a logarithmic quantizer.

Similar results are obtained for the other BER coding schemes that can operate at bit-rates between 4 and 8 kbits/sec. The subjective quality of the encoded speech varies from "good" at 4 kbits/sec, to very good communications quality at 8 kbits/sec. The speech quality at 6 kbits/sec is equivalent to that obtained from other well established Analysis by Synthesis speech coders like CELP.

The vector-quantizer optimisation algorithm can be generalised to include

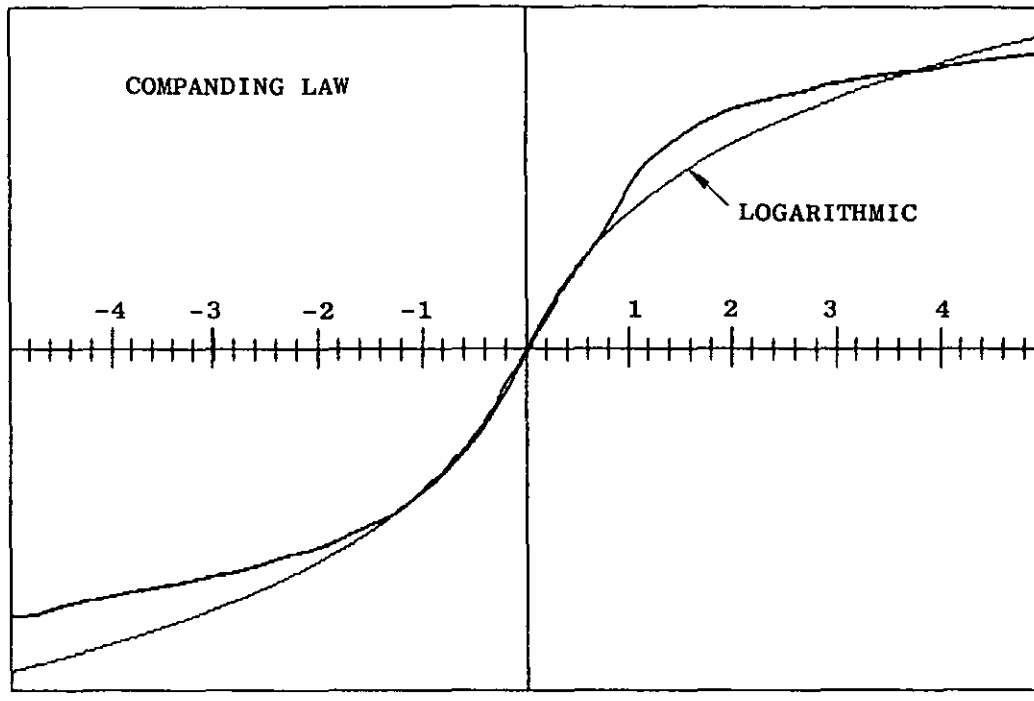


FIGURE 6.8.1 The companding law (thick line) corresponding to a 7-bit Scalar Quantizer, designed for a BER coder that defines a single Adaptive Excitation Source and employs the $M_1(4)$ Filter-Optimisation algorithm, is shown. The Scalar Quantizer was designed using the optimisation method described in Section 6.8.

The companding characteristic of a μ-law logarithmic quantizer (thin line) with μ=10 and v_{max}=6, is shown for comparison.

more than one FIR filter sections. A separate quantizer can then be designed for each section, or a single quantizer can be designed for all the FIR filter coefficients. By using the former method, the quantizers may be included in the filter optimisation process of a BER coder, so that the coefficients of each FIR section are quantized before the next FIR section is optimised. This approach offers the advantage that quantization errors introduced in the early stages of the filter optimisation process, can be partially compensated for by the following optimisation stages.

6.9 Conclusions

The proposed BER coder operates differently from conventional speech coders which employ the source-filter speech model, in that it defines the excitation source in a backward adaptive manner, and relies solely on the adaptation of the synthesis filter parameters to reconstruct the speech waveform. The adaptation of the stored excitation sequences and the optimisation of the various sections of the synthesis filter, can be done in a number of ways. As a result, a multitude of BER coding schemes with a wide range of properties can be defined.

The excitation sequences employed by a BER coder may have speech-like or noise-like charactersitics, and may or may not be allowed to adapt to the changing properties of the speech signal. The estimation of the synthesis filter coefficients can be done using either the Linear Prediction or the Distortion Minimisation methods, and the coefficients of the various filter sections may be estimated jointly or separately. The filter optimisation and the excitation adaptation are two separate operations in a BER coding system, and the various algorithms that perform these two functions can be combined in many different ways. Some of these combinations can be more successful than others.

The BER coding schemes that define an excitation source with speech-like characteristics usually perform better than the BER systems that define a noise-like excitation. The two types of excitation can be mixed to produce an even better BER coding system. Many BER coding schemes are able to operate with very small frames, thus reducing the encoding delay to a value as low as 3 ms. This low delay property can become a valuable asset in speech coding applications where the system's overall coding delay must be kept to a minimum.

A BER coder can take many forms, and one configuration in particular corresponds to a Code Excited LPC coder that employs an excitation codebook with overlapping entries. Another configuration corresponds to the Self-Excited Vocoder. These two schemes can be considered as special cases of the general BER coder, and there are other configurations of the BER coder that perform better than either of them.

The BER coder in most cases shows a remarkable ability to adapt to the changing characteristics of the speech waveform, in spite of the fact that its backward adaptive operation slows down its response to signal transitions. This ability is preserved even at low transmission bit rates, and various BER coding schemes can operate successfully at bit rates between 4 and 8 kbits/sec. The speech quality at 4 kbits/sec can be judged as "good", while at 8 kbits/sec very good communications quality speech can be achieved.

BER coders are sensitive to quantization errors and require an accurate quantization of the coefficients of the FIR filter sections. A method that can be used to design "optimum" vector-quantizers for the FIR filter coefficients has been presented, which is based on the minimisation of the average distortion introduced by the BER coding process. Scalar quantizers optimised using the proposed method, perform better than uniform and logarithmic quantizers, when the number of bits allocated to the quantization of each FIR filter coefficient is less than 6.

The backward adaptation of the excitation sequences increases the sensitivity of a general BER coder to transmission errors. This sensitivity can be reduced by including standard reinitialisation procedures, or by using a "fixed" sequence as one of the excitation sources employed by the BER coder.

REFERENCES

[6.1] M.R.Schroeder, B.S.Atal, "Code-Excited Linear Prediction (CELP): High quality speech at very Low Bit Rates", Proc ICASSP 1985, pp 937

[6.2] P.Kroon, B.S.Atal, " Strategies for improving the performance of CELP coders at Low Bit Rates", IEEE Proc ICASSP, New York 1988, pp 151

[6.3] N.Gouvianakis, C.Xydeas, "BER-LPC Speech Coding Schemes at 4 to 9.6 kbits/sec", Final Report, British Telecom Research Laboratories, Contract No 610249, April 1986

[6.4] N.Gouvianakis, C.Xydeas, "Advances in Analysis by Synthesis LPC Speech Coders", IERE Journal, Supplement on Mobile Radio, Vol 57, No 6, Nov/Dec 1987, pp S272-S286

[6.5] L.Ljung, T.Soderstrom, "Theory and Practice of Recursive Identification", MIT Press, Cambridge, Mass., 1983

[6.6] J.V.Candy, "Signal Processing: The Model-Based Approach", McGraw-Hill Book Company, New York 1986

[6.7] Y.Monden et al, "Fast Algorithm for Identification of an ARX Model and its Order Determination", IEEE Trans. ASSP, Vol 30, No 3, June 1982, pp 390-399

[6.8] Y.F.Huang, A.K.Rao, "Application of a Recursive Estimation Algorithm with Information-Dependent Updating to ARMAX Models and ARMA models with Unknown Inputs", IEEE Proc. ICASSP, Dallas 1987, pp 1007-1010

[6.9] S.Singhal, B.S.Atal, "Optimizing LPC parameters for Multi-Pulse Excitation", IEEE Proc. ICASSP, 1983, pp 781

[6.10] J.Picone et al, "Joint Estimation of the LPC parameters and the Multi-Pulse Excitation", Speech Communications, Vol 5 (1986), pp 253

[6.11] D.T.Lee et al, "Recursive Least Squares Ladder Estimation Algorithms" IEEE Trans. on Circuits and Systems, Vol CAS-26, No 6, June 1981 pp 467-481

[6.12] A.Benveniste, C.Chaure, "AR and ARMA Identification Algorithms of Levinson Type: An Innovations Approach", IEEE Trans. on Automatic Control", Vol AC-26, No 6, December 1981, pp 1243-1261

[6.13] B.Friedlander, "Lattice Filters for Adaptive Processing", Proceedings of the IEEE, Vol 70, No 8, August 1982, pp 829-867

[6.14] D.T.Lee, "Recursive Ladder Algorithms for ARMA Modeling", IEEE Trans. on Automatic Control, Vol AC-27, No 4, August 1982, pp 753-764

[6.15] H.Lev-Ari et al, "Least-Squares Adaptive Lattice and Transversal

Filters: A Unified Geometric Theory", IEEE Trans. on Information Theory, Vol IT-30, No 2, March 1984, pp 222-236

[6.16] E.Karlsson, M.H.Hayes, "Least Squares ARMA Modeling of Linear Time-Varying Systems: Lattice Filter Structures and Fast RLS Algorithms", IEEE Trans. ASSP, Vol 35, No 7, July 1987, pp 994-1014

[6.17] R.M.Johnson, "Theory and Applications of Linear Differential and Difference Equations: A Systems Approach in Engineering", Ellis Horwood Ltd., Chichester 1984, John Wiley & Sons Inc.

[6.18] W.B.Kleijn et al, "Improved Speech Quality and Efficient Vector Quantization in SELP", IEEE Proc ICASSP, New York 1988, pp 155-158

[6.19] P.Kroon, E.F.Deprettere, "A Class of Analysis-by-Synthesis Predictive Coders for High Quality speech coding at rates between 4.8 and 16 kbits/sec", IEEE Journal on Selected Areas in Communications, Vol 6, No 2, Feb 88, pp 353-363

[6.20] G.Ohyama, "A Novel Approach to Estimating Excitation Code in Code-Excited Linear Prediction Coding", IEEE Proc. ICASSP, 1986, pp 3067

[6.21] D.Lin, "Speech Coding Using Efficient Pseudo-Stochastic Block Codes", IEEE Proc. ICASSP, Dallas 1987, pp 1354-1357

[6.22] R.C.Rose, T.P.Barnwell, "The Self Excited Vocoder - An alternative approach to Toll Quality at 4800 bps",IEEE Proc ICASSP, Tokyo 1986, pp 453-456

[6.23] T.P.Barnwell et al, "A Real-Time Implementation of a 4800 bps Self Excited Vocoder using the AT&T WE-DSP32 Signal Processing Micro-processor", Speech Tech 1987, pp 263-267

[6.24] K.Nayebi et al, "Analysis of the Self-Excited Subband Coder: A new approach to Medium Band Speech Coding", IEEE Proc. ICASSP, New York 1988, pp 390-393

[6.25] Y.Linde et al, "An Algorithm for Vector Quantizer Design", IEEE Trans on Communications, Vol 28, No 1, January 1980, pp 84-95

RECAPITULATION AND SUGGESTIONS FOR FURTHER RESEARCH

When considering an application that requires Digital-Coding of speech, one is faced with the task of choosing the "best" digital coder for the particular application. Various conditions usually have to be met by the coder, such as robustness in the presence of channel errors, low delay characteristics, and low cost of implementation. In addition, the coder must be able to reproduce speech with quality which is acceptable to the user.

The technology to achieve high speech quality is already well developed for bit rates above 16 kbits/sec. Today, the major research activity is focused at lowering the bit rate to 4.8 kbits/sec without degrading speech quality. Already, a new class of speech coding methods has produced very good results at 10 kbits/sec, and the trend is for even lower bit rates. The new methods use existing models of speech but employ complex algorithms, known as Analysis-by-Synthesis (AbS), to optimise the model parameters. The recent availabilty of powerful Digital Signal Processors has simplified the task of implementing such complex Speech-Coding algorithms.

This "algorithmic" approach has also been followed in this thesis, in trying to develop new and efficient speech coding algorithms, based on existing models derived from Linear Prediction Theory. Various AbS speech coding algorithms have been studied and compared in terms of their performance (SNR) and their complexity (number of operations). The SNR was chosen to measure the coders' performance because it can yield meaningful results when coders of the same "nature" are compared. It was found that when "similar" coders were compared, the results of listening tests agreed with the ranking obtained by using SNR measurements.

Multipulse Excitation (MPE) algorithms were considered in Chapter 3, where a general classification of the MPE optimisation algorithms was presented. Some of the algorithms mentioned, are highly complex and can be used to find the upper limit in the performance of conventional MPE coders. Simpler MPE algorithms were described in Chapter 4, that can be implemented in real-time using the currently available VLSI technology. The MPE algorithms were compared at various pulse-rates, and it was found that a significant advantage (in terms of speech quality) can be gained at high pulse rates by using more "sophisticated" MPE coding methods with a moderate increase in complexity.

A fast implementation of a complex MPE algorithm (method MS5) was developed in Chapter 4, based on the Gram-Schmidt orthogonalization procedure. Method MS5 gives very good results, and is still simple enough to allow real-time implementation. The proposed Block-Search methods were also found to have low-complexity and high performance characteristics, comparable to that of the "best" Multi-Stage MPE systems. One of the Block-Search methods in particular (Method BS1) gave the best results out of all the methods examined in Chapter 4.

One further conclusion was that the presence of the noise-shaping filter helps to improve the efficiency of the pulse search procedure, when simple MPE optimisation algorithms are used. The overall effect is to improve the performance of the simple algorithms, while the performance of the complex algorithms remains almost unaffected. The effectiveness of the noise shaping filter in shaping the noise spectrum was found to be small at low pulse rates.

MPE coders can produce very good communications quality speech at a bit rate of 9.6 kbits/sec. At lower bit rates, efficient quantization methods must be used to avoid loss of speech quality. A process that designs "optimum" scalar quantizers for the pulse amplitudes is proposed in Chapter 4. The method defines a Gamma-PDF model which is based on the experimental-PDF data, and takes into account the dependency of the PDF on the pulse rate. Quantizers based on the Gamma-PDF model perform better than optimised-uniform quantizers, when the number of bits per pulse is small.

An efficient method for coding the pulse positions in a MPE coding system, has been proposed in Chapter 5. A codebook of position-patterns is employed by the MPE coder, so that the pulse positions are specified by transmitting the index of a codebook entry. It was found that by using a position-codebook, the number of bits required for the coding of the pulse positions is approximately one third of the number of bits required by a conventional MPE coder. The released bits can be allocated to other parts of the MPE coder, thus improving its performance. Alternatively, the bit rate of the MPE coder can be reduced without affecting the quality of the recovered speech.

A fast algorithm was proposed to design "unstructured" position-codebooks, based on the maximisation of a weighted-SNR measure. The designed

codebooks were compared to random codebooks, and they were found to perform better, resulting a greater efficiency in the coding of the pulse positions. It also became clear that the complexity of the CS-MPE coder can be reduced by introducing structure into the codebooks, and an example from the use of a random-tree-codebook was given. It is reasonable to assume that a "design" process similar to the one used for the unstructured codebook, can be used to define tree-codebooks with better properties than random-tree-codebooks. Large tree-codebooks could also be designed (since large unstructured codebooks are impractical) that would permit the use of larger MPE frames, and would improve the performance of the CS-MPE coder at bit rates below 8 kbits/sec.

The codebook "design" process can generally be used in speech coders which model the speech or the excitation signal with a weighted sum of "primary waveforms". MPE coders for example, employ a set of primary waveforms (pulses) to model the excitation. Sinusoidal coders on the other hand, may employ a set of sine-waves to model speech. The same approach can therefore be used to design codebooks for a number of speech coders, so that each entry of the codebook specifies a different combination of primary waveforms. Further research must be carried out to determine whether this codebook approach can improve the performance of other speech coding systems. Another question that also needs to be answered is whether the codebook design process can be combined with another process that "designs" the primary waveforms.

In Chapter 6, the proposed BER coders where examined. Various algorithms were proposed for the synthesis-filter optimisation and the backward excitation-adaptation. It was found that a BER system which defines the excitation from the past synthesised speech samples can give very good results, and can operate with very small encoding delays of the order of 3 ms. Two special cases of the BER coder were found to be the CELP coder (with overlapping codebook entries) and the Self-Excited Vocoder. Many other BER coding schemes were also investigated which performed better than the CELP and Self-Excited coders.

BER coders can produce good quality speech at 4.8 kbits/sec, and very good communications quality speech at 8 kbits/sec. At the lower bit rates more efficient quantization procedures must be used in order to avoid loss of speech quality. A process was proposed in Chapter 6, that designs a Vector

Quantizer for the FIR-section coefficients of the BER synthesis filter. The process uses a measure of the coder's performance, and designs the vector quantizer by maximising the performance measure. It was found (for the case of a scalar quantizer) that a quantizer designed by using the proposed method, can be more efficient than logarithmic quantizers when 5 bits or less are allocated for the quantization of each coefficient.

The BER schemes can be simplified by exploiting the structure of the matrices derived in the filter calculations. Even though many BER shemes can be implemented in real-time at 4.8 kbits/sec, operation at higher bit rates is accompanied by a substantial increase in complexity. Further research must therefore be carried out to simplify the BER algorithms employed at 6 kbits/sec and 8 kbits/sec. The application of the proposed Vector Quantization process must also be studied at these higher bit rates.

NOTE ON PUBLICATIONS

The Block-Search MPE optimisation algorithms presented in Chapter 4, were also included in :

N. Gouvianakis, C. Xydeas, "A Multipulse Excited LPC coder implementation based on a Block Solution approach", Proc. Int. Conf. Digital Processing of Signals in Communications, IERE, Loughborough 1985, pp 85-92

N. Gouvianakis, C. Xydeas, "A Comparative Study of Multistage Sequential and Block Sequential Search Multipulse LPC algorithms", Proc. Int. Conf. IASTED, Paris, Jun 1985, pp 122-125

N. Gouvianakis, C. Xydeas, U.K. Patent Application No. 8508669 and 8515501

The BER coding algorithms presented in Chapter 6, were also included in :

N. Gouvianakis, C. Xydeas, "Advances in Analysis by Synthesis LPC Speech Coders", IERE Journal, Supplement on Mobile Radio, Vol 57, No 6, Nov/Dec 1987, pp S272-S286

N. Gouvianakis, C. Xydeas, U.K. Patent Application No. 8720388

APPENDIX A

The one-sided z-transform equivalent of Eq 6.3.6 is :

$$S_y(z) = \sum_{m=1}^{l} a_m \left[S_y(z)\ z^{-m} + \sum_{k=o}^{m-1} s_y(k-m)\ z^{-k} \right] +$$

$$+ \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \left[U_j(z)\ z^{-n-d_j-k} + \sum_{i=o}^{n+d_j+k-1} u_j(i-n-d_j-k)\ z^{-i} \right]$$

(Eq A.1)

A rearrangement of the terms of Eq A.1 produces the equation :

$$S_y(z) \left[1 - \sum_{m=1}^{l} a_m\ z^{-m} \right] = \sum_{m=1}^{l} a_m \sum_{k=o}^{m-1} s_y(k-m)\ z^{-m} + \sum_{j=1}^{n_b} U_j(z) \sum_{k=o}^{q_j} b_k^j\ z^{-n-d_j-k} +$$

$$+ \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n+d_j+k-1} u_j(i-n-d_j-k)\ z^{-i}$$

(Eq A.2)

By rearranging the summation terms of the first term in the right-hand side of Eq A.2, and by using equations 6.2.1 and 6.2.2, Eq A.2 is transformed to the equivalent equation :

$$S_y(z) = A(z) \sum_{k=o}^{l-1} z^{-k} \sum_{m=1}^{l-k} a_{k+m}\ s_y(-m) + A(z) \sum_{j=1}^{n_b} U_j(z)\ B_j(z) +$$

$$+ A(z) \sum_{j=1}^{n_b} \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n+d_j+k-1} u_j(i-n-d_j-k)\ z^{-i}$$

(Eq A.3)

which gives the equations 6.4.2 and 6.4.3.

APPENDIX B

Using Eq 6.4.8, the energy of the signal distortion is found to be :

$$\mathbf{e}_a^T \mathbf{e}_a = \left[\mathbf{s}-\mathbf{m}_y - \sum_{j=1}^{n_b} \mathbf{Q}\,\mathbf{U}_j\mathbf{b}_j \right]^T \left[\mathbf{s}-\mathbf{m}_y - \sum_{j=1}^{n_b} \mathbf{Q}\,\mathbf{U}_j\mathbf{b}_j \right] \qquad (Eq\ B.1)$$

or equivalently :

$$\mathbf{e}_a^T \mathbf{e}_a = (\mathbf{s}-\mathbf{m}_y)^T(\mathbf{s}-\mathbf{m}_y) - (\mathbf{s}-\mathbf{m}_y)^T \left[\sum_{j=1}^{n_b} \mathbf{Q}\,\mathbf{U}_j\mathbf{b}_j \right] - \left[\sum_{j=1}^{n_b} \mathbf{b}_j^T\mathbf{U}_j^T\mathbf{Q}^T \right] (\mathbf{s}-\mathbf{m}_y) +$$

$$+ \left[\sum_{j=1}^{n_b} \mathbf{Q}\,\mathbf{U}_j\mathbf{b}_j \right] \left[\sum_{j=1}^{n_b} \mathbf{b}_j^T\mathbf{U}_j^T\mathbf{Q}^T \right] \qquad (Eq\ B.2)$$

The right-hand side of Eq B.2 can be rewritten as product of two composite matrices :

$$\mathbf{e}_a^T \mathbf{e}_a = \left[\mathbf{s}-\mathbf{m}_y \;\vdots\; -\mathbf{Q}\mathbf{X}_1\mathbf{b}_1 \;\vdots\; \cdots \;\vdots\; -\mathbf{Q}\mathbf{X}_{n_b}\mathbf{b}_{n_b} \right]^T \left[\mathbf{s}-\mathbf{m}_y \;\vdots\; -\mathbf{Q}\mathbf{X}_1\mathbf{b}_1 \;\vdots\; \cdots \;\vdots\; -\mathbf{Q}\mathbf{X}_{n_b}\mathbf{b}_{n_b} \right]$$

$$(Eq\ B.3)$$

By separating the filter coefficients in Eq B.3, the final result is obtained :

$$\mathbf{e}_a^T\mathbf{e}_a = \begin{bmatrix} 1 \\ \hline -\mathbf{b}_1 \\ \hline \vdots \\ \hline -\mathbf{b}_{n_b} \end{bmatrix}^T \left[\mathbf{s}-\mathbf{m}_y \;\vdots\; \mathbf{Q}\mathbf{X}_1\mathbf{b}_1 \;\vdots\; \cdots \;\vdots\; \mathbf{Q}\mathbf{X}_{n_b}\mathbf{b}_{n_b} \right]^T \left[\mathbf{s}-\mathbf{m}_y \;\vdots\; \mathbf{Q}\mathbf{X}_1\mathbf{b}_1 \;\vdots\; \cdots \;\vdots\; \mathbf{Q}\mathbf{X}_{n_b}\mathbf{b}_{n_b} \right] \begin{bmatrix} 1 \\ \hline -\mathbf{b}_1 \\ \hline \vdots \\ \hline -\mathbf{b}_{n_b} \end{bmatrix}$$

$$(Eq\ B.4)$$

Equation B.4 is clearly the same as Eq 6.4.10.

APPENDIX C

Equation 6.6.1.2 is transformed in the one-sided z-transform domain as :

$$U_j(z) = \sum_{k=o}^{q_j} b_k^j \left[U_j(z)\ z^{-n-d_j-k} + \sum_{i=o}^{n+d_j+k-1} u_j(i-n-d_j-k)\ z^{-i} \right] + U_{j-1}(z)$$

(Eq C.1)

Rearranging the terms of Eq C.1 gives :

$$U_j(z) \left[1 - \sum_{k=o}^{q_j} b_k^j\ z^{-n-d_j-k} \right] = \sum_{k=o}^{q_j} b_k^j \sum_{i=o}^{n+d_j+k-1} u_j(i-n-d_j-k)\ z^{-i} + U_{j-1}(z)$$

(Eq C.2)

Using Eq 6.2.1, Eq C.2 is transformed to Eq 6.6.1.3