TIME AND FREQUENCY DOMAIN ALGORITHMS

FOR SPEECH CODING

by

Francis Song Chian YEOH,

B. Sc. (Loughborough University, England)

A Doctoral Thesis submitted in partial fulfilment of
the requirements for the award of Doctor of Philosophy
of the Loughborough University of Technology
October, 1983

SUPERVISOR: Costas S. Xydeas,
M. Sc., Ph. D, M.I.E.E., M.I.E.R.E., M.I.O.A.
Department of Electronic and Electrical Engineering

To my parents

君子以自強不息

"The destiny of man lies in his unremitting efforts to constantly strive for improvement."

- Confucius (Book of Changes)

SYNOPSIS

The promise of digital hardware economies (due to recent advances in VLSI technology), has focussed much attention on more complex and sophisticated speech coding algorithms which offer improved quality at relatively low bit rates.

This thesis describes the results (obtained from computer simulations) of research into various efficient (time and frequency domain) speech encoders operating at a transmission bit rate of 16 Kbps.

In the time domain, Adaptive Differential Pulse Code Modulation (ADPCM) systems employing both forward and backward adaptive prediction were examined. A number of algorithms were proposed and evaluated, including several variants of the Stochastic Approximation Predictor (SAP). A Backward Block Adaptive (BBA) predictor was also developed and found to outperform the conventional stochastic methods, even though its complexity in terms of signal processing requirements is lower. A simplified Adaptive Predictive Coder (APC) employing a single tap pitch predictor considered next provided a slight improvement in performance over ADPCM, but with rather greater complexity.

The ultimate test of any speech coding system is the perceptual performance of the received speech. Recent research has indicated that this may be enhanced by suitable control of the noise spectrum according to the theory of auditory masking. Various noise shaping ADPCM configurations were examined, and it was demonstrated that a proposed pre-/post-filtering arrangement which exploits advantageously the predictor-quantizer interaction, leads to the best subjective

performance in both forward and backward prediction systems.

Adaptive quantization is instrumental to the performance of ADPCM systems. Both the forward adaptive quantizer (AQF) and the backward one-word memory adaptation (AQJ) were examined. In addition, a novel method of decreasing quantization noise in ADPCM-AQJ coders, which involves the application of correction to the decoded speech samples, provided reduced output noise across the spectrum, with considerable high frequency noise suppression.

More powerful (and inevitably more complex) frequency domain speech coders such as the Adaptive Transform Coder (ATC) and the Sub-band Coder (SBC) offer good quality speech at 16 Kbps. To reduce complexity and coding delay, whilst retaining the advantage of sub-band coding, a novel transform based split-band coder (TSBC) was developed and found to compare closely in performance with the SBC.

To prevent the heavy side information requirement associated with a large number of bands in split-band coding schemes from impairing coding accuracy, without forgoing the efficiency provided by adaptive bit allocation, a method employing AQJs to code the sub-band signals together with vector quantization of the bit allocation patterns was also proposed.

Finally, 'pipeline' methods of bit allocation and step size estimation (using the Fast Fourier Transform (FFT) on the input signal) were examined. Such methods, although less accurate, are nevertheless useful in limiting coding delay associated with SBC schemes employing Quadrature Mirror Filters (QMF).

ACKNOWLEDGMENTS

It is my pleasure to express my utmost gratitude to Dr. Costas Xydeas, whom I am privileged to have as my supervisor, for his continuous guidance and inspiration throughout the course of this research. His extensive knowledge and experience in the field of digital speech coding have been invaluable to the development of the work, while his unceasing enthusiasm and optimism were a tremendous encouragement.

I am also grateful to British Telecom Research Laboratories for providing the necessary financial support for the project. In particular, I would like to express my thanks to Mr. Chris Wheddon for initialising the research contract, Mr. Fred Westall, Dr. Roger Hanes, Mr. Adrian Farrell and others in R18.3.1, for their interest in the work and their helpful comments and suggestions. I am thankful too, for the assistance rendered by R18.3.1 in carrying out the informal listening tests and obtaining the spectrograms for the work presented in chapter four of the thesis.

My thanks are also due to Professors I.R. Smith and J.W.R. Griffiths, present and former heads of the the Department of Electronic and Electrical Engineering, for providing the necessary research facilities. I am grateful too for the kind assistance in various matters of members of academic staff of this department, as well as the technicians and the ladies of the general office. The staff of the University's Computer Centre deserves special mention for their willingness and readiness to help with problems in computing - they are certainly a credit to the university. Mr. Graham Gerrard and Mr. Geoff Harris have been

especially helpful beyond the call of duty.

My many colleagues in the lab and department have contributed to a cheerful and conducive environment for research. Apart from the numerous enlightening (and entertaining!) debates (over coffee) on subjects as diverse as politics and car maintenance, I have also benefitted immensely from the many fruitful discussions on technical matters and the mutual exchange of ideas. Dr. T.C. Kok, Mr. S.N. Koh and Mr. W.K. Cham have been particularly helpful in their advice and assistance.

I am greatly indebted to my beloved parents who have given so sacrificially for my education all these years while I have been away from home, and who have taught me many of life's important lessons. My brothers and sisters too, have been unfailing in their expression of affection and support. My youngest sister Emily, in particular, has been a source of much joy and comfort. I wish to thank also my brother-in-law, Mr. Yeo Siew Khim for his excellent Chinese calligraphy. I am deeply grateful to Miss Karol Tong for her moral support and her assistance in many little ways.

My long stay in Loughborough and England has been a pleasant and memorable experience, due largely to the many good friends I have made over the years. My deepest appreciation goes to these wonderful friends of mine, too numerous to mention, for their constant love, concern, support and encouragement, as well as to those friends far away who have continued to write faithfully to me despite the barriers of time and distance. God bless you all!

LIST OF PRINCIPAL SYMBOLS

$x(n), x_n$	: input speech sample
$\hat{x}(n), \hat{x}_n$	: recovered speech sample
$x(t)$	: input speech waveform
$\hat{x}(t)$	: recovered speech waveform
$\{x_n\}, \{x(n)\}$	: sequence of x(n) samples
a_k	: kth predictor coefficient
p	: order of predictor
$\tilde{x}(n)$	: predicted speech sample
A_{opt}	: optimum (mmse) predictor coefficients vector
R	: autocorrelation matrix in LPC analysis
	: total number of bits for SBC/ATC
C	: autocorrelation vector in LPC analysis
$y(n)$	: locally decoded DPCM sample
$e(n)$	: prediction residual sample
$q(n)$	: quantization noise sample
Δ	: quantizer step-size
$\rho(i), R(i)$	: ith autocorrelation coefficient
$E[x]$	: expectation of x
$<.>$	: average value
G_{pcm}	: gain of DPCM over PCM
$w(n)$	: data window sample
Φ	: covariance matrix
k_m	: mth reflection coefficient
g_m	: mth log area coefficient
$A(n)$	: predictor coefficients vector
$K(n)$	: gain vector for Kalman predictor

List of Symbols

$V_a(n)$	: symmetrical matrix used in Kalman prediction
V_v	: scalar constant
V_w	: symmetrical matrix of noise terms
I	: identity matrix
f_m	: mth forward residual of lattice predictor
b_m	: mth backward residual of lattice predictor
C_m	: mth partial sum (numerator) used in lattice prediction
D_m	: mth partial sum (denominator) used in lattice prediction
g	: constant used in sequential predictors
ε	: small quantity
N	: blocksize of prediction
	: blocksize of cosine transform
C(p)	: p-shift autocorrelation function
M	: pitch period
	: no. of samples between BBA predictor update
G_1, G_2	: optimising constants
B	: number of quantizer bits
$x_c(n)$	: compressed speech sample
V	: maximum amplitude of PCM quantizer
p(x)	: probability of x
σ	: standard deviation
α	: scaling constant
	: noise shaping factor
	: multiplier value
M(.)	: time invariant multiplier function
c	: gain constant
β	: scaling constant
	: quantizer leakage factor

List of Symbols

f_s	: sampling frequency
f_c	: signal bandwidth
T	: sampling period
	: number of FIR filter taps
b_k	: kth coefficient of noise feedback filter
W(f)	: weighting function in frequency domain
$f_i(n)$	: quantizer correction factor
$\hat{x}(n)$	: corrected decoded sample
b	: number of bands in sub-band coder
h1(n)	: high-pass QMF used in sub-band coder
h2(n)	: low-pass QMF used in sub-band coder
R_i	: number of bits allocated to ith frequency component
$\bar{R}$	: average bit rate
a(i)	: correlation coefficient for ith sub-band
G_{tc}	: gain of transform coder over PCM
$X_c(k)$	: cosine transform components
X_n	: input signal vector
Y_n	: cosine transform of X_n
B_N	: cosine transform basis matrix
β_N	: NxN square matrix
A	: blocksize of parameter update
u(t)	: vocal tract impulse response
e(t)	: excitation waveform
C(t)	: cepstrum
H(S)	: source entropy
S(n,k)	: short-time signal spectrum
R(n,k)	: short-time residual spectrum
$X(\omega)$	: Fourier transform of x(n)

C O N T E N T S

Contents (cont.)

Contents (cont.)

Contents (cont.)

Contents (cont.)

Contents (cont.)

Contents (cont.)

Contents (cont.)

CHAPTER ONE INTRODUCTION

1.1 COMMUNICATION BY SPEECH

Communication is essentially a social affair. Man has evolved a host of different systems which render his social life possible - social life not in the sense of living in packs for hunting or for making war, but in a sense unknown to the lower animals[1]. Most prominent among all these systems of communication is of course human speech and language. Indeed, man is unique among all life forms in this world, in his ability to acquire and use speech. Human language is not to be equated with the sign systems of animals, for man is not restricted merely to calling his young, or suggesting mating, or shouting cries of danger; he can with his remarkable facilities of speech give utterance to almost any thought. Like animals, we too have our inborn instinctive cries of alarm, pain, etc.; we say 'oh' or 'ah'; we have smiles, groans and tears; we blush, shiver, yawn and frown, but such reflexes do not form part of the true human language. A hen can set her chicks scurrying up to her by clucking - communication established by a release mechanism - but human language is vastly more than a complicated system of 'clucking'.

Because man lives in an air atmosphere, it is not surprising that he learned to communicate by producing longitudinal vibrations (acoustic waves) in the air medium[2,3]. At the acoustic level, speech consists of rapid and deterministic fluctuations in air pressure. These sound pressures are generated and radiated by man's vocal apparatus, they are

detected by his ear and apprehended by his brain.

The specialised code of speech did not develope overnight. Passage of untold time probably witnessed the gradual evolution of human speech from the grunts and barks of man's fellow creatures to the level of sophistication we know today. The earliest forms of communication were probably mainly tactile and visual[4]. At least one speculation holds that man's first means of communication were probably hand signals - speech perhaps evolved when man discovered he could supplement his hand signals by audible and distinctive gestures of his vocal tract. As Sir Richard Paget puts it, "It was the continual use of man's hands for craftsmanship, the chase, and the beginnings of art and agriculture that drove him to find other methods of expressing his ideas - namely, by a specialised pantomime of tongue and lips."[5]

Speech and writing are by no means our only systems of communication. Social intercourse is greatly strengthened by habits of gesture - little movements of the hands and face, or the so-called 'body language'. With nods, smiles, frowns, handshakes, kisses, fist shakes and other gestures, we can convey the most subtle understanding[6]. However, life in the modern world is coming to depend more and more upon 'technical' means of communication - telephone and telegraph, radio and printing. Without such technical aids, the modern city-state could not exist one week, for it is only by means of them that trade and business can proceed, transport systems run on schedule, that law and order are maintained and education is possible. Communication renders true social life practicable, for communication means organisation. Communication engineers have altered the size and shape of the world[1].

From time immemorial, man has sought to communicate over distances by various means - by the beat of drums, by beacons on hill-tops, by carrier pigeons and by coded flag signals. For example, long and short smoke signals were used by the Red Indians, high and low pitch drums by African tribesmen[4]. History records that in the sixth century B.C., Cyrus the Great of Persia is supposed to have established lines of signal towers on high hill-tops, radiating in several directions from his capital. On these vantage points, he stationed 'leather lunged' men who shouted messages along, one to another. Similar 'voice towers' were reportedly used by Julius Caesar in Gaul[2], as well as by the ancient Chinese, who used such 'voice transmission systems' to herald the arrival of the emperor.

Despite the desires and motivations to accomplish communication at long distances, it was not until man learned to generate, control and convey electrical current that telephony could be brought into the realm of reality. In 1876, the invention of the telephone by Bell[7] made conversations at a distance far beyond the range of the human voice possible for the first time. Its use spread rapidly, and over the years, the laying of telephone cables across the continents and along ocean floors has enabled conversations to be carried out between almost any two parts of the earth.

Basically, the telephone converts an acoustical signal by means of transducers into an electrical signal which can be transmitted over long distances along wires at a very high speed (the speed of light). At the destination (or receiver), this electrical signal is re-converted back to acoustical energy to yield a close replica of the original waveform. The communication engineer is primarily concerned with efficient

communication i.e. the transmitting of messages (or information) between two points over a channel as quickly as possible and with minimum error [8]. Numerous communication systems have evolved over the years since the advent of telephony, each new development usually being an attempt to improve on the efficiency of its predecessor.

Until recently, most communication systems have been concerned with the transmission of continuous or analogue signals which can take on an infinite number of variations. In contrast, one can conceive of a system which involves the transmission of one of a finite number of waveform elements or messages. A simple example of this is observed in the transmission of an English text using the Morse code. Here, the problem of transmission is reduced to one of transmitting a sequence of messages, each of which is selected from a specified and finite set. This type of communication is termed 'digital communication'[8].

1.2 DIGITAL SPEECH COMMUNICATION

Digital communication systems therefore involve the transmission and detection of one of a finite set of known waveforms (or digital data), as opposed to analogue systems, where an infinitely large number of messages exist and the corresponding waveforms are not at all known. Pulse code modulation (PCM[9]) is an example of a digital communication technique used to transmit continuous data. The transformation from analogue to digital data is made possible by the process of quantization which essentially approximates the continuous signals so that they assume only certain discrete amplitudes. This is the process of digitising the data, which can now be transmitted by a finite number of symbols (or levels). Digital methods of speech coding have been

proposed more than three decades ago, but only attracted serious attention and interest during the era of the transistor. However, this interest has since intensified and accelerated virtually without bounds, fueled by the advances in transistor technology, switching circuits, the advent of the microprocessor and important breakthroughs in device technology - VLSI (very large scale integration) and CCDs (charged coupled devices). Presently, digital techniques are entering telecommunication networks very quickly[10] - massive investments have been made in digital transmission systems around the world in recent years. It is envisaged that by the turn of the century, if not sooner, most existing telecommunication networks would have gone fully 'digital'.

The reasons for this overwhelming interest in digital speech communication are numerous. A few of the more commonly advanced advantages associated with digitising speech (and other types of information) will be briefly considered[11].

(1) Digital encoding is able to provide for the transmission of information over long distances and varying network topology with minimal degradation to speech quality,since digital signals can be accurately regenerated by repeaters placed along the transmission path.

(2) Time division multiplexing (TDM) can be applied very simply and cheaply to telephone transmission lines and switching devices, using economic digital circuitry, thereby increasing channel capacity. In contrast, frequency division multiplexing (FDM) techniques employed in analogue transmission systems are considerably more expensive, requiring the use of complex filters.

(3) Different types of signals can be encoded to a uniform digital for-

mat and transmitted over the same communication system. Thus a digital system can handle a variety of signals, such as video, facsimile data, computer data, news despatches, etc.

(4) Digital speech and other data are in a convenient form for processing using digital computers. Thus complex signal processing can be easily applied. Also, the ease of encryption of digital data makes it especially suitable for military communications where secrecy is essential.

(5) The lower power requirement of digital, compared to analogue transmission provides higher reliability and thus better suitability for satellite and computer-controlled communication. Moreover, high redundancy can be introduced into the transmitted codes to improve detection accuracy in noisy channels.

(6) The rapid advance of device technology in terms of digital hardware and VLSI has led to immense economies in the realisation of digital circuits. In digital speech applications, numerous dedicated chips and chip sets have been developed. Also, voice communication with computers is a possibility made available by digital techniques. Speech synthesis has generated considerable interest with the introduction of the Texas Instrument's 'speak-and-spell' synthesiser chip,which carries important implications in the realm of education. Speech recognition is also a rapidly expanding field - effective computer recognition of digitised speech commands could enable users to interact with the computer easily via speech digitisation terminals. This could have far-reaching consequences in terms of the assimilation of computers and robots into the everyday routine of man in the future.

It is not surprising therefore, that a tremendous amount of investment and research has gone into the area of digital speech coding. Indeed, the term 'digital' has itself become something of a fashionable 'catch-word' in the seventies, and will doubtless be even more so, in the eighties.

In the field of digital speech coding and transmission, as in any field, one is concerned with efficiency. Specifically, an efficient speech digitiser should possess good data compression capability (so that transmission bandwidth is reduced without leading to degradation in the quality of the digitised speech) and low implementation cost. Obviously, these two requirements are often diametrically opposed to one another and frequently some sort of compromise has to be sought. An abundance of methods towards achieving this dual requirement has emerged over the relatively brief history of digital speech coding[12,13]. Traditional techniques have sought to preserve the signal waveform. Such 'waveform encoders' can be designed in the time as well as the frequency domain, and provides good quality speech at relatively high transmission bit rates. A different approach seeks to transmit a parametric representation of the speech signal, based on some appropriate model of speech production, in an attempt to obtain very high transmission bandwidth economies. The synthesised speech which derives from such crude representations are often of vastly inferior quality, although intelligibility can be quite high. Another class of coders, the so-called 'hybrid coders' covers the 'middle ground' between these two methods, seeking to combine the advantages of both.

The work to be described in this thesis is concerned with 'waveform encoding' at a transmission bit rate of about 16 Kbps. Various

techniques in both time and frequency domain were investigated. These include differential and predictive coding, noise spectral shaping and adaptive quantization (in the time domain), and sub-band coding and adaptive transform coding (in the frequency domain). In each area, attention is focussed on new methods or modifications to existing methods which can lead to an improvement in performance in terms of quality enhancement, bit rate reduction or coder simplification.

1.3 ORGANISATION OF THESIS

The contents of the thesis will now be outlined.

Following this section, the main results obtained during the course of the research will be highlighted. The experimental procedure, which involves mostly computer simulation is then briefly described in the next section. Details of the input speech data used, the methods of assessment employed, the equipment required and the structure of computer programs are presented.

Chapter two provides a survey of the field of digital speech coding, covering the main areas of current interest. This is intended to be non-technical as much as possible to enable the non-specialised reader to be acquainted with existing speech digitisation techniques. The three broad areas of speech coding are included, namely, analysis-synthesis vocoder systems, waveform coding (in the time and frequency domain) and hybrid coding methods. Various other related issues which have developed alongside the mainstream of speech coding are also considered. The problems of transmission over noisy channels, the effects of delays in the system, the use of variable rate coding are

all important factors to be considered in the design of a communication system or network. Implementation in hardware is also fast becoming an important area of development and follow-up to research, especially with the apparently unceasing advance in device technology. The abundance of systems with varying claims of good performance in the research arena has led to efforts to introduce a more realistic and uniform means of performance assessment. In the context of speech coding, the ultimate measure is the perceptual quality of the output speech. A variety of subjective tests have been designed for this purpose and some of these are discussed. The chapter concludes with an overview of the entire area of speech coding, with projections about future trends and directions of research.

In chapters three through six, the research work conducted is presented. Chapter three considers the subject of adaptive prediction in differential coding systems (ADPCM and APC) with particular emphasis on backward modes of predictor adaptation. Chapter four extends the work on ADPCM further by incorporating the additional feature of noise shaping into the coder. Adaptive quantization, probably the central element in digital coding systems is covered in chapter five. Chapter six is concerned with frequency domain techniques of speech coding and the principles and performance of sub-band coding and adaptive transform coding schemes are examined in detail.

The final chapter, chapter seven provides a recapitulation of the work described and the new ideas proposed. Suggestions are made for further research along the directions already investigated and a final conclusion is made. The appendices, which consist mainly of mathematical derivations precede an exhaustive list of references.

1.4 SUMMARY OF MAIN RESULTS

The main findings during the course of the research are outlined in the following:

Chapter three examines various forms of backward predictor adaptation in the context of ADPCM (adaptive differential pulse code modulation) coding. Several modifications were made to the conventional sequential gradient predictor algorithm in an attempt to improve its efficiency of adaptation during signal transitions. Although some evidence of a quicker transitional response was observed, overall performance did not indicate any significant SNR gain. A backward block adaptive (BBA) predictor was next proposed and evaluated. This was found to provide better prediction efficiency with lower complexity. An attempt was also made to reduce the complexity of the adaptive predictive coder (APC) to an 'implementable' level. Unfortunately, the heavy dependance of the coder on accurate pitch detection presented some difficulties in the simplification process.

In chapter four, the concept of noise spectral shaping was examined in relation to both forward and backward adaptive ADPCM systems employing 2-bit quantization. It was found, in the forward adaptive cases, that a simple fixed pre-/post-filtering method of implementing noise shaping is adequate for coarse quantization applications, providing equivalent quality to that obtained with the more complicated noise-feedback coder. Two backward adaptive noise shaping coders employing the BBA predictor were also proposed and studied. These were found to provide significant improvement over the decoded speech quality of conventional ADPCM. One of them, using a backward pre-filtering method of noise shaping was able

to yield a speech quality at 16 Kbps comparable to that obtained from 7-bit log PCM.

In chapter five, a new method was proposed to reduce quantization noise in ADPCM systems by applying correction to the decoded signals at the receiver. This led to an improvement in SNR as well as subjective quality.

Chapter six examines the performance of two frequency domain coders, namely the sub-band coder (SBC) and the adaptive transform coder (ATC). In an effort to reduce the delay and complexity associated with these two powerful techniques without losing their advantages, a new 'transform-based' approach to split-band coding was proposed. This method provides comparable performance to the SBC but with substantially reduced coder complexity and delay. Further efforts to reduce the delay and complexity of split-band schemes were next investigated. The use of a simple form of vector quantization technique to transmit the adaptation information for split-band schemes results in a sizeable reduction in side information for coders with a large number of bands. Finally, a proposed parallel method of bit allocation has been able to reduce the overall coding delay of the sub-band coder, but this resulted in some degradation in the speech quality.

1.5 BACKGROUND INFORMATION

Some background information is presented in this section. This includes details of the input speech data used,the methods of assessment employed for the systems tested and other relevant information pertaining to the research work.

1.5.1 Input Data

The input speech material consists of three separate data files, which will be referred to throughout the thesis as MALE, FEMALE and SISTER.

(1) MALE - This contains the utterance,

"There was an old man called Michael Finnegan,
He grew whiskers on his chinagen."
spoken by a male speaker.

(2) FEMALE - This contains the same two sentences, spoken by a high-pitched female speaker.

(3) SISTER - This consists of a collection of isolated words spoken by a male speaker and selected for their high fricative or unvoiced content. The words are,

"Sister, father, S. K. Harvey, shift, thick, fist, talk, spent, vote."

All three speech files are band-limited from 0 to 3400 Hz and sampled at 8000 samples per second. MALE and FEMALE are each of approximately five seconds' duration and SISTER is a little more than six seconds (including pauses). These were all obtained from analogue speech using a twelve-bit analogue to digital (A/D) converter.

1.5.2 Computer Facilities

All the results presented have been obtained via simulation on the interactive PRIME 400 mini-computer[14-16] of the Loughborough University's Computer Centre. The programs are all written in the FORTRAN66 language, with graphic facilities provided by the GINO

software routines[17,18].

Emphasis is placed on structured programming - to provide clarity of organisation, ease of debugging and portability. Liberal use is made of subroutines and function segments that form the basic building blocks from which an entire system is constructed. Thus the main routine of a system need only consist of some necessary input/output facilities and a series of subroutine calls.

1.5.3 Assessment of Performance

In order to obtain a reliable assessment of the performance of the various coders simulated, a number of performance criteria (both subjective and objective) are used. These are:-

(1) Total SNR (TSNR) and average segmental SNR (SSNR)

The SNR is possibly the quickest means of determining how well a coder performs in terms of waveform preservation[12,19,20]. Total SNR is given by,

$$\text{TSNR} = 10 \log_{10} \frac{\sum_n x^2(n)}{\sum_n [\hat{x}(n)-x(n)]^2} \quad \text{(dB)} \qquad (1.1)$$

where $x(n)$ and $\hat{x}(n)$ denote the nth input and decoded speech sample, respectively, and the summations are over the duration of the speech file used. However, in the results presented in the thesis, all SNR values quoted were obtained from a summation over about two seconds of speech (60 blocks of 256 samples). This was found to be statistically adequate and indeed, SNR results obtained for the entire utterance of

five seconds are often very similar.

The average segmental SNR is given by,

$$SSNR = 1/K \sum_{j=1}^{K} SNR(j) \qquad (1.2)$$

where SNR(j) is the SNR of the jth block and K = 60. In the computation of the segmental SNR, blocks containing silence are not included in the averaging process so as not to affect the accuracy of the measure. Nevertheless, it is widely recognised that SNR values can be extremely deceptive on occassions as an indicator of output speech quality and thus, this measure must be supplemented by other methods of assessment [12,19].

(2) Spectral plots of the long-term average output noise

This provides a means of observing the distribution of noise energy across the frequency spectrum and can be a useful indicator of the subjective quality of the output speech. The average level of the spectrum indicates the quantity of the noise energy present in the received speech while its shape can provide useful information about the nature of the subjective distortion. For example, a concentration of noise in the low frequency region could indicate 'roughness' or 'rumble' while the same level at the high part of the spectrum will probably be apparent as a background 'hiss'. This measure is particularly relevant for coders which seek to manipulate the shape of the output noise spectrum to exploit masking properties which can be effective in reducing the perception of noise.

The output noise signal is obtained as the difference between the input and the received speech waveforms. In deriving the spectral

characteristics, 60 blocks of this noise signal are used. Frequency analysis is provided by a 256-point FFT (available on the NAG computer software library[21]) on Hamming windowed samples of the noise signal, using an overlap of 50%. The logarithm (to base 10) of the Fourier magnitude components are taken for each block and averaged. The final noise spectrum consists of these averaged components. Because of the limited number of blocks used (60), the averaged noise spectrum produced tends to be characterised by jagged edges. Consequently, a simple moving average smoothing process (over 3 adjacent components) is carried out to round off these sharp edges, without altering the general shape of the spectrum too greatly.

(3) Informal subjective listening tests

The ultimate test of any speech coder is the subjective quality of the output speech produced. Coder assessment is therefore not complete until some listening tests have been carried out. Exhaustive and long drawn out formal listening tests are extremely expensive in terms of both time and effort and are certainly not a necessity at this research stage. Informal listening comparisons are often quite adequate.

To perform the listening test, the digital speech output produced by each coder has to be converted to analogue form and recorded on tapes or cassettes. This process involves the following stages:

(i) The output speech written on disk memory on the computer is transferred onto magnetic tapes, using the MAGNET software package[14] available on the PRIME system.

(ii) This data is next reformatted by the Hewlett Packard HP7970E magnetic tape unit and computer[22] to a form compatible with real-

time output.

(iii) The reformatted data is then ready to be transferred onto recording tapes or cassettes via a twelve-bit D/A converter. An analogue low-pass filter is used during the transfer to remove out-of-band noise in the output speech.

For the purposes of listening tests, data files are processed in their entirety. For each coding system or variation, at least two files are used (usually the MALE and FEMALE data), giving a total of 4 sentences on which assessment may be performed. Where relevant, comparisons are also made with the quality of speech produced by well-known systems such as log PCM. For 16 Kbps coding, 6 and 7 bit log PCM are probably of the most interest. These listening tests are carried out using both headphones and loudspeakers.

Sound spectrograms were also produced on one occassion (chapter 4) with kind assistance from British Telecom Research Labs. However, these are necessarily restricted owing to the difficulties involved in gaining access to the equipment.

CHAPTER TWO DIGITAL CODING OF SPEECH - A REVIEW

2.1 INTRODUCTION

The underlying goal of any speech coding system is to transmit speech, with the highest possible quality, using the least channel capacity and at the lowest cost. Obviously, these are all mutually conflicting aims since, for a given coding scheme, quality is generally proportional to channel capacity and complexity (which is invariably correlated with cost). In most situations, therefore, the need inevitably arises for obtaining a compromise solution, which is optimum for the particular environment and application.

Current speech coding techniques have come a long way since the days of direct quantization of digitized speech using pulse code modulation (PCM)[9]. Most present day algorithms seek to exploit, with varying degrees of complexity, the intrinsic characteristics of speech signals in order to achieve better signal compression and hence higher efficiency. Studies of complex (and potentially efficient) speech coding algorithms have often been deterred by the spectre of high costs, although this situation is gradually changing as a result of recent rapid advances in VLSI (very large scale integration) technology. At the present time, digital speech coders of moderate complexity are already implementable on a single chip, and thus, with further advances in digital technology imminent, research into efficient high-complexity algorithms are considered with rather more than mere academic interest.

The complete design of any transmission system involves the optimal selection of a number of factors, such as signal quality, transmission bit rate, coding delay, complexity and cost. The choice of a particular system would obviously be very much dependant upon the transmission environment (for example, terrestial wire, glass fibre, radio, satellite). Related issues such as the effects and types of transmission errors, multiple (tandem) coding, etc., would also influence coder design.

Digital speech coding techniques may be broadly classified into three main areas, according to the principles employed in their design[12,13]. The first of these is the class of waveform coding methods. These essentially strive for facsimile reproduction of the signal waveform and hence could be used for coding non-speech signals equally well. More efficient speech-specific techniques however, seek to exploit properties of the speech waveform to achieve better signal compression. Waveform coders are generally fairly robust for a wide range of talker environment and are normally of low and moderate complexity.

A second class of speech coders derives from modelling the speech production source. Such source coders, known as vocoders (VOice CODERS) attempt to provide a parsimonious description of speech (using a given model of the speech production mechanism), which could be parameterized and transmitted with minimal channel capacity. Consequently, vocoders are able to achieve high economies in transmission bandwidth. However, the somewhat simplistic model of speech generation employed imposes a severe limit to the quality of the speech produced by these means. Vocoder speech tends to sound 'synthetic' and 'machine-like' - talker recognition is difficult, although high intelligibility is possible.

The 'middle ground' between waveform coders and vocoders is an area receiving increasing recognition as a viable alternative for producing reasonable quality speech at low bit rates. Such hybrid methods offer the attraction of combining some advantages of both waveform coders and source coders.

2.2 TRANSMISSION BIT RATES IN SPEECH CODING

The key issue in transmission systems is perhaps the efficient utilisation of channel capacity. Transmission bit rate is thus a major consideration in the design of speech coders. Figure 2.1 shows a spectrum of speech coding transmission rates currently of interest, and the quality of speech reproduction obtainable at a prescribed bit rate. The quality of reproduced speech is broadly denoted in descending order as, commentary, toll, communications and synthetic.

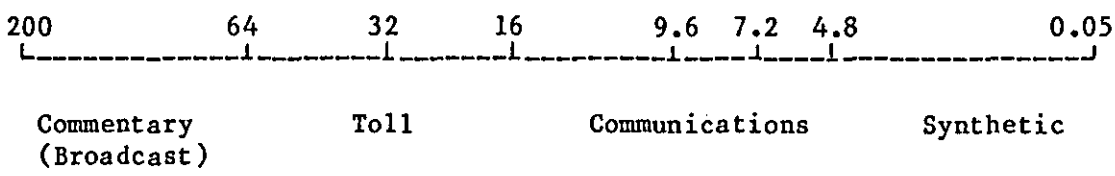


Fig. 2.1 Speech Coding Transmission Bit Rates (Kbps) and Associated Quality

Commentary, or broadcast quality speech is, as its name implies, high quality speech which is suitable for some forms of broadcast material. Its bandwidth is typically from 0 to 7 kHz (wide-band speech) which is much wider than normal narrow-band telephone (300 - 3400 Hz). Toll quality is used loosely to denote a quality of narrow-band speech which is without perceptible distortion. Presently, this is achievable at bit

rates of 16 Kbps and above. The next grade, communications quality, represents a speech quality which possesses noticeable degradation with perhaps lessened talker recognition, although intelligibilty is still high. This is the quality associated with waveform or hybrid coders operating in the range of 9.6 to 7.2 Kbps. Finally, at the very low bit rate range (< 4.8 Kbps), source coders are able to provide intelligible synthetic quality speech with significant loss of 'naturalness' and substantially degraded talker recognition.

The complexity of speech coders tends to be a function of the transmission bit rate. At the upper end of the scale (⩾ 32 Kbps), relatively simple waveform coding techniques are adequate to provide an accurate representation of the signal. As available bits are reduced, more sophisticated implementations become increasingly necessary to retain the same speech quality. Figure 2.2 illustrates the present 'state of art' in the field of speech coding, in terms of speech quality as a function of bit rate[12]. The vertical axis represents a hypothetical quality rating, ranging from a value of 1 (which denotes a quality essentially indistinguishable from the original) to 0 (which denotes extremely poor and unintelligible speech). It is important to realise however, that in reality, subjective quality is a much more complex attribute than is implied by this simple scale.

It can be seen that, as the bit rate decreases from 64 Kbps, first the static, and then the dynamic characteristics of speech signals are exploited to improve coding efficiency. Indeed, as the bit rate is reduced even more, the quasi-periodic nature of speech signals (due to the pitch structure) is also used to effect further signal compression. In addition, advantage may be taken of the perceptual characteristics of

the human ear. It is known that the perception of noise in a given frequency band may be diminished in the presence of high energy speech components in the same band. This phenomenon of 'auditory masking' is the principle behind coders employing 'noise shaping' methods to control the distribution of the noise spectrum in the decoded speech, to provide a more palatable output. The broken line in figure 2.2 represents the 'middle ground' region of speech quality obtainable with hybrid coding techniques, which attempt to bridge the gap in quality between unnatural vocoder speech (which cannot be improved whatever the bit rate after about 2.4 Kbps) and the relatively high quality speech provided by waveform coders(> 16 Kbps).

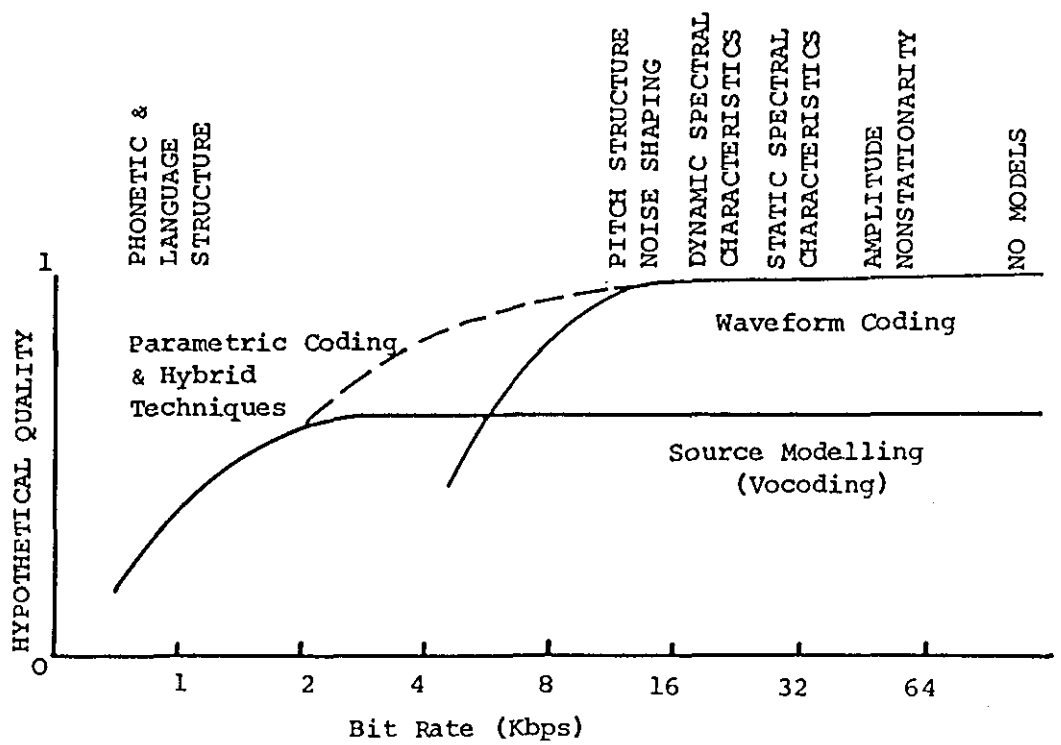


Fig. 2.2 Quality vs Bit Rate for Speech Coding

2.3 VOCODERS

2.3.1 Speech Production Model

The principle of vocoders is the parameterization of speech signals according to a linear quasi-stationary model of speech production, which is based on a crude simplification of the vocal tract. A schematic diagram of the vocal tract is shown in figure 2.3[2].

The vocal tract is a non-uniform acoustical tube, between 15 to 17 cm in length, which extends from the lips to the glottis, and varies its shape as a function of time. This time varying change is caused by movements of the lips, jaws, tongue and velum which are known as the articulators. The lungs, trachea, larynx, throat, nose and mouth all contribute to the production of speech. Speech is produced when air is expelled from the lungs into the trachea and forced between the vocal cords and then through the length of the vocal tract to the oral and nasal outputs. Speech sounds may be broadly classified as either voiced or unvoiced. For voiced sounds, such as /i/ in eve, the expelled air causes the vocal cords to vibrate as a relaxation oscillator (the frequency of vibration determines the pitch), and the air stream is modulated into discrete puffs or pulses. Unvoiced sounds are generated either by passing the air stream through a constriction in the tract, or by making a complete closure, building up pressure behind the closure, and abruptly releasing it. The former gives rise to fricatives such as /f/ in fish, while the latter results in transient stops or plosive sounds, such as /p/ in pickle.

The traditional model of speech production in vocoders is the source system model shown in figure 2.4[12]. Several assumptions are inherent

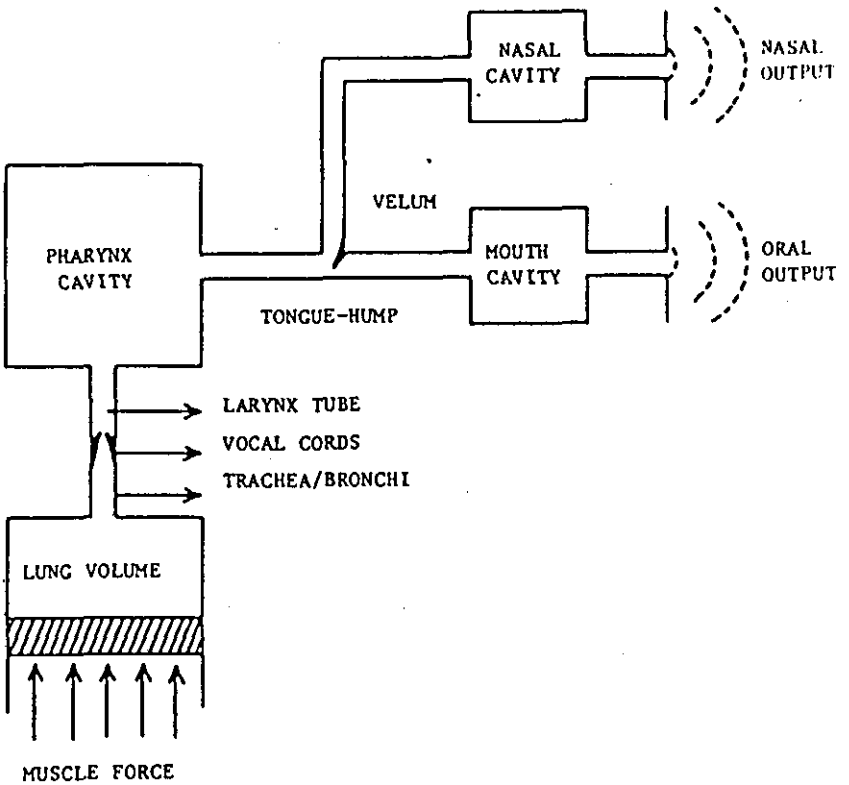


Fig. 2.3 Schematic Diagram of Vocal Tract

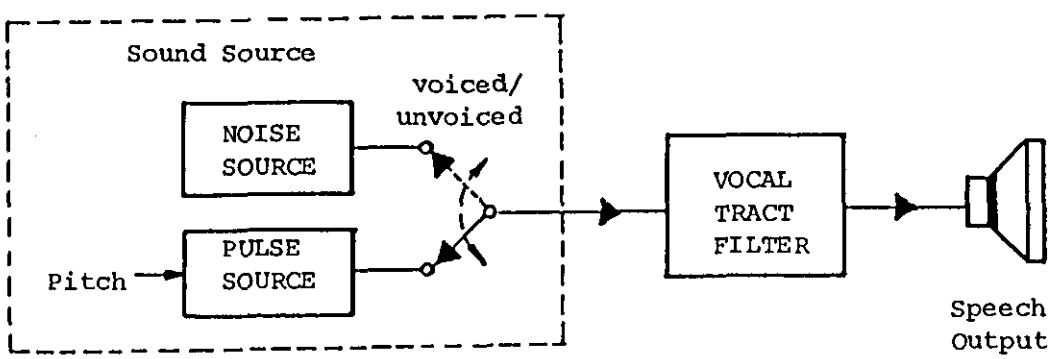


Fig. 2.4 Source-System Speech Production Model

in this model. The sound generating mechanism (the source) is assumed to be linearly separable from the intelligence-modulating vocal tract filter (the system). Also, speech sounds are assumed to be either voiced or unvoiced, and are generated either from quasi-periodic vocal cord pulses or from random sound produced by turbulent air flow.

2.3.2 Principles of Vocoders

The vocoding procedure may be divided into an analysis and a synthesis process[2,12,23,24]. The analysis is performed at the transmitter, where the vocal tract and excitation parameters are extracted from the input speech and transmitted. At the receiver, these parameters are used in the synthesis process to reproduce the original speech sounds.

Synthesis is carried out using a periodic pulse generator to represent voiced sounds, and a random noise generator for unvoiced sounds. The two sources are mutually exclusive, and a parametric signal from the transmitter operates the switch between them. Intensity of sound excitation is also represented parametrically by a gain value, and pitch is specified by a parametric pitch signal. Voiced pitch is very much talker dependent, typically spanning a two-octave range, from 50 to 200 Hz for men, and 100 to 400 Hz for women.

Following the linear source-system model of figure 2.4, the sound output of the vocal tract may be represented as a convolution in time of the excitation waveform e(t) and the impulse response u(t) of the vocal system, thus,

$$x(t) = u(t)*e(t) \qquad (2.1)$$

where * denotes convolution. In the frequency domain, this convolution

is equivalent to a multiplication of the Fourier transforms of u(t) and e(t):-

$$X(\omega) = U(\omega).E(\omega) \qquad (2.2)$$

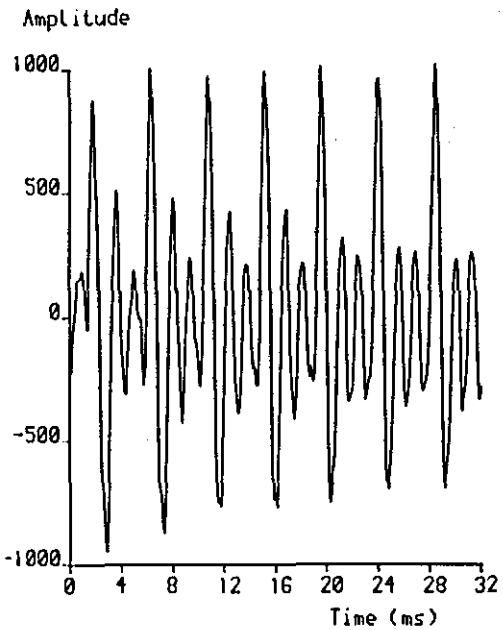
Taking magnitudes gives,

$$|X(\omega)| = |U(\omega)|.|E(\omega)| \qquad (2.3)$$

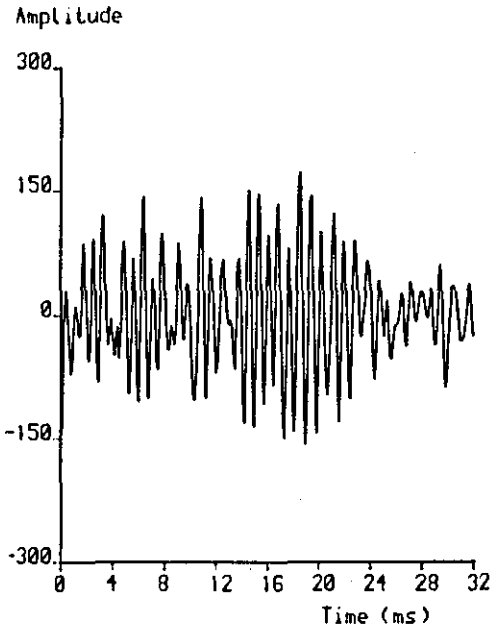
Thus the magnitude spectrum of speech consists of two components: a smooth envelope given by $U(\omega)$ (the frequency response of the vocal tract), and a fine structure corresponding to the excitation term $E(\omega)$. For voiced speech, $E(\omega)$ is a fine line structure and the envelope $U(\omega)$ has several well-defined peaks (typically 3 or 4 for telephone speech), whose centre frequencies are called formants. For unvoiced speech, $E(\omega)$ is noise-like (as e(t) is the result of air turbulence in the vocal tract), and $U(\omega)$ usually have one or two formants above 3 kHz. Typical magnitude spectra of voiced and unvoiced speech segments (for 8 kHz sampled speech) are shown in figure 2.5.

Vocoders depend on a parametric description of the vocal tract transfer function which can take on a variety of forms. These variations in parameter extraction techniques give rise to numerous vocoder designs in both time and frequency domains. In all of the designs however, the dependance upon the signal model of figure 2.4 places a ceiling on the quality of speech that is obtainable. Present research seeks to improve the capabilities of low bit rate vocoders by progressing beyond the simple source-system model.

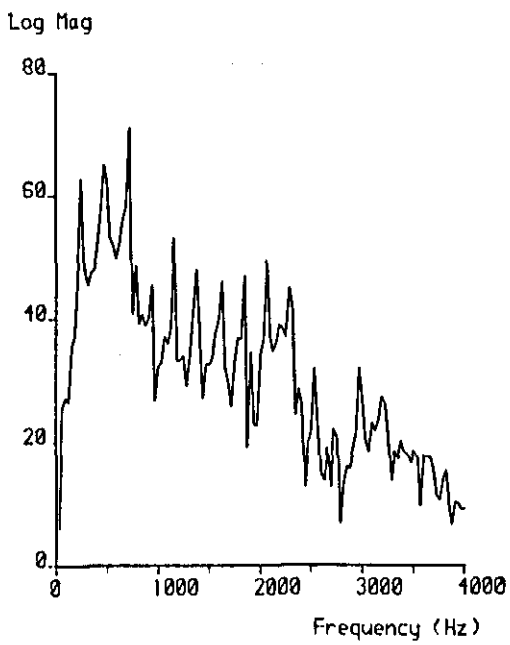
A brief description of the better known vocoder designs will be given in the following.



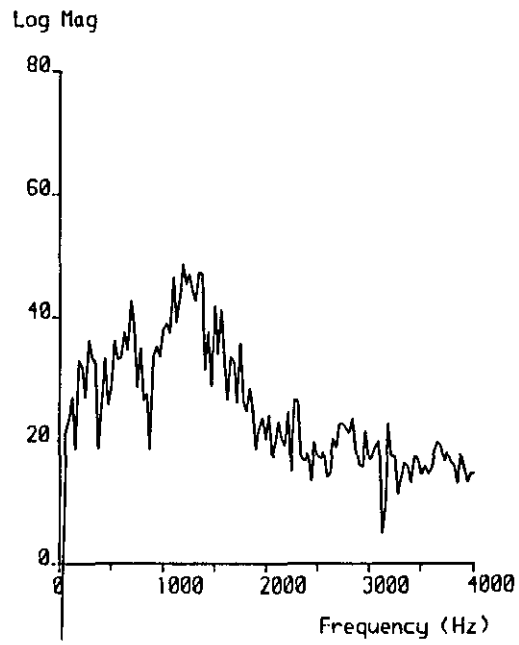
(a) Voiced Speech Waveform



(b) Unvoiced Speech Waveform



(c) Magnitude Spectrum of (a)



(d) Magnitude Spectrum of (b)

Fig. 2.5 Speech Waveforms and Magnitude Spectra for Voiced and Unvoiced Speech

2.3.3 Channel Vocoder

The earliest, and possibly the most well-known vocoder is the channel vocoder[2,12,23-28] invented by Homer Dudley in 1939[25]. The channel vocoder takes into consideration two important features of speech production and perception[26]:-

(1) that the vocal excitation of voiced speech is quasi-harmonic and that of unvoiced speech is a random wide-band signal,

(2) that the perception of speech depends largely upon the preservation of the shape of the short-time amplitude spectrum.

A block diagram of the channel vocoder is given in figure 2.6. A bank of band-pass filters separates the input signal at the transmitter (analyser) into contiguous spectral bands, typically 10-20 bands, each with a bandwidth of 300-150 Hz. The output of each band-pass filter, after rectification and low-pass filtering, represents the time varying signal amplitude of each frequency band. Also included in the analyser are a voiced/unvoiced detector and a pitch detector, which determines the pitch during voiced speech. This information is multiplexed with the spectrum defining channel signals and transmitted.

At the receiver (synthesiser), the speech spectrum is reconstructed from the transmitted data. Excitation, either from a pitch modulated pulse generator (voiced speech) or from a broad-band noise generator (unvoiced) is applied to an identical set of band-pass filters. The output from the filters are amplitude modulated by the spectrum defining signals. The sum of the filter bank outputs yields the reconstructed speech which possesses a short-term spectrum similar to the input. Thus, by utilising and transmitting the short-term spectral content of

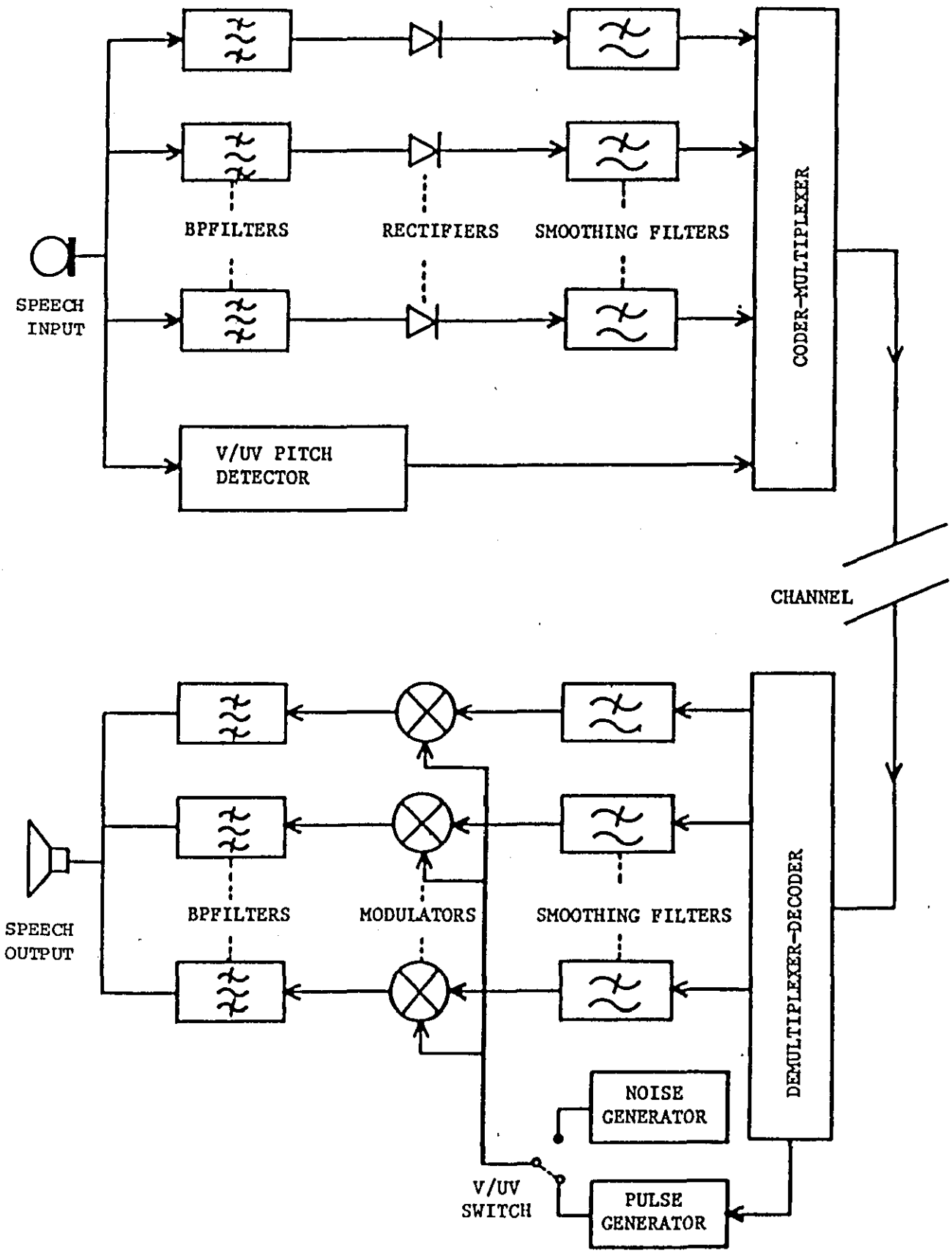


Fig. 2.6 Block Diagram of a Channel Vocoder

speech signals, instead of directly coding the waveform, the channel vocoder is able to effect substantial bandwidth reduction (typically by a factor of 10). Holmes described in detail a 19-channel vocoder developed for the U.K. Government's Joint Speech Research Unit (JSRU) based on the above principles[27].

2.3.4 Formant Vocoder

An even more efficient description of the speech information may be obtained by specifying only the frequencies of peaks (or formants) in the amplitude spectrum[2,26]. This is the principle employed in the formant vocoders, which are able to operate at bit rates as low as 1.2 Kbps. Figure 2.7 shows a block diagram of such a formant vocoder with three formants[26]. The analyser divides the speech spectrum into frequency bands and measures the average frequency f, and the amplitude A of the formants. These parameters, together with the voiced/unvoiced decision and pitch information are then coded and transmitted. At the receiver, the parameters f_1,f_2 and f_3 and the excitation (either f_0 or random noise) are applied to three variable resonators, whose resonant frequencies are determined by the appropriate f value. These signals from the resonators are multiplied by the A signals, and summed to provide the synthesised speech.

2.3.5 Pattern Matching Vocoder

This vocoder achieves even further bit rate reduction and is able to operate at 400 to 800 bps. In this scheme, the short-time speech spectrum is compared with a set of stored spectra, each identifiable by a binary code[2,29]. The code corresponding to the best match is

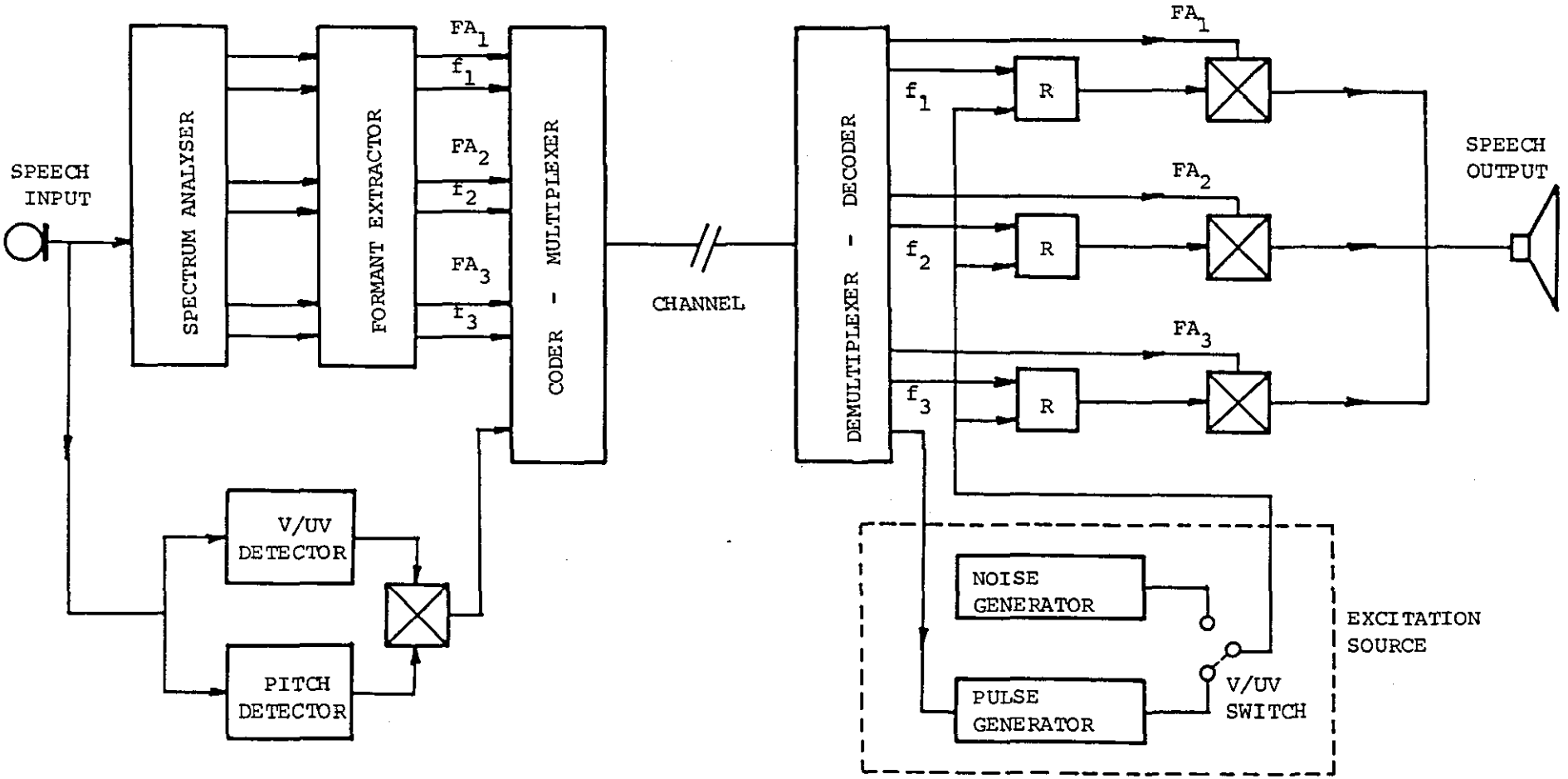


Fig. 2.7 Block Diagram of a Formant Vocoder

transmitted, together with the usual pitch and voiced/unvoiced information. The receiver uses the stored spectrum indicated by the received code to synthesise the speech signal. This principle of pattern matching is similar to the recently proposed vector quantization techniques for speech coding (see section 2.4.1.8).

2.3.6 Homomorphic Vocoder

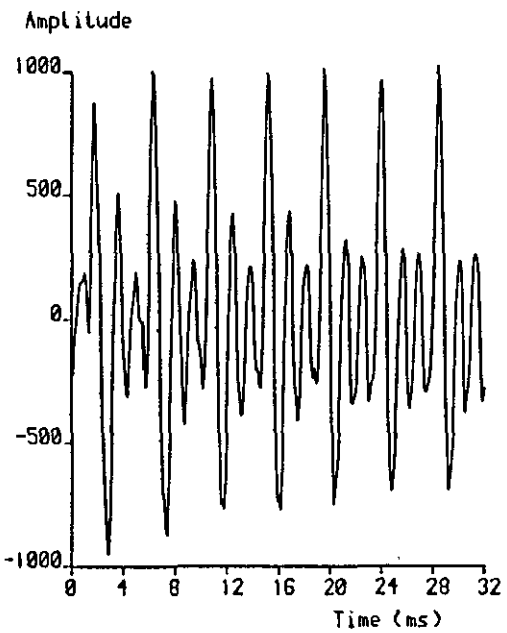
The advent of fast Fourier transform (FFT) techniques in the latter half of the 1960s made feasible the implementation of high resolution spectral analysis of speech. This technical advance, together with research into deconvolution methods, led to the development of the homomorphic vocoder[2,23,24,26,30]. The principle behind this vocoding algorithm is the observation that the mouth output pressure is approximately the linear convolution of the vocal excitation signal and the impulse response of the vocal tract, as given by equations (2.1) to (2.3). Taking logarithm of (2.3) yields,

$$\log|X(\omega)| = \log|U(\omega)| + \log|E(\omega)| \qquad (2.4)$$

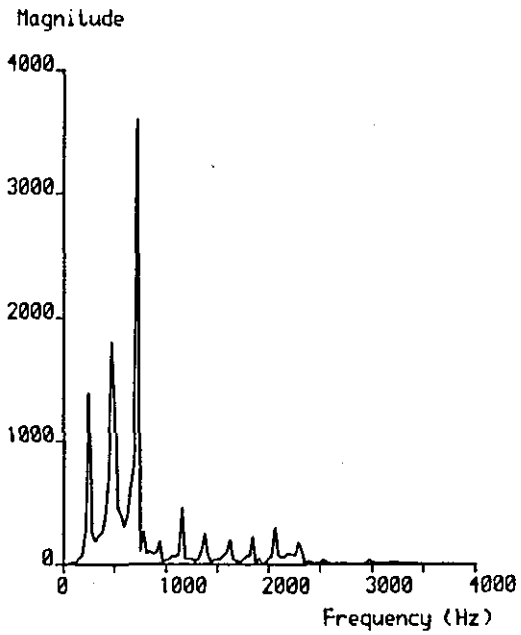
The convolution operation is reduced to an addition of two terms which can now be separated by a filtering process. The inverse Fourier transform of equation (2.4) gives the cepstrum C(t),

$$C(t) = \mathrm{IDFT}(\log|X(\omega)|) = \mathrm{IDFT}(\log|U(\omega)|) + \mathrm{IDFT}(\log|E(\omega)|) \qquad (2.5)$$

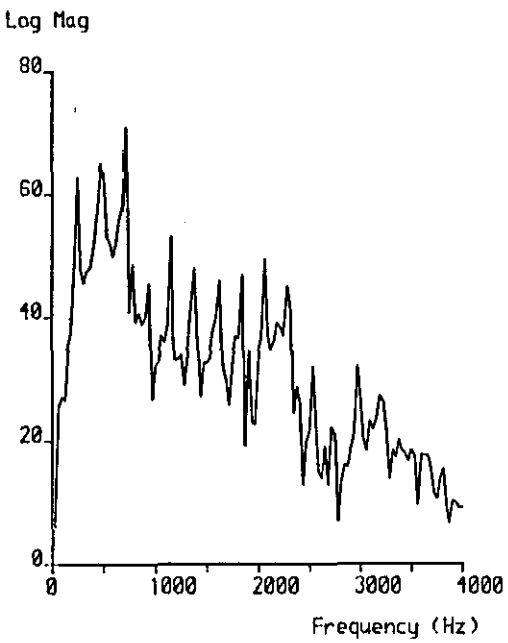
C(t) contains two components - a 'low time' component containing vocal tract information and a 'high time' component due to the excitation. This capability of the cepstrum to isolate the excitation component has led to its widespread use as a pitch detector[31]. Figure 2.8 shows the waveforms corresponding to each stage of signal processing performed during the analysis stage of the homomorphic vocoder. Figure 2.9 is a



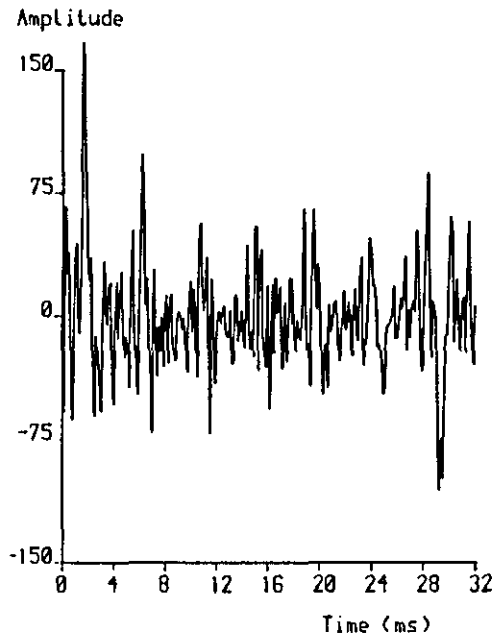
(a) Speech Waveform



(b) Magnitude Spectrum



(c) Log Magnitude SPectrum



(d) Cepstrum

Fig. 2.8 Analysis Stages of Homomorphic Vocoder

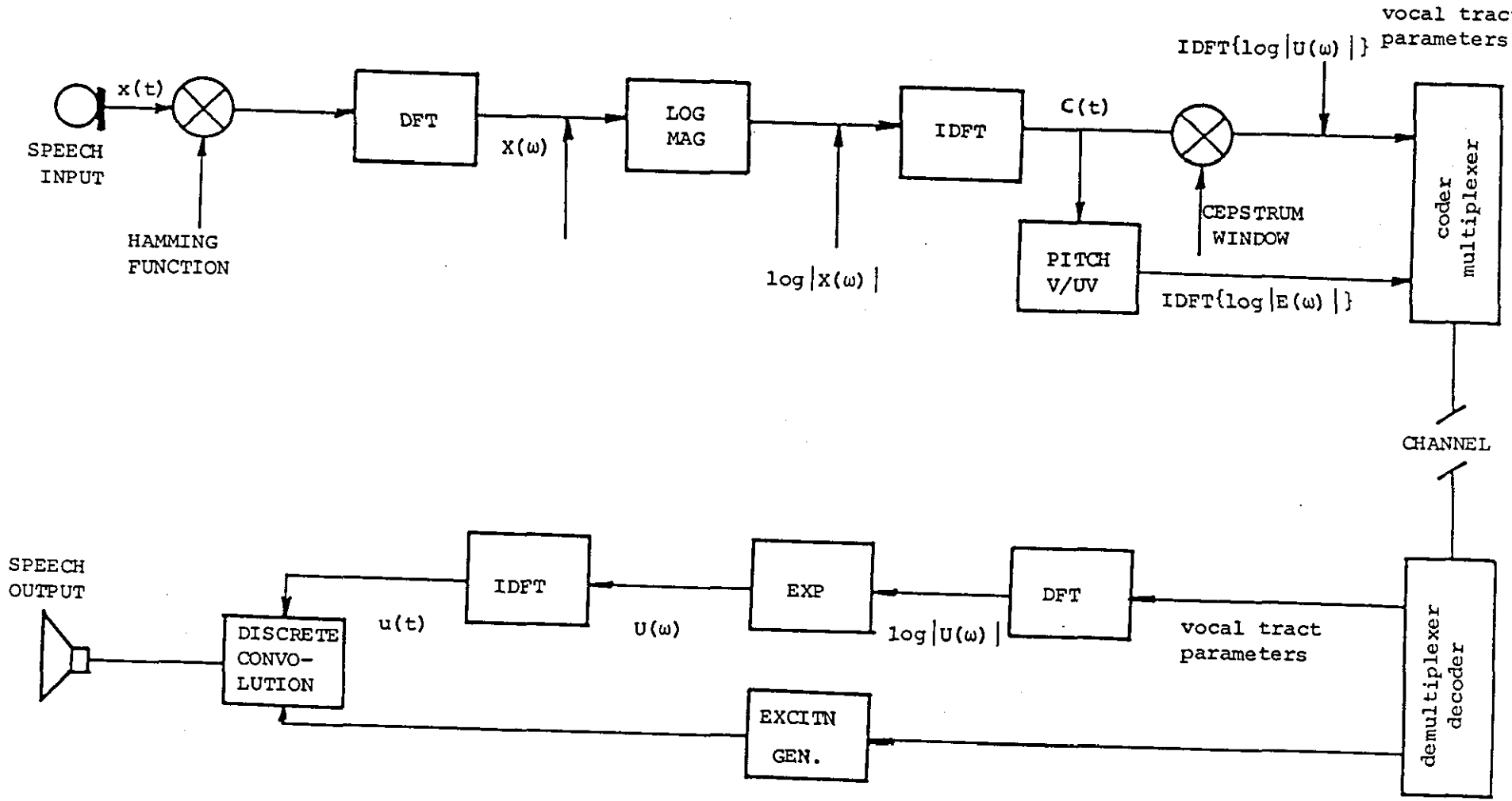


Fig. 2.9 Block Diagram of Homomorphic Vocoder

block diagram of the analysis and synthesis operations involved. The analyser performs the operations specified by equations (2.1) to (2.5) in extracting the cepstrum, which is then suitably truncated to obtain the vocal tract information. The result is the signal c(t), which together with the excitation information constitute the transmission parameters.

Synthesis is accomplished using the signal c(t), which is Fourier transformed, exponentiated, inverse Fourier transformed and finally convolved with the excitation source. The homomorphic or cepstrum vocoder yields good synthetic speech at about 7.8 Kbps, and its implementation has been eased recently with the advent of charged coupled devices (CCD's).

2.3.7 Linear Predictive Coding (LPC) Vocoder

In the linear predictive coding vocoder[2,12,23,24,26,32-36], modelling of the speech waveform is carried out in the time, rather than the frequency domain, thereby avoiding difficulties associated with frequency domain techniques, such as the accurate location of formants. The most commonly used model is the all-pole (or autoregressive) filter given by,

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \qquad (2.6)$$

where G is the amplitude of the input excitation, and the coefficients a_k specify a pth order all-pole approximation of the short-term speech spectrum (p is typically ≥ 8). The complex roots of equation (2.6)

gives the location of the formants and their bandwidths.

For every speech sample x(n) at the input, a linear prediction $\tilde{x}(n)$ is formed from the previous p samples according to :

$$\tilde{x}(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{2.7}$$

The filter coefficients a_k are determined by minimising the square of the prediction error i.e. minimising $(x(n)-\tilde{x}(n))^2$ over an analysis interval that spans typically several pitch periods. The solution of the minimisation process gives[2,12,33,37],

$$A_{opt} = R^{-1}C \tag{2.8}$$

where R is the autocorrelation/covariance matrix, C is the autocorrelation/covariance vector and A_{opt} represents the optimum (i.e. minimum squared error) filter coefficients (see also section 3.2). The block diagram of a LPC vocoder is given in figure 2.10. Analysis consists of extracting the pitch information and the amplitude of excitation G, performing a voiced/unvoiced decision and solving (2.8) for the filter coefficients. Synthesis is accomplished by a recursive filter (formed as the inverse of the linear predictor) fed with the excitation, which are either pitch modulated pulses or random noise. LPC vocoders provide good performance for bit rates in the 2.4 to 4 Kbps range.

The bulk of linear prediction modelling has been on the all-pole model given by equation (2.6). Recent research has suggested the use of a pole-zero or auto-regressive moving-average (ARMA) model which is particularly efficient for modelling unvoiced sounds[32,33,38,39]. The

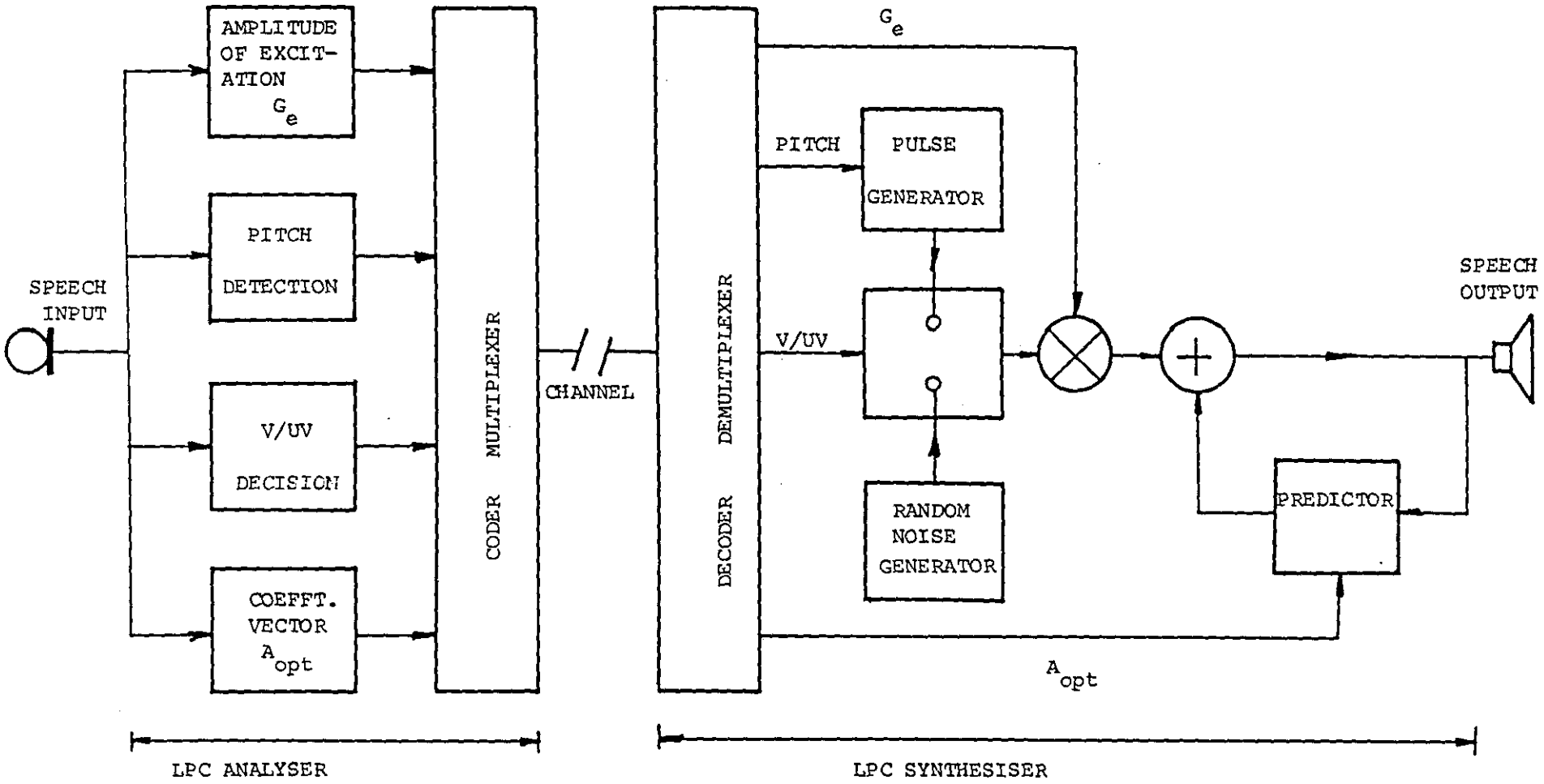


Fig. 2.10 Block Diagram of LPC Vocoder

main difficulty with ARMA modelling is the greater complexity - the optimisation of the filter coefficients leads to the solution of a set of non-linear equations.

2.4 WAVEFORM CODING

Unlike vocoder techniques discussed in the preceding section, waveform coding methods do not consider reproduction of speech in terms of excitation descriptions, vocal tract resonances or articulatory parameters. Instead, an attempt is made to perform a straight-forward reconstruction of the acoustic waveform. Such waveform approximating methods are generally necessary to provide speech of a quality sufficient for commercial telephony. Traditional waveform coding techniques, such as pulse code modulation (PCM), differential pulse code modulation (DPCM) and delta modulation (DM) have been relatively simple. Present day waveform coders, however, are substantially more complex as the search for improved efficiency is spurred by the promise of implementabilty resulting from advances in device technology.

Waveform coder algorithms may be conveniently categorized into time domain and frequency domain classes, but it is important to realise that coders in different classes can be equivalent in terms of the properties of speech that they exploit. For example, adaptive predictive coding (APC - which is a time domain algorithm) and adaptive transform coding (ATC - which is a frequency domain technique) exploit the same redundancy in the speech signal and are therefore considered 'equivalent' in this sense.

2.4.1 Time Domain Methods

2.4.1.1 Pulse Code Modulation (PCM)

Perhaps the simplest form of waveform coding is that of linear pulse code modulation (PCM)[9,12,37], in which an analogue signal is uniformly quantized in a rectangular grid in time and amplitude. This is an approach widely used in methods of analogue-to-digital conversions. Since it does not seek to exploit any properties of speech, it is not constrained to this class of signals and does not possess any inherent data compression capability.

Historically, PCM is the first method used for digital transmission of speech. It was proposed by Reeves in 1938[40] and analysed in detail by Cattermole[9]. The operation of PCM may be summarised into the following steps:

(1) The band-limited analogue signal is first sampled at or above the Nyquist frequency i.e. a frequency twice the signal's bandwidth.

(2) The amplitude of each signal sample is quantized into 2^B levels, where B is the number of bits allocated for the encoding of each sample.

(3) The discrete amplitude levels are represented by distinct binary words of length B, which are transmitted.

(4) The decoder converts the binary words back into amplitude levels and the resulting amplitude-time pulse sequence is low-pass filtered to yield the recovered analogue signal.

It is clear that the only source of noise in PCM is due to the quantization error, which is proportional to the quantizer step-size, assuming that the amplitude range of the input signal does not exceed

that of the quantizer (i.e. no 'overload' occurs). Thus high fidelity reproduction of speech can be achieved by employing a large number of closely spaced quantization levels, but this would involve excessive and unacceptable bit rate requirements. Linear PCM is clearly a highly inefficient means of quantizing speech signals as it does not take into account the characteristics of the input. More effective methods utilise either non-uniform quantization or adaptive quantization.

(a) Non-uniform Quantization

Non-uniform quantization[37,41-45] is characterised by fine quantizer steps for the very frequently occurring low amplitudes of speech signals and much coarser steps to take care of the occassional large amplitude excursions.Such characteristics are termed 'companding' characteristics, from the fact that the step-sizes are COMpressed for the low amplitudes, and exPAND rapidly outwards to cover the range of the signal to be quantized (see figure 2.11). Two non-uniform quantizers widely used in commercial telephony applications (denoted as A law and μ law PCM) utilise a logarithmic charateristic for the quantizer steps. These are defined as follows[9,12,37,42] (for $x(n) > 0$);

μ law: $$x_c(n) = \frac{V \ln (1 + \mu x(n)/V)}{\ln (1 + \mu)} \quad ; 0 < x(n) \leq V \qquad (2.9)$$

A law: $$x_c(n) = \frac{Ax(n)}{1 + \ln A} \quad ; 0 \leq x(n) \leq V/A$$

$$x_c(n) = \frac{V[1 + \ln (Ax(n)/V)]}{1 + \ln A} \quad ; V/A \leq x(n) \leq V \qquad (2.10)$$

where $x(n)$ is the input and $x_c(n)$, the compressed quantizer output. μ

and A are parameters controlling the shape of the logarithmic characteristic, and V is the maximum amplitude of the input signal. The use of a logarithmic characteristic allows the quantizer to span the large dynamic range encountered in typical speech communication.

Another approach to non-uniform quantization seeks to tailor the quantizer characteristic to the probabilty density function of the input signal. Max[43] proposed an iterative method for obtaining the optimum (i.e. minimum mean squared error distortion) quantizer input/output threshold levels for signals with a Gaussian density. Paez and Glisson [45] extended this work to signals with Laplacian and gamma distributions, both of which are fairly good models of long-term speech amplitudes. These pdfs(with a standard deviation = σ) are defined as follows:-

$$\text{Gaussian pdf} \quad : \quad p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-x}{\sigma\sqrt{2}}\right]^2 \tag{2.11}$$

$$\text{Laplacian pdf} \quad : \quad p(x) = \frac{1}{2\beta} \exp\{\frac{-|x|}{\beta}\} \qquad \text{with } \sigma = \sqrt{2\beta} \tag{2.12}$$

$$\text{gamma pdf} \quad : \quad p(x) = \frac{\sqrt{k}}{2\sqrt{\pi|x|}} \exp(-k|x|) \qquad \text{with } \sigma = \frac{\sqrt{0.75}}{k} \tag{2.13}$$

and their characteristics are shown in figure 2.12.

(b) Adaptive Quantization

The dynamic range of speech signals in typical voice communication systems can vary by as much as 40 dB. While logarithmic quantization is able to capture this wide variation to some extent, better results can

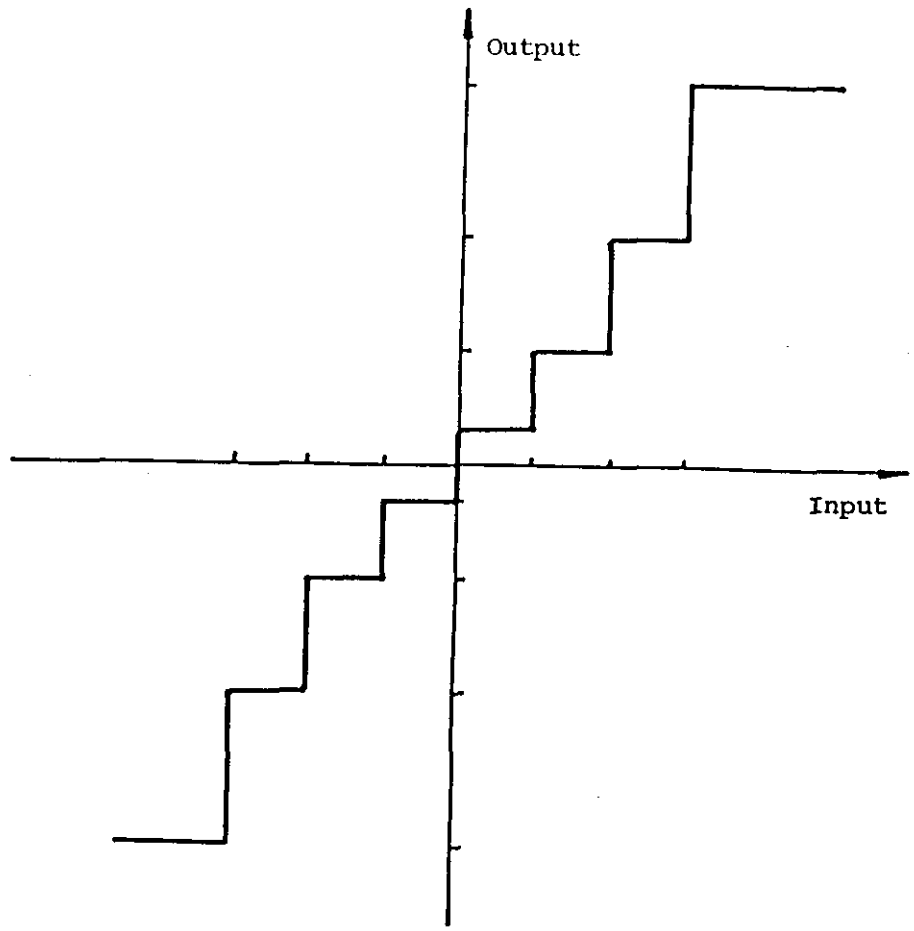


Fig. 2.11 Illustration of Non-uniform Quantizer Characteristics

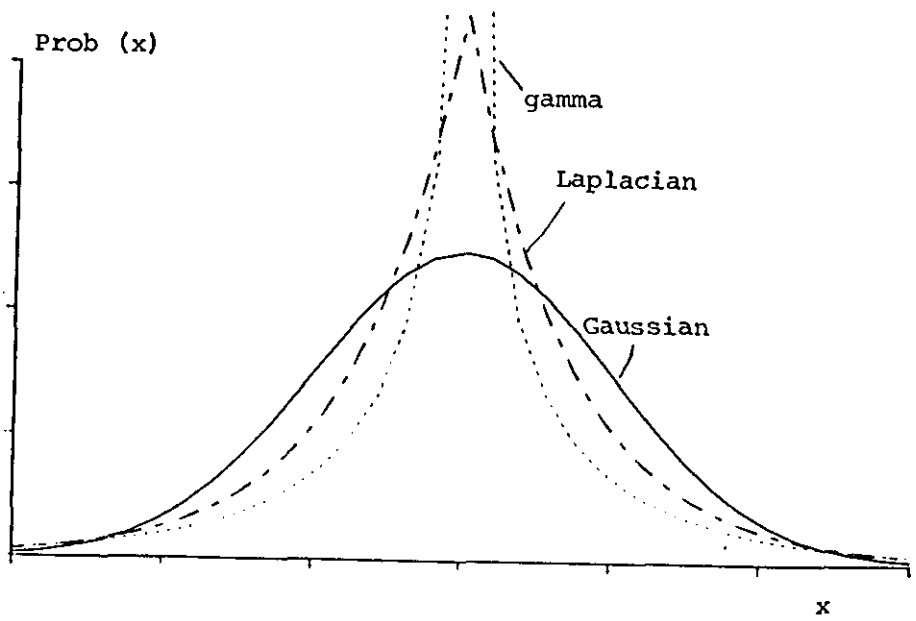


Fig. 2.12 Models of Speech Amplitude Distribution

be obtained by employing a quantizer which is able to adapt its range according to the non-stationary nature of speech signals. Adaptive quantization utilises a quantizer characteristic (uniform or non-uniform) that shrinks or expands in time like an accordion, to adapt to low and high speech powers respectively[12,20,37]. Although speech signals have a large dynamic range over a long period of time, input power levels vary slowly enough to facilitate the design of simple adaptation algorithms to track these power variations. These adaptations may proceed either on a 'block' basis, as in forward block quantization (AQF) or on a sample by sample basis, as in the well-known one-word memory quantizer (AQJ) algorithm developed by Jayant.

(i) Forward Adaptation

In forward block adaptive quantization[19,20,46-48], the quantizer step-size Δ is calculated for a block of N input samples (typically 4-16 ms duration) and transmitted to the receiver. This step-size is normally obtained from the root-mean-square (rms) value of the block of signal samples as,

$$\Delta = \alpha \frac{1}{N} \sqrt{\sum_{j=1}^{N} x^2(n-j)} \qquad (2.14)$$

where α is an appropriate constant weighting factor which depends on the number of bits used in the quantizer. This optimum step-size is then used to quantize the same block of the signal. Naturally, the use of such 'look ahead' features ensures that the quantizer step-size is always matched to the power of the signal, and thus provide substantially improved performance over time-invariant quantizers. The price to be paid for this advantage is the introduction of a time delay

into the system (equal to the duration of update of the quantizer step-size) and the need for additional 'side information' to be transmitted to the receiver. Optimum quantizers may be employed with such forward block adaptations to obtain minimum distortion. In this case, the standard deviation of the block of samples is used to normalise the signal, before quantization by a unit variance optimum quantizer. For relatively short duration blocks (4-8 ms), the speech amplitude distribution is approximately Gaussian. As the blocksize is increased however, it tends toward Laplacian, and for the long-term, it becomes very much gamma distributed.

(ii) Backward Adaptation

Perhaps the best known adaptive quantizer[37,47,49] in recent years is the one-word memory sequential adaptation algorithm developed by Jayant [49]. This provides a means of matching the quantizer step-size to the signal variance using quantizer memory. The principle is to modify the step-size of the quantizer for every new input sample, by a factor depending on the knowledge of which quantizer slot was occupied by the previous sample. The step-size adaptation evolves according to,

$$\Delta(n+1) = \Delta(n).M(|H(n)|) \tag{2.15}$$

where $\Delta(n)$ is the step-size at the nth instant, and $M(.)$ is a time-invariant multiplier function that depends on the magnitude of the transmitted codeword at time n, denoted by $|H(n)|$. The characteristics for a 3 bit Jayant quantizer is shown in figure 2.13.

A quantization technique similar to Jayant's algorithm is the variance estimating quantizer studied by Stroh[41], Noll[20] and Castelino[50], where the input signal is normalised by the square root of a maximum

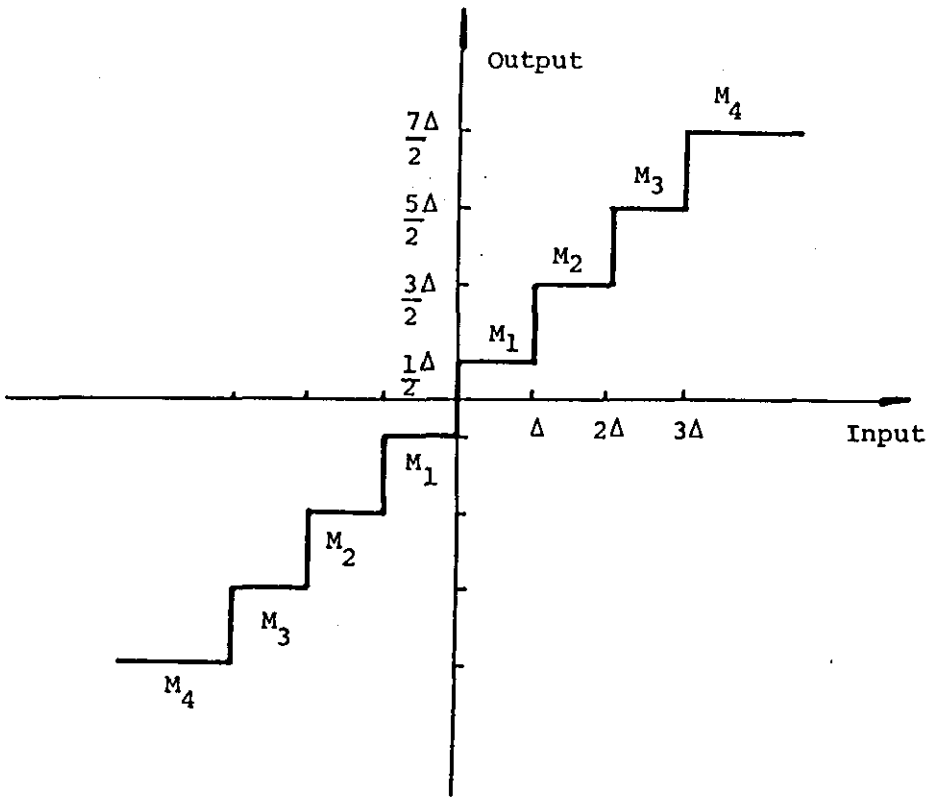


Fig. 2.13 3 Bit Jayant Quantizer Characteristics

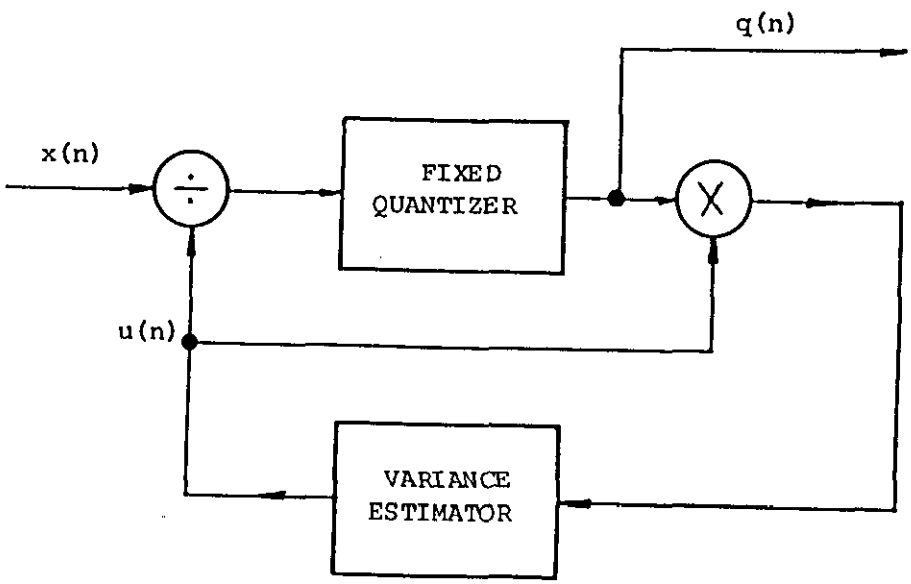


Figure 2.14 Variance Estimating Quantizer

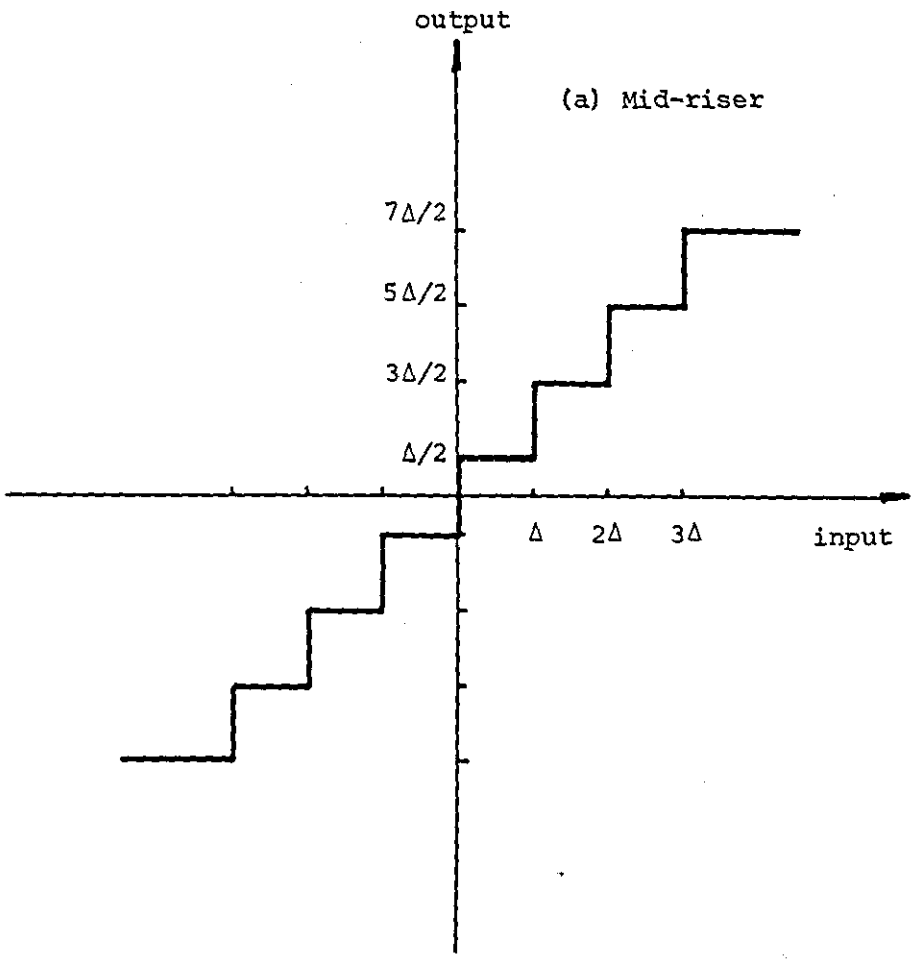
likelihood estimate of its variance at every sampling instant, and the result is quantized by a fixed quantizer (see figure 2.14). The normalising value is made proportional to a moving estimate of the decoded signal's standard deviation in order to obtain a unit variance signal which can then be optimally quantized.

Another companding technique along the same lines is the proposal of Wilkinson[51]. In his scheme, the step-size Δ is adapted with a time constant of about 5-10 ms rather than for every sample. Xydeas[11,52, 53] proposed a dynamic ratio quantizer (DRQ) which utilises an instantaneously adaptive non-linear element to normalise the input signal prior to quantization.

Most of the adaptive quantization techniques proposed provide an SNR advantage over logarithmic PCM of between 3 and 5 dB. Adaptive quantization will be considered in greater detail in chapter 5.

(c) Mid-rise and Mid-tread Quantizer Characteristics

Since speech signals are symmetrical about the time axis, quantizers are likewise symmetrical. Depending on the input/output quantizer staircase characteristics, two versions of the quantizer may be identified - namely the mid-rise and the mid-tread, shown in figure 2.15. The mid-rise quantizer has its decision level at the origin, while the mid-tread has a zero output level. Mid-riser characteristics are preferred, mainly due to the fact that it uses an even number of levels, which makes it compatible with binary representation. The mid-tread quantizer however, has superior idle channel performance due to the existence of a zero level output. The use of a switch that exploits both mid-rise and mid-tread characteristics has been suggested by



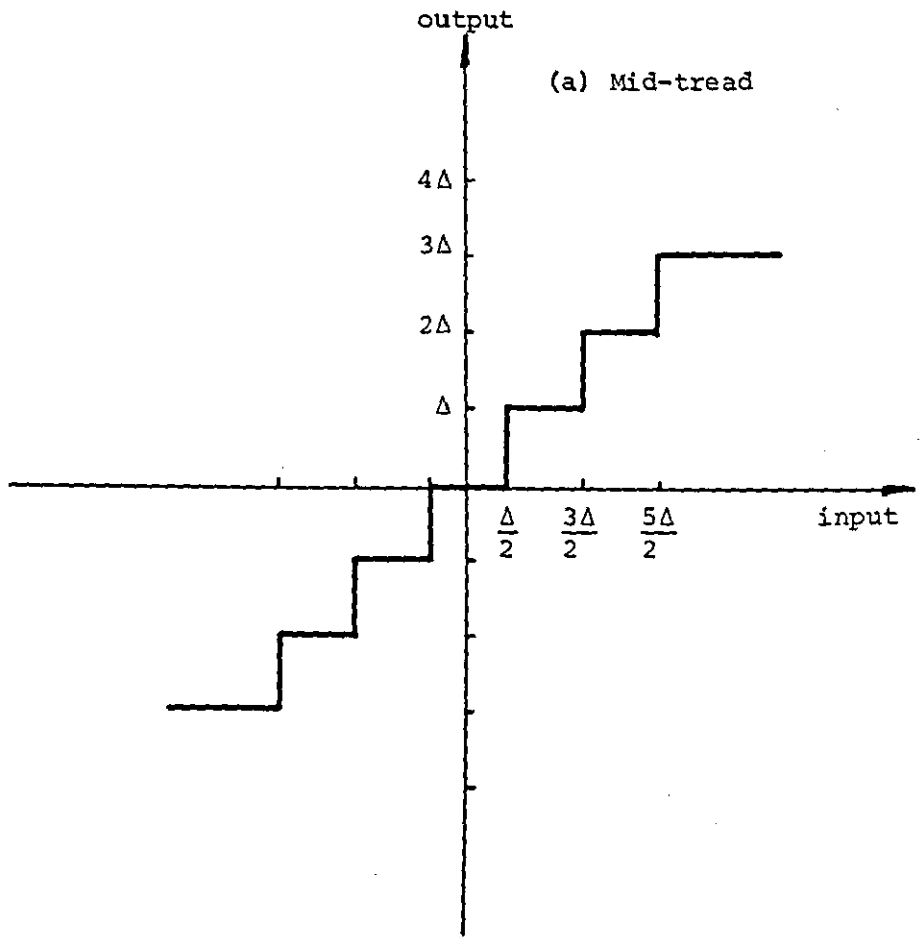


Fig. 2.15 Mid-rise and Mid-tread Quantizer Characteristics

Crochiere [54].

2.4.1.2 Differential Pulse Code Modulation (DPCM)

Adjacent amplitudes in speech waveforms sampled at the Nyquist frequency are often highly correlated. One consequence of this correlation is that the variance of the difference e(n) between speech samples x(n) and x(n-1) is much smaller than the variance of x(n). Since the quantization error power is proportional to the quantizer input power for a given fineness of quantization, it is advantageous to quantize and transmit the difference between adjacent samples of speech instead of the speech sample itself. The reconstruction of the original speech sample can be performed by a simple process of integration.

This is the basic principle of differential pulse code modulation (DPCM) [11,12,24,37,45,55-64], which is based on an invention by Cutler[61]. If the variance of the quantizer input is reduced by a factor G, the variance of the quantization error is also reduced by G and thus the signal to noise ratio (SNR) will be similarly increased by G. If the correlation between adjacent samples of the speech signal is c_1 (by definition $-1 < c_1 < 1$), it can be shown that the value of G for the first order differential coding scheme (i.e. one in which the difference between adjacent samples is transmitted) is $\{2(1-c_1)\}^{-1}$. More generally, if the difference between x(n) and a weighted version of x(n-1), say $a_1x(n-1)$ is used as the quantizer input, the variance of this signal is minimum when $a_1=c_1$. In this case, G is given by $(1-c_1^2)^{-1}$, a gain which is greater than unity for all values of c_1. The quantity $a_1x(n-1)$ can be considered as a first order prediction of x(n) and the corresponding differential coding scheme is a predictive

coder. MacDonald[62] found that apart from superior SNR performance, the choice of $a_1=c_1$ for a first order predictor provides better tolerance to channel errors.

Figure 2.16 shows a block diagram of a generalised DPCM coder and decoder, where P represents a pth order fixed linear predictor

$$P(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{2.16}$$

and $\hat{e}(n)$ denotes the quantized value of e(n). From the figure, it can be seen that the locally decoded speech sample at the nth instant is,

$$\hat{x}(n) = \hat{e}(n) + y(n) \tag{2.17}$$

where y(n) denotes the prediction of x(n). Also,

$$\hat{e}(n) = e(n) + q(n) \tag{2.18}$$

where q(n) is the quantization error. As,

$$e(n) = x(n) - y(n) \tag{2.19}$$

it follows from (2.17) to (2.19) that,

$$\hat{x}(n) = x(n) + q(n) \tag{2.20}$$

Therefore, the decoded sample $\hat{x}(n)$, is the sum of the input sample x(n) plus the quantization error q(n) arising from the quantization of the difference sample e(n). Note that this condition occurs because of the feedback round the quantizer. y(n) is thus a prediction obtained from the previous p decoded samples, and not the input samples.

The formal design of the DPCM predictor is given in chapter 3.

2.4.1.3 Adaptive Differential Pulse Code Modulation (ADPCM)

The term DPCM is normally used to denote the differential coder configuration of figure 2.16 which employs a fixed (i.e time-invariant)

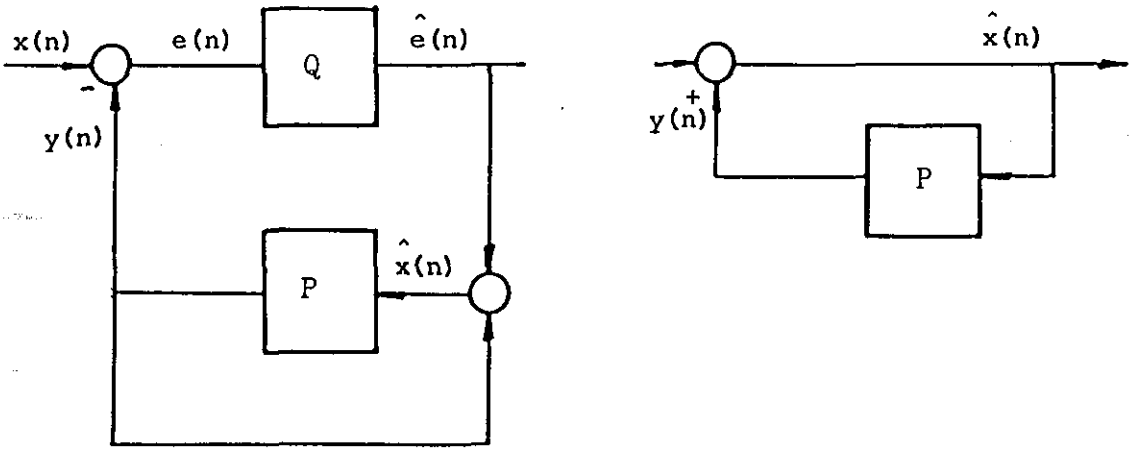


Fig. 2.16 Block Diagram of DPCM Coder

quantizer and a fixed predictor, whose coefficients are optimised for long-term speech characteristics. It is apparent however, that predictors or quantizers whose design are based on long-term statistics cannot be optimum at all times because of the non-stationary nature of speech signals, and the quite significant talker variability encountered in practical voice communication systems. Consequently, practical versions of DPCM are likely to employ adaptive quantizers and/or adaptive predictors - the former to follow changes in signal power and the latter to respond to variations in the short-term speech spectrum. Coders incorporating such adaptive features are known as adaptive differential pulse code modulation (ADPCM)[37] coders. There has been a vast amount of research on ADPCM speech encoding systems over the years and most of these are concerned with various methods of adapting the quantizer and the predictor.

(a) Adaptive Quantization

The same principles of adaptive quantization[12,20,37,64] as mentioned in section 2.4.1.1(b) with reference to PCM coding are applicable to DPCM. The only difference is that, instead of the input speech signal, it is now the difference sequence which has to be quantized. Quantizer adaptation may again be either in a forward mode or a backward mode. If a forward block method is used, an estimation of the quantizer step-size will have to be made using the input signal, since the feedback DPCM configuration of figure 2.16 (with the quantizer inside the loop) does not permit the accumulation of error samples for estimation purposes.

Backward adaptive quantization techniques in DPCM are basically similar to those of PCM, and the various adaptations discussed in section 2.4.1.1(b)(ii) are also directly applicable. The one-word memory

quantizer of Jayant is widely used in DPCM systems. Adaptation is similar to the PCM case, but the multiplier values are different (see section 5.2.2.1)[64].

Stroh's method of backward variance estimation[41] can easily be applied to DPCM. The normalising value is still a moving estimate of the quantizer input, which in this case, is the sequence of quantized difference samples.

One inadequacy in most adaptive quantization algorithms is the inability to adapt sufficiently quickly to the large amplitude excitation pulses, which characterise the prediction residual signal. The consequent 'clipping' of the residual could lead to significant losses in SNR as well as perceptible distortion in the form of 'clicks' in the decoded speech. To overcome this problem, Cohn and Melsa proposed a pitch compensating quantizer (PCQ)[66] which uses two modes of operation: an envelope detector for the syllabic adaptation, and a Jayant (AQJ) loop for pitch compensation. A five level quantizer is used, with the two outermost levels placed further apart than usual, to capture the high amplitude excitation pulses. Qureshi and Forney[67] suggested a rather similar scheme which uses two Jayant loops with different adaptation characteristics - one for syllabic companding and the other for pitch compensation.

Further discussion of adaptive quantization in DPCM systems will be deferred until chapter 5.

(b) Adaptive Prediction

While adaptive quantizers seek to follow the power level of the input signal, adaptive predictors[12,19,65] offer the possibility of tracking

the short-term input signal's spectral characteristics, in order to achieve greater variance reduction. Most forms of adaptive prediction ADPCM provide 2 to 3 dB advantage in SNR, compared to fixed prediction, under otherwise identical conditions. As in adaptive quantization, predictor adaptation may proceed either on a forward block mode or a backward sequential basis.

In forward block adaptive prediction[19,55], the optimum predictor coefficients are calculated to minimise the forward prediction error over a block of input samples, normally between 8 to 32 ms duration. Since adaptation proceeds on a block basis, a data buffer is required at the transmitter to collect and store incoming input samples until the minimisation can be performed. This introduces a delay to the system which is equal to the time duration of the block. At the same time, because the predictor coefficients are obtained from the input signal, they are not available at the receiver and have to be transmitted as side information[68].

The need for side information and delay may be avoided if predictor adaptation is performed in a backward mode[19,55,68-75]. Such backward adaptations usually proceed on a sequential or sample by sample basis. The predictor coefficients are continually updated to minimise some error criterion according to the general formula,

$$a_k(n+1) = a_k(n) + \varepsilon \quad ; k = 1,2,....p \tag{2.21}$$

where ε is derived from information available at both transmitter and receiver. This usually includes previously decoded error and signal samples. Most backward adaptive schemes employ some form of steepest descent or gradient algorithm using minimum mean square error criteria.

Gibson investigated the performance of the stochastic approximation predictor[69] and the Kalman predictor[71] in speech coding applications and found the latter to be slightly superior. Cummiskey[76] proposed a similar backward adaptation technique based on the minimisation of the absolute, instead of the squared prediction error. In general, sequentially adaptive predictors tend to be rather sensitive to transmission errors, which can easily lead to filter instability - an obviously unacceptable condition in practical applications. This drawback may be avoided to some extent if, instead of a transversal predictor structure, a lattice configuration is employed. Indeed, much interest has been focussed on the use of adaptive lattice predictors in ADPCM in recent years[77-79]. The details of this and various other adaptive schemes are covered in chapter 3.

The advantage of DPCM over direct PCM may be eroded if the signal to be transmitted possesses statistically different characteristics from speech. For example, some telecommunication networks might be required to carry data, as well as speech signals. In such cases, the need might well arise for designing a predictor which is able to perform well for both speech and data inputs. Predictors which are designed for more than one type of signal are termed 'compromise predictors' since such predictors will inevitably be a sub-optimum compromise for the different signals individually. O'Neal and Stroh[59] studied several cases of compromise prediction used in DPCM, and showed that these provide superior performance over PCM. Not unexpectedly, however, the SNR obtained with such compromise predictors is always less than the case where the DPCM coder is optimised and used for each type of signal individually.

2.4.1.4 Pitch Predictive Coder

While ADPCM systems are concerned only with exploiting the short-term spectral envelope redundancy of speech signals, a more sophisticated class of speech coders attempts to effect even further signal compression by taking advantage of the longer-term pitch redundancy present in voiced speech. Perhaps the most well-known research effort in this direction is the adaptive predictive coding (APC) system developed by Atal and Shroeder[12,19,37,80-82]. The APC coder (shown in figure 2.17) can be considered as an 'enhanced' version of ADPCM and incorporates two adaptive predictors; a short-term vocal tract predictor (similar to ADPCM) given by,

$$P_1(z) = \sum_{k=1}^{p} a_k z^{-k} \qquad (2.22)$$

and a long-term pitch predictor, given by,

$$P_2(z) = \beta z^{-M} \qquad (2.23)$$

where β is a gain parameter, M represents the pitch period in number of samples and p is typically ≥ 8. All adaptation proceed on a forward block basis. The optimisation of the coder parameters is performed on the input speech and transmitted to the receiver periodically. It was found that if the pitch predictor is modified to span two pitch periods, i.e.

$$P_2(z) = \beta_1 z^{-M} + \beta_2 z^{-2M} \qquad (2.24)$$

better prediction is achieved. Using this APC system, Atal and Shroeder reported a synthesised speech quality at a transmission bit rate of about 10 Kbps, better than 6 bit log PCM.

In later versions of the APC[81], the pitch predictor was again modified to:

$$P_2(z) = \beta_1 z^{-M+1} + \beta_2 z^{-M} + \beta_3 z^{-M-1} \qquad (2.25)$$

The three amplitude coefficients provide a frequency dependent gain factor which improves the prediction at higher frequencies, giving an average 3 dB prediction gain over the first order case. At the same time, Atal and Shroeder also introduced the concept of noise shaping (see section 2.4.1.6(a)) to their APC system, to yield good subjective quality speech at a bit rate below 16 Kbps.

Goldberg and Schafer[83] described a real-time mini-computer implementation of a simplified APC system operating at 6400 Kbps using a 4th order short-term predictor and a pitch predictor based on the computationally efficient average magnitude difference function (AMDF) [84,85] given by,

$$AMDF(j) = \sum_{n=1}^{T} |x(n) - x(n-j)| \qquad (2.26)$$

The AMDF(j) is calculated for all j of interest (i.e. within the block of T samples) and the value of j which minimises the AMDF is the estimated pitch period. The quality of the synthesised speech was described as 'reverberant' and contains perceptible granular noise.

Jayant investigated the performance of two pitch predictors in his pitch adaptive DPCM coder[86] intended for operation at 16 Kbp - one uses the AMDF and the other is based on the autocorrelation function. After experimenting with various combinations of long and short-term predictors, he reported that the best results were obtained with a prediction scheme using a fixed 3-tap short-term predictor for

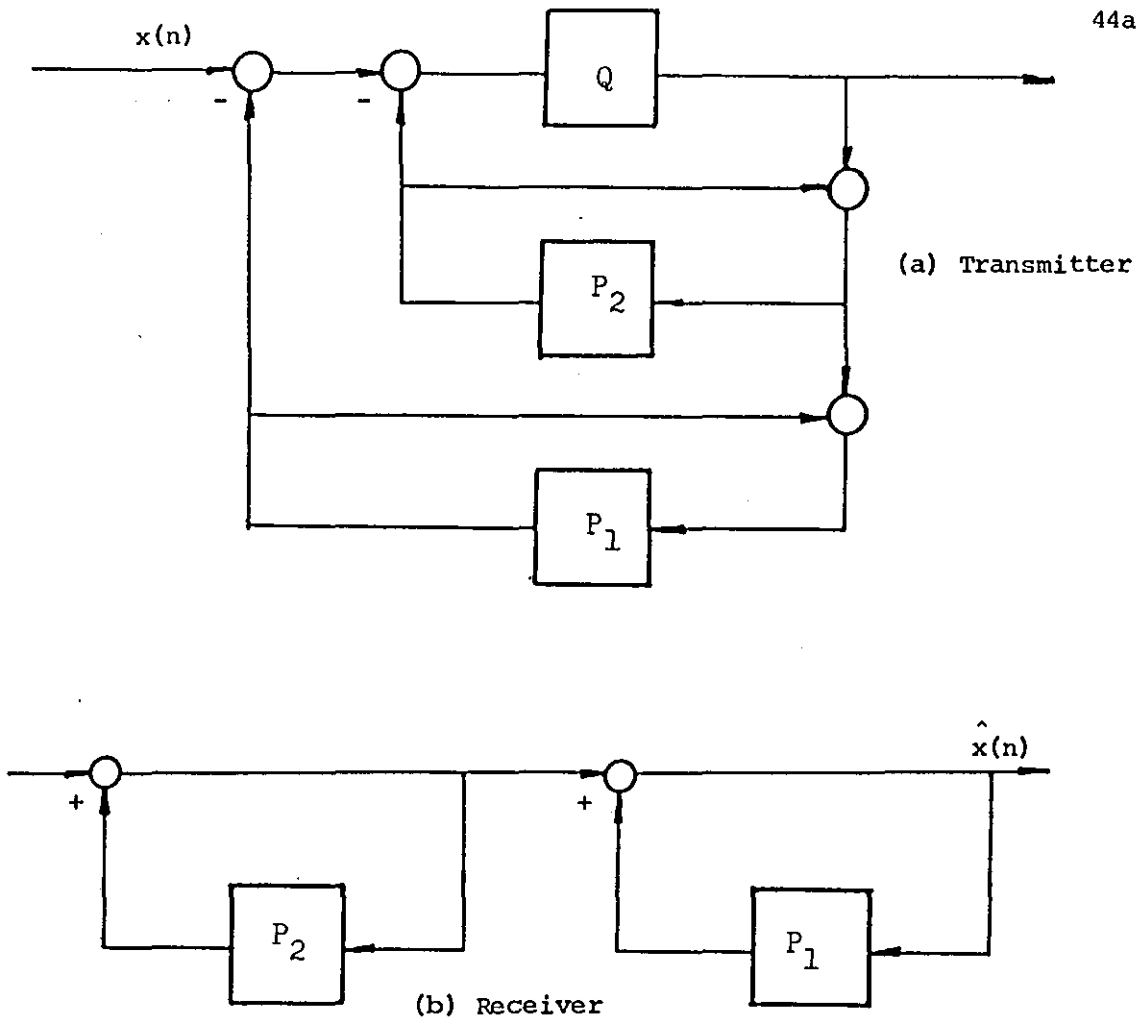


Fig. 2.17 Bloc Diagram of APC System

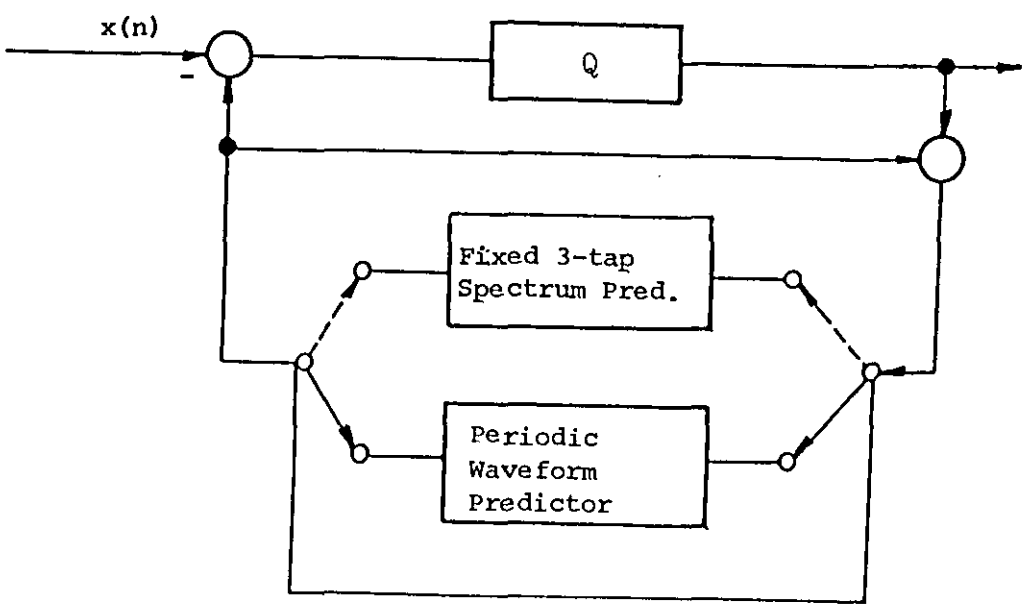


Fig. 2.18 Block Diagram of Pitch Adaptive Coder

non-periodic speech segments, switching to a 1-tap pitch predictor (preferably based on the AMDF algorithm) upon detection of strong periodicity. A block diagram of his system is shown in figure 2.18. Xydeas[11,87,88] proposed a similar pitch synchronous DPCM scheme which aligns adjacent pitch periods correctly before obtaining the difference signal to be quantized. This ensures that the prediction residual is always kept very small.

Unlike the short-term predictor, pitch predictors are not easily amenable to backward adaptation, due to the long time lags involved. Attempts to develope viable sequential backward gradient techniques have met with little success. It appears that although differential coding schemes employing pitch prediction offers much potential as an effective means of signal compression, their one major drawback is the dependance on accurate pitch extraction for efficient performance. Apart from the substantial delay incurred (typically one to two pitch periods), accurate pitch detection generally requires highly complex implementations. Indeed, because of the computational complexity involved, the otherwise powerful APC scheme of Atal and Shroeder have not been suitable for use in most real-time applications with current technology[37].

The adaptive predictive coder will be examined at greater length in chapter 3.

2.4.1.5 Delta Modulation (DM)

DPCM coders exploit the high adjacent sample correlation found in Nyquist-sampled speech to produce a difference signal that can be

quantized using fewer levels than PCM, for the same SNR performance. This suggests the possibility of reducing the number of quantization levels even further if signal correlation can be correspondingly increased. Consequently, one could consider a differential coder which uses the minimum number of quantizer levels (2 levels, 1 bit) and a simple predictor in a feedback loop. Delta modulation (DM) is precisely such a one-bit version of DPCM which combines low complexity with good waveform tracking properties[37,89-91]. A thorough and comprehensive examination of delta modulation encoding techniques is given by Steele [89]. In its simplest form, the DM coder operates by approximating an input time function by a series of linear segments of constant slope. Such a coder is therefore referred to as a linear or non-adaptive delta modulator (LDM). Not unexpectedly, as in PCM and DPCM, more efficient versions of DM coders exist, where the slope of the approximating function is variable - and these are referred to as adaptive delta modulation (ADM) systems.

(a) Linear Delta Modulation (LDM)

Figure 2.19 shows the block diagram of a linear delta modulator. The input analogue signal x(t) is appropriately band-limited and sampled at a frequency much higher than the Nyquist frequency, to give the highly correlated sequence {x(n)}. A first order prediction based on the previous locally decoded speech sample is subtracted from the input sample to form the error signal,

$$e(n) = x(n) - \hat{a}x(n-1) \qquad (2.27)$$

e(n) is then quantized by the two-level quantizer (essentially a sign extractor) to yield b(n)(either +1 or -1) which is coded and transmitted. The receiver integrates the received b(n) to give a signal

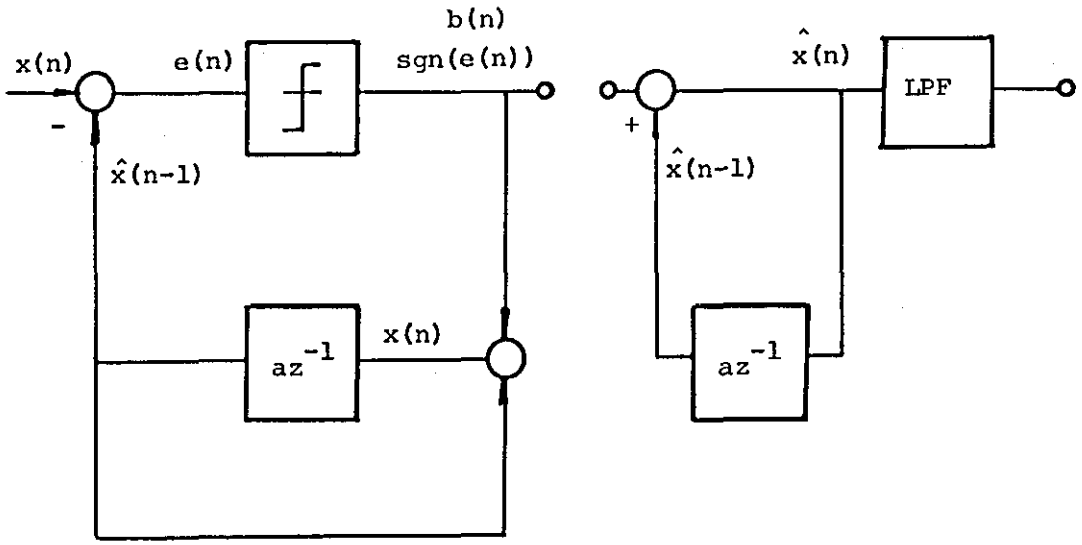
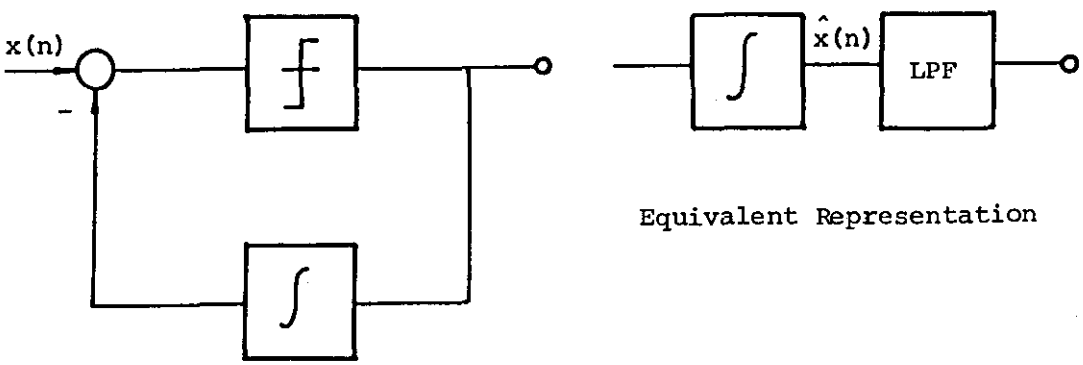


Fig. 2.19 Block Diagram of a Delta Modulation System



Equivalent Representation

that is a staircase approximation of the original speech, i.e.

$$\hat{x}(n) = a\hat{x}(n-1) + \Delta \text{sgn}(e(n)) \qquad (2.28)$$

with a = 1 for perfect integration,

< 1 for leaky integration

where Δ is the DM step-size. Finally, a low-pass filter at the receiver removes the out-of-band noise introduced by the sharp edges of the staircase approximation. The filtered signal yields the recovered speech.

The choice of the step-size Δ in equation (2.28) determines the type and extent of noise present in the DM coder. As in DPCM, the noise in DM coders are either granular noise or slope overload distortion. These are illustrated in figure 2.20. Slope overload occurs when Δ is too

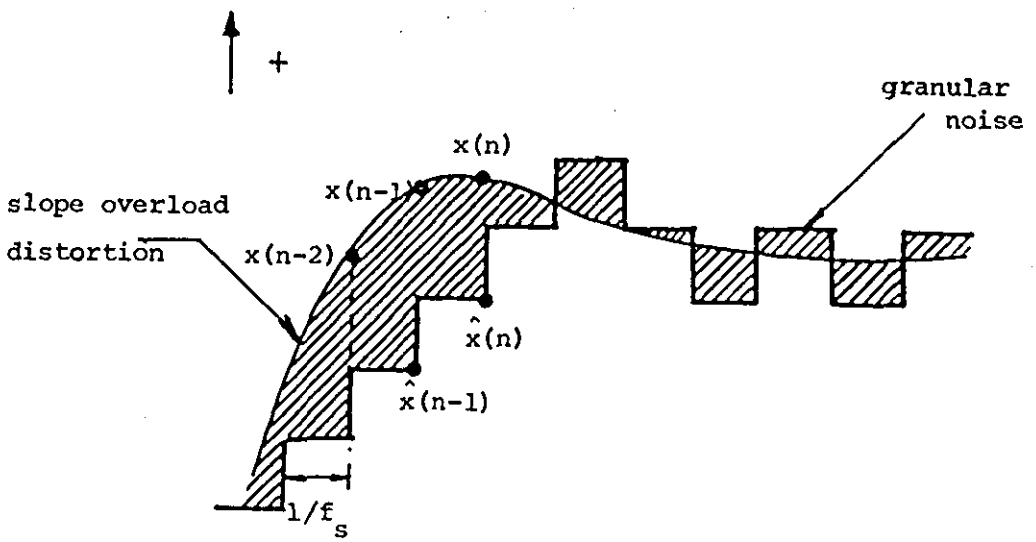


Fig. 2.20 Illustration of quantization noise in Linear Delta Modulation

small and the staircase waveform is unable to track the rapid amplitude changes of the input signal effectively. The error in the decoded signal is thus greater than the step-size. Slope overload may be

avoided if the following condition is met,

$$\left.\frac{dx(t)}{dt}\right|_{max} \leq \frac{\Delta}{T} \tag{2.29}$$

where $T = 1/f_s$ is the sampling period, f_s is the sampling frequency and dx/dt is the derivative of the input signal. For example, if the input signal is a sine wave, $x(t) = V\sin\omega_o t$, then

$$\left.\frac{dx(t)}{dt}\right|_{max} = V\omega_o \tag{2.30}$$

and no slope overload occurs if

$$V\omega_o \leq \Delta f_s \tag{2.31}$$

Granular noise, on the other hand, arises when tracking is correctly maintained but the step-size is too large relative to the local slope characteristics of the input. It is apparent therefore, that small values of Δ accentuate slope overload, while large values increase granularity. Given the input signal statistics, it would be possible to obtain the optimum step-size Δ_{opt} which would provide the minimum total error power. Abate[91] suggested a simple rule for determining Δ_{opt} using the equation,

$$\Delta_{opt} = \langle x(n) - x(n-1) \rangle^{\frac{1}{2}} \ln(2F) \tag{2.32}$$

$$F = f_s/2f_c \tag{2.33}$$

where f_c is the bandwidth of the input signal, and F is the over-sampling index, which is generally much greater than 1. De Jager [90] derived an empirical expression for the quantization noise power, σ_n^2 in LDM systems,

$$\sigma_n^2 = K\frac{f_c}{f_s}\Delta^2 \tag{2.34}$$

where K is an empirical constant. From this expression, the SNR for a

LDM coder may be obtained as,

$$SNR = \frac{\sigma_x^2}{\sigma_n^2} = \frac{f_s^3 \sigma_x^2}{K f_c \Delta^2} \tag{2.35}$$

From (2.31), the maximum amplitude V_{max} of the sinusoid $V\sin\omega_o t$ which does not overload the coder is given by,

$$V_{max} = \frac{\Delta f_s}{2\pi f_o} \tag{2.36}$$

where $2\pi f_o = \omega_o$. Hence, noting that $\sigma_x^2 = V_{max}^2/2$, the peak SNR is

$$SNR_{peak} = \frac{f_s^3}{8\pi^2 K f_c f_o^2} \tag{2.37}$$

Equation (2.37) shows the important result that the SNR in LDM is proportional to the cube of the transmission bit rate.

Research on LDM quantization noise normally involves separate treatments of granular noise (Van De Wag[92], Goodman[93]) and overload distortion (Prontanotarios[94], Greenstein[95]). O'Neal[96] examined both types of noise and estimated the total noise power from the sum of the individual noise variances. Recently, Steele[97], using the expression for slope overload derived by Greenstein, produced equations for the peak SNR of LDM for Gaussian inputs, which are as simple as de Jager's formula and more accurate than Abate's.

The performance of LDM may be improved using double integration[90] i.e. two integrators in series. This allows the prediction samples x(n) to respond faster to the amplitude changes in the input signal, so that a smaller step-size can be used, thereby leading to a direct reduction of granular noise without the penalty of increased overload distortion.

The disadvantage of fast adaptation, however, is the greater risk of instability[76,98]. This problem may be partially overcome using delayed encoding techniques, where the encoder is allowed to 'look ahead' at the input signal and slow down the rate of response accordingly[99,100].

Another LDM configuration is the delta sigma modulator (DSM)[101] shown in figure 2.21, where the integrator is placed in front of the quantizer, and the receiver consists simply of a low-pass filter. With such an arrangement, the error signal is integrated prior to quantization, and slope overload is made independent of the signal frequency.

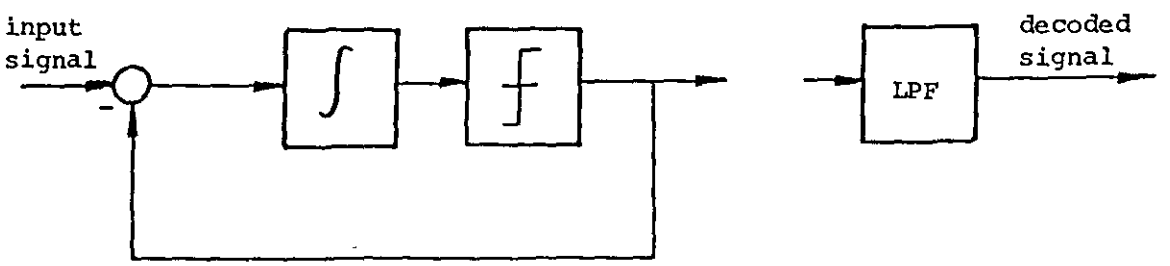


Fig. 2.21 Delta Sigma Modulator (DSM)

(b) Adaptive Delta Modulation (ADM)

From the preceding discussion on slope overload and granularity in LDM systems, it is clear that instead of attempting to obtain a fixed step-size which is a compromise between the conflicting requirements for minimising either distortion, a better solution would be to allow the step-size to adapt optimally to the local signal characteristics.

This is the principle employed in adaptive delta modulation (ADM) systems. Various different ADM strategies have appeared in the literature[37,91,102,103] although the underlying operation is the same i.e. to decrease the step-size when the slope of the input signal is small, but to allow it to expand rapidly upon detection of overload.

The first ADM system to appear in the literature is the high information delta modulator (HIDM) proposed by Winkler[102], and shown in figure 2.22. An adaptation logic incorporated into the LDM structure allows the step-size Δ to adapt according to observations of past quantizer outputs. A sequence of identical bits at the quantizer output indicates a possible overload condition while alternative polarity bits suggest that Δ is larger than necessary. Specifically, the step-size adaptation is as follows:

(i) Δ is doubled if the current and previous two binary outputs are of the same polarity,

(ii) Δ is halved if the last two output bits are of opposite polarity,

(iii) Δ is unchanged in all other cases.

This simple adaptation strategy provides greatly improved dynamic range over LDM. Numerous other variants of this instantaneously companded delta modulator (ICDM) followed. Perhaps the most notable of these is the one-word memory ADM of Jayant[103]. In this scheme, successive bits b(n) and b(n-1) are compared to detect probable slope overload (b(n) = b(n-1)) or probable granularity (b(n) $\neq$ b(n-1)). The step-size adapts according to,

$$\Delta(n) = \Delta(n-1).M^{b(n)b(n-1)} \quad ; M \geq 1 \qquad (2.38)$$

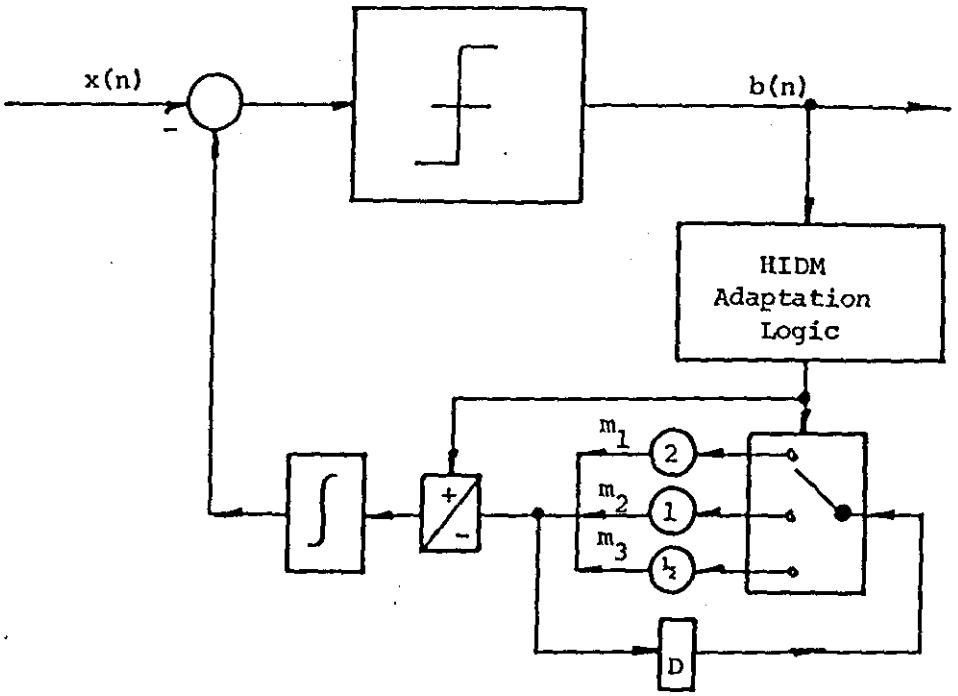


Fig. 2.22 Winkler's High Information Delta Modulator

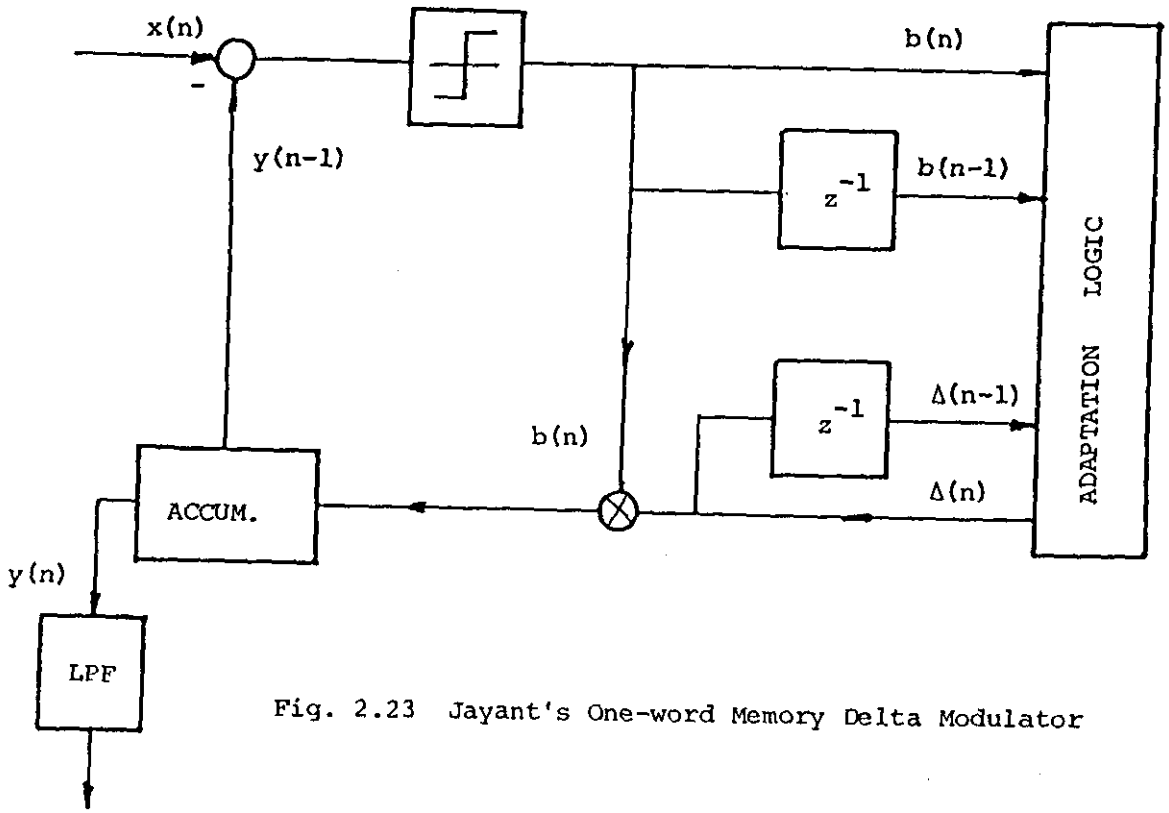


Fig. 2.23 Jayant's One-word Memory Delta Modulator

i.e. Δ is either multiplied by M or 1/M at each instant. The rate of step-size increase or decrease is governed by the single factor M, with M=1 representing the case of non-adaptive LDM. Jayant, using a value of M=1.5 reported a 10 dB advantage over LDM for simulations with narrow-band speech sampled at 60 kHz. A block diagram of the system is shown in figure 2.23. Kyaw and Steele[104] extended this idea to include the effects of the current plus the two most recent polarity outputs. This gives rise to 8 possible binary patterns (3 bits), which are paired appropriately to give 4 different multiplier values. For a Gaussian input band-limited to 3.1 kHz, they reported a 4.5 dB advantage over Jayant's method at 40 Kbps.

A different class of ADM utilises syllabic companding techniques, where the step-size changes much more slowly than the instantaneous adaptations, and follows the variations of the signal envelope[105-109]. Such systems are very robust to errors in transmission. An example is the continuous variable slope delta (CVSD)[108] modulator shown in figure 2.24. The DM step-size is determined by the output bit stream (3 or 4 bits) stored in a shift register. When all the bits in the shift register are of the same polarity, a pulse H is generated, and activates the syllabic filter (with a suitably adjusted time constant). A pulse of height H_o (which is usually much smaller than H) is added to H to ensure that the minimum step-size is not zero. The output of the syllabic filter, with coefficient a_2 is multiplied with the transmitted bit to give the step-size $\Delta(n)$ which is fed to the leaky integrator with coefficient $a_1 = 0.99$. The step-size adaptation is thus,

$$\begin{aligned} \Delta(n) &= a_2\Delta(n-1) + (1-a_2)(H+H_o) \quad \text{for} \quad b(n) = b(n-1) = b(n-2) \\ &= a_2\Delta(n-1) + (1-a_2)H_o \qquad \text{otherwise} \end{aligned} \tag{2.39}$$

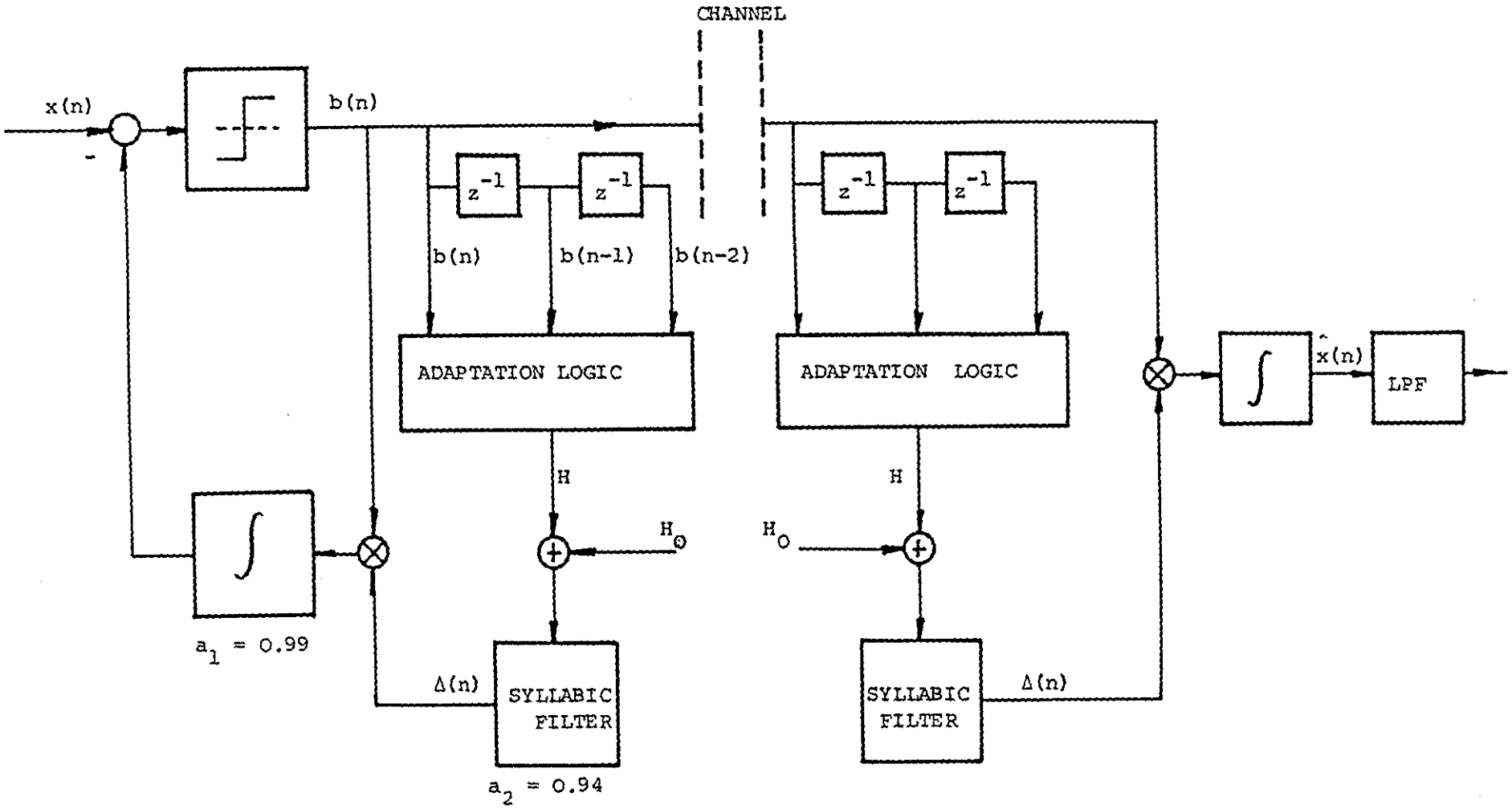


Fig. 2.24 Continuous Variable Slope Delta (CVSD) Modulator

Other versions of syllabic companded ADM systems include the proposal of Tomozawa and Kaneko[109], Brolin and Brown[105], and the continuous delta modulator (CDM) of Greefkes and De Jager[107], which incorporates an extractor for the signal envelope used for the control of the step-size.

Finally, forward transmission of the DM step-size has also been proposed with ADM[46]. Such systems, denoted ADM-AQF operates on the same principle as ADPCM-AQF - the optimum step-size is calculated from a block of input samples and transmitted to the receiver. The explicit transmission of the step-size provides better robustness to channel errors.

2.4.1.6 Other Differential Coder Configurations

DPCM, APC, and DM are all particular cases of the broad class of differential encoding systems. Indeed, it can be seen that the APC structure of figure 2.17 collapses to the DPCM coder (figure 2.16) if the pitch loop is removed. Additionally, if the quantizer is reduced to just two levels, and the predictor restricted to one tap, the delta modulator of figure 2.19 results. A thorough survey of this class of differential encoding system structures is provided by Gibson[19]. Apart from the more familiar coders discussed hitherto, several other configurations are of interest.

(a) Noise Feedback Coder (NFC)

The noise feedback coder (NFC)[81,110-113], illustrated in figure 2.25, operates on a different principle from DPCM or APC. Instead of using feedback to predict the input signal, the goal of NFC is to shape the

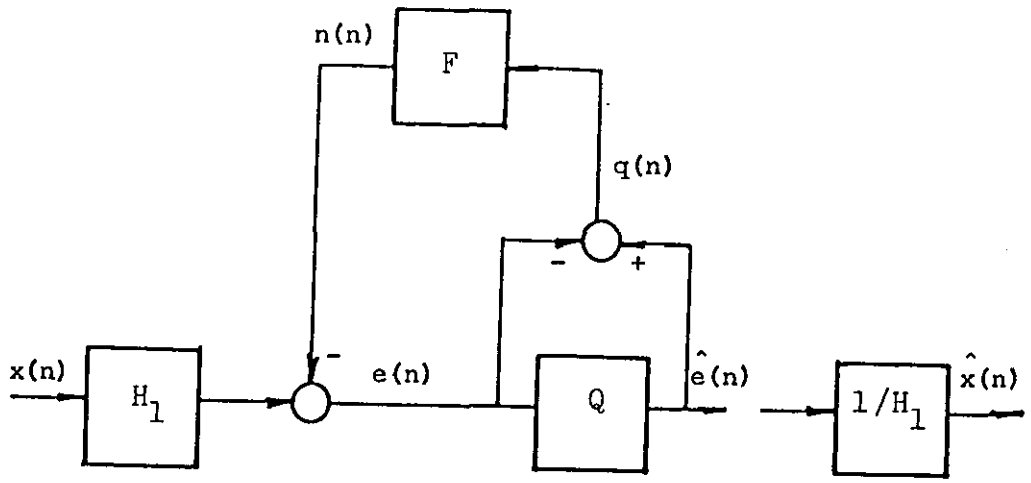


Fig. 2.25 Noise Feedback Coder

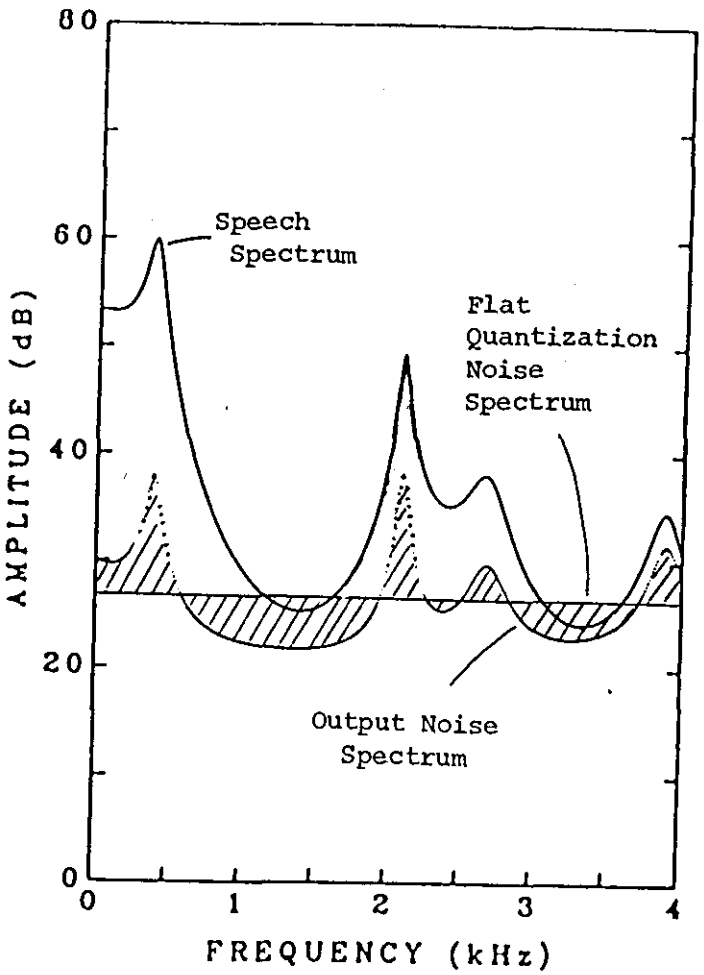


Fig. 2.26 Illustration of the Shaping of the Output Noise Spectrum for Perceptual Improvement

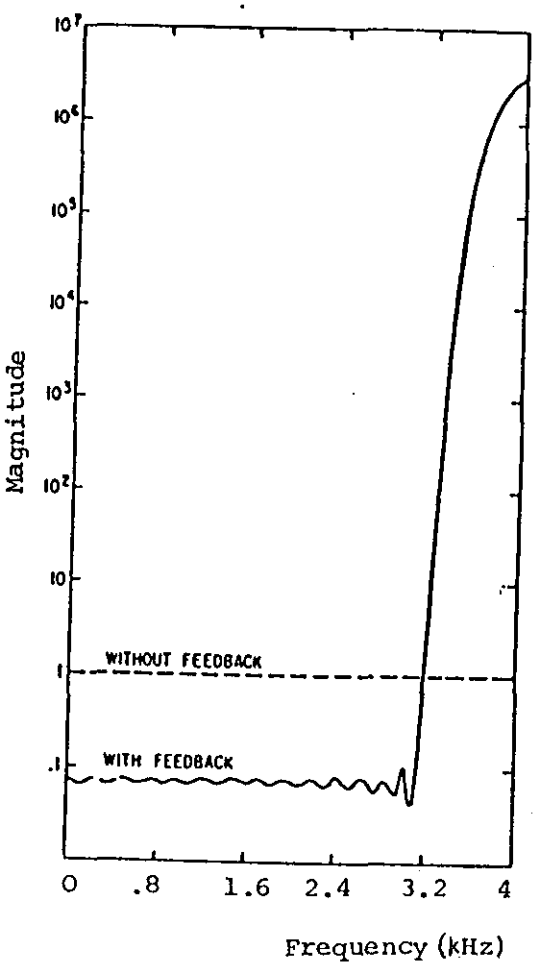


Fig. 2.27

Illustration of the Reduction of Quantizer Noise by Noise Shaping

output noise spectrum to produce a perceptually more pleasant output. To accomplish this, the quantization noise (i.e. the difference between the quantizer input and output) is fed back through the appropriately adjusted filter F. Frequently also, NFC is used together with a pre-filter H_1 in the transmitter and a corresponding post-filter $1/H_1$ at the receiver. H_1 is thus also available for adjustment, although it is normally pre-selected from redundancy removal considerations. NFC can therefore be used in conjunction with redundancy removing schemes such as APC and DPCM, and recent efforts in this area have proven quite successful[81,112]. The use of noise spectral shaping in speech coders arises from the theory of auditory masking, which suggests that noise in the low frequency formant region is normally masked by the high energy speech components so that much of the perceived distortion in the decoded speech comes from the high frequency region where the signal level is low[81,115]. The idea then, is to modify the shape of the output noise spectrum (known to be relatively flat for APC/DPCM systems) so that it follows the speech spectrum and remains below it at all frequencies. Figure 2.26 shows the desired shape of the output noise spectrum, together with the speech spectrum and the unshaped typically flat spectrum of APC or ADPCM. It has been shown, under the assumption of white (uncorrelated) quantization noise, that the shaded areas above and below the flat noise level are equal (but note the logarithmic scale of the vertical axis). Thus noise in one frequency region may be reduced only at the expense of greatly increasing it in another region. This however, allows sufficient control of the spectrum to reduce perceptual distortions in the decoded speech, as has been demonstrated by Atal and Shroeder for APC[81], and by Makhoul and Berouti for ADPCM[112].

An earlier approach to the concept of noise shaping employed the NFC to obtain a reduction in quantization noise. This is achieved by proper selection of the noise feedback filter F, such that the output noise is pushed into the out-of-band frequency region (this assumes a sampling frequency greater than Nyquist), where it could be filtered out[111] (see figure 2.27).

The use of noise shaping features in differential coders will be investigated in greater detail in chapter 4.

(b) Direct Feedback Coder (DFC)

Another differential coder structure is the direct feedback coder (DFC) [116], shown in figure 2.28, in which a filter is placed in the forward path, rather than the backward path of the quantizer. If the quantizer uses only two levels, and G_2 is an integrator, the DFC becomes the delta sigma modulator (DSM) of figure 2.21.

(c) Prediction Error Coder (PEC/D*PCM)

A differential coder that is more amenable to mathematical analysis than the preceding configurations is the feed-forward predictive subtractive coder[57], also known as a prediction error coder (PEC)[19], an adaptive residual coder (ARC)[73] or (as will be referred to here, using Noll's notation) as D*PCM[110]. Although attractive analytically, D*PCM has not received much attention because of the effect of 'noise accumulation' at the decoder. This is due to the fact that while the predictors at both transmitter and receiver are the same, their inputs are not. The transmitter predictor operates on the undegraded input while the receiver predictor uses an input that is corrupted by

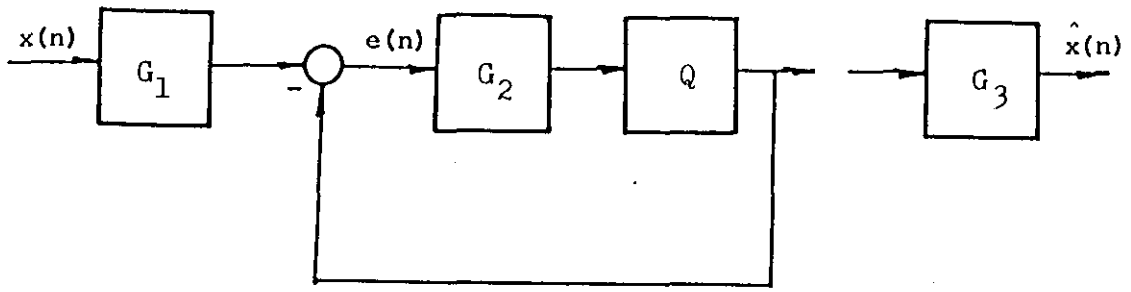
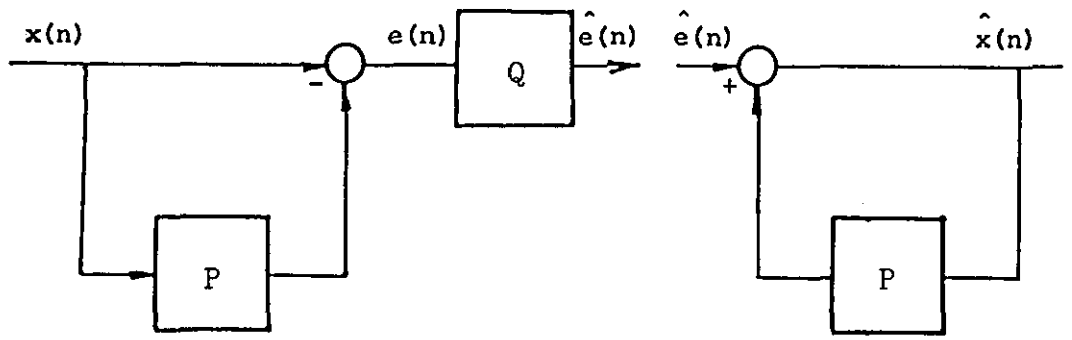


Fig. 2.28 Direct Feedback Coder



(a) Transmitter (b) Receiver

Fig. 2.29 D*PCM Configuration

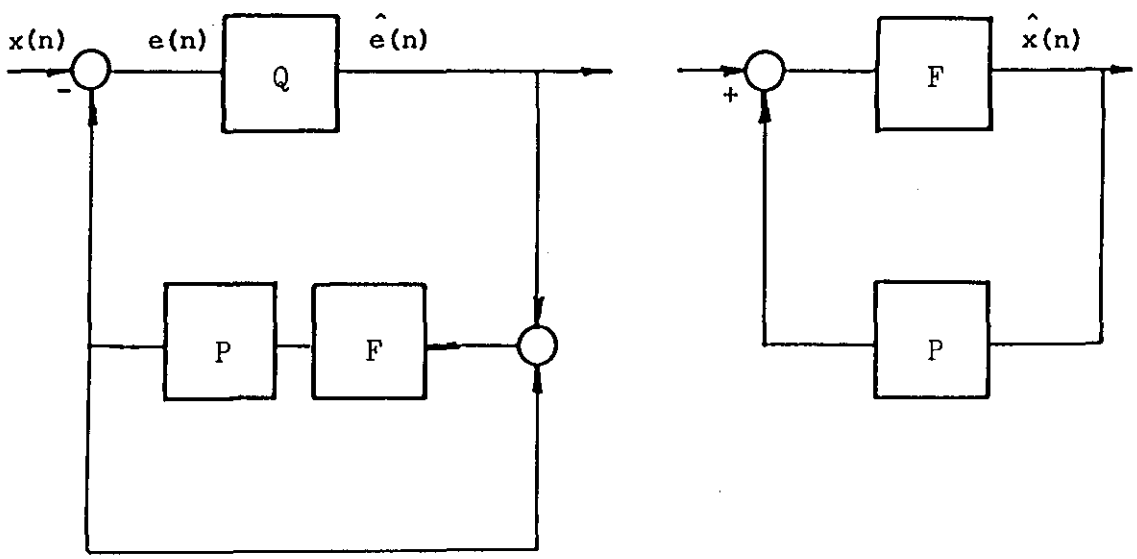


Fig. 2.30 DPCM System with Filtering

quantization noise. It can be easily shown[110] that because of the positive feedback at the receiver, any quantization noise present tends to be emphasised, so that the noise variance of D*PCM is always greater than that of DPCM. Bodycomb and Haddad[117] studied the performacne of D*PCM for Gauss-Markov inputs with the predictor and quantizer separately optimised from a mean square error criterion. They found that D*PCM provided no improvement over direct quantization of the input, because of the noise accumulation effect at the receiver. For speech inputs however, this noise accumulation is offset by the advantage of variance reduction brought about by exploiting the high signal correlation, so that D*PCM provides an overall superior performance over PCM. In fact the effect of noise accumulation produces a shaping of the output noise spectrum which follows closely the frequency response of the receiver synthesis filter [112]. D*PCM can thus be used to provide noise spectral shaping. Indeed, the noise shaping APC coder of Atal and Shroeder[81] employs precisely the basic D*PCM structure, together with a noise feedback filter to provide fine control of the noise spectrum. This can be realised from the noise feedback coder of figure 2.25 by setting $H_1=1-P$.

(d) DPCM with Filtering

Another approach to reduce quantization noise effects in DPCM is to use a filter in series with, and preceding the predictor at the transmitter, and a similar filter in the forward path of the receiver, as shown in figure 2.30. The object of this is to modify the input to the predictor in some way so as to improve its performance. Melsa[118] used a Kalman filter for this purpose in ADPCM, APC and CVSD coders and Gibson[70] employed the same filter in his sequentially adaptive ADPCM system.

2.4.1.7 Entropy Coding

It has been customary, in the design of quantizers, to adjust quantization intervals so as to minimise the mean square error for a given number of quantizer levels L. The optimum quantizer input/output characteristics are thus determined by the probability distribution of the signal to be quantized. Most optimum quantization schemes usually assume that the quantized values are then binary coded for transmission i.e. for L levels, $\log_2 L$ bits are used to code each level. This is equivalent to assuming that all levels are equally likely, which is contrary to the initial assumption of a specific distribution. If, instead of assigning the same length code for every output of the quantizer, a variable code length is used, whereby highly probable levels are assigned shorter codewords and vice versa, then the average code-length would be less than the case where uniform length codes are used, thus leading to a reduction in average transmission bit rate. Entropy coding is one such variable source encoding technique which utilises this principle of unequal code-lengths. When the symbols to be transmitted (in this case the quantizer levels) are independent, it is possible to generate codes with an average word-length approximating the entropy of the symbols. The concept of entropy will now be formally defined.

Suppose that a source S outputs statistically independent symbols S_i, i=1,2,...q, and the probability associated with S_i are p_i, i=1,2,...q. The entropy of the source is defined as[119]:

$$H(S) = -\sum_{i=1}^{q} p_i \log p_i \qquad (2.40)$$

Each S_i symbol can be uniquely represented by a codeword B which is a

sequence of j symbols, $B = (b_1, b_2, \ldots . b_j)$ and B is a member of a finite set of codewords $[B_1, B_2, \ldots . B_q]$ having length ℓ_i. The average length $\bar{L}$ of this coding procedure is defined as:

$$\bar{L} = \sum_{i=1}^{q} p_i \ell_i \tag{2.41}$$

and the following important property of the entropy can be proved,

$$H(S) \leq \bar{L} \tag{2.42}$$

Equation (2.42) shows that the entropy of the source is the lower bound of the average codeword length. This means that the best coding procedure, where codewords B_i are efficiently assigned to source symbols S_i could provide a minimum average codeword length $\bar{L}_{min}$ equal to the entropy of the source. The ratio $H(S)/\bar{L} = E$ is defined as the efficiency of the coding procedure, while (1-E) represents the redundancy.

In waveform coding methods such as DPCM, where signal redundancy is removed prior to coding, the use of entropy coding on the coder output sequence can result in a further SNR improvement at a given transmission rate[120,121]. O'Neal[120] studied the performance of DPCM with entropy coding on signals with a Laplacian distribution and found that when the number of quantization levels is large, entropy coding could provide about 5 dB improvement over normal DPCM. Cohn and Melsa[66], and Qureshi and Forney[67] also employed entropy coding in their ADPCM systems with backward adaptive prediction, and a pitch compensating quantizer. In these schemes, a 5 level quantizer is used, with the 2 outermost levels set further apart than usual, to 'capture' the high amplitude excitation pulses of the residual signal. As these high amplitudes occur very infrequently (typically only 1% of the time),

variable rate coding has to be used to ensure a reasonable transmission rate. Atal also uses entropy coding for his 'APC system with improved quantization'[122] for the same reason.

The use of entropy coding implies the need for a buffer at both the transmitter and receiver, so that a signal coded into a variable length code can be transmitted over a channel at a uniform rate. This also means that a delay proportional to the buffer length will be incurred. Long buffers are clearly undesirable because of the problems associated with excessive delays, while short buffers are more susceptible to overflow and loss of synchronization. Systems employing entropy coding will thus have to incorporate appropriate buffer management measures suitable for the particular environment. Synchronisation of the variable length codes is also an important aspect of entropy coding, and numerous self synchronising codes have been proposed. Possibly the best known of these is the Huffman code, which has been used extensively over the years. The procedure of generating such a code is given by Huffman[123] for the case of binary coding. Makhoul and Berouti employed a simple variant of the Huffman code which is useful in the case of channel errors[112]. The set of codes has all ones in each code, except for the last bit which is zero, as shown in table 2.1. This enables the receiver to re-synchronise every time it receives a zero.

Code Length	Code
1	0
2	10
3	110
4	1110
5	11110
6	111110

Table 2.1 Example of a Self-synchronising Code

2.4.1.8 Multipath Search Coding (MSC)

The performance of most conventional waveform coding schemes is generally poor at low bit rates, when only 1 bit or less is allowed for coding each signal sample. One class of waveform coders, directed at improving the performance at this range of bit rates, uses multipath search strategies based on a delayed decision about binary data representing speech signals[124-133,136-139].

Conventional waveform coders such as PCM and DPCM can be considered as single path coders. They are based on instantaneous decision: the encoder converts an input sample x(n) into a channel codeword c(n), which contains information about x(n) (as in PCM) or on x(n) and its predecessors x(n-1),x(n-2),....(as in DPCM). The decoder converts the received channel codeword c(n) into an output sample y(n). In contrast, multipath search coding(MSC) schemes consider future values x(n+1), x(n+2).... as well, before a (delayed) decision is made about the optimum c(n) to be released. Figure 2.31 shows the structure of MSC schemes. Samples x(n) of the input signal are fed into the input buffer of length N. The encoder compares the buffered samples X_k with a collection of possible output sequences Y_k, $k=1,2,...2^N$, where $Y_k^T = \{y_{k1},y_{k2},....y_{kN}\}$. The collection of these sequences which are either stored or deterministically generated when needed, must be available at

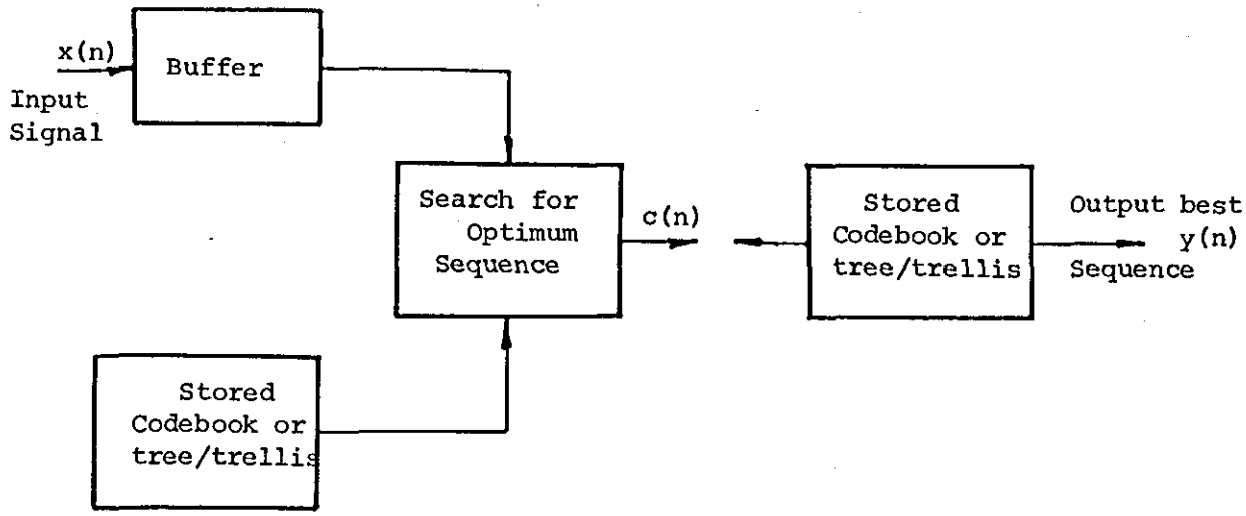
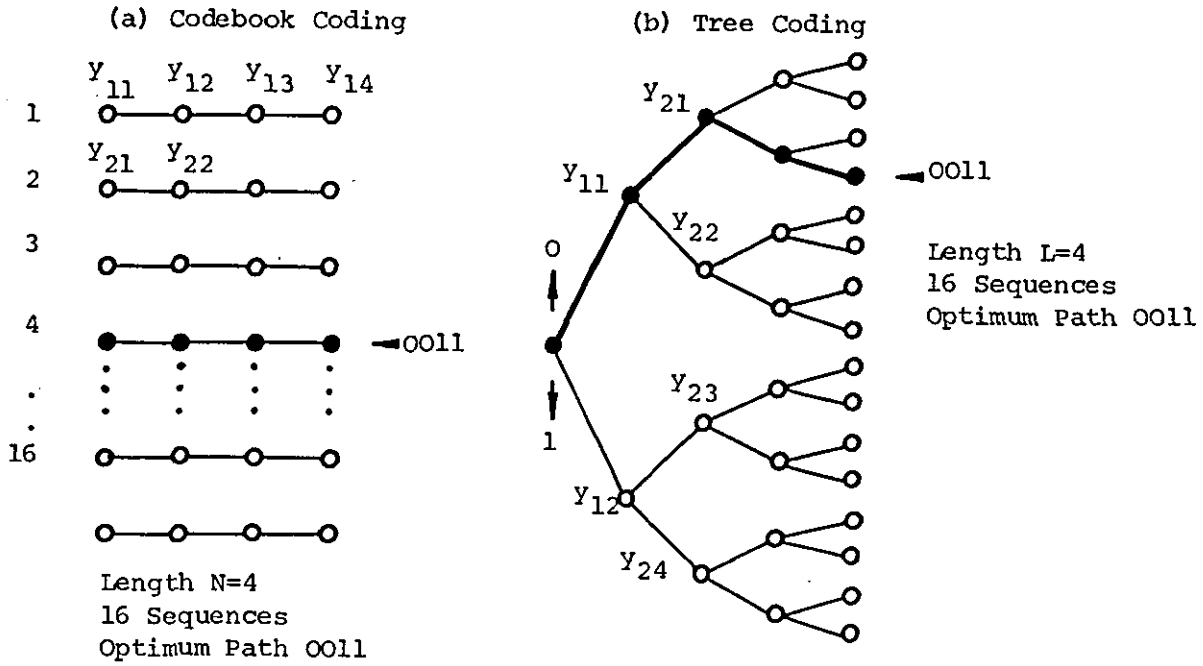


Fig. 2.31 Multipath Search Coding Schematic Diagram



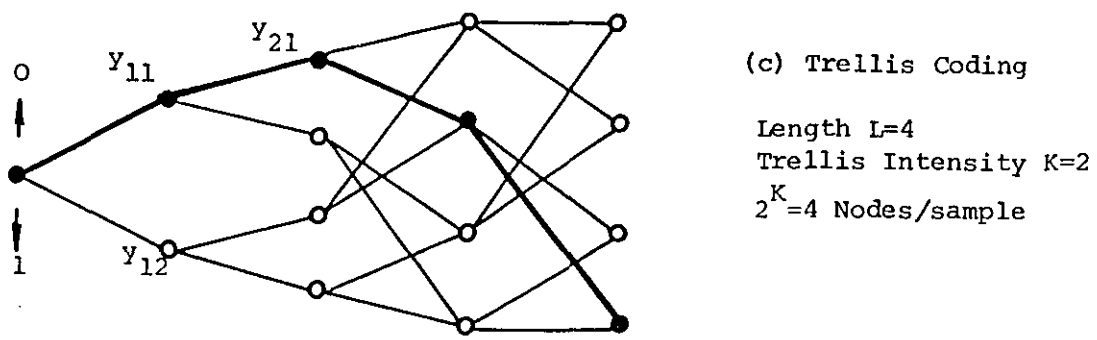


Fig. 2.32 Multipath Search Coding Schemes

both transmitter and receiver. The optimum output sequence is the nearest neighbour sequence i.e. the sequence with the minimum squared error,

$$E_k = (X_k - Y_k)^T(X_k - Y_k) \qquad (2.43)$$

The decoder is informed about the chosen output sequence by a binary channel sequence C_k.

MSC coding strategies may be classified into 3 main classes: (a) Codebook coding (b) Tree Coding (c) Trellis Coding

In codebook coding schemes, also known as list coding or vector quantization[125,126], the set of possible output sequences Y_k, $k=1,2,...2^N$ is arranged in a finite size codebook whose elements are not restricted in any way. When the optimum output sequence has been found, the corresponding index of that sequence is transmitted as the channel sequence in a binary format using N bits (see figure 2.32(a)).

In tree and trellis coding schemes[124,131-134,139], the output sequences of length L are arranged in the form of a tree or trellis of depth L (see figures 2.32(b) & (c)). Its branches are populated with reconstruction values. Different sequences therefore have a number of common elements. Each sequence forms a path through the tree or trellis. The channel sequence, known as the path map, provides information about how to trace through the tree or trellis. There is a slight difference between the tree and the trellis. In tree coding, the number of branches from each node is fixed (typically 2) and the tree expands outwards, doubling the number of possible paths at each stage. For the trellis coder, the number of paths is limited to 2^K per sample, where K is termed the intensity of the trellis. So the trellis starts

as a tree, which then collapses to the specific structure of the trellis when $L \geqslant K$.

The main problem in MSC is to fill the codebook or to 'populate' the branches of the tree or trellis with elements, in a way that 'typical' output sequences result. One method is to generate the elements successively on a sample by sample basis using an algorithm known to both coder and decoder. In such deterministic schemes, possible channel digits not only define a path but they are also assigned amplitude values. Another possibility is to have at the encoder and decoder, stored codebooks or tree/trellis sequences which have been determined beforehand. Such stochastic schemes are much less restricted in providing typical output sequences.

Codebook coding or vector quantization has been applied to the coding of transmission parameters such as the reflection coefficients of an LPC system[134,135]. Buzo and Gray[126] reported equivalent performance in an LPC system using 10 bits/frame vector quantization for coding the transmission parameters as one using 35 bits/frame scalar quantization - an advantage of 25 bits/frame! The criterion used in locating the optimum output code is the minimisation of the widely used Itakura-Saito [134] distortion measure. More recent work[136] suggested that in addition to the reduction in bit rate afforded by vector quantization, better quality synthesised speech, compared to scalar quantization, is also obtained.

Various algorithms for tree/trellis encoding of speech have been investigated by Anderson[137], who reported impressive gains of up to 7 dB over DPCM, in addition to the advantages of better dynamic range and

more resistance to channel errors. A particularly simple and effective procedure is the M algorithm or the (M,L) algorithm[138]. In this procedure, the search progresses through the tree one level at a time, and a maximum of only M lowest distortion paths are retained at each level. At the next level, the next 2M extensions of these paths are compared and the worse M paths eliminated. This process is continued until the level L is reached, at which point the accumulated error over the past L samples is examined and the best path which minimises the error is determined. This algorithm has been used by Atal for his APC scheme[82] and by Jayant and Christensen[138] in conjunction with adaptive quantization. Fehn and Noll[124] obtained more modest SNR gains of about 3 dB in their experiments, and observed that the increases in SNR occurred mainly in voiced speech segments where the SNR values were already rather high. As such, perceptual improvements were smaller than suggested by the SNRs. Other notable contributions in the area of multipath search coding include the work of Matsuyama[127,128], Linde[129], Berger[130], Viterbi[131], Jelinik[132] and Wilson[133].

2.4.2 FREQUENCY DOMAIN TECHNIQUES

In time domain techniques of waveform coding, the input speech signal is treated as a single full-band signal. Redundancy is removed using various means of prediction prior to quantization and coding, and then re-inserted at the decoder. The main differences among the various time domain coders lie in the degree of prediction or interpolation that is attempted, and the differing algorithms for adapting the system parameters.

A more recent class of waveform coders seeks to exploit to a greater extent, the models of speech production and perception, without making the algorithms totally dependant on these models, as in vocoders. This is the general category of frequency domain coders[12,140], in which the approach is to divide the speech signal into a number of frequency components and to encode each of these components separately. By this means, different frequency bands can be preferentially encoded according to perceptual or minimum mean square error criteria for each band, and quantization noise can be contained within bands. Thus, encoding accuracy is always placed where it is needed and indeed, bands with little or no energy may not be encoded at all.

The variety of algorithms in frequency domain coding is perhaps not as diverse as in the more traditional time domain methods. The complexity associated with techniques in the frequency domain may well be a possible reason for this rather lesser interest in such schemes, but advances in device technology are gradually changing the situation. Two techniques in the class of frequency domain coders which have received possibly the greatest amount of interest in recent years are sub-band coding (SBC) and adaptive transform coding (ATC). These have been reported to provide good quality speech at relatively low bit rates.

2.4.2.1 Sub-band Coding (SBC)

In the sub-band coder[12,141,142], the speech spectrum is partitioned into typically 4 to 16 contiguous bands by means of a bank of band-pass filters. Each band is then low-pass translated and downsampled to a frequency twice its bandwidth and digitally encoded using adaptive step-size PCM (APCM) with an accuracy determined by some appropriate

criterion, subject to the number of bits available. At the receiver, the reverse process is performed - the sub-band signals are upsampled, translated back to their original frequency location and summed to give a close replica of the original speech signal. A block diagram of the sub-band coder is shown in figure 2.33.

Apart from the advantage of containing quantization noise within bands, encoding in sub-bands also enables the use of different adaptive quantizer step-sizes in different bands. Thus bands with lower signal energy will have smaller quantizer step-sizes and contribute less noise. In practice, a large number of bits is usually allocated to the lower frequency bands where pitch and formant structure must be accurately preserved to retain speech fidelity. For the higher frequency bands where fricatives occur, a much smaller number of bits is normally adequate. At the same time, this process of bit allocation can also be used to control the shape of the output noise spectrum to satisfy perceptual consideration.

Early versions of the sub-band coder employ large finite impulse response (FIR) band-pass filters[143] to partition the speech signal into sub-bands. Each sub-band is then low-pass translated (by a modulation process), sampled at its Nyquist rate and digitally encoded. The large FIR filters are necessary to provide very sharp cut-off characteristics to minimise the effects of signal aliasing which occurs during decimation (or down-sampling) of the sub-band signals[144]. Crochiere[141] proposed an integer band sampling method for performing the low-pass to band-pass translations which eliminates the need for modulators, and is thus better suited to hardware realisation. A more elegant approach to split-band coding however, is the use of quadrature

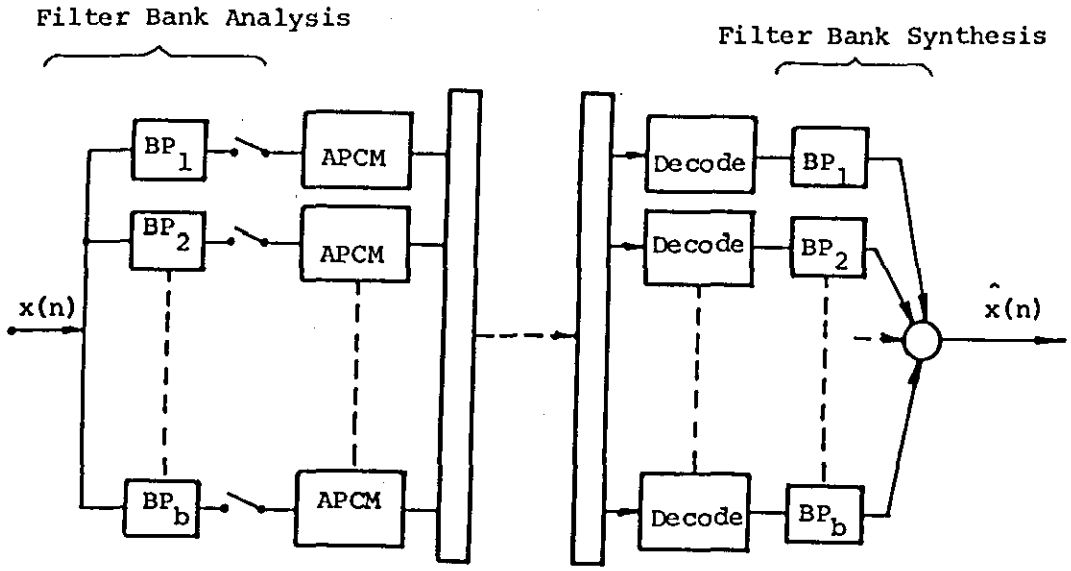


Fig. 2.33 Block Diagram of Sub-band Coder (SBC)

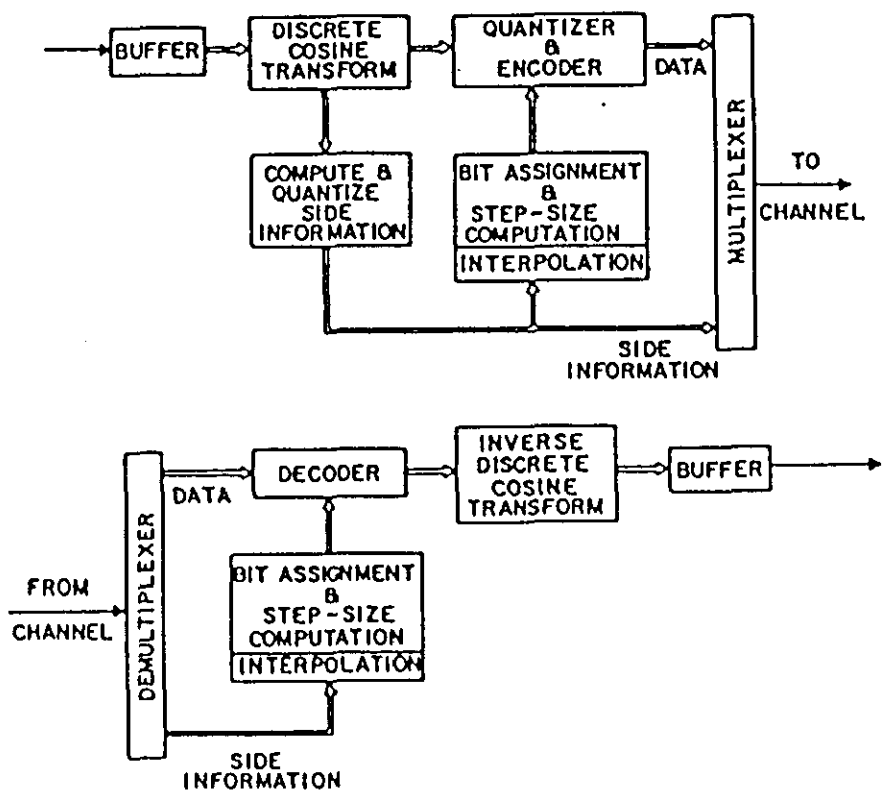


Fig. 2.34 Block Diagram of Adaptive Transform Coder (ATC)

mirror filters (QMF) for performing the band-splitting operation as proposed by Esteban[145]. These filters have the highly desirable property for canceling effects of aliasing and imaging in the sample rate conversion processes and thus allow the use of much shorter filters (32 taps or less). Indeed, the advantages offered by QMF's have reduced significantly the complexity of sub-band coders to the extent that a complete two-band SBC is currently implementable in hardware using just a single signal processing chip [146-148] (see section 2.7.2). Recently also, British Telecom developed a 6 band sub-band coder which uses two signal processing chips, one for the encoder and the other for the decoder[149].

The sub-band coder has clearly been established as a viable technique in speech coding (as evident from the huge amount of interest it has received)[141,142,145-160], offering good quality speech at relatively low bit rates and moderate complexity. The trend in recent research efforts has been toward increasing the number of bands employed in the SBC (to exploit further the advantages of split-band coding) - from the original proposal of 3 or 4, to 8, 16 and even 32 bands[157]. Further discussion on the sub-band coder will be given in chapter 6.

2.4.2.2 Adaptive Transform Coding (ATC)

The adaptive transform coder (ATC)[12,140,161,162] operates on the same principles as the sub-band coder, in that the input speech is divided into a number of bands, and each of these bands is preferentially encoded according to some perceptual or minimum mean square error criterion. The important differences are that, the number of 'bands' involved in ATC is very much greater and that a block transformation,

rather than a filter bank is used to achieve the 'band-splitting'. For this reason, the ATC and the SBC have been described as narrow-band and wide-band analysis/synthesis coders respectively[140].

A block diagram of the adaptive transform coder proposed by Zelinski and Noll[161] is shown in figure 2.34. The transmitter transforms a block of N normalised input samples into the frequency domain using a N-point discrete cosine transform (DCT)[12,164]. These frequency components are then quantized (with different number of bits, determined by an adaptive bit allocation process) and transmitted. The step-sizes of the quantizers are obtained from a coarse description of the short-time DCT spectrum. At the receiver, inverse transformation on the received frequency samples yields the recovered speech.

The ATC coder described above has been reported to provide excellent quality speech at 16 Kbps. Below this bit rate however, quality deteriorates rapidly - a 'low-pass' effect becomes increasingly evident and a 'burbly' distortion is manifested[162]. This is due to the severely inaccurate preservation of the frequency spectrum as the coder becomes 'starved' for bits. Tribolet proposed a more complex low bit rate 'speech specific' ATC coder which uses the pitch information to provide a more detailed estimate of the short-time signal spectrum[165]. He reported good quality speech at a transmission rate of less than 9.6 Kbps using this technique (see figure 2.35).

Other discrete transforms besides the DCT may also be used in transform coding schemes. However, the DCT has been shown to be superior in many ways in its compaction ability for speech and video signals, and to approximate closely the performance of the optimal (signal dependant)

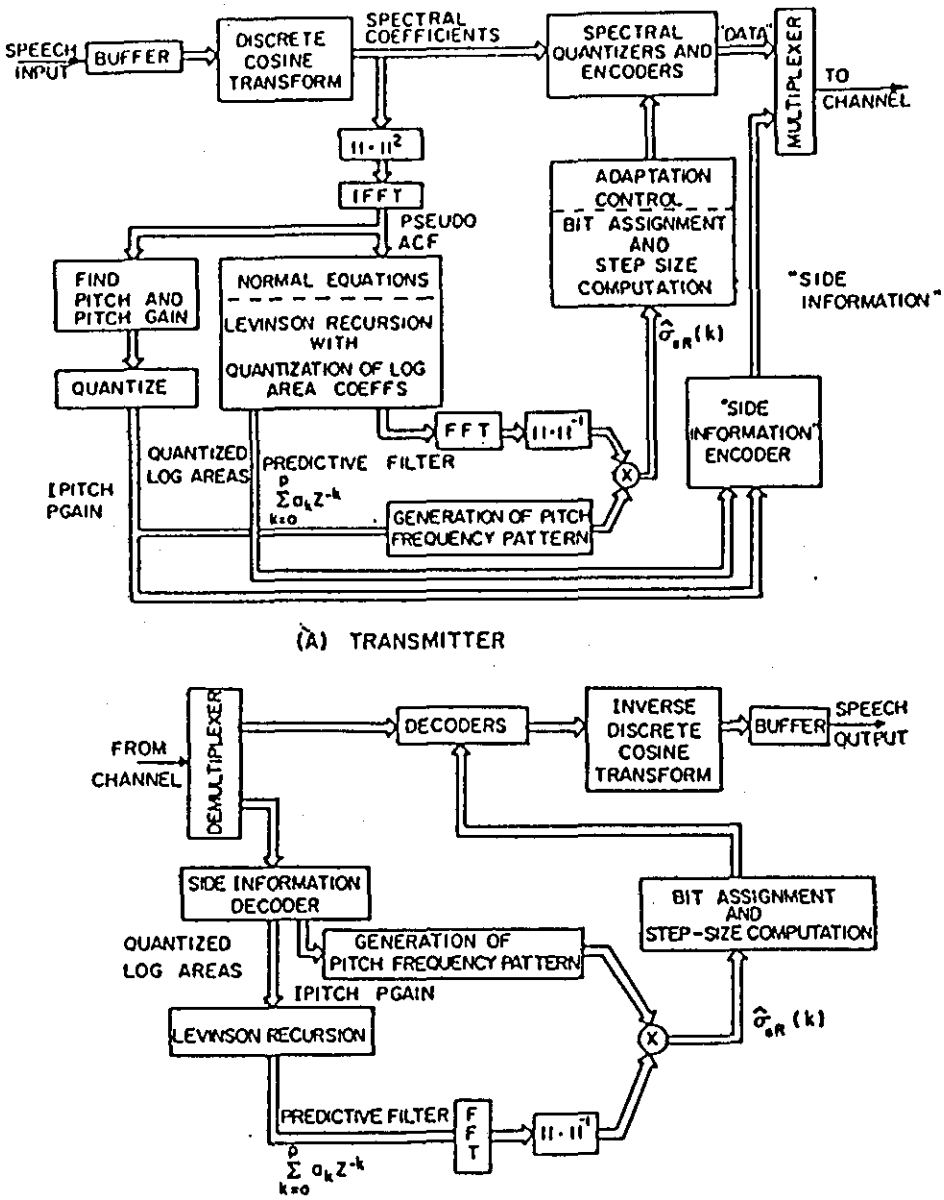


Fig. 2.35 Block Diagram of Vocoder-driven Adaptive Transform Coder

Karhunen-Loeve transform (KLT)[161,166]. Adaptive transform coding techniques will be discussed in greater detail in chapter 6.

2.4.2.3 Phase Vocoder

The phase vocoder, developed by Flanagen and Golden[167] is similar in principle to the ATC and the SBC. Here, the short-time spectral components of speech are converted to magnitude and phase derivative components which are subsequently coded for transmission. Typically 30 frequency channels are used in the phase vocoder, giving it a frequency resolution between that of the sub-band coder and the transform coder. Techniques for adaptively quantizing the channel signals of the phase vocoder, similar to those of SBC and ATC can be used. Portnoff described an implementation of the digital phase vocoder using fast Fourier transform (FFT) techniques[168].

2.4.2.4 Polar Plane Coding

Another related frequency domain technique is that of polar plane coding investigated by Gethoffer[169]. In this scheme, the magnitude and phase components of the input signal are computed and quantized separately with differing accuracy. Good results were reported at bit rates below 16 Kbps using very large transform sizes (up to 8192).

2.5 HYBRID CODING TECHNIQUES

A third general class of speech coding methods utilises various combinations of features associated with time and frequency domain waveform coders as well as parametric and vocoding techniques[12,13].

The voice-excited vocoder is one such hybrid method, where part of the signal is coded using waveform coding methods (either time or frequency domain) and the other part coded by means of parametric representation. Another general class of hybrid techniques attempts direct bit rate reduction by parametrically compressing the speech signal in bandwidth and sampling rate prior to coding, using various harmonic scaling algorithms. Such methods are able to provide high quality speech at relatively low bit rates (< 16 Kbps).

2.5.1 Voice-excited Vocoding Techniques

There has been considerable recent interest in hybrid methods of speech coding which covers the 'middle ground' between waveform coders and vocoders, operating in the range between 4.8 to 9.6 Kbps. This interest arises from several directions[170]:

(1) the demand for a speech quality that is better than that currently available from vocoders - proverbially, vocoders put 'marbles in the talker's mouth', eliminate a talker's individuality so that all talkers sound alike, and make speech sound inhuman and machine-like.

(2) the difficulty and complexity of accurate pitch prediction required by most vocoders - in many practical instances, this sensitivity to pitch errors preclude satisfactory performance.

(3) the unavailability of wide-band channels (data rates above 16 Kbps) due to economical and other factors.

(4) the recent availabilty of modems that operate reliably in the data range rates around 9.6 Kbps over regular telephone lines.

Hybrid methods of speech coding utilise the principles of both waveform

coders and vocoders, in an attempt to provide acceptable and natural sounding speech at a higher bit rate than that required by vocoders. The design of most hybrid coders is very similar to the LPC vocoder (see section 2.3.7), the main difference being that a portion of the original signal or residual waveform (normally a low-pass filtered version of the full band signal) is transmitted in place of the pitch information. In this way, the excitation information is contained in the transmitted residual, and the complexity and difficulties associated with explicit pitch extraction are avoided. At the receiver, some form of high frequency generation is employed to produce a full band residual, which is then applied to the LPC synthesis filter to yield the recovered speech.

2.5.1.1 Residual-excited Linear Predictive (RELP) Coder

Un and Magill[171] described a residual-excited linear predictive (RELP) coder suitable for operation at a transmission rate below 9.6 Kbps. A block diagram of this is shown in figure 2.36. The LPC analysis is performed on overlapping Hamming-windowed speech samples. The prediction residual from the LPC inverse filtering is band-limited to 800 Hz, down-sampled and transmitted using adaptive delta modulation (ADM) with hybrid (i.e. both syllabic and instantaneous) companding (see section 2.4.1.5(b)). At the decoder, the received residual is interpolated to restore the original sampling rate, and then spectrally flattened to generate high frequency harmonics. The spectral flattening process is shown in figure 2.37. The baseband of the residual is retained undistorted in the upper path, while in the lower path, the high frequency harmonics of the residual are generated by

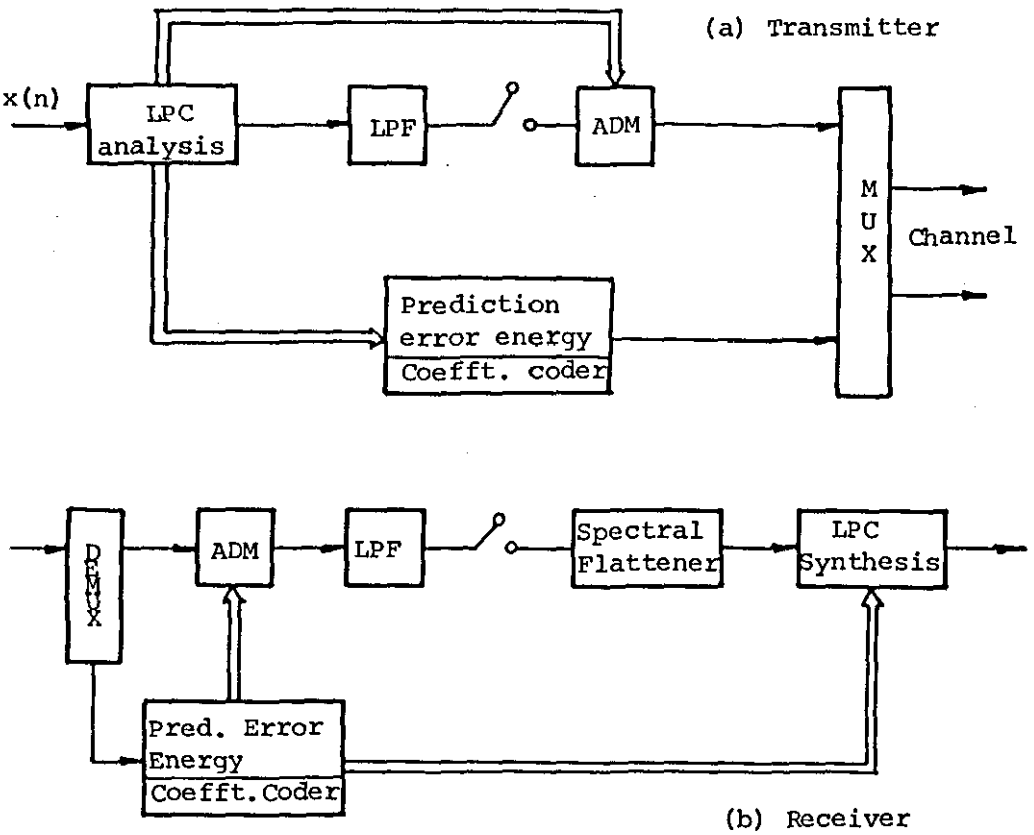


Fig. 2.36 Block Diagram of Residual Excited Linear Predictive Coder

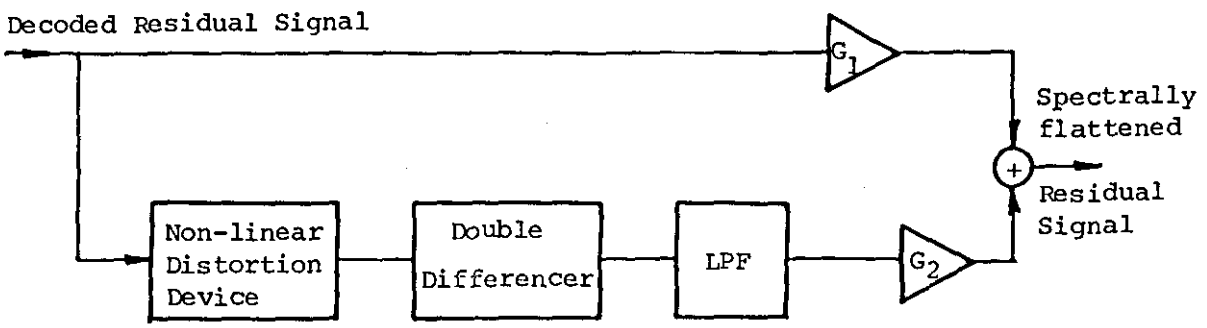


Fig. 2.37 Spectral Flattener for Baseband Residual

full-wave rectification. The energy of these harmonics is further enhanced by a double differencer, and then high-pass filtered to remove signal components in the baseband frequency region. This high frequency signal is then summed with the original baseband in the upper path (with G_1 and G_2 controlling the optimum mix between them) to yield the spectrally flattened residual. The input to the LPC synthesis consists of this spectrally flattened residual plus a suitably controlled amount of random noise. Un and Magill reported significant improvement in the quality of the synthesised speech for this RELP coder, over conventional vocoders. Furthermore, as no pitch extraction is required, the coder is robust in any operating environment, and provides a speech quality which degrades very gradually as the bit rate is lowered from 9.6 Kbps to about 4.8 Kbps.

The LPC analysis for such RELP coders is often performed using the autocorrelation method[33]. The parameters for the synthesis filter (which may be a transversal filter or a lattice configuration) are normally transmitted as reflection (PARCOR)[134,135] coefficients or as log area coefficients (see section 3.3.1)[34]. Frequently, the use of pre-emphasis on the input speech is recommended before LPC analysis [170-172]. This reduces the short-term spectral dynamic range of the signal, enhances the high frequency components present and improves the accuracy of LPC parameter quantization.

2.5.1.2 Voice-excited Linear Predictive (VELP) Coder

A very similar technique to the RELP coder is the voice-excited linear predictive (VELP) coder where the transmitted excitation baseband is obtained from the original speech signal instead of the LPC residual

(see figure 2.38). At the receiver, the decoded baseband speech is added to the high-pass filtered output of the LPC synthesiser to form the reconstructed output speech. Note however, that the term 'voice-excitation' has been used as a generic term to denote both voice-excitation and residual excitation. Viswanathan[170] compared the performance of RELP and VELP coders operating under identical conditions and found that speech from the RELP coder is more 'crisp', less muffled and generally less noisy than speech from the VELP coder.

2.5.1.3 Spectral Flattening

The quality and 'naturalness' of hybrid coders such as the RELP and VELP coders are very much dependant on the high frequency content of the synthesised speech. Since only the low frequency baseband signal is normally transmitted, the process of spectral flattening or regenerating high frequency components in the excitation signal is of considerable significance.

Numerous methods of high frequency regeneration have appeared in the literature[171-181]. It is well known that if the baseband of speech contains either the fundamental pitch or at least two adjacent harmonics, then a waveform containing all the harmonics can be generated by feeding the baseband signal to an instantaneous, zero memory non-linear device. The spectral shape of the regenerated harmonic structure may be quite arbitrary and must be flattened to provide a suitable excitation. Figure 2.39 shows a generalised high frequency regeneration system applicable to voice-excited LPC systems. High frequencies are introduced by applying some form of non-linear distortion to the baseband signal. To avoid 'roughness' in the

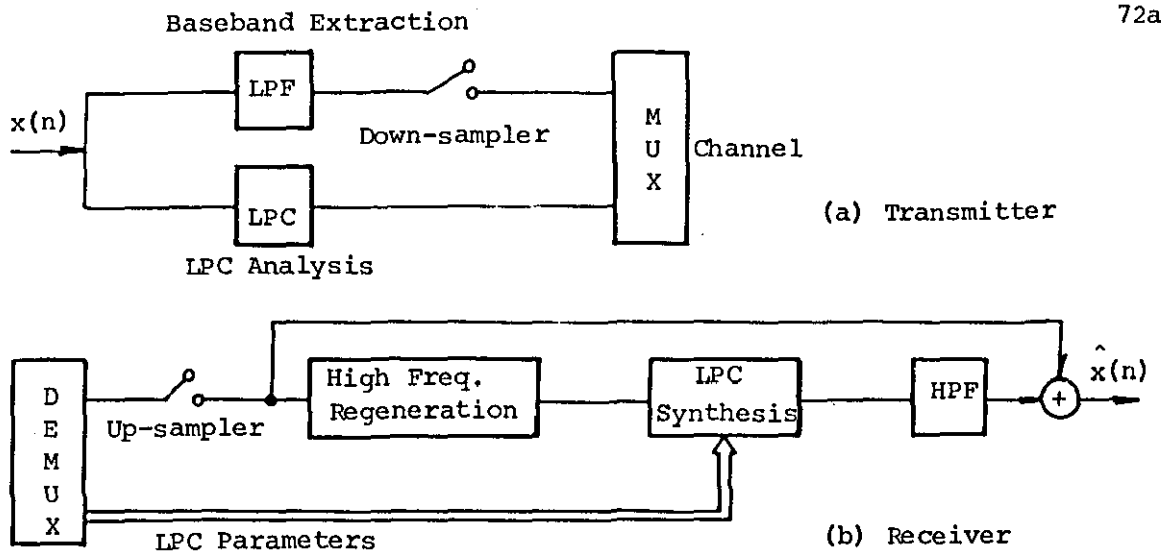


Fig. 2.38 Block Diagram of Voice-excited Linear Prediction Coder

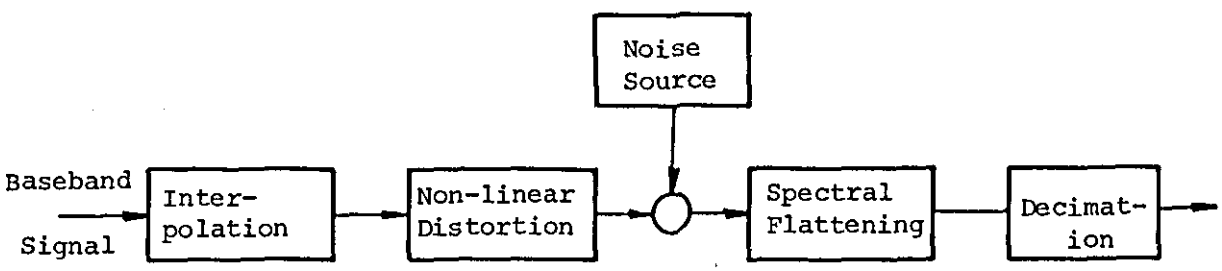


Fig. 2.39 Generalised High Frequency Regeneration Structure for Voice-excited LPC

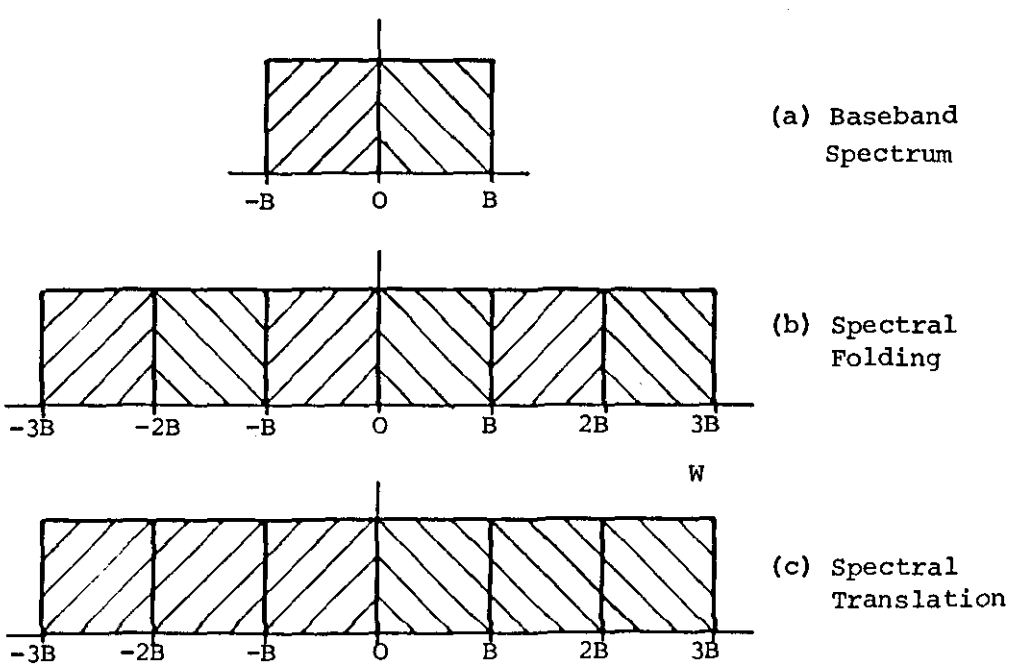


Fig. 2.40 Spectral Duplication Methods for High Frequency Regeneration

recovered speech due to spectral aliasing, it is recommended that the baseband be upsampled to at least twice the original sampling rate before applying the distortion and spectral flattening[171,172]. It will have to be subsequently decimated to the correct sampling frequency before being fed to the LPC synthesiser. Frequently also, a noise source is added to the distorted signal to compensate for the loss of high frequencies in fricatives[171-173], but some proposals have dispensed with its use [170].

Rectification is a commonly used non-linear distortion scheme[172-175]. In general, a rectifier operating on a signal x(t) has the following input/output characteristics[173]:

$$y(t) = 1/2\{(1+\alpha)|x(t)| + (1-\alpha)x(t)\} \quad ;0 \leq \alpha \leq 1 \qquad (2.44)$$

where $|.|$ denotes absolute value, and α represents the extent of rectification, with $\alpha = 0$ giving half-wave rectification and $\alpha = 1$ corresponding to full-wave rectification. Both values, as well as a value of $\alpha = 0.5$, have been used.

Another method of spectral flattening and high frequency generation employs spectral duplication using the transmitted baseband spectrum. Makhoul and Berouti presented two methods, spectral folding and spectral translation, by which this may be done[173]. Figure 2.40 illustrates the two spectral duplication methods, for a baseband with bandwidth B, obtained from a signal with bandwidth W = 3B. Spectral folding is in fact simply the process of upsampling by inserting zeroes between the samples of the baseband. It is important, in performing spectral folding, that the short-term dc value should be subtracted prior to the operation, and added on afterwards, to eliminate a distortion at the folding frequency introduced by the process. Spectral translation can

be done by applying two complementary band selection filters to the spectrally folded full band signal. Spectral duplication gives rise to low level background tones in the synthesised speech, which although different from the 'roughness' characteristic of rectification methods of high frequency regeneration techniques, is not necessarily preferable perceptually. Makhoul and Berouti also suggested an alternative method of spectral duplication which seeks to eliminate these background tones by preserving the harmonic structure of the baseband. This is done by adjusting the width of the baseband spectrum to be a multiple of the short-term pitch fundamental frequency. Frequency domain coding would obviously be easier in this case - and the use of ATC for coding the baseband signal was proposed[173]. A related method of ensuring that spectral duplication is optimally aligned to the harmonic structure of the input speech uses the short-term magnitude and phase components of the speech segment. The magnitude spectrum is duplicated at higher frequencies by shifting it through a pitch adaptive distance[176]. This optimal shift is determined at the transmitter by cross-correlating the high frequency spectrum of the signal with the transmitted baseband and 'peak-picking' the result [175]. Note that this process may also be performed using the cosine magnitude spectrum.

Un and Lee proposed a hybrid method of spectral flattening, in which the high frequency signal is generated by a conventional non-linear distortion device (such as a rectifier) using the baseband, and passed through a band-pass filter[177]. The output of the band-pass filter is added to its baseband and then spectrally folded to yield the full-band excitation. This was reported to result in considerable reduction in the tonal noise associated with straight-forward spectral duplication.

A further proposal used a split-band coding method[172], to split the baseband into two bands, leaving a spectral gap between them to conserve transmission bandwidth. When the non-linear process is applied to these split bands, harmonics are generated at frequencies of integer multiples of the sums and differences of the frequency components in the basebands. Since these frequency components spread more broadly, more high frequencies can be expected.

Numerous other spectral flattening techniques for voice-excited LPC have been proposed, with varying claims for their effectiveness, and these may be found in references 178-180.

2.5.1.4 Baseband Encoding

The coding of the baseband in voice-excited LPC systems may be done as in normal waveform coding using any suitable strategy. Differential coding does not offer any particular advantage in this case, due to the lack of correlation in the signal. ADM with hybrid companding has been used [171,175] as well as log PCM [181] and APCM[170,178]. Abzug[179] utilises an adaptive method of quantizing the baseband, where the signal samples are coded with differing accuracy according to their energy. Sub-band coding of the baseband has also been proposed - Esteban's voice-excited predictive coding (VEPC) scheme employs a bank of quadrature mirror filters to split the baseband into eight equal bands [172]. These are coded individually using a block companding PCM technique, with the number of bits allocated to each band varied adaptively on a block basis.

If spectral flattening is performed in the frequency domain at the receiver, it is more convenient to also code the baseband in the frequency domain. ATC methods using both the cosine and Fourier transforms have been suggested for this purpose[173]. In the Fourier domain, the magnitude and phase components of the baseband may be coded with different accuracy according to their contribution to the perceptual quality of the synthesised speech[176].

2.5.2 Harmonic Scaling Techniques

Harmonic scaling, which has evolved from concepts of phase vocoding (see section 2.4.2.4) is not in itself a speech coding method. It is rather, a pre-processing technique which compresses the input speech by typically a factor of two, prior to coding and transmission, leading to a direct bit rate reduction. At the decoder, the received signal is appropriately expanded by a complementary process to yield the reconstructed speech[146,159,182-186].

Methods of harmonic scaling have been realised in both the time and the frequency domain, and they focus primarily on redundancies in speech due to pitch structure and local stationarity. Time domain harmonic scaling (TDHS) has in particular, been demonstrated to be an effective means of achieving bandwidth reduction whilst maintaining good clean speech reproduction. The TDHS algorithm developed by Malah [182-184], compresses the bandwidth and sampling rate of the input signal by a factor of two at the transmitter and expands it back at the receiver. This is accomplished as a time domain realisation through pitch synchronous processing. Figure 2.41(a) illustrates the compression process. The input speech x(n) is divided into blocks of 2P samples,

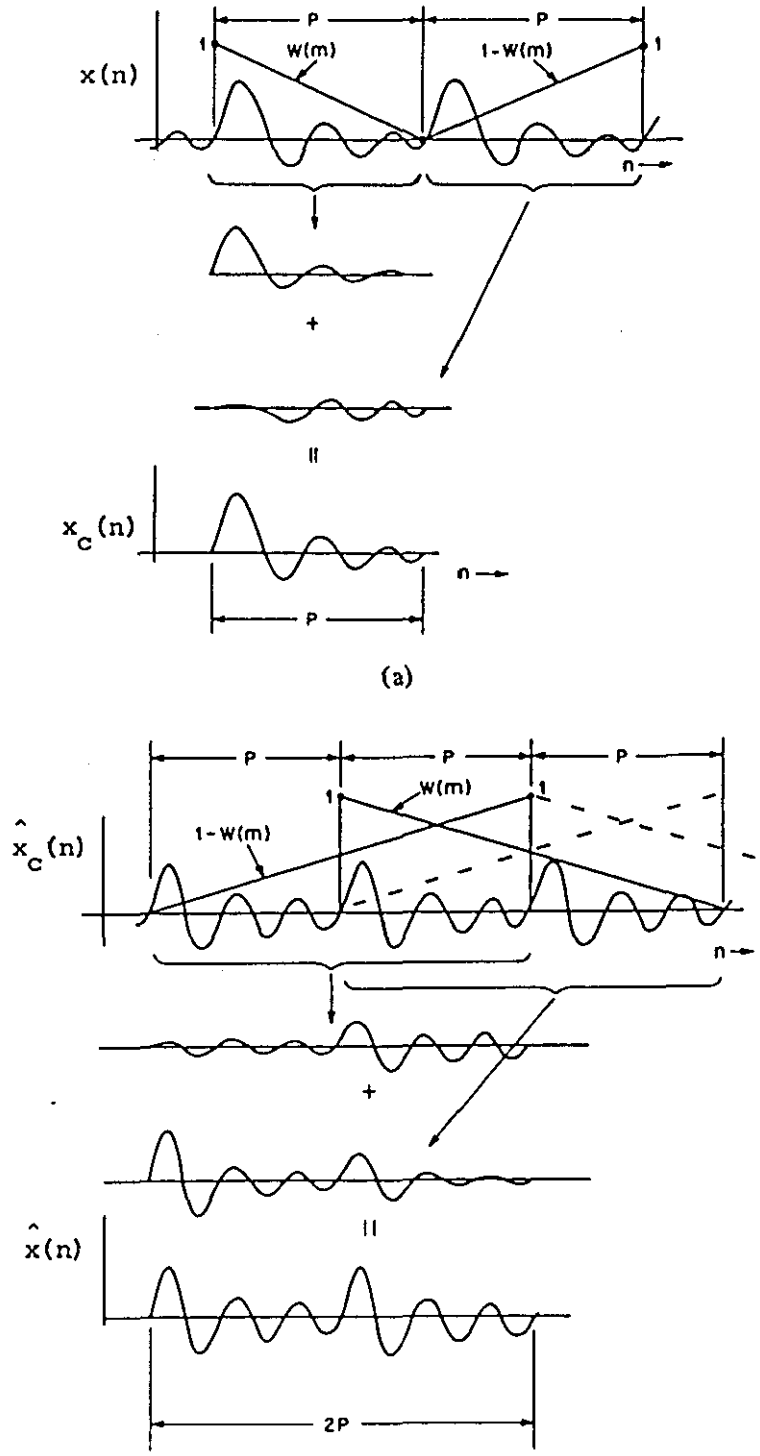


Fig. 2.41 Illustration of Time Domain Harmonic Scaling
(a) Compression (b) Expansion

where P is the measured pitch period. This is compressed to P samples as follows: The first block of P samples is weighted by a window W(n) which linearly decreases from 1 to 0 across the block. The second block is similarly weighted with a window 1-W(n) which linearly increases from 0 to 1. The sum of the two weighted blocks then produces one block of P samples of the compressed signal $x_c(n)$, which looks like the first block of x(n) at its beginning and like the second block of x(n) at the end. In this way, the concatenation of the blocks of $x_c(n)$ forms a continuous waveform without block end discontinuities. The inverse process of TDHS expansion is depicted in figure 2.41(b). In this case, 3P samples of $\hat{x}_c(n)$ (the received $x_c(n)$) are used to compute 2P samples of $\hat{x}(n)$ using the 2P sample overlapped windows shown by the solid lines. The windows are then shifted by P samples and the next 2P samples of $\hat{x}(n)$ are computed in a similar process. Thus, for every P samples of the compressed signal $\hat{x}_c(n)$, 2P samples of the expanded signal $\hat{x}(n)$ are produced, such that $\hat{x}(n)$ is continuous across the boundaries of the concatenated output blocks.

The frequency domain harmonic scaling (FDHS) technique[182,184,186], based on the short-time complex Fourier spectrum, aims at scaling the individual pitch harmonics of voiced speech signals, as in the phase vocoder[167]. However unlike the latter, which uses only the phase derivative, FDHS seeks to perform frequency scaling without discarding the phase information. A qualitative model for frequency division is shown in figure 2.42.

Malah and Flanagan presented a unified description and assessment of TDHS and FDHS and investigated a hybrid scaling method in which compression is performed by TDHS and expansion by FDHS[182]. They

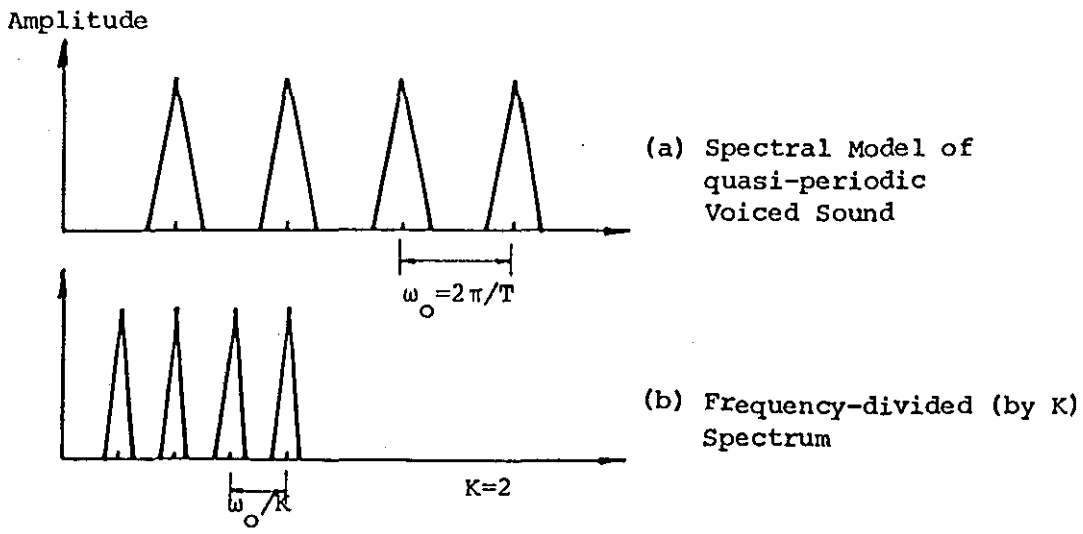


Fig. 2.42 Qualitative Model for Frequency Division of Voiced Sounds

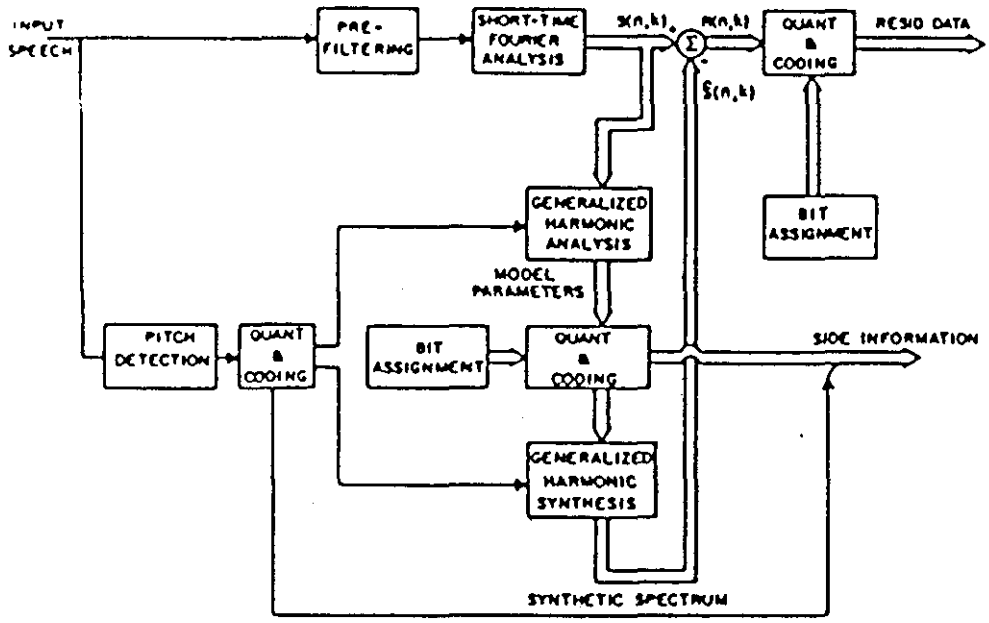
concluded that FDHS is more robust than the simpler TDHS because it is not explicitly dependant on pitch extraction. For clean (i.e. uncorrupted) speech inputs, compression with TDHS results in better reconstructed speech quality. On the other hand, for noisy inputs, in addition to possible failure of the pitch detector at high noise levels, the TDHS expansion process tends to structure the noise, producing a perceptually annoying effect. They also reported that in applications where pitch extraction is feasible but where pitch data transmission is to be avoided, the hybrid TDHS-FDHS system provided better overall speech quality than TDHS or FDHS alone. The additional advantages of the hybrid system, such as reduction of noise structuring and high immunity to channel errors, compared to TDHS alone; and the lower complexity and higher quality, as compared to FDHS alone, makes it the best solution for a variety of applications.

As mentioned above, harmonic scaling is used in conjunction with standard waveform coding techniques, and in this respect, TDHS has

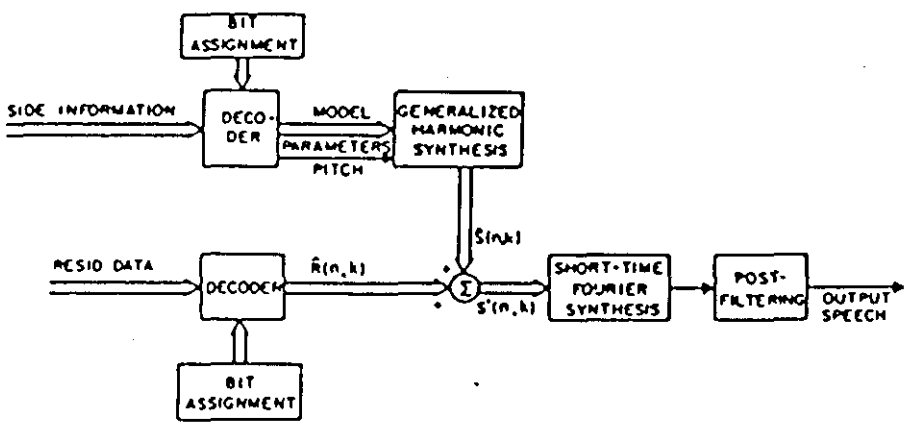
received proportionately greater attention than FDHS because of its relative simplicity and the high quality recovered speech it provides. Malah and Crochiere investigated the performance of TDHS with sub-band coding and adaptive transform coding[159]. They found that bit rate advantages of 7 and 4 Kbps were obtained over SBC and ATC when TDHS is used, at a bit rate of 9.6 and 7.2 Kbps respectively. In addition, TDHS algorithms appear to perform well on the speech of several simultaneous speakers. More recently, Crochiere, Cox and Johnston were able to perform real-time simulations of these combinations using a multi-processor approach[146]. TDHS has also been investigated by Melsa in conjunction with backward adaptive ADPCM (which he termed an 'adaptive residual coder') and variable Huffman coding[185]. He reported good quality speech with a bit rate of 9.6 to 16 Kbps.

2.5.3 Harmonic Coding

Another attempt to close the performance gap between waveform coders and vocoders is the harmonic coder proposed recently by Almeida and Tribolet [187]. Figure 2.43 shows a generalised harmonic coder diagram. At the transmitter, the data is pre-filtered, windowed and transformed to the frequency domain to yield the short-time spectrum $S(n,k)$. This short-time spectrum is then analysed into generalised harmonics, according to the estimate of the pitch. The model parameters i.e. the complex amplitudes of the generalised harmonics are then quantized and used to synthesise the modeled spectrum $\hat{S}(n,k)$, using a non-stationary spectral model. The residual spectrum $R(n,k) = S(n,k) - \hat{S}(n,k)$ is then quantized and transmitted along with the pitch and model coefficients. At the receiver, the residual data are decoded and added to the



(a)



(b)

Fig. 2.43 Block Diagram of Harmonic Coder

(a) Transmitter (b) Receiver

synthetic spectrum, and then fed to the short-time Fourier synthesiser and post-filter. Almeida and Tribolet reported promising preliminary results for high quality speech reproduction using the harmonic coder for bit rates from 4.8 to 9.6 Kbps.

2.6 TRANSMISSION ISSUES

Much of the work on speech coder design largely ignores transmission issues, but sometimes transmission factors are critical to the choice or design of a coding strategy. This section will deal with some transmission considerations[10,12,19].

2.6.1 Channel Errors

For most speech coding studies, the channel is assumed to be ideal. The principal reason for this assumption is that it is necessary to determine whether a speech coder will achieve the desired performance in an ideal environment before complicating the problem with channel effects. Once an attractive speech coder design is obtained however, it is imperative that the effects of channel errors be examined[19]. An investigation into the effects of channel errors on the SNR performance of several speech encoding schemes is given by Noll[188].

Subject to some qualifications and exceptions, one can say that the 'tolerable' bit error rates in most speech coding procedures are in the order of 10^{-3}[10]. One typically gains order of magnitude advantages (10^{-2} or more) by using so-called robust versions of coding algorithms and by using explicit methods of bit protection (error correction/ coding) or by speech smoothing operations at the receiver. For

applications in telecommunication networks, the International Telegraph and Telephone Consultative Committee (CCITT) has defined certain requirements with respect to the performance of speech coding algorithms in the presence of transmission errors[10]:

(1) Algorithms must remain stable at the decoding end when disturbed by an error rate of 10^{-3}.

(2) Coded speech must remain understandable up to this error rate.

(3) With a more common error rate of 10^{-6} or 10^{-7}, the quality must remain subjectively equivalent to (or better than) the PCM quality under the same condition.

Some of the common measures employed to combat the effects of transmission errors will be discussed in the following.

(a) Subdued Quantizer Adaptation

Quantizer adaptation strategies which rely on memory in their adaptation (such as Jayant's one-word memory algorithm) are naturally more sensitive to errors in transmission, because of the effect of error propagation. One method of checking this effect is to allow the error to 'leak' away within an acceptable time, at the expense of a slight degradation in performance. For example, the one-word memory algorithm (equation 2.15) can be replaced by a 'leaky' adaptation logic[189,190],

$$\Delta(n) = \Delta^{\beta}(n-1).M(|H(n)|) \qquad (2.45)$$

where β (typically just smaller than 1, e.g. 63/64) is the leakage factor which controls the speed of error dissipation. This modification, proposed by Goodman, has been employed successfully in time domain (ADPCM) as well as frequency domain (SBC) coding[12]. A similar robust version of ADM is the syllabic (as opposed to

instantaneous) companding continuous variable slope delta (CVSD)[108] modulator with an adaptation algorithm given by equation (2.39).

(b) Subdued Prediction

The same error propagation effect is true for speech coding systems using backward adaptive prediction as noted by Moye[191], Qureshi and Forney[67], among others. The usual approach is again to fade the memory of the adaptive algorithm in some fashion, although this can substantially reduce the efficiency of prediction. One way to restrict this reduction in performance is to fade the memory only when an error occurs, but this would entail added complexity (incurred by incorporating error detectors) and possibly an increased data rate[19].

(c) Explicit Transmission of Coder Parameters/Error Protection

The problem of sensitivity to transmission errors which is inherent in backward adaptive quantizer or predictor strategies may be avoided to some extent by dedicating a fraction of the coder bit rate for explicit transmission of adaptation information. This would obviously be better suited for forward block adaptation strategies such as forward block prediction (forward adaptive ADPCM, APC) and quantization (AQF, ATC). Additionally, when the channel error rates are very high, these parameters can be coded in a special error-protected format, by allowing a further increase in bit rate. Jayant[46] investigated the effectiveness of error protection for mobile telephony employing DM and DPCM and suggested two coders suitable for operation in that environment - a DM-AQF coder with bit scrambling, and an error-protected 3-bit DPCM-AQF. In the latter scheme, the most significant bit (MSB) is transmitted 3 times, the next bit twice and the least significant

bit (LSB) once. At the receiving end, the MSB is determined by a majority vote over the 3 received versions; the magnitude of the middle bit is forced to its smaller magnitude if the two received versions do not agree, and the LSB is accepted as correct. This provides good error protection at the expense of a doubling of the bit rate, and would perhaps only be justified in applications where error probability is high (10^{-2} or more), such as in the case of mobile telephony considered. Steele proposed several error protection coding methods based on statistical criteria for use with DPCM encoding schemes[192-193]. One method transmits a PCM word representing the true amplitude of the signal at the end of every block of DPCM samples. If the decoded DPCM speech differs from the PCM sample, one or more errors exist in the block, and a search based on a simple statistical criteria can be used to locate and correct the erroneous sample(s). Other information derived from the input speech can also be used for error-protection purposes, and another method transmits the maximum difference between adjacent samples within a block. If the adjacent difference between recovered speech samples at the receiver exceeds this transmitted value, then an error is indicated and appropriate correction may be applied.

Crochiere performed an analysis on the performance of 4 and 5 band sub-band coders in the presence of transmission errors[150]. Using the robust quantizer (equation 2.45) and partial bit protection (protecting the sign and the MSB) in the lower sub-bands, he found that intelligibilty of the recovered speech is maintained for error rates as high as 10^{-1}. Viswanathan[194] examined the noisy channel performance of a 16 Kbps APC coder with entropy coding using the Hamming (7,4) code (i.e. protect 4 data bits by adding 3 parity bits) to protect the

important coder parameters, and reported slight degradation in the speech quality for error rates up to 10^{-2}. In many cases, channel errors may lead to instability in the feedback filter of ADPCM systems, especially when the predictor is backward adaptive and of a high order. One recent suggestion inverts the ADPCM predictor structure at both ends, so that the receiver becomes an all-zero filter[195]. Another method implements a 4th order filter using two stages of cascaded second order predictors, where each section is optimised individually [196]. Lattice filters have also been used extensively, in place of tranversal filters in ADPCM or LPC systems[78,79,197-201] - these have the advantage of preventing instability in the decoder filter due to transmission errors, if the filter coefficients are constrained to be in the range +1 to -1 (see section 3.3.2).

2.6.2 Tandem Coding

As present telecommunication networks are still mostly analogue, with digital sections only in some parts, the need for more than one coding-decoding process is not uncommon[10]. Indeed, as a worst case in an international communication, CCITT does not exclude the possibility of up to 14 coding-decoding processes in cascade. Such tandem codings of speech may involve identical or non-identical stages. If the encoding stages are separated by intermediate operations of digital-to-analogue conversions, the distortions introduced by different coding stages tend to be statistically independent, and therefore additive in some sense. Although there is a tendency for quality loss to occur most during the first stage, each subsequent coding-decoding operation will contribute not insignificantly to quality deterioration.

Crochiere[150] investigated the performance of the sub-band coder for up to 4 tandem codings, and found a 3 dB drop in SNR per doubling of the number of tandem coders. Quality degradation is perceptible after three coding-decodings and becomes quite obvious with 4 tandem coders. Le Guyader[200] also studied the effect of tandem coding in ADPCM and PCM systems and found noticeable degradation in speech quality for all systems after 8 coding-decoding processes.

2.6.3 Delay

Another constraint of telecommunication networks is a limitation on the processing delay. Increasing the delay in a telecommunication 4-wire link will make communication more sensitive to echoes. Disturbing echoes can be eliminated by echo suppressors or echo cancellers, but their use is not recommended, for economic reasons. In some applications however, such as satellite communications, the propagation delay is so large that there is no real constraint on processing delay, since the latter typically constitutes only a small fraction of total delay. For terrestial links, the use of echo suppressors can be avoided for up to possibly 20 ms delay, although CCITT recommends values much lower than that[10].

Apart from the very simple algorithms, most speech coding methods utilise some form of 'look-ahead' techniques in order to achieve better signal compression and hence bit rate reduction. Forward block adaptive predictors or quantizers (AQF)[20,41,47] will obviously incur a delay equal to the blocksize of adaptation; typically in the range of 8 to 32 ms. Other methods which employ similar block processing operations, such as ATC, LPC, RELP are also subject to the same delay. For the

sub-band coder, substantial delay is incurred by the filtering process involved in the splitting of the signal spectrum, and this is proportional to the number of bands used. Although quadrature mirror implementations have reduced considerably the length of the FIR filters required, the delay is nonetheless not insignificant. If, in addition, forward adaptive bit allocation and quantization is used, further delays will be necessitated [201,202].

Processing delay is obviously a drawback in terms of the implementability of any algorithm, and should be taken into account in the assessment of a system.

2.6.4 Encryption

One of the attractions of digitised speech is the ease with which it can be encrypted. Digital encryption can be accomplished either by masking speech carrying bits with a psuedo-random binary noise sequence known at the receiver, or by permuting their positions within a block of a certain length. In general, the residual intelligibility from permutation is always higher than in masking, but it does decrease with the length of the block used. Sometimes, the encryption procedure necessitates a delay, which for the block permutation method, is equal to the size of the block. For medium rate speech coding, such as 16 or 24 Kbps ADPCM, a blocklength of 16 (with a delay of 1 ms) would be adequate for providing casual privacy in applications like mobile radio [12].

Although traditionally, encryption or scrambling is used to offer communication privacy, recent research have applied scrambling

techniques to embed data into speech or video signals with significant success[203,204].

2.6.5 Variable Rate Coding

In the design of digital speech coders, it is often assumed that the coder and channel operate at fixed bit rates. In reality however, speech is an intermittent and non-stationary process, and in many applications, user demand on a communication system is variable. In practice, these intermittent properties can be utilised to improve the design of a communication system, such as is done in TASI (Time Assignment Speech Interpolation), or DSI (Digital Speech Interpolation) systems. The other property - that of a variable demand on the system has also been explored for use in packet transmission systems[12,23,206].

In both the above systems, the important element is the variable rate coder. In its simplest form, it may amount to a trivial transmit/no transmit decision as was used in initial TASI systems. In such systems, a group of N users share M channels ($M < N$) at any instant. Only active parts of communications are transmitted, and during pauses between sentences, words or even syllables, the channel is allocated to another active user. Since in a typical conversation, less than 50% of the time on average is spent on active talking, a concentration factor of at least 2 is generally considered possible when the number of channels N is greater than 100. Even then, the probability (however remote) of 'freeze out' exists (i.e. when there are more than M simultaneous speakers). When freeze out occurs, some active channels cannot be transmitted and the effect is subjectively very disturbing. One way to

avoid or at least limit this freeze out effect is to associate TASI with variable length coding schemes - rather than cutting some active channels completely in the event of freeze out, it is subjectively preferable to smoothly decrease the quality of all (or part) of the active channels by assigning fewer bits per sample to them.

Generally, variable rate coding may be characterised according to the configuration shown in figure 2.44, where both the source activity and the channel rate are assumed to be variable[12]. The buffer is used to take up the 'slack' between the source and the channel, and to smooth out fluctuations. A block processing approach is often used, in which a block of N samples is encoded with a total of B bits such that the average transmission rate is B/N bits per sample. The allocation of bits across the block can be made according to rate distortion relations, and is given by the well-known equation[205]:

$$R(n) = \delta + 1/2 \log_2 \frac{\sigma^2(n)}{d^2} \qquad (2.46)$$

where R(n) is the number of bits for coding the nth sample in the block, δ is a constant dependant on the characteristic of the quantizer and the probability distribution of the signal, $\sigma^2(n)$ is the variance of the signal as a function of time and d^2 is the variance of the quantization noise,

$$d^2 = d^2(n) \qquad ;n = 1,2,.....N \qquad (2.47)$$

Dubnowski[206] analysed the theoretical SNR for quantizing the block of N samples using variable rate coding and provided the following SNR formula,

$$SNR\Big|_{var} = 20(\frac{B}{N} - \delta)\log 2 + 10 \log \frac{\frac{1}{N}\sum_{n=0}^{N-1} \sigma^2(n)}{\left[\prod_{n=0}^{N-1} \sigma^2(n)\right]^{\frac{1}{N}}} \qquad (2.48)$$

The first term of (2.48) represents the SNR for a fixed rate coder and the second term gives the improvement in block SNR that is possible using variable rate coding. This gain is given by the ratio of the arithmetic and geometric means of the signal variance across the block. If the speech is highly non-stationary across the block (i.e. the signal variance fluctuates greatly), a large gain can be expected. For a single speaker, speech is locally stationary over about 30-50 ms, and the blocksize N would have to be much greater (> 100 ms) in order to obtain a significant advantage. For the case of multiple users however, as in TASI, P speakers can share a single channel by assigning each user a sub-block of N/P samples and concatenating the sub-blocks into one large block.

An important aspect of variable rate coding is the problem of buffer management. Long buffers can cause unacceptable delays in the system while short buffers are subject to a greater risk of overflow which can cause excessive distortion. Dynamic buffer control techniques have been proposed, based on observations of either the output bit stream of the coder or the input samples. Dubnowski used a method of buffer control which is similar in many respects to the one-word memory algorithm of Jayant[48]. This is given by,

$$d^2(n) = d^2(n-1).H(b(n-1)) \qquad (2.49)$$

where $d^2(n)$ denotes the distortion level in the quantizer at time n, and $H(b(n-1))$ is a multiplier factor which is dependant on the number of bits $b(n-1)$ in the transmitter buffer at time n-1. $H(b(n-1))$ is a monotonically increasing function of $b(n-1)$, being less than 1 when $b(n-1)$ is near zero and greater than 1 when $b(n-1)$ is near B, the buffer

length. When b(n-1) is small, $d^2(n)$ decreases (from 2.56), so that greater quantizer accuracy (i.e. more bits) is permissible. Conversely, when b(n-1) is large, coarser quantization (less bits) will have to be used to prevent buffer overflow and $d^2(n)$ is increased appropriately. An ADPCM system using variable rate coding was also demonstrated by Dubnowski. In this scheme, the ADPCM coder output is framed into packets of 60 bits, with a 2-bit header preceding each packet. Each packet is encoded with either 2,3,4 or 5 bits per sample corresponding to 30,20,15 or 12 signal samples per packet, respectively. The choice of the number of bits to quantize each sample is computed at the transmitter and transmitted as the 2 header bits for each packet. This explicit transmission of the bit information provides more robustness to transmission errors, which might otherwise lead to synchronisation problems. Packet switching is often employed in a network consisting of a number of communication terminals. In such cases, each packet must contain various other overhead information, such as the destination and source, the type of information contained etc.[23]

2.7 HARDWARE ISSUES

Although the bulk of research into speech coding algorithms has been carried out using computer simulations, the ultimate aim of these efforts is to produce systems which can be physically built and used in real-life applications. The last decade has seen a phenomenal advance in device technology and in particular, the advent of high speed micro-processors and programmable ICs. In the field of digital speech coding, implementation of potential coders in hardware has become a

major issue, and numerous special purpose ICs have been developed for this purpose [12,146-148,209]. This section provides a brief survey on the current state of VLSI technology with respect to speech coding applications[207].

2.7.1 Custom Chips and Devices

A number of custom chips and chip sets have recently been introduced that are specifically intended for digital speech applications.[12] In the area of waveform coding, chips for complete μ law A/D and D/A conversions (including anti-aliasing filtering) have recently been developed and are of interest in applications of digital telephony. Chips for ADM (adaptive delta modulation) have also been available, and with growing interest in telephony at 32 Kbps, chips for ADPCM are expected to follow.

In the vocoder area, synthesiser chips or chip sets which realise the speech production model of figure 2.44 have also recently become available. A notable example which has generated considerable interest is the Texas Instrument's 'speak and spell' chip which has been used for voice response in educational toys. A number of other devices for applications in voice response and announcement systems have since followed.

2.7.2 High Speed Microprocessors and Programmable ICs

Another area of VLSI technology that is currently having a strong impact in digital speech applications is that of high speed microprocessors and programmable integrated circuits. A notable example is the Bell

Laboratories' Digital Signal Processing (DSP) IC[146,147,207]. The DSP is a powerful single-chip programmable microprocessor that is especially suited for performing digital signal processing operations. Figure 2.45 shows a block diagram of this processor. Its main elements are:

(1) a 1024 x 16 bit ROM memory for storage of the programs, tables and various constants,

(2) a 128 x 20 bit RAM memory for storage of dynamic data and state variables,

(3) a main Arithmetic Unit (AU) with provision for multiplications, full product accumulation, rounding and overflow protection,

(4) An Address Arithmetic Unit (AAU) with address registers for controlling memory access and provision for updating these addresses,

(5) an I/O unit to control serial data transmission in and out of the circuit, and

(6) a control unit which provides instruction decoding and process synchronisation.

The processor operates with a 800 ns machine cycle time, which is established by a 5 MHz clock.

Crochiere described the implementation of various speech coder algorithms using the DSP[146]. A low complexity design, such as ADPCM using a backward adaptive quantizer (AQJ) can be realised easily on a single chip. In fact, one such ADPCM encoder or decoder uses no more than a quarter of the real-time capability of the DSP, 3 percent of RAM and 15 percent of program memory. This suggests that 4 ADPCM encoders or 4 decoders, or 2 encoder-decoders could be implemented on a single DSP. A medium complexity technique such as the sub-band coder (2 and 4

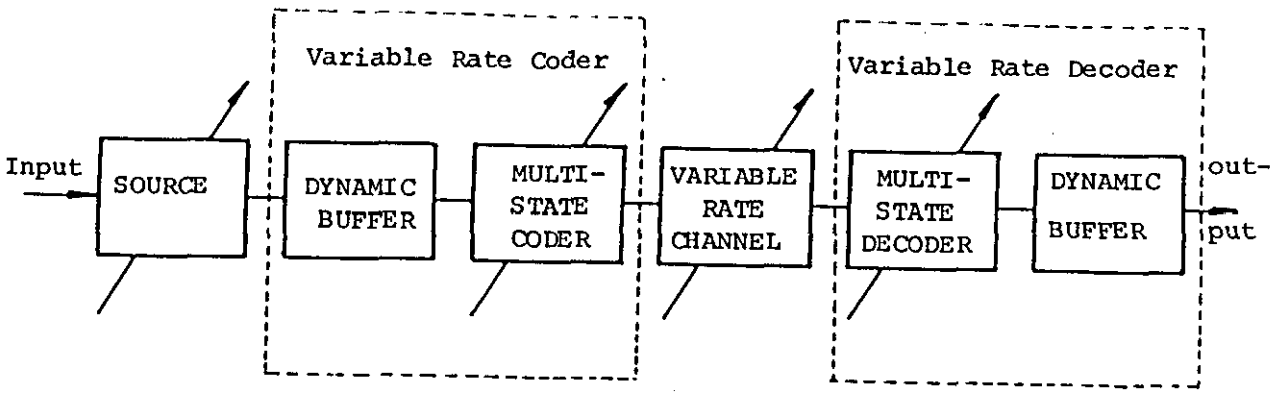


Fig. 2.44 Variable Rate Coding Configuration

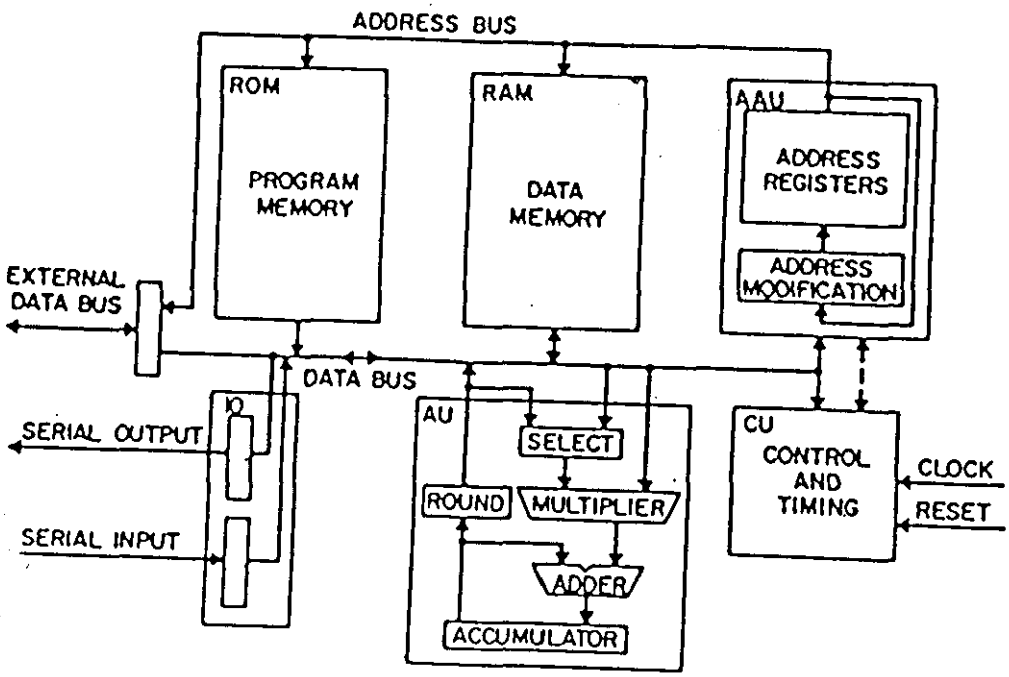


Fig. 2.45 Block Diagram of the Bell Lab's Digital Signal Processor (DSP)

bands) using quadrature mirror filters requires one chip for the implementation of either encoder or decoder, using almost all the DSP capability and memory. For algorithms which are too complex for a single DSP but which can be broken down into smaller modules, a multiple DSP approach was used. An example of such an algorithm is time domain harmonic scaling combined with SBC. This is realised using 3 DSPs for the encoder (which involves pitch detection and harmonic compression) and 2 for the decoder (no pitch extraction needed).

Another signal processing chip gaining widespread acceptance in digital speech applications is the NEC7720[148,209,210]. A block diagram of the chip is shown in figure 2.46. It uses a 'Harvard' architecture, in which the instruction store is separated from data storage. There is space for 512 instructions held in ROM, 23 bits wide. Instructions are of 3 types:

(1) program control,including 32 conditional jumps and subroutine calls,
(2) immediate loading of 16 bit data,
(3) a general purpose format which can simultaneously control 6 different functions.

Data storage is provided separately for fixed data in ROM (512 words) and for variable data in RAM (128 words). The data wordlength is 16 bits, with limited facilities for double length working, and with fixed data held only to 13 bit precision. Arithmetic facilities are provided by a 16 x 16 multiplier which is pipelined into a conventional arithmetic unit, both operating simultaneously at the 4 MHz instruction rate. Data transfer between memory units, arithmetic registers and input-output takes place over a 16 bit internal data bus. The NEC7720 has been used to implement, amongst other things, the LPC vocoder, the

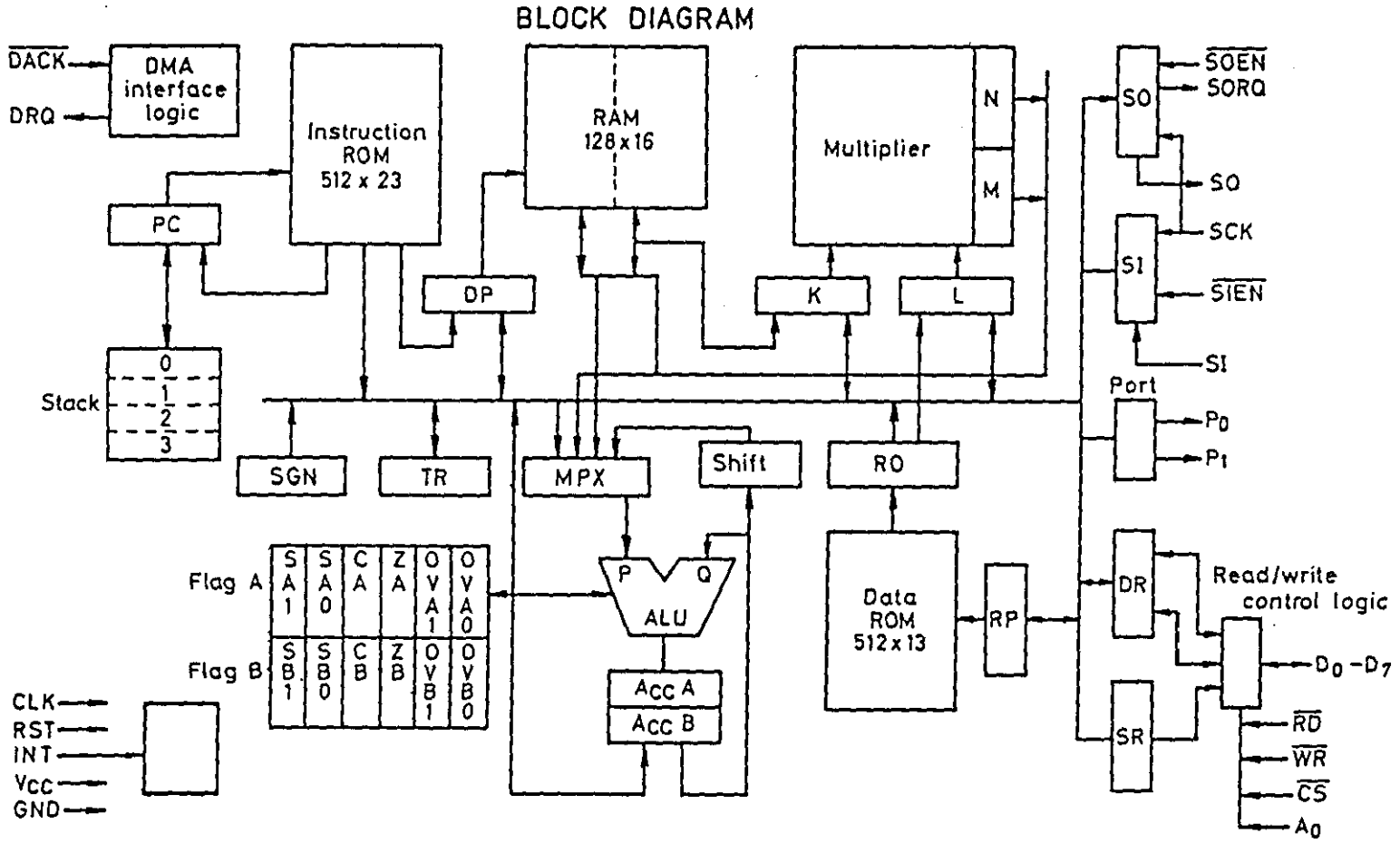


Fig. 2.46 Block Diagram of the NEC7720

channel vocoder and a two-band sub-band coder[148].

One major problem with the present signal procesors such as the Bell Labs' DSP and the NEC7720 is the difficulty of implementing a divide operation, so divisions are usually avoided by various means. For example, in the implementation of the widely used AQJ adaptation, division is eschewed by storing the quantizer step-sizes and inverse step-sizes in ROM[146,148].

It is clear that device integration technology has advanced to a stage where many algorithms regarded as too complex a few years ago, are now being seriously considered for implementation. It is envisaged, with the ever increasing capability and decreasing cost of digital hardware, that the time will soon come when algorithm complexity ceases to be such a critical factor in the choice of a system.

2.8 PERFORMANCE INDICATORS

Although objective performance measures are highly desirable in the assessment of speech coders, these are not sufficiently well established and are generally only used as guideposts in coder design. Formal judgments on coded speech quality must almost inevitably depend on subjective testing. Nevertheless, objective measures such as signal to noise ratios have been useful as complements to the more reliable listening tests. Several performance indicators will be discussed in this section.

2.8.1 Objective Assessment

The single most widely used indicator of speech coder performance is the long-term SNR[9,12,19,20,37,211], defined by,

$$\text{SNR (dB)} = 10 \log_{10} \frac{\sum_{n} x^2(n)}{\sum_{n} \{\hat{x}(n)-x(n)\}^2} \tag{2.50}$$

where the summations are typically over the duration of a sentence length utterance. Since waveform coders attempt to preserve the input signal waveform, the SNR, which is the ratio of the signal variance to the noise variance has the potential of characterising waveform coder quality. Indeed, the SNR would be a meaningless quantity in systems which are not based on waveform preservation. The SNR measure as given in (2.50) is however, strongly influenced by the high energy components of the speech waveform, and does not reflect the performance for low energy segments whose preservation is perceptually very important. An improved measure which takes this into account computes the SNR averaged over short-time segments of active speech (discarding silence). The average segmental SNR over K blocks is defined as[19,20],

$$\text{SSNR} = 1/K \sum_{j=1}^{K} \text{SNR}(j) \tag{2.51}$$

where SNR(j) is the SNR of the jth block, or segment measured in dB. The segmental SNR is a particularly useful performance indicator for coders that adapt quantizers or predictors in a block fashion.

A related distortion measure used particularly for predictive coding systems is the signal-to-noise improvement ratio (SNRI or SNI), also known as the signal-to-residual ratio (SRR), given by[19],

$$\text{SNRI (dB)} = 10 \log_{10} \frac{\sum_n x^2(n)}{\sum_n e^2(n)} \qquad (2.52)$$

where e(n) is the residual signal or the prediction error. This is useful for measuring the effectiveness of the predictor employed in a DPCM or LPC configuration.

Objective measures such as the SNR can also be formulated in the frequency domain, and these might be more relevant to frequency domain coders. Such spectral SNRs reflect the accuracy of preservation of the short-time magnitude spectra of speech segments which are known to be perceptually important. It is also known that the human ear makes a crude Fourier analysis of signals and does not pay much attention to phase - so that some loss of phase information is indeed permissible [2,12,26]. More recent work has aimed to use the short-time spectral envelope to develope perceptually meaningful objective spectral distance measures that can be accumulated over the running signal. The general approach has been to evaluate the short-time amplitude spectrum on a frequency-warped scale (corresponding to the equal articulation bands, or to the critical bands), and a non-linear transformation of spectral magnitudes to approximate the relationship between subjective loudness and amplitude[81,110,140].

Sound spectrograms are also useful in determining how well the spectral characteristics of the recovered signal are matched to the input, and provide an easy visual comparison between different coders. Figure 2.47 shows an example of contour spectrograms obtained from about 5 seconds of male and female speech for the utterance, "There was an old man called Michael Finnegan, he grew whiskers on his chinagen." The

(a)

(b)

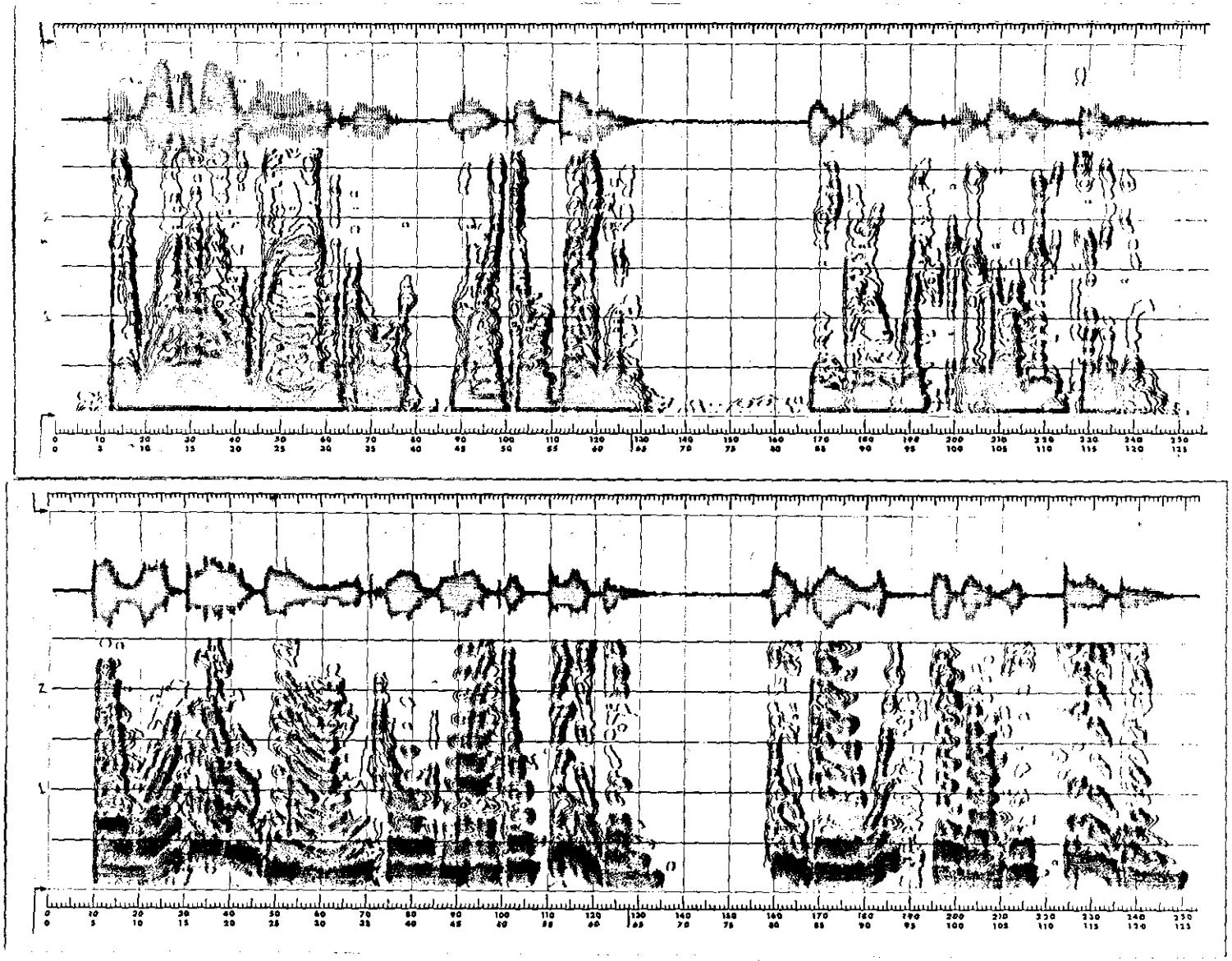


Fig. 2.47 Time Waveforms and Contour Spectrograms for (a) Male (b) Female Speech

speech data is sampled at 8 kHz with a bandwidth from 0 to 3400 Hz. The signal waveform is also shown above the spectrogram, which is essentially a plot of the frequency characteristics across time. The amplitude of the frequency components is indicated by the intensity of the plot. The dark bars in the low frequency region corresponding to the high amplitude formants are clearly shown. Note that the spectrograms shown in the figure are incomplete, as frequency components above 2.5 kHz had been left out to provide room for the input signal waveform.

Finally, the distribution of output noise across the frequency spectrum also provides an indication of the kind of distortion to be expected in the recovered speech. In particular, the long-term average noise spectrum has proved useful in applications where noise shaping is applied to improve the perceptual quality of recovered speech[81, 212-215].

2.8.2 Subjective Assessment

Subjective assessment of a speech coder may be considered under the following sections:

(a) Intelligibility

Speech intelligibility is usually not a problem in waveform coded speech unless the bit rate is very low and there are demanding transmission requirements, such as a high bit error rate or multiple tandem codings. It is perhaps more relevant to vocoder synthesised speech, which is often dependant on the input speech - the sex and age of the speaker, additive noise and other distortions introduced in the system.

Intelligibility is also heavily influenced by speech content - real life speech is often characterised by considerable redundancies, so that a listener may well understand what is being said without having to hear every word. To minimise such effects, intelligibility tests may employ 'logatomes', which are meaningless words with a structure consisting of consonant-vowel-consonant e.g. bon, vin, although there is often an element of unreality about them[199].

(b) Talker Recognition

Talker recognisability is important, not only in telephone conversations among friends, but even more so in many business and government transactions by voice[12]. Again this tends to be a minimal issue in waveform coding. It is important however, in the assessment of vocoders, since many vocoders have a tendency to make everybody sound alike, and thus have poor speaker recognisability[26].

(c) Listener Acceptance

Listener acceptance is probably the most commonly encountered subjective testing procedure[199,216,217]. Its purpose is to produce a comparative rating of several different coders, either as a means of comparing their relative performance or to calibrate a particular coder in terms of others whose degradation are better known (such as log PCM). Typically, subjects are presented with pre-recorded speech material via loudspeakers or headphones, and asked to indicate their responses appropriately. Formal listening tests are often a long drawn out process, and a number of pre-requisite conditions are recommended[183]:

- the listening level must be comfortable,
- the subject matter must be varied, and the phrases used must be phonetically balanced,

- the number of listeners must be sufficient (at least 20)
- reference conditions such as ambient noise, must be defined to ensure repeatability of the experiments.

Many types of tests are available for evaluating the acceptability of a signal of satisfactory intelligibility. In the isopreference method [183], the subject is presented with first, the input signal degraded in a measurable and reproducible manner and second, the input signal delivered by the coder under test. These are presented randomly, in pairs, and the subject is asked to make a forced decision on which he prefers. The value of degradation at which listening acceptance is comparable for the two signals is used to characterise the coder. The most frequently used degradation is multiplicative noise. A related test is the relative preference method, where the coded signal is compared directly with signals affected by a known degree of degradation, such as 7 bit or 8 bit PCM. For communication applications, 7 bit PCM coded speech is generally accepted as the lower limit of quality permissible in the telephone network. A more elaborate testing procedure is the method of judgment by categories of degradation[199,216,217]. The subject is asked to classify, with respect to the input signal, a series of different coder outputs, using a scale containing 5 marking levels covering the whole range of degradation. A commonly used scale is the CCIR scale drawn up by the International Consultative Committee for Radiodiffussion, which consists of the following 5 grades:

5 Imperceptible degradation

4 Perceptible but not annoying degradation

3 Slightly annoying degradation

2 Annoying degradation

1 Very annoying degradation

This corresponds to the rating categories "excellent - good - fair - poor - unsatisfactory" used by Bell Laboratories in Holmdel, New Jersey [217]. The signals are presented randomly in repeated pairs A-B,A-B, where A is the reference signal and B the coded signal. Frequently, a training sequence of about 10 pairs of signals is first presented to the subjects to give them a 'feel' of the experiment. The results of the classification are analysed by plotting the histograms of the votes received for each category of rating of each coder tested. An example of a typical histogram is shown in figure 2.48, where the signal evaluated is obviously of a high quality.

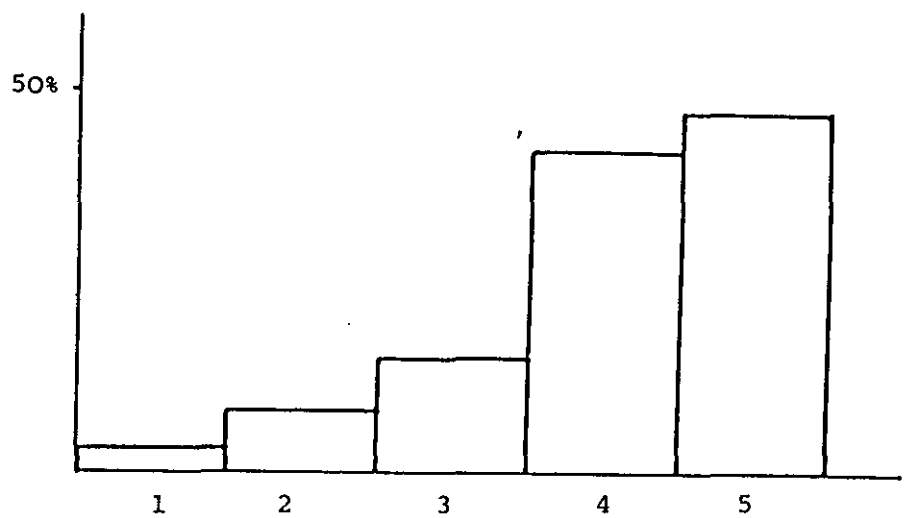


Fig. 2.48 Illustrative Histogram for Subjective Testing Results (Judgment by Categories of Degradation)

Although formal listening tests are probably the most reliable indicators of performance with regards to speech coders, it is not usually resorted to because of the complexities and difficulties involved in carrying out the test. During the early design phases in

particular, informal listening tests, supplemented by SNR values and perhaps spectrograms are often quite adequate[19].

2.9 CONCLUSION

The choice and design of a particular speech coding system for a specific application is often dictated by the requirements of the local environment, and the constraints imposed. With the huge amounts of research efforts expended in the field of speech coding over the years, a wealth of information is available to the designer to cover virtually every conceivable area of interest. Frequently, and particularly when local constraints are not overly rigid or severe, the task of the designer becomes one of deciding among several viable alternatives. This would ultimately involve an exercise in evaluating each potential system, and determining the optimal trade-off between such factors as speech quality, complexity and bit rate, subject to the local conditions. A knowledge of the performance of a wide spectrum of speech coders, together with the operating details associated with each, would thus be essential. This section considers briefly several issues related to the assessment and comparison of a range of speech coding algorithms.

2.9.1 Coder Complexity

Complexity is obviously an important issue in coder design, since it is invariably tied up with implementability and cost. Flanagan, et al[12] provided an approximate ranking of a number of speech coders in terms of complexity, by comparing each to the simple adaptive delta modulator (ADM), which is assigned a complexity factor of unity.

Relative Complexity		Coder
1	ADM	Adaptive Delta Modulation
1	ADPCM	Adaptive Differential PCM
5	SBC	Sub-band Coder (with CCD filters)
5	PP-ADPCM	Pitch Predictive ADPCM
50	APC	Adaptive Predictive Coder
50	ATC	Adaptive Transform Coder
50	ϕV	Phase Vocoder
50	VEV	Voice Excited Vocoder
100	LPC	Linear Predictive Coefficient Vocoder
100	CV	Channel Vocoder
200	ORTHOG	LPC Vocoder with Orthogonalised Coefficients
500	FORMANT	Formant Vocoder
1000	ARTICULATORY	Vocal-tract synthesiser; synthesis from printed English text.

2.9.2 Speech Quality and Transmission Bit Rate

Good speech quality is the ultimate aim of any speech coding system, and this is generally a function of the transmission bit rate. The quality associated with known coders operating at or above a particular bit rate is given below[12]:

Quality	Coder	Bit Rate (in Kbps)
Toll quality	Log PCM	56
	ADM	40
	ADPCM	32
	Sub-band	24
	PP-ADPCM	24
	APC,ATC,ϕV,VEV	16
Communications Quality	Log PCM	36
	ADM	24
	ADPCM	16
	Sub-band	9.6
	APC,ATC, V,VEV	7.2
Synthetic quality	CV,LPC	2.4
	ORTHOG	1.2
	FORMANT	0.5

Toll quality digital transmission can be achieved with simple coders at 40 Kbps (ADM), 32 Kbps (ADPCM) and 24 Kbps (SBC). Mobile radio telephone quality at 24 Kbps with the same relatively simple coders also seems feasible. With increased complexity (APC,ATC), toll quality at 16 Kbps can be attained.

Future research in the ever expanding area of speech coding is envisaged to continue with increased interest. With hardware technology advancing in parallel, attention would be expected to be focussed on pushing the lower bit rate limit for toll quality speech even further, using more sophisticated techniques. At the same time, methods for elevating speech quality at data-coding speeds (7.2 to 9.6 Kbps), and for moderating the effects of transmission errors on conventional systems is an area of substantial interest. Very often, there is a tendency for practical applications in any field to lag quite a way behind current understanding of the subject. While this may be true also in the field of speech coding, it is a healthy sign that international telecommunications organisations such as CCITT are taking an active interest in up-to-date algorithms and research efforts in this area. Indeed, over the past few years, CCITT has embarked on an extensive programme to evaluate potential speech coding algorithms in an attempt to define new standards for future network requirements[10].

CHAPTER THREE ADAPTIVE PREDICTION IN DIFFERENTIAL CODING SYSTEMS

3.1 INTRODUCTION

Speech that is sampled at the Nyquist rate exhibits significant correlation between adjacent samples. This means that a particular speech sample may be predicted to a good degree of accuracy from knowledge of previous samples. This predictability, or redundancy property is exploited by differential coding schemes in which a signal obtained from the difference between the original signal sample and a prediction of it based on previous samples is quantized and transmitted. This de-correlating (or whitening) process results in a transmitted signal of substantially reduced variance, compared to the original speech, and thus leads to a direct bit rate reduction for the same SNR performance (see also section 2.4.1.2).

The generalised structure of a differential coder is shown in figure 2.16, and reproduced for convenience in figure 3.1. The blocks labelled P and Q are the predictor and quantizer respectively, both of which may be either fixed or adaptive. This will be referred to as the DPCM configuration[37,41,45,46,55-64]. Note that the APC coder[80-82] of figure 2.17 in which an additional long-term pitch predictor is employed, also belongs to this general class of differential coders, since it also utilises the predictability property of speech. The function of the predictors in DPCM or APC coders is to produce as accurate as possible a prediction sequence {y(n)} of the incoming speech {x(n)}, based on previously decoded samples, such that the prediction

error sequence {e(n)} is minimised. The simplest DPCM coder arises when both the quantizer and the predictor are fixed, and P is a simple one-tap delay, with a constant scaling factor which is either 1 (perfect integration) or less than 1 (leaky integration). Generally however, for more efficient performance, either or both P and Q are designed to adapt to the input signal's characteristics. Such systems, referred to as adaptive differential PCM (ADPCM) will be of concern in this chapter. In particular, the predictor part of the coder will be examined in detail while the quantizer will be covered in chapter 5.

In the following sections, various known predictor algorithms are examined and their performance and limitations discussed. Then several novel predictor adaptations which seek to improve on the performance of standard methods are introduced. In later sections, the APC coder is considered, together with some pitch extraction methods. A simplified APC coder, designed to operate at 16 Kbps, is then described and evaluated. The one-word memory adaptive quantizer[49] is used in all computer simulations in this chapter.

3.2 FIXED PREDICTION

The predictor in DPCM is traditionally a transversal filter which can be represented in the z domain as,

$$P(z) = \sum_{k=1}^{p} a_k z^{-k} \qquad (3.1)$$

where p denotes the predictor order, and $\{a_k,\ k=1,2....p\}$ are the p coefficients of the filter (see figure 3.2). For fixed predictors, these coefficients are optimised using long-term statistics of the input signal and remain unchanged after that. Selection of the optimum

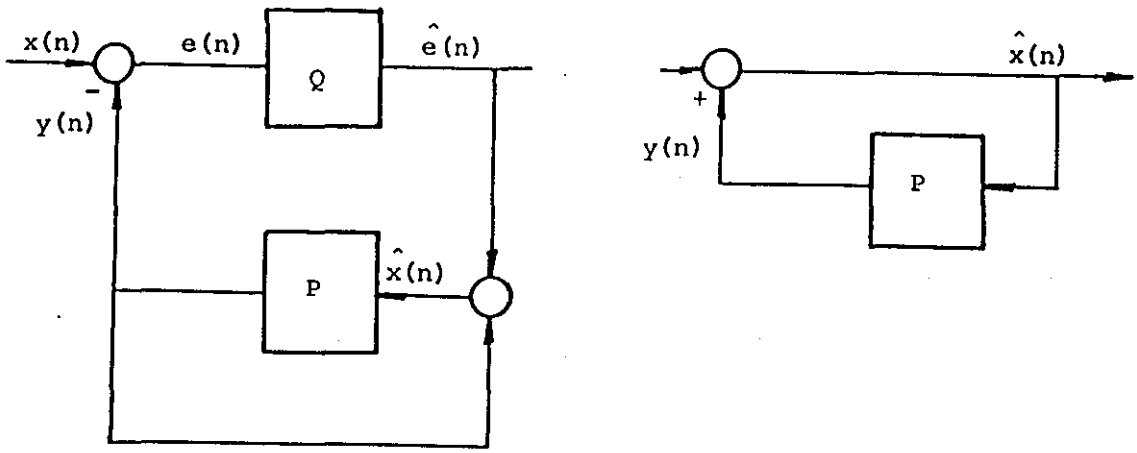


Fig. 3.1 Generalised Differential Coder Configuration

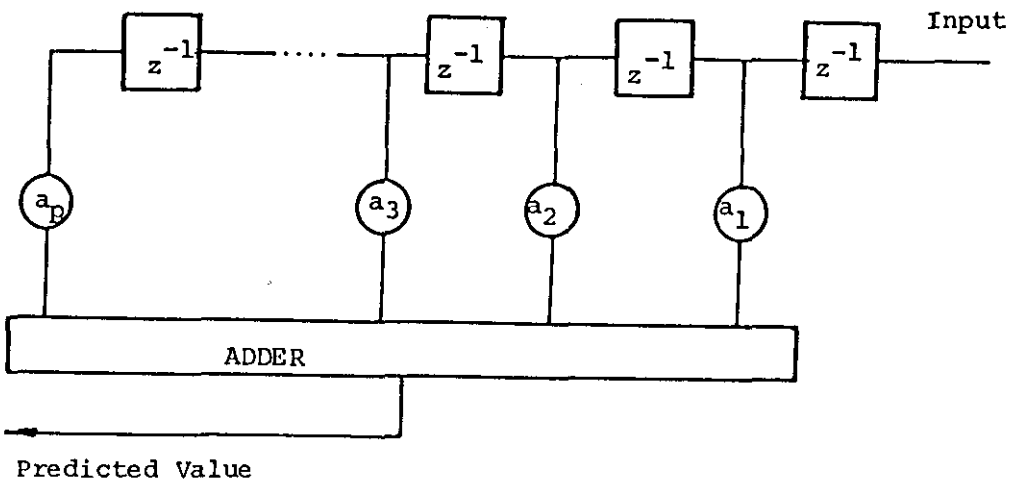


Fig. 3.2 Structure of a pth Order Linear Predictor

predictor coefficients is based on a minimum mean square error criterion. From figure 3.1, the error signal at the nth instant is the difference between the input sample x(n) and a prediction y(n) based on past reconstructed values of the input, i.e.,

$$e(n) = x(n) - y(n)$$

$$= x(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k) \tag{3.2}$$

Note that the feedback round the quantizer insures that the error in the reconstructed signal $\hat{x}(n)$ is precisely the quantization error of e(n) and not an accumulation of previous quantization errors. The SNR gain over PCM, G_{pcm} is given by the ratio of the input speech to that of the prediction error signal. To maximise G_{pcm} therefore, the predictor coefficients a_k are chosen to minimise the prediction error variance i.e.

$$\min_{a_k} \{ \langle e^2(n) \rangle = \langle [x(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k)]^2 \rangle \} \tag{3.3}$$

$$= \min_{a_k} \{ \langle [x(n) - \sum_{k=1}^{p} a_k x(n-k)]^2 \rangle + \langle q^2(n) \sum_{k=1}^{p} a_k^2 \rangle \} \tag{3.4}$$

where q(n) is the quantization noise, given by,

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

Equation (3.4) assumes that terms involving correlation between the input signal and the quantization noise are negligible. In addition, if the coder can be assumed to be good enough such that,

$$\langle q^2(n) \rangle \ll \langle x^2(n) \rangle \tag{3.6}$$

then the second term of (3.4) can also be neglected, leading to,

$$\underset{a_k}{\text{Min}} \quad \{ \langle [x(n) - \sum_{k=1}^{p} a_k x(n-k)]^2 \rangle \} \tag{3.7}$$

which is the classical Wiener filtering procedure in parameter estimation theory[218] (also encountered in the design of the LPC vocoder - section 2.3.7). Rewriting equation (3.7) in terms of expectations (letting σ_e^2 representing the variance of e(n)),

$$E[e^2(n)] = \sigma_e^2 = E[\{x(n) - \sum_{k=1}^{p} a_k x(n-k)\}^2] \tag{3.8}$$

Expanding the right hand side,

$$\sigma_e^2 = E[x^2(n)] - 2 \sum_{k=1}^{p} a_k E[x(n)x(n-k)] + \sum_{k=1}^{p} \sum_{\ell=1}^{p} a_k a_\ell E[x(n-k)x(n-\ell)] \tag{3.9}$$

In matrix notation, (3.9) becomes

$$\sigma_e^2 = \sigma_x^2 - 2A^T C + A^T R A \tag{3.10}$$

where σ_x^2 is the variance of the input signal and

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \qquad C = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(N) \end{bmatrix} \qquad R = \begin{bmatrix} \rho(0) & \rho(1) & \cdots & \rho(N-1) \\ \rho(1) & \rho(0) & \cdots & \rho(N-2) \\ \cdot & \cdot & & \\ \cdot & \cdot & & \\ \rho(N-1) & \cdots & & \rho(0) \end{bmatrix}$$

The elements of C and R are the values of the autocorrelation function of the input sequence i.e.

$$\rho(i-j) = E[x(i)x(j)] \tag{3.11}$$

The optimum set of predictor coefficients A_{opt} which minimises σ_e^2 is formed by equating the derivative of σ_e^2 with respect to A to zero, thus

$$\left.\frac{d\sigma_e^2}{dA}\right|_{A=A_{opt}} = 0 \qquad (3.12)$$

or

$$-2C + 2AR = 0 \qquad (3.13)$$

The solution is therefore,

$$A_{opt} = R^{-1}C \qquad (3.14)$$

Using equations (3.10) and (3.14), the minimum variance of the prediction error σ_e^2 can be formed as,

$$\sigma_e^2 = \sigma_x^2 - C^T R^{-1} C = \sigma_x^2 - A_{opt}^T C \qquad (3.15)$$

Note that the variance σ_e^2 of the error sequence is not constant or monotonically reduced as the order p of the predictor increases. This is because speech is not perfectly predictable from its past samples and so as p becomes large, σ_e^2(min) approaches a finite non-zero value. Noll [60] investigated the SNR gain G_{pcm}, of DPCM over PCM for various order predictors, and showed that G_{pcm} typically saturates for all practical purposes at p=2. This observation is shown in figure 3.3 for both low-pass filtered (0 - 3400 Hz) and band-pass filtered (300 - 3400 Hz) speech. Note the higher asymptotic G_{pcm} value for low-pass filtered speech. This is expected, as low-pass filtered speech has more low frequency energy and hence greater adjacent sample correlation. This implies greater possibility of redundancy removal by differential coding and thus a higher G_{pcm}.

McDonald[62] considered the performance of DPCM systems on voice signals and produced some useful data on the long-term autocorrelation values of speech sampled at 9.6 and 8 kHz. Using these values, the optimum coefficients for various order fixed predictors can be found using

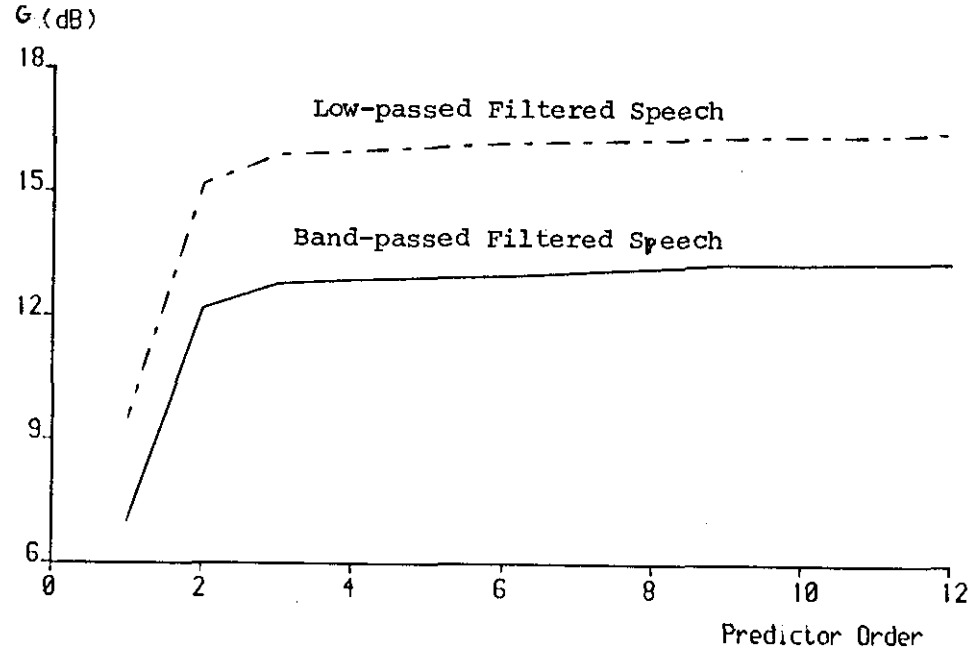


Fig. 3.3 Optimum SNR Gain, G vs Order of Predictor, p

(3.14). These autocorrelation values and the corresponding predictor coefficients are shown in table 3.1.

Table 3.1 Long-term Normalised Autocorrelation of 8 KHz Sampled Speech[62] and the Corresponding Optimum Predictor Coefficients.

Normalised Autocorrelation	
$\rho(1)$	0.8644
$\rho(2)$	0.5570
$\rho(3)$	0.2274
$\rho(4)$	-0.0297
$\rho(5)$	-0.1939
$\rho(6)$	-0.2788
$\rho(7)$	-0.3030
$\rho(8)$	-0.2823
$\rho(9)$	-0.2208
$\rho(10)$	-0.1330

Order	Optimum Predictor Coefficients							
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
1	.864							
2	1.515	-0.752						
3	1.748	-1.223	.310					
4	1.793	-1.401	.566	-0.147				
5	1.777	-1.338	.412	.051	-0.110			
6	1.776	-1.338	.415	.041	-0.097	-0.008		
7	1.776	-1.341	.416	.057	-0.148	.061	-0.039	
8	1.775	-1.340	.412	.058	-0.137	.024	.010	-0.027

3.3 ADAPTIVE PREDICTION

While fixed predictors are designed on the basis of long-term signal statistics, adaptive predictors seek to provide better prediction of the input speech by varying their coefficients according to the short-term local signal characteristics. Predictor adaptation may proceed either in a forward mode or a backward basis. Block adaptation is often associated with the forward mode while sequential adaptation occurs

mainly with backward prediction.

3.3.1 Forward Block Adaptive Prediction

In this method[33,68,80-82,112], the optimum predictor coefficients are calculated to minimise the forward prediction error, e^2, over a given range of x(n) which is chosen to cover between 8 to 32 ms of speech data, i.e. minimise,

$$e^2 = \sum_n \left[x(n) - \sum_{k=1}^{p} a_k x(n-k)\right]^2 \qquad (3.16)$$

This procedure is basically similar to the optimisation process encountered in DPCM (equation 3.7) except that in this case the minimisation is performed in the short-term over a very much smaller block of samples. Setting the derivative of e^2 with respect to the a_k's to zero yields the normal equations (see also equations 3.9-3.14):

$$\sum_{k=1}^{p} a_k \sum_n x(n-k)x(n-i) = \sum_n x(n)x(n-i) \qquad (3.17)$$

$$1 \leq i \leq p$$

According to the way the range of the minimisation procedure is specified, two cases arise from (3.17), leading to two methods of solution.

In the <u>autocorrelation</u> method[33,112], e^2 is assumed to be minimised over the infinite duration $-\infty < n < \infty$. Equation (3.17) then becomes,

$$\sum_{k=1}^{p} a_k R(i-k) = R(i) \qquad ; 1 \leq i \leq p \qquad (3.18)$$

where

$$R(i) = \sum_{n=-\infty}^{\infty} x(n)x(n-i) \tag{3.19}$$

is the autocorrelation function of the signal and is an even function with respect to i. The coefficients of R(i,k) form what is known as the autocorrelation matrix (hence the name autocorrelation method) which is a symmetrical Toeplitz matrix (a Toeplitz matrix is one in which all elements along each diagonal are equal). As the range of x(n) is only over a finite interval, a window w(n) can be applied to x(n), to obtain another signal x'(n) which is zero outside the interval concerned;

$$\begin{aligned} x'(n) &= w(n)x(n) \qquad ; 0 \leqslant n \leqslant N-1 \\ &= 0 \qquad\qquad\quad ; \text{otherwise} \end{aligned} \tag{3.20}$$

The autocorrelation function is then,

$$R(i) = \sum_{n=0}^{N-1-i} x'(n)x'(n-i) \qquad ; i \geqslant 0 \tag{3.21}$$

For the relatively short-term stationarity characteristics of speech signals, data windows such as the Hamming window or the Hanning window are appropriate, although for most ADPCM applications, the much simpler rectangular window is normally adequate.

Unlike the autocorrelation method, the <u>covariance</u> method[33,80-82] minimises the prediction error over a finite interval, say $0 \leqslant n \leqslant N-1$. Equation (3.17) becomes,

$$\sum_{k=1}^{p} a_k \Phi(k,i) = \Phi(0,i) \tag{3.22}$$

where

$$\Phi(i,k) = \sum_{n=0}^{N-1} x(n-i)x(n-k) \tag{3.23}$$

is the covariance of the signal x(n) in the given interval. Again, the

name of the method arises from the fact that the coefficients of $\Phi(k,i)$ form a covariance matrix. This matrix is also symmetrical, but unlike the autocorrelation matrix, it is not Toeplitz. Note from (3.23) that the values of the signal x(n) for the range $-i \leq n \leq N-1$ must be known, a total of N+p samples. The covariance method reduces to the autocorrelation method as the interval over which n varies goes to infinity.

Numerous solutions for the normal equations of (3.17) have been presented in the literature for both the autocorrelation and the covariance methods. Covariance matrices are symmetrical and in practice, usually positive definite. Thus (3.22) can be solved efficiently by the square root or Cholesky's decomposition method[219], which requires about half the computation ($p^2/6$) and storage ($p^2/2$) of the more general methods such as Gauss's elimination or Crout's reduction. The Toeplitz characteristics of autocorrelation matrices permit even further reduction in computation and storage. Levinson[220] derived an elegant recursive procedure for solving such Toeplitz matrix equations, and Durbin[221], exploiting the fact that the column vector R(i) in the right-hand-side of (3.18) comprises the same elements found in the autocorrelation matrix, produced a recursion twice as fast as Levinson's, requiring only p^2 operations and 2p storage locations - a substantial saving over the more general methods (see Appendix A, Durbin's recursion). In solving for the coefficients of a pth order predictor, Durbin's method also computes the solutions for all predictors of order less than p. An important by-product of this process is the set of reflection coefficients k_m, also known as PARCOR (PARtial CORrelation)[34,134] coefficients, which are related to

the a_k's by,

$$k_m = a_m^m \tag{3.24}$$

where a_i^j denotes the ith linear prediction coefficient for a jth order predictor. The k_m coefficients have the important property that if

$$|k_m| < 1 \quad ; i = 1,2...p \tag{3.25}$$

the linear prediction filter is guaranteed to be stable. For the autocorrelation method, the k_m's are always less than unity, so that stability is theoretically assured. The PARCOR coefficients also possess desirable quantization properties, and they are often used as transmission parameters in place of the a_k coefficients[134,135]. This is because the latter are extremely sensitive to errors - small perturbations can cause radical changes in the filter's frequency characteristics which may lead to instability. For the k_m's however, filter stability is assured by (3.25), while at the same time, the smaller dynamic range ($-1 \leq k_m \leq 1$ for all m) offers more accurate quantization. In practice, optimal quantization is obtained by transforming the k_m's into log area coefficients g_m, given by the relation,

$$g_m = \log \frac{1 + k_m}{1 - k_m} \tag{3.26}$$

and linearly quantizing them[134].

The choice between autocorrelation or covariance methods of solving for the optimum predictor coefficients in terms of output speech quality is not clear at the present, and no specific comparisons between the two methods for DPCM appears to have been documented. The computational efficiency of the autocorrelation method is an obvious advantage, although this would be more than offset by the substantial amount of

multiplications required if a data window (other than the rectangular window) is applied. The covariance method does not assume that all samples outside the analysis block are zero, and is possibly slightly more accurate as a result. In terms of SNR and subjective speech quality in DPCM systems however, indications are that differences between the two methods, if any, are negligible[19]. On the whole, the autocorrelation method appears to be more widely used in ADPCM because of the guaranteed stability of the filter produced [112,171]. The covariance method has been used for adaptive predictive coding (APC) systems[80-82].

It should be emphasised that the solution of the normal equations does not form the major computational load - most of the operations required in forward adaptive prediction systems involve the computation of the autocorrelation or covariance coefficients, which requires pN operations. This can dominate the computation time if $N >> p$ as is often the case[33].

In addition to direct methods of solving the normal equations of (3.17), various iterative solutions exist[33,222]. In these methods, one begins with an initial guess of the solution. This is then updated by adding a correction term, which is normally based on the gradient of some error criterion. Such iterative methods generally require more computation than the direct methods unless one begins with a good initial guess. They are useful however, for adaptations where the whole signal is not available at once, and the solution has to be updated based on every new observation. The amount of change is usually proportional to the difference between the new observation and the predicted value given the present solution. This is indeed the principle of operation of backward

sequential predictors to be discussed next.

3.3.2 Backward Sequential Adaptive Prediction

Most backward adaptive prediction algorithms used in ADPCM investigated to date, allow the predictor coefficients to evolve sequentially according to:

$$a_k(n+1) = a_k(n) + G(n)\hat{e}(n) \quad ; 1 \leqslant k \leqslant p \tag{3.27}$$

where $a_k(n)$ is the value of the kth predictor coefficient at the nth instant and G(n) is a gain term[65-67,69,72-75]. Equation (3.27) can be viewed as a sequential solution to the set of linear simultaneous equations or as an estimation theory-based algorithm for parameter estimation. The form of equation (3.27) arises from a sequential minimisation of the squared quantized prediction error $\hat{e}(n)$[69].

At the nth instant, the square of the quantized prediction error is given by,

$$\hat{e}^2(n) = \{ \hat{x}(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k) \}^2 \tag{3.28}$$

Differentiating with respect to a_j gives,

$$\frac{\partial \hat{e}^2(n)}{\partial a_j} = -2\hat{e}(n)\hat{x}(n-j) \quad ; 1 \leqslant j \leqslant p \tag{3.29}$$

To minimise $\hat{e}^2(n)$ with respect to the jth predictor coefficient, a_j must be corrected in a direction opposite to the gradient of the error in (3.29) giving,

$$a_k(n+1) = a_k(n) + g(n)\hat{x}(n-k)\hat{e}(k) \quad ; 1 \leqslant k \leqslant p \tag{3.30}$$

where g(n) is an appropriately optimised gain constant controlling the speed of predictor adaptation. The differences among backward adaptive predictors studied by various investigators evolve around the selection

of the gain term g(n). A simple form commonly used is[69,72,75]:

$$g(n) = \frac{G}{\gamma + \frac{1}{M}\sum_{j=1}^{M} \hat{x}^2(n-j)} \tag{3.31}$$

where γ and G are scalar constants determined experimentally. The second term in the denominator is the variance of the M most recently decoded samples and acts as a normalisation factor or automatic gain control, so that the coefficient adaptation is not input amplitude dependant, while the constant γ is included to prevent division by zero during silence. Frequently, M is set equal to p, the order of the predictor. Backward sequential predictors with the general form of (3.30) have been studied by Gibson[68,69,72,75], Moye[191], Cummiskey [76], Jones, Cohn and Melsa [73,75], Qureshi and Forney[67], for ADPCM coding and by Melsa and Goldberg[223] for APC systems. The update procedure given by (3.31) was investigated by Gibson[68,69] who referred to it as the stochastic approximation predictor (SAP). Cohn and Melsa[73] studied a slightly different formulation where the normalisation factor is an exponentially weighted function of previous decoded values,

$$a_k(n+1) = a_k(n) + \frac{G\,\hat{x}(n-k)\hat{e}(n)}{(1-\alpha)\sum_{j=0}^{\infty} \alpha^j \hat{x}^2(n-j) + \gamma} \tag{3.32}$$

Cummiskey[76] proposed a simpler adaptation based on the sign of the quantized error;

$$a_k(n+1) = a_k(n) + \frac{G\,\hat{x}(n-k)}{\sum_{j=1}^{p} |\hat{x}(n-j)|}\,\mathrm{sgn}(\hat{e}(n)) \tag{3.33}$$

This reduces hardware complexity with a possible loss in performance.

Evci, Xydeas and Steele[55,74,224] proposed a sequential gradient estimation predictor (SGEP) which is based on the general form of (3.30). In this scheme, the gain term g(n) is computed separately for each coefficient, using a sequential technique to estimate the gradient of the prediction error with respect to each a_k at each time instant. They reported improved results over SAP at the expense of greater complexity.

Gibson, Jones and Melsa also compared the performance of ADPCM systems using the SAP predictor with those using the Kalman predictor and found a slight advantage in the latter in terms of SNR[75]. The Kalman algorithm[19,75] is a more complicated estimation procedure which can be represented by the following equations in vector notation. The a_k coefficient vector A(n) is updated by the general form of (3.30),

$$A(n+1) = A(n) + K(n)\hat{e}(n) \tag{3.34}$$

where the gain vector K(n) is given as,

$$K(n+1) = \frac{V_a(n)\hat{X}(n-1)}{V_v + \hat{X}^T(n-1)V_a(n)\hat{X}(n-1)} \tag{3.35}$$

$V_a(n)$ is a pxp symmetric matrix defined as,

$$V_a(n) = \left[I - K(n-1)\hat{X}^T(n-2)\right] V_a(n-1) + V_w \tag{3.36}$$

$\hat{X}^T(n) = \{\hat{x}(n),\hat{x}(n-1),.....\hat{x}(n-p+1)\}$; V_w is a pxp symmetric matrix of w(n) where $w^T(n) = \{w_1(n),w_2(n),....w_p(n)\}$ is a vector of zero mean white noise terms, I is the pxp identity matrix and V_v is a scalar constant.

Obviously, the Kalman algorithm is a comparatively more complex adaptation procedure than the SAP predictor. However, improved performance should be obtained since a different gain is computed for each coefficient a_k, in contrast to SAP which uses a single gain G for all the coefficients (equation (3.31)). Note also that SAP is in fact a particular case of the Kalman algorithm. When $V_a(n) = I$ and $V_v = \gamma$, equation (3.35) becomes:

$$K(n) = \frac{\hat{X}(n-1)}{\gamma + \hat{X}^T(n-1)X(n-1)} \qquad (3.37)$$

which is essentially the SAP equation of (3.31).

One disadvantage associated with backward sequentially adaptive predictors in ADPCM is the risk of instability of the system - there is no guarantee that the predictor coefficients at any given instant constitute a stable filter. Errors in transmission and too rapid adaptation of the coefficients can often give rise to stability problems. To minimise this risk of filter instability, the change in the magnitude of the predictor coefficients at each time instant is frequently made very small. This however, means that predictor performance will be curtailed during periods of transition in the input speech (such as between silence and voiced speech), when quick adaptation is desirable. Stability problems in linear predictors may be avoided if the filter is configured in a lattice structure instead of the conventional transversal structure. Indeed, recent trends have indicated a shift in favour of lattice implementations in both ADPCM and LPC research[77-79,198-200].

The lattice filter, depicted in figure 3.4 arises directly from the computation of the least square error predictor coefficients by the autocorrelation method using Levinson's or Durbin's recursion (see section 3.3.1 and Appendix A). Stability of the filter is assured if the PARCOR or reflection coefficients, k_m are constrained to be of magnitude less than one. Note that increasing the predictor order is achieved by adding more lattice sections in cascade, without changing any of the previous sections[77]. This nesting property implies that a lattice predictor of order p contains implementations of all orders less than p, as noted in Durbin's recursion.

The following time-varying relations hold, from figure 3.4,

$$f_0(n) = b_0(n) = x(n) \tag{3.38}$$

$$f_m(n) = f_{m-1}(n) + k_m(n)b_{m-1}(n-1) \tag{3.39}$$

$$b_m(n) = k_m(n)f_{m-1}(n) + b_{m-1}(n-1) \tag{3.40}$$

where $f_m(n)$ and $b_m(n)$ are the mth stage forward and backward residuals respectively at time instant n, and the prediction residual e(n) is given by the pth stage forward residual $f_p(n)$.

In ADPCM applications, the lattice predictor is configured in a feedback loop with the quantizer as in figure 3.1; the input is the quantized signal sequence $\{\hat{x}(n)\}$ and the output is the linear prediction y(n). When used in conjunction with forward block processing schemes, the lattice implementation provides identical results with the transversal filter structure, and the reflection coefficients and predictor coefficients are related by (3.24). The attraction of the lattice however, is its efficiency when used in backward sequential adaptation, where the k_m's are updated at every sampling instant based on newly arrived information. Equations (3.39) and (3.40) show the explicit

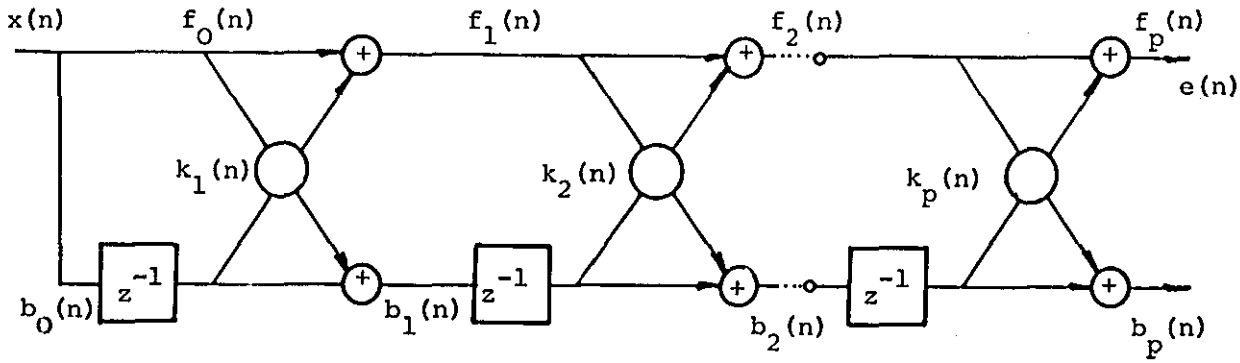


Fig. 3.4 pth order Lattice Filter

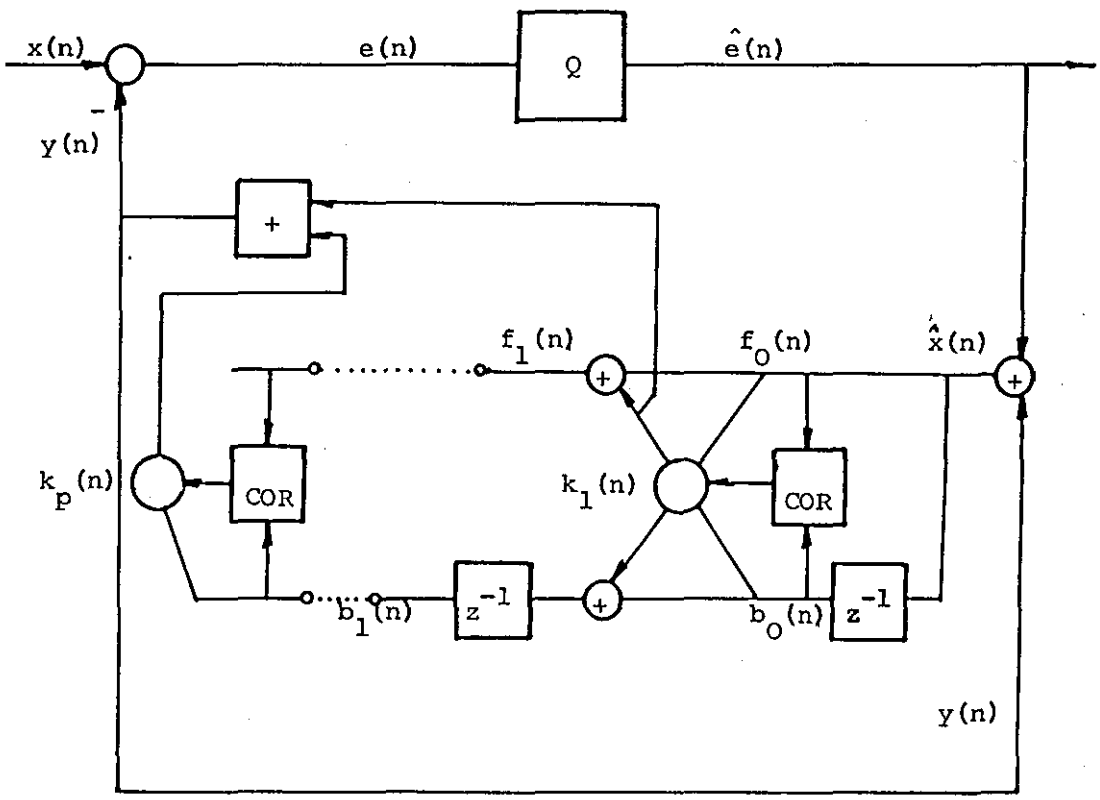


Fig. 3.5 ADPCM Using Lattice Predictor

dependance of k_m on time as $k_m(n)$. Figure 3.5 shows an ADPCM coder employing the sequentially adaptive lattice predictor. The adaptation of the predictor begins with the lowest reflection coefficient and propagates across the higher sections. $k_1(n)$ is first updated (by the box labelled COR) from information which includes $b_0(n)$ and $f_0(n)$ (the latest decoded sample). Then, the forward and backward residuals for the next stage, $f_1(n)$ and $b_1(n)$ are formed from the updated $k_1(n)$ according to (3.39) and (3.40). These residuals are then used for updating the next reflection coefficient k_2 and so on. This sequential propagation in the calculation of the k_m's contributes to the better convergence properties of the lattice predictor, compared to gradient-type algorithms. The output of the predictor is given by,

$$y(n) = \sum_{j=1}^{p} k_j(n) b_{j-1}(n) \qquad (3.41)$$

Various methods have been proposed for computing the k_m coefficients sequentially - these usually involve the minimisation of the variance of the forward residual or the backward residual or a combination of the two.

One typical method, employed by Makhoul[78,79] is based on minimising a weighted mean-square type of error of the form,

$$E(n) = \sum_{k=-\infty}^{n} w(n-k) e_m^2(k) \qquad (3.42)$$

where $e_m^2(k)$ is a function of the forward and backward residuals, given by,

$$e_m^2(k) = (1 - \gamma) f_m^2(k) + \gamma b_m^2(k) \qquad ; 0 \leqslant \gamma \leqslant 1 \qquad (3.43)$$

and $w(n)$ is a window that weights the residual energy into the past. The constant γ determines the mix between forward and backward

residuals. Minimising (3.42) with respect to $k_m(n)$ gives the update equation:

$$k_m(n+1) = - \frac{\sum_{j=-\infty}^{n} w(n-j) f_{m-1}(j) b_{m-1}(j-1)}{\sum_{j=-\infty}^{n} w(n-j) \left[\gamma f_{m-1}^2(j) + (1-\gamma) b_{m-1}^2(j-1)\right]}$$

$$= - \frac{C_m(n)}{D_m(n)} \qquad (3.44)$$

The window determines the rate at which past samples are progressively 'forgotten' and is typically a real pole filter of the form (in z-transform notation):

$$W(z) = \frac{1}{(1 - \beta z^{-1})^N} \quad ; \; 0 < \beta < 1 \qquad (3.45)$$

where the order N and the parameter β controls the decay characteristics. The effect of a higher N is to provide more weighting to the relatively short duration in the immediate past and 'forget' the more distant past quickly. β controls the general decay rate of the window, and for a given rate, N determines the relative weighting of the windowed samples. This is illustrated in figure 3.6.

For N=1, (3.45) can be expanded as an infinite series,

$$W(z) = 1 + \beta z^{-1} + \beta^2 z^{-2} + \beta^3 z^{-3} + \ldots . \qquad (3.46)$$

and the right-hand side of (3.44) becomes,

$$- \frac{f_{m-1}(n) b_{m-1}(n) + \beta f_{m-1}(n-1) b_{m-1}(n-2) + \beta^2 f_{m-1}(n-2) b_{m-1}(n-3) \ldots .}{\left[\gamma f_{m-1}^2(n) + (1-\gamma) b_{m-1}^2(n-1)\right] + \beta \left[\gamma f_{m-1}^2(n-1) + (1-\gamma) b_{m-1}^2(n-2)\right] + \beta^2 [\ldots .}$$

(3.47)

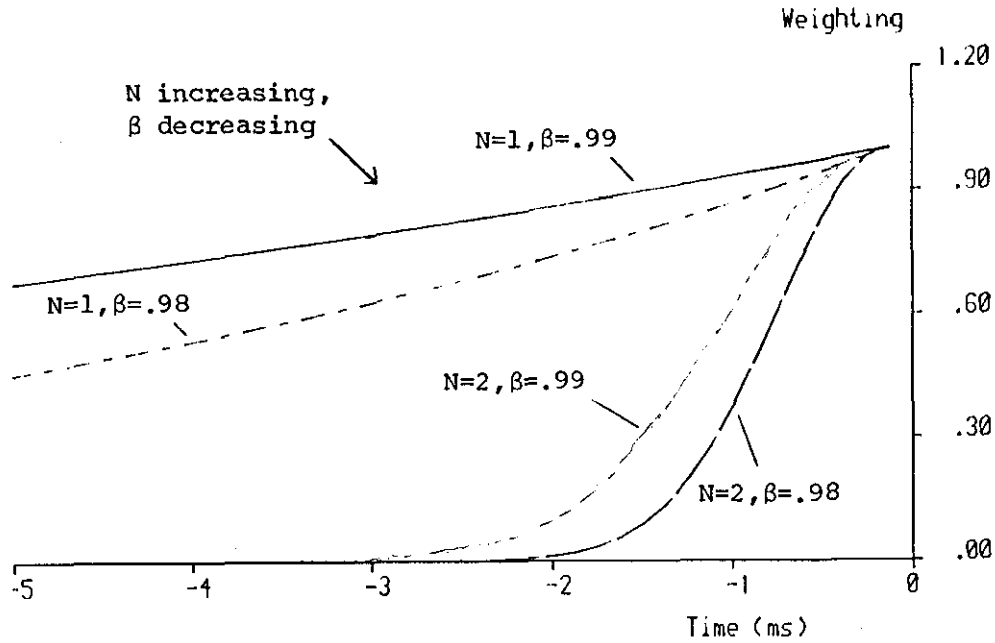


Fig. 3.6 Illustration of Window Characteristics Used in Lattice Predictors

i.e. the quantities $C_m(n)$ and $D_m(n)$ can be computed recursively as:

$$C_m(n) = \beta C_m(n-1) + f_{m-1}(n)b_{m-1}(n-1) \quad (3.48)$$

$$D_m(n) = \beta D_m(n-1) + [\gamma f_{m-1}^2(n) + (1-\gamma)b_{m-1}^2(n-1)] \quad (3.49)$$

For linear prediction applications, Makhoul recommended a 3-pole window with the optimum parameters $\gamma=1$ and β between 0.984 and 0.988[79].

Applications of the sequential lattice predictor to ADPCM have been investigated extensively by le Guyader and Gilliore[200]. They used a 1-pole window, with update equations given by,

$$C_m(n) = (1-\gamma)C_m(n-1) - 2\gamma f_m(n)b_m(n) \quad (3.50)$$

$$D_m(n) = (1-\gamma)D_m(n-1) + \gamma[f_m^2(n) + b_m^2(n)] \quad (3.51)$$

and,

$$k_{m+1}(n) = C_m(n)/D_m(n) \quad (3.52)$$

where γ is chosen to be a power of 2 (e.g. 2^{-6}) so that multiplications involving γ can be reduced to simple shift operations. They also proposed a simplified adaptation procedure which requires no multiplications or divisions. This so-called sign product method is given by the equations,

$$k_m(n) = \sin[(\pi/2)k_m'(n)] \quad (3.53)$$

where $k_m'(n)$ is derived recursively as,

$$k_{m+1}'(n+1) = (1-\gamma)k_{m+1}'(n) - \gamma\ \mathrm{sgn}\{f_m(n)\}.\mathrm{sgn}\{b_m(n)\} \quad (3.54)$$

Their main conclusions were that the lattice ADPCM coders out-perform the gradient adapted coders, and that the simple sign product adaptation is more robust to transmission errors. The latter observation is not surprising since the sign product method is a form of subdued prediction (see section 2.6.1(b)).

3.4 PROPOSED BACKWARD ADAPTIVE ALGORITHMS

The speech quality provided by fixed prediction ADPCM is acceptable for bit rates higher than 32 Kbps, where inaccuracies in prediction are compensated by sufficient fineness in quantization. As the bit rate is reduced however, the quality steadily deteriorates, and at 16 Kbps, the degradation in the recovered speech is clearly unacceptable. Adaptive prediction is able to provide at this bit rate, about 3 dB advantage in SNR and substantially improved perceptual quality. Generally, forward adaptive predictors are simpler in terms of signal processing requirements and more efficient in terms of error minimisation[68,225]. However, the need for side information and coding delay associated with forward adaptation can be a serious disadvantage. Backward adaptive predictors which do not have this drawback are therefore more attractive in many applications despite their greater complexity. In terms of performance, Gibson noted that there is little difference between the two[19,68]. Consequently, our investigation is focussed on the area of backward adaptive prediction. Several such adaptive predictor algorithms were developed for use in ADPCM and are considered in the following sections.

3.4.1 Sequential Adaptation

3.4.1.1 Modified SAP (SAPM)

Our starting point is the transversal predictor structure of figure 3.2, in which the predictor coefficients are updated according to the sequential adaptation algorithm governed by the general SAP equations of (3.30) and (3.31). As noted above, the conventional method of updating the predictor coefficients using the SAP algorithm has been to

apply (3.30) with the same prediction gain constant, G to all the coefficients. This assumes, rather without justification, that the a_k's are independent and that the optimum gain value is the same for each of them. Also, from (3.30), it is seen that the adaptation of the kth predictor coefficient at the nth instant, depends only on the latest quantized error sample $\hat{e}(n)$, and the decoded signal sample $\hat{x}(n-k)$, and not on the more recent decoded samples $\hat{x}(n)$, $\hat{x}(n-1)$.....$\hat{x}(n-k+1)$ which are available at the receiver. We investigated a slight modification to this procedure which attempts to provide for the inter-relatedness of the a_k coefficients (as is done in the Kalman algorithm) as well as to allow more recently decoded samples to affect the adaptation of higher coefficients in a similar manner to the lattice implementations. This modified SAP algorithm, denoted as SAPM, involves the following steps:-

(1) The first predictor coefficient a_1 is first updated in the conventional manner according to (3.30).

(2) This updated a_1 is then used to define a new error function using (3.28).

(3) The new error is differentiated with respect to a_2 to provide the gradient for the adaptation of a_2.

(4) Using the updated a_1 and a_2, another error function is formed, and this is differentiated with respect to a_3 to provide for the adaptation of a_3.

(5) This procedure continues until all coefficients are updated.

Consider the square of the quantized prediction error at the nth instant (from (3.28)),

$$e_1^2(n) = \{\hat{x}(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k)\}^2 \tag{3.55}$$

Differentiating with respect to a_1 gives,

$$\frac{\partial e_1^2(n)}{\partial a_1} = -2e_1(n)\hat{x}(n-1) \tag{3.56}$$

The first coefficient a_1 is updated as,

$$a_1(n+1) = a_1(n) + ge_1(n)\hat{x}(n-1) = a_1(n) + \gamma_1(n) \tag{3.57}$$

where g is of the form of g(n) given by (3.31). Using the updated a_1, a second error function $e_2(n)$ can be formed i.e.

$$e_2^2(n) = \{\hat{x}(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k) - \gamma_1(n)\hat{x}(n-1)\}^2 \tag{3.58}$$

Differentiating with respect to a_2 gives,

$$\frac{\partial e_2^2(n)}{\partial a_2} = -2e_2(n)\hat{x}(n-2) \tag{3.59}$$

and the second coefficient is similarly updated according to,

$$a_2(n+1) = a_2(n) + ge_2(n)\hat{x}(n-2) = a_2(n) + \gamma_2(n) \tag{3.60}$$

This process is continued for all the coefficients, giving,

$$\begin{bmatrix} a_1(n+1) \\ a_2(n+1) \\ \vdots \\ a_p(n+1) \end{bmatrix} = \begin{bmatrix} a_1(n) \\ a_2(n) \\ \vdots \\ a_p(n) \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_p \end{bmatrix} = \begin{bmatrix} a_1(n) \\ a_2(n) \\ \vdots \\ a_p(n) \end{bmatrix} + \begin{bmatrix} ge_1(n)\hat{x}(n-1) \\ ge_2(n)\hat{x}(n-2) \\ \vdots \\ ge_p(n)\hat{x}(n-p) \end{bmatrix} \tag{3.61}$$

By expressing the higher order errors and γ values in terms of g and e_1 (see Appendix B), the update equation can be shown to be,

$$\begin{bmatrix} a_1(n+1) \\ a_2(n+1) \\ \vdots \\ a_p(n+1) \end{bmatrix} = \begin{bmatrix} a_1(n) \\ a_2(n) \\ \vdots \\ a_p(n) \end{bmatrix} + g\hat{e}(n) \begin{bmatrix} \hat{x}(n-1) \\ [1-g\hat{x}^2(n-1)]\hat{x}(n-2) \\ \vdots \\ [1-g\hat{x}^2(n-1)][1-g\hat{x}^2(n-2)]..\hat{x}(n-p) \end{bmatrix} \tag{3.62}$$

Or more generally,

$$a_k(n+1) = a_k(n) + g\hat{e}(n)\hat{x}(n-k).F(k)$$

where,

$$F(k) = 1 \qquad ; k = 1$$

$$= \prod_{m=1}^{k-1}\left[1 - g\hat{x}^2(n-m)\right] \qquad ; 1 < k \leq p \qquad (3.63)$$

is a function of the p most recently decoded samples. Notice that the SAP algorithm is obtainable from (3.63) by setting F(k) = 1 for all k.

3.4.1.2 Adaptive Gain SAP (SAPA)

In the general SAP algorithm given by (3.30), the actual amount of change to the kth predictor coefficient at each time instant is governed by the term $g(n)\hat{e}(n)\hat{x}(n-k)$ in (3.30), where G (from (3.31)) effectively controls the adaptation rate since the normalisation provided by the denominator of (3.31) cancels out the magnitude variations due to $\hat{e}(n)\hat{x}(n-k)$ to some extent. Normally, G is optimised experimentally and kept fixed for a particular class of signals. Also, in order to minimise the risk of instability in the system, G is often kept rather small, to prevent too rapid changes occurring in the predictor coefficients. In reality however, the optimum rate of adaptation of the coefficients varies with time and according to the short-term signal characteristics. Slow variations are desirable during steady-state segments of voiced speech where the signal is locally stationary, while rapid adaptation is essential for efficient prediction in periods of transitions between silence or unvoiced sounds to voiced sounds or vice versa. A constant G is thus a sub-optimal compromise between these conflicting requirements. Better prediction could perhaps be achieved if G itself were made to adapt to the short-term signal characteristics.

We investigated a simple method of achieving some form of adaptation in G using ideas borrowed from adaptive delta modulation (section 2.4.1.5(b)). Specifically, the magnitude of G is permitted to increase or decrease depending on the direction of change in the predictor coefficients of past sampling instants. If the previous adaptations to a particular a_k coefficient were all in one direction (whether increasing or decreasing), then a more rapid change is desired, and G is multiplied by a factor α ($\alpha > 1$). Conversely, past adaptations of opposite polarity indicate probable local stationarity of the signal for which a smaller adaptation is preferable, and G is reduced appropriately by dividing by α. However, the variations of G must be necessarily bounded because of stability reasons.

Several variations on this theme were explored. The simplest version, denoted as SAPA-1 switches between 2 values of G i.e. $G\alpha$ and G/α depending on the polarity of predictor adaptation for the present and previous instants. If the same direction is indicated for both instants, the larger gain ($G\alpha$) is employed in the update equation - otherwise, the smaller gain is used. SAPA-2 extends this logic further, using 3 values of G i.e. $G\alpha$, G and G/α. $G\alpha$ is used when the directions of the previous 2 adaptations are the same as that indicated for the current adaptation; G/α is used when the 3 adaptations are of alternating polarity, and G is used in all other cases. Another variation, SAPA-3 allows G to assume values over a broader range, for quicker adaptation. Instead of using only 2 or 3 fixed values, G is permitted to vary freely between acceptable limits based on the same logic as above. In this case, the predictor gain G is a function of

time and is denoted by $G'(n)$. Table 3.2 shows the logic governing the variation of G for each of the SAPA schemes.

Table 3.2 Adaptive SAP (SAPA) - Variations in Predictor Gain G

	Scheme	Direction of Correction for time instant (n-2)	(n-1)	n	Predictor Gain for nth instant
1	SAPA-1		+	+	$G\alpha$
			−	−	
			+	−	G/α
			−	+	
2	SAPA-2	+	+	+	$G\alpha$
		−	−	−	
		+	−	+	G/α
		−	+	−	
		otherwise			G
3	SAPA-3	+	+	+	$G'(n-1)\alpha$
		−	−	−	
		+	−	+	$G'(n-1)/\alpha$
		−	+	−	
		otherwise			$G'(n-1)$

We note that the adaptive SAP algorithm described in the preceding section is similar in form to the 'fast converging stochastic gradient algorithm (FSAP)' mentioned by Farhang-Boroujeny and Turner[226] for non-speech applications. The adaptation of FSAP is given as:

$$a_k(n+1) = a_k(n) + 1/2\beta(1-q)G_f(n) + q(a_k(n)-a_k(n-1)) \qquad (3.64)$$

where $G_f(n)$ represents the conventional prediction gradient of the form $g(n)\hat{e}(n)\hat{x}(n-k)$, with $g(n)$ as given in (3.31), and q and β are constants determined experimentally. The similarity between SAPA and FSAP is most simply shown by considering the contribution of $q(a_k(n)-a_k(n-1))$ to the adaptation procedure for the kth coefficient. It is clear from (3.64) that a_k is updated by two quantities: (i) the term $1/2\ \beta\ (1-q)G_f(n)$ derived from the conventional SAP equation (3.30), and (ii) the previous magnitude of correction, $[a_k(n)-a_k(n-1)]$ weighted by the leakage factor

q (q < 1). If the previous and present adaptations are in the same direction, a_k is increased or decreased by the sum of the two quantities. On the other hand, if the present and previous directions of adaptation are not the same, then a_k is only changed by an amount equal to the difference of the two quantities, which would normally be very small. Thus FSAP may be considered as a more complex adaptive SAP algorithm with the variation of the a_k's proceeding rather more slowly compared to the SAPA schemes, particularly SAPA-3 (sums and differences as opposed to products and quotients).

3.4.1.3 Computer Simulation Results

Computer simulations were carried out to evaluate the performance of the various predictors considered. In each case, the predictor is configured in an ADPCM structure employing the 2-bit one-word memory quantizer. The parameters used in the various algorithms are those which provide the best average SNR performance over the input speech files used. These are determined experimentally to be:

(1) SAP (eqn. (3.30) & (3.31)) - G = 0.015, M = p (order of predictor) γ = 100.

(2) SAPM - as SAP.

(3) SAPA - as SAP, with α= 2.0
for SAPA-3, G'(n) is bounded by the limits 0.001 and 0.032.

(4) FSAP - G = 0.015, q = 0.95, β = 0.2.

In the simulations performed, a few instances of system instability were observed. These normally occur when a high predictor gain G is used.

The risk of instability, however remote, is clearly unacceptable, and in practical systems, appropriate preventive measures will be necessary. One relatively simple method (at least in computer simulation terms) of checking for possible instability in ADPCM systems employing linear predictors is to convert the a_k's into the corresponding reflection coefficents k_m and check that $|k_m| < 1$. This can be done using a backward recursion derived from Durbin's algorithm (Appendix A). An unstable filter can be made stable by reflecting the poles outside the unit circle inside, such that the magnitude of the frequency response remains the same[33]. For sequential predictors, a simpler alternative is to revert to the previous set of stable coefficients upon detection of possible instability. When this feature is incorporated in all the prediction algorithms examined, no more problems associated with instability were encountered for a wide range of G values.

The various predictors were evaluated based on their SNR performance and convergence rate[77], which is the time taken by the algorithm to respond to sudden changes in the signal statistics. This is related to the rate of adaptation of the predictor coefficients, and the following experiment using second order predictors was designed to observe this adaptation. Four blocks (each of 32 ms duration) of speech data were used for the experiment - these were obtained by taking two blocks of female speech (part of the utterance 'There') which contains a transition between silence and voiced speech, and reflecting these to obtain the third and fourth blocks to provide a similar transition from voiced speech to silence. These 4 blocks of experimental data are shown in figure 3.7. The predictor coefficients were all initialised to zero and adaptation was allowed to proceed. Figure 3.8 shows the adaptation

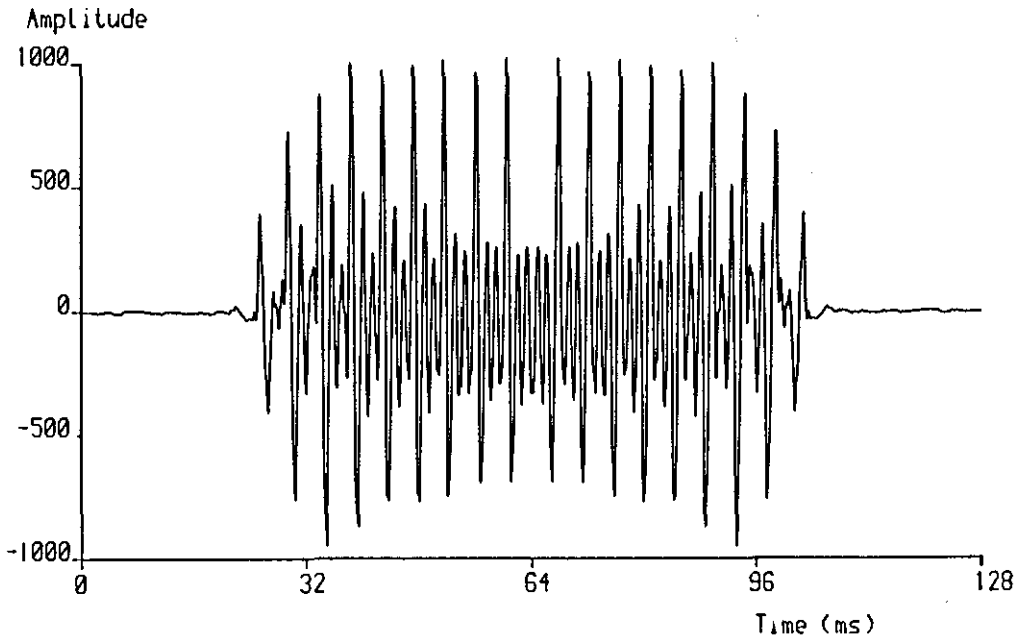


Fig. 3.7 Input Speech Data Used for Predictor Evaluation

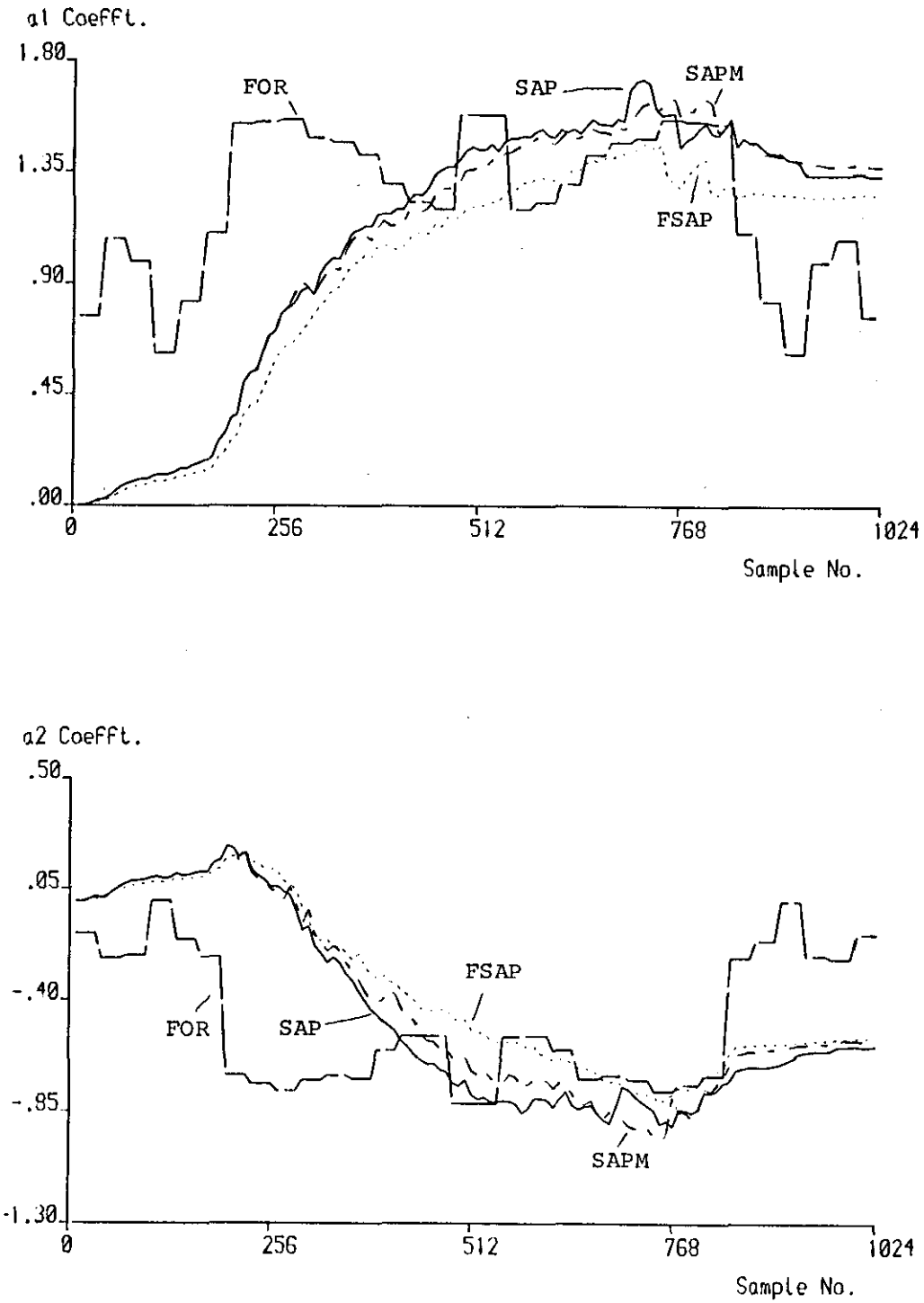
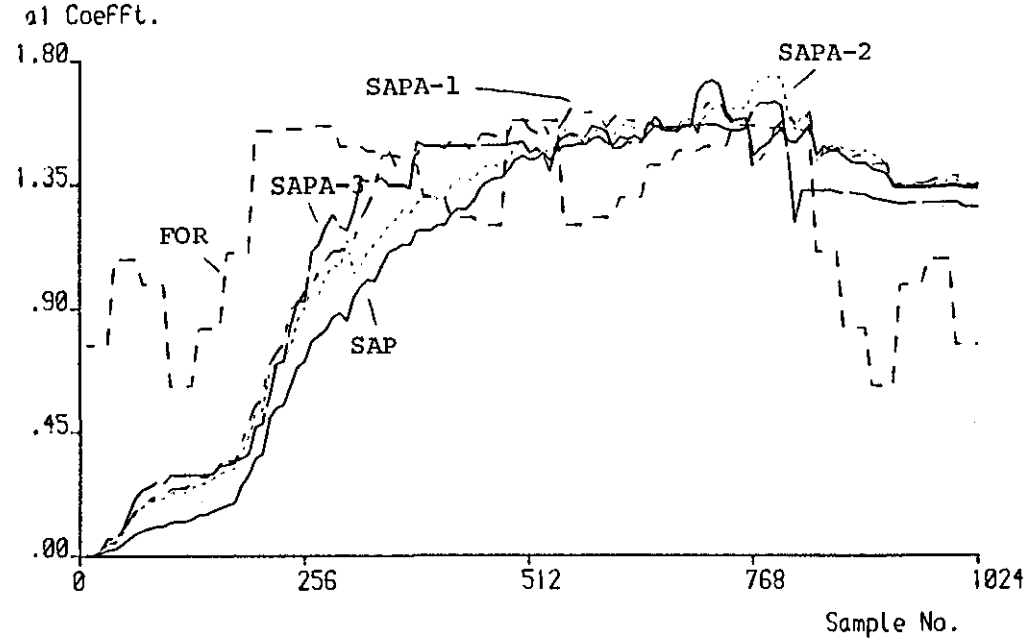
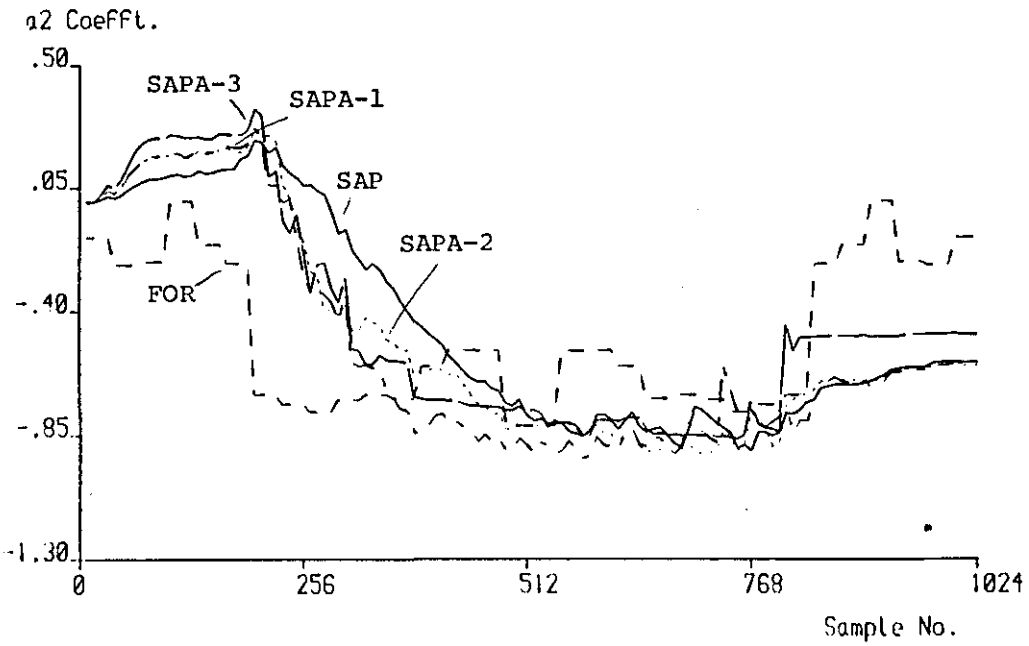


Fig. 3.8 Adaptation of Sequential Predictor Coefficients (Second Order)


a1 Coefft.
1.80
1.35
.90
.45
.00
SAPA-2
SAPA-1
SAPA-3
FOR
SAP
0
256
512
768
1024
Sample No.


a2 Coefft.
.50
.05
-.40
-.85
-1.30
SAPA-3
SAPA-1
SAP
SAPA-2
FOR
0
256
512
768
1024
Sample No.

of the coefficients a_1 and a_2 over time for the various predictors investigated. As the coefficients vary very slowly, only every eighth value was used to plot the adaptation trajectory. Also, as a comparison, the 'optimum' predictor coefficients for the same data calculated using forward block adaptation (section 3.3.1) with a blocksize of 32 samples (4 ms) were included. Although these coefficients are not necessarily the optimum for any particular sample, they do provide nonetheless, a useful indication of where the optimum values are.

It can be seen from the figures that the coefficients of the backward adaptive predictors seek to track the forward coefficients. The variation of the SAPM coefficients is very close to that of SAP, as would be expected for the low order predictor considered. However, the former coefficients appear to vary over a smaller amplitude range. The effect of an adaptive G for the SAPA algorithms is clearly evident from the figure - in all the variations of SAPA, the predictor coefficients approach the 'optimum' values from zero much more rapidly than SAP. Once the steady state is reached however, adaptation begins to slow down quite considerably.

Although this should not be considered as a proper evaluative test for the predictors concerned, it does nevertheless provide a useful indication of the speed of adaptation of the predictor coefficients to changing signal statistics. The experiment demonstrates the advantage of having an adaptive, rather than a fixed predictor gain G, and even the simple two-value switched G scheme of SAPA-1 is able to provide quicker adaptation. This faster adaptation is also reflected in the SNR values measured over the 4 blocks of data - all the SAPA schemes gave

better SNR performance than the SAP algorithm. This advantage however, is not as clearly apparent when the predictor coefficients adapt from values closer to the optimum, rather than from zero, as is done in the experiment. When the a_k's were initialised to the optimum fixed predictor coefficients given in table 3.1, it was found that no SNR advantage was discernible over the 4 blocks considered. Indeed, the long-term SNRs for all the sequential prediction algorithms investigated (2nd and 8th orders) do not show any significant differences (see Table 3.3). Examination of the individual block SNRs however, reveal that the SAPA variations tend to perform better during periods of transition in the signal, where comparatively large changes in the coefficients are desirable. Subjective listening tests conducted indicate a very slight preference in favour of SAPM, and particularly, SAPA-2 over conventional SAP.

Table 3.3 SNR Performance of ADPCM Coders with Backward Adaptive Prediction

2nd Order

Predictor Used	MALE SSNR	MALE TSNR	FEMALE SSNR	FEMALE TSNR	SISTER SSNR	SISTER TSNR
SAP	18.79	17.54	18.24	16.33	13.88	12.54
SAPM	18.62	17.21	18.51	16.68	13.92	12.55
SAPA-1	18.65	17.34	17.83	16.42	12.59	11.05
SAPA-2	18.57	17.08	18.35	16.28	13.27	10.88
SAPA-3	18.42	17.30	18.30	16.54	13.64	11.39
FSAP	18.52	17.74	18.40	16.54	13.90	13.05
BBA	18.53	17.68	17.94	16.39	15.50	16.53
LAT	18.91	17.46	18.79	16.78	15.15	15.64
LAT-SP	18.82	17.61	18.73	16.63	13.52	11.49
8th order						
SAP	18.66	17.48	19.08	17.40	12.57	9.44
SAPM	18.65	17.60	19.18	17.07	12.24	9.34
SAPA-1	18.68	17.16	18.78	16.74	12.91	10.67
SAPA-2	18.49	17.51	19.38	17.16	12.22	9.72
SAPA-3	17.70	16.62	18.52	16.57	13.17	11.65
FSAP	18.42	17.15	19.11	17.14	12.32	9.64
BBA	18.82	17.73	19.51	17.82	15.49	16.26
LAT	19.15	17.16	19.88	17.97	14.69	14.02
LAT-SP	19.06	17.68	19.08	16.93	13.13	10.67

3.4.2 Block Adaptation

Block adaptive predictors are normally associated with forward adaptation while sequential predictors frequently operate in a backward mode. However, this need not always be the case. Indeed, the advantages of backward sequential prediction (no side information or delay) may be combined with those of forward block adaptation (more

robustness, lower complexity) with some (inevitable) sacrifice of accuracy. Such a backward block adaptive (BBA) prediction technique is now proposed and described in the following.

3.4.2.1 Backward Block Adaptive (BBA) Predictor

The BBA predictor coefficients are computed in the same way as the forward predictor coefficients i.e. based on the short-term autocorrelation function calculated over a block of signal samples. The main difference is that, in the BBA case, no 'look-ahead' advantage is permitted (to avoid coding delay) - the predictor coefficients are optimised for a block of decoded samples and used to predict incoming speech samples which are not used in the optimisation process. The inherent assumption in this adaptation technique is that the statistics of speech signals do not vary drastically within short time segments of a few milli-second duration. Thus, the predictor coefficients optimised for a particular block of samples will be expected to provide good prediction when used for samples immediately outside the block considered. Also, since optimisation is performed using previously decoded samples which are available at the receiver, the need for transmitting side information does not arise.

Figure 3.9 illustrates how the BBA predictor adaptation proceeds[213, 215]. Assume that at time instant T, a new set of predictor coefficients is required, for the (T+1)th sample. These coefficients are computed from the autocorrelation function derived from the previous N decoded samples i.e. $\hat{x}(T)$, $\hat{x}(T-1)$, $\hat{x}(T-2)$....$\hat{x}(T-N+1)$, and are fixed and used for the next M incoming samples, x(T+1), x(T+2).....x(T+M). When x(T+M) has been processed, a new set of coefficients is required,

and this is computed in a similar manner, from the updated autocorrelation function, derived from the current block of most recently decoded N samples, i.e. $\hat{x}(T+M),\hat{x}(T+M-1)....\hat{x}(T+M-N+1)$. The autocorrelation analysis is thus performed on a sliding window over the N most recent decoded samples, with an overlap of (N-M) samples between adjacent blocks. The two main parameters involved in the BBA predictor are M and N, where N is the number of samples (blocksize) over which coefficient optimisation is performed, and (N-M) represents the amount of overlap between adjacent blocks. The optimum N would be similar to the forward adaptive case - typically spanning 8-32 ms of speech. The amount of overlap (N-M) can obviously affect the accuracy of prediction. A large overlap will presumably provide coefficients closer to the optimum, while at the same time increasing the computational load in the coder since the optimisation process has to be carried out more frequently over a given time period. Too little overlap on the other hand, could mean that changes in signal statistics may not be detected sufficiently quickly, resulting in a mismatch between the calculated and the optimum coefficients for certain blocks. Clearly, a reasonable compromise between complexity and efficiency has to be determined. The performance of the BBA predictor is examined in detail in the following section.

3.4.2.2 Computer Simulation Results

The first task is to determine how the amount of overlap between adjacent blocks of decoded samples affect the performance of the predictor used in an ADPCM system. For a blocksize N of 256 samples (32 ms) which was found to provide satisfactory results, the frequency of

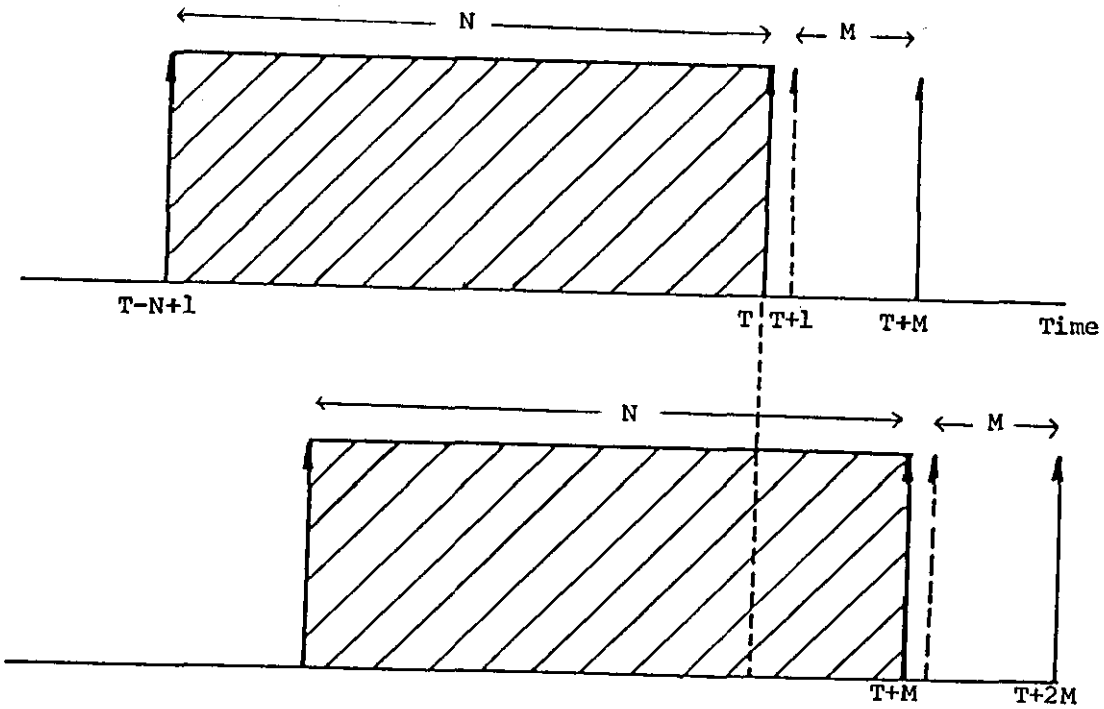


Fig. 3.9 Adaptation of Backward Block Adaptive (BBA) Predictor

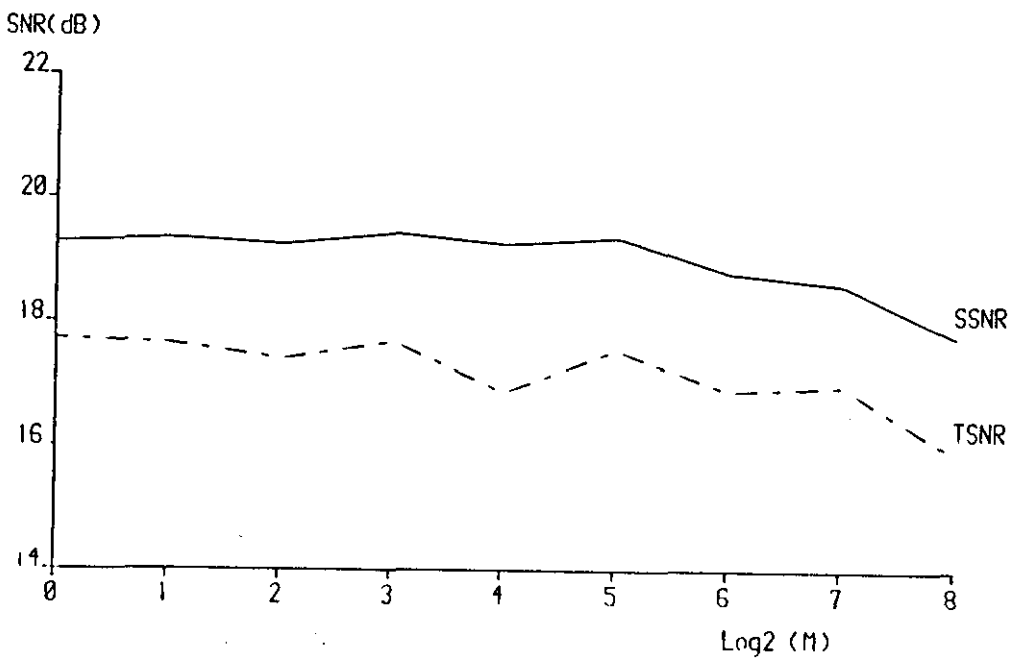


Fig. 3.10 SNR vs Update Frequency of ADPCM Coder Using BBA Predictor

update of the predictor coefficients for various order BBA predictors was varied, and their performance observed. Figure 3.10 shows the variations of the segmental and total SNRs (SSNR & TSNR) as a function of M for an 8th order predictor.

It can be seen that the SNRs remain relatively constant over a wide range of M values, indicating that excessively frequent updating of the predictor coefficients offers little advantage. A value of M=32 (which is a reasonable compromise between complexity and performance) was selected for use in subsequent simulations, although limited tests using larger M values provided similar results. The effect of windowing on the autocorrelation block to provide more weight to the more recently decoded samples was also investigated. This additional complexity did not contribute significantly however, to the performance of the coder and was therefore rejected.

The variation of the BBA predictor coefficients with time was next observed. Figure 3.11 shows the adaptation of the second order BBA predictor coefficients for the first 10 blocks of male speech compared to the forward predictor coefficients calculated for a blocksize of 32 and 256 samples (denoted as FOR32 and FOR256). It can be seen that the variations of the BBA coefficients are rather gradual and are bounded by the variations of the more rapidly changing FOR32 coefficients. Also, the BBA coefficients follow the general direction of the FOR256 coefficients but are consistently of lower magnitude. This could be due to the fact that while the forward predictor coefficients are optimised from the input signal, the computation of the BBA coefficients is based on decoded samples which contain quantization noise. Note however, that as long as performance is not appreciably impaired, lower magnitude

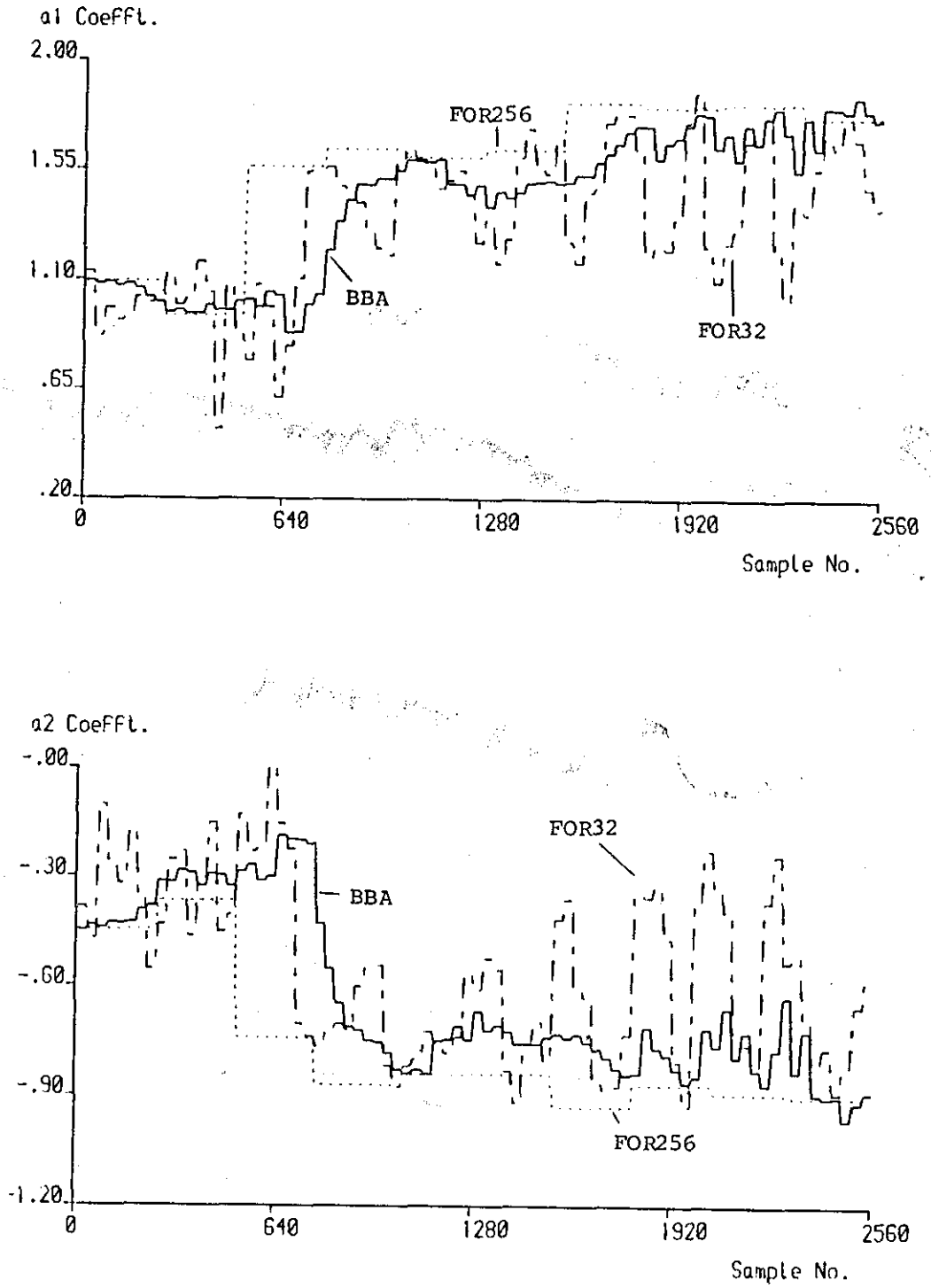


Fig. 3.11 Adaptation of BBA Predictor Coefficients

coefficients, which imply smaller power gains in the filter are desirable, because of the lessened risk of instability[81,82]. This is an advantage over SAP, (figure 3.8) whose coefficients are generally larger in magnitude than either block methods.

The BBA predictor was evaluated further by comparing its performance in ADPCM to the other algorithms considered in the preceding sections as well as to the adaptive lattice predictor (equations (3.50)-(3.54)). Table 3.3 summarises the SNR performance of the same ADPCM coder employing each of the different prediction algorithms. LAT denotes the lattice adaptation given by equations (3.50)-(3.52) and LAT-SP denotes the sign-product method of (3.53) & (3.54)[200]. These results were obtained from 60 blocks (about 2 s) of each data file. It can be seen that for the male and female sentences, the SNR values do not vary significantly among coders employing second order prediction. This is probably due to the fact that differences among the various algorithms are not fully manifested at such a low order of prediction. Perhaps a clearer indication of predictor efficiency is provided by the results for eighth order prediction. The figures show that the lattice and the BBA predictors are ahead of the rest by an average of half a dB. This advantage is also perceptible subjectively. Listening tests indicate that the recovered speech obtained from the ADPCM coder using the BBA and LAT predictors contains less high frequency distortion than those obtained using the other sequential predictors. This becomes clear from observation of the distribution of the output noise associated with each system. Figure 3.12(a) shows the output noise spectra obtained from male speech for ADPCM systems employing 8th order SAP, BBA and LAT predictors. The noise power of the coder employing SAP is generally

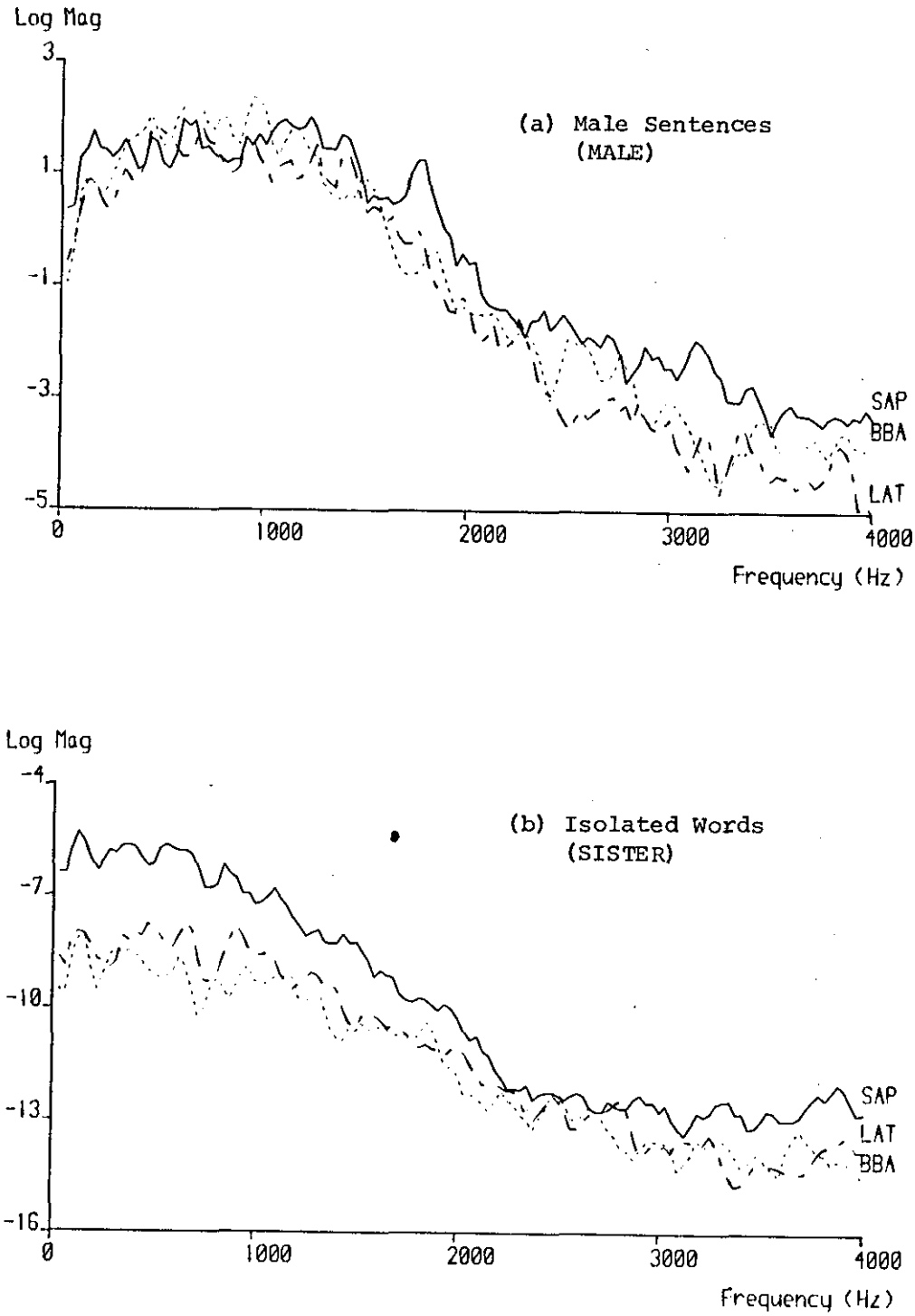


Fig. 3.12 Output Noise Spectra for SAP, BBA and Adaptive Lattice Predictors in ADPCM

greater than the other two systems across the whole of the spectrum and particularly so at the high frequency region, which accounts for the more audible 'hiss' in the recovered speech. The SNR results for the data file SISTER in Table 3.3 merit some comments. The actual figures are generally much lower than those obtained for the male and female sentences. Paradoxically also, the results for 8th order prediction are no better (or even worse) than the second order case. This observation can be explained by the atypical nature of the data file, which contains isolated words with substantially higher than normal unvoiced content. This lowers the average correlation of the waveform and hence, the predictability of the signal so that little, if any advantage is obtained by increasing the order of prediction. This lower predictability also accounts for the smaller SNR figures, and affects the adaptation of gradient-type algorithms such as SAP to quite an extent. What comes out beyond any doubt for this data file is the superior performance of the BBA predictor compared to the sequential predictors, including the lattice. Total SNR values of up to 6 dB advantage is recorded in some cases! This advantage is clearly seen from the output noise spectral plots shown in figure 3.12(b). The output noise power of the BBA system is considerably lower than both SAP and LAT.

3.4.3 Assessment of Prediction Algorithms

We now consider the merits of each of the prediction algorithms proposed in terms of such factors as performance, complexity and robustness.

3.4.3.1 Performance

The simulation results presented above indicate the potential of the BBA predictor used in ADPCM systems. In terms of both SNR and subjective preference, the BBA predictor was found to perform discernably better than the conventional gradient adaptations. The modification to the SAP algorithm (SAPM) to provide for the inter-relatedness of the predictor coefficients, although intuitively appealing, do not, unfortunately produce sufficient evidence of improved performance. Its SNR is no different from SAP and its recovered speech quality offers little, if any, advantage. Complexity-wise however, a significantly greater amount of signal processing is required.

The SAPA variations appear to be able to produce more rapid adaptation in the predictor during periods of transition in the signal, with little increase in complexity over SAP. This advantage again, is not apparent from the SNR values since transitions in the speech signal occur relatively infrequently in the sentences considered. Listening tests seem to indicate a slight preference for SAPA over SAP nevertheless. Predictors employing a combination of the SAPM and SAPA techniques were also investigated and found to offer a performance not far from either. It would appear, from these observations, that the term g(n) (from (3.31)) governing the change of the predictor coefficients do not in general constitute an overly critical factor as long as it is constrained to be within an appropriate range. Indeed, the SNR for ADPCM systems employing the SAP algorithm measured as a function of the predictor gain G was found to have a rather flat characteristic over a wide range of values, as shown in figure 3.13. In addition, the fact that Cummiskey reported satisfactory performance for his simplified

adaptation algorithm based only on the sign of the quantized residual (equation (3.34)) appears to support this observation. Recently, an experimental comparison carried out on various sequential prediction algorithms concluded that, "in the context of ADPCM, the extra computational burden associated with more complex adaptive linear prediction algorithms outweighs the accompanying improvement in system performance" [197]. It must be noted however, that the comparison is carried out at a particular transmission bit rate, and the criteria used are the mean-square prediction error and the average SNR. This does not take into account the subjective quality of the received speech, for which differences among dissimilar classes of predictors at a different bit rate may well be significant.

Indeed, a difference in subjective quality which might not be adequately reflected in SNR values certainly exist between the recovered speech produced in ADPCM coders employing the BBA predictor and those employing the gradient algorithms.

3.4.3.2 Complexity

The complexity of an algorithm is often considered in terms of its ease of implementation in hardware, and this might be influenced by such factors as the design and architecture of the particular hardware chip, which may be unrelated to the algorithm concerned. For simplicity however, we shall consider complexity in relation to the amount of signal processing operations required to perform a certain task. In this section in particular, we shall be concerned only with the multiplications and divisions involved, with a division being considered as equivalent to two multiplications.

To provide a basis for comparison between block and sequential methods of adaptation, the number of multiplications required for each algorithm over a block of N samples is calculated. The forward block adaptive (FBA) predictor is also included in the comparison, since it is closely related to the BBA predictor. Appendix C shows how the number of multiplications for each algorithm is obtained. Figure 3.14 illustrates graphically, the complexity of the algorithms concerned. Note however, that this complexity measure does not take into account the computation involved in ensuring filter stability, some form of which will be required for the sequential algorithms, in practice. The vertical axis represents the number of multiplications involved in the processing of a block of 256 samples. Table 3.4 provides expressions for the multiplication operations required for each algorithm, as a function of p, the predictor order and N. The expression for the BBA predictor is also dependant on the parameter M.

Table 3.4 Complexity of Adaptive Predictor Algorithms

Predictor	No. of Multiplications Required
FBA	(p+1)N + p(p+5)/2
BBA	N[(p+1) + p(p+3)/M]
SAP	(p+4)N
SAPM	(4p+1)N
SAPA	(p+4)N
FSAP	(2p+4)N
LAT	5pN
LAT-SP	(look-up table)

It can be seen from figure 3.14 that even without considering the computations required for ensuring predictor stability, the complexity

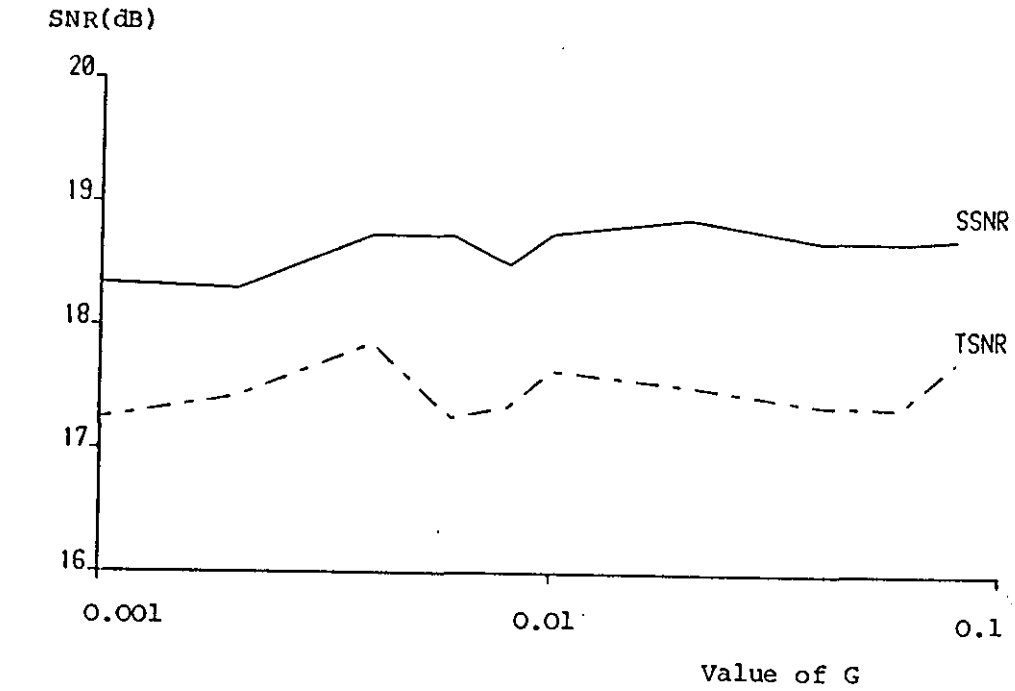


Fig. 3.13 SNR vs G for ADPCM Coder Using 2nd Order SAP Prediction

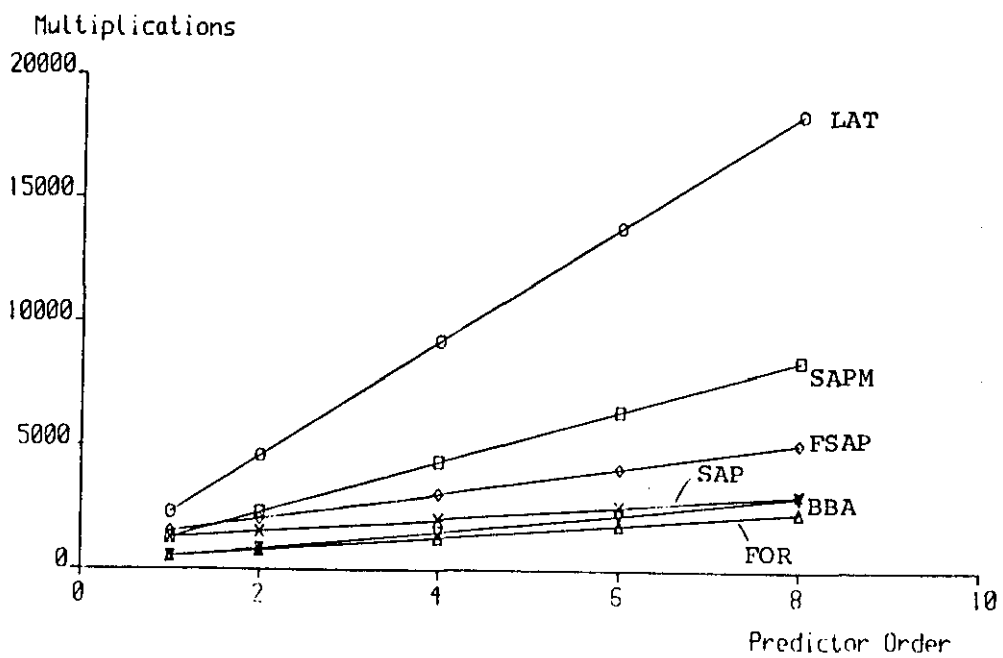


Fig. 3.14 Complexity of Adaptive Prediction Algorithms in Terms of the Number of Multiplications Required

of backward sequentially adaptive predictors, especially the lattice methods is still higher than the forward case. As noted in section 3.3.1, the bulk of the computational load in forward block adaptive prediction lies in the derivation of the autocorrelation function, as fast algorithms, such as Durbin's recursion exist for the solution of the resulting Toeplitz matrix equation. For this reason, the complexity of the BBA predictor is also lower than the sequential methods since its coefficients are essentially computed in a similar way as the forward predictor. Compared to the forward system, the BBA predictor requires, due to the overlap between blocks, an additional (N/M-1) computations for the predictor coefficients per block of N samples, as the autocorrelation calculations are the same for both cases. This is because in the derivation of the autocorrelation function, multiplications involving samples common to adjacent blocks, due to the overlap, need only be performed once, and the results stored for future use. With N=256, p=8, the BBA predictor requires about 13% and 27% more computation than the forward predictor, for M=64 and 32, respectively.

Although the BBA predictor coefficients are computed from a block of N decoded samples, it is not necessary to have these N values in memory at any sampling instant. Depending on the value of M, the autocorrelation function can be accumulated in partial sums (each of M values), and updating can proceed on a block (of M samples) basis. It is shown in Appendix D, that instead of storing all products used in the sequential computation of the autocorrelation function (which would require of the order of Np memory locations!), a much reduced buffer of size (p+1)N/M + p is sufficient. For N=256, M=32, p=8, the storage required is reduced quite substantially from 2048 to 80!

Thus, as far as complexity is concerned, it is quite apparent that the BBA predictor is superior to the sequential schemes.

3.4.3.3 Stabilty and Robustness

System stability is not a major problem in the BBA predictor because of the method by which the predictor coefficients are computed. Durbin's algorithm ensures the stability of the filter since the reflection coefficients, $\{k_m\}$ are always less than one[33]. Likewise, in the lattice predictor, stability can be guaranteed by constraining the sequentially calculated k_m's to be less than one at each stage.

For the sequential gradient adaptations, stability is not automatically assured and some measures might have to be incorporated to prevent instability occurring. This would mean additional complexity and expense.

A further advantage of a block method of predictor adaptation (as opposed to a sequential method), is the possibly better 'robustness' to transmission errors, owing to the averaging process involved in the computation of the predictor coefficients. In the BBA predictor adaptation, the coefficents do not change directly in response to erroneous samples (unlike the sequential methods), but are kept fixed for up to M sampling instants. Burst errors in particular, would have a far less detrimental effect on the BBA predictor than on typical sequential predictors. Because the latter adapts instantaneously to the received residual signal, a succession of errors in the magnitude of this received signal would most certainly cause a total collapse of the system.

3.5 DISCUSSION AND CONCLUSION

We have introduced and described in the preceding sections, several backward adaptive predictor algorithms based on a transversal filter structure, which are suitable for use in ADPCM systems. These were evaluated using computer simulation and compared to known techniques such as the stochastic approximation predictor (SAP) and the adaptive lattice.

We first attempted to improve on the SAP algorithm by modifying the general equation for the predictor adaptation. One method sought to provide some inter-relation between the individual predictor coefficients and to permit the adaptation of higher order coefficients to be affected by the magnitude of the most recent decoded sample. Another variation provides an adaptive predictor gain constant which varies according to the estimation of the input signal's statistics - taking on a large value during periods of signal transition (for faster adaptation) and switching over to a smaller value on detection of signal stationarity. Although there was evidence of improved performance in SNR during periods of signal transition in the latter scheme, overall SNR results were inconclusive. Subjective improvement over the conventional SAP was also slight. Further experimentation suggests that improvements over SAP, based on modifying the conventional equation is very limited, due to the relative insensitivity of predictor performance to changes in the adaptation equation.

This leads to a move away from the SAP algorithm to the development of a backward block adaptive prediction algorithm, which was found to out-perform the gradient methods when employed in ADPCM, and to compare

well with the adaptive lattice. Its superiority over the gradient adaptations is particularly significant for signals with a high unvoiced content, and hence lower correlation, for which an SNR advantage of up to 6 dB has been observed. More importantly, the improvement in performance is perceptible subjectively as a reduction in high freqeuncy noise in the recovered speech. In terms of algorithm complexity, the BBA predictor was also found to require substantially less computation for its predictor coefficients compared to the lattice and gradient methods. At the same time, the nature of adaptation of the BBA predictor promises greater robustness to transmission errors, and particularly burst errors since the coefficients do not respond to changes in single samples, but are optimised from a fairly large block of decoded samples.

We conclude that the BBA predictor offers considerable potential for use in ADPCM systems, providing a performance comparable to the adaptive lattice predictor, but with much lower complexity and possibly better robustness. Moreover, as will be seen in chapter 5, the BBA predictor structure permits backward noise shaping features to be conveniently incorporated into the ADPCM coder producing significant improvement in the subjective performance of the coder without incurring any penalty in terms of increased transmission rate or coding delay[213,215].

3.6 PITCH ADAPTIVE CODING SCHEMES

While ADPCM coders seek to remove the redundancy between adjacent speech samples by short-term prediction, a more sophisticated class of predictive coders attempts also to exploit the quasi-periodic nature of the speech wave to obtain more complete signal prediction. Probably the

best known pitch predictive scheme in recent times is the adaptive predictive coder (APC) described by Atal and Shroeder[80-82].

3.6.1 Adaptive Predictive Coding (APC)

The block diagram of the APC is shown in figure 3.15. Signal redundancy is removed in two stages: first by the conventional vocal tract predictor P_1, and then again by the pitch predictor P_2, which in its simplest form is a tap and delay adjustment given by,

$$P_2(z) = \beta z^{-M} \tag{3.64}$$

where M represents a relatively long delay (2-20 ms) usually corresponding to a pitch period, and β is a scaling factor. The order of the predictors may be interchanged, but recent studies suggest that the order as given in figure 3.15 is the better arrangement[82]. The combined predictor is then given by:

$$P(z) = P_1(z) + P_2(z)[\,1 - P_1(z)] \tag{3.65}$$

Notice from figure 3.15 that the quantizer is again inside both predictor loops, to ensure that no noise accumulation occurs in the receiver. In terms of its predictive properties however, the APC may be represented by the two-stage feed-forward structure shown in figure 3.16 [37]. The delay M, of the pitch predictor is chosen so that the correlation between speech samples which are M samples apart is highest. The parameter β is then obtained as[80]:

$$\beta = \frac{< x(n)x(n-M) >}{< x(n-M)^2 >} \tag{3.66}$$

where x(n) is the nth speech sample and <.> indicates the averaging over all the samples in a given time segment. It was found that more

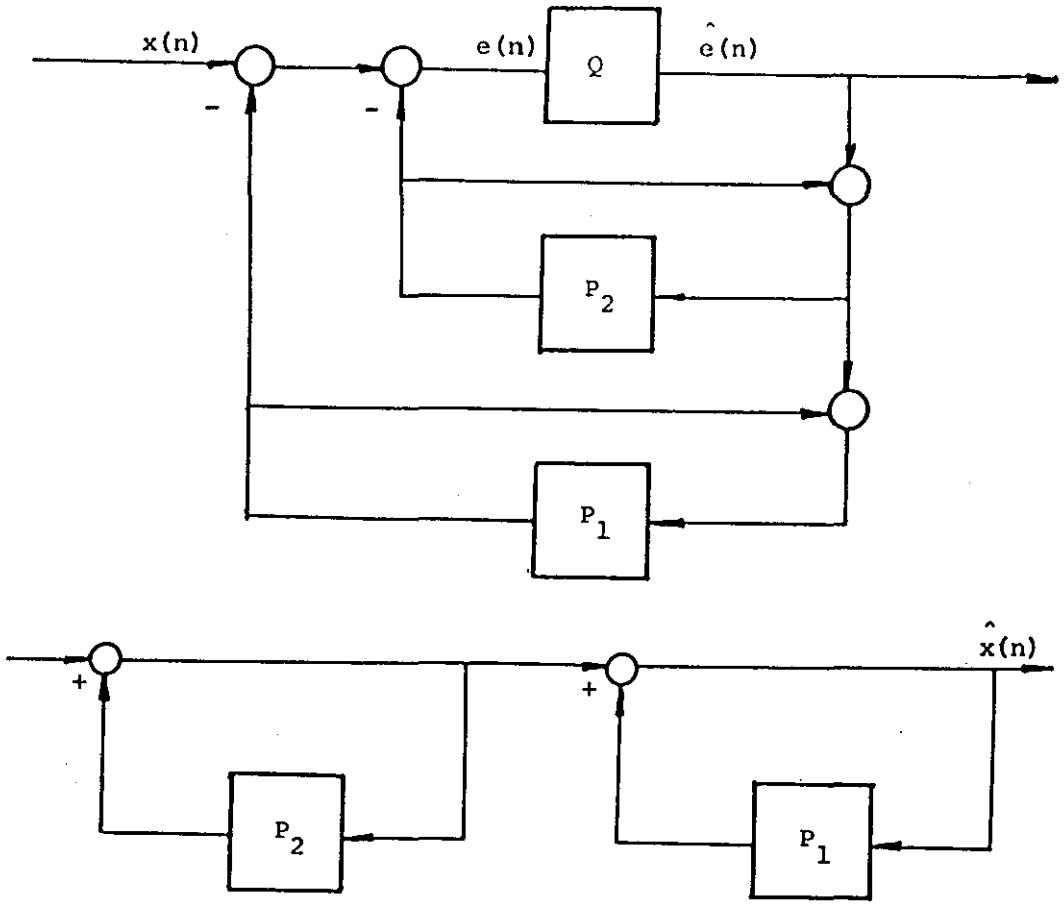


Fig. 3.15 APC Block Diagram

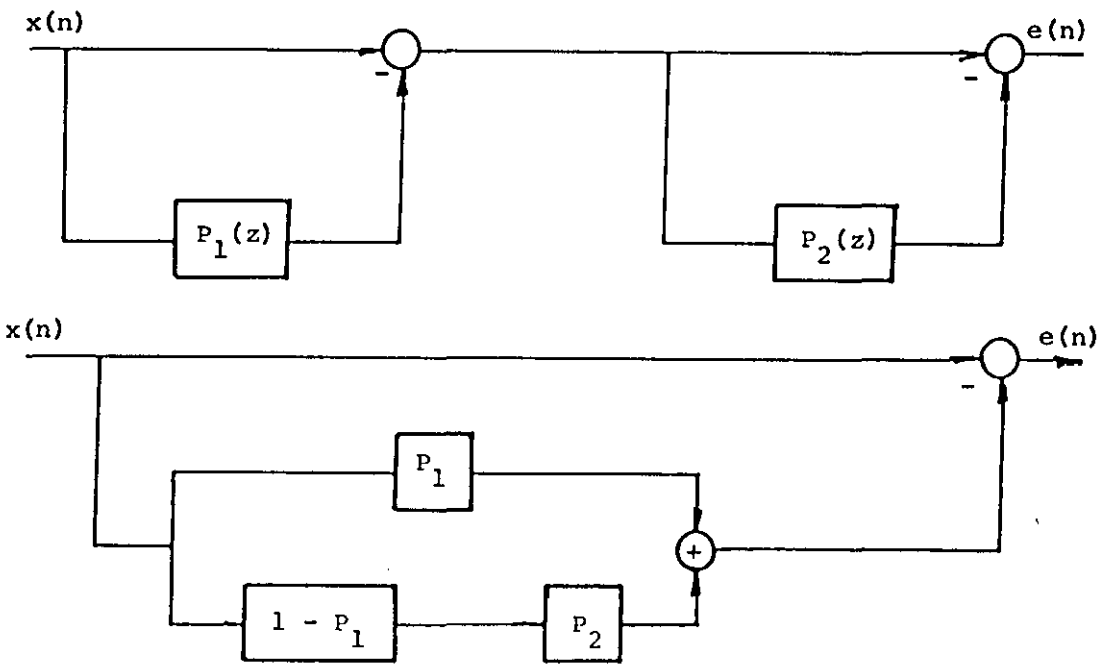


Fig. 3.16 Two-stage Prediction Representation of APC

accurate pitch prediction can be achieved if additional samples on both sides of M are also used in the prediction process[81], i.e.

$$P_2(z) = \beta_1 z^{-M+1} + \beta_2 z^{-M} + \beta_3 z^{-M-1} \quad (3.67)$$

Atal reported a 3 dB advantage in prediction gain for this 3-tap predictor over the one-tap case. With this highly complicated configuration, good quality speech at much less than 16 Kbps has been achieved.

In later work by Atal and Shroeder, the concept of noise shaping was applied to the APC coder to enhance the quality of the recovered speech with notable success[81]. Indeed, much of the current interest in noise shaping techniques has been largely generated as a result of their work on APC. More recently, in an attempt to push the bit rate further down without sacrificing speech quality, entropy coding was applied to the APC residual to ensure even more efficient utilisation of available bits [122]. This was soon followed by an exceedingly complicated split-band APC scheme in which, in addition to all the previous modifications, the input signal is first split into frequency sub-bands, before being preferentially encoded using APC[226].

3.6.2 Pitch Extraction Methods

The difference between ADPCM and APC is the use of an additional pitch predictor in the latter, which accounts for its more efficient (and complete) prediction. Accurate pitch prediction is thus instrumental to the performance of APC. However, although pitch extraction has been an important area of interest for a long time (particularly in the field of speech synthesis and vocoders), techniques suitable for use in time domain coders such as APC have not been too numerous. Most of these

evolve around some form of measurement of the signal correlation with varying degrees of complexity (which is often proportional to accuracy). Two pitch extraction methods relevant to the present context will be discussed in some detail in the following.

3.6.2.1 Average Magnitude Difference Function (AMDF) Pitch Detector

The average magnitude difference function (AMDF)[19,84,85] pitch detector avoids the heavy computational requirements associated with direct determination of the autocorrelation function by considering only the average difference between samples shifted by a constant amount within a block. Specifically, the AMDF is defined as:

$$AMDF(p) = \text{Average} \left| z(n) - z(n-p) \right| \qquad p = P_{min}, \ldots\ldots\ldots P_{max} \tag{3.68}$$

where z(n) is the nth sample of the block, which may contain the input speech signal or the prediction residual (after vocal tract prediction) or the quantized versions of either. p represents the amount of shift, and is bounded at each end by the minimum and maximum expected pitch period, P_{min} and P_{max}. The pair of samples z(n) and z(n-p) are such that both lie within a defined block of W samples, which is typically greater than the maximum pitch period (see figure 3.17). For each block of W samples, the AMDF is formed for all possible pairs of samples z(n) and z(n-p). The pitch period P is given as P_{est} if:

$$AMDF(P_{est}) < AMDF(p) \text{ for all } p \tag{3.69}$$

i.e. the pitch period P is the separation (in number of samples) which gives the minimum AMDF. However, due to the wide range of voiced pitch variation, pitch period multiples may sometimes be identified instead. An additional condition frequently applied is that, for waveform

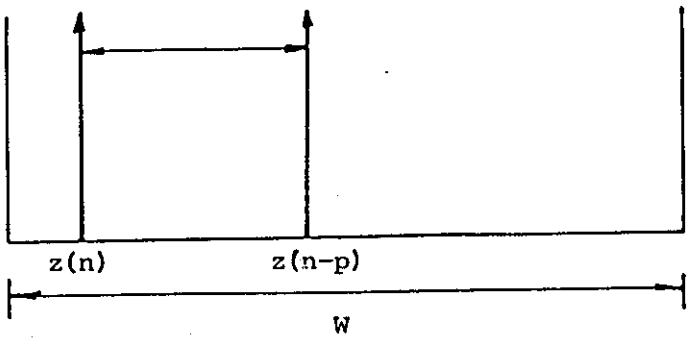


Fig. 3.17 Average Magnitude Difference Function (AMDF) Pitch Extraction

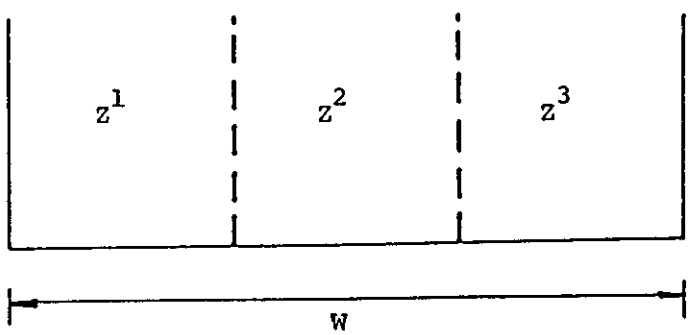


Fig. 3.18 Autocorrelation Method of Pitch Extraction

periodicity,

$$AMDF(P_{est}) < G_1 Average(|z(n)|) \text{ for all } n \quad (3.70)$$

where G_1 (typically 0.5) is a threshold that is used to hypothesise waveform periodicity with varying degrees of confidence. For highly periodic segments, $AMDF(P_{est}) << Average(|z(n)|)$, so the threshold G_1 can be used to ensure that pitch values are not assigned to non-periodic segments.

3.6.2.2 Autocorrelation Method of Pitch Detection

The method used by Atal[37,80] for determining the pitch period in a block of speech samples (3.66) involves obtaining all correlations between P_{min} and P_{max} and requires a huge amount of signal processing, which may be unacceptable for many applications. A simpler, but inevitably less accurate method based on the same principle, utilises only the sign information in the computation of the autocorrelation [19,84]. The autocorrelation function in this case is defined as:

$$C(p) = Average\ (sgn\ z(n).\ sgn\ z(n-p)) \quad (3.71)$$

Again, C(p) is calculated for all pairs of samples z(n) and z(n-p) so that both are within the block. The pitch period P is given as P_{est} if,

$$C(P_{est}) > C(p) \text{ for all } p \quad (3.72)$$

In addition, $C(P_{est})$ must usually also satisfy two further conditions,

$$C(P_{est}) > Z_{clip} \quad (3.73)$$

and,

$$C(P_{est}) > G_2 \quad (3.74)$$

where G_2 is typically 0.2, and Z_{clip} is a clipping threshold given by:

$$Z_{clip} = 0.64 \max(|z|^1_{max} , |z|^3_{max}) \quad (3.75)$$

$|z|^1_{max}$ *is the maximum z value in the first third of the block and*

$|z|^3_{max}$ is the maximum in the last third of the block (see figure 3.18). The inclusion of these two conditions have been quite effective in mitigating spurious peaks in the C(p) function, and provides for better accuracy in the prediction.

3.6.2.3 Other Pitch Extraction Techniques

Numerous other pitch extraction techniques in both time and frequency domain have been documented in the literature[19,31,84-88,228,229]. These include the cepstral method (widely used in vocoder applications) [31], the parallel processing method, techniques based on linear predictive coding (LPC) analysis, inverse filtering, etc. A thorough comparison of some of these pitch detectors is provided by Rabiner, et al[84]. Generally, block methods of pitch detection such as the AMDF and the autocorrelation methods described above are attractive because they are relatively simple, and also because the delay in the system is confined to only one block of samples. More recently, Miller[228] proposed a pitch detection algorithm in which pitch markers are identified by a series of elimination processes and logical tests. This method requires a large data file (up to 10000 samples) for successful operation. Although reliability was reported to be high, the need for an immense amount of storage (and the corresponding delay) renders it clearly unsuitable for most speech coding purposes.

3.7 PROPOSED PITCH ADAPTIVE DIFFERENTIAL CODER

While the efficiency of pitch adaptive coders such as the APC is without doubt, the complexity involved has limited its applicability to a great extent. Much of the complexity in APC is due to the pitch predictor,

which requires a large amount of signal processing operations for efficient performance. In order to produce a viable APC system for practical purposes, the complexity of the coder will need to be reduced quite substantially. In addition, forward adaptive prediction (for both predictors) as used by Atal will also be unacceptable for 16 Kbps transmission using constant rate coding because of the requirement of side information. In fact, Atal uses an 8th (or 10th) order vocal tract predictor, a 3-tap pitch predictor and a forward adaptive Gaussian quantizer (AQF) giving a sizeable side information overhead of 3-4 Kbps [81].

We decided to investigate the effects on speech quality of greatly simplifying the APC so that it is suitable for operation at 16 Kbps without (or with minimal) side information. Obviously, quality deterioration is to be expected - the object of the exercise is to determine the extent of the degradation and to compare the results of such a simplified pitch adaptive coder with other techniques of comparable complexity at the same bit rate.

3.7.1 System Description

The proposed simplified APC system follows the same general configuration of figure 3.15 . Two bits are assigned for quantizing the prediction residual e(n) using the backward adaptive Jayant's quantizer, to give a nominal transmission bit rate of 16 Kbps. Ideally, no side information should be required, to avoid any increase in bit rate. This means that all necessary adaptation should proceed in a backward mode, although this might not always be possible. The pitch information in particular, needs to be extracted from the input samples, as backward

adaptive methods of pitch detection are known to be highly unreliable [19,230]. For simplicity, the pitch predictor used is the single tap gain and delay arrangement of equation (3.64), which is defined by only two parameters, M and β . The few bits of additional information associated with M and β may be embedded in the transmitted data stream of the residual by 'stealing' bits from some of the samples, if it is essential that the total bit rate be strictly confined to 16 Kbps. For a typical pitch adaptation period of 32 ms, this side information represents an insignificant proportion of the total transmitted bits (about 10 out of 512) and may be easily accommodated without affecting the performance of the system. The vocal tract predictor was chosen to be either fixed or backward adaptive in view of the constraint on side information, and the pitch information is updated once every 32 ms (256 samples). Both the AMDF and the autocorrelation methods of pitch extraction were examined. P_{min} and P_{max} are set to 16 and 160, to cover a range of pitch frequencies between 50 and 500 Hz.

3.7.2 Pitch Synchronisation

One important feature of the APC coder not mentioned by Atal but noted by Xydeas[87,88], is the need to align samples in adjacent pitch periods correctly before removing the pitch redundancy in the signal. The residual signal after vocal tract prediction consists typically of a rapidly varying random signal with distinct 'spikes' at time intervals corresponding to the pitch period (see figure 3.19). These pitch periods usually vary in length gradually, increasing or decreasing by a few sampling instants at a time. The function of the pitch predictor P_2 is to obtain the difference between samples separated by the estimated

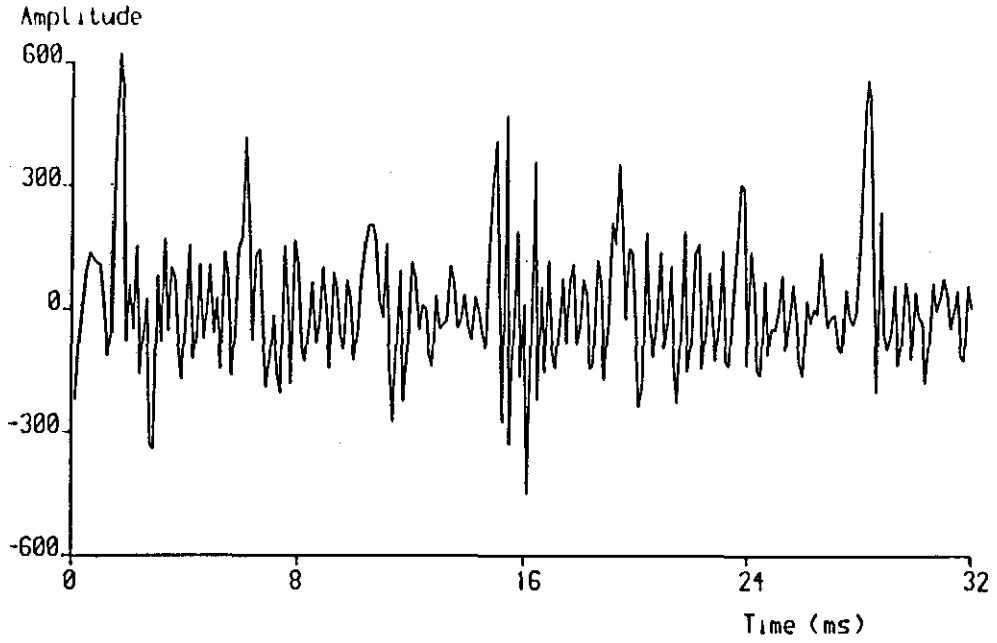


Fig. 3.19 Typical Speech Residual after Vocal Tract Prediction

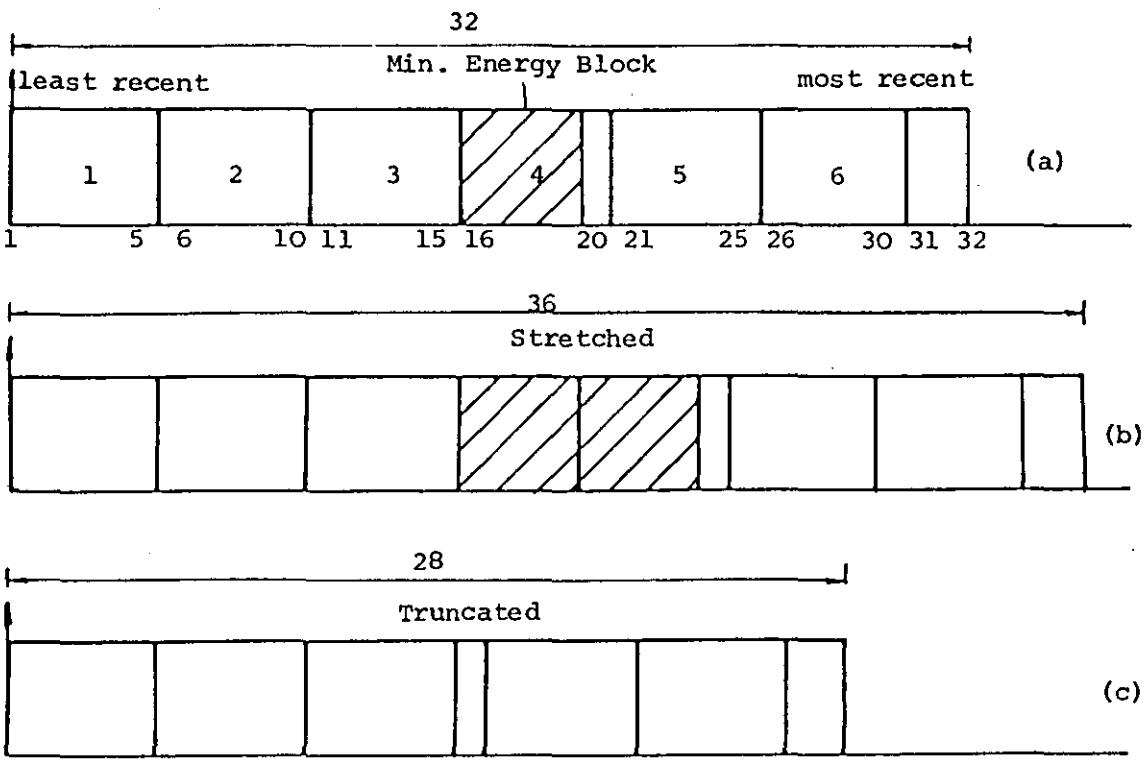


Fig. 3.20 Illustration of Filter Adaptation for Pitch Synchronisation in APC Scheme

(a) Original Filter Length
(b) Filter Stretched by Repeating Minimum Energy Region
(c) Filter Truncated by Discarding Minimum Energy Region

pitch period, with the aim of removing these spikes. To do this efficiently, it is important that adjacent pitch periods are correctly aligned before the subtraction is performed, i.e. the predictor buffer must be either 'stretched' or 'squeezed' in anticipation of the expected pitch of the incoming block of speech, to ensure that subtraction is performed between corresponding high amplitude samples. The pitch predictor is thus a linear filter whose length varies according to the estimated pitch period. With most pitch detection methods based on time domain measurements, there is a possibility that the estimates obtained are multiples of the actual pitch period. For the APC coder, such 'errors' are not likely to affect performance provided that the same estimate is obtained for the duration of the voiced utterance so that the pitch predictor filter is not subject to drastic changes in length. It is preferable in practice therefore, to include some form of check for such occurrences to ensure a smooth transition between pitch periods. The method employed to provide for pitch synchronous operation in the proposed APC coder will now be described.

The parameters M and β are obtained from a block of 256 samples of the input speech using either the AMDF or the autocorrelation method of pitch extraction. The length of the pitch predictor filter P is lengthened or shortened (or remains the same) according to the updated value of M. To ensure correct alignment of the pitch pulses, changes are only made to the stored samples in the predictor filter whose magnitudes are relatively small. The way in which this is done is best illustrated by an example. Assume that the present length of the pitch predictor filter is 32 taps and it is required to be changed to 36 taps for the next block. The 32 stored samples in the filter are divided

into sub-blocks of 5 samples each, as shown in figure 3.20(a), starting from the least recent sample. Sample(s) which are left over (such as samples 31 & 32 in this example) are excluded from consideration. The average energy of samples within each sub-block is calculated and the block with the lowest energy is identified. Assume that this is block number 4. The filter is then 'stretched' at this point by inserting the required number of samples (in this case 4) between the original samples 15 and 16. This is done by duplicating samples 16 to 19 at this location as shown in figure 3.20(b), thereby extending the filter length to 36 taps. Truncation of the filter is performed in a similar way. Consider, for instance the case when the filter is required to be shortened from 32 to 28 taps. In this case, the 4 samples of sub-block number 4 are simply removed and the least recent 15 samples shifted up. This ensures that the positions of corresponding pitch pulses are properly aligned, and that any necessary modification involves only the small magnitude segments. Alternatively, instead of duplicating samples in the former case, zeros could be inserted in the appropriate buffer locations.

Various other conditions have to be imposed on the pitch predictor to allow for deviations from normal operation. When no pitch periodicity is detected in the signal (as during unvoiced speech or pauses), β is set to zero, M is unchanged and the system becomes an ADPCM coder. A simple detection logic for identifying pitch period multiples is also included. Although multiples in pitch do not affect the coder performance in theory, it is important that changes in the filter length do not occur too drastically, such as from say, 33 to 66 or 99! Certain intuitive tests can be carried out to detect the occurrence of pitch

multiplicity. The following simple procedure was found to be adequate for the data files used in the investigation. Let the current pitch period be M_1 and the estimated pitch period for the next block be M_2. Pitch multiplicity is characterised by the observation that the quotient of the larger and the smaller pitch values is close to an integer greater than unity. Specifically, one of the pitch period is considered to be a multiple of a pitch value near (or equal to) the other if:

$$n - \varepsilon < \text{Max}\,(M_1/M_2,\ M_2/M_1) < n + \varepsilon \qquad (3.76)$$

where ε is a suitable small quantity (e.g. 0.2) and n is an integer greater than one. If M_2 is found to be the pitch multiple, then the required change ΔM, in the length of the filter is given by,

$$\Delta M = \text{NINT}\,(M_1 - M_2/n) \qquad (3.77)$$

where NINT(.) denotes the nearest integer. On the other hand, if M_1 is the pitch multiple, the corresponding change will be,

$$\Delta M = M_1 - nM_2 \qquad (3.78)$$

By this means, changes in the filter length is kept within reasonable limits. The pitch predictor adaptation logic is summarised in the flow chart of figure 3.21.

3.7.3 Computer Simulation Results

Prior to obtaining results for the complete APC system, initial tests were carried out to determine the most suitable pitch detection algorithm to be employed. The AMDF and the autocorrelation pitch extractors were both evaluated by comparing the pitch estimates each produces with values measured from the actual data. The optimum parameters for the two methods were experimentally determined to be:

$G_1 = 0.5$ for the AMDF algorithm,

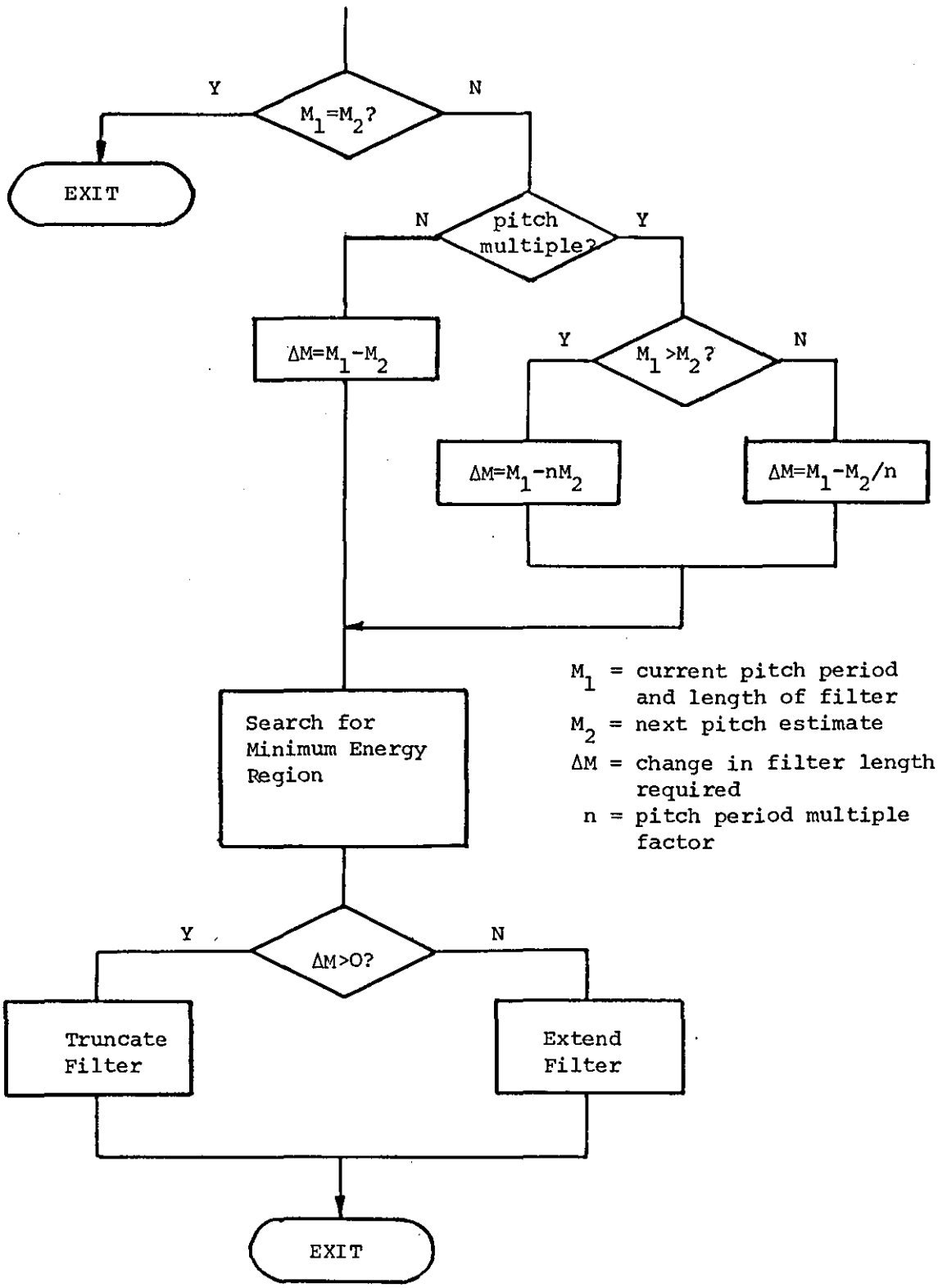


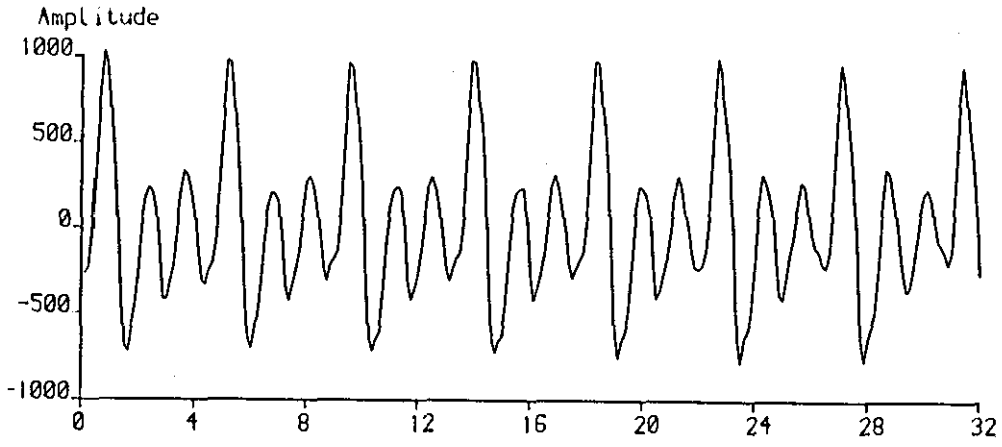
Fig. 3.21 Flow-chart for Pitch Predictor Adaptation (APC)

and $G_2 = 0.9$,

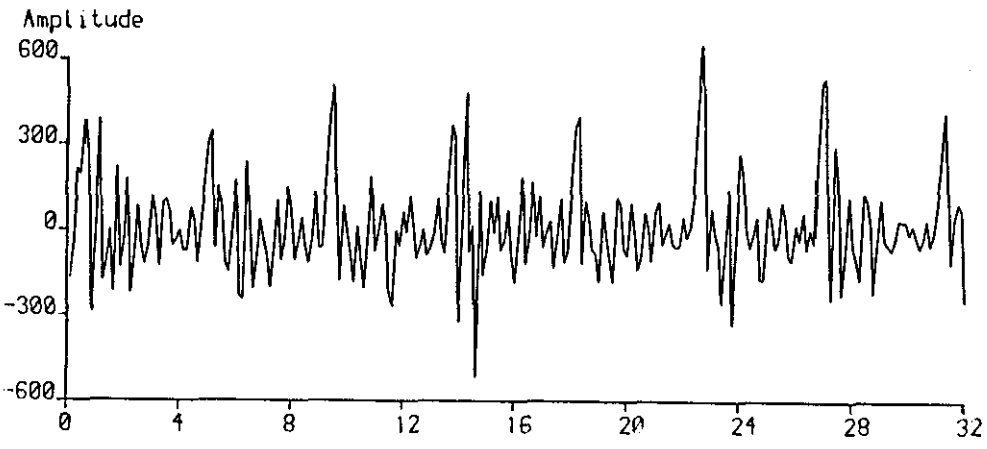
$Z_{clip} = 0.64$ for the autocorrelation method.

The recommended threshold of 0.2 for G_2[19] was found to be too low, and resulted in the failure of the algorithm to detect many blocks which are unquestionably periodic. Raising it to 0.9 provided considerably improved detection. The parameter β was found to vary between 0.6 and about 1.2 with values concentrated around 0.8-0.9, suggesting that it could possibly be kept fixed for simplicity.

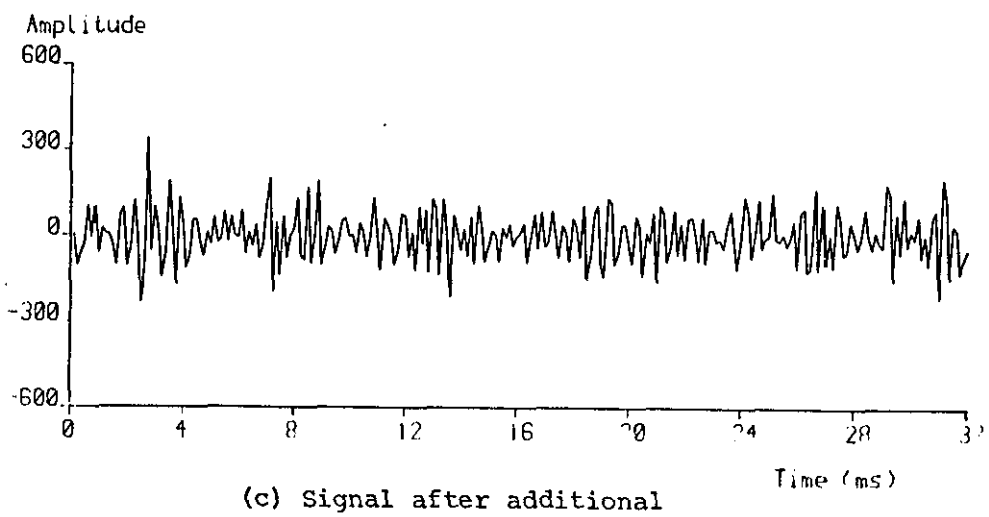
The APC scheme was first simulated using a fixed first order vocal tract predictor. Figure 3.22 shows the signals corresponding to about 100 ms of female (voiced) speech after each stage of prediction. The pitch periodicity is clearly evident in the residual signal after some adjacent sample redundancy had been removed by the vocal tract predictor (figure 3.22(b)). These pitch pulses were largely removed in the next stage by the pitch predictor (figure 3.22(c)). Figure 3.23 illustrates the gain in segmental SNR due to pitch prediction, for both male and female speech, over 60 blocks (2 s) of the data. The considerable improvement due to the more complete prediction of APC over simple ADPCM is apparent in the figure, and this advantage appears to be greater for female speech. The latter observation is not surprising as female speech waveforms are generally more periodic and better structured than male speech, and are therefore more suited for pitch adaptation. Indeed for the same utterance, a larger number of blocks in the female speech was classified as periodic. This is also reflected in the average SNR values obtained - the inclusion of pitch prediction provided a 4 dB advantage for the female speech compared with only 1 dB for male speech.



(a) Original signal

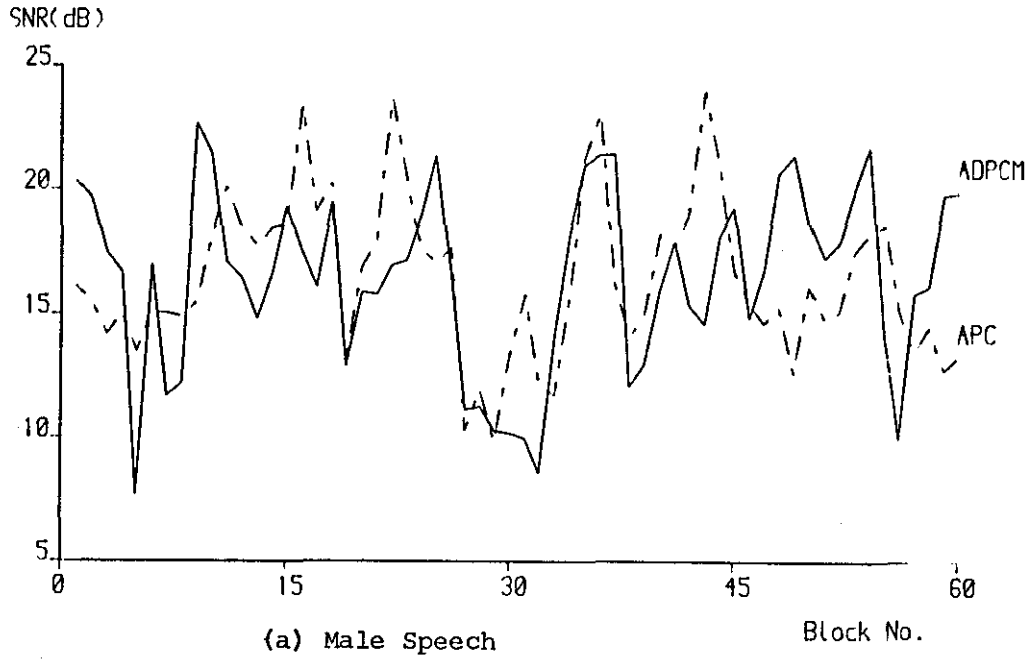


(b) Signal after vocal tract Prediction

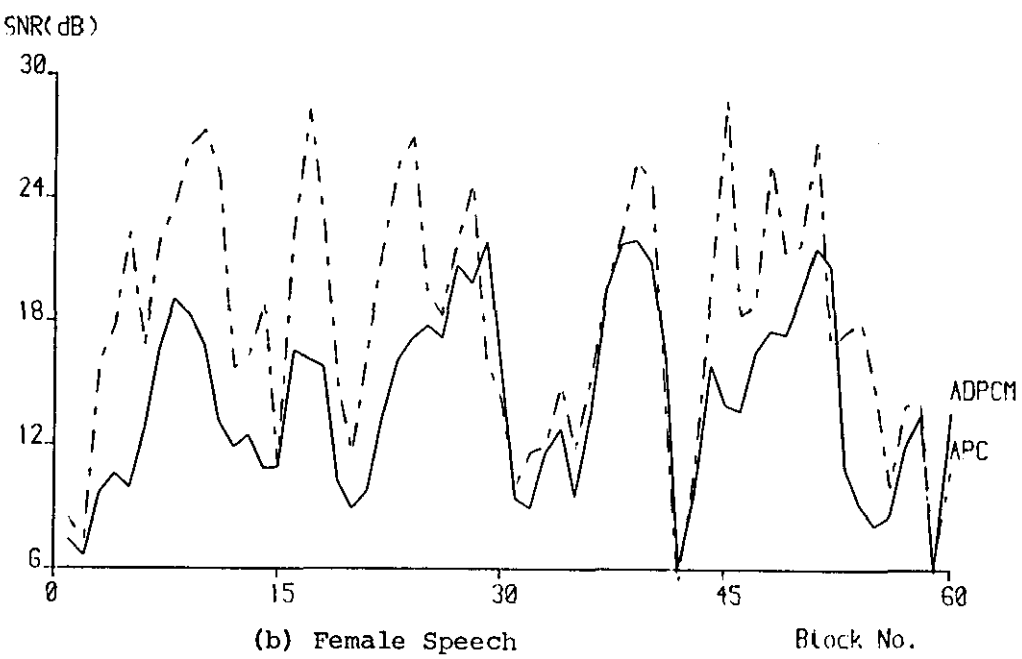


(c) Signal after additional pitch prediction

Fig. 3.22 Illustration of Speech Waveform after each Stage of Prediction



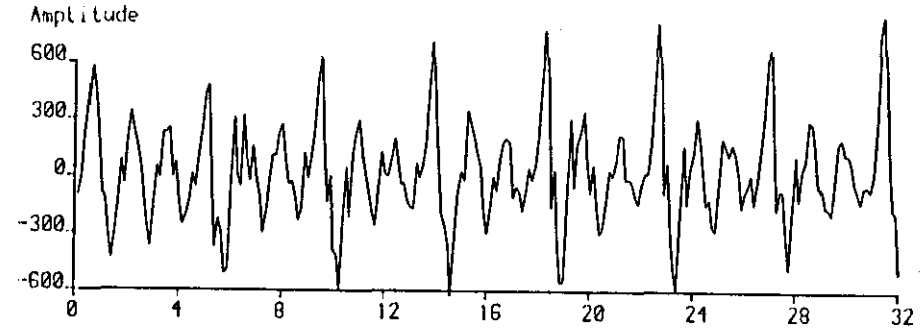
(a) Male Speech



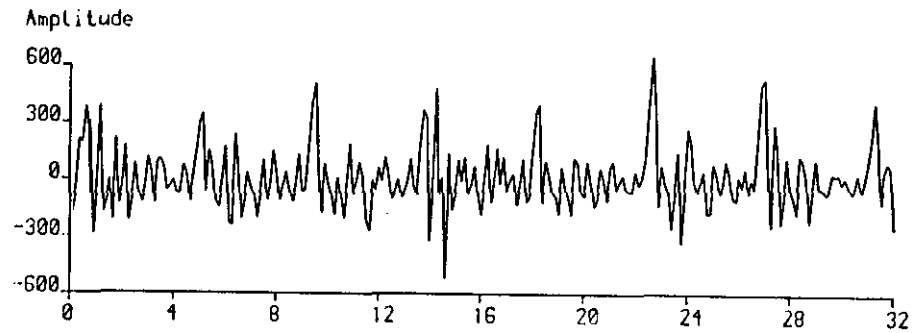
(b) Female Speech

Fig. 3.23 Segmental SNR of APC and ADPCM using First Order Predictor

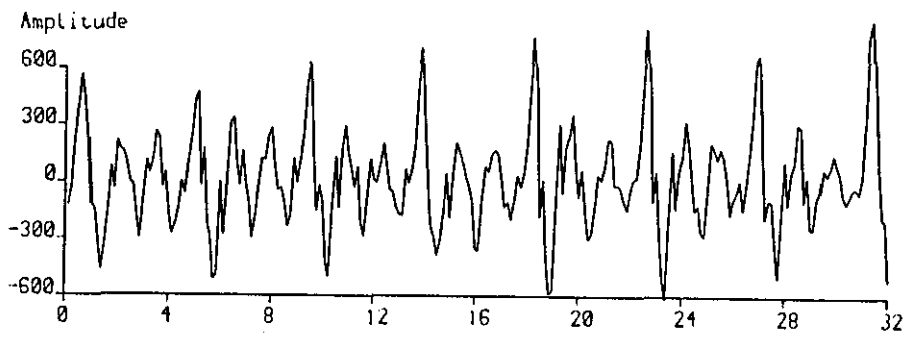
The performance of the APC coder with different vocal tract predictors was next investigated to see if the same advantage provided by the pitch predictor could be maintained when higher order (and by implication more efficient) vocal tract predictors are used. Figure 3.24 compares the residual signal after each stage of prediction using 1st and 2nd order fixed and 1st and 2nd order forward adaptive predictors. The effect of employing a higher order predictor can be clearly seen in the residual after the first stage of prediction. Because of the better decorrelating ability of 2nd order prediction, the resulting residual signal is reduced in magnitude by a greater extent and contains a significantly greater proportion of high frequency components. For the 2nd order fixed predictor, the periodic pulses are still retained, and these are quite successfully removed in the subsequent pitch prediction process, although the final residual appears to be no better (in fact, slightly worse) than when a 1st order predictor was used. The coefficients of the forward adaptive predictors are optimised from the short-term signal correlation and they are therefore able to remove a greater amount of redundancy from the input signal compared to the fixed case. However, this more efficient decorrelating process appears to produce a more random residual whose pitch structure is not as clearly defined in certain places (see figure 3.24(a)(iii)). As a result, the ability of the pitch predictor to effect further signal compression is affected to a degree so that the final residual signal produced (figure 3.24(b)(iii)) does not show as much improvement. Indeed, for the signal segment considered, the 2nd order forward adaptive predictor produces the least amount of signal compression of the four cases. Similar observations were made when the vocal tract predictor was replaced by a backward adaptive predictor.



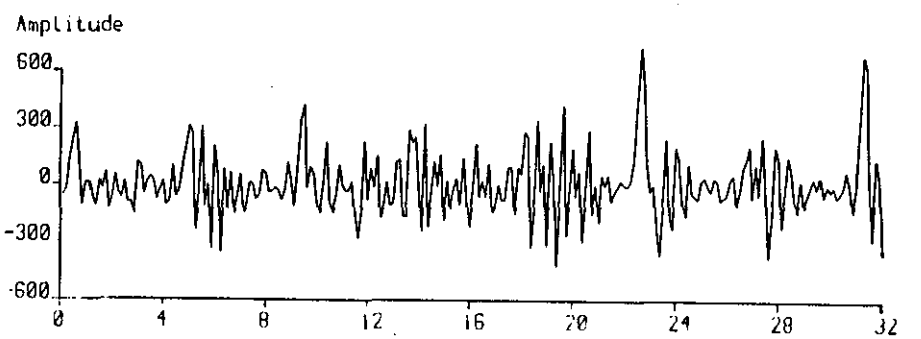
(i) 1st Order Fixed Predictor



(ii) 2nd Order Fixed Predictor

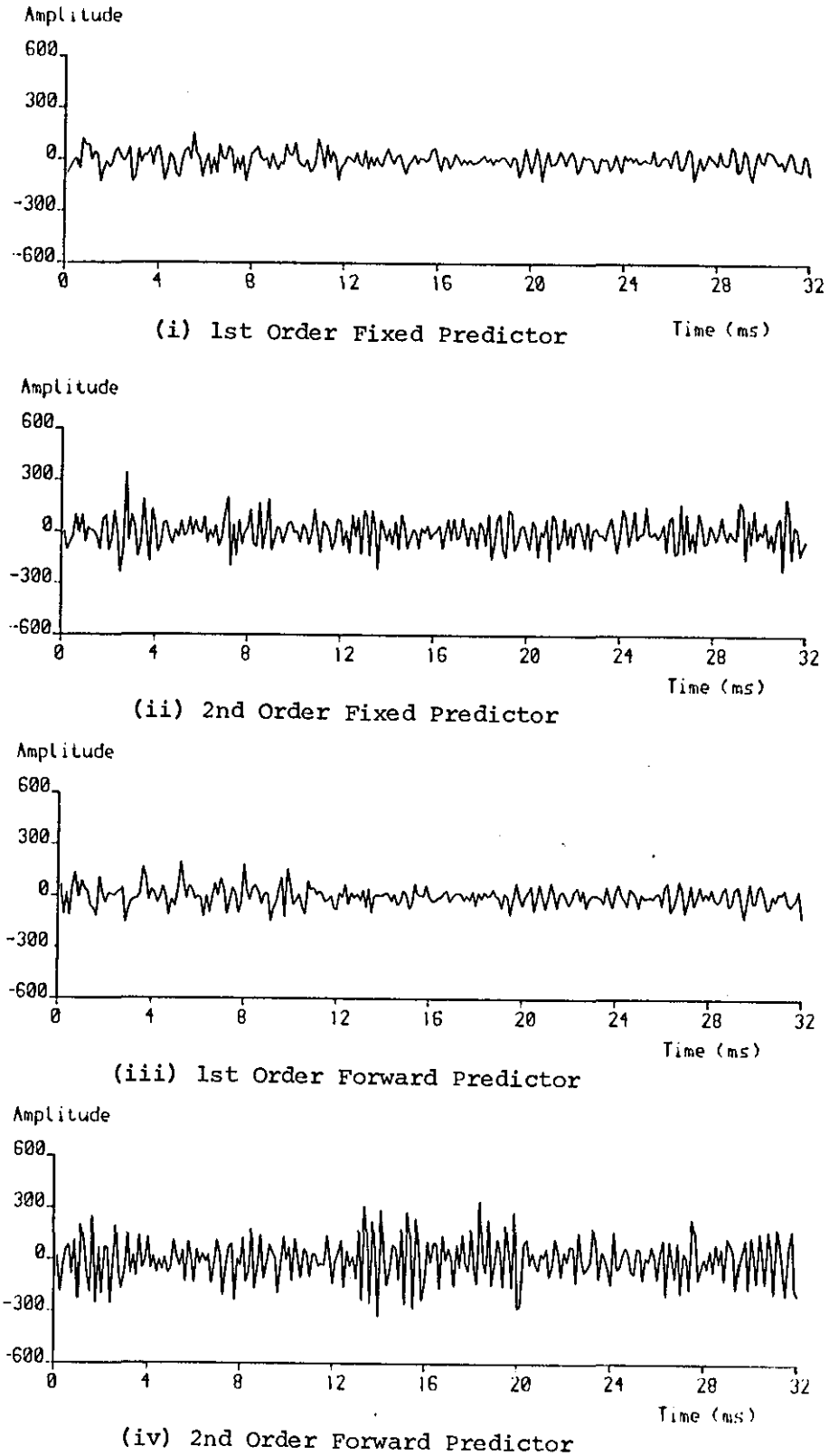


(iii) 1st Order Forward Predictor



(iv) 2nd Order Forward Predictor

Fig. 3.24(a) Speech Residual after Vocal Tract Prediction



(i) 1st Order Fixed Predictor

(ii) 2nd Order Fixed Predictor

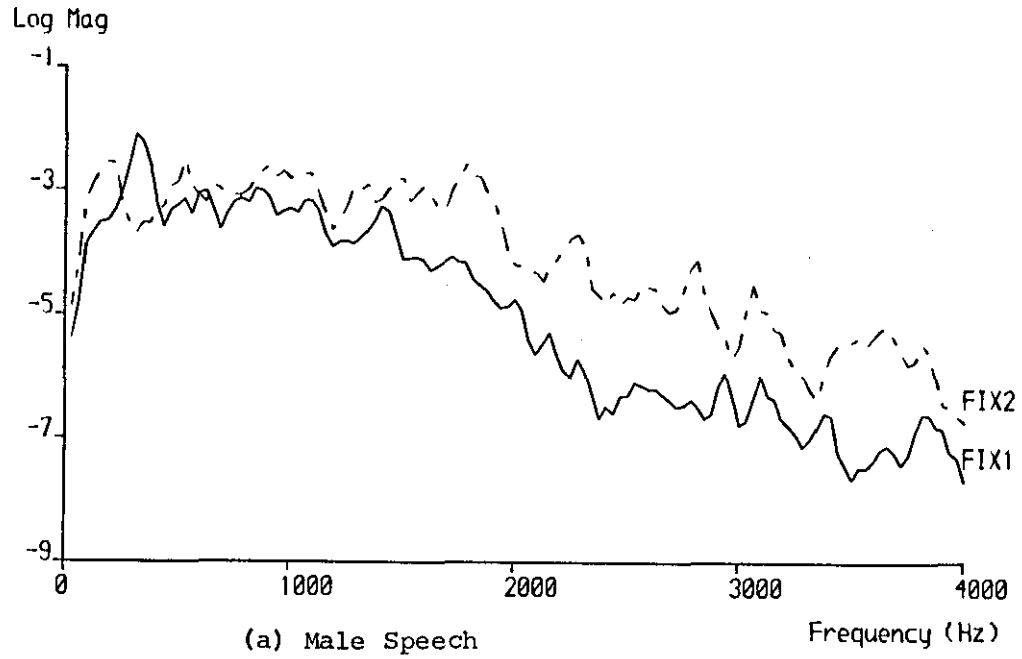
(iii) 1st Order Forward Predictor

(iv) 2nd Order Forward Predictor

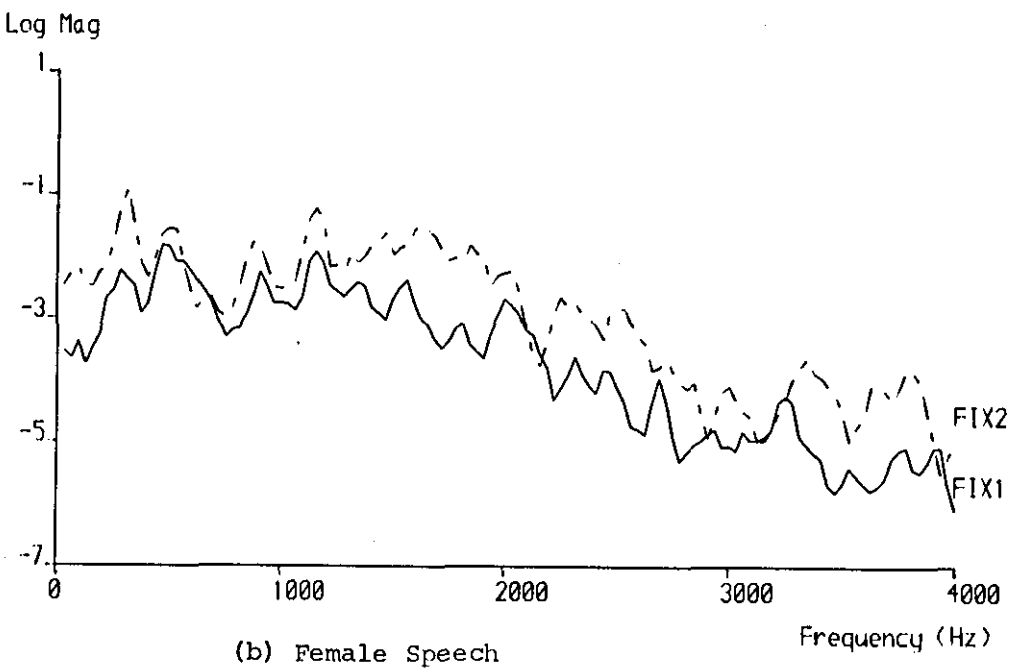
Fig. 3.24(b) Speech Residual After Further Pitch Prediction

Figure 3.25 shows the long-term average output noise spectra of the APC schemes employing 1st and 2nd order fixed prediction. It is clear that the simpler 1st order predictor, owing to its lower efficiency, was able to assist the pitch prediction process more, to give lesser overall output noise. Figure 3.26 provides a comparison of the noise spectra of both male and female speech for 3 schemes, namely, ADPCM with 2nd order fixed prediction, ADPCM with 2nd order forward prediction, and APC using first order vocal tract prediction. The advantage of APC over fixed prediction ADPCM is evident and expected. However, its performance with respect to the comparatively simpler forward adaptive ADPCM is not as impressive. For female speech, APC is possibly slightly better, while for male speech it is actually worse. Table 3.5 shows the average segmental SNR obtained for the various coding schemes considered.

It appears that the simplified APC system is unable to provide the required level of performance to justify the complexity involved in the use of the pitch predictor. Its SNR at best is no better than the simpler adaptive prediction ADPCM. Subjectively also, the decoded speech quality of the APC system offers little, if any perceptible advantage over that of 2nd order forward adaptive ADPCM.

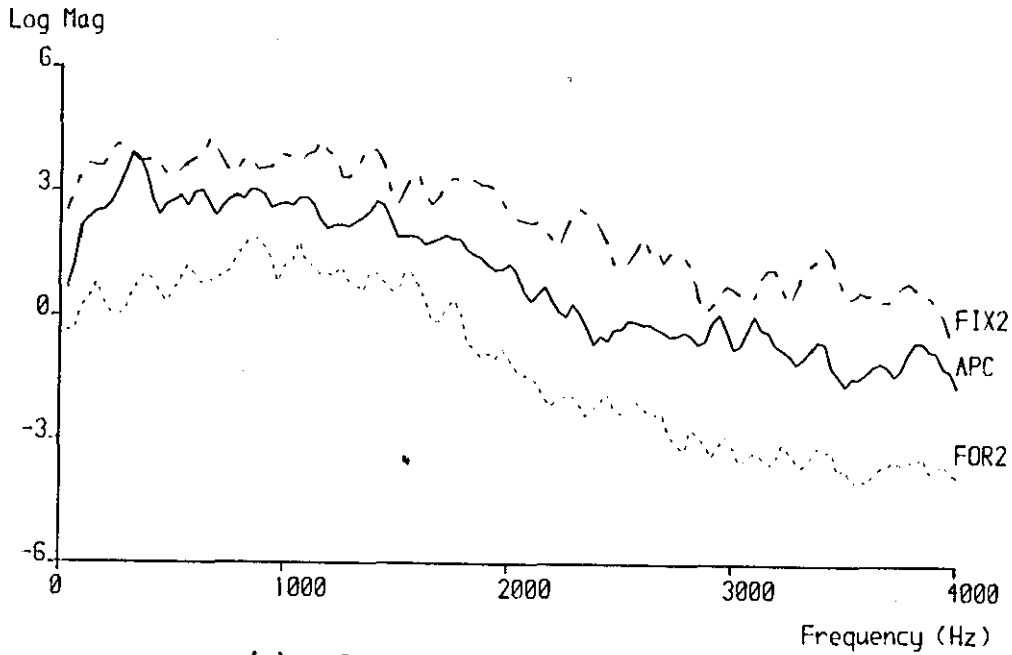


(a) Male Speech



(b) Female Speech

Fig. 3.25 Output Noise Spectra of APC Schemes Employing 1st and 2nd Order Fixed Vocal Tract Prediction



(a) Male Speech

(b) Female Speech

Fig. 3.26 Output Noise Spectra of
(i) ADPCM with 2nd Order Fixed Predictor
(ii) ADPCM with 2nd Order Forward Adaptive Predictor
(iii) APC with 1st Order Fixed Vocal Tract Predictor

Table 3.5 SNR Performance of Various Predictive Coding Schemes

Scheme	MALE	FEMALE
ADPCM:		
Fixed Predictor		
1st order	16.44	14.32
2nd order	16.02	15.26
Forward Adaptive Predictor		
1st order	16.81	14.54
2nd order	19.05	18.89
APC		
Fixed Predictor		
1st order	17.39	18.98
2nd order	16.53	18.39
Forward Adaptive Predictor		
1st order	17.20	18.73
2nd order	16.83	17.98

3.7.4 Discussion

Although the quasi-periodic nature of speech signals has been extensively studied for a long time, attempts to fully exploit this property to achieve efficient signal compression in speech coding applications have been largely unsuccessful without recourse to highly complicated implementations. Our investigation into simplified pitch adaptive schemes seems to have borne this out.

Apart from the sophisticated APC system proposed by Atal, a number of other pitch adaptive differential coders of varying degrees of complexity have been investigated by sundry researchers. The main problem with many of these schemes is the difficulty of accurate and reliable pitch extraction. Errors in pitch estimate also tend to

propagate (because of the length of filters used), giving rise to a reverberant quality in the recovered speech[83,112]. Another problem with APC systems encountered in our studies is the interaction between the two predictors employed. Short-term predictors which are efficient when used in isolation (as in ADPCM) proved to be less effective when combined with the long-term pitch predictor in APC. Our limited experiments appear to indicate that this is because the former produces a more random residual with the pitch structure blurred to some extent, thereby upsetting the operation of the latter, whose performance depends entirely on the accurate preservation of the pitch information. This observation was also noted by Jayant[86] during his investigation into pitch adaptive DPCM coding schemes. After performing several simulation studies, Jayant arrives at a configuration that uses a 3-tap fixed short term predictor, switching to a single-tap long-term predictor upon detection of strong periodicity (see figure 2.18). This pitch adaptive system was reported to provide a 3.8 dB advantage over the fixed 3-tap DPCM coder for female speech - a result which is in agreement with our simulation studies in the preceding section.

Backward adaptive APC systems have been investigated by Melsa, et al [223,230], using gradient algorithms or Kalman type adaptations, with little success. The main problem as before, is the difficulty of accurate pitch detection.

We conclude that while pitch adaptive schemes possess considerable merit as a powerful speech coding technique, its general applicability and usefulness has hitherto been largely limited by the complexity associated with reliable and accurate pitch prediction. Our studies indicate the difficulty of obtaining a relatively low complexity version

of the coder without significantly curtailing its potential. Indeed, the use of a pitch predictor in differential speech coders is by no means always desirable - Makhoul and Berouti decided in fact to discard this long-term predictor from their adaptive predictive scheme on the ground that its inclusion provides more bad than good on balance[112]. The reason is that the pitch predictor is not always effective, and errors present in the system tend to be propagated over long periods of time owing to the necessary length of the filter used.

3.8 CONCLUSION

Adaptive prediction is undoubtedly a promising and important area in speech coding, as is evident from the vast amounts of research devoted to the subject. Various forms of predictor adaptation have been examined in this chapter, including several novel variations on certain known algorithms. In the context of ADPCM coding of speech, the superiority of adaptive over fixed prediction has been unquestionably established. Variations in performance among different efficient adaptive algorithms however, are not as immediately apparent, and often other factors such as complexity and robustness predominate in the selection of an algorithm for a particular application. The backward block adaptive (BBA) prediction algorithm described in section 3.4.2.1 has been shown to provide good performance with relatively low complexity. Also, the block adaptation employed could possibly offer better robustness to transmission errors, although further experiments will have to be carried out for confirmation.

Pitch adaptive speech coding schemes have also been examined in some detail. While undoubtedly powerful in theory, such schemes are

unfortunately heavily dependant on accurate pitch prediction for efficient performance, and this has proved to be a severe limitation to their potential. Accurate pitch prediction is invariably linked with high complexity and/or long delays.

Algorithms for predictor adaptation are largely based on some form of minimum mean square error criterion, and predictor efficiency is often measured in terms of its SNR. Recent evidence has suggested however, that the SNR measure does not accurately reflect the subjective quality of the recovered speech, which is the ultimate test of any speech coding system. Much current interest has therefore been centred on various subjective criteria for use in speech coder assessment which will be more reliable indicators of speech quality. In particular, the concept of noise shaping to improve the perceptual quality of decoded speech has found widespread applications in a range of speech coders [81-82,112, 113,231-233]. This subject will be treated in more detail in the following chapter.

CHAPTER FOUR ADAPTIVE NOISE SPECTRAL SHAPING IN ADPCM SYSTEMS

4.1 INTRODUCTION

Traditionally, waveform coders have attempted to minimise the mean square error difference between the original and coded speech waveforms, and methods of assessing coder efficiency have conventionally been in terms of some form of signal-to-noise ratio (SNR) measurement[9,12,19, 20,37,211]. Recent studies have indicated however, that the perception of signal distortion is not based on the SNR alone. Indeed, it is now well recognised that the subjective loudness of distortion (or noise) in a coder depends to a considerable extent on both the short-time spectrum of the quantizing noise and its relation to the short-time spectrum of the speech signal. The theory of auditory masking suggests that noise in the formant regions could be partially or totally masked by the typically high energy low frequency components of the speech signal, so that much of the perceived noise in a coder comes from the high frequency regions where the signal level is low. Thus, the frequency components of the noise around the formant regions can be permitted to have higher energy relative to the components in the inter-formant and the high frequency regions[81,82,110,112,113,115,231-233].

Waveform coders which are designed based on a minimum mean-square error criterion produces an output noise signal which has a typically flat spectrum[81,82,110]. The subjective loudness of this noise could be reduced by appropriate shaping of its spectrum, trading a decrease in

the energy of the high frequency components for an increase in noise in the low frequency formant region. The principle of noise shaping is illustrated in figure 4.1. For minimum perceptual distortion, the noise spectrum should remain below the signal spectrum at all frequencies. However, for effective noise masking, the gap between the signal and noise levels must be sufficiently large (typically 20 dB or more)[233]. Techniques for performing such noise spectral shaping have been devised for both time and frequency domain speech coders and these have been applied with considerable success[12,19,40].

In this chapter, we consider only the technique of noise shaping applied to time domain coders, and in particular to the ADPCM coder operating at or around 16 Kbps. The theory of noise shaping is first reviewed and the various noise shaping coder configurations described. Simulation results are subsequently presented for two noise shaping ADPCM coders where parameter adaptation proceeds on a forward mode. Following this, backward adaptive methods for performing noise shaping are investigated. These have the advantage of not requiring side information for adaptation, so that the bit rate can be kept at 16 Kbps. Subjective listening tests on the recovered speech demonstrate the significant perceptual advantage provided by noise shaping, whether applied in a forward or a backward mode.

4.2 NOISE SPECTRAL SHAPING

Much of the current interest in the area of noise spectral shaping has arisen as a result of the work on APC of Atal and Shroeder[81], and Makhoul and Berouti[112], although the idea of shaping the noise spectrum has been present in the literature for a long time. Generally,

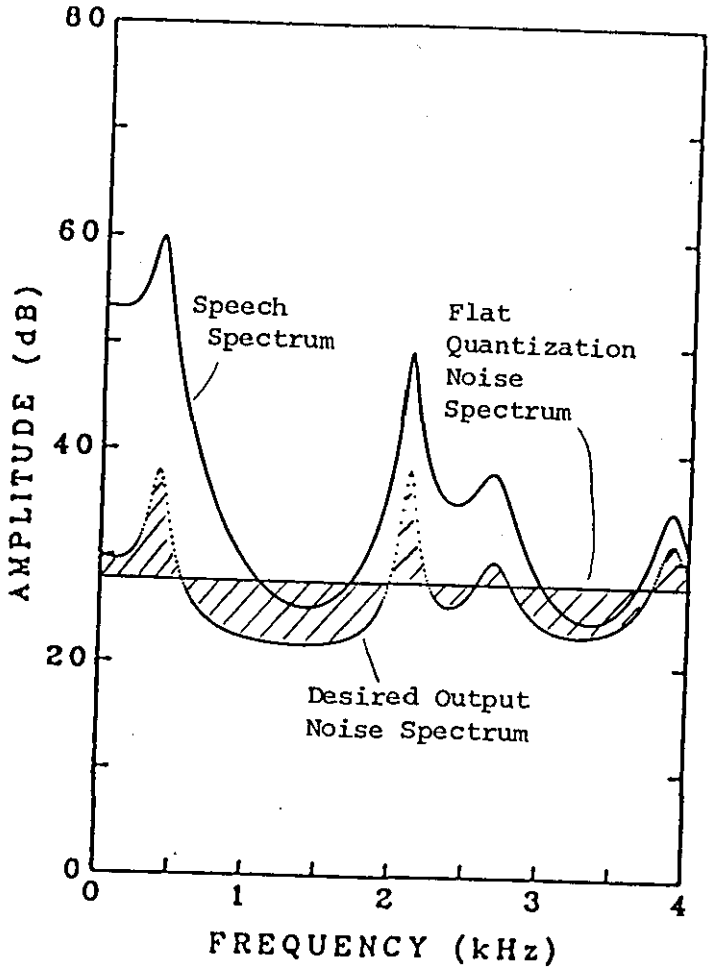


Fig. 4.1 An Example Showing the Output Noise Spectrum Shaped to Reduce Perceptual Distortion

control of the noise spectrum may be achieved in one of two ways - either by using noise feedback or by employing some form of signal pre-filtering.

4.2.1 Quantization Noise Feedback

In both the work of Atal and Makhoul, control of the noise shape is realised by incorporating an additional filter on the differential coder (APC or ADPCM) which feeds back the quantization noise i.e. the difference between the quantizer input and output. Figure 4.2 shows the noise-feedback coder employed by Atal, where P is the normal pth order linear predictor,

$$P(z) = \sum_{k=1}^{p} a_k z^{-k} \tag{4.1}$$

and F is a transversal filter given by,

$$F(z) = \sum_{k=1}^{m} b_k z^{-k} \tag{4.2}$$

Note that while his final design includes the pitch predictor, this has been left out in the analysis for simplicity, on the ground that it does not affect the basic principle involved. The quantizer input in figure 4.2 can be easily shown to be,

$$e(n) = x(n) - \sum_{k=1}^{p} a_k x(n-k) - \sum_{k=1}^{m} b_k q(n-k) \tag{4.3}$$

where,

$$q(n) = \hat{e}(n) - e(n) \tag{4.4}$$

denotes the quantization error at the nth instant. The coder output is now given as,

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

i.e.

$$\hat{e}(n) = \hat{x}(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k) \tag{4.5}$$

It follows from (4.3) to (4.5) that,

$$q(n) = \hat{x}(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k) - \{x(n) - \sum_{k=1}^{p} a_k x(n-k) - \sum_{k=1}^{m} b_k q(n-k)\}$$

i.e.

$$q(n) - \sum_{k=1}^{m} b_k q(n-k) = \hat{x}(n) - x(n) - \sum_{k=1}^{p} a_k \{\hat{x}(n-k) - x(n-k)\} \tag{4.6}$$

In frequency domain notation, (4.6) can be written as,

$$\hat{X}(\omega) - X(\omega) = Q(\omega) \frac{1 - F(\omega)}{1 - P(\omega)} \tag{4.7}$$

where $Q(\omega)$, $F(\omega)$ and $P(\omega)$ are the Fourier transforms of the quantization noise, $F(z)$ and $P(z)$ respectively.

For F=P, the output noise is the same as the quantizer noise, giving a flat frequency spectrum. However, with F≠P, the coder of figure 4.2 is able to control the shape of the output noise spectrum with appropriate choice of the feedback filter F. Under the assumption that the quantization noise is white, the spectrum of the coder output noise is determined only by the factor (1-F)/(1-P) as implied by (4.7). It can also be easily shown (see Appendix E) that the following constraint holds[81],

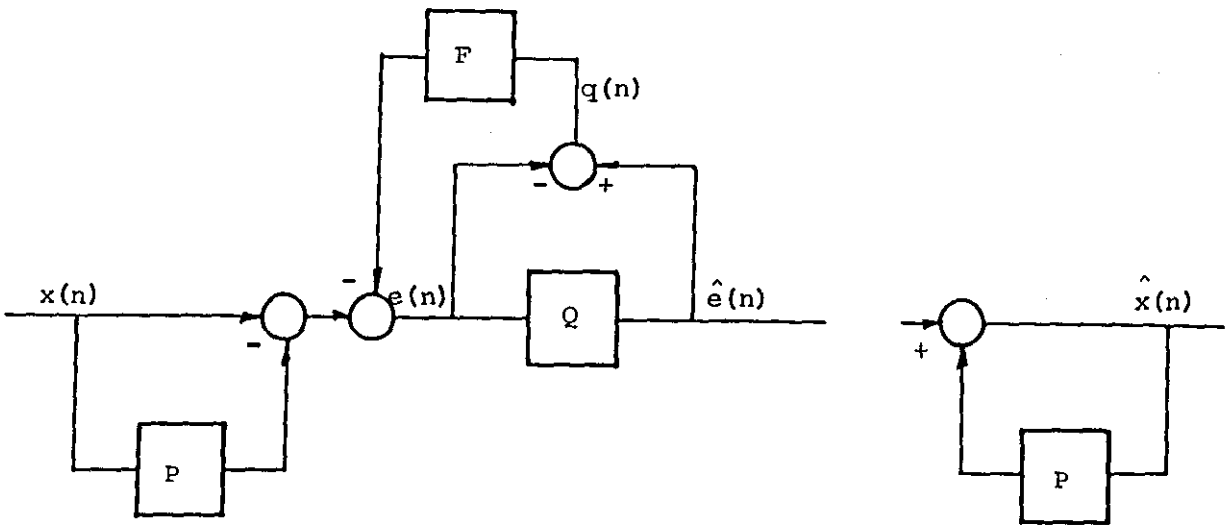


Fig. 4.2 Noise Shaping Coder Configuration Employed by Atal

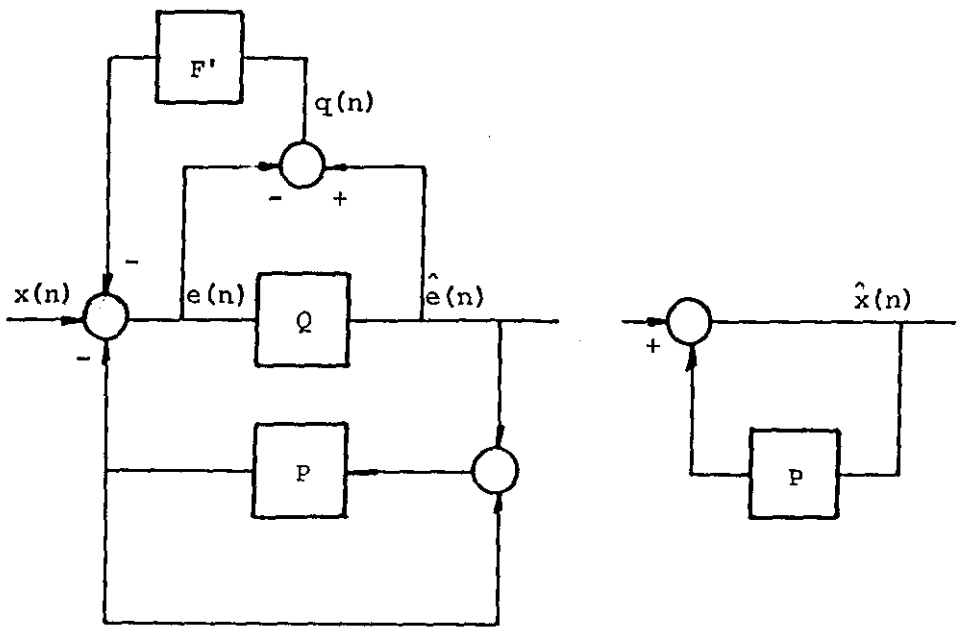


Fig. 4.3 Noise Shaping Coder Employed by Makhoul

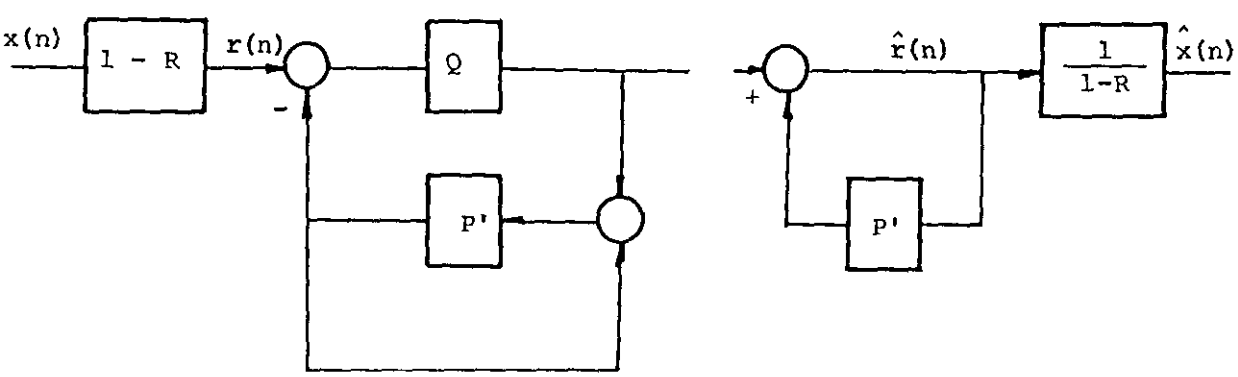


Fig. 4.4 Noise Shaping Coder Using Pre- and Post-filtering

$$1/f_s \int_0^{f_s} \log \Gamma(f)\,df = 0 \qquad (4.8)$$

where $\Gamma(f)$ is the squared magnitude of the factor $(1-F)/(1-P)$ at a frequency f,

$$\Gamma(f) = \left| \frac{1 - F(e^{2\pi jfT})}{1 - P(e^{2\pi jfT})} \right|^2 \qquad (4.9)$$

and T is the sampling interval.

The interpretation of (4.8) is that, assuming that the power of the quantizing noise is not changed significantly by the feedback loop, the average value of log power spectrum of the output noise is determined solely by the quantizer and is not affected by the choice of the filters F or P. In this way, the spectrum of the output noise can be shaped to suit perceptual requirements by reducing noise from one frequency region at the expense of increasing it in another (see figure 4.1). However, the constraint of (4.8) is in terms of log power spectrum, so that any deviation from the flat (minimum mean-square error) case will result in increased total noise power, although the areas above and below the average level i.e. the shaded areas will always be equal.

Atal suggested selecting the filter F to minimise an error measure in which the noise is weighted according to some subjectively meaningful criterion. This could be done by weighting the noise power at each frequency f by a function W(f). Since the ratio of noise power to signal power at any frequency f is proportional to $|1-F(e^{2\pi jfT})|^2$, one could choose F to minimise,

$$E = \int_0^{f_s} | 1 - F(e^{2\pi jfT}) |^2 W(f) df \qquad (4.10)$$

subject to the constraint,

$$\int_0^{f_s} \log \{ |1 - F(e^{2\pi jfT}) |^2 \} df = 0 \qquad (4.11)$$

Several interesting choices for W(f) were discussed. The first assumes that W(f) is constant for all frequencies, giving a solution F=0. The result is the feedforward D*PCM structure[110] (see section 2.4.1.6(c)), where the coder output noise has the same spectral envelope as the input speech. SNR is low and the reconstructed speech contains perceptible low frequency 'roughness'. Another choice is to let $W(f)=|1-P|^{-2}$, giving F=P, and the coder becomes effectively the ADPCM structure. This results in minimum unweighted noise power in the recovered speech, yielding a flat noise spectrum and a high SNR. The subjective quality is much less noisy than the previous case although a high frequency 'hiss' is audible. An intermediate choice between the two extremes can be made by letting

$$F(z) = P(z/\alpha) = \sum_{k=1}^{p} a_k \alpha^k z^{-k} \qquad (4.12)$$

where α controls the extent of noise shaping, from the flat minimum mean-square error case (F=P, α=1) to the fully shaped case (F=0, α=0). A value of α=0.7 was reported to provide the best subjective performance, eliminating the high frequency hiss without introducing low frequency roughness and yields an SNR slightly lower than the mmse case.

Noll[110] undertook a rigorous mathematical analysis of the generalised noise feedback coder (NFC) of figure 4.2 and showed that both DPCM and D*PCM are special cases of the NFC. DPCM is described as a fully

whitening filter while D*PCM only performs partial whitening.

The perceptual advantages obtained by shaping the output noise spectrum of APC coders was also investigated by Makhoul and Berouti[112,113]. The configuration used by them is shown in figure 4.3, where F′ is given by,

$$F'(z) = \sum_{k=1}^{m} b'_k z^{-k} \tag{4.13}$$

Again, the pitch predictor is not included in the analysis and in fact, it was discarded by Makhoul in his final design. The difference between this configuration and that employed by Atal is in the position of the vocal tract predictor P. Nevertheless, the coders are the same with regard to their noise shaping ability. From figure 4.3, the quantizer input is given by,

$$\hat{e}(n) = x(n) - \sum_{k=1}^{m} b'_k q(n-k) - \sum_{k=1}^{p} a_k \hat{x}(n-k) \tag{4.14}$$

The receiver is similar to that of figure 4.2, so the recovered output is,

$$\hat{x}(n) = \hat{e}(n) + \sum_{k=1}^{p} a_k \hat{x}(n-k) \tag{4.15}$$

From (4.4), (4.14) and (4.15),

$$q(n) = \hat{x}(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k) - \{x(n) - \sum_{k=1}^{m} b'_k q(n-k) - \sum_{k=1}^{p} a_k \hat{x}(n-k)\}$$

i.e.

$$\hat{x}(n) - x(n) = q(n) - \sum_{k=1}^{m} b'_k q(n-k) \tag{4.16}$$

(4.16) can be written in frequency domain notation as,

$$\hat{X}(\omega) - X(\omega) = [\ 1 - F'(\omega)\ ]Q(\omega) \qquad (4.17)$$

In this case, the shape of the output noise spectrum is determined by the factor $[1-F'(\omega)]$, where F' as before can be chosen to satisfy perceptual criteria. From (4.7) and (4.17), it can be seen that the noise shaping coders of figures 4.2 and 4.3 can be made equivalent by setting,

$$1 - F'(z) = \frac{1 - F(z)}{1 - P(z)}$$

giving,

$$1 - F(z) = [\ 1 - F'(z)\ ][\ 1 - P(z)] \qquad (4.18)$$

To obtain the same noise shape as before,

$$F'(z) = 1 - \frac{1 - P(z/\alpha)}{1 - P(z)} \qquad (4.19)$$

Note that in both coders, the introduction of noise shaping involves only the modification of the transmitter of the ADPCM structure - the receiver remains the same.

4.2.2 Adaptive Pre-filtering

A third configuration for shaping the output noise spectrum in a similar manner consists of a pre- and post-filtering arrangement on a differential coder[82], as shown in figure 4.4. In this case,

$$1 - R(z) = \frac{1 - P(z)}{1 - P(z/\alpha)} \qquad (4.20)$$

It is clear from the figure that,

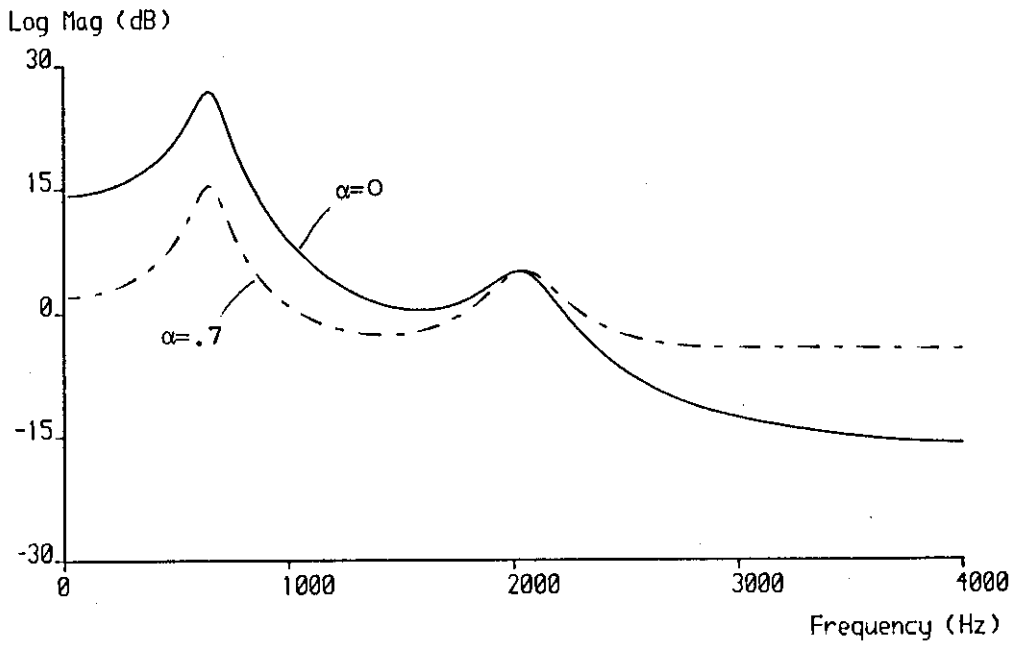
$$\hat{X}(\omega) - X(\omega) = \frac{Q(\omega)}{1 - R(\omega)} = Q(\omega)\ \frac{1 - P(z/\alpha)}{1 - P(z)} \qquad (4.21)$$

i.e. the quantization noise spectrum is again shaped by the factor (1-F)/(1-P). Note that the predictor P′ in figure 4.4 is optimised for the pre-filtered speech {r(n)}, while P in (4.20) is optimised from the original speech in the same way as the previous two configurations. The structure of figure 4.4 is a relatively less studied noise shaping coder. This could be due to the fact that a fully adaptive version of the coder requires two sets of predictor coefficients to be computed and transmitted i.e. the normal coefficients for P′ plus the coefficients for R required for noise shaping. The consequent increase in delay, transmission rate (due to the additional side information) and complexity is generally difficult to justify.

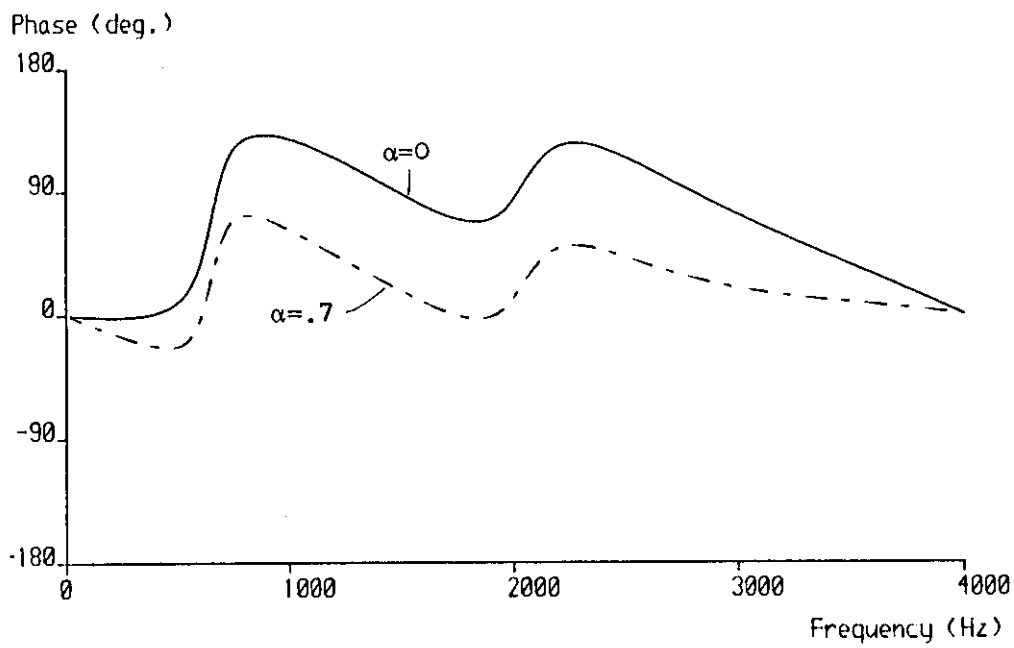
The final output noise for all 3 coders is the flat quantization noise $Q(\omega)$ shaped by the factor (1-F)/(1-P). The frequency response of this noise shaping transfer function (1-F)/(1-P) is given in figure 4.5 for an 8th order filter. The solid curve represents the envelope of the speech input, modelled by the filter 1/(1-P) and the broken curve illustrates a typical noise shape which results from the factor (1-F)/(1-P) with α=0.7. The two formants of the speech waveform can be clearly seen in the figure.

4.2.3 Discussion

Atal[81] reported good toll quality speech from his APC coder with noise shaping, using a 10th order vocal tract predictor, a 3-tap pitch predictor (equation (3.67)) and a 3-level forward adaptive quantizer (AQF) optimised for a Gaussian distributed signal. With an input sampling frequency of 8 kHz, the transmission bit rate is in the region of 16 Kbps. Makhoul[112] also obtained good quality speech "virtually



Magnitude Response



Phase Response

Fig. 4.5 Frequency Response of (1-F)/(1-P) with α=0.7

indistinguishable from the original" at 16 Kbps using an 8th order predictor and a 19-level entropy coder for the prediction residual. He uses a lower sampling rate of 6.67 kHz for his input however, thus allowing more bits effectively to code the residual signal and side information. While there is little doubt that the two schemes are able to achieve very good quality coded speech at 16 Kbps, their one common drawback is the high complexity involved; in the former case, with a complicated pitch predictor and in the latter, with variable bit rate entropy coding. Obviously, such complex implementations contribute greatly to the overall system performance, and could possibly 'mask' the full potential of noise shaping. It would be interesting therefore, to consider the effectiveness of noise shaping applied to coders at a lower level of complexity.

We investigate in the following sections such less complex differential speech coders which utilise the concept of noise shaping. These may be divided into two groups and considered separately, depending on whether forward or backward adaptation of the parameters is employed.

4.3 FORWARD ADAPTIVE NOISE SHAPING

In the noise shaping coders proposed by both Atal and Makhoul, the coefficients of the noise feedback filter F and F' are obtained from those of the vocal tract predictor, and these parameters are derived using forward block adaptive (FBA) prediction[33,47] (see section 3.3.1). The quantizer employed is also forward block adaptive (see section 2.4.1.1 b(i)). The use of such forward adaptation implies the need for delay and side information transmission. Generally, the delay would be equal to the blocksize used for the calculation of the optimum

predictor coefficients, since a block of input samples have to be buffered for this purpose. The quantizer step-size can be estimated from the same block of input samples so that no additional delay is required. The adaptation rate of the predictor need not be the same as that for the quantizer. Frequently, the predictor is able to tolerate less frequent updating of its coefficients, and consequently the blocksize used is also larger.

It was decided to investigate the effectiveness of noise shaping techniques applied to the simple ADPCM coder, employing 2-bit quantization. To keep the complexity to a minimum, with a view on practical implementability, only the basic features of the noise shaping coder as described by Atal or Makhoul were retained. The noise shaping coder of figure 4.3 (which shall be denoted as NSF1) was simulated and compared with the pre-/post-filter configuration of figure 4.4 (denoted as NSF2). Note that the coder of figure 4.2 is equivalent to figure 4.3, given the relation of (4.18) and provides identical results with the same choice of noise shaping factor α.

4.3.1 Computer Simulation Results

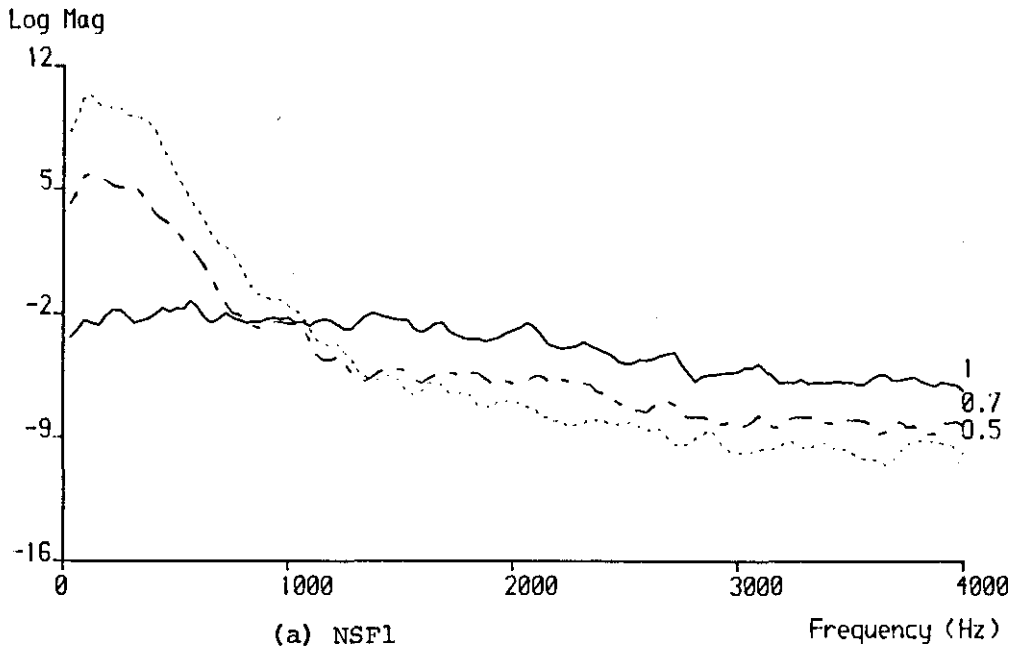
Preliminary experiments were conducted to determine the optimum values of parameters to be used in the simulation. The predictor coefficients are computed from the input signal every 256 samples (32 ms) and transmitted as side information. A 4th order predictor is used. An estimate of the standard deviation of the prediction residual (which is the quantizer input) is made every 64 samples (8 ms) from the input signal, using feed-forward adaptive prediction. This is given as:

$$\sigma = c \frac{1}{M} \sqrt{\sum_{j=1}^{M} \{ x(n-j) - \sum_{k=1}^{p} a_k x(n-j-k) \}^2} \qquad (4.22)$$

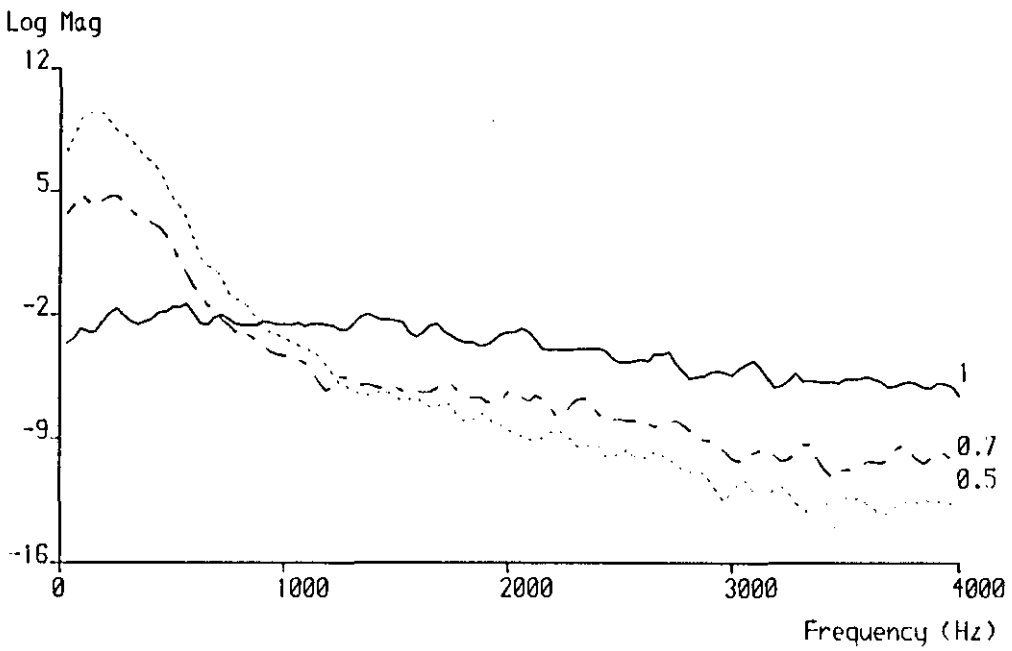
with M=64 and p=4. The optimum scaling factor c, to account for the quantization noise present in the actual quantizer input was determined experimentally to be 1.5. The quantizer used is optimised for signals with a Gaussian distribution. Quantizers optimised for other distributions (such as the Laplacian, gamma and uniform pdfs[43,45]) were also tried but were found to provide inferior results in terms of SNR. The exception is the Laplacian quantizer, which appears to perform rather well, particularly for female speech.

For each of the schemes NSF1 and NSF2, the noise shaping factor α was varied over the range between 0 and 1. Figure 4.6 shows the long-term average log magnitude spectra of the output noise produced by each scheme for various α, and clearly demonstrates the effect of noise shaping. Table 4.1 summarises the total and segmental SNR values for 2 seconds of speech obtained.

The SNR generally decreases as the extent of noise shaping is increased, as expected since the total noise power is also increased. It is interesting however, that at all levels of noise shaping, the SNR of NSF2 is better than that of NSF1. This observation is borne out by the comparison of the output noise spectra produced by the two schemes (for α=0.5 and 0.7) as shown in figure 4.7, where it can be seen that the output noise level of NSF2 is consistently lower than that of NSF1.



(a) NSF1



(b) NSF2

Fig. 4.6 Output Noise Spectra for NSF1 and NSF2 for Different Noise Shaping Factors

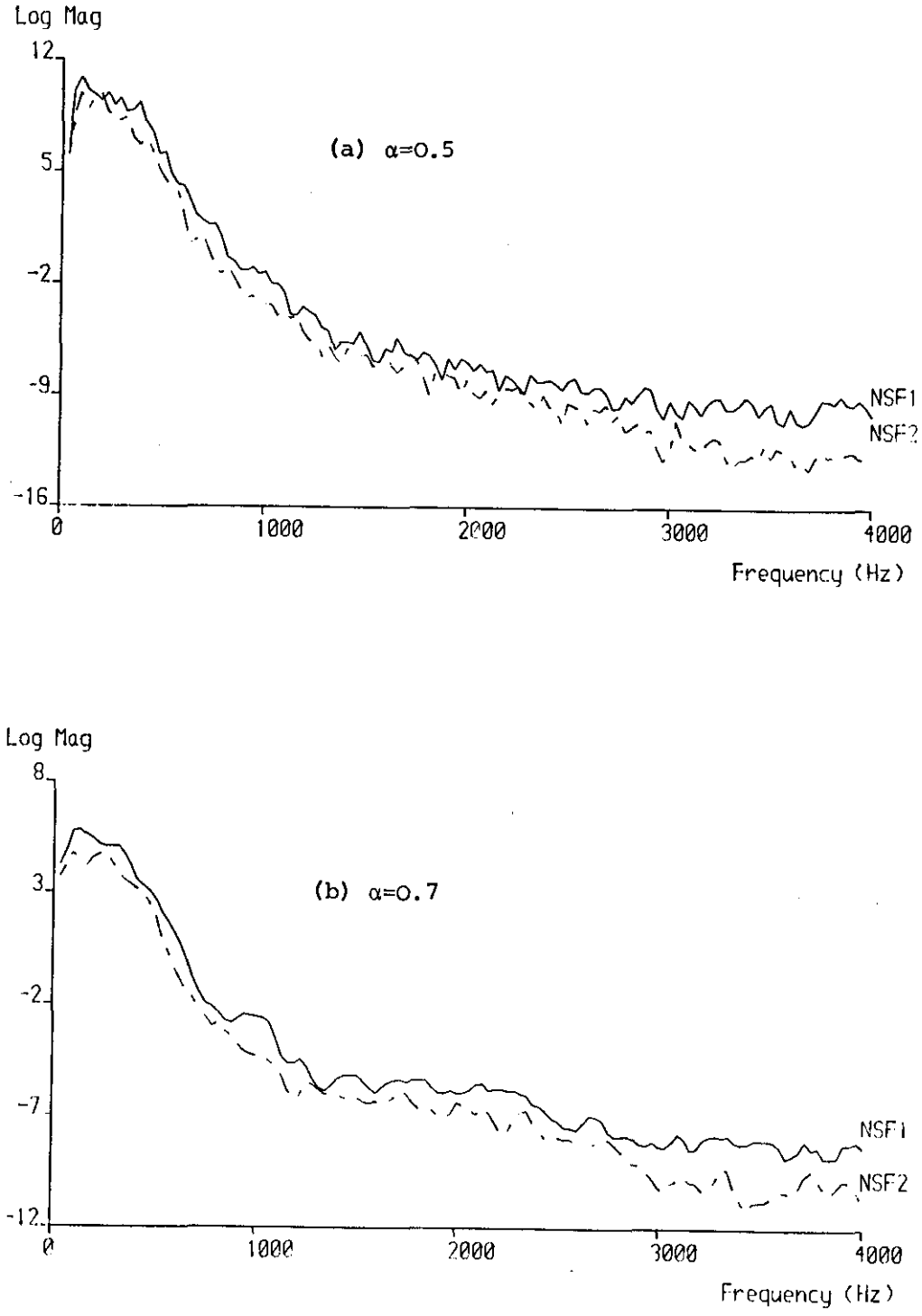


Fig. 4.7 Output Noise Spectra of NSF1 and NSF2 with 2-Bit Quantization

Table 4.1 SNR performance of Noise Shaping Coders NSF1 and NSF2 (2 s of Male and Female Speech)

Noise Shaping Factor α	NSF1 CODER MALE SSNR	NSF1 CODER MALE TSNR	NSF1 CODER FEMALE SSNR	NSF1 CODER FEMALE TSNR	NSF2 CODER MALE SSNR	NSF2 CODER MALE TSNR	NSF2 CODER FEMALE SSNR	NSF2 CODER FEMALE TSNR
1.0	21.45	20.59	21.41	20.49	21.45	20.59	21.41	20.49
0.9	21.28	20.81	21.28	20.20	21.61	20.86	21.63	20.88
0.8	20.22	20.10	20.26	19.46	20.93	20.33	21.55	20.46
0.7	18.70	18.95	18.92	18.18	20.01	19.41	20.95	20.16
0.6	17.56	18.01	17.63	17.17	18.67	18.44	19.84	18.86
0.5	15.99	16.75	16.05	15.59	17.11	17.25	18.72	17.98
0.0	7.78	6.74	8.81	7.96	11.08	10.97	13.33	12.63

Recordings of the output speech were made for a range of α values and the best subjective performance was found to be for α=0.6 to 0.7, a finding in good agreement with Atal and Makhoul. Listening tests indicate a clear preference for the decoded speech produced by NSF2, consistent with the above observation on SNR and output noise spectra.

4.3.2 Discussion of Simulation Results

The quite significant difference in performance between the two noise shaping coders is unexpected as it has been generally accepted (albeit without experimental evidence) that they should produce very similar results[82]. A possible explanation for this observation is given as follows[212]:- In the NSF1 coder, the predictor P is optimised from the input signal but operates on the decoded speech samples which are corrupted by quantization noise. This limits the accuracy of the prediction process and produces a certain power of the residual signal. The residual signal in turn determines the quantization noise power

(assuming no quantizer overload) which defines the level about which eventual noise shaping is performed. Thus in figure 4.6(a), the noise spectra for various α values are shaped about the level given by α=1, so that the areas bounded by each curve above and below this line are approximately equal. In the NSF2 scheme however, the pre-filter 1-R (which is essentially a spectral flattener) operates in a quantization noise-free environment on the input speech and reduces the power of the signal to be presented to the ADPCM encoder to follow. The combined action of the pre-filter and the ADPCM predictor P′ results in a prediction residual at the quantizer input, which has a variance smaller than that of NSF1. The flat quantization noise spectrum of this residual is thus also lower. This lower quantization noise however, is obtained only at the expense of noise accumulation at the receiver, when post-filtering has to be applied to restore the spectral balance of the signal. The amount of noise accumulation will be proportional to the extent of whitening produced by the pre-filter. But in this case, this necessary noise accumulation process is used to advantage to perform the noise shaping. The post-filter 1/(1-R) shapes the noise about the reduced noise level, to give an output noise spectrum which possesses the same shape as that produced by NSF1 (for the same α), but with a consistently lower magnitude across all frequency components. Figure 4.8 illustrates the different levels about which the output noise is shaped, for the two schemes.

The better performance of NSF2 is due to a net gain from the effects of two conflicting processes:-

(i) the feed-forward pre-filter, which reduces the variance of the quantizer input and hence the quantization noise, and

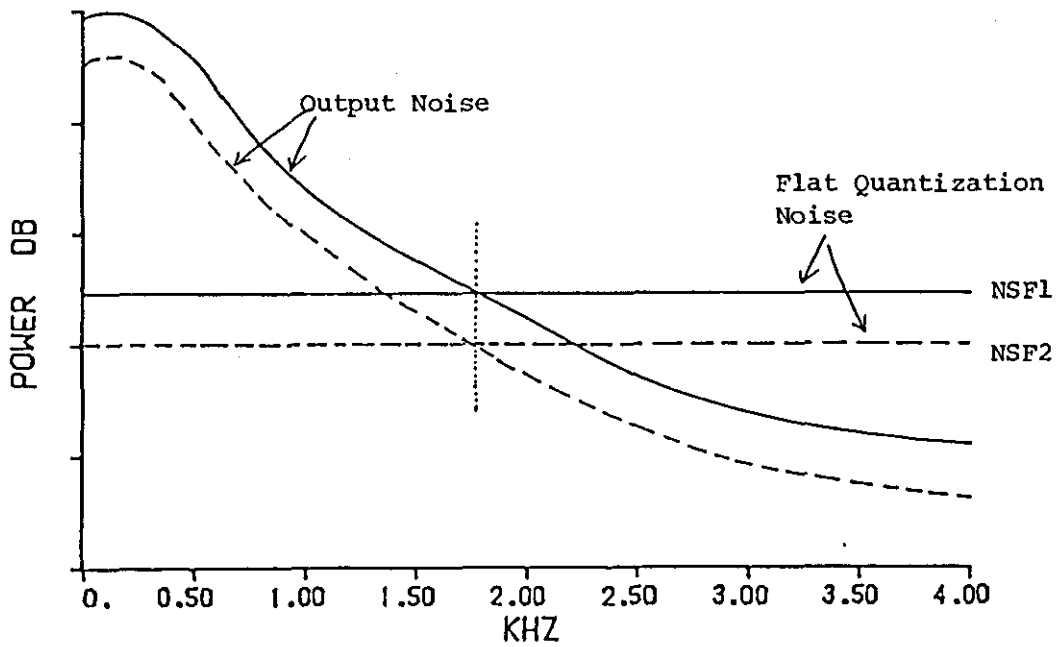


Fig. 4.8 Illustration of Noise Shaping Levels for NSF1 and NSF2

(ii) the post-filter which accumulates noise at the receiver.

In the results obained, it would appear that the reduction in noise due to the pre-filter is greater than the corresponding noise accumulation, so that overall improvement over NSF1 is obtained. Indeed, from table 4.1, it can be seen that the SNR of NSF2 is actually increased (albeit only slightly) when a small degree of noise shaping is applied (α=0.9). The relative contribution of the pre- and post-filter to the performance of NSF2 appears to be a function of the fineness of quantization employed. When quantization is coarse, the effectiveness of the ADPCM predictor P′ is quite severely limited by the relatively greater amounts of quantization noise present in its input, so that the effect of the pre-filter is predominant. As the number of quantizer levels is increased however, the contribution of the pre-filter would also be diminished, and the noise accumulation at the receiver becomes more significant. Hence, it would be expected that the margin of improvement of NSF2 over NSF1 would be inversely related to the fineness of quantization. To investigate the validity of this hypothesis, the performance of the two coders were examined under conditions of fine (4-bit) quantization. Figure 4.9 shows the output noise spectra of the two schemes obtained with 4-bit quantization, for two values of α. It can be seen that the superiority of NSF2 over NSF1 does indeed diminish as finer quantization is employed.

4.3.3 Fixed Pre-filtering

The better performance of NSF2 over NSF1 is obtained at the expense of increased complexity, delay and transmission bit rate, since an additional set of predictor coefficients needs to be computed and

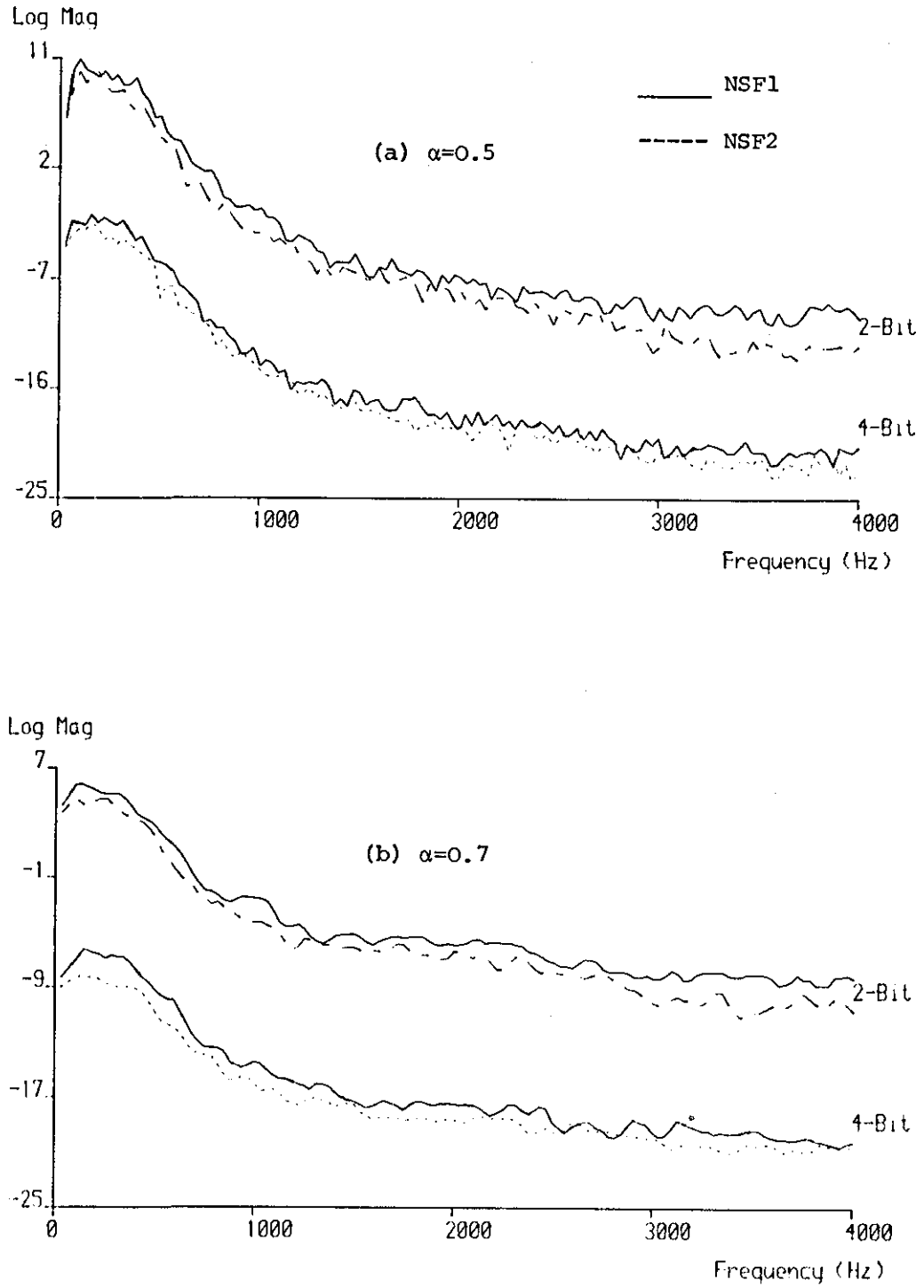


Fig. 4.9 Output Noise Spectra of NSF1 and NSF2 Under Conditions of Coarse (2-bit) and Fine (4-bit) Quantization

transmitted. It was decided to investigate if this performance could be maintained without the penalty of a higher bit rate. One way to keep the same bit rate for both NSF1 and NSF2 is to allow the pre-filter in the latter scheme to be fixed. Two fixed pre-filters were examined in relation to NSF2. The first, denoted FP1 is a simple first order pre-emphasis, given by,

$$1 - R(z) = 1 - \beta z^{-1} \tag{4.23}$$

and the second (FP2) is a second order filter of the form given by (4.20), where the coefficients a_1 and a_2 are derived from the long-term autocorrelation function of speech. The parameters of FP1 and FP2 used in the simulation were determined experimentally to be β=0.8 and α=0.7. Figure 4.10 shows the output noise spectra of both schemes compared to NSF2. It is seen that the FP2 codec fails to suppress the high frequency noise to the extent of either FP1 or NSF2 and provides a 'hump' in the noise spectrum corresponding to the poles of the pre-filter used. FP1, on the other hand, is able to provide a well-balanced low and high frequency performance, giving a long-term average noise spectrum rather similar to NSF2. It must be remembered however, that unlike the latter system where the noise tracks the short-term speech spectrum, the shaping provided by FP1 is non-adaptive. The subjective quality of the recovered speech produced by the FP1 codec is a little worse than that of NSF2 but is comparable, if not slightly better than NSF1.

The predictor-quantizer interaction noted in the preceding discussion for the adaptive pre-filtering scheme NSF1 is also applicable to FP1. Figure 4.11 shows the segmental SNR of FP1 as a function of the pre-emphasis coefficient, for 2 and 4 bit quantization. It can be seen that

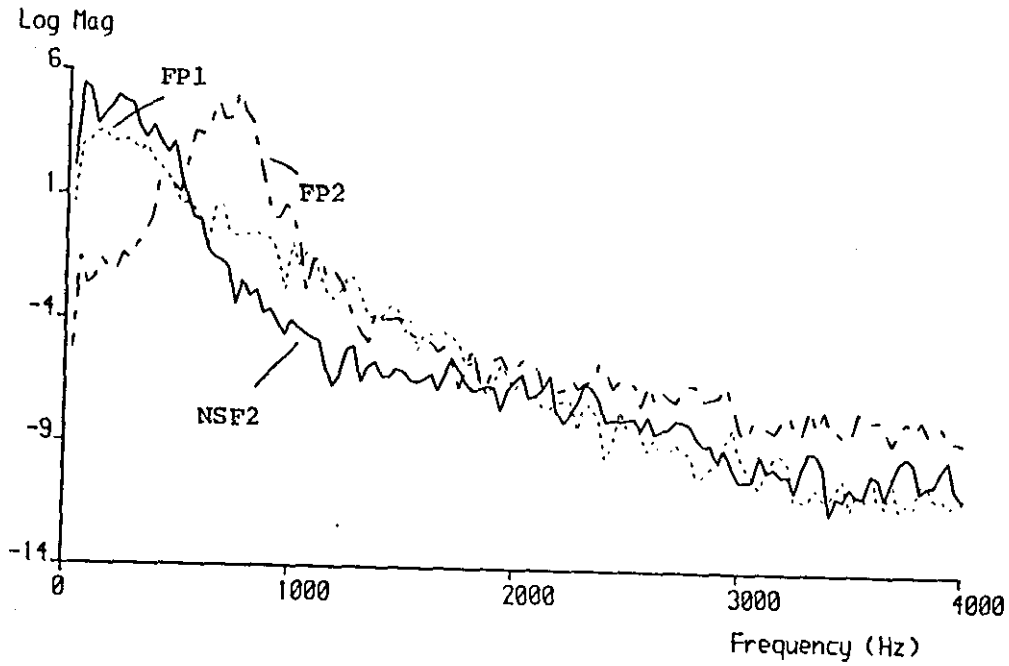


Fig. 4.10 Output Noise Spectra of Fixed Pre-filter Schemes FP1 & FP2 Compared to NSF2

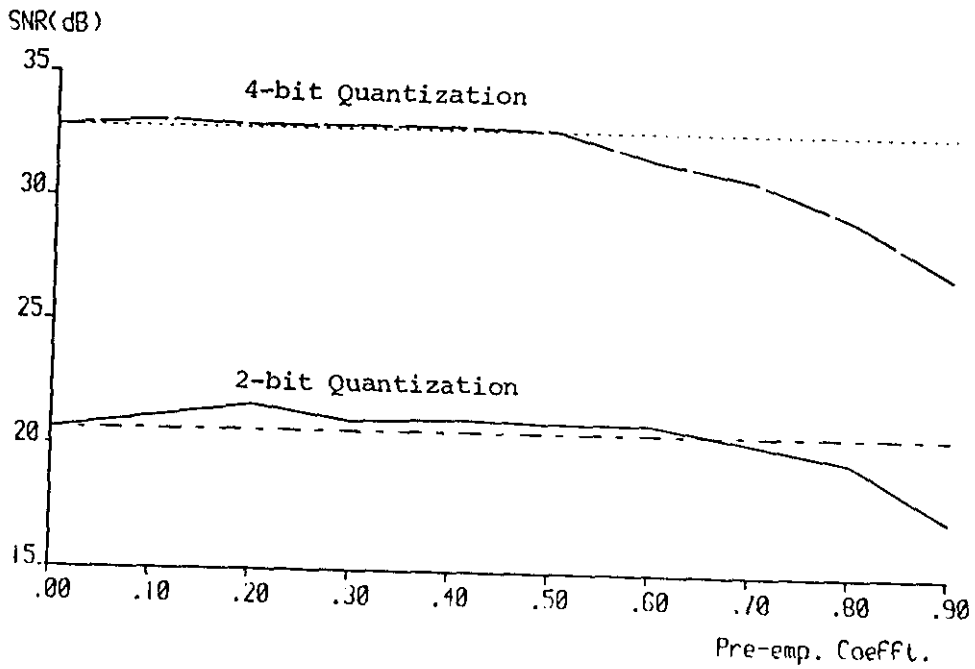


Fig. 4.11 Segmental SNR of FP1 as a Function of the Pre-emphasis Coefficient

for coarse quantization, over a sizeable range of pre-emphasis values, the SNR is higher than when no pre-emphasis is applied. For fine quantization however, the effect of pre-emphasis is clearly to reduce the SNR, and at a quicker rate too.

4.3.4 Conclusion

From the preceding investigation, the effect of applying noise shaping to improve the perceptual quality of ADPCM decoded speech has been demonstrated at a transmission bit rate of about 16 Kbps. At this bit rate, and with a relatively low level of coder complexity, communications quality speech is possible using the basic adaptive prediction ADPCM coder with noise shaping. Interaction between the various components of the coder can be exploited to provide improved performance in the application of noise shaping. It was found that for the same noise shape, the use of an adaptive pre-/post-filtering arrangement to perform noise shaping produces better results than the conventional adaptive noise feedback coder. Specifically, for a relatively simple ADPCM coder operating under coarse quantization conditions, noise shaping is best applied using a fixed pre-emphasis. This is able to produce a quality of the recovered speech equivalent to or better than the more complex adaptive noise feedback coder.

The same experiments as described above were performed on wide band speech, band-limited from 0 - 7 kHz and sampled at 16 kHz, giving a transmission rate of 32 Kbps plus side information. Similar observations to the narrow band speech were obtained in all cases[212].

4.3.5 Note on Publication

A paper entitled, "Noise Spectral Shaping Applied to Coarse Quantization Differential Speech Coders" was presented at the Mediterranean Electrotechnical Conference (MELECON 1983) in May 1983 and was recorded in the Conference Proceedings p. C1.08. This paper was written in co-authorship with Dr. C.S. Xydeas and Mr. S.N. Koh and covers the work described in section 4.3 of this chapter.

4.4 BACKWARD ADAPTIVE NOISE SHAPING

The work described in the preceding sections, and indeed previous work documented in the literature on noise shaping in predictive coding schemes, have largely involved systems employing forward adaptive predictors[81-82,112] and/or quantizers[81-82,112,231]. The delay and side information associated with such forward adaptation has been a major drawback of these otherwise effective systems. Backward adaptive schemes do not have these problems and are therefore more attractive in many applications[68]. We develope in the following sections, methods for applying the concept of noise shaping to enhance the quality of the coded speech produced by differential speech coders, which are not excessively complex, and for which no delay or side information are required. These are able to operate at 16 Kbps using 2-bit quantization.

4.4.1 Description of Backward Noise Shaping Coder

The coder to be employed for this purpose is of the general adaptive differential structures shown in figures 4.2 to 4.4. The constraint on

side information implies the need for a backward strategy for all adaptation - prediction, quantization and noise shaping. Results on backward adaptive predictors in chapter 3 suggest the use of the backward block adaptive (BBA) predictor for the purposes of adaptive prediction and noise shaping. The similarity of the BBA to the FBA predictor allows most of the work developed for the latter to be directly and conveniently applied. Furthermore, because of the noise accumulation effect inherent in noise shaping schemes, the use of block adaptation is to be preferred to sequential adaptation as far as the risk of instability is concerned. And obviously, with the amount of adaptation involved, the computational demands of the BBA predictor are relatively modest compared to the sequential methods[225]. Backward quantizer adaptation is easily implemented using the 2-bit Jayant quantizer (AQJ)[49].

Two fully backward adaptive noise shaping schemes, denoted as NSB1 and NSB2 are proposed and described in the following[213,215]. In both cases, an 8th order BBA predictor is used. The predictor coefficients are computed from past decoded signal samples using a blocksize of 256 samples, and these are updated every 32 sampling instants, as described in section 3.4.2.1.

4.4.1.1 Scheme 1 (Quantization Noise Feedback)

The first scheme follows directly from the conventional noise shaping coder structure of figure 4.3. The noise feedback filter F' adapts according to (4.19), where P is the BBA predictor. The effect of noise shaping is clearly seen in the output noise spectra for different values of α, shown in figure 4.12. Listening tests indicate an optimum α value

of 0.7, which is the same value obtained in the forward adaptive cases. This value of α provides the best compromise in terms of subjective quality between the high and low frequency distortions, eliminating much of the high frequency 'hiss' without increasing low frequency 'rumble' appreciably. As shaping is increased (by decreasing α) however, the low frequency 'roughness' and 'breathiness' becomes increasingly apparent and the quality deteriorates.

4.4.1.2 Scheme 2 (Adaptive Pre-filtering)

The work on noise shaping in forward adaptive ADPCM systems (section 4.3.2)[212] suggests the possibility of exploiting predictor-quantizer interaction to reduce the level of quantization noise about which shaping is performed. It was found, in the simulation of the forward adaptive schemes, that the application of noise shaping using a pre-/post-filter arrangement on the basic ADPCM coder provides a clear perceptual and SNR advantage over the conventional noise feedback coder.

We decided to investigate if the same observation is true for the similarly configured backward adaptive noise shaping system. We note above, that the adaptive pre-filter scheme NSF2 requires the transmission of the pre-filter coefficients in addition to the ADPCM predictor parameters - a requirement which is clearly unacceptable for operating at a transmission bit rate of 16 Kbps using 2-bit quantization. Since the BBA predictor adapts in a backward mode, the pre-filter coefficients can also be made to adapt according to the BBA predictor to avoid the transmission of side information. The configuration used to incorporate such a backward adaptive pre-filter

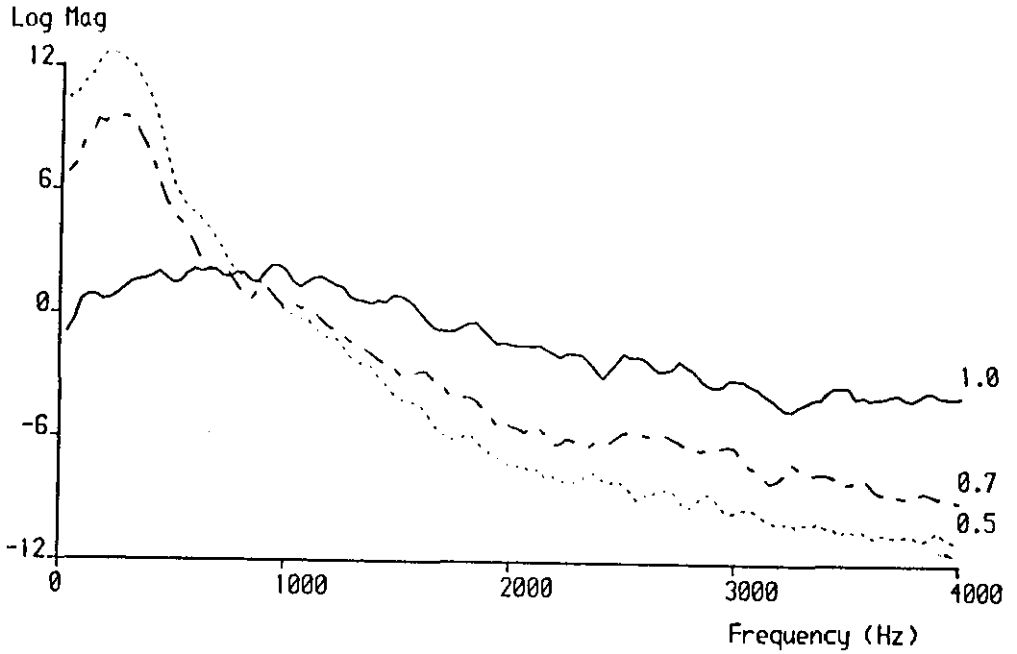


Fig. 4.12 Output Noise Spectra of NSB1 Coder for Different Noise Shaping Factors

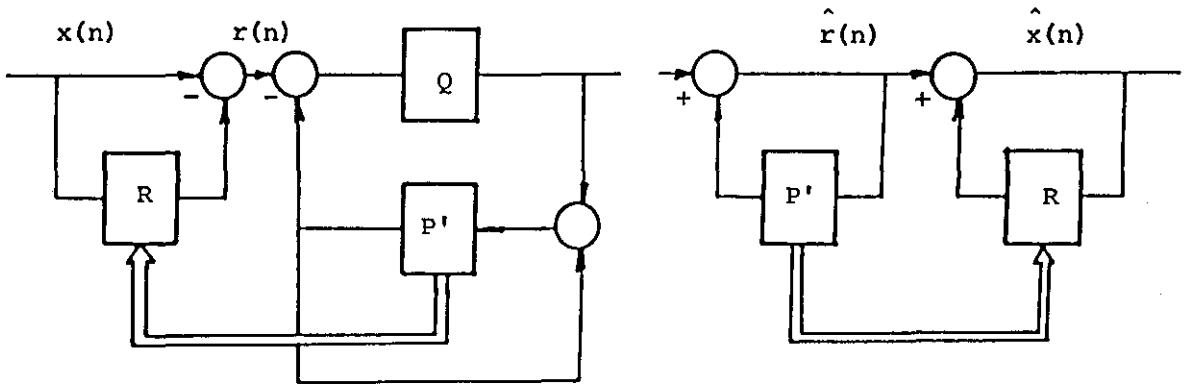


Fig. 4.13 Noise Shaping ADPCM Coder Using Backward Adaptive Pre- and Post-filtering

into the ADPCM coder is shown in figure 4.13. The filter R adapts according to:

$$R(z) = 1 - \frac{1 - P'(z)}{1 - P'(z/\alpha)} \qquad (4.24)$$

where P′ is the BBA predictor optimised from past samples of the pre-filtered signal {r(n)}. At the receiver, corresponding post-filtering is applied to the received r(n) to recover the input speech.

For this backward adaptive arrangement, the interaction among the various elements in the system is rather more complex, although the same general explanation as that for the forward adaptive case applies to a great extent. The effect of the pre-filter, whether forward or backward adaptive, is still the same i.e. to produce a smaller residual signal and hence a lower level of quantization noise. Once again, the spectral plots of the output noise provide much insight into the operation of the coder. From figure 4.14, it is apparent that the backward pre-filter arrangement produces certain desirable characteristics. Specifically, it is able, for the same noise shaping factor α=0.7, to provide a noise shape similar to NSB1 over the high frequency part of the spectrum, but it performs the task more efficiently, by not pushing up the low frequency noise to the same extent. Hence, for the same (tolerable) low frequency noise level, NSB2 will provide even more suppression of the high frequency distortion present, which would lead to an enhancement in the quality of the received speech. The noise spectral plot for NSB2 with α=0.2 (figure 4.14) illustrates this effect.

Listening tests confirmed the deduction from the output noise spectra. The quality of the recovered speech produced by NSB2 (with α=0.2) was found to be significantly superior to that of NSB1 (α=0.7). The total

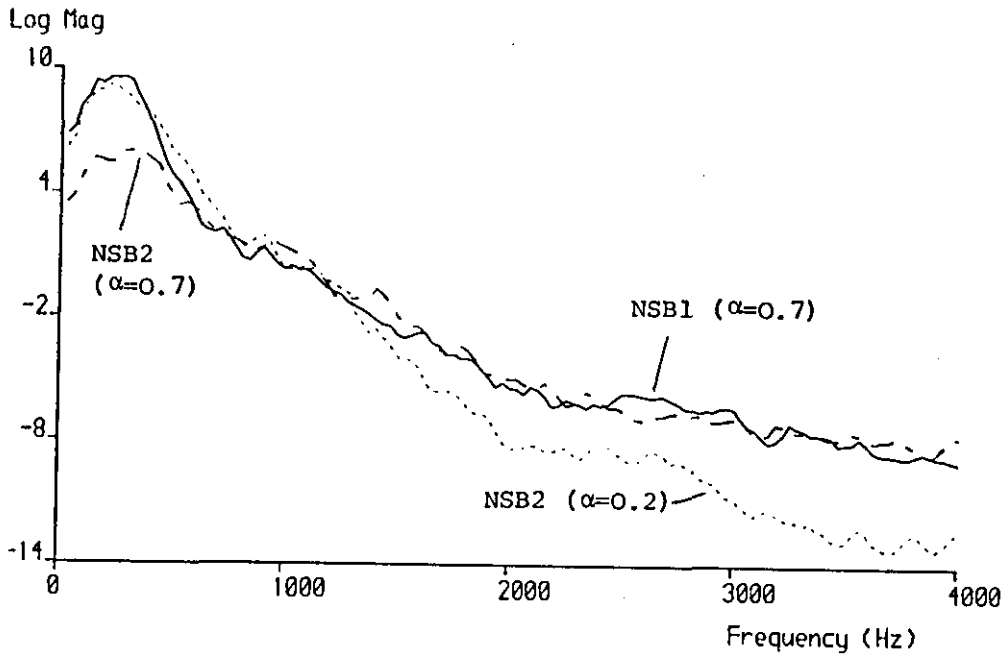


Fig. 4.14 Output Noise Spectra of Backward Adaptive Noise Shaping Schemes

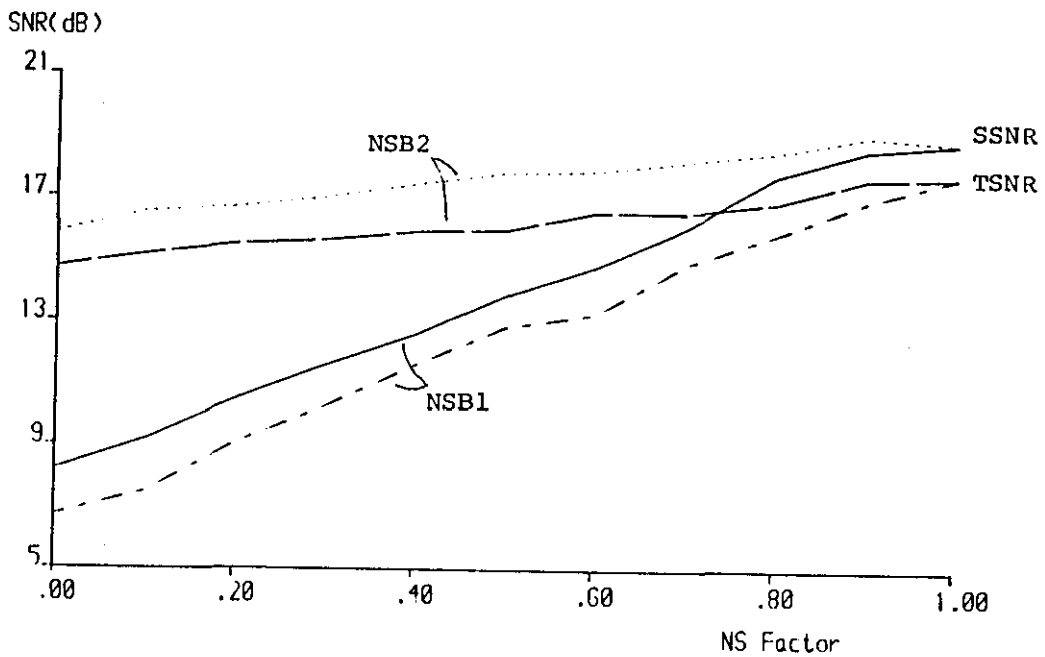


Fig. 4.15 Total and Segmental SNR Performance as a Function of Noise Shaping Factor for Schemes NSB1 and NSB2

and segmental SNR for various levels of noise shaping, for the two schemes are plotted in figure 4.15. Due to the relatively smaller low frequency noise level of NSB2, its SNR drops less rapidly as noise shaping is increased, compared to NSB1.

4.4.2 Subjective Listening Test

In order to obtain a more realistic assessment of the two proposed noise shaping coders, an informal subjective listening test involving a total of twenty five subjects was conducted. The recovered speech from the schemes NSB1 and NSB2 were compared to that obtained from 6 and 7 bit μ law log PCM[9] (denoted as PCM6 and PCM7), equivalent to bit rates of 48 and 56 Kbps respectively. Each of the four schemes was compared with every other scheme (except for PCM6 vs PCM7 for obvious reasons) in a randomly ordered A-B paired comparison test. The recovered speech from two schemes were presented to the subjects each time, and they were asked to respond with either a preference for one over the other or with no preference at all. Male and female sentences were separately tested. The results are summarised in table 4.2.

For male speech, there is undoubted preference (at least 80%) for both noise shaping schemes over PCM6. NSB1 is adjudged to be about the same as PCM7, while NSB2 is clearly superior to all the others. For female speech, the pattern is not as clear-cut - NSB1 is preferred to PCM6 but not to PCM7, while NSB2 is deemed slightly better than PCM7, and clearly superior to PCM6.

Table 4.2 Results of Subjective Listening Tests (in percentages)

Schemes A	Schemes B	MALE SPEECH pref A	MALE SPEECH pref B	MALE SPEECH No pref	FEMALE SPEECH pref A	FEMALE SPEECH pref B	FEMALE SPEECH No pref
PCM6	NSB1	0	92	8	20	36	44
PCM7	NSB1	12	20	68	40	12	48
PCM6	NSB2	0	100	0	0	72	28
PCM7	NSB2	0	80	20	40	52	8
NSB1	NSB2	4	80	16	4	72	24

Figure 4.16 provides a quick summary of the paired comparison test results (obtained from the average of the individual tests for male and female speech), and illustrates quite clearly, the overall superior quality provided by NSB2. Figure 4.17 shows the contour spectrograms of the recovered male speech sentences corresponding to each of the four schemes evaluated, together with that of the original unprocessed speech. It can be seen, by comparing with the original, that a considerable amount of additive noise is present in the high frequency region of the spectrum for the PCM schemes (note in particular the beginning of the sentence). This gives rise to a high frequency background 'hiss' in the speech, which although small in amplitude, is nevertheless perceptually annoying. In contrast, the noise shaping coders NSB1 and NSB2 are able to suppress this high frequency noise successfully and thus reduce significantly the background hiss. It is clear from the figure that the spectrogram corresponding to NSB2 provides the closest resemblance to the original, as would be expected from the results of the subjective test.

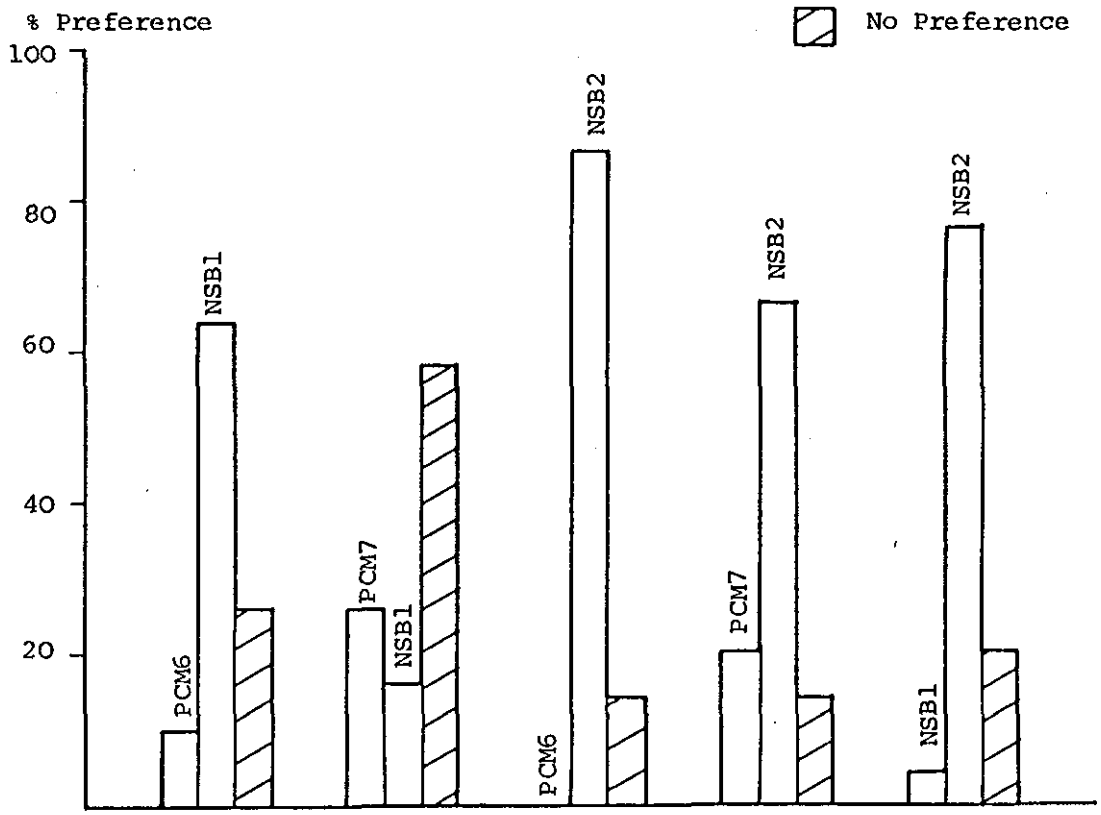


Fig. 4.16 Summary of Paired Comparison Test Results
(Subjective Listening Tests)

(a)

Original

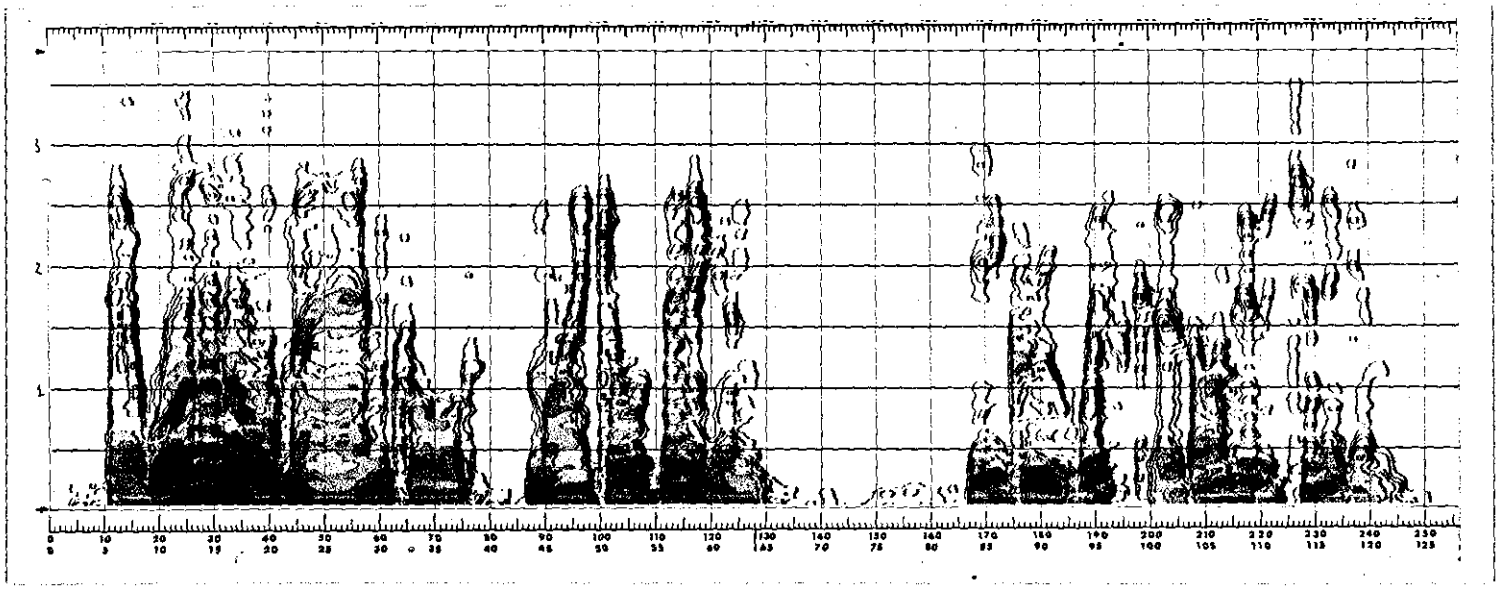
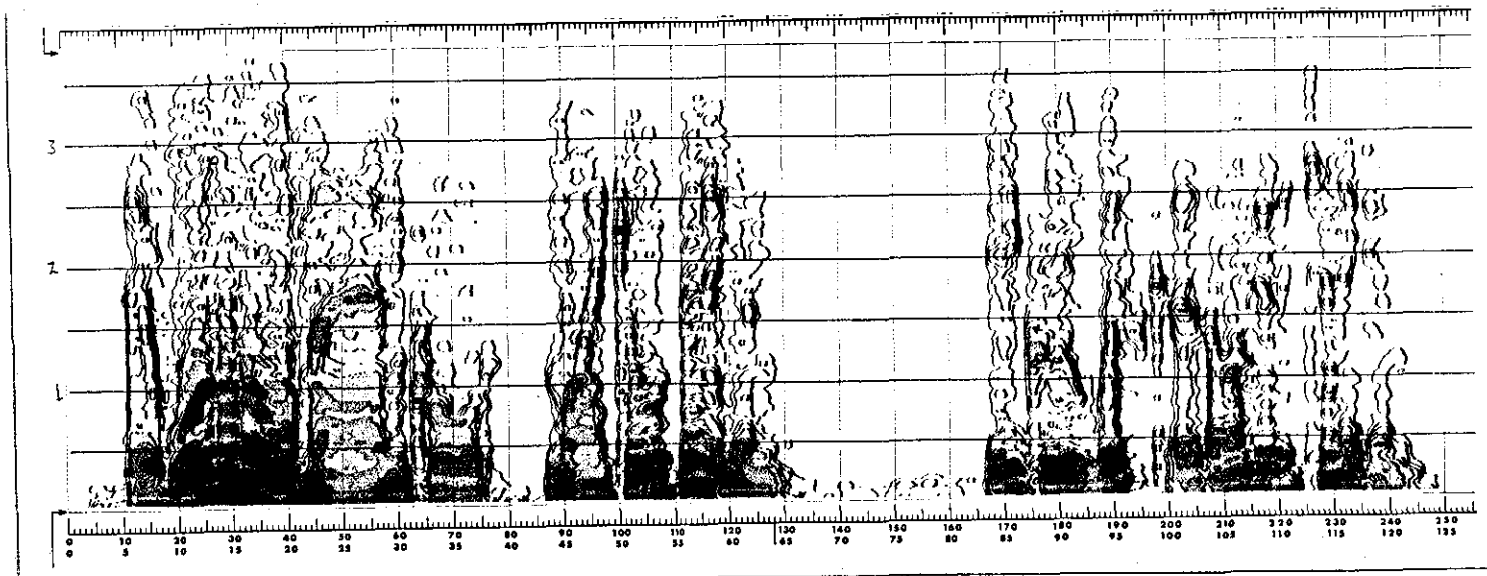


Fig. 4.17 Contour Spectrograms for the Utterance, " There was an old man called Michael Finnegan, He grew whiskers on his chinagen." (MALE)
(a) Original (b) PCM6 (c) PCM7 (d) NSB1 (e) NSB2

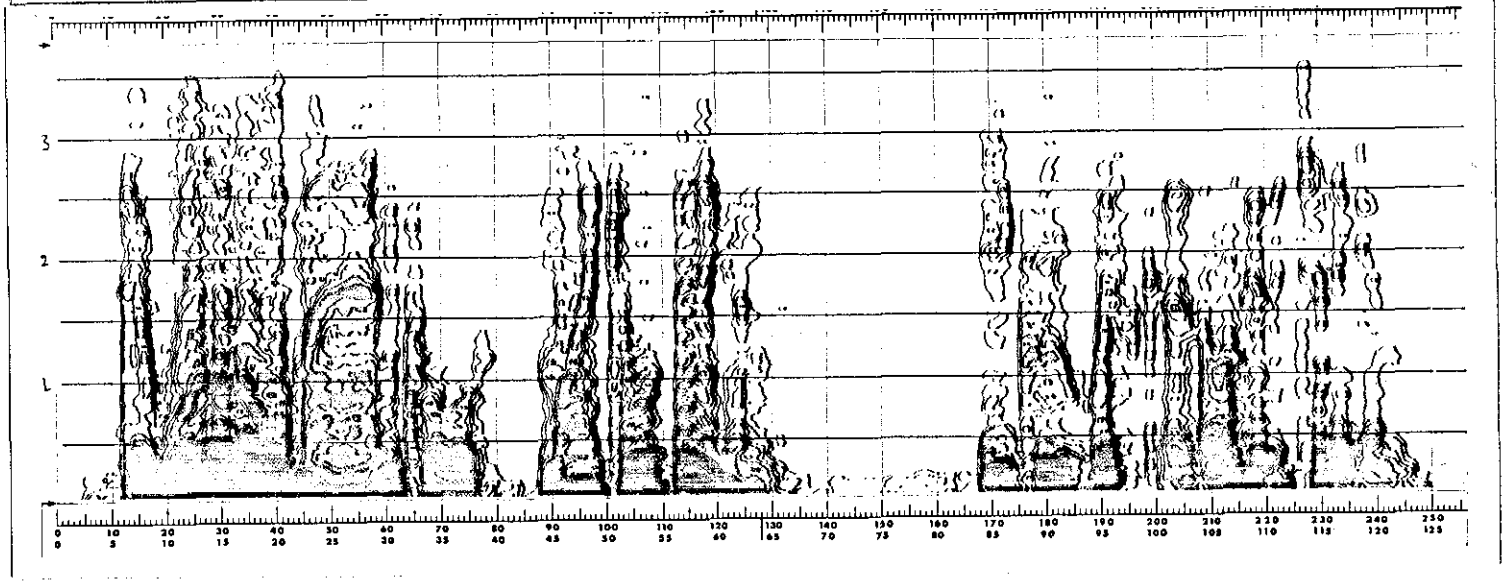
(b)

PCM6



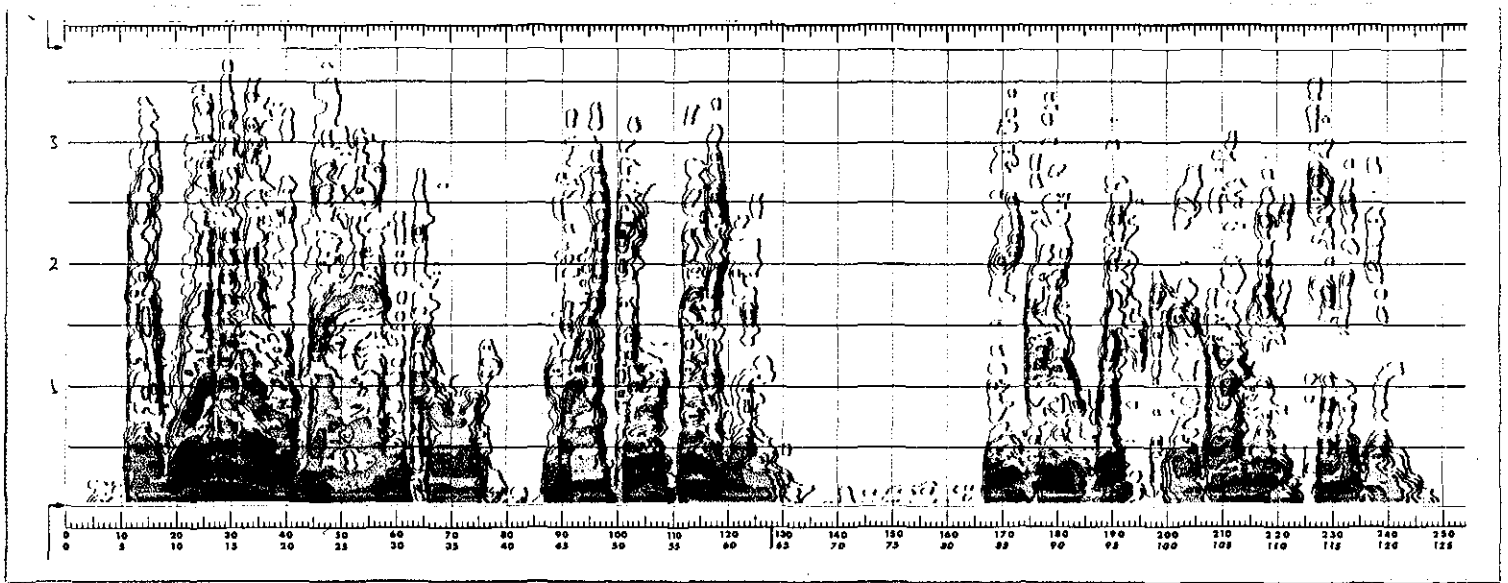
(c)

PCM7



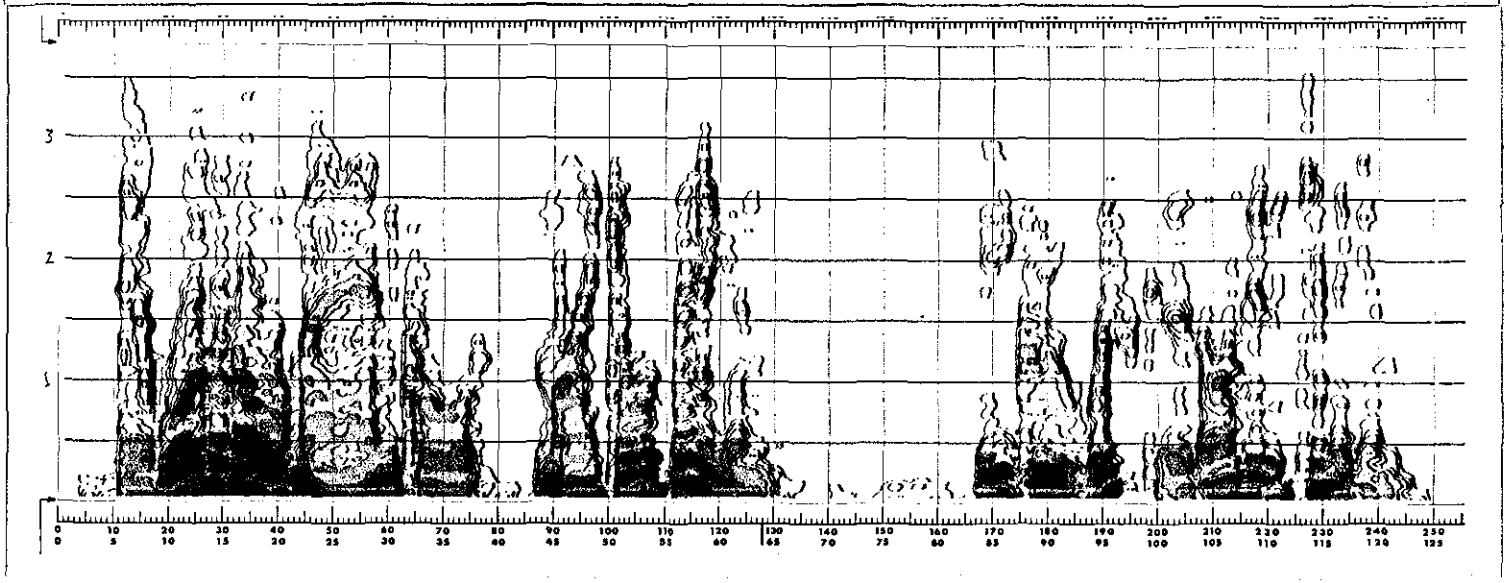
(d)

NSB1



(e)

NSB2



It is interesting to note the SNRs associated with the four coders evaluated. While the noise shaping coders have SNR values in the region of 15-17 dB (see figure 4.15), the corresponding SNR obtained using 6 and 7 bit log PCM coding are about 25 and 30 dB respectively - a difference of up to 15 dB! This demonstrates without doubt, the fallibility of using objective measurements such as signal-to-noise ratios as a means of assessing the performance of coders belonging to different classes, an observation noted by many researchers[12,19,82].

4.4.3 Note on Publications

A paper entitled, "16 Kbps ADPCM with Backward Noise Spectral Shaping" has been accepted for presentation at the Second International Conference on New Systems and Services in Telecommunications to be held in Liege, Belgium in November 1983. This paper is written in co-authorship with Dr. C.S. Xydeas and covers the work presented in section 4.4 of this chapter.

A more complete version of this paper, entitled, "Noise Shaping in Backward Adaptive ADPCM at 16 Kbps" which also covers the work on the backward block adaptive (BBA) predictor in section 3.4.2 has been submitted for publication to the IERE Proceedings. This paper is also written in co-authorship with Dr. C.S. Xydeas.

4.5 CONCLUSION

In this chapter, we have examined the application of noise spectral shaping to relatively simple ADPCM coders operating at a nominal bit rate of 16 Kbps. The aim has been to investigate the subjective

performance of such noise shaping ADPCM systems which do not involve a high level of complexity as those proposed by other researchers.

Forward adaptive noise shaping systems were first studied, and two methods of achieving noise spectral shaping were simulated, one based on the conventional noise feedback coder and the other utilising an adaptive pre- and post-filtering arrangement on the basic differential coder structure. The latter scheme was found to provide better SNR and subjective performance, due largely to the effect of predictor-quantizer interaction which works to advantage under the coarse quantization conditions considered. Unfortunately however, the better performance was achieved at the expense of a slight increase in side information and hence transmission bit rate. To avoid this additional side information requirement, a simple fixed pre-filter arrangement was considered, and this was found to provide a decoded speech quality a little worse than the adaptive case although comparable, if not slightly better than the more complicated conventional adaptive noise feedback coder.

The use of noise shaping techniques for improving the perceptual performance of differential coders has been demonstrated. With coarse quantization however, the level of noise present in the recovered speech can be a limiting factor to the effectiveness of such techniques. Various studies have indicated that, for effective auditory masking of noise, the noise power must be about 20 dB below the signal power at the same frequency[233]. Nevertheless, even when this condition is not satisfied (as in the case of coarse quantization), noise shaping can still be useful as a means of obtaining the optimum balance between low and high frequency distortion to produce the most subjectively pleasing

output[212].

Our simulations have shown that for a relatively 'un-sophisticated' 4th order adaptive prediction ADPCM scheme, the use of noise feedback to provide noise shaping is unwarranted, since equivalent, if not better performance can be obtained using a much simpler fixed pre-filter for the same purpose.

In an attempt to avoid the transmission of side information and the need for coding delay associated with 'look-ahead' forward adaptive strategies, the possibility of applying noise shaping in a backward manner was explored. Two schemes for achieving backward adaptive noise shaping, based on the configurations considered in the forward adaptive coders, were developed and evaluated. It was found that shaping of the output noise was again able to provide significant improvement in the output speech quality over the unshaped case. The backward adaptive pre-filter scheme in particular, was able to exploit predictor-quantizer interaction efficiently, to produce an impressive decoded speech quality comparable to that obtained using 7-bit log PCM coding[213,215].

In the work described hitherto, little attention is paid to the quantizer, in order not to detract from the main theme of the respective chapters. However, having discussed the prediction and noise shaping aspects of differential coding schemes in sufficient detail, the time has now come to consider the intricacies of adaptive quantization, which is crucial to the efficient performance of speech coding schemes. This will form the subject matter for the next chapter.

CHAPTER FIVE ADAPTIVE QUANTIZATION

5.1 INTRODUCTION

The performance of differential coders (DPCM, ADPCM, APC) is determined by two factors - prediction and quantization. The predictor attempts to reduce the variance of the input signal by removing redundancies present in its waveform while the quantizer seeks to represent the resultant prediction residual in terms of discrete amplitudes, with minimum distortion subject to the constraint on the number of levels it can employ for this purpose. Provided that the noise introduced by the quantization process does not affect the prediction, the final SNR of such coding schemes is therefore governed by the general equation[12],

$$SNR = SNR_p + SNR_q \qquad (5.1)$$

where SNR_p depends on the estimation accuracy of the prediction process employed and is sometimes referred to as the signal-to-noise ratio improvement (SNRI)[19]. SNR_q is the SNR produced by the quantization of the residual signal.

For efficient performance, both predictor and quantizer are normally required to be adaptive. In practice, adaptive prediction is not a critical requirement when the transmission bit rate is sufficiently high (above 32 Kbps). Adaptive quantization however, is rather more crucial to system performance. Noll[47] has shown that a DPCM system employing adaptive quantization and fixed prediction produces a massive 7 dB advantage over logarithmic PCM. When adaptive prediction is used, this

advantage is increased by a further 3-4 dB, with an additional 2-3 dB improvement possible with entropy coding. These results were obtained using 3-bit quantization.

This chapter is concerned with the quantization aspect of speech coding schemes, and in particular, with those schemes operating at a transmission bit rate of about 16 Kbps. At this bit rate, and with the input speech sampled at 8 kHz, only 2 bits are allowed for the quantization of each transmitted signal sample (assuming no signal decimation or entropy coding) so that some form of quantizer adaptation is a virtual necessity. The adaptive quantization methods used for the ADPCM systems in chapters 3 and 4 are examined in greater detail in this chapter. Other quantizer adaptation techniques are also considered. Following this, a simple novel approach to reducing quantization noise in ADPCM systems is proposed and described. This is evaluated using computer simulation on the 2-bit one-word memory backward adaptive quantizer[49].

5.2 ADAPTIVE QUANTIZATION TECHNIQUES

The basic function of the quantizer is to assign to each input sample, one of a set of several discrete magnitude levels, which is closest to the input sample. In a B bit uniform quantizer, the number of these discrete amplitude levels is 2^B. Hence, the quantization error power is proportional to the square of the quantizer step-size i.e. the distance between adjacent amplitude levels. In typical voice communications systems, the dynamic range of speech signals considering inter-talker as well as intra-talker variations can be as much as 40 dB[37]. Early attempts to accommodate this large signal dynamic range has been in the form of time-invariant non-uniform quantizers, with fine quantizer steps

in the small amplitude region and much coarser steps for large amplitudes (see section 2.4.1.1.(a))[9,42,44]. Logarithmic PCM[9,12,37], which is still being used in many communication networks, is one such non-uniform quantization technique. Other methods have sought to match the quantizer input-output levels to the input signal's statistics[43], and various quantizers optimised for signals with Gaussian, Laplacian and gamma distributions[43,45] have been designed. Such time-invariant techniques however, fail to recognise that the large dynamic range of speech signals is the result of a non-stationary or time-varying process, and these methods are therefore only optimal for a specific input signal power. Better results can be obtained using a quantization strategy that is variable in time i.e. with a characteristic that adapts to the input signal level.

Adaptive quantization[47,49-53] utilises a quantizer characteristic that shrinks or expands in time like an accordion, depending on the input signal power. Typically, speech power levels vary sufficiently slowly in time to allow simple adaptation strategies to be designed to track these variations. In differential coding schemes, the quantizer input is the prediction residual which has a much reduced dynamic range compared to the corresponding speech signal. Nevertheless, the power variation is still considerable, and adaptive quantization is no less desirable[60, 64]. Adaptive quantizers may be either forward or backward adaptive, depending on whether adaptation is based on the input samples or on the quantized output, respectively. These two main classes shall be considered separately in the following.

5.2.1 Forward Adaptive Quantization (AQF)

Forward adaptive quantizers (AQF)[19,41,46-48] normally adapt its step-size on a block basis. A block of N input samples is buffered and the average energy of the signal samples within the block is obtained. This value determines the step-size Δ, which is then used to quantize the same block of samples. Thus,

$$\Delta = \alpha\left[\frac{1}{N}\sum_{j=0}^{N-1} x^2(n-j)\right]^{\frac{1}{2}} \qquad (5.2)$$

where α is an appropriate constant weighting factor which depends on the number of bits used in the quantizer. By using the actual quantizer input to obtain the step-size, this method ensures that the quantizer range is always matched to the signal. If it is required that the quantizer characteristics be designed for a particular signal distribution, (5.2) can be used to estimate the standard deviation of the block of samples. α would obviously be different in this case.

For ADPCM applications, quantization of the prediction residual is performed on a sample-by-sample basis, so that it is not possible to buffer a block of residual signal samples for the purpose of calculating the optimum quantizer step-size. In this case, some form of step-size estimation will have to be made. Figure 5.1 shows an ADPCM codec employing forward adaptive quantization. A block of N input samples is buffered, the feed-forward prediction residual is formed, and an estimate of the step-size for the block is made, based on the average residual energy. Several methods for calculating the step-size for DPCM and ADPCM systems have been suggested.

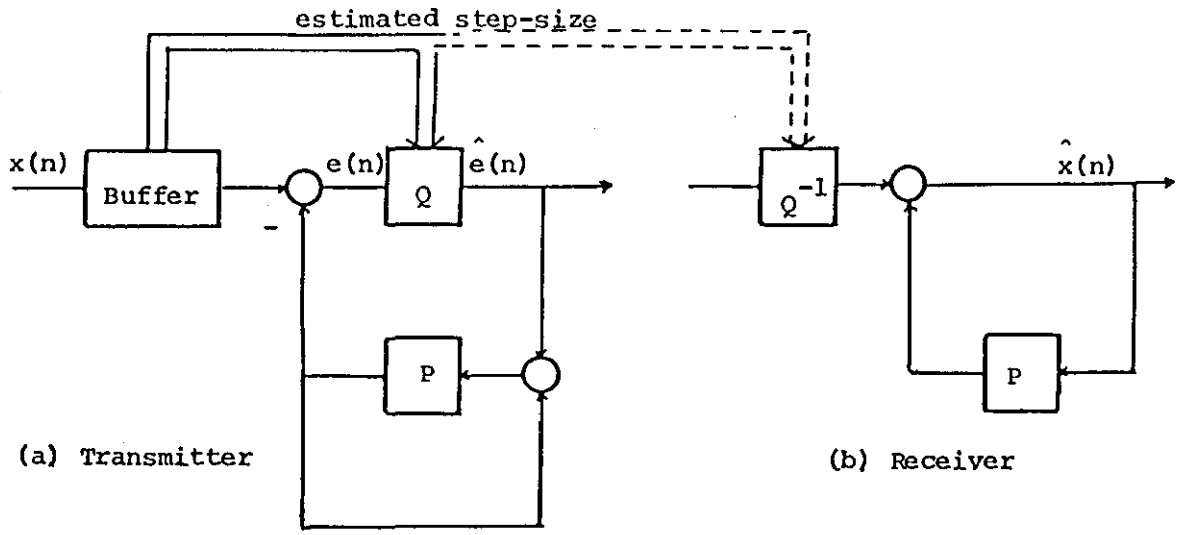


Fig. 5.1 ADPCM-AQF Codec

Noll[47] proposed using the maximum difference between adjacent samples within a block to determine the optimum step-size Δ, for a single-tap DPCM system:

$$\Delta = \alpha \text{Max}\{ |x(n-j) - x(n-j-1)| \} \qquad (5.3)$$
$$j = 0,1,2....N-2$$

where α is an optimising parameter. Jayant[46] presented a similar formula for estimating the AQF step-size which uses the average forward prediction error:

$$\Delta = \alpha \frac{1}{N-1} \sum_{j=0}^{N-2} |x(n-j) - a_1 x(n-j-1)| \qquad (5.4)$$

where

$\alpha = 0.50$ for B = 3

$= 0.25$ for B = 4

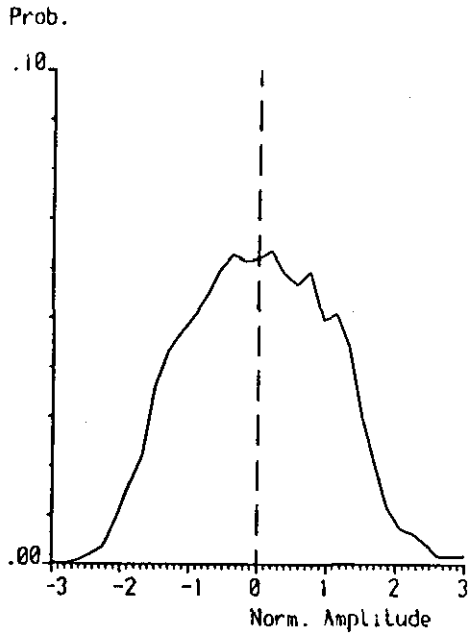
a_1 is the first order predictor coefficient and B is the number of bits per sample employed by the quantizer. For higher order predictors, Noll [47] suggested using,

$$\Delta = \alpha \sqrt{\sum_{j=1}^{N-p-1} \{x(n-j) - \sum_{k=1}^{p} a_k x(n-j-k)\}^2} \qquad (5.5)$$

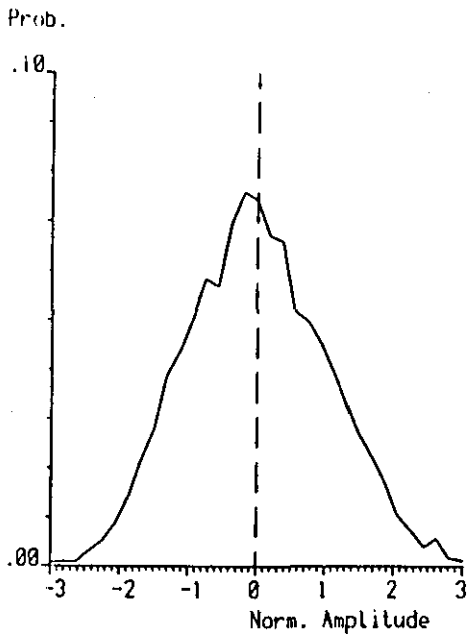
where α is a parameter and the $\{a_k, k=1,2,...p\}$ are the p predictor coefficients. Optimum quantizers may also be employed in DPCM coding, in which case the block of N error samples are first normalised by the estimated standard deviation of the block before being quantized by a unit variance optimum quantizer. The standard deviation of the error samples estimated from the input signal is normally modified by a weighting factor greater than unity, to account for the presence of additive quantization noise in the actual error signal that is quantized[41].

Figure 5.2 shows the probability density functions (pdf) of two seconds of male and female speech. The short-term pdfs (figure 5.2(a)) were obtained by averaging over all normalised short-time pdfs, taken in blocks of 64 samples (8 ms). These are very much Gaussian when the blocksize used is small and tends toward Laplacian as the blocksize increases. The long-term pdfs (figure 5.2(b)) obtained from the full two seconds of speech are also shown. These are undoubtedly gamma distributed, due to the presence of proportionately greater amounts of low amplitude components in a typical speech utterance (including pauses and silence). Figure 5.3 shows the similar pdfs of the speech residual, obtained using second order feed-forward adaptive prediction on the same speech data. Clearly, the distribution of the residual signal is not very much different from that of the original speech.

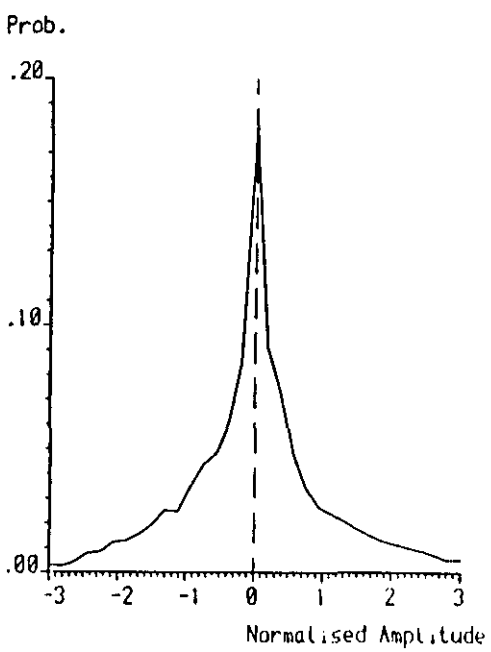
From these observations of the pdfs, the use of quantizers optimised for specific distributions can be expected to yield better performance for



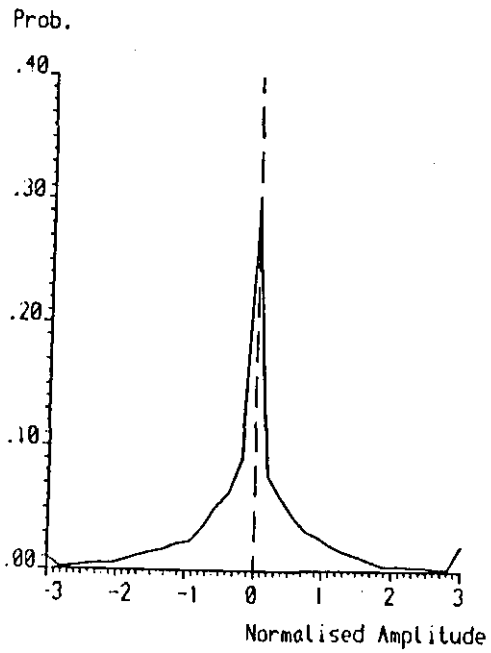
(a) Short term pdf (MALE)



(b) Short term pdf (FEMALE)

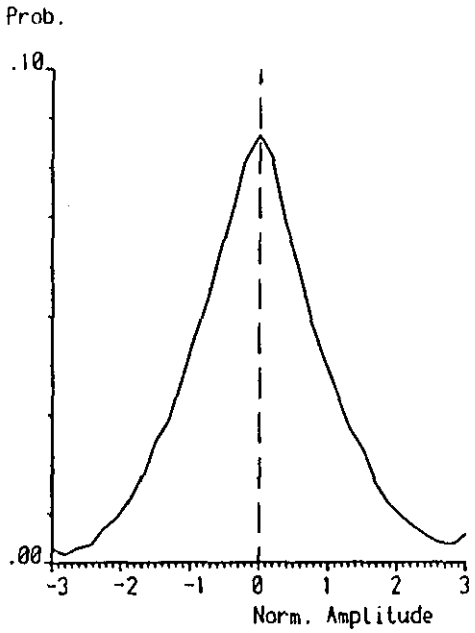


(c) Long term Pdf (MALE)

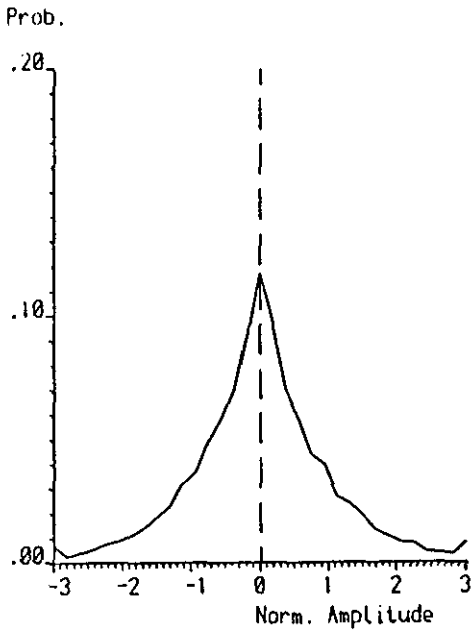


(d) Long term Pdf (FEMALE)

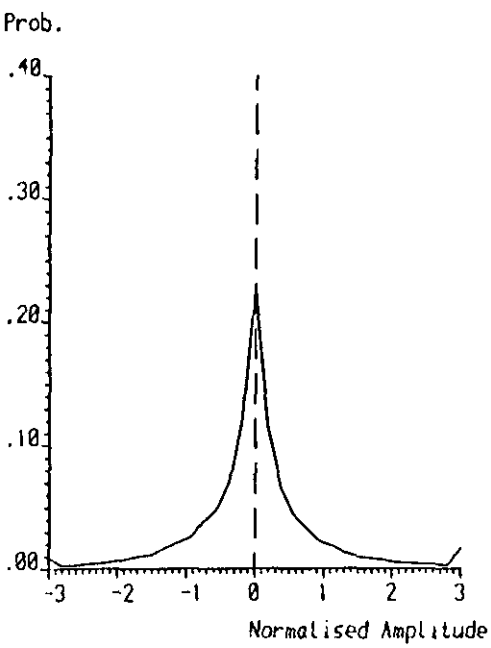
Fig. 5.2 Normalised Pdfs of Speech Signals



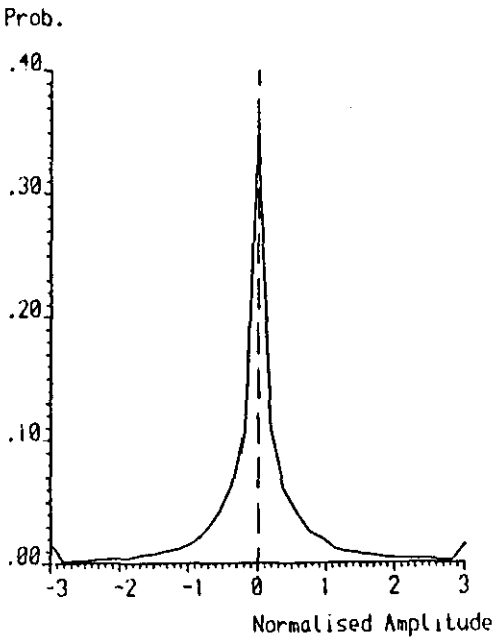
(a) Short term pdf (MALE)



(b) Short term Pdf (FEMALE)



(c) Long term Pdf (MALE)



(d) Long term Pdf (FEMALE)

Fig. 5.3 Normalised Pdfs of Speech Residual Signals

both APCM and ADPCM coding using AQF. Higher SNR has indeed been reported by Noll[47], for the use of such 'optimum' quantizers. For 2-bit quantization however, the advantage of these over uniform quantization is slight (half a dB or less on average).

The 'look-ahead' facility of forward adaptive quantizers necessitate a delay in the system, since a block of input samples has to be buffered in order to estimate the step-size. In addition, this step-size estimate needs also to be communicated to the receiver, thus requiring additional channel capacity. (In practice, it is the quantized version of the step-size that is used at both transmitter and receiver to ensure identical operation.) The delay and side information requirement which might be undesirable or unacceptable in certain applications may be avoided if quantizer adaptation is made to proceed in a backward mode, based on past output samples, which are available at both transmitter and receiver.

5.2.2 Backward Adaptive Quantization (AQB)

The attraction of backward adaptive quantizers[12,20,37,49,50,64] as noted above, lies in their ability to operate without delay or side information. Essentially, the adaptation involves some form of 'prediction' of the incoming signal power which is used to update the quantizer step-size. Since no prior information about signal energy is available, adaptation must be made based on the most recently decoded samples at a given time instant in order to maximise prediction accuracy. Consequently, backward adaptive quantizers usually vary their step-sizes instantaneously, at every sampling instant, as opposed to the block adaptation of AQF. Several backward quantizer adaptation

algorithms will now be considered.

5.2.2.1 Jayant's Adaptive Quantizer (AQJ)

Undoubtedly the most well-known and widely used backward adaptive quantizer is the instantaneous one-word-memory algorithm developed by Cummiskey, Flanagan and Jayant, and commonly referred to as the Jayant's quantizer (AQJ)[49,64]. This provides a simple means of matching the step-size of the quantizer to its input, using quantizer memory. Specifically, if the outputs of a uniform B-bit quantizer are of the form,

$$\hat{x}(n) = H(n)\frac{\Delta(n)}{2} \quad ; |H(n)| = 1,3,.....2^B-1 \quad (5.6)$$
$$\Delta(n) > 0$$

the step-size $\Delta(n+1)$ is given by the previous step-size multiplied by a time-invariant function of the code-word magnitude $|H(n)|$; i.e.

$$\Delta(n+1) = \Delta(n).M(|H(n)|) \quad (5.7)$$

where M(.) denotes the multiplier function. By this means, the quantizer seeks to expand or contract its amplitude range according to the variance of the incoming input samples. Figure 5.4 illustrates the input-output characteristics of a 3-bit Jayant quantizer. Note that the number of multiplier values is given by 2^{B-1}. Since adaptations follow quantizer output rather than input, step-size information in this scheme need not be explicitly communicated but, in the case of error-free transmission, can be re-created exactly by the receiver.

Alternatively, this adaptive quantizer can be viewed as one which normalises the input samples $x(n)$ with a state variable $u(n)$ and uses a fixed range quantizer to quantize the result, as shown in figure 5.5.

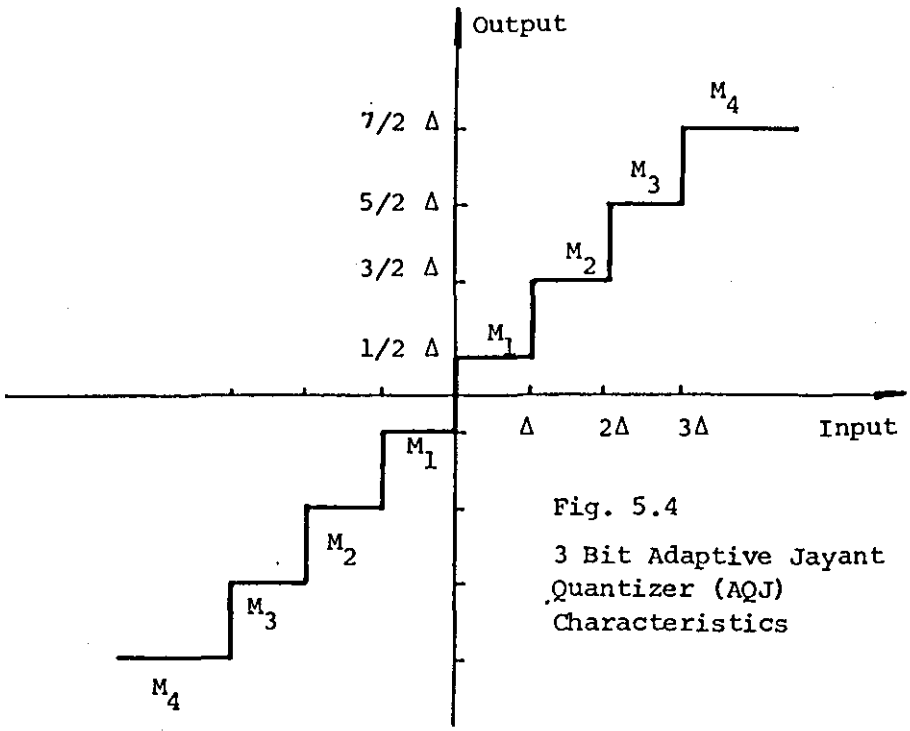


Fig. 5.4
3 Bit Adaptive Jayant Quantizer (AQJ) Characteristics

It can be seen that the state variable u(n) is updated in the same way as Δ(n) of (5.7), being the product of its previous value and the multiplier associated with the previous quantizer slot occupied i.e.

$$u(n) = u(n-1).M(|H(n-1)|) \qquad (5.8)$$

The reverse process i.e. multiplication by u(n) is performed to obtain the decoded sample.

Notice in figure 5.4, that the step-size adaptation is based only on the magnitude of the latest decoded output and not on its sign. This is a consequence of the observation that the probability density function of speech signals is expected to be symmetrical about a mean value of zero. Table 5.1 shows the recommended multiplier values provided by Jayant, for B = 2,3 and 4 bit quantizers for both PCM and DPCM coders. These recommended multipliers do not in general constitute overly critical target values. It is important however, that step-size increases should

be more rapid than step-size decreases. This has to do with the following comparison of two basic types of quantization errors: overload errors, which occurs when the step-size is too small and the signal sample falls outside the quantizer range, and granular errors that are inherent in quantization, even when the quantizer range is adequate. Granular errors are bounded by the step-size and are therefore relatively smaller in magnitude compared to overload errors. As a result, they also tend to be less harmful to SNR.

Table 5.1 Step-size Multipliers for the One-Word Memory Quantizer

CODER	PCM			DPCM		
B	2	3	4	2	3	4
M1	0.6	0.85	0.8	0.8	0.9	0.9
M2	2.2	1.00	0.8	1.6	0.9	0.9
M3		1.0	0.8		1.25	0.9
M4		1.5	0.8		1.75	0.9
M5			1.2			1.2
M6			1.6			1.6
M7			2.0			2.0
M8			2.4			2.4

Although the one-word memory quantizer performs well in ideal channels, the sequential adaptation it employs renders it extremely susceptible to transmission errors. A robust version of this quantizer, proposed by Goodman[190], modifies the step-size adaptation algorithm of equation (5.7) to incorporate a leakage factor β,

$$\Delta(n+1) = \Delta^{\beta}(n).M(|H(n)|) \qquad ;0 < \beta < 1 \qquad (5.9)$$

β is normally just less than unity (for example, 63/64) and controls the rate at which the effects of transmission errors are dissipated. This modification improves the quantizer's error performance considerably, at the cost of slightly reduced efficiency.

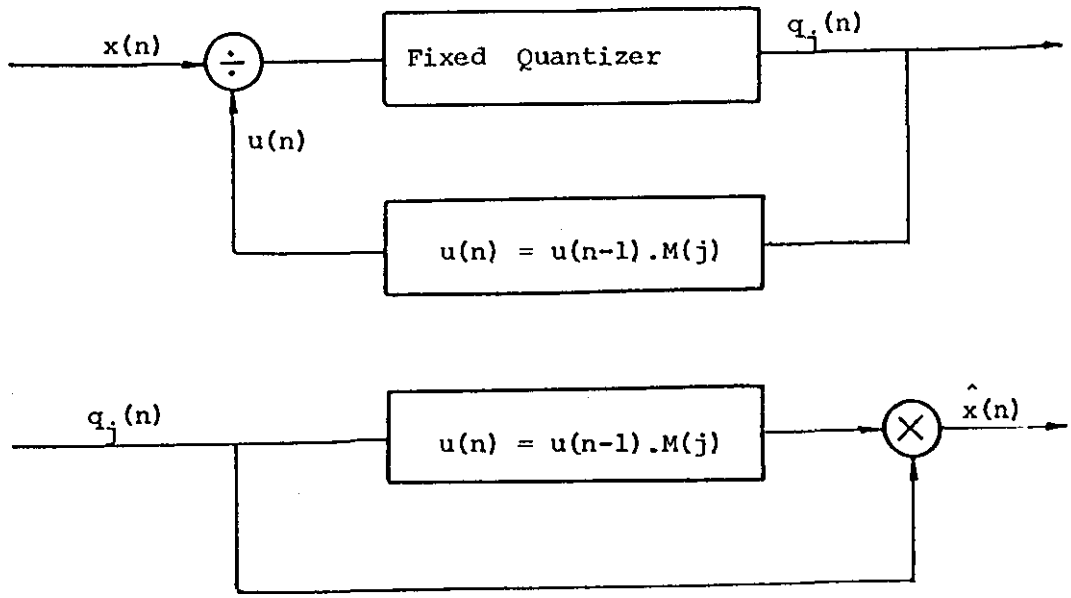


Fig. 5.5 Block Diagram of Jayant's Quantizer

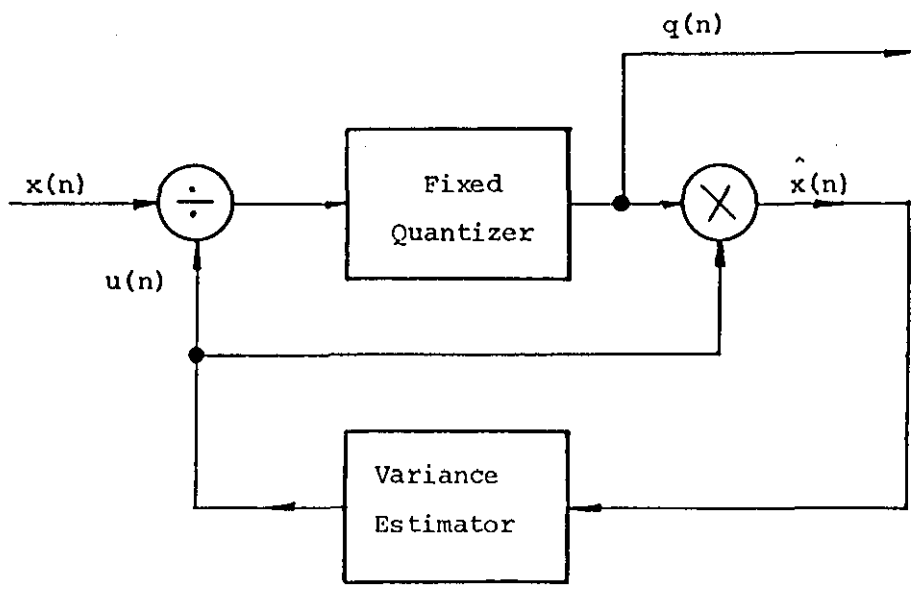


Fig. 5.6 The Variance Estimating Adaptive Quantizer

The AQJ has been used extensively in DPCM and ADPCM systems for the quantization of the prediction residual signal. Notice the slight but important difference in the optimum multiplier values used for DPCM coding (see table 5.1). This has to do with the fact that while high adjacent sample correlation exists at the input of a PCM quantizer (this being equal to the correlation between Nyquist sampled speech), the same is not true for DPCM quantizer inputs, due to the differentiating (or high-passing) process involved. DPCM quantizer inputs are generally much less correlated and thus step-size increases must be even more rapid than step-size decreases[37,49].

Goodman[189] conducted a theoretical study of the AQJ and considers its performance in terms of such factors as its range fluctuation, adaptation speed and the stability of the process. He showed that the sequence of quantizer ranges is a stochastically stable process, and that the steady state fluctuation of the normalising factor u(n) is related to a function of the maximum and minimum multiplier values by,

$$R = \log_2 \frac{\text{Max } M(.)}{\text{Min } M(.)} \qquad (5.10)$$

The adaptation response is then inversely related to R. Thus with appropriate choice of the multiplier values, the optimal trade-off between adaptation speed and accurate steady-state performance for a particular application may be obtained.

5.2.2.2 Variance Estimating Quantizer (VEQ)

A backward quantizer adaptation strategy similar to Jayant's algorithm is the variance estimating quantizer (VEQ) studied by Noll[20], Stroh

[41] and Castelino[50]. The VEQ, shown in figure 5.6, normalises the input signal by the square root of a maximum likelihood estimate of its variance and quantizes the resulting ratio using a fixed quantizer. The normalising variable u(n) is made proportional to a moving estimate of the decoded signal's standard deviation in order to obtain a unit variance signal which can then be optimally quantized. Thus, u(n) is given by,

$$u^2(n) = \alpha^2 \frac{1}{N} \sum_{j=1}^{N} \hat{x}^2(n-j) \tag{5.11}$$

where α is an optimising constant. An exponential average of previous quantizer outputs have also been used. This is of the form,

$$u^2(n) = \alpha^2 \sum_{j=1}^{\infty} (1-\gamma)\gamma^{j-1} \hat{x}^2(n-j) \tag{5.12}$$

where the effective memory of the variance estimator varies by changing the value of the leakage constant γ. The introduction of γ weights each decoded sample into the past, attaching more weight to the more recent samples and gradually 'forgetting' distant samples. The formulation of (5.12) can be expressed in recursive form as,

$$u(n) = \left[\alpha^2(1-\gamma)\ \hat{x}^2(n-1) + \gamma u^2(n-1)\right]^{\frac{1}{2}} \tag{5.13}$$

From figure 5.6, it can be seen that,

$$\hat{x}(n-1) = u(n-1).q(n-1) \tag{5.14}$$

Substituting into (5.13) gives,

$$u(n) = u(n-1)\ [\ \alpha^2(1-\gamma)\ q^2(n-1) + \gamma\ ]^{\frac{1}{2}} \tag{5.15}$$

Clearly, (5.15) is the same as the Jayant adaptation of the normalising factor given in (5.8) if:

$$M(|H(n-1)|) = [\ \alpha^2(1-\gamma)\ q^2(n-1) + \gamma\ ]^{\frac{1}{2}} \qquad (5.16)$$

and consequently, the variance estimating quantizer is equivalent to Jayant's quantizer.

5.2.2.3 Pitch Compensating Quantizer (PCQ)

The lack of a 'look-ahead' facility in the AQJ algorithm renders it rather susceptible to overload during the occurrence of sudden transitions in the input signal. This is particularly so when the quantizer is used in differential coding structures where its shortcomings are manifested in the clipping of the high amplitude residual samples related to the speech excitation or pitch pulses. Such 'clipping' can produce annoying 'clicks' in the decoded speech and a reduction in SNR . A fast adaptation response to avoid overload is possible with suitable choice of the multiplier values, but this will, on the other hand, increase the granular noise during the low amplitude segments of the signal. The obvious solution is to have some form of variable adaptation algorithm which is able to increase the quantizer step-size rapidly upon detection of overload, without sacrificing performance during the less rapidly varying segments of the signal. At least two such quantizers which attempts to incorporate some compensation for the pitch pulses, have been proposed.

The first pitch compensating quantizer (PCQ) proposed by Cohn and Melsa [66], uses two adaptive u(n) estimators simultaneously. One is an envelope estimator (denoted $u_e(n)$) which computes a moving average of the magnitudes of previous quantized samples. The other, $u_j(n)$ is a Jayant's estimator with non-uniform quantization levels and specially selected multiplier values. These multipliers are all less than unity

except for the two outermost levels which are set at values much higher than normal. For example, in a 5-level quantizer, the multipliers are given as: M(1) = 0.4, M(2) = 0.8 and M(3) = 2.2. The quantizer characteristic is shown in figure 5.7.

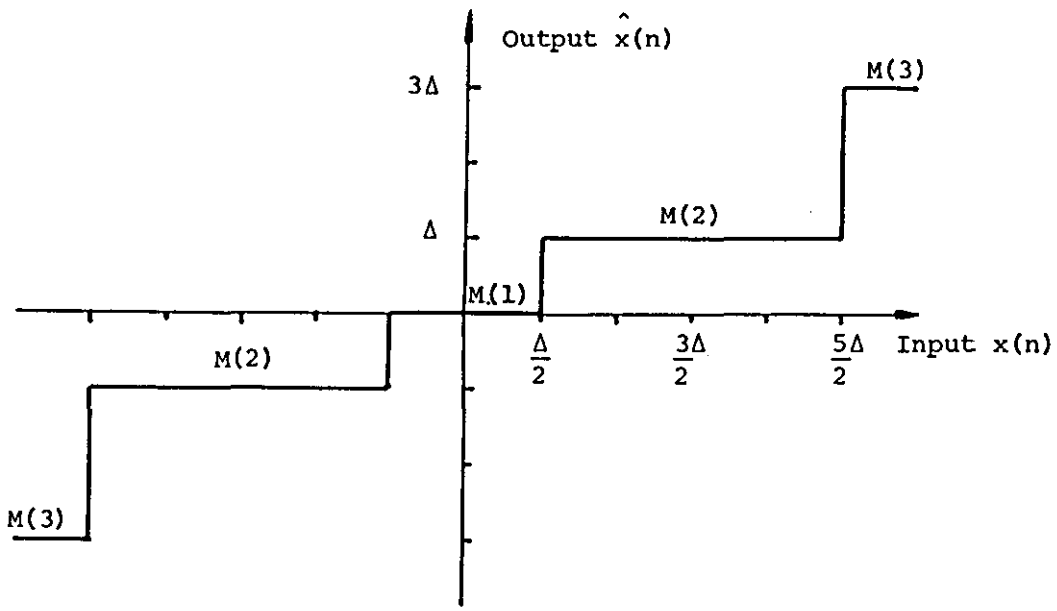


Fig. 5.7 Characteristics of Pitch Compensating Quantizer (PCQ)

Both u_e and u_j are updated at each time instant and the actual normalising factor u(n) used is the larger of the two, i.e.

$$u(n) = \text{Max}[\, u_e(n), u_j(n) \,] \qquad (5.17)$$

When signals with slowly varying amplitudes are being quantized, u_j assumes small values because only the multipliers less than unity are being used. In such cases, u_e provides a more accurate estimation of the signal variance and is taken as the normalising factor. When the quantizer detects a possible pitch pulse with one of its outermost levels however, u_j increases rapidly due to the high multiplier value associated with the outermost levels, and becomes greater than u_e.

Consequently, the step-size increases quickly to 'capture' the high amplitude sample(s). After the pitch pulses have been quantized, u_j decays just as quickly so that the envelope estimator takes over again.

The second pitch compensating quantizer developed by Qureshi and Forney [67] employs two Jayant estimators, one for tracking syllabic variations of the input signal and the other for providing large values of u(n) upon detection of possible pitch pulses, by using high values for the outermost levels as before. The quantization strategy is similar to the first method except that the envelope estimator is substituted with a Jayant's estimator whose multipliers are set close to unity so that its output follows the long-term syllabic variations of the input signal. The adaptation process is best understood by considering logarithms. Defining $U(n) = \log_2 u(n)$, Qureshi's PCQ adapts according to,

$$U(n) = U_1(n) + U_2(n) + U_{min} \tag{5.18}$$

where U_{min} is a constant and defines the minimum value of U(n). $U_1(n)$ is related to the normalising factor of the first Jayant estimator (pitch compensator) and is updated according to:

$$U_1(n) = \gamma_1 U_1(n-1) + M_1(n-1) \tag{5.19}$$

where M_1 is a set of multipliers which are all zeroes except for the value which corresponds to the outermost levels of the quantizer. γ_1 is a leakage constant less than unity which causes $U_1(n)$ to decay exponentially after the occurrence of the outermost quantization level. $U_2(n)$ is related to the second Jayant estimator and is similarly defined as,

$$U_2(n) = \gamma_2 U_2(n-1) + M_2(n-1) \tag{5.20}$$

where again, γ_2 is a leakage factor and M_2 is a set of coefficients which are close to zero except for the outermost levels. The quantizer

gain, or normalising factor is finally given by,

$$u(n) = \text{Int}\left[\, 2^{\text{Int}[U(n)]}(1 + U(n) - |U(n)|) \,\right] \qquad (5.21)$$

where Int[.] means "the integer part of".

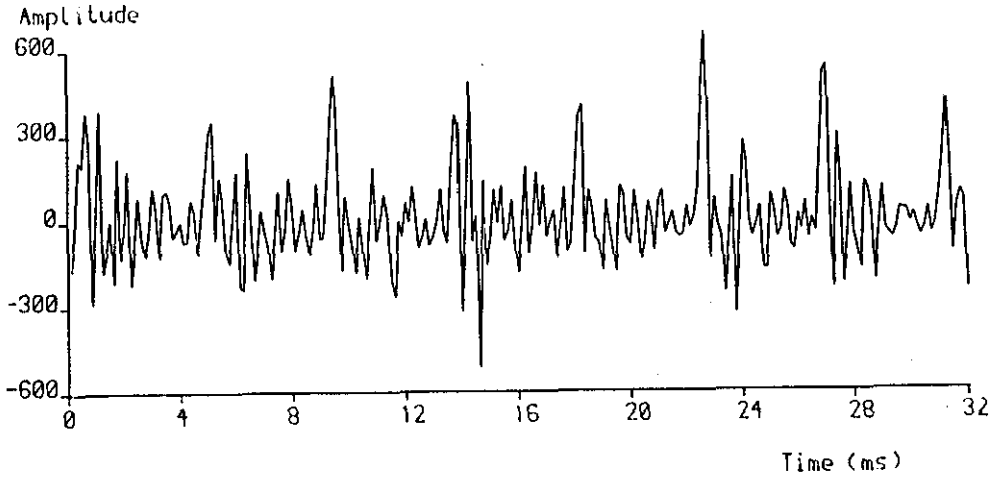
5.2.3 Discussion

The use of pitch compensating methods such as those described above to improve on the performance of AQJ has certainly led to a reduction in clipping errors in the quantization of the prediction residual, with a consequent increase in SNR over the un-compensated case. Unfortunately however, the techniques proposed require variable rate coding, with its attendant problems of delay, synchronisation and buffer management (see section 2.6.5). In many applications, the difficulties associated with variable rate coding would usually outweigh any advantages over fixed rate coding that could be expected.

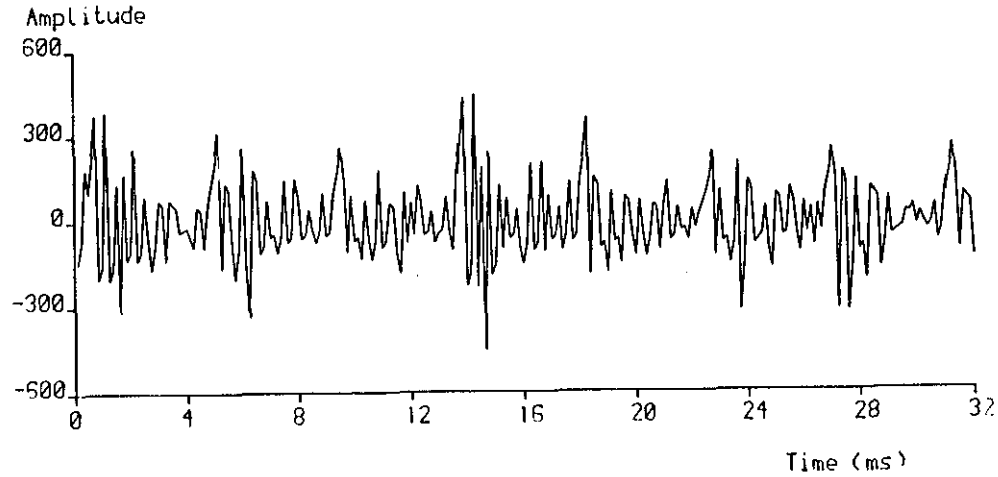
In the following sections, we describe a new approach to the problem of quantizer compensation, which do not attempt to modify the basic AQJ algorithm in any way. Instead, correction is made to the decoded speech samples at the receiver only, based on simple statistical measurements.

5.3 QUANTIZER CORRECTION

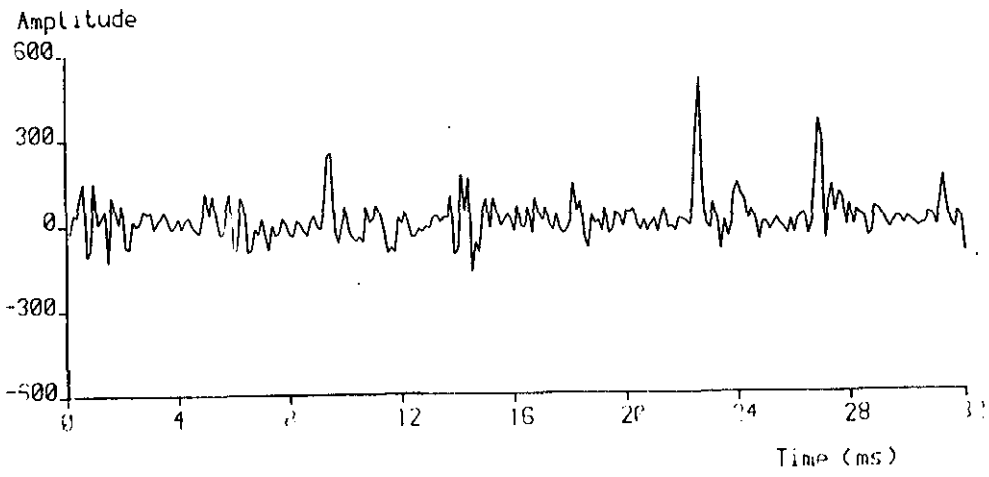
The work on quantizer correction has arisen out of our efforts to seek improved quantizer performance for the 2-bit Jayant quantizer in the context of DPCM or ADPCM coding. With only 2-bits (4 levels) assigned to code each signal sample, quantization accuracy is obviously limited and 'clipping' of the residual signal frequently occurs. Figure 5.8



(a) Typical Prediction Residual



(b) Quantized Version of (a) Using 2-bit AQJ



(c) Quantization Error

Fig. 5.8 Illustration of Quantizer Clipping

illustrates this effect. The first plot, labelled (a) shows a typical DPCM residual signal (obtained using second order fixed prediction) with the distinct high amplitude periodic excitation pulses. Figure 5.8(b) shows the quantized version of the same signal obtained with 2-bit AQJ, which clearly demonstrates the clipping of the pitch pulses. Apart from this clipping effect, it can also be seen that the quantizer output tends to decay too slowly following the occurrence of a large output.

We decided to investigate if these limitations of the quantizer can be corrected without attempting to modify the basic DPCM coder configuration, and without requiring any additional information to be communicated to the receiver. The last constraint implies that all necessary information must be obtained from the quantizer output bit stream.

5.3.1 Correction Technique

Consider a DPCM coder (figure 5.9) where $x(n)$, $\hat{x}(n)$, $e(n)$ and $\hat{e}(n)$ denote the input speech, decoded speech, the quantizer input (prediction residual) and output, respectively. The proposed quantizer correction technique is based on observing the quantizer output sequence $\{\hat{e}(n)\}$ in small blocks at a time and then applying appropriate correction, based on these observations, to the corresponding decoded speech sequence $\{\hat{x}(n)\}$. The amount of correction to be applied depends on the distribution of the block of $\hat{e}(n)$ samples and these can be obtained from long-term statistics. Figure 5.10 shows how the correction is applied at the DPCM decoder. The quantized output sequence is extracted from the transmitted bit stream and fed to the box labelled COR where correction is made to the appropriate decoded speech samples. This

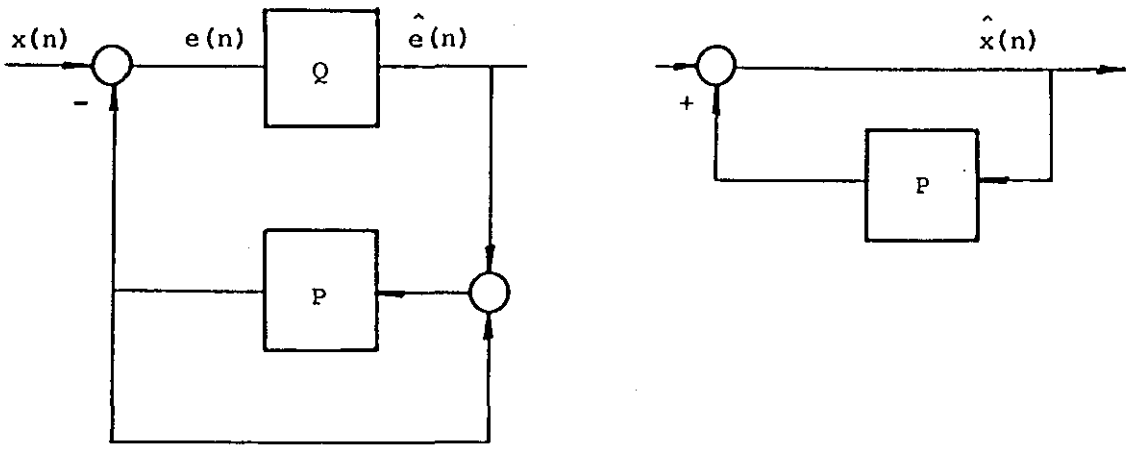


Fig. 5.9 DPCM Coder

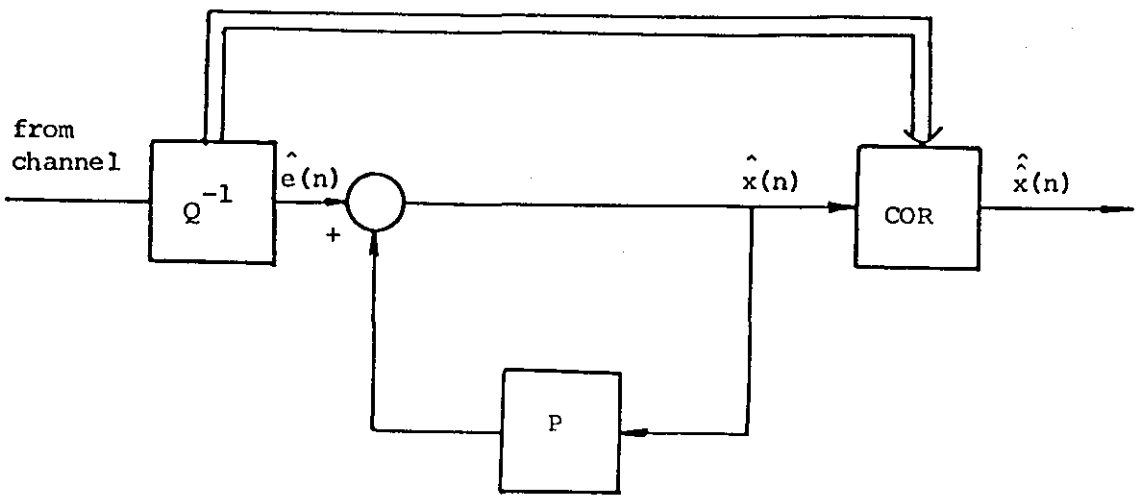


Fig. 5.10 DPCM Receiver with Quantizer Correction

quantizer correction procedure will now be described[214].

Consider the quantizer input-output relation at the transmitter in terms of small blocks of 3 contiguous samples. The quantization process introduces an error given by $e(n)-\hat{e}(n)$, which, in the absence of transmission errors, is equal to the output noise $x(n)-\hat{x}(n)$. If the polarity of this error is known, some form of correction can be applied to $\hat{x}(n)$ to provide a reduction in the output noise. A correction factor $f_i(n)$ for the ith block can be defined as,

$$f_i(n) = \frac{e_i(n) - \hat{e}_i(n)}{\hat{e}_i(n)} \qquad ; n = 1,2,3 \qquad (5.22)$$

where n denotes the position of the sample within the block. Adjacent blocks slide forward by one sample each time instant to give an overlap of 2 samples between blocks. When $f_i(n) > 0$, it implies that $|\hat{e}_i(n)| < |e_i(n)|$ i.e. the magnitude of the quantized value is smaller than the actual sample. An appropriate correction to increase the magnitude of the corresponding decoded signal $\hat{x}_i(n)$ would thus lead to lower noise for this particular sample. In the same way, the $f_i(n) < 0$ condition indicates that a decrease in the magnitude of $\hat{x}_i(n)$ is desirable. The decoded samples can be therefore corrected according to:

$$\hat{\hat{x}}_i(n) = \hat{x}_i(n) + \bar{f}_i(n)\,\hat{e}_i(n) \qquad (5.23)$$

where $\hat{\hat{x}}_i(n)$ is the corrected sample and $\bar{f}_i(n)$ represents a fixed correction, optimised from long term characteristics. Obviously, a correction using $f_i(n)$ itself (from (5.22)) would lead to zero noise in the decoded speech.

For a 2-bit (4 level) quantizer, one bit is required to carry the sign information, leaving only one bit for the magnitude. We shall denote the lower and upper magnitude levels by 1 and 2 respectively.

Each of the 3 sample groups is identified according to its magnitude sequence. Of the 8 possible sequences, only the following four symmetrical patterns were considered in the correction process.

(a) 2 2 2 (b) 1 1 1 (c) 1 2 1 (d) 2 1 2

For each of these patterns, a further classification into 4 possible sub-groups is performed, depending on the sequence of the signs i.e.

sequence 1 : + + + or - - -

sequence 2 : + + - or - - +

sequence 3 : + - - or - + +

sequence 4 : + - + or - + -

This grouping of the sign sequences follows from the symmetrical properties of the quantizer input about the zero axis.

This analysis was performed separately on all the input speech data files using 4 different prediction techniques (all 2nd order) on the ADPCM coder. These are:

(1) Fixed prediction - with the predictor coefficients obtained from the long-term autocorrelation of speech (see section 3.2)[62].

(2) Forward block adaptive (FBA) prediction - using a blocksize of 256 samples (see section 3.3.1)[41].

(3) Backward sequentially adaptive prediction - using the SAP algorithm (section 3.3.2)[75].

(4) Backward block adaptive (BBA) prediction - with the predictor coefficients updated every 32 samples using a blocksize of 256 samples

(section 3.4.2.1)[213,215].

In each case, the speech data was coded using the respective ADPCM coder with 2-bit AQJ and the statistics of $f_i(n)$ (from (5.22)) were obtained. The percentage of occurrence of each pattern, as well as the probability distribution of each factor $f_i(n)$ were noted. Table 5.2 shows an example of the analysis performed for the case of fixed prediction ADPCM, and provides the following information:

(1) The percentage of occurrence of each sign sequence (1,2,3 and 4) related to each magnitude sequence a,b,c and d.

(2) The statistics of the 3 correction factors f(n) associated with each sign sequence.

(3) Other useful information regarding the probability distribution of each f(n), such as its mean value, the average of its positive values and the average of its negative values. The variance of f(n) is also indicated by the cumulative percentage entries, which gives the proportion of f(n) greater or less than a certain value.

Looking at table 5.2(a) i.e. the statistics of pattern (a), it can be seen that the application of a positive correction to the decoded output block at the receiver corresponding to pattern(a) sequence 1 would result in lower noise more than 90% of the time for the first two samples and about 70% for the third sample in the block. This particular combination corresponds to the magnitude sequence 222 with all samples of the same polarity, and indicates quite strongly the occurrence of pitch pulses. As the quantizer requires a few sampling instants to respond to these high amplitude samples, much of the 'clipping' occurs on the rising edges of the transition. Hence, the correction factors f(n) associated with the first 2 samples of this particular output sequence are very largely positive. In obtaining

	Sequence 1			Sequence 2			Sequence 3			Sequence 4		
	1	2	3	1	2	3	1	2	3	1	2	3
Total		467			48			22			13	
%		84.91			8.73			4.00			2.36	
%>0	90.4	96.6	61.5	100.0	39.6	27.1	27.3	81.8	86.4	61.5	23.1	7.7
%<0	9.6	3.4	38.5	0.0	60.4	72.9	72.7	18.2	13.6	38.5	76.9	92.3
%>3	3.0	6.6	3.2	4.2	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
%>2	6.9	13.9	6.6	16.7	2.1	0.0	0.0	0.0	9.1	0.0	0.0	0.0
%>1	31.7	39.0	18.8	47.9	4.2	2.1	0.0	0.0	22.7	7.7	0.0	0.0
%>0.5	64.2	67.9	33.2	91.7	14.6	2.1	0.0	45.5	50.0	23.1	7.7	0.0
%<-.2	3.2	0.6	22.5	0.0	33.3	39.6	40.9	4.5	9.1	7.7	38.5	69.2
%<-.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
%<-.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
%<-.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Av>0	0.93	1.18	0.95	1.25	0.52	0.30	0.09	0.50	0.76	0.46	0.20	0.00
Av<0	-0.17	-0.14	-0.20	0.00	-0.20	-0.18	-0.21	-0.13	-0.17	-0.12	-0.17	-0.23
Mean	0.82	1.14	0.51	1.25	0.09	-0.05	-0.13	0.38	0.64	0.24	-0.08	-0.21

(a) Pattern (a) 222

	Sequence 1			Sequence 2			Sequence 3			Sequence 4		
	1	2	3	1	2	3	1	2	3	1	2	3
Total		1353			855			1102			1258	
%		29.62			18.72			24.12			27.54	
%>0	39.6	38.5	52.5	63.2	25.3	38.5	21.3	74.0	43.8	18.6	35.9	49.7
%<0	60.4	61.5	47.5	36.8	74.7	61.5	78.7	26.0	56.2	81.4	64.1	50.3
%>3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
%>2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
%>1	0.1	0.2	1.1	0.6	0.5	0.9	0.1	1.4	0.6	0.2	0.2	1.2
%>0.5	14.0	14.0	25.7	23.6	6.5	19.1	5.5	42.3	18.3	2.9	9.1	23.6
%<-.2	46.9	50.8	38.7	26.2	65.8	52.4	67.3	19.0	46.6	69.4	50.1	41.8
%<-.4	31.9	37.5	28.5	14.6	53.3	41.5	50.1	12.0	34.4	53.4	35.9	30.6
%<-.6	19.0	22.8	18.9	7.1	38.1	30.3	30.5	8.1	22.5	35.6	21.5	20.1
%<-.8	7.2	11.1	9.9	2.9	20.2	15.6	13.8	4.3	11.8	19.3	9.6	9.9
Av>0	0.40	0.40	0.49	0.40	0.34	0.48	0.32	0.53	0.43	0.25	0.34	0.48
Av<0	-0.44	-0.50	-0.50	-0.37	-0.57	-0.55	-0.51	-0.43	-0.51	-0.53	-0.46	-0.50
Mean	-0.11	-0.15	0.02	0.12	-0.34	-0.15	-0.33	0.28	-0.10	-0.39	-0.18	-0.02

(b) Pattern (b) 111

Table 5.2 Statistics of Correction Factors for 2nd Order Fixed Prediction ADPCM

	Sequence 1			Sequence 2			Sequence 3			Sequence 4		
	1	2	3	1	2	3	1	2	3	1	2	3
Total		1176			532			721			834	
%		36.04			16.30			22.10			25.56	
%>0	79.1	45.2	42.6	87.6	17.9	24.4	11.2	61.6	36.9	4.7	15.0	27.2
%<0	20.9	54.8	57.4	12.4	82.1	75.6	88.8	38.4	63.1	95.3	85.0	72.8
%>3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
%>2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
%>1	1.1	0.3	0.8	1.5	0.2	0.2	0.0	0.6	0.0	0.2	0.0	0.0
%>0.5	44.8	2.4	17.7	58.8	1.9	7.5	2.6	5.1	13.0	1.3	1.3	7.7
%<-.2	12.2	19.9	47.5	8.1	51.5	67.3	83.4	11.1	54.6	91.7	48.3	62.8
%<-.4	6.6	0.0	34.6	4.5	0.0	55.8	71.8	0.0	43.3	82.3	0.0	48.8
%<-.6	4.1	0.0	23.0	3.2	0.0	41.7	58.3	0.0	31.2	63.5	0.0	35.3
%<-.8	2.0	0.0	10.6	2.6	0.0	25.2	35.0	0.0	17.1	38.7	0.0	16.8
Av>0	0.53	0.18	0.44	0.62	0.19	0.37	0.32	0.21	0.39	0.32	0.17	0.35
Av<0	-0.33	-0.16	-0.50	-0.41	-0.22	-0.60	-0.66	-0.14	-0.56	-0.68	-0.20	-0.55
Mean	0.35	-0.01	-0.10	0.49	-0.14	-0.37	-0.55	0.08	-0.21	-0.63	-0.15	-0.30

(c) Pattern (c) 121

	Sequence 1			Sequence 2			Sequence 3			Sequence 4		
	1	2	3	1	2	3	1	2	3	1	2	3
Total		705			235			177			552	
%		42.24			14.08			10.61			33.07	
%>0	49.8	87.9	42.3	77.9	17.0	47.2	46.3	74.6	51.4	8.2	1.1	33.9
%<0	50.2	12.1	57.7	22.1	83.0	52.8	53.7	25.4	48.6	91.8	98.9	66.1
%>3	0.0	0.0	0.0	0.4	0.0	0.4	0.6	0.0	0.6	0.0	0.0	0.0
%>2	0.0	0.0	0.1	1.7	0.0	0.9	1.1	0.0	0.6	0.0	0.0	0.0
%>1	0.3	2.7	1.4	6.4	0.0	3.0	5.6	0.0	5.1	0.2	0.2	0.5
%>0.5	1.8	52.1	6.7	20.9	6.0	13.2	13.0	45.2	18.1	0.9	0.7	2.9
%<-.2	18.0	6.5	27.4	6.4	77.4	28.1	27.1	20.9	28.8	61.6	97.5	33.5
%<-.4	0.0	2.7	0.0	0.0	63.8	0.0	0.0	13.0	0.0	0.0	93.1	0.0
%<-.6	0.0	2.1	0.0	0.0	54.9	0.0	0.0	8.5	0.0	0.0	77.0	0.0
%<-.8	0.0	1.3	0.0	0.0	33.2	0.0	0.0	4.0	0.0	0.0	48.7	0.0
Av>0	0.19	0.55	0.29	0.41	0.35	0.43	0.46	0.56	0.44	0.19	0.68	0.20
Av<0	-0.16	-0.32	-0.18	-0.13	-0.65	-0.20	-0.18	-0.47	-0.21	-0.22	-0.74	-0.19
Mean	0.01	0.45	0.02	0.29	-0.48	0.10	0.11	0.30	0.13	-0.19	-0.73	-0.06

(d) Pattern (d) 212

Table 5.2 Statistics of Correction Factors

	Pattern(a)			Pattern(b)			Pattern(c)			Pattern(d)		
Factors	1	2	3	1	2	3	1	2	3	1	2	3
Sequence 1	0.8	1.1	0.4	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.4	0.0
Sequence 2	1.4	0.0	0.0	0.0	0.0	0.0	0.3	-.2	-.2	0.5	-.2	0.0
Sequence 3	0.0	0.0	0.0	-.4	0.4	0.0	-.6	0.0	-.3	0.0	0.4	0.0
Sequence 4	0.0	0.0	0.0	-.3	-.2	0.0	-.6	-.2	-.2	-.2	-.7	0.0

Table 5.3 Optimised Correction Factors

these correction factors, one is clearly exploiting certain specific and fairly well defined properties which are peculiar to the speech residual signal. The actual amount of correction for each individual sample is determined experimentally.

It was observed that the long-term statistics for the factors f(n) exhibit very little variations among the different speech files used and the various prediction algorithms employed. It is possible therefore, to obtain an universal set of optimised correction factors based on all data files and averaged over the prediction schemes considered. This set of optimised correction factors is shown in table 5.3. The same analysis was also performed on the 4 non-symmetrical magnitude sequences, 112, 122, 211 and 221. It was found however, that the f(n)'s associated with these sequences were very much less well-defined, and little advantage results from using these factors.

The underlying assumption in associating specific correction factors with particular quantizer output sequences as done above is that certain redundancy or predictability still remains in the speech residual signal and these appear to be quite independent of the type of prediction employed. Obviously, it would not be possible to obtain any sensible relationship when the signal to be quantized is a random signal. Consequently, in the analysis performed to obtain the f(n) statistics, blocks containing low amplitude random noise (which are actually silence segments) must be excluded from consideration. A simple silence detector was used for this purpose. This consists of measuring the average signal energy in blocks of 20 samples and comparing it to a threshold value. When the average signal energy falls below this threshold, the block is deemed to be silence, and is not considered in

the analysis. This simple procedure was found to be quite effective in eliminating unwanted contributions from segments of silence in the speech signals.

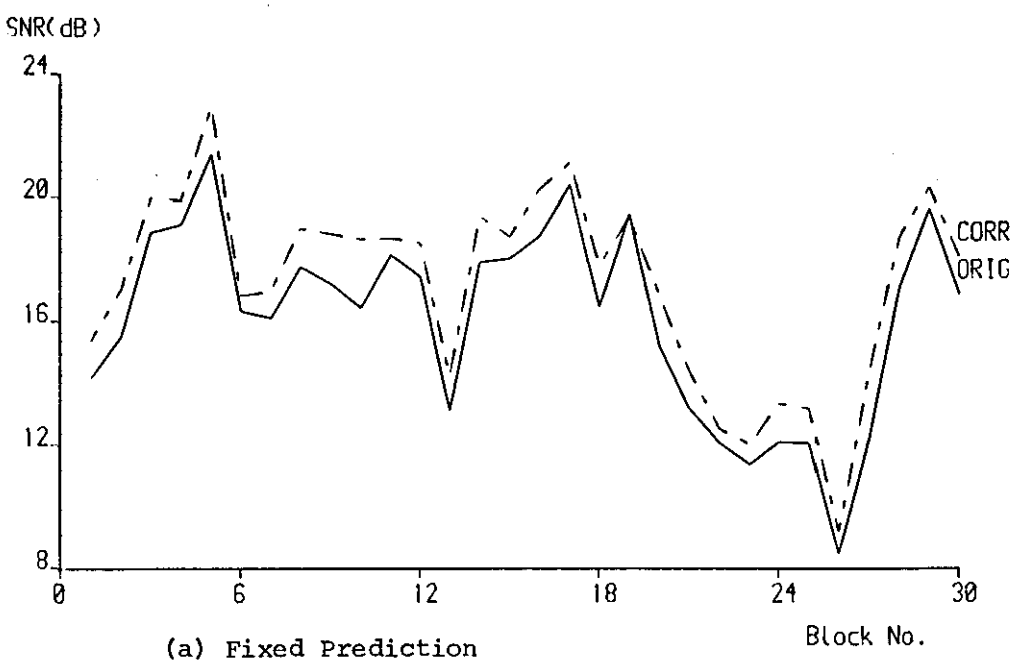
5.3.2 Computer Simulation Results

The technique of quantizer correction using the factors given in table 5.3 was applied to the ADPCM systems considered. In all cases, an improvement in performance was recorded. Table 5.4 shows the total and segmental SNR obtained for each case before and after the application of correction, obtained from 2 seconds of speech from each data file.

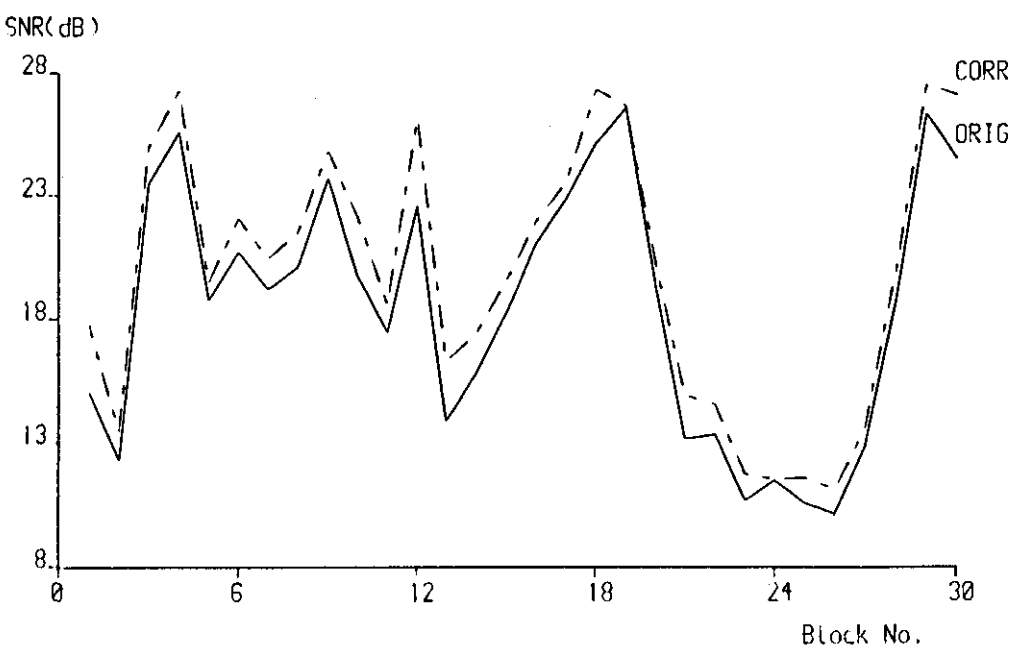
Table 5.4 SNR Results for Various Second-order ADPCM Systems (With and Without Quantizer Correction)

	MALE		FEMALE		SISTER	
Predictor Used	SSNR	TSNR	SSNR	TSNR	SSNR	TSNR
FIXED						
(a) Original	16.02	16.49	15.26	15.01	14.26	17.30
(b) Corrected	17.03	17.65	16.34	16.30	15.85	18.48
FORWARD (FBA)						
(a) Original	19.05	18.19	18.89	16.97	15.61	17.01
(b) Corrected	20.03	19.38	19.96	18.24	17.08	18.43
SAP						
(a) Original	18.77	17.54	18.04	16.33	13.88	12.54
(b) Corrected	19.91	18.86	19.26	17.61	14.58	13.91
BBA						
(a) Original	18.53	17.68	17.94	16.39	15.50	16.53
(b) Corrected	19.78	19.17	19.05	17.65	16.28	18.02

The average improvement in SNR is between 1.25 to 1.5 dB, but this does not reflect the actual performance of individual blocks, since those blocks which are not strongly periodic do not register very much gain. Figure 5.11 shows the segmental SNR for 1 second of male speech, before



(a) Fixed Prediction



(b) BBA Prediction

Fig. 5.11 Segmental SNR for ADPCM Systems Before and After Applying Quantizer Correction

and after applying quantizer correction, obtained for the fixed and the BBA prediction ADPCM coder. It can be seen that increases in SNR of as much as 2 dB or more can be achieved in some segments of the signal. This improvement in segmental SNR is also reflected in the corresponding output noise spectra plots shown in figure 5.12. In addition to the reduced noise power level across the frequency spectrum, considerable high frequency noise suppression is also achieved by the quantizer correction process. The same observations were obtained for the other prediction schemes and for all the data files considered. More importantly, informal listening tests conducted indicate a decided preference for the corrected speech over the normal ADPCM decoded speech. The background hiss characteristic of ADPCM systems at this low bit rate, although still audible, is perceptibly reduced after correction[214].

5.3.3 *Note on Publication*

A paper entitled, "Noise Reduction in ADPCM AQJ Systems Using Quantizer Correction at the Receiver" has been published in the IEE Electronics Letters, vol. 19, no. 11, pp. 420-421, May 1983. It was written in co-authorship with Dr. C.S. Xydeas and covers the work described in section 5.3 of this chapter.

5.4 SUMMARY AND CONCLUSION

It has long been recognised that in speech digitisation schemes, the quantizer plays a central role in determining system performance[47]. Consequently, much early study has concentrated on the efficient design of the quantizer, in attempts to match the quantizer

(a) Fixed Prediction

(b) BBA Prediction

Fig. 5.12 Output Noise Spectra for ADPCM Systems Before and After Applying Quantizer Correction

characteristics with either assumed or measured probability densities of speech. More recent designs take into account the short-term stationarity of speech signals and the importance of adapting the range of the quantizer to match the local signal strength of its input[41,48]. Such adaptive quantizers have provided significantly improved performance, both objectively and subjectively.

In this chapter, we have examined both classes of adaptive quantizers:

(i) the forward adaptive scheme - where the adaptation decision is based on the unquantized input data and communicated to the receiver as side information, and

(ii) the backward adaptive procedure - where adaptation is based on the received quantized signal, and can therefore be replicated at the receiver with no auxiliary information.

Computer simulation results confirmed that the forward method is slightly better if no cost is assessed for the side information. Practical considerations however, appear to favour backward adaptation[19,68].

In the area of backward adaptive quantization, the one-word memory (AQJ) quantizer proposed by Jayant[49] (and the later robust version of Goodman[190]) has stood the tests of time and emerged undoubtedly as the most widely used quantization scheme in waveform coding applications. Efforts to improve further on the AQJ have been numerous. The most notable of these are perhaps the attempts to provide for quicker adaptation response to the infrequent large amplitude excitation pulses, characteristic of the speech prediction residual signal of DPCM and ADPCM systems. The pitch compensating quantizer (PCQ) designs of Cohn and Melsa[66], and Qureshi and Forney[67], have succeeded in arresting these large amplitude excursions of the residual signal to some extent,

but this was achieved at the cost of having to resort to variable bit rate coding, which is unacceptable for many applications. Frequently, in this, as in many other areas, improvement to an existing scheme is only possible at some cost, which in this case could be in terms of complexity, robustness and practicability. Ultimately, the designer will have to select a design which offers the best compromise for his particular application.

We have introduced a new approach to the problem of improving the performance of the AQJ, which consists of applying correction to the DPCM or ADPCM decoded signal samples, based on information obtained from the quantizer output sequence. This method seeks to compensate for the limitation of the AQJ in its quantization of the residual signal, by modifying the recovered speech signal in an appropriate way. Experiments on the 2-bit AQJ have indicated that improvement in SNR has indeed been achieved by the correction process. Output noise is decreased over the entire frequency spectrum, with most of the reduction occurring in the high frequency region. Perceptual improvement in the coded speech has also been obtained, in the form of lessened background noise.

While the simple quantizer correction process is able to provide noise reduction to some extent, its limitation lies in its use of fixed correction factors, obtained from observations of the long-term quantizer input-output statistics. As a consequence, a 'compromise' amount of correction is used for a given quantizer output sequence, which is too little for some cases and too much for others. Better performance could be achieved if the correction factors are made to adapt to local signal conditions. Further research is required in this

area to explore the possibilities of providing adaptive correction.

The work described so far, in the last three chapters has been on rather 'traditional' waveform coding methods which operate in the time domain on the speech signal waveform. Recent trends in the area of speech coding have indicated a shift towards more complex frequency domain techniques which are able to exploit the properties of the speech waveform more effectively, to provide even better signal compression. These more powerful waveform coding techniques will be considered in the chapter to follow.

CHAPTER SIX FREQUENCY DOMAIN SPEECH CODING

6.1 INTRODUCTION

The rapidly increasing capability and decreasing cost of digital hardware in recent years has brought about renewed interest in sophisticated speech coding algorithms which are able to operate efficiently at relatively low transmission bit rates. One consequence of this advance in digital technology has been a noticeable drift away from the 'traditional' time domain speech coders into the realm of frequency domain coding. The basic concept in frequency domain coding is to divide the speech spectrum into frequency bands or components using either a filter bank or a block transform analysis. After encoding and decoding, these frequency components are used to re-synthesise a replica of the input waveform by either filter bank summation or inverse transformation. By splitting the input speech in this manner, different frequency bands can be preferentially encoded according to perceptual or minimum mean-square error criteria for each band. At the same time, quantization noise can be contained within bands, and prevented from creating out-of-band harmonic distortions[140].

Two basic types of frequency domain speech coders are considered in this chapter, namely, the sub-band coder (SBC)[141] and the adaptive transform coder (ATC)[161]. In the first case, the speech spectrum is partitioned into a set of typically 4 to 16 contiguous sub-bands by

means of a filter bank analysis. In the second, a block transform analysis is used to decompose the signal into typically 64 to 512 much finer frequency components. Both techniques attempt to perform some sort of short-time spectral analysis of the input signal, although the spectral resolution achieved by the two methods are quite different. The sub-band coder provides rather coarse frequency resolution, with the frequency components consisting of broad bands ranging from about 200 to 1000 Hz in width. The adaptive transform coder, on the other hand, seeks to model the detailed structure of the speech waveform, and permits much finer frequency analysis. These two methods have therefore been referred to as 'wide-band' and 'narrow-band' analysis/synthesis coders, respectively[140].

The sub-band coder and the adaptive transform coder are described in detail in the following sections. Their performance, for a range of parameter values is examined via computer simulation. Problems and practical difficulties associated with each coder, such as complexity and delay, are also discussed. Finally, a new approach to split-band coding schemes is proposed and presented. This combines the techniques of sub-band and transform coding methods, and provides a performance comparable to either, in terms of SNR and decoded speech quality, but with lower complexity and shorter coding delay.

6.2 SUB-BAND CODING (SBC)

The sub-band coder (figure 6.1)[12,141,142] partitions the input signal spectrum into typically 4 to 16 frequency sub-bands via a bank of band-pass filters. Each sub-band is in effect, low-pass translated to zero frequency by a modulation process, decimated to its Nyquist rate (twice

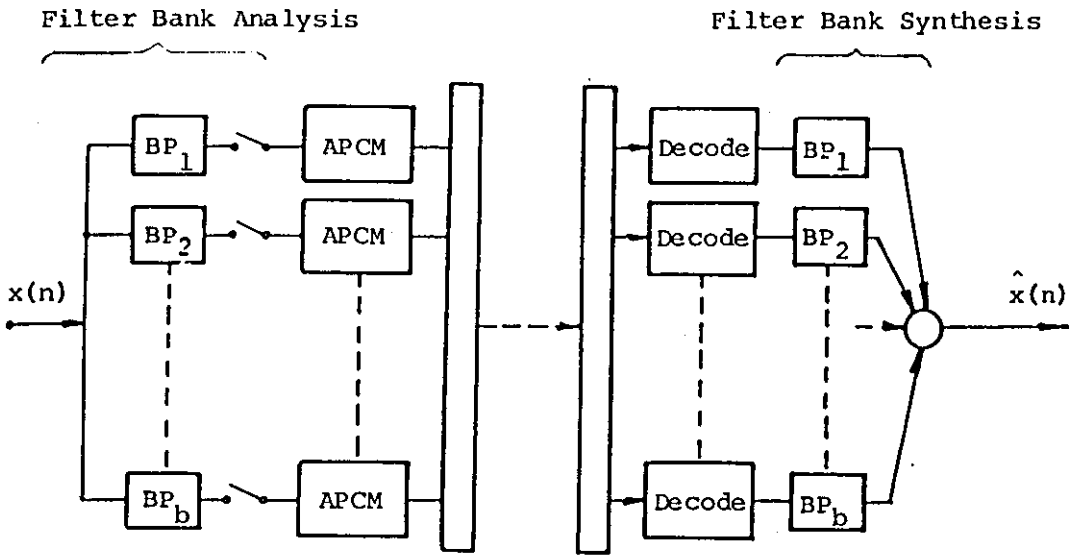


Fig. 6.1 Block Diagram of Sub-band Coder

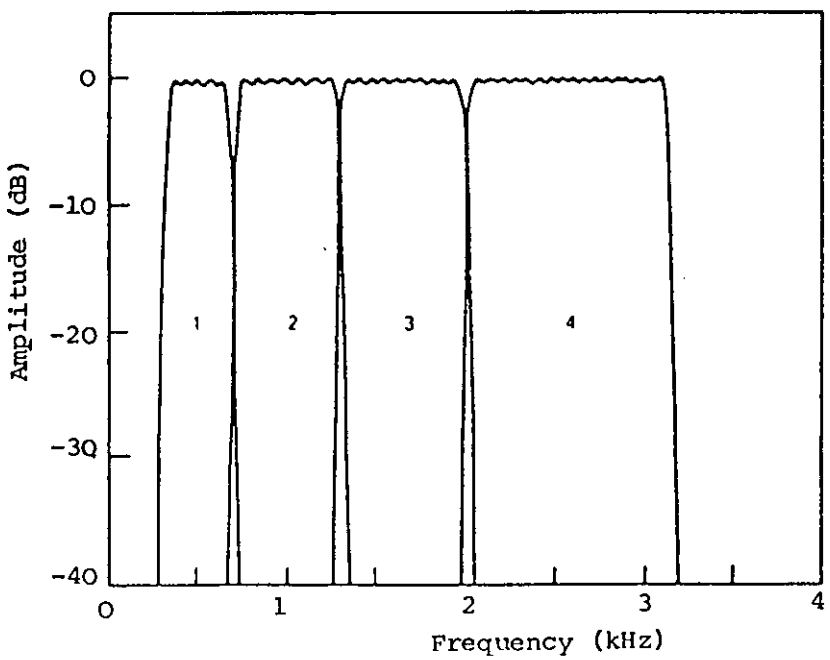


Fig. 6.2 Partitioning of Speech Spectrum into 4 Contiguous Bands, Contributing Equally to Articulation Index

the width of the band) and digitally encoded using adaptive step-size PCM (APCM). The number of bits employed for each band is determined by some perceptual or minimum mean-square error criterion. On reconstruction, the sub-band signals are decoded, modulated back to their original locations and then summed to give a close replica of the original signal.

Coding of the speech signal in sub-bands offers several advantages. Quantization noise can be contained within frequency bands to prevent masking of one frequency range by noise in another. Also, as noted earlier, bands can be preferentially encoded i.e. more bits can be assigned to the high energy low frequency bands where pitch and formant structure must be accurately preserved, and less bits to the upper frequency region where fricatives and noise-like sounds occur. Additionally, by appropriate assignment of bits to the sub-bands, the shape of the output noise spectrum may be suitably controlled to satisfy perceptual requirements[12,140].

6.2.1 Partitioning of Frequency Bands

The central feature of the sub-band coder is the splitting of the input signal into frequency bands. Early proposals to perform the band splitting employed large finite impulse response (FIR) band-pass filters [141,142]. These are necessary to provide the very sharp cut-off characteristics required to minimise the effects of signal aliasing, which occurs during the decimation of the sub-band signals[144].

Initial designs of sub-band coders consist of relatively few sub-bands. Band partitioning was made according to perceptual criteria, so that

each band contributes equally to the so-called articulation index (AI). The AI concept[234] is based upon a non-uniform division of the frequency scale for the speech spectrum. Twenty non-uniform contiguous bands are derived, each of which contributes 5% to the total AI. One early design uses 4 sub-bands, covering 200-700 Hz, 700-1310 Hz, 1310-2020 Hz and 2020-3200 Hz. Each of these bands contribute about 20% to AI, giving a total of 80%. Figure 6.2 illustrates this partitioning of the speech spectrum[141].

6.2.1.1 Integer Band Sampling

Crochiere, one of the pioneers of sub-band coding, proposed an integer band sampling technique for performing the low-pass to band-pass translations which eliminates the need for modulators and are therefore more easily realised in hardware[141]. This is illustrated in figure 6.3. The speech band is partitioned into b sub-bands by band-pass filters BP_1 to BP_b. The output of each filter in the transmitter is re-sampled at the rate of $2f_i$, where f_i is the bandwidth of the ith sub-band. These decimated signals are then digitally encoded and multiplexed for transmission. At the receiver, the decoded sub-band signals are upsampled to their original sampling rate by inserting zero-valued samples. These signals are then filtered by another set of band-pass filters, identical to those at the transmitter. Finally, the outputs of these filters are summed to give a reconstructed replica of the original input signal.

The integer band sampling method imposes certain constraints on the choice of sub-bands, as illustrated in figure 6.3. Sub-bands are required to have a frequency range between $m_i f_i$ and $(m_i+1)f_i$, where m_i

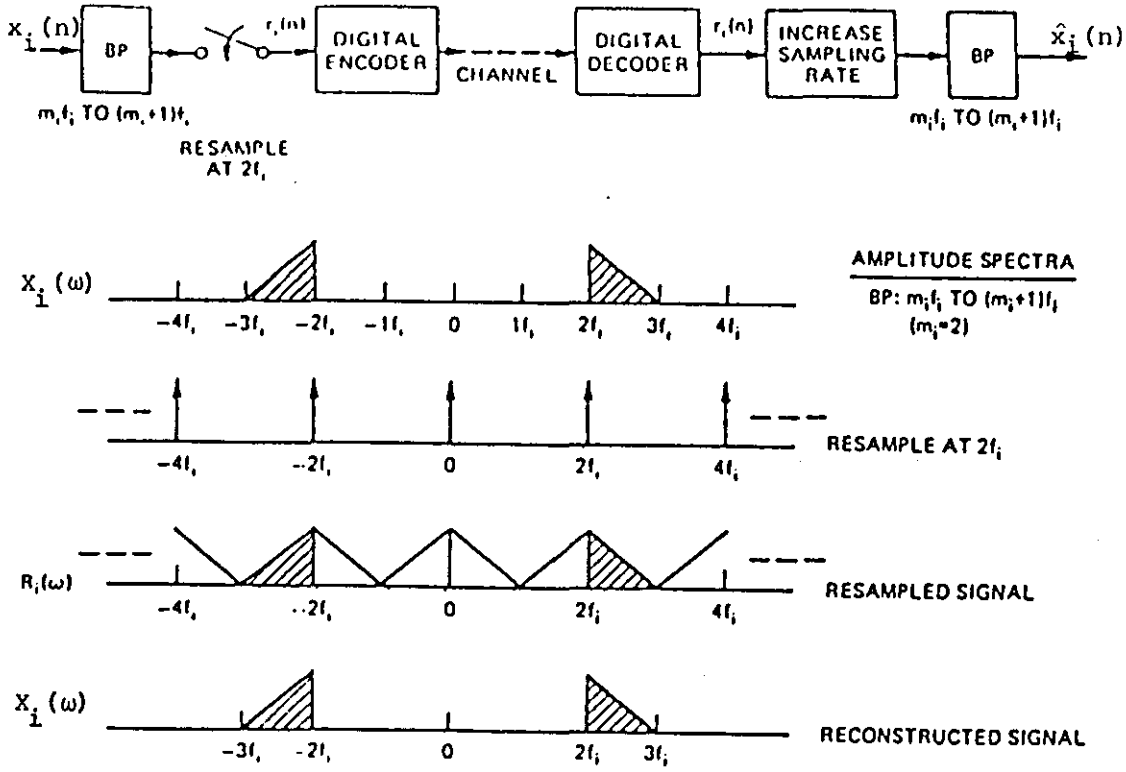


Fig. 6.3 Integer Band Sampling Technique for Low-pass to Band-pass Translation

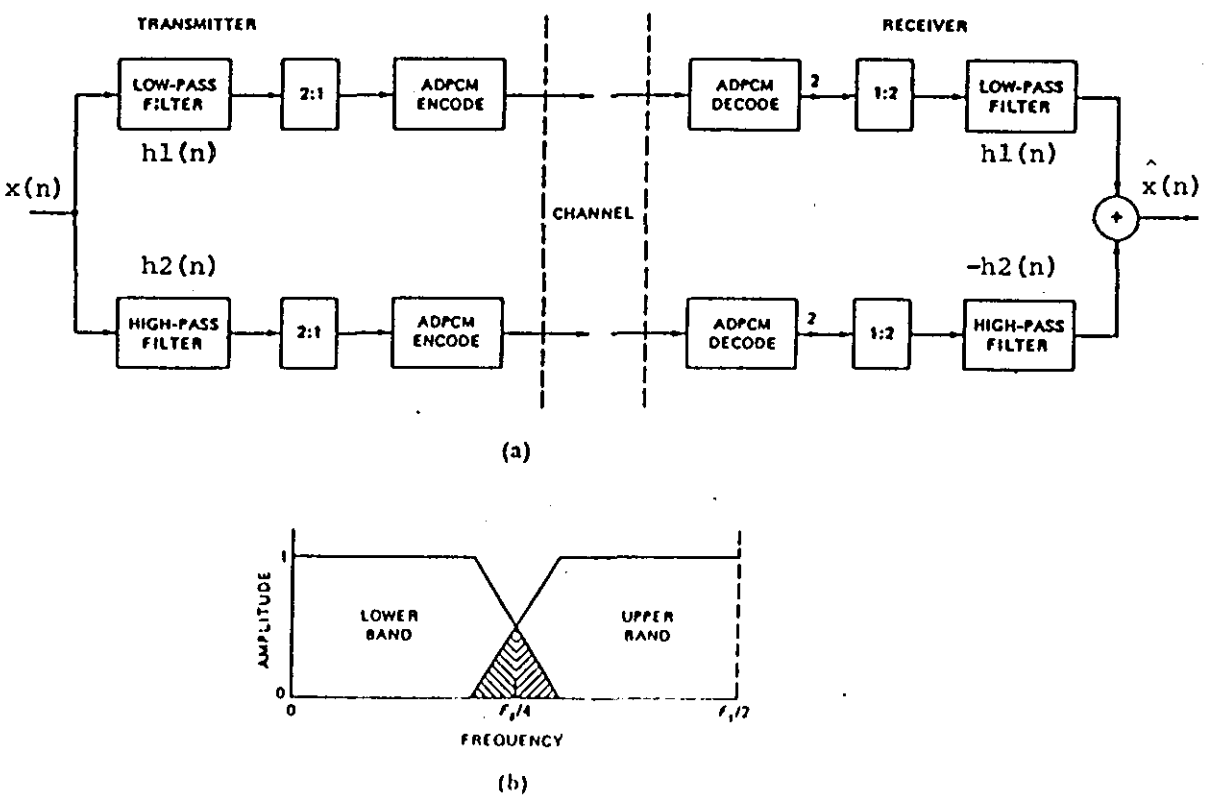


Fig. 6.4 (a) Block Diagram of a 2-Band Sub-band Coder
(b) Spectral Description of the Sub-bands

is an integer. This constraint is necessary to avoid aliasing in the sampling process.

6.2.1.2 Quadrature Mirror Filter (QMF) Bank

Although the integer band sampling method has been reported to produce encouraging results, very long filters[142] (175-200 tap FIR designs) are necessary to provide the sharp cut-off characteristics required in order to reduce aliasing or inter-band 'leakage' arising from the sampling processes. A more elegant design, proposed by Esteban[145], allows for almost perfect cancellation of this aliasing effect, by utilising a set of low and high-pass filters which possesses 'quadrature' relationships. This quadrature mirror filter (QMF) approach will be described in the following.

Consider the design of a 2 (equal) band sub-band coder which uses a low-pass and a high-pass filter to split the bands, as shown in figure 6.4. The down-sampling processes in both upper and lower bands introduce aliasing terms in each of the sub-band signals. In the lower band, the signal frequency above $f_s/4$ is folded down into the range 0 to $f_s/4$, and appears as aliasing in this signal, as illustrated by the shaded region in figure 6.4(b). Similarly, for the upper band, any signal energy below $f_s/4$ is folded upward into its Nyquist band $f_s/4$ to $f_s/2$. The amount of this mutual aliasing of energy or inter-band leakage is directly dependant on the degree to which the filters h1(n) and h2(n) approximate ideal low-pass and high-pass filters, respectively.

In the re-construction process, the sub-band sampling rates are

increased by inserting zeroes between each sub-band sample. This introduces a periodic repetition of the signal spectra in the sub-band. For example, in the lower band, the signal energy from 0 to $f_s/4$ is symmetrically folded around $f_s/4$ into the range of the upper band. This unwanted signal energy or 'image' is filtered out by the low-pass filter h1(n) at the receiver. The filtering operation effectively interpolates the zero-valued samples that have been inserted between the sub-band signals to values that appropriately represent the desired waveform. In the same way, the 'image' from the upper band is reflected to the lower sub-band and filtered out by the filter -h2(n).

Because of the quadrature relationships of the sub-band signals in the QMF bank, the remaining components of the images can be exactly cancelled by the aliasing terms introduced in the analysis (in the absence of transmission errors). In practice, this cancellation is obtained down to the level of the quantization noise of the coders.

To obtain this cancellation property in the QMF bank, the filters h1(n) and h2(n) must be symmetrical FIR designs with even numbers of taps i.e.

$$h1(n) = h2(n) = 0 \qquad \text{for } n < 0 \text{ and } n \geq T \tag{6.1}$$

where T (even) is the number of taps. The symmetrical property implies that,

$$h1(n) = h1(T-1-n) \tag{6.2a}$$

and

$$h2(n) = -h2(T-1-n) \qquad ; n = 0,1,2....T/2-1 \tag{6.2b}$$

The QMF bank further requires that the filters satisfy the condition,

$$h2(n) = (-1)\, h1(n) \qquad ; n = 0,1,2....T-1 \tag{6.3}$$

which is the mirror image relationship of the filters.

With the above constraints, the aliasing cancellation property of the QMF bank can be easily verified[145,147] as shown in appendix F. It can be seen, from the appendix, that the filters must also satisfy the condition,

$$|H1(e^{j\omega})|^2 + |H2(e^{j\omega})|^2 = 1 \qquad (6.4)$$

where $H1(e^{j\omega})$ and $H2(e^{j\omega})$ denote the Fourier transforms of h1(n) and h2(n), respectively. The requirement of (6.4) can be very closely approximated for modest values of T. Johnston[158] describes a procedure based on the Hooke and Jeeves optimisation algorithm and presents a set of filter designs for various number of taps, from 8 to 64. Less optimal filters can also be obtained using conventional Hanning window designs [143]. Figure 6.5 shows the frequency response for a 32 tap filter design obtained by Johnston (32 D design). It can be seen that the requirement of (6.4) is satisfied to within ± 0.025 dB, which is more than satisfactory for good SBC performance.

For band-splitting into more than two bands, the basic QMF bank can be repeated in a tree structure. Figure 6.6 shows the use of QMF in a 8 band sub-band coder. Notice the order of the filters h1 and h2 at each stage of the tree. This arrangement, as shown in the figure, ensures that the parallel outputs of the encoders E1 to E8 corresponds to the 8 equal sub-bands arranged in ascending order of frequency. Furthermore, h2, instead of -h2 can be used at the receiver if the signs of all outputs from h2 are reversed. Sub-band coders with non-uniform bands (such as octave designs) may also be obtained using the QMF bank approach, subject to some limitations. This is done by truncating

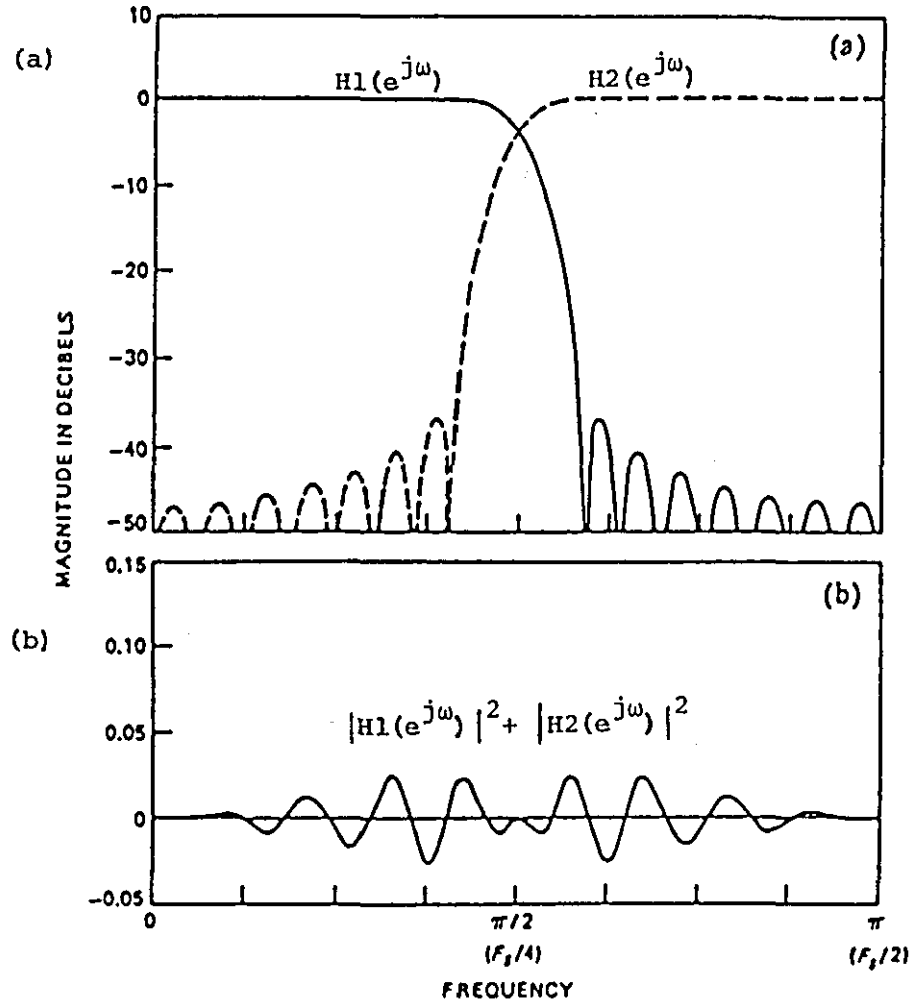


Fig. 6.5 Frequency Response of a 32-tap Quadrature Mirror Filter Design for a Two-band Sub-band Coder

(a) Magnitude Response of Individual High and Low pass Filters

(b) Magnitude Response of the Composite System

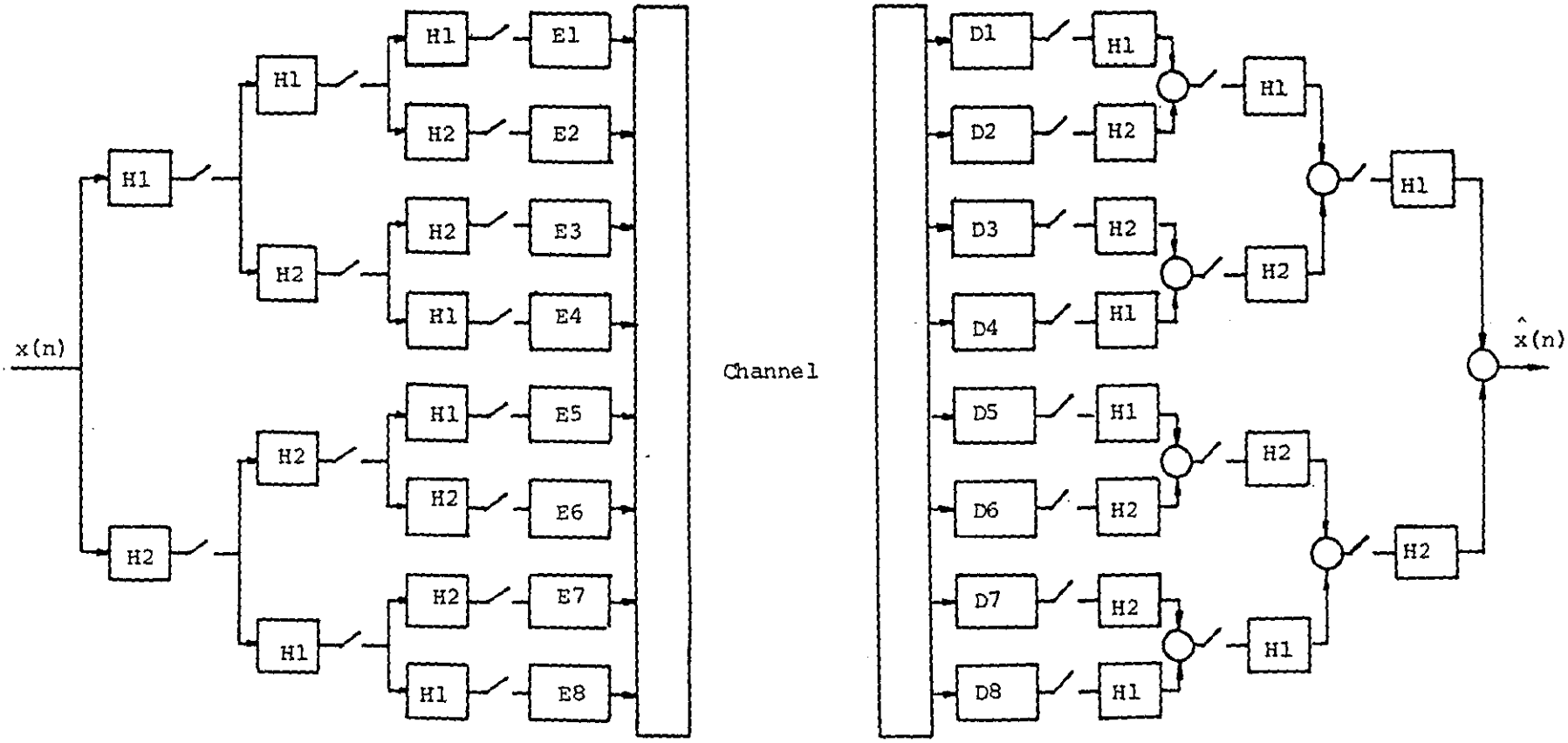


Fig. 6.6 Block Diagram of a Tree-Structured 8 Band QMF Subband Coder

certain sections of the tree as shown in figure 6.7 for a 5 band sub-band coder[153].

The use of symmetrical FIR filters in the QMF bank introduces a delay in the system equal to (T-1)/2 samples at each stage. However, because the sampling rate of the sub-band signals is halved at each stage, the actual amount of delay (referred to the original sampling rate) increases up the tree. Considering the delay at both analysis and synthesis stages, the total delay introduced by the tree-structured b-band QMF bank is given by (T-1)(b-1) samples, assuming the use of uniform filters at all stages[145].

Studies have indicated that the tree-structured QMF sub-band coder yields much inproved processed speech quality compared to the integer band sampling technique, despite the latter's use of long FIR filters [147,153]. Consequently, virtually all current implementations of sub-band coders use the QMF bank.

6.2.2 Coding of Sub-band Signals

One advantage of sub-band coders noted previously, is the exploitation of the non-flat spectral density of speech signals which allows unequal quantization to be applied to the frequency bands. The allocation of bits for coding each sub-band may be fixed or adaptive.

6.2.2.1 Fixed Bit Allocation

In early designs, the number of bits assigned for coding each sub-band signal is determined from long-term signal statistics, and are fixed for a given coder. Crochiere[141,150,153] uses the backward adaptive Jayant

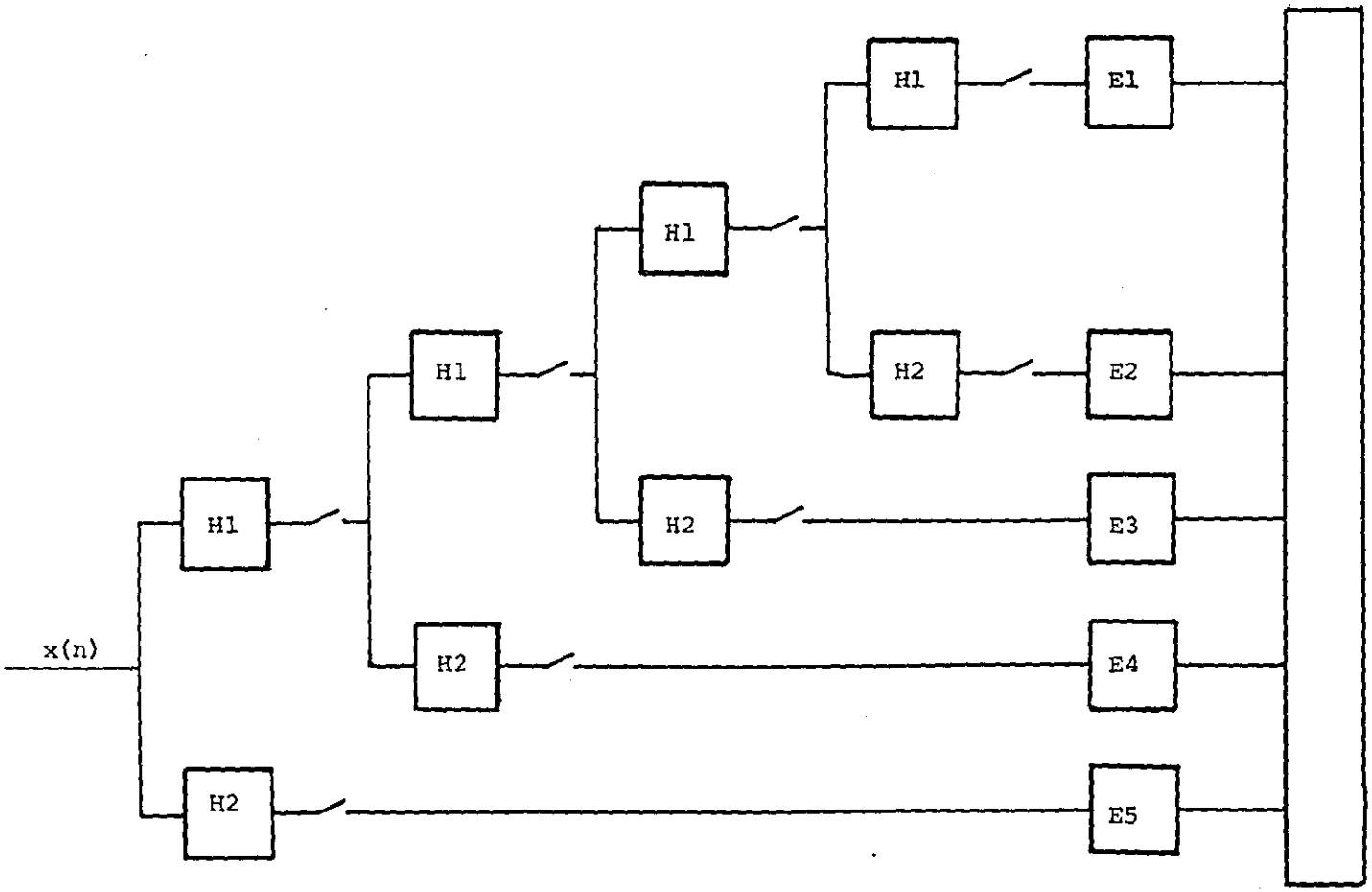


Fig. 6.7 5 Band Sub-band Coder with Non-uniform Spacing of Bands

quantizer (AQJ)[49] for his schemes, while Esteban[145] employs block quantization with forward transmission of step-sizes (AQF)[20, 41]. For a fairly large number of bands, the constraint on available quantizer bits do not in general allow the assignment of 2 bits to code the high frequency bands, a condition which is necessary for the backward adaptation of the AQJ. Crochiere[142] suggested using the 1 1/k bit quantizer, a modification of the AQJ, proposed by Goodman. In this approach, the sign of the signal is encoded every sample, and the magnitude is transmitted with one bit every k samples. The sign bit transmits essentially the 'zero crossing' or phase information and the magnitude bit conveys the amplitude information of the waveform at a reduced rate.

6.2.2.2 Adaptive Bit Allocation

As speech is a non-stationary signal, fixing the number of bits (from long-term consideration) for coding each sub-band will necessarily be sub-optimum in the short-term. Better results can be obtained by allowing the number of bits assigned to each frequency band to vary according to local signal statistics. Adaptive or dynamic techniques of bit allocation attempt to distribute available bits more efficiently by assigning bits to the sub-bands according to their energy composition over a short segment of typically 10 to 30 ms of speech. In this way, efficient coding is maintained and no bits are 'wasted'. Naturally, adaptive bit allocation requires the transmission of side information periodically so that the receiver is kept informed of the update in the bit allocation patterns. The optimum assignment of bits is based on a minimum mean square error criterion and is given by the well-known

equation[12,48,140,161]

$$R_i = d + 1/2 \log_2 \frac{\sigma_i^2}{D^*} \qquad ; i = 1,2,....b \tag{6.5}$$

where σ_i^2 is the variance, and R_i, the optimum number of bits for the ith sub-band. b is the number of bands in the sub-band coder, or the number of bands considered in the allocation process, since certain frequency bands beyond the signal cut-off frequency may be omitted. d is a correction term that reflects the performance of practical quantizers, and D* denotes the noise power,

$$D^* = 1/b \sum_{i=1}^{b} e_i^2 \tag{6.6}$$

where e_i^2 is the noise power incurred in quantizing the ith sub-band. The bit assignment obtained from (6.5) must satisfy the constraint of available bits, R

$$R = \sum_{i=1}^{b} R_i \tag{6.7}$$

It is easy to obtain the result that all bands must have the same distortion. The optimum bit assignment is then,

$$R_i = \bar{R} + 1/2 \log_2 \frac{\sigma_i^2}{\left[\prod_{j=1}^{b} \sigma_j^2\right]^{1/b}} \tag{6.8}$$

where $\bar{R}$ is the average bit rate, given by,

$$\bar{R} = 1/b \sum_{i=1}^{b} R_i \tag{6.9}$$

The R_i's calculated from (6.8) cannot take on negative or fractional values in practice since they represent the number of quantizer bits to

be used. Hence, rounding to the nearest positive integer or zero is necessary, and this must be done without violating the constraint of (6.7).

The bit allocation equation given by (6.5) can be modified slightly to provide some control of the output noise shape which might be desirable from a perceptual point of view[12,140]. However, the relatively small number of frequency bands in sub-band coders does not allow much room for manouvre in this respect. Such frequency domain noise shaping is more appropriate in the context of adaptive transform coding (see section 6.3.3 below).

6.2.3 Computer Simulation

6.2.3.1 General Procedure

The uniform tree-structured QMF implementation of the sub-band coder is simulated on the computer. The same number of taps is used for the low and high-pass filters at every stage of the tree. 32 taps are used for the 2,4, and 8 band SBCs and 16 taps, for the 16 band case. The filter coefficients are obtained from Johnston's '32 tap(E)' and '16 tap(C)' designs[158]. These are shown in table 6.1

When the number of sub-bands is sufficiently large, certain bands in the high frequency end of the spectrum may not need to be transmitted at all, since they correspond to information beyond the bandwidth of the input signal. The input speech data used in the simulation is band-limited to 3400 Hz and sampled at 8000 Hz, so the frequency band between 3400 and 4000 Hz theoretically does not contain any speech information. Hence, for the so-called 8 and 16 band sub-band coders, effectively only

7 and 14 bands, respectively, are actually transmitted. This is useful in conserving quantizer bits.

Table 6.1 Coefficients for 32 and 16 tap FIR Quadrature Mirror Filters

(a) 32 tap

h1(0) = 0.005123 = h1(31)	h1(8) = -0.014569 = h1(23)
h1(1) = -0.011276 = h1(30)	h1(9) = -0.038306 = h1(22)
h1(2) = -0.000962 = h1(29)	h1(10) = 0.026624 = h1(21)
h1(3) = 0.015681 = h1(28)	h1(11) = 0.055707 = h1(20)
h1(4) = -0.002612 = h1(27)	h1(12) = -0.051383 = h1(19)
h1(5) = -0.021038 = h1(26)	h1(13) = -0.097684 = h1(18)
h1(6) = 0.007380 = h1(25)	h1(14) = 0.138764 = h1(17)
h1(7) = 0.028123 = h1(24)	h1(15) = 0.459646 = h1(16)

(b) 16 tap

h1(0) = 0.006526 = h1(15)	h1(4) = -0.026276 = h1(11)
h1(1) = -0.020488 = h1(14)	h1(5) = -0.099296 = h1(10)
h1(2) = 0.001991 = h1(13)	h1(6) = 0.117867 = h1(9)
h1(3) = 0.046477 = h1(12)	h1(7) = 0.472112 = h1(8)

Figure 6.8 shows the decimated sub-band signals of the 8 band SBC, obtained from a typical segment of voiced speech. Notice the characteristic concentration of signal energy in the lower frequency bands and also, the lack of correlation in the signals after decimation. The signal correlation in the sub-bands decreases as the number of bands is increased, since the corresponding spectra becomes progressively 'flatter' as the width of the frequency bands gradually narrows. Table 6.2 shows the average first shift autocorrelation coefficients obtained from the sub-band signals for the 2, 4 and 8 band coders. It can be seen that, apart from the first band of the two-band SBC, little correlation can be expected in the sub-band signals. Correlation values

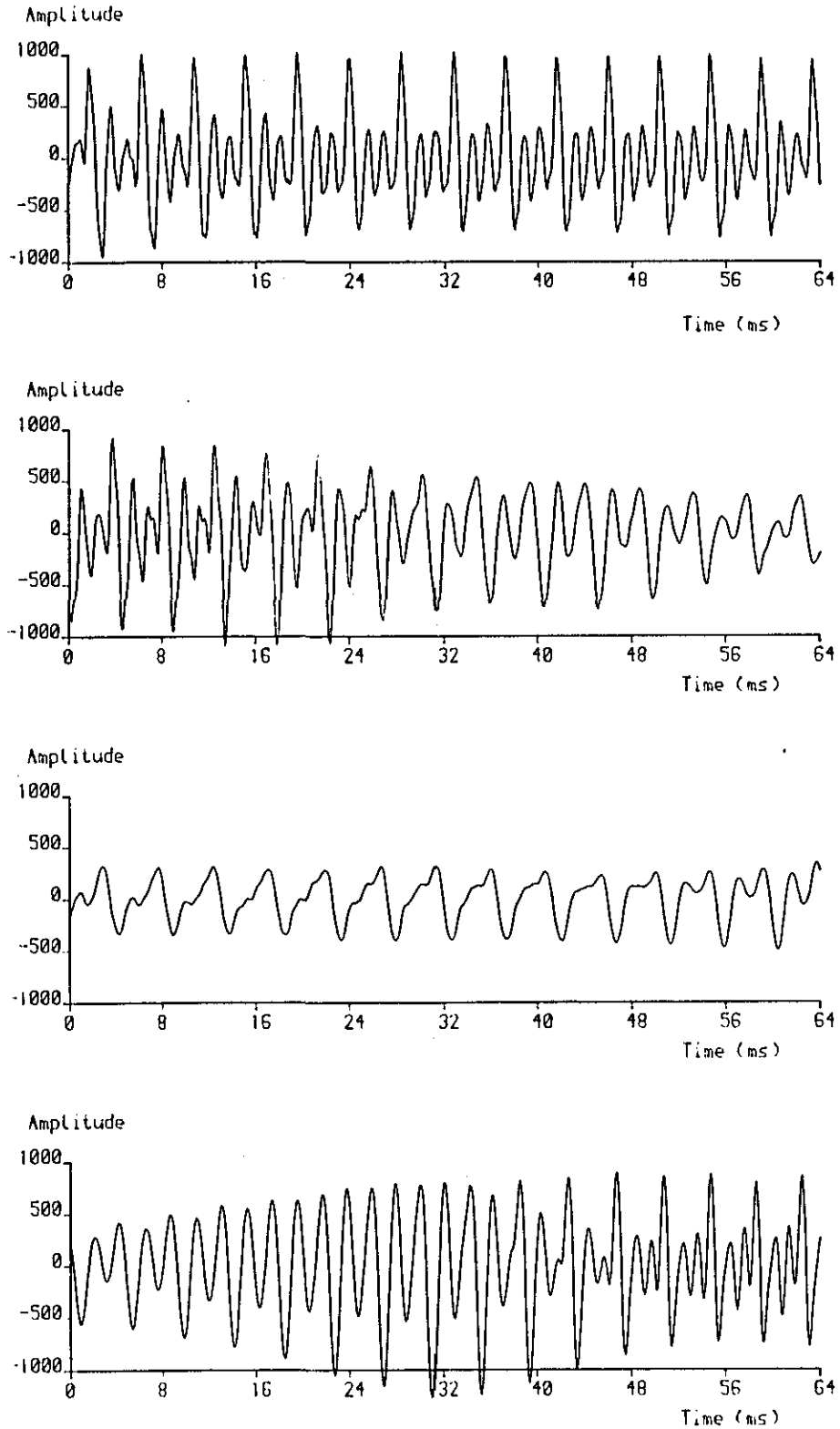


Fig. 6.8(a) Full Band Input Signal Waveform

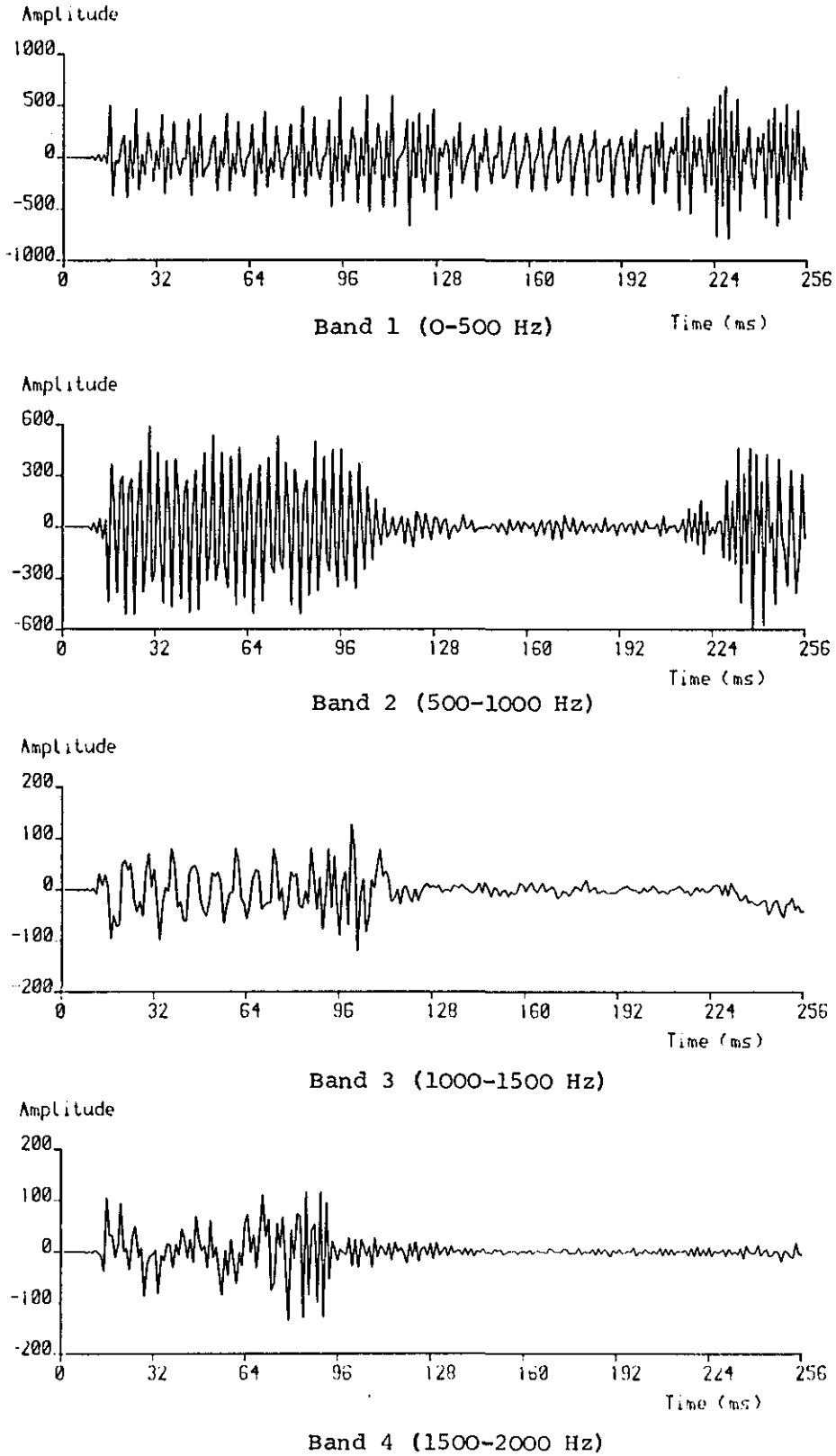
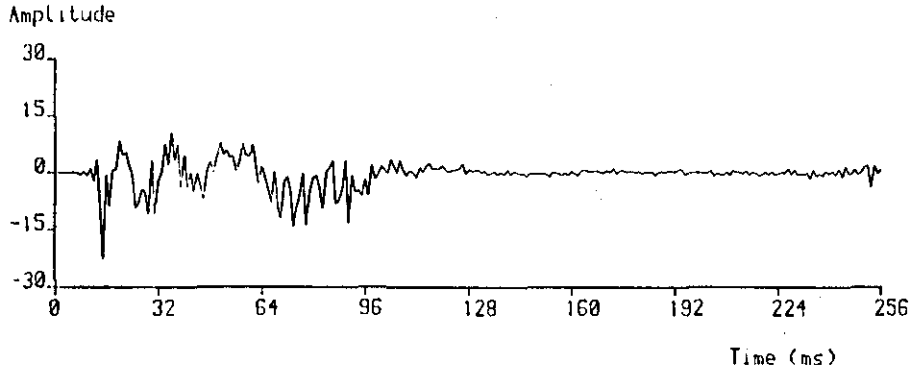
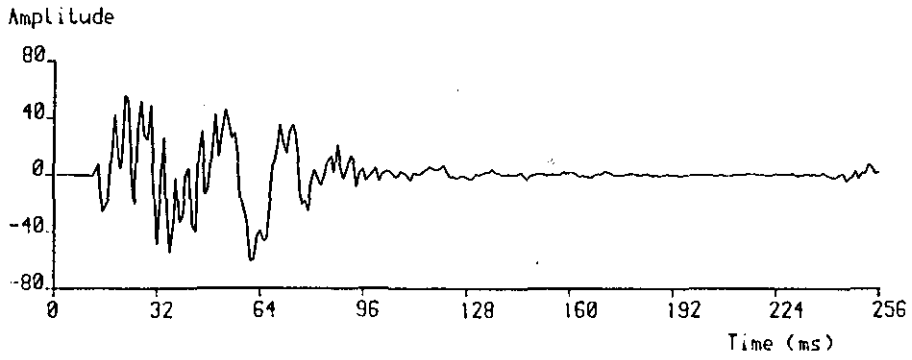


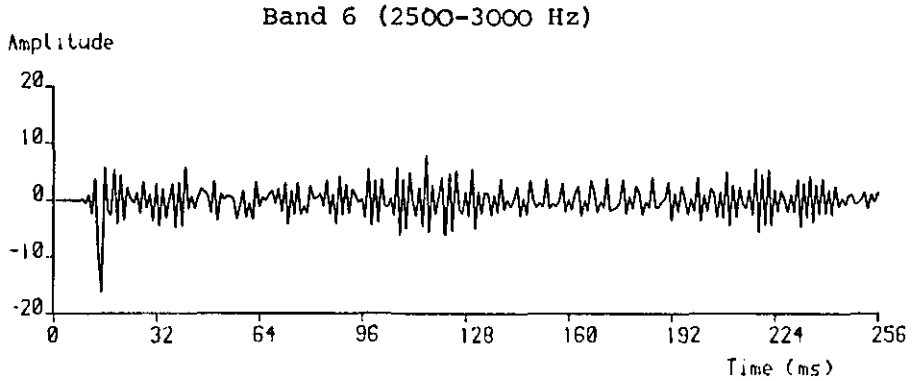
Fig. 6.8(b) Sub-band Signals of a 8 Band Sub-band Coder



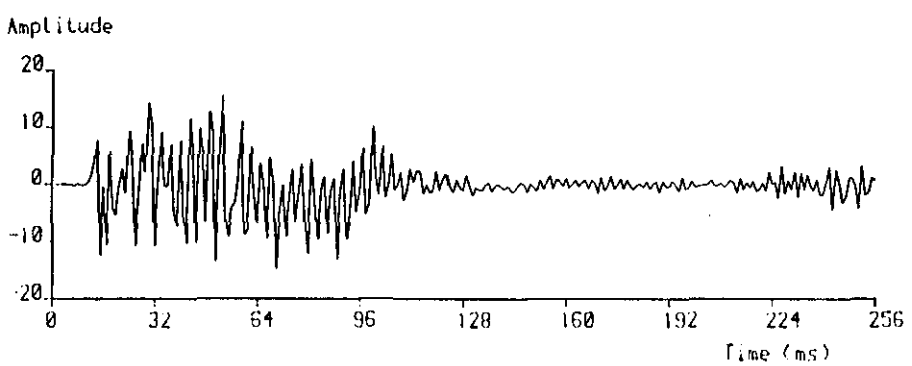
Band 5 (2000-2500 Hz)



Band 6 (2500-3000 Hz)



Band 7 (3000-3500 Hz)



Band 8 (3500-4000 Hz)

Fig. 6.8 Sub-band Signals

for the same frequency bands also vary widely among different input data. Therefore, the use of differential techniques to encode the sub-band waveforms does not offer any advantages[145], and consequently, in the simulations performed, all encoding is done using APCM.

Table 6.2 Correlation Coefficients for Sub-band Signals

	a(1)	a(2)	a(3)	a(4)	a(5)	a(6)	a(7)	a(8)
MALE								
2-band	0.832	-0.074						
4-band	0.580	-0.405	0.116	0.304				
8-band	0.147	-0.302	0.397	-0.168	-0.294	-0.017	0.047	-0.016
FEMALE								
2-band	0.603	-0.364						
4-band	0.321	-0.412	0.402	-0.071				
8-band	-0.279	-0.256	0.302	-0.255	-0.035	0.258	-0.335	0.077
SISTER								
2-band	0.763	-0.139						
4-band	0.585	-0.304	0.183	0.166				
8-band	0.406	-0.337	0.262	-0.263	-0.220	-0.037	0.172	-0.047

6.2.3.2 Bit Allocation

Both fixed and adaptive methods of assigning bits to code the sub-band signals were investigated. Adaptive bit allocation is performed using the alternative formulation of (6.8)[161],

$$R_i = \bar{R} + 1/2 \log_2 \sigma_i^2 - \frac{1}{2b} \sum_{j=1}^{b} \log_2 \sigma_j^2 \qquad (6.10)$$

By changing the geometric mean term of (6.8) into an arithmetic mean in

(6.10), implementation on the computer is greatly simplified. As R_i can only take on integer values, each value as derived from (6.10) must be rounded to the nearest positive whole number or zero. Following this, further adjustments must be made to ensure that the integer bit assignment satisfies the constraint on available bits given by (6.7). The full bit allocation procedure as implemented in the simulation involves the following steps:

(1) The variances σ_i^2 of each sub-band signal over an appropriate time segment (typiclally 8-32 ms) are first calculated.

(2) Sub-bands which are beyond the input signal's frequency range (such as band 8 for the 8 band coder, bands 15-16 of the 16 band coder) are effectively prevented from being assigned bits by dividing their variances by a constant factor (e.g. 10) before including them in the bit allocation process. This method provides virtually identical bit allocation patterns to the case when the out-of-range bands are excluded from consideration, and can be more conveniently implemented on the computer.

(3) These values of σ_i^2 are then used in the bit assignment equation of (6.10) to obtain the R_i's. The average bit rate $\bar{R}$ used in the equation must be modified to account for channel capacity occupied by the side information.

(4) The R_i's are then rounded up or down to the nearest integer value to give the bit assignment map.

(5) Further adjustments are necessary to ensure that the constraint on available bits (equation (6.7)) is satisfied and that no band receives more than the maximum allowable number of bits (7 in this case). If more bits than available have been allocated, then the excess bits are taken away from bands which least deserve them i.e.

for which the integer rounding process adds the greatest amount. For example, a band with an initial R_i of 3.6, rounded up to 4 is deemed to be less deservant than one with an initial R_i of 4.8 rounded up to 5. Similarly, when the number of bits allocated is fewer than available, the extra bits are given to bands which most deserve them i.e. the bands from which the integer rounding process takes away the greatest amount.

The flow chart of the bit allocation procedure is shown in figure 6.9. R_i is the number of bits assigned to the ith band from (6.10), R_i' is R_i rounded to the nearest integer, R is the total number of bits available and R_{max} is the maximum allowable number of bits for each band.

Adaptive bit allocation is generally used with forward adaptive quantization of the sub-bands, where the sub-band signal variances are transmitted to the receiver. The quantized version of these variances are used at both transmitter and receiver to compute the bit allocation pattern and the quantizer step-sizes. This ensures that the parameters used at both ends are identical. Consequently, the bit allocation equation of (6.10) uses $\hat{\sigma}_i$, instead of σ_i, in practice. The fixed bit allocation map may be obtained by using the same procedure and averaging the bits assigned to each frequency band over the long-term. However, to prevent loss of bandwidth in the synthesised speech, at least one bit must be assigned to each frequency band, even though some of the high frequency bands contain insignificant information most of the time. Because of this inefficient utilisation of available bits, and the inability to properly track the short-term signal spectral variations, the performance of sub-band coders employing fixed bit allocation is

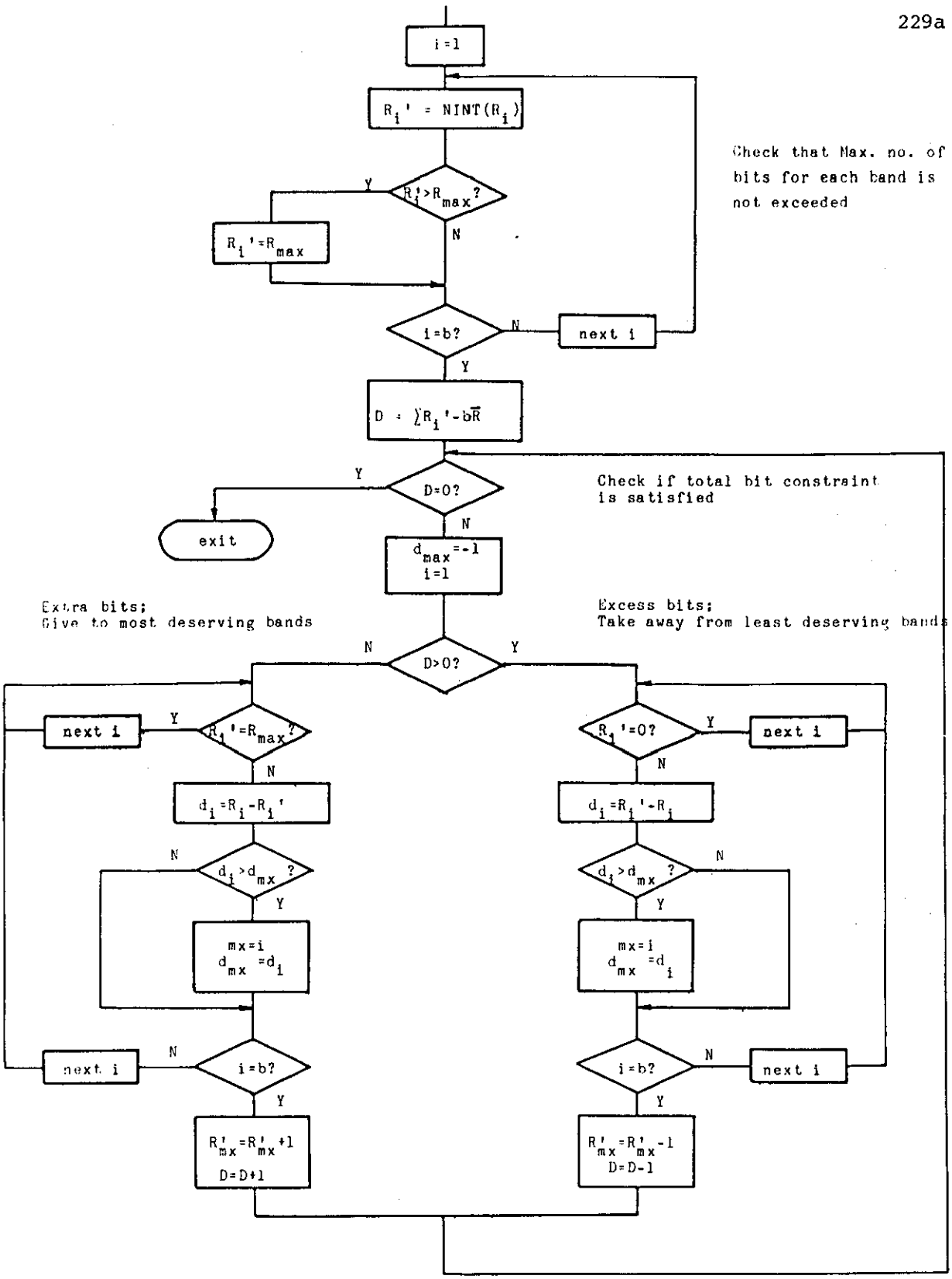


Fig. 6.9 Flow-chart for the Adaptive Bit Allocation Procedure

necessarily inferior to the fully adaptive case. Its advantages however, lies in its much reduced complexity. Esteban[145] proposed the bit allocation pattern 3333 1111 for an 8 band SBC operating at 16 Kbps using an input signal band-limited from 0 to 4 kHz.

6.2.3.3 Quantization

The sub-band signals are normally coded using APCM-AQF, particularly when the number of bands is large. The step-sizes employed in the quantization are determined from the signal variance of each band, which are transmitted as side information. The proportion of available bits assigned for the side information depends on the frequency of update of the quantizer step-sizes. Table 6.3 shows the segmental SNR results obtained for the 2,4,8 and 16 band sub-band coders simulated, where the quantizer step-sizes (and bit allocation patterns) are updated after every 256, 128, 64 and 32 input samples. Allowance has been made for the side information required for transmission of the sub-band variances (5 bits each per block), so the results apply for a total transmission rate of 16 Kbps.

Table 6.3 Segmental SNR performance for Sub-band Coder Employing Adaptive Bit Allocation and APCM-AQF (16 Kbps)

Update Blocksize	256		128		64		32	
	Male	Female	Male	Female	Male	Female	Male	Female
2-band	18.76	18.48	19.10	18.74	19.30	18.95	19.26	18.95
4-band	23.03	22.18	22.09	20.65	22.71	21.05	18.50	16.16
8-band	23.64	22.73	23.71	22.44	22.37	20.37		
16-band	23.80	22.66	22.99	21.18	18.16	17.34		

It can be seen that the SNR generally increases with the number of sub-bands and reaches its peak when b=8. SNR also falls as the blocksize for updating the quantizer is reduced, since proportionately less bits are available for signal coding, due to the resulting increase in side information. A quantizer update blocksize of 128 samples (or 16 ms) appears to be a good compromise in terms of performance and delay.

Figure 6.10 shows the output noise spectra for the 4, 8 and 16 band coders employing adaptive bit allocation, with the parameters updated every 16 ms. The lower noise level of the 8 and 16 band coders over the 4 band case is clearly demonstrated.

6.2.3.4 Subjective Quality

Recordings were made of the decoded speech from sub-band coding schemes using various combinations of parameters. Informal listening tests indicate a high quality of received speech generally, for the bit rate concerned. The high frequency hiss characteristic of time domain coders such as ADPCM at this bit rate is virtually absent, as can be deduced from figure 6.10. For the 4 and 8 band coders however, a whistling distortion is quite clearly audible. This was found to be due to the high frequency peaks (fig. 6.10) which were not totally removed by the analogue filter used in the recording. These however, could be removed by digitally filtering the output speech using a 33-tap FIR low-pass filter on the computer. Nevertheless, a 'whispery' distortion remains, accompanied by a hollowness when the number of bands is small. The 'whisper' is due to aliasing effects which occurs in the synthesis process when one or more bands (usually the high frequency bands) are

not transmitted, and a folded scaled down image of the low frequency band(s) occupies the spectral gaps in the signal. The 'hollowness' is due to these spectral gaps. For 16 bands however, this hollowness disappears and the whisper is much less noticeable. In fact, the quality obtained for the 16 band coder is excellent and very close to the original.

Recordings were also made for the same sub-band systems, which have the maximum number of bits allowed in each band reduced to 5. Although quality is still generally good, the distortions in this case are considerably more apparent.

Before discussing the merits or demerits of the sub-band coder further, we shall pause briefly to consider the other powerful frequency domain coder which provides even finer frequency analysis. The adaptive transform coder will be described in the next section.

6.3 ADAPTIVE TRANSFORM CODING (ATC)

The adaptive transform coder (ATC)[12,140,161,162] is a more complex frequency analysis technique which involves block transformations of windowed segments of the input speech. Each segment is represented by a set of transform coefficients which are separately quantized and transmitted. At the receiver, the quantized coefficients are inverse transformed to produce a replica of the original segment. Adjacent segments are then joined together to form the synthesised speech.

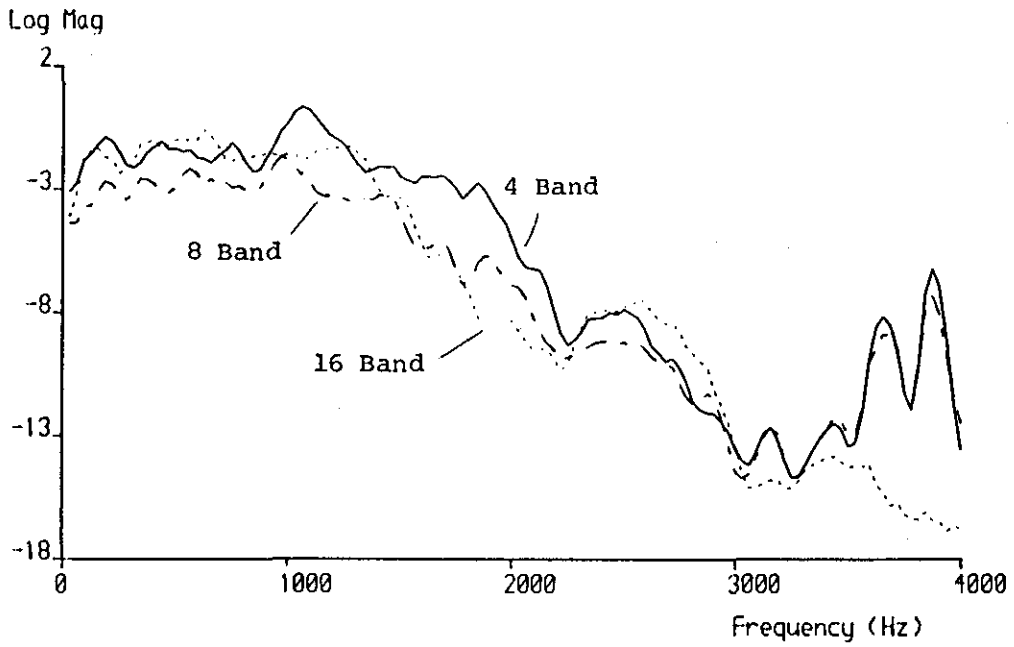


Fig. 6.10 Output Noise Spectra of 4,8 and 16 band Sub-band Coder Using Adaptive Bit Allocation and APCM-AQF (Male Speech)

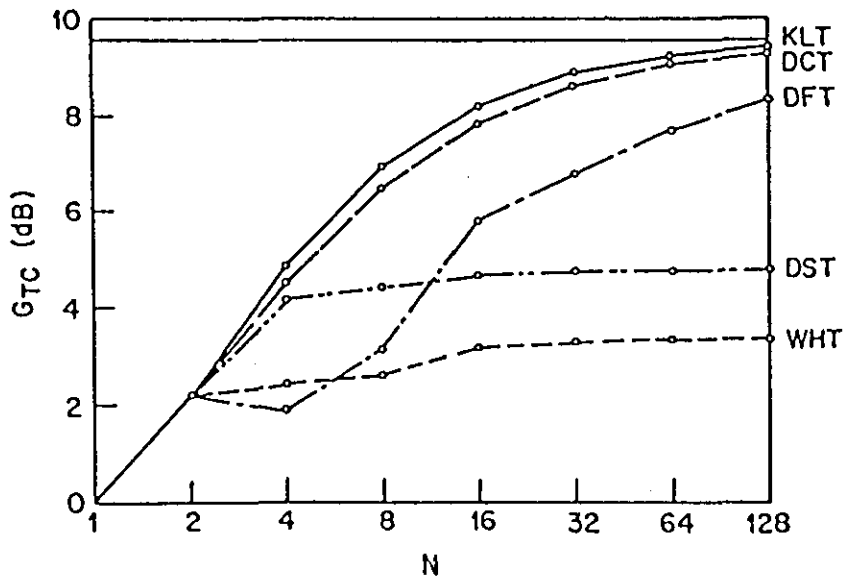


Fig. 6.11 Theoretical Gains in SNR over PCM of Various Unitary Transforms

6.3.1 The Block Transformation

Block transformation techniques have been widely used in image coding systems with much success[235], but it was only recently applied to speech coding. The class of transforms of interest for speech processing are the orthogonal time-to-frequency transformations.

It can be shown[48,161] that the gain of a transform coding scheme (using an N point transform) over PCM can be given as,

$$G_{tc} = \frac{\sigma^2}{\left[\prod_{j=1}^{N} \sigma_j^2\right]^{1/N}} \qquad (6.11)$$

where σ^2 represents the variance of the signal and σ_j^2 are the variances of the N transform coefficients. This gain is in fact the ratio of the arithmetic and geometric means of the variances of the transform coefficients, since the signal variance σ^2 for unitary transforms is equal to the average of the variances of the transform coefficients.

$$\sigma^2 = 1/N \sum_{j=1}^{N} \sigma_j^2 \qquad (6.12)$$

Zelinski and Noll[161] obtained the value of G_{tc} for various unitary transforms, using a stationary tenth order Markov process whose first ten autocorrelation coefficients are equal to the first ten long-term autocorrelation coefficients of speech. Figure 6.11 shows the results obtained using various blocksizes of the Karhunen-Loeve, discrete cosine, discrete Fourier, discrete slant, and the Walsh-Hadamard transforms. Note that the discrete cosine transform (DCT) has a performance very close to the optimum signal-dependant Karhunen-Loeve transform (KLT) and significantly superior to the others.

Indeed, the DCT has been found to be ideally suited for the coding of speech as well as picture signals[12,140,161,164,235]. Apart from its signal independence, and its approximation to the KLT, its even symmetry helps to minimise end effects encountered in block coding methods. The DCT of an N-point sequence is formally defined as,

$$X_c(k) = \sum_{n=0}^{N-1} x(n)\,c(k)\cos\left(\frac{(2n+1)k\pi}{2N}\right) \qquad (6.13a)$$

$$k = 0,1,2.....N-1$$

where $c(k) = 1, \quad k = 0$

$= \sqrt{2}, \quad k = 1,2,.....N-1$

The inverse DCT is defined as,

$$x(n) = 1/N \sum_{k=0}^{N-1} X_c(k)\,c(k)\cos\left(\frac{(2n+1)k\pi}{2N}\right) \qquad (6.13b)$$

$$n = 0,1,2.....N-1$$

Fast algorithms have been derived for implementing the DCT with great computational efficiency, comparable to the FFT[236-238].

6.3.2 Quantization of the Transform Coefficients

The quantization of the transform coefficients is of fundamental importance since it determines the accuracy of preservation of the short-time signal spectrum, and hence the quality of the synthesised speech. Usually these coefficients are individually quantized, with the step-sizes and number of bits determined from the energy distribution in the cosine basis spectrum. For minimum mean-square error distortion, the number of bits assigned for coding the N transform coefficients is

determined by the same bit allocation equations used for sub-band coding i.e. equations (6.5) to (6.9), with b (the number of sub-bands) replaced by N (the number of transform coefficients). Unlike the SBC however, fixed bit allocation is not applicable to ATC. This is because the latter operates by adapting to the fine resolution short-term frequency characteristics of speech, which may vary drastically from block to block. Consequently, a bit assignment pattern based on long-term statistics would be severely sub-optimum, as has been demonstrated by Zelinski and Noll[161]. Further, as was observed previously with regard to SBC, fixed bit allocation requires the assignment of at least one bit to each frequency component to prevent loss of bandwidth in the synthesised signal. This would result in substantial 'wastage' of bits for the transform coder which has typically 128-256 transform coefficients.

6.3.3 Noise Shaping

As in time domain waveform coding techniques, the noise spectrum of frequency domain coders may also be shaped appropriately to improve the perceptual quality of the decoded speech[12,140]. The bit assignment rule prescribed by (6.5) produces an output noise with flat spectral characteristics, which is known to be perceptually sub-optimal. This flat noise spectrum however, could be controlled to some extent by performing the bit assignment based on a different criterion. The modified bit assignment rule to permit control of the noise spectrum [12,140,239] is given by,

$$R_i = d + 1/2 \log_2 \frac{W_i \sigma_i^2}{D^*} \quad ; i = 0,1,.....N-1 \qquad (6.14)$$

where W_i represents a positive weighting. By changing the weighting function W_i, the shape of the output noise spectrum can be varied, from the flat minimum distortion case to a shape which follows the input signal's spectral envelope. For any particular transmission bit rate, the perceptually optimum value of W_i can be determined by means of listening tests.

6.3.4 Adaptation Strategy

The adaptive bit assignment used in ATC schemes seeks to exploit the non-flatness of the speech signal density, by distributing bits unevenly across the spectrum. The actual step-sizes to be used in the quantizer however, needs to be estimated, since the expected spectral levels of the transform coefficients are not known a priori. Thus, some side information which reflects the dynamic properties of speech must be transmitted. This adaptation information is used at both transmitter and receiver to determine the bit assignment pattern and the quantizer step-sizes for the block and is therefore of critical importance. Two basic adaptation techniques will now be considered.

6.3.4.1 Zelinski and Noll's Scheme

The best known adaptive transform coder for speech applications is probably the proposal of Zelinski and Noll shown in block diagram form in figure 6.12[161,162]. A block of N input speech samples is first normalised by its estimated standard deviation and then transformed into

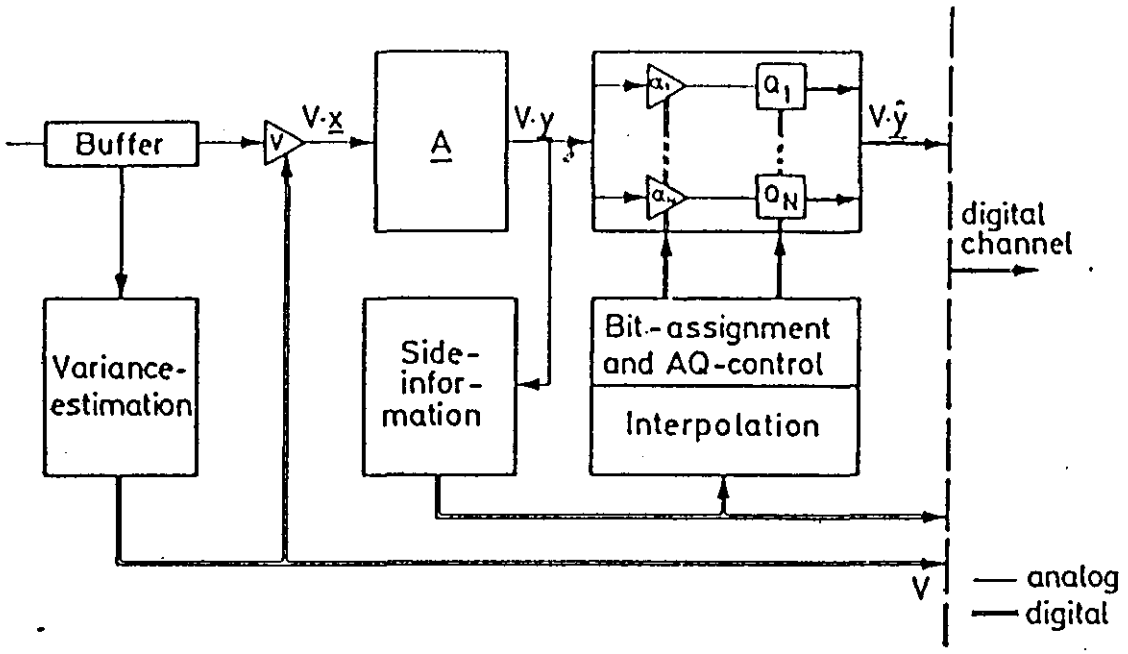


Fig. 6.12 Block Diagram of an Adaptive Transform Coder

a set of frequency domain coefficients via an N-point DCT. A coarse description of the cosine basis spectrum is extracted and transmitted to the receiver as side information. This (quantized) coarse spectral estimate is used at both transmitter and receiver to calculate the optimum assignment of bits and the quantizer step-sizes for coding the coefficients. The spectral estimate consists of a small number of samples computed by averaging the DCT spectral magnitudes (figure 6.13). These samples are then geometrically interpolated (i.e. linearly interpolated in log magnitude) to yield the expected spectral levels at all frequencies used for determining the quantizer parameters. Excellent synthesised speech quality was reported using this method at 16 Kbps.

As the bit rate is reduced however, it becomes increasingly difficult to accurately encode the fine structure (pitch details) of the DCT

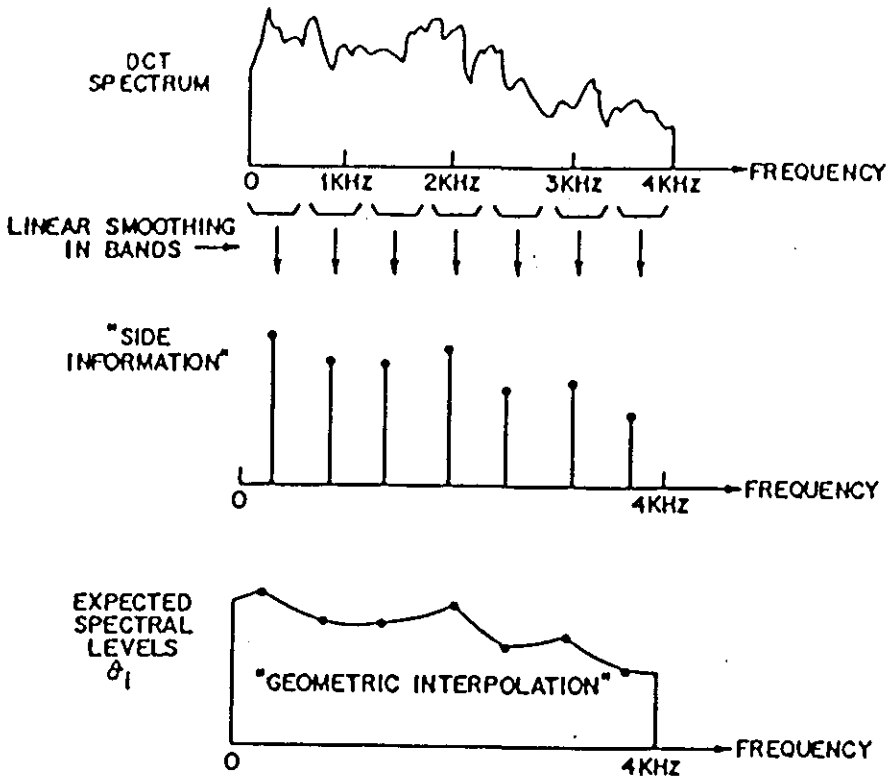


Fig. 6.13 Spectral Estimation Procedure for ATC

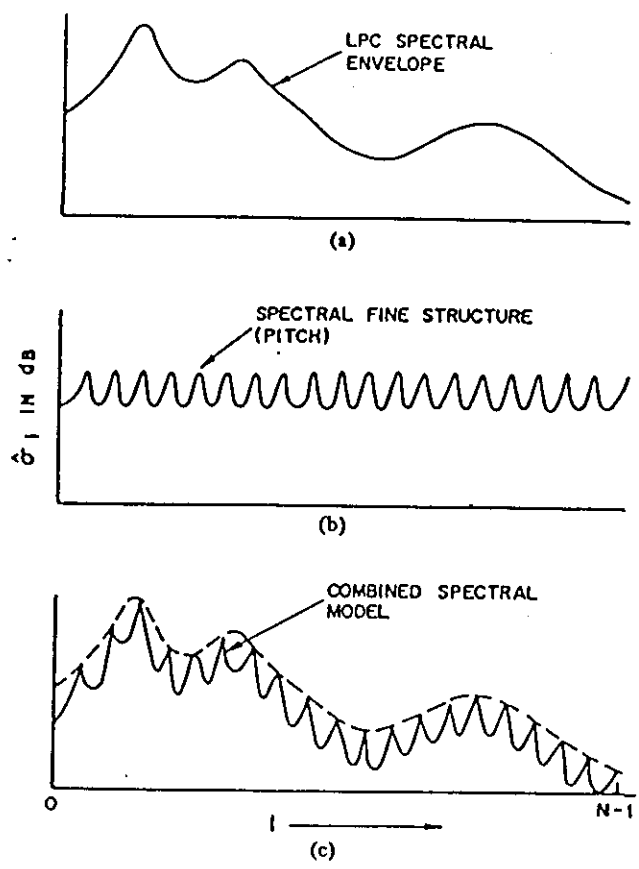


Fig. 6.14

Speech Spectrum Model

(a) Spectral Envelope

(b) Spectral Fine Structure (pitch)

(c) Combined Model

spectrum, and this gives rise to a 'burbly' distortion in the recovered speech. At the same time, the shortage of bits results in wide gaps in the spectrum, as a substantial proportion of coefficients are not transmitted. This leads to significant loss of bandwidth and the so-called 'low-pass' effect[12,140,162].

A number of remedial measures have been proposed to combat this quality deterioration at low bit rates. These include uneven spacing of the side information spectral estimates (to give more emphasis to perceptually important frequency regions[162,239]), ensuring that a minimum proportion of transform coefficients are transmitted and substituting non-transmitted coefficients with an amount of noise (to reduce the low-pass effect), and more efficient quantization of the side information by exploiting various redundancies present[162]. However, these attempts have not succeeded in adequately correcting for the inaccuracy of preservation of the short-time spectrum, which is the predominant cause of the performance degradation.

6.3.4.2 Vocoder Driven Adaptive Transform Coder

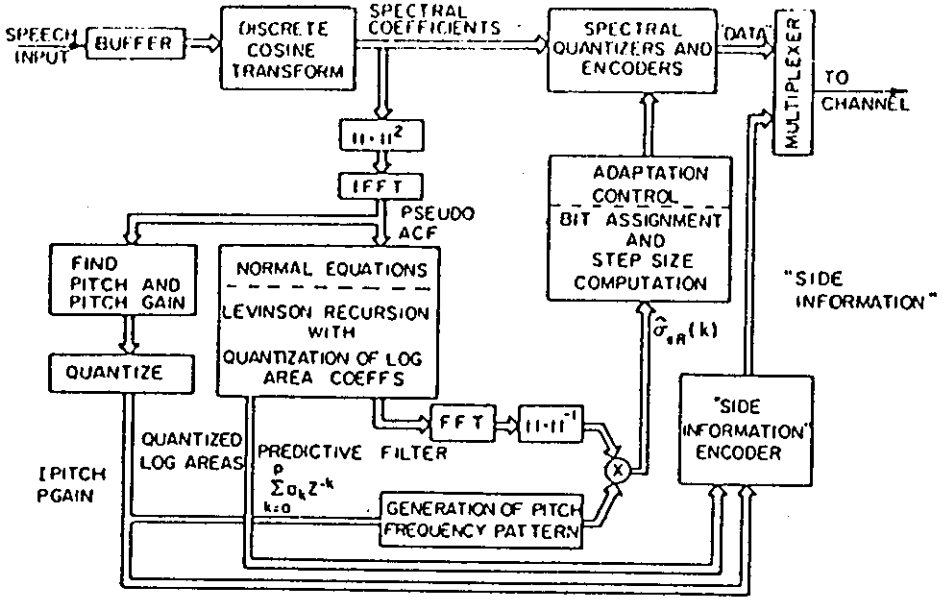
A later proposal for low bit rate ATC schemes utilises a more complex 'speech specific' adaptation algorithm based on the traditional model of speech production to predict the DCT spectral levels. The prediction involves two components as illustrated in figure 6.14. The first is associated with the spectral envelope and the second with the harmonic (fine) structure of the spectrum. This so-called vocoder driven ATC[12, 140,165,240] is able to provide a more parsimonious allocation of available bits according to the fine structure of the spectrum and thus avoid the quality degradation encountered at low bit rates. A block

diagram of the system is shown in figure 6.15. The estimate of the short-term DCT spectrum is obtained as follows. The original DCT spectrum is first squared and inverse transformed with an inverse DFT to yield a 'psuedo' autocorrelation function (ACF) rather similar to the normal ACF. The first p+1 values of this function are used to define a correlation matrix in the usual normal equations formulation sense. The solution of these equations yield s an LPC filter of order p, whose inverse spectrum provides the estimate of the formant structure of the DCT spectrum (figure 6.14(a)). The spectral fine structure is obtained from a pitch model, derived from the maximum value of the psuedo-ACF above the range p+1. The corresponding pitch gain G is the ratio of the psuedo-ACF at this maximum value, over its value at the origin. With these two parameters, a pitch pattern can be generated (figure 6.14(b)). The two spectral components are multiplied and normalised to yield the final spectral estimate (figure 6.14(c)) used in the bit assignment and step-size adaptation process.

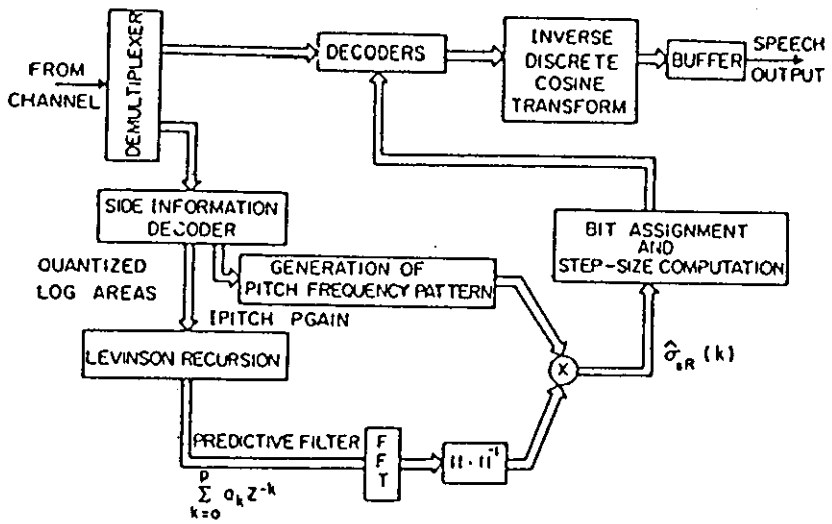
6.3.5 Computer Simulation

As we are concerned with evaluating the performance of speech coders operating at 16 Kbps, the use of the highly complicated vocoder-driven adaptive strategy is not warranted, since the simpler (although still highly complex) model of Zelinski and Noll is adequate at this bit rate. This ATC design was therefore chosen for simulation on the computer.

A 128-point DCT was used to perform the block transformation. The basis spectrum is estimated using 16 uniformly spaced support values, each obtained by averaging over 8 neighbouring transform coefficients. For example, the first support value, obtained from the average variance of



(A) TRANSMITTER



(B) RECEIVER

Fig. 6.15 Block Diagram of Vocoder-driven Adaptive Transform Coder

the first 8 coefficients is positioned at location 4, the next at location 12, then 20, and so on until location 124. These support values are then quantized with 2 bits before interpolation (on the log magnitude) is performed to obtain the complete spectral estimate. Figure 6.16 shows how this spectral estimate compares with the spectrum for a typical segment of speech. The bit allocation procedure using this spectral estimate is performed in the same way as for the sub-band coder, shown in the flow-chart of figure 6.9. The number of bits assigned to each frequency component must be rounded to the nearest integer. Excess bits are taken from the least deserving coefficients and extra bits are given to the most deserving cases in the same manner as before. With 5 bits used for coding the block standard deviation (for normalisation purposes) and 32 bits for the 16 support values, a total of 219 bits per block of 128 samples are available for distribution among the transform coefficients. Base 2 logarithm is taken of the support values before quantization to ensure a more uniform amplitude distribution. All quantizers are designed for signals with a Gaussian density[43].

The segmental SNR obtained for the male and female speech files are respectively 24.47 and 22.40 dB. Subjective quality of the recovered speech is extremely good for the male speech, where distortion is barely perceptible. For the female speech however, a slight 'buzz' can be heard in the background, due possibly to edge effects related to the use of block transforms[140].

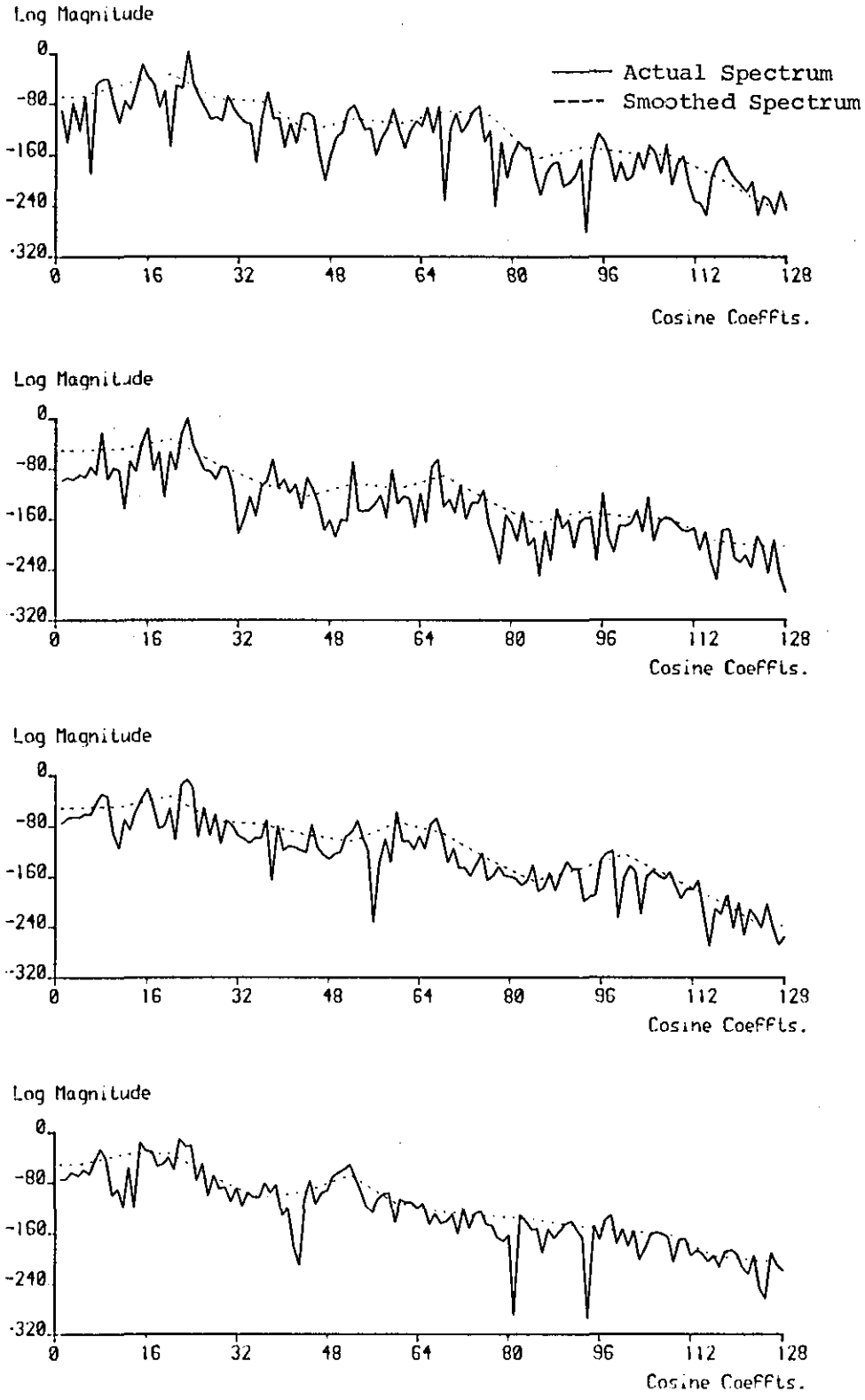


Fig. 6.16 Short-term DCT Magnitude Spectra and the Corresponding Estimated Spectra for Typical Segments of Speech

6.4 DISCUSSION

The efficiency of frequency domain speech coding has been amply demonstrated by the sub-band and transform coders described above. Much of the superiority of such frequency domain coders over their time domain counterparts, lies in the effective exploitation of the non-flat spectral density of speech and the use of different encoding accuracy for different frequency regions. This flexibility ensures that the 'usefulness' of every available bit is maximised.

Variations to the basic structure of the coders described have been proposed by several researchers, but most of these involve very minor modifications. In sub-band coding, much of the more recent research efforts have concentrated on simplifying the bit allocation process[160] and reducing side information bit rate by exploiting spatial redundancies in the signal energy[154,157]. Pitch prediction has also been incorporated in some systems[151,153,209] although the justification for this substantial additional complexity is dubious. One important development in sub-band coding has been the use of polyphase filter designs in the implementation of the QMF bank[147]. This has resulted in an appreciable reduction in the amount of signal processing required in the filtering process, compared to the direct time convolution methods.

For the adaptive transform coder, some improvement in the synthesised speech quality has been reported using a post-processing enhancement scheme on the vocoder-driven ATC[240]. Also in the same area, another notable effort seeks to reduce coder complexity by employing small (9-point) transforms[241]. These were aimed at providing good quality

speech at very low bit rates (< 10 Kbps). More recently, an attempt to bridge the gap between wide-band and narrow-band frequency domain coders came in the form of a 32 band sub-band coder[157], which uses vector quantization techniques for adapting the bit allocation and quantizer step-sizes in order to minimise side information requirement. This highly complex scheme was reported to provide comparable quality with ATC at the same bit rate.

Obviously, the advantages of these powerful techniques over time domain methods have not been achieved without a cost. Frequency domain coders are generally much more complex, and usually require long coding delays.

The use of FIR filter banks with their inherent delay, has been a limiting factor in sub-band coders. This delay and the computational complexity of the analysis/synthesis filter bank processes increase proportionately with the number of bands and could prove prohibitive even with the use of quadrature mirror filters and polyphase implementations. The sub-band coding approach for digitizing speech can thus be quite demanding in terms of both delay and complexity, especially at low bit rates(< 16 Kbps), where fine frequency resolution is essential to enable the coder to efficiently adapt to the short-term speech spectral variations. At the same time, when the number of bands is large, it becomes increasingly necessary to employ adaptive (or dynamic) bit allocation and forward block adaptive quantization (AQF) so that available bits are optimally allocated to each sub-band. This unfortunately, imposes a further delay on the system (equal to the quantizer/bit allocation update blocksize) in addition to the filter propagation delay.

The delay in the adaptive transform coder depends on the size of transform used, which is usually sufficiently large to provide adequate frequency resolution. While this delay is generally less than that of sub-band coders, the complexity of ATC is much higher, since the encoding and bit allocation processes are effectively performed for a considerably larger number of frequency bands. This complexity issue renders the otherwise powerful ATC unattractive for many applications [242,243]. A reduction in blocksize has been suggested as a possible means of coder simplification[162,241]. Unfortunately, the advantages of coding in the frequency domain also tends to be eroded when the transform size is small, and the resultant performance degradation far outweighs the reduction in complexity.

6.5 A TRANSFORM APPROACH TO SPLIT-BAND CODING

In the following sections, we propose a 'transform based' split-band coding (TSBC) approach, which permits fine spectral resolution (and therefore a more accurate representation of the short-term speech spectral variations) without the accompanying increase in delay and complexity encountered in conventional sub-band coding systems. A block transformation is used to perform the band-splitting into a number of equally or unequally spaced bands. The time signals corresponding to these bands can be coded in the same way as in SBC, using fixed or adaptive bit allocation with forward or backward adaptive quantization. The proposed method allows for a more flexible design approach to frequency domain coding, as a whole range of trade-offs between performance, delay and complexity is possible, to suit specific

applications. More importantly, the delay and complexity of the proposed system (in terms signal processing operations) is substantially reduced, compared to sub-band coders employing filter banks[201,202].

6.5.1 System Description

The generalised structure of the proposed split-band scheme is shown in figure 6.17. For simplicity, the following description will be for the case where the input signal is split into a number of uniform frequency bands.

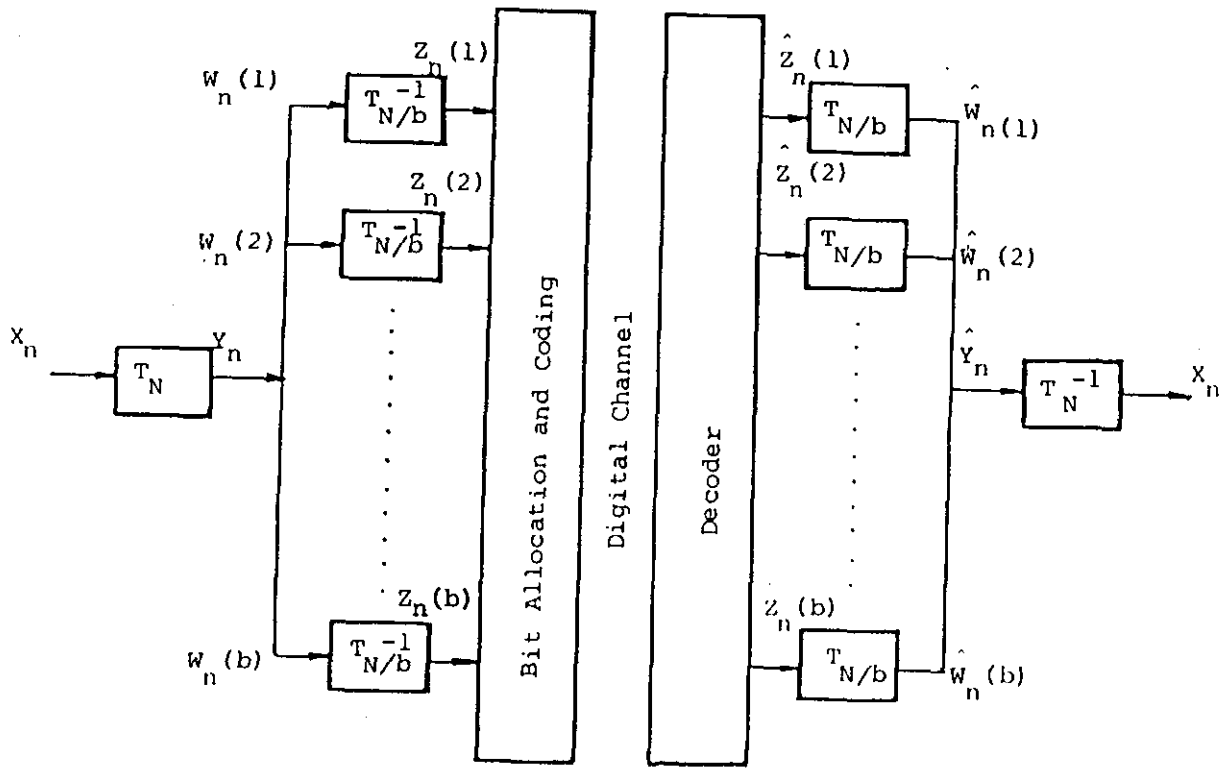


Fig. 6.17 Block Diagram of the Transform-based Split-band Coder

The sequence of input samples $\{x_n\}$ is segmented into blocks X_n, of N samples. Each block X_n is transformed via an N-point discrete cosine transform (DCT) to yield a block Y_n, of N transformed coefficients. Y_n is then divided into contiguous blocks $W_1(1), W_2(2), \ldots\ldots W_n(b)$, each containing N/b samples, where b is the number of frequency bands in the

TSBC system. Each of these smaller blocks $W_n(i)$ is separately de-transformed via an N/b point inverse DCT to give the 'psuedo' sub-band signals $Z_1(1), Z_2(2), \ldots\ldots Z_n(b)$. The energy $E_n(i)$ in each of the $Z_n(i)$ bands is computed and transmitted as side information. The quantized version of this information is used at both the transmitter and the receiver to compute the optimum number of bits assigned for the coding of each sub-band signal $Z_n(i)$, as well as the step-sizes for the individual quantizers. At the receiver, the reverse process is performed - the decoded 'psuedo' sub-band signals, $\hat{Z}_n(i)$ are forward transformed with an N/b point DCT to give the signals $\hat{W}_n(i)$. These are then combined in the correct order to form $\hat{Y}_n$. A final N point inverse DCT on $\hat{Y}_n$ yields the recovered signal $\hat{X}_n$.

The blocksize for the update of the bit allocation and quantizer step-sizes need not be equal to the transform size. When the latter is relatively small, the side information is calculated and transmitted only once over a number of input transform blocks. This is to ensure that the side information bit rate remains a fairly small proportion of the total transmission rate, so that sufficient bits are available to accurately code the sub-band signals.

The splitting of the input signal X_n can also be considered in terms of matrix operations as:

$$\underset{Nx1}{\begin{bmatrix} z_n(1) \\ z_n(2) \\ \vdots \\ z_n(b) \end{bmatrix}} = \underbrace{\begin{bmatrix} B^t_{N/b} & 0 \cdots 0 \\ 0 & B^t_{N/b} \cdots 0 \\ \vdots & \vdots \\ 0 & 0 \quad B^t_{N/b} \end{bmatrix}}_{N/b \quad N-N/b} \underset{NxN}{\begin{bmatrix} \\ B_N \\ \\ \end{bmatrix}} \underset{Nx1}{\begin{bmatrix} \\ X_n \\ \\ \end{bmatrix}} \qquad (6.15a)$$

At the receiver, the synthesis procedure to recover X_n takes the form,

$$\underset{Nx1}{\begin{bmatrix} \\ \hat{X}_n \\ \\ \end{bmatrix}} = \underset{NxN}{\begin{bmatrix} \\ B^t_N \\ \\ \end{bmatrix}} \underset{NxN}{\begin{bmatrix} B_{N/b} & 0 & 0 \cdots 0 \\ 0 & B_{N/b} & 0 \cdots 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 \cdots & B_{N/b} \end{bmatrix}} \underset{Nx1}{\begin{bmatrix} \hat{Z}_n(1) \\ \hat{Z}_n(2) \\ \vdots \\ \hat{Z}_n(b) \end{bmatrix}} \qquad (6.15b)$$

This pair of matrix equations can be represented by,

$$Z_n = \beta^t_{N/b} B_N X_n \qquad (6.16a)$$

and

$$\hat{X}_n = B^t_N \beta_{N/b} \hat{Z}_n \qquad (6.16b)$$

respectively, where B_N is the cosine basis matrix for an N point transform and B^t_N, which denotes the transpose of B_N is also the inverse N point cosine basis matrix (using the symmetrical definition of the DCT pair). $\beta_{N/b}$ represents the NxN square matrix containing b sub-matrices $B_{N/b}$ along its 'diagonal' and zeroes elsewhere. Z_n is the (Nx1) dimensional matrix formed from b (N/b x 1) dimensional sub-matrices

$Z_n(i)$, i = 1,2....b.

The value of b determines the spectral resolution (number of bands) of the split-band system, which can vary from the fine resolution provided by ATC to the 'one-band' case of waveform coding schemes. Specifically, 3 cases arise,

(i) b = N

i.e. the number of frequency bands is equal to the transform blocksize. In this case,

$$B_{N/b} = B^t_{N/b} = 1 \quad \text{(single value)} \tag{6.17}$$

and

$$\beta_{N/b} = \beta^t_{N/b} = I_N \tag{6.18}$$

where I_N is the NxN identity matrix. From (6.16a) and (6.16b) therefore,

$$Z_n = B_N X_n = Y_n \tag{6.19a}$$

and

$$\hat{X}_n = B^t_N \hat{Z}_n = B^t_N \hat{Y}_n \tag{6.19b}$$

Thus, the transform coefficients Y_n are in fact coded individually, and the system becomes an adaptive transform coder.

(ii) b = 1

Equations (6.16a) and (6.1b) yield,

$$Z_n = \beta^t_N B_N X_n = X_n \tag{6.20a}$$

and

$$\hat{X}_n = B^t_N \beta_{N/b} \hat{Z}_n = \hat{Z}_n \tag{6.20b}$$

i.e no splitting of the signal is performed and the full band signal is directly coded.

(iii) $1 < b < N$

A range of differing degrees of spectral resolution can be achieved, with b defining the fineness of resolution.

Non-uniform splitting of bands (such as octave sub-band designs) can be simply realised by dividing the transform coefficients Y_n into unequal parts before carrying out the second stage transformations.

Bit allocation and quantization is performed in the same way as in sub-band coding (sections 6.2.3.2 and 6.2.3.3).

6.5.2 Computer Simulation Results

The performance of the proposed TSBC system was evaluated using computer simulation. Many combinations of various parameters are possible. The three main variables in the system are:

(a) b - the number of frequency bands

(b) N - the blocksize of the initial transform

(c) A - the blocksize for the update of side information, which defines the rate of adapting the sub-band bit allocation and quantizer step-sizes to the short-term speech spectral variations.

Computer simulation results related to the variations of these parameters are outlined as follows:-

(1) Varying the number of bands, b

With A fixed at 256, the levels of the output noise spectra for various

values of b are shown in figure 6.18 for transform sizes 128 and 64. It can be seen that the level of output noise is generally inversely related to the number of bands, except for the case when b = 32. This is due to the fact that increasing b increases the side information requirements (for the same update blocksize A) so that proportionately fewer bits are available to code the subband signals. The allocation of 5 bits for the energy of each band takes up more than 25% of the available bits, leaving less than 75% to code the $Z_n(i)$ signals, which in turn leads to poorer performance. Thus, in this case, the advantage of better spectral resolution is partially offset by lower encoding accuracy, due to the proportionately larger side information requirements.

(2) Varying the Transform size, N

With A again fixed at 256 samples, the effect on the output noise spectra, of the variation of the transform size N is demonstrated in figure 6.19 for the 8 and 16 band cases. Not unexpectedly, the noise level is reduced when the transform blocksize N increases, since N determines the fineness of frequency resolution of the first block transformation.

(3) Varying the blocksize A, for bit allocation and quantization

Fixing the transform size N to 64, the blocksize A is varied for the 16 and 32 band cases (see figure 6.20). Increasing A reduces the side information requirements and hence releases more bits to quantize the sub-band signals, resulting in lower output noise as shown. However, this only applies to a certain extent, since the accuracy of the quantizer step-size estimate is also reduced when A is excessively large. Additionally, the delay associated with a large A may be a more immediate constraint in practical terms.

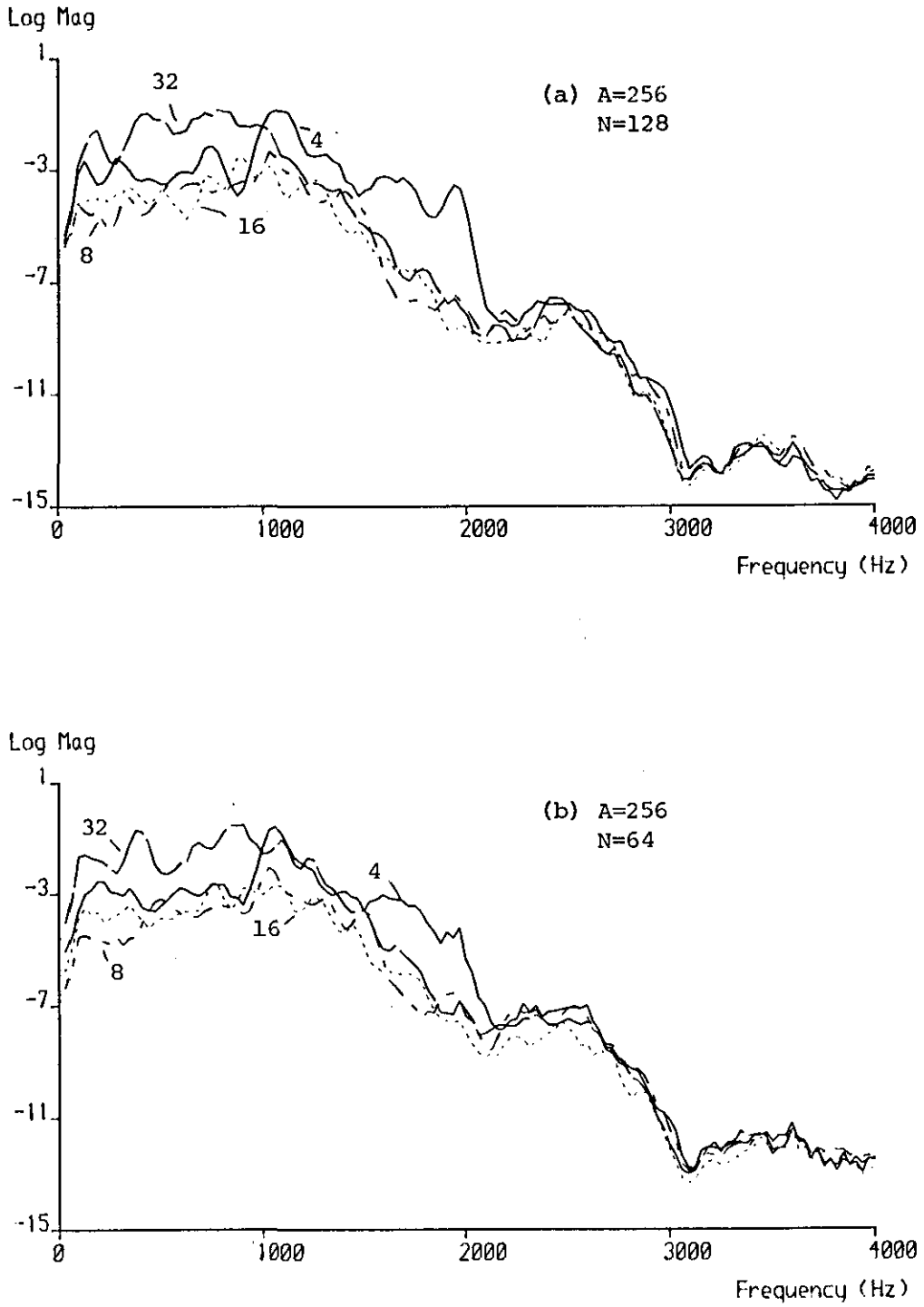


Fig. 6.18 Output Noise Spectra of TSBC System
Variable Number of Bands, b

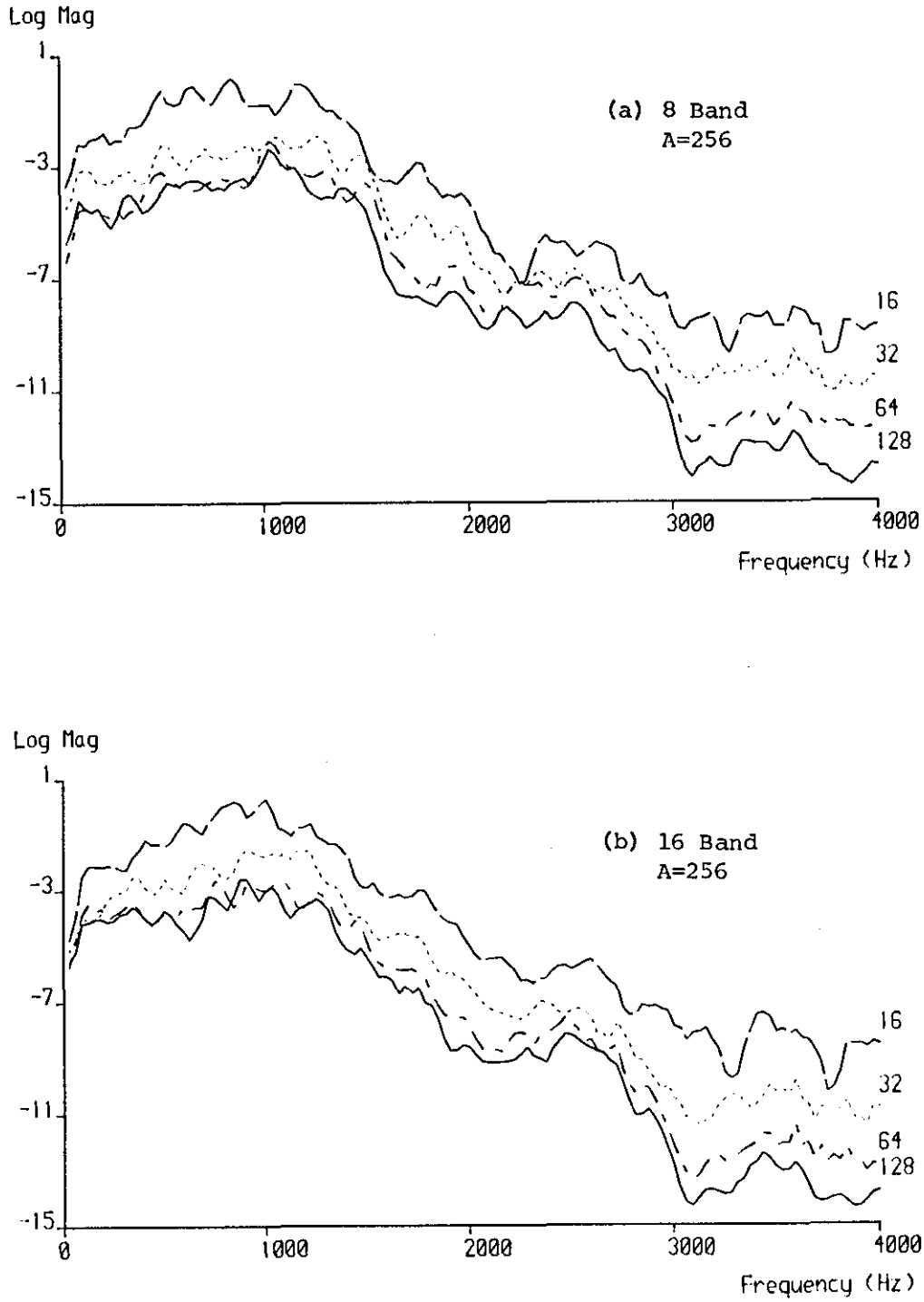


Fig. 6.19 Output Noise Spectra of TSBC System
Variable Transform Size, N

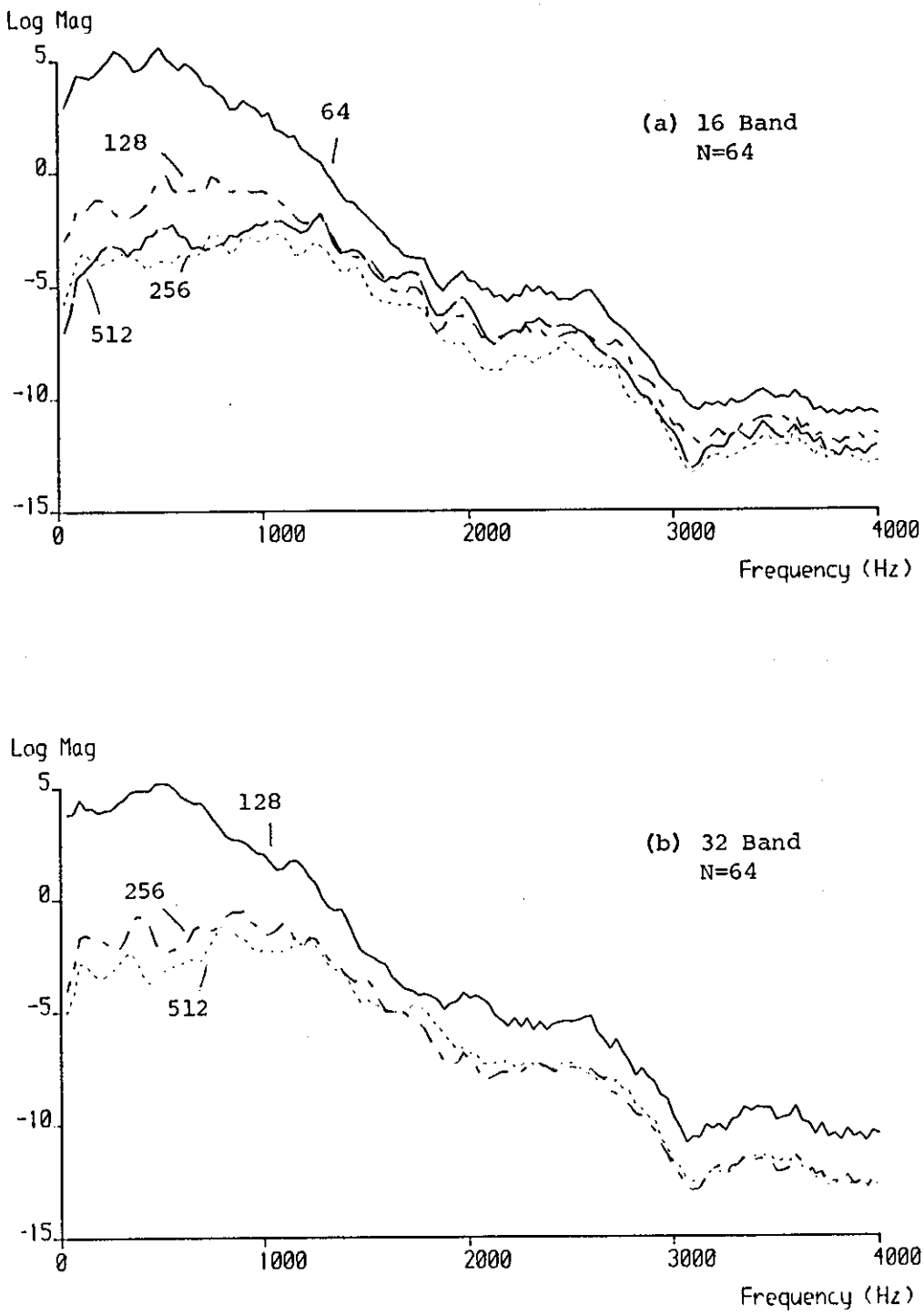


Fig. 6.20 Output Noise Spectra for TSBC System
Variable Update Blocksize, A

A summary of computer simulation results of the proposed coder in terms of segmental SNR performance is shown in figure 6.21. The performance of the conventional QMF sub-band coder is also included for comparison. It can be seen that the proposed split-band scheme compares favourably with the SBC when the transform size is fairly large. As a further comparison, figure 6.22 shows the output noise spectra of the conventional 16-band SBC, the ATC and ADPCM together with the proposed split-band coder for N=128, b=16 and A=256 (all at 16 Kbps). The ADPCM coder employs second order forward adaptive prediction[20] (with a blocksize of 256 samples) and the one-word memory quantizer[49]. Clearly, the average noise level of TSBC is comparable to the SBC and ATC. Figure 6.22 also illustrates the superiority of frequency domain coders over simple time domain ADPCM schemes.

The results of the noise spectra corresponds closely to the perceptual quality of the received speech. The subjective performance of TSBC is generally comparable to the SBC but the perceived distortion is different. In the SBC, a 'whispery' distortion and 'hollowness' (section 6.2.3.4) is present in the perceived speech when the number of bands is small. For the proposed scheme, the 'whisper' is present to a lesser extent but a further low amplitude 'buzz' is audible, and becomes gradually more apparent as N and/or b is reduced. This 'buzz' is also discernible in the ATC, particularly for the female speech (see section 6.3.5) and is possibly due to edge effects related to the use of block transforms. However, for reasonably high values of N (e.g. 64, 128) and a large number of bands (8,16), this distortion is barely perceptible using headphones and only slightly audible over the

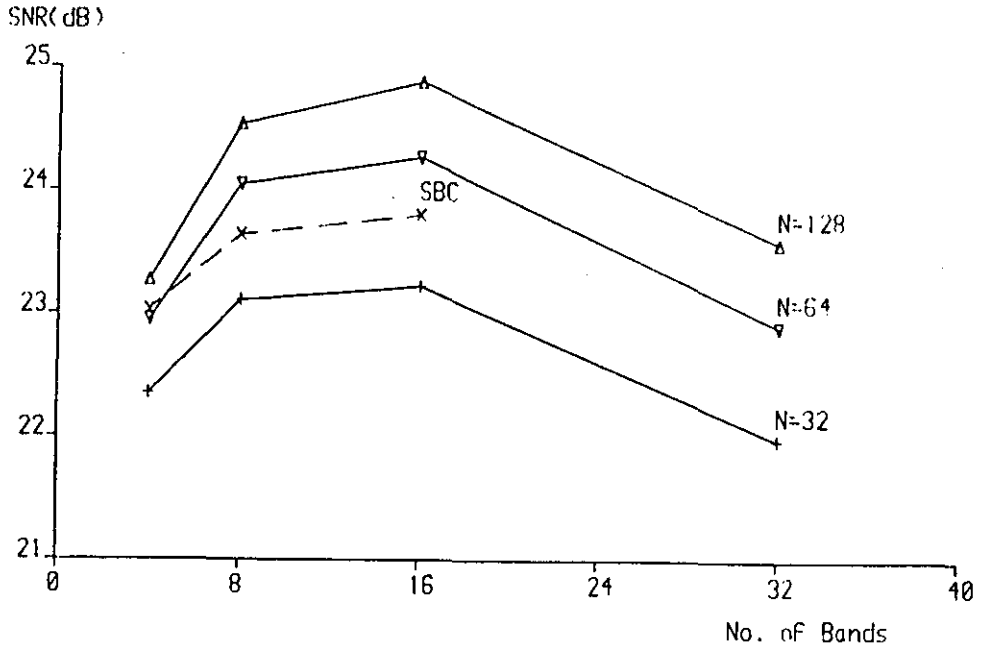


Fig. 6.21 Segmental SNR Performance of TSBC Schemes (A=256)

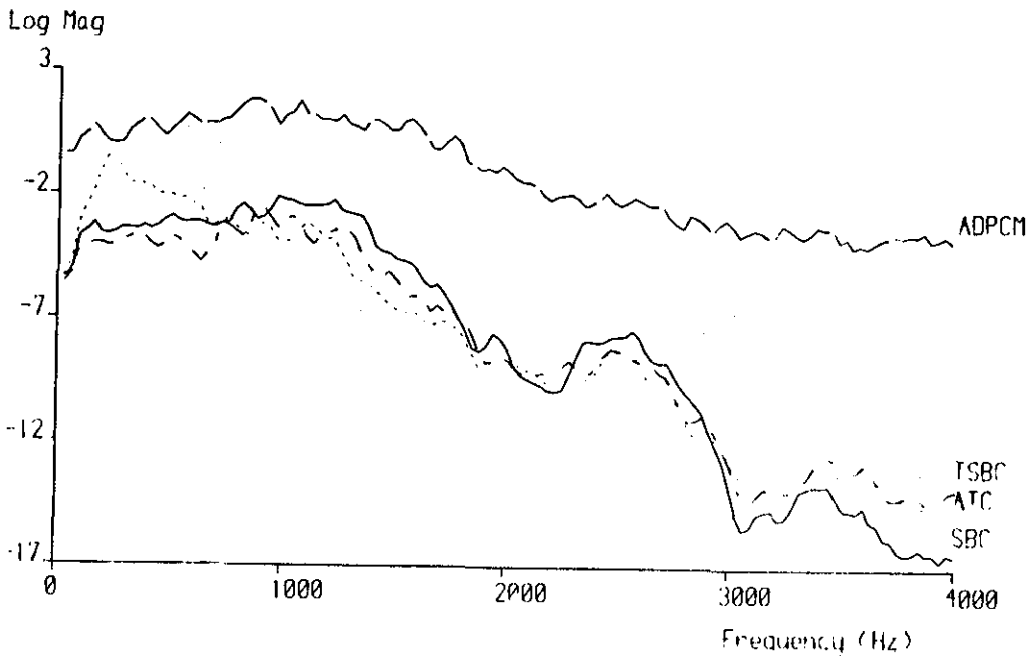


Fig. 6.22 Output Noise Spectra of Various Speech Coding Schemes

loudspeakers. As the rate of update of side information increases, proportionately fewer bits are available to code the sub-band signals, and the result is a 'burbly' distortion similar to that obtained with ATC at low bit rates[162]. Also, when the transform size, N is reduced, spectral resolution of the main transformation becomes coarser, and the recovered speech possesses a certain 'roughness'.

6.5.3 Discussion

A realistic assessment of speech coding schemes must necessarily consider aspects of practical implementations. Good performance is obviously the primary aim of any coding system, although this must be weighed against the complexity involved (which determines the 'implementability' and cost), the robustness to transmission errors, and the delay required, amongst other factors.

The analysis/synthesis transform approach to split-band coding proposed here involves shorter delays and requires a significantly smaller amount of computation, compared to SBC schemes operating under the same conditions. These two factors are discussed in greater detail in the following.

6.5.3.1 Delay

The delay in the sub-band coder consists of two components:-

(1) the analysis/synthesis propagation delay through the quadrature mirror filter banks, given as (b-1)(T-1) samples for a b band SBC using uniform T-tap filters. For example, a 16 band SBC employing 16 tap QMFs, will incur a delay of 225 sampling periods.

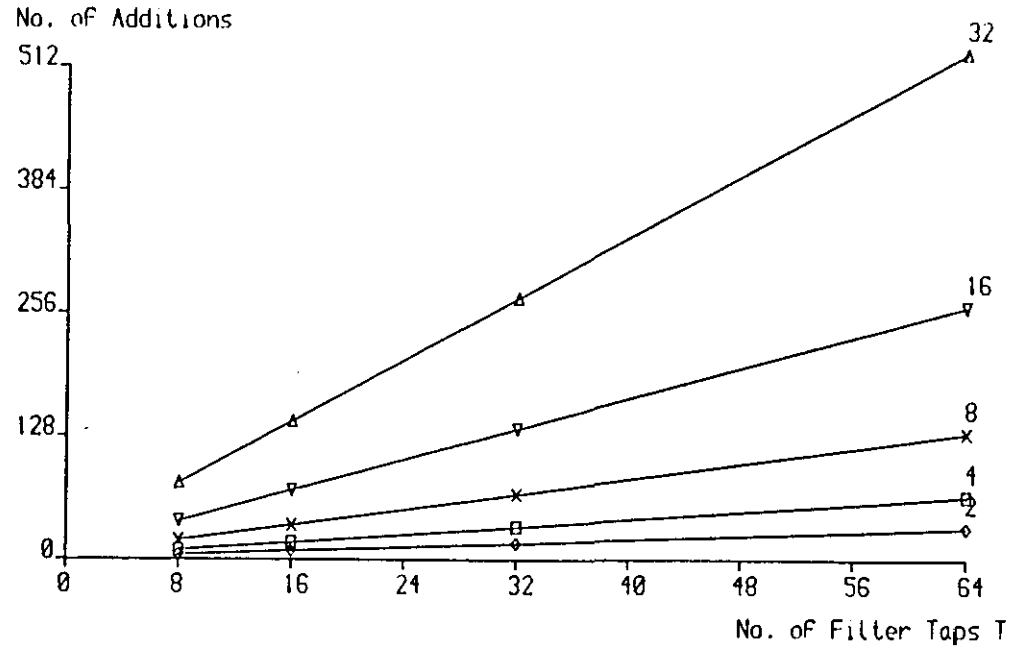
(2) the delay introduced by forward adaptive bit allocation and quantization of the sub-band signals. This is defined by the parameter A.

The delay in the proposed scheme is independent of b and is significantly lower than that imposed on the corresponding sub-band coder since the band splitting and bit allocation processes can be performed within the same block A, and no additional filter propagation delay arises. For the same 16 band example, there would be a reduction in delay of 225 samples (~28 ms), assuming that the time taken to perform the band-splitting is relatively insignificant.

6.5.3.2 Complexity

The complexity of an algorithm is normally considered in terms of the amount of signal processing involved and the storage required. A reasonable measure of signal processing requirements is the number of multiplication and addition operations employed per sample of the input signal.

Quadrature mirror designs[145] of FIR filters and their polyphase[147] implementation allow the number of filtering operations in sub-band coders to be appreciably reduced over the case where direct filtering is employed. Specifically, the filtering process involves $T/2 \log_2 b$ real multiplications and $(T/2+1)\log_2 b$ real additions (see appendix G). Several fast algorithms exist for the computation of the DCT[236-238]. One such algorithm[236] entails $3N/2(\log_2 N - 1) + 2$ real additions and $N\log_2 N - 3N/2 + 4$ real multiplications for an N point DCT. For the two stage transformation employed in the proposed scheme, the computational requirements per input sample are: (see appendix H)



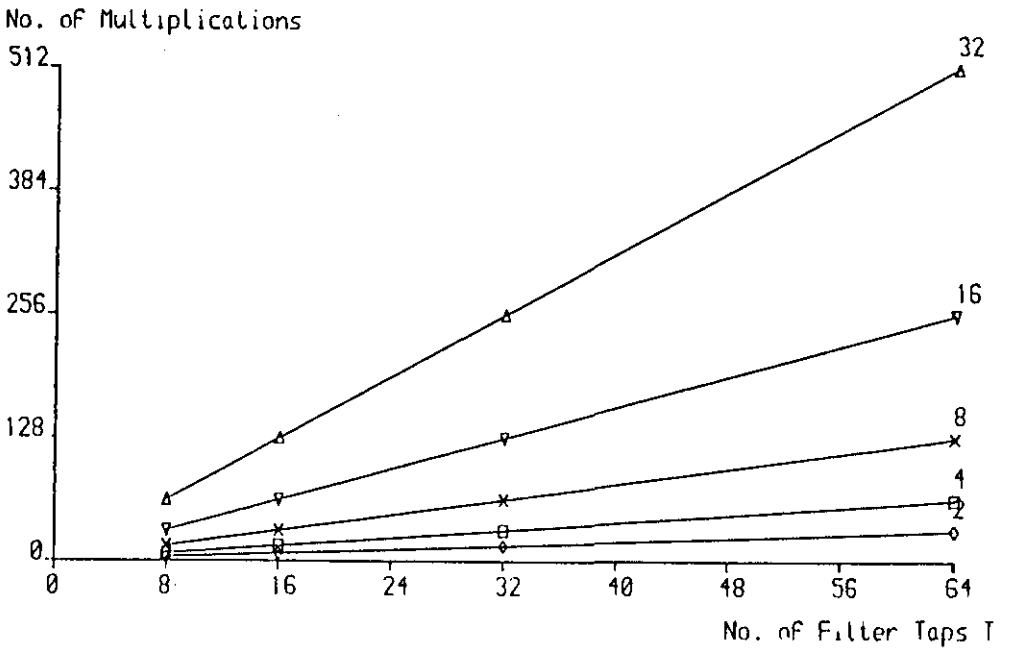
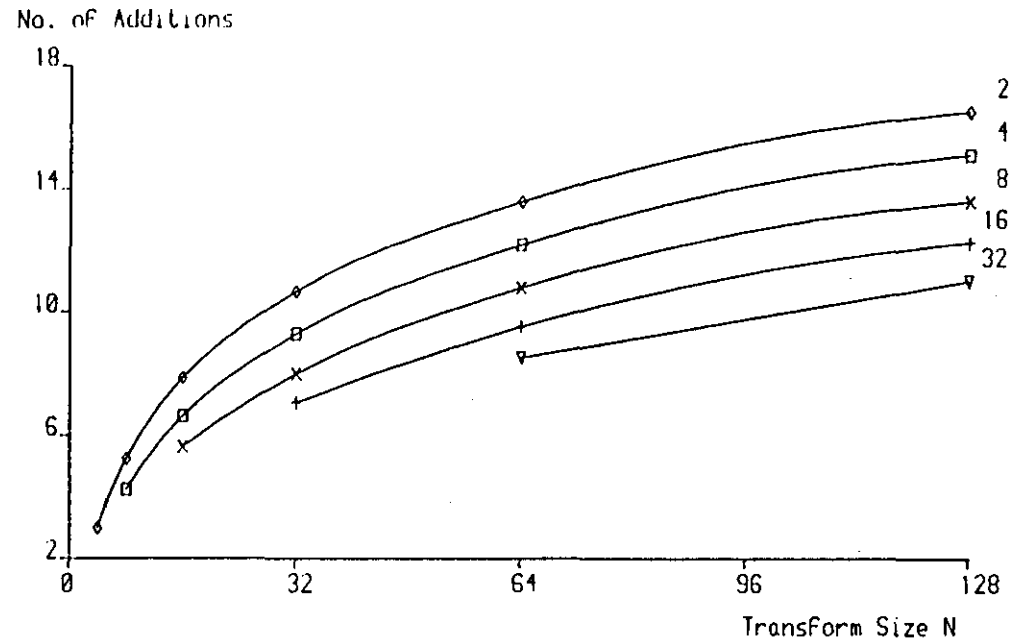


Fig. 6.23 Computational Requirements of SBC and TSBC Systems
(a) Sub-band Coder



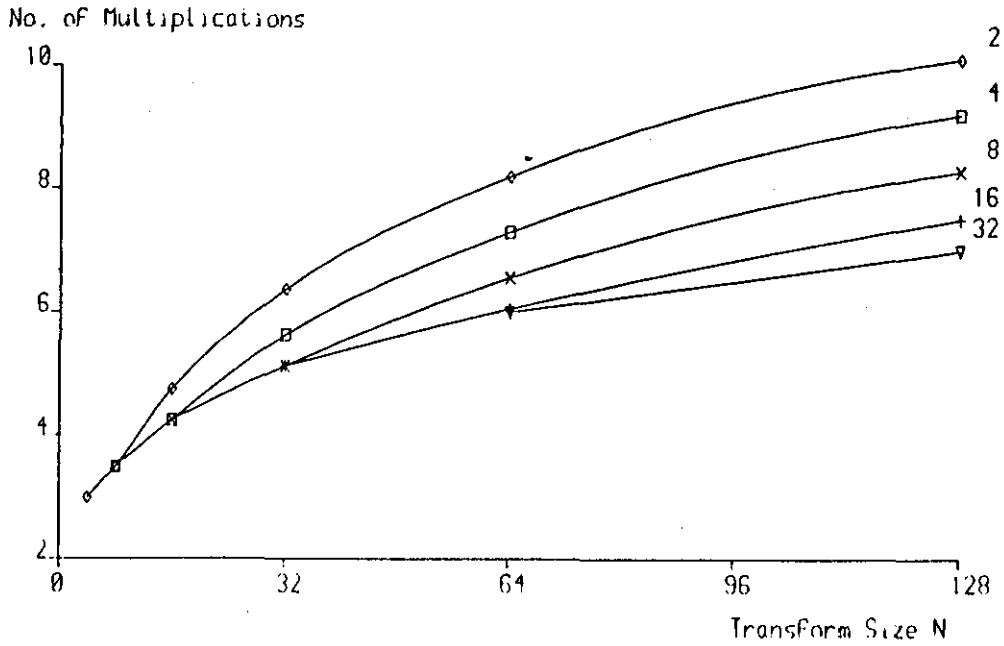


Fig. 6.23 (b) TSBC System

$3\log_2 N - 3 - 3/2\log_2 b + 2(b+1)/N$ real additions and $2\log_2 N - 3 - \log_2 b + 4(b+1)/N$ real multiplications. The computational requirements for the two methods of band splitting are shown in figure 6.23. Note the different scales on the vertical axes. It is clear that the effective number of signal processing operations (per input sample) in the proposed scheme is substantially lower than that of the sub-band coder. This is a significant advantage in implementation terms.

The storage requirements of the sub-band coder increase proportionately with the number of bands and the length of filters used since all intermediate samples propagating through the filters must be retained. Assuming equal length filters at all stages, this storage requirement is given by $(b - 1)T$ real locations. Obviously, if forward block adaptive quantization is employed, there would be further demands on memory storage determined by the blocksize of adaptation. The fixed memory requirements for storing the filter coefficients are relatively modest - because of the symmetrical properties of QMFs, only one half (T/2) of the filter coefficients need to be stored[147].

For the proposed scheme, the cosine basis functions of the DCT matrix need to be stored in fixed memory. However, only the values in the first quadrant are required, since the other functions are obtainable via symmetry[244]. For a N point transform, N fixed storage locations are required. The dynamic memory requirement is determined by the size of transform used and the update blocksize. Again, for forward quantization and bit allocation, the 'block transform' approach allows considerable memory to be 'shared' between the split-band analysis and the adaptive bit allocation process. The dynamic memory requirements of TSBC is therefore, generally lower than that of the sub-band coder.

6.5.4 Note on Publications

A paper entitled, "A Transform Approach to Split-band Coding Schemes" written in co-authorship with Dr. C.S. Xydeas has been accepted for publication in the IEE Proceedings on Communications, Radar and Signal Processing (Pt. F). This covers the work described in section 6.5 and some parts of section 6.6 below.

A shorter version of the paper entitled, "Split-band Coding of Speech Signals Using a Transform Technique" has been submitted for consideration to the International Conference on Communications (ICC '84) to be held in Amsterdam on May 14-17, 1984.

6.6 FURTHER CONSIDERATION ON BIT ALLOCATION AND QUANTIZATION

The effect on system performance of each of the parameters in the TSBC scheme has been demonstrated in the preceding sections. Generally, the performance of the coder improves (to a limit) with the transform size, N, the number of bands, b and the blocksize for parameter adaptation, A. On the other hand, complexity and delay also increase in the same direction. In practical implementations, there is inevitabaly a trade-off in terms of performance, complexity and delay, and the TSBC scheme offers a flexible design approach to satisfy a range of constraints. Some room for similar manouvre also exists for the conventional QMF sub-band coder, albeit to a lesser extent. The number of bands, the length of filters used and the rate of parameter adaptation can all be controlled.

In this section, we consider some issues related to the bit allocation and quantization procedures for sub-band signals relevant to both SBC

and TSBC schemes.

6.6.1 Forward and Backward Adaptation Variations

Forward adaptive quantization of the sub-band signals although undoubtedly efficient, becomes progressively less attractive as the number of bands employed increases. This is because the side information requirements also become increasingly non-trivial and coding accuracy can be seriously affected. A further disadvantage associated with all forward adaptive schemes is of course the need for a delay.

Fixed bit allocation, if used together with backward adaptive quantization offers a distinct advantage in terms of available bits for coding the sub-band signals (as no side information is required) and a reduction in coder delay. Unfortunately however, as noted previously, the inability to track the short-term frequency variations in the input signal imposes a severe limit to performance, especially with a large number of bands. Also, in such cases, a significant proportion of available bits are 'tied up' by the high frequency bands (to prevent loss of bandwidth) leading to a reduction in overall coding efficiency.

Backward adaptive bit allocation with backward quantization, which offers the promise of dynamic assignment of bits without the need for side information is an attractive proposition. The bit allocation can be made to vary according to the relative energy composition of previously decoded sub-band samples. Unfortunately, although theoretically possible, most conceivable forms of backward bit allocation adaptation would be extremely sensitive to transmission errors. Once the bit allocation pattern in the receiver is not matched

to that at the transmitter, the system collapses unless some form of recovery is incorporated (which inevitably means more complexity and loss of performance).

Another possible combination is to employ forward adaptive bit allocation with backward quantization. In this case, the adaptive bit allocation process is performed at the transmitter and the bit allocation map is communicated to the receiver. This method would retain the advantage of optimum bit allocation, with reduced side information and lower receiver complexity (as the bit allocation procedure need not be repeated at the receiver). The reduction in side information arises because, unlike the signal variances which must be fairly accurately quantized, the information concerning the bit allocation pattern can only take on a very limited range of integer values, and thus can be transmitted with a smaller number of bits.

Figure 6.24 shows as an example, the histograms for the number of bits assigned to each sub-band for the 8-band TSBC scheme, using a transform size of 128, with the bit allocation updated every 32 ms (256 samples). It can be seen that generally, the bit information for the lower sub-bands of the signal can be coded with 2 bits (4 possible values) while the same information related to the higher part of the spectrum requires no more than 1 bit. This provides a saving of 3 to 4 bits for each band, compared to the case where the average energy of each band is coded with 5-bit accuracy. The saving is substantial when the spectral resolution is high, as in the 32 band case, where the increased side information can seriously impair coding efficiency. This method of transmitting the bit allocation map may be considered as a simple form of vector quantization (see section 2.4.1.8), where the codebook

contains a set of all bit allocation patterns of practical interest, and a codeword is transmitted once per block of samples to indicate which pattern is to be used.

A potential problem exists with the use of instantaneous backward adaptive quantizers (such as the AQJ[49]) with adaptive bit allocation. The adaptation algorithm of AQJ requires a minimum of 2 bits to allow the step-size to adapt to the magnitude variations of the quantizer input but the high frequency bands are often only assigned one bit. One method to overcome this difficulty uses the 1 1/K bit quantizer[142], where the sign information is transmitted with one bit every sampling instant, while the magnitude is encoded with an additional bit every K samples. We propose another method where an approximation for the magnitude of the 1 bit AQJ output is obtained from a suitably scaled version of the output of one of the lower bands. The actual ratios for scaling depend on the energy in the reference band (which would be indicated by the number of bits assigned to it) and can be optimised from long-term statistics. Using this technique, the important zero-crossing (sign) information is preserved for these high frequency bands and the magnitudes follow a scaled down version of the signal envelope of one of the lower bands. The use of forward adaptive bit allocation with AQJ is probably more relevant to the TSBC system, since no advantage in terms of a reduction in delay is available for the SBC.

Hybrid methods of quantization may also be used, where some bands (especially the 1-bit high frequency bands) are coded with AQF, while the lower bands use AQJ - the particular design chosen would obviously depend on the environment and application. For the TSBC scheme, one way of avoiding long coder delays is by using smaller transform sizes, if

the resulting degradation is tolerable. Unfortunately, apart from degraded performance, a smaller blocksize means more overhead side information and therefore less bits available for coding.

6.6.2 Parallel Bit Allocation

Typically about half the delay incurred by the SBC is due to the use of forward adaptive bit allocation and quantization, since the adaptation is based on the outputs of the QMF bank. While this delay may be avoided by employing fixed bit allocation with AQJ, the resultant degradation in performance is unfortunately far from acceptable.

We examine a method by which this delay can be eschewed by attempting the bit allocation for the sub-band signals during the necessary time delay incurred by the QMF bank. Figure 6.25 illustrates how this 'pipe-line' or parallel bit allocation approach operates, compared to the conventional 'serial' method. For an 8 or 16 band SBC, the delay due to the QMF analysis bank is typically about 15 ms, which is a suitably long time for bit allocation and quantizer adaptation.

This parallel bit allocation can be carried out by performing a spectral analysis on the input signal segment, while it is propagating through the analysis filter bank. One way to do this is by using the discrete Fourier transform. The short-time Fourier spectrum of the input speech segment provides an estimate of the energy distribution in the various frequency bands. The accuracy of estimation might not be sufficiently high to permit the use of AQF (with transmission of step- sizes) although the relative energy composition of the various bands should be adequate to provide a bit allocation pattern (for use with AQJ) which

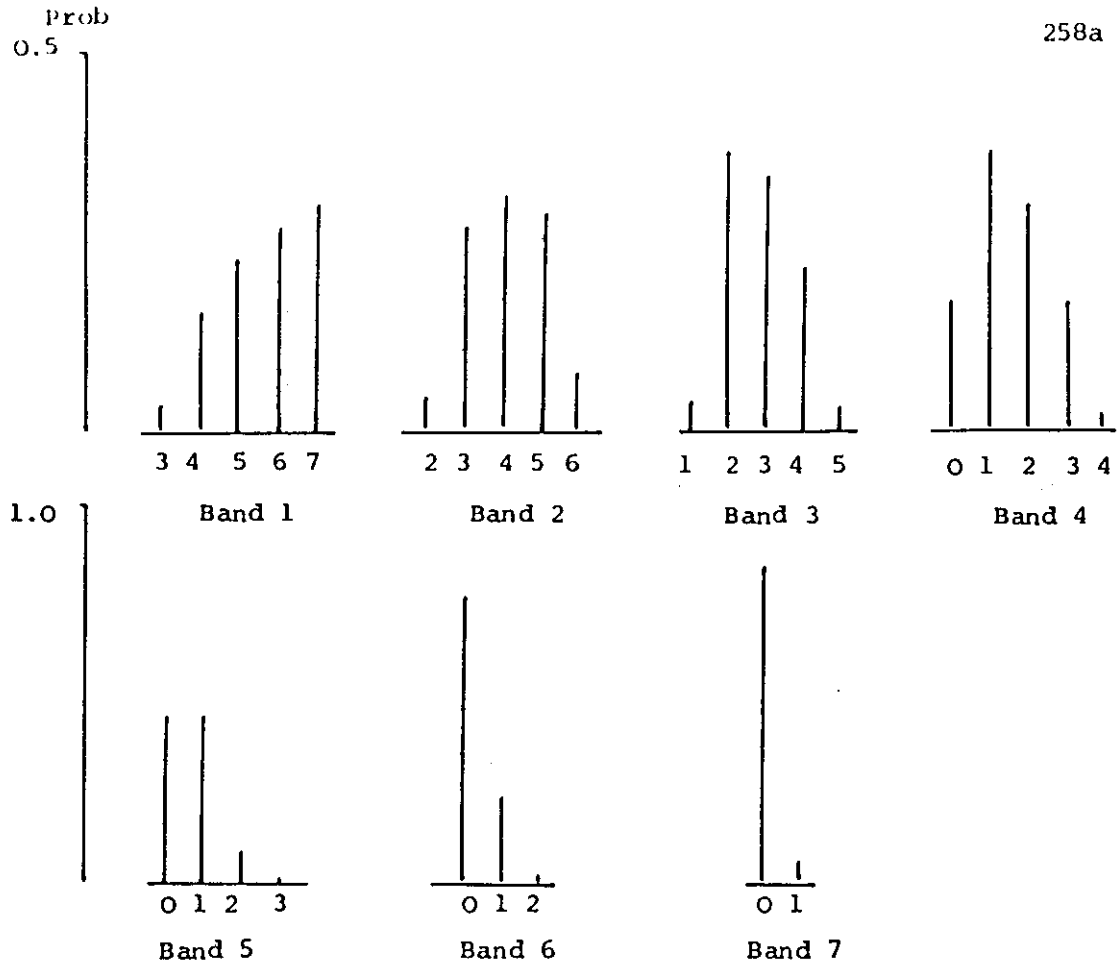


Fig. 6.24 Histograms of Bit Allocation for 8 Band TSBC
(N = 128, A = 256)

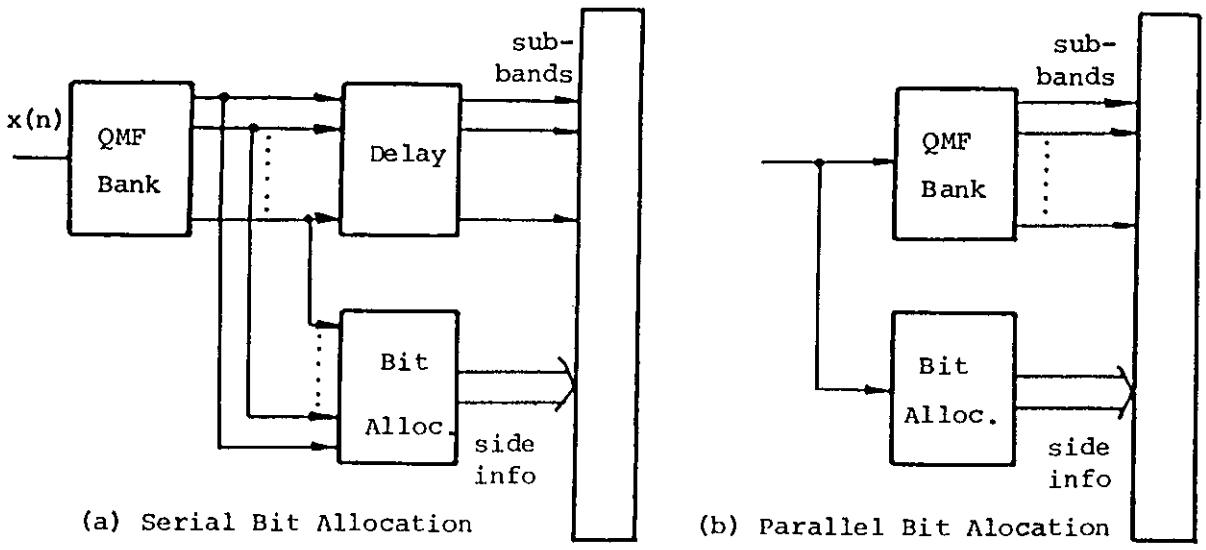


Fig. 6.25 Serial and Parallel Bit Allocation for Sub-band Coder

reflects the dynamic spectral variations of the input signal.

6.6.3 Computer Simulation

The adaptation strategies discussed above are investigated via computer simulation.

The proposed method of scaling the 1 bit AQJ magnitude to the output of a high energy reference band is first examined. The relationship between the energy of bands assigned 1 bit, to that of the reference band must be determined to obtain a suitable scaling constant. This scaling ratio depends on a number of factors:

(1) the number of bands in the coder,

(2) the position of the 1 bit band with respect to the reference band, since bands assigned 1 bit might have different energy levels at different parts of the spectrum (The errors of rounding to the nearest integer in the bit allocation process would be expected to be largest for the 1 bit bands); and

(3) the number of bits assigned to the reference band.

Experiments were carried out on both the SBC and TSBC. In each case, adaptive bit allocation is performed on the sub-band signals, a reference band is selected, and the ratio of the variance of bands assigned 1 bit to that of the reference band is obtained. To maximise estimation accuracy, the reference band must be a band with consistently high energy. Consequently, the first band is chosen as the reference for the 4 and 8 band coders, the second band for the 16 band coder and the third band for the 32 band case. Fortunately, the experiments performed revealed very little variability in the scaling ratios. The ratio between the energy of the 1 bit band to that of the reference

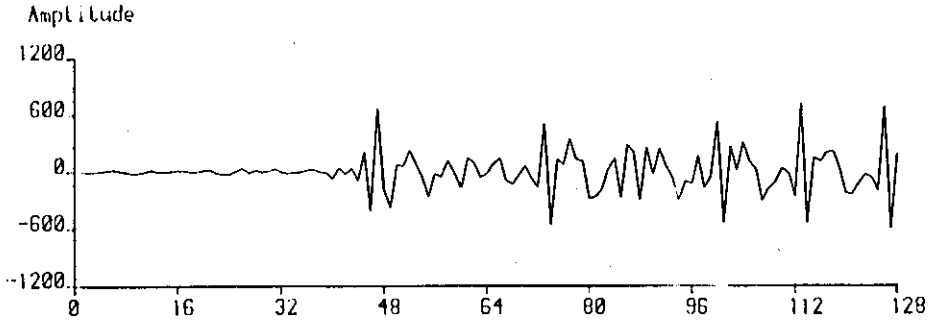
appears to be quite independent of the former's position in the frequency spectrum, as well as to the number of bands employed in the coder. A simple table based on long-term statistics can thus be drawn up (table 6.4). This table of scaling factors are used in all subsequent simulations.

Table 6.4 Scaling Constants for the One-bit Sub-band

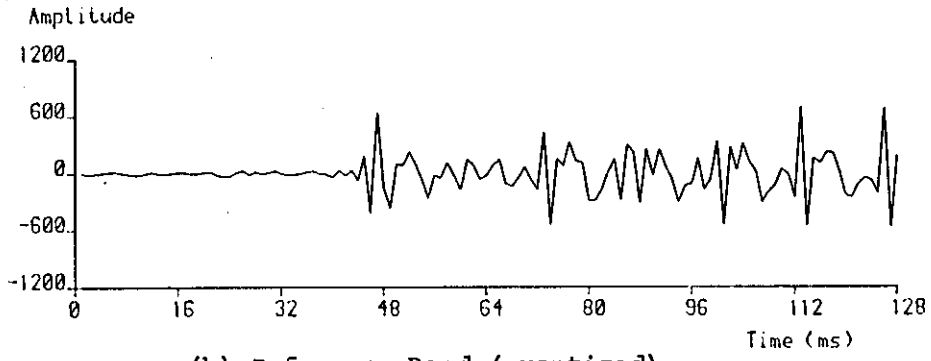
No. of Bits in Reference Band	3	4	5	6	7
Scaling Factor for 1-bit Band	0.29	0.12	0.06	0.03	0.015

Jayant[49] provided the optimum multiplier values for the AQJ algorithm for 2,3,4 and 5 bit quantizers. As the maximum number of bits used in the sub-band schemes is 7, the multipliers related to the 6 and 7 bit AQJ have to be determined. Experiments were performed for this purpose and the optimum multiplier values obtained empirically (maximum SNR) are shown in table 6.5.

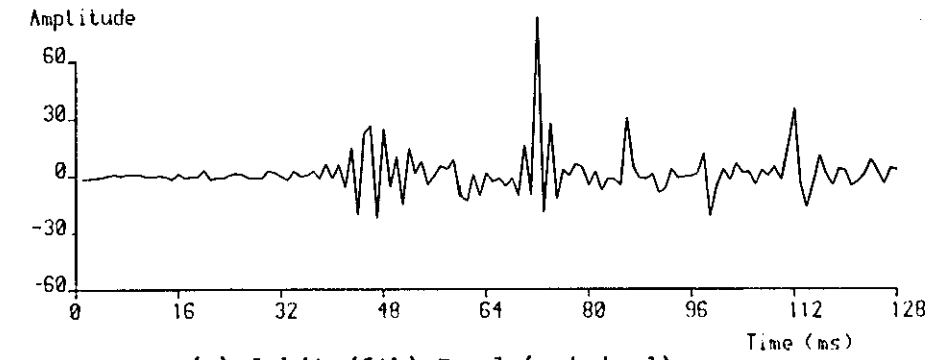
Figure 6.26 illustrates how the method of scaling the 1 bit AQJ output to a reference band compares with the original unquantized signal. The example is for the sixth band of a 8 band SBC when the reference (first) band is assigned between 5 and 7 bits. Notice that the signal envelope for the 1 bit band has been reasonably well preserved.



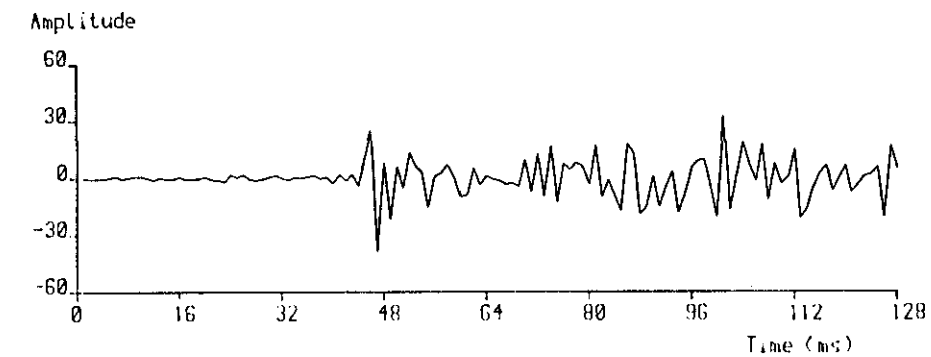
(a) Reference Band (original)



(b) Reference Band (quantized)



(c) 1 bit (6th) Band (original)



(d) 1 bit Band (scaled to Reference Band)

Fig. 6.26 Illustration of the Scaling of the 1 bit AQJ Output to a Reference Lower Band

Table 6.5 Optimum Multiplier Values for 6 and 7 Bit AQJ

(a) 6 Bit

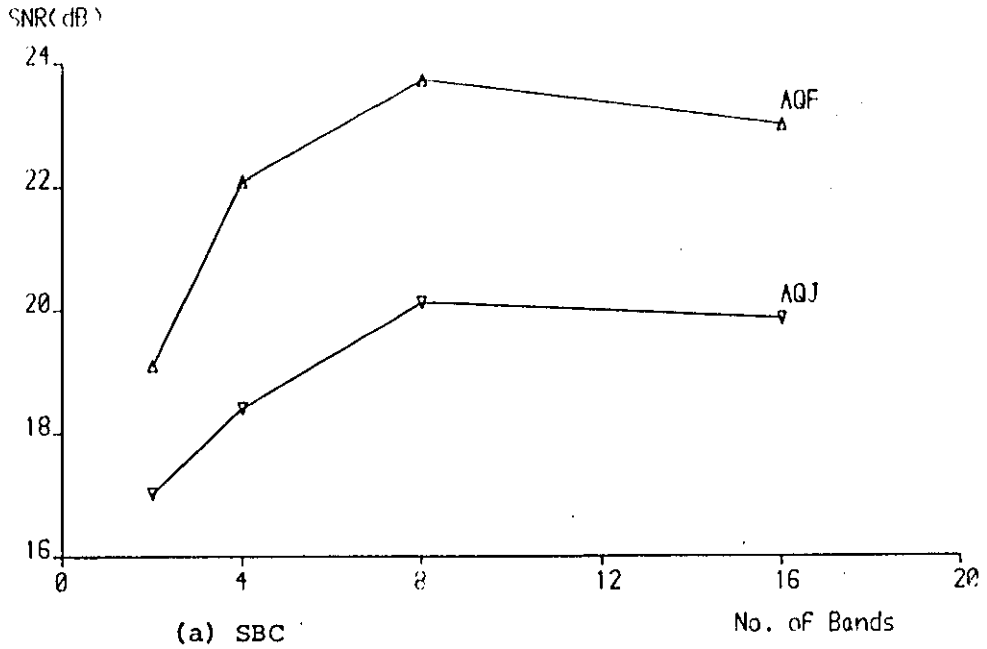
0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8
1.9	2.0	2.1	2.2	2.4	2.6	2.8	3.0

(b) 7 Bit

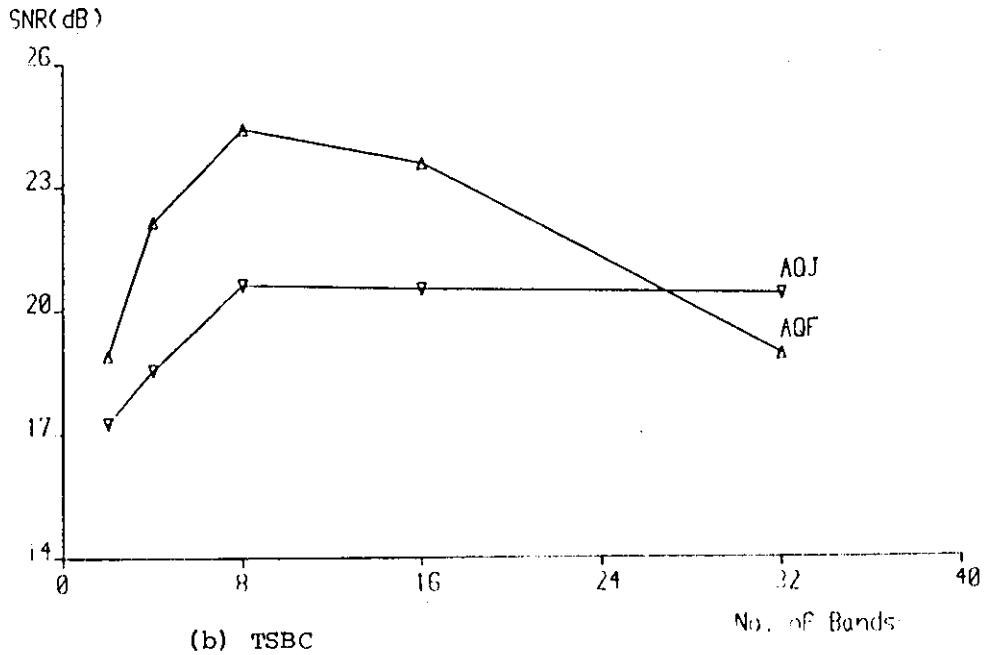
0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
1.05	1.10	1.15	1.20	1.25	1.30	1.35	1.40
1.45	1.50	1.55	1.60	1.65	1.70	1.75	1.80
1.85	1.90	1.95	2.00	2.1	2.2	2.3	2.4
2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.2

Figure 6.27 shows the SNR performance of SBC and TSBC using AQF with forward transmission of the sub-band variances, and AQJ with vector quantization of the bit allocation pattern. It can be seen that for the SBC, the use of AQF results in better SNR performance for all cases, even though less quantizer bits are actually being used for coding the sub-band signals, compared to the AQJ case. The advantage of explicit transmission of quantizer amplitude information appears to be much greater than the less accurate instantaneous magnitude adaptation of AQJ. For the TSBC systems, the same observation can be made. Note however, the better performance of the 32 band case using AQJ. This is due to the fact that the coding efficiency for the sub-band signals is quite seriously affected by the heavy side information incurred by the 32 band coder using AQF.

The output noise spectral plots provide the same observations. Figure 6.28 shows the output noise level for the 16 band SBC for three cases:

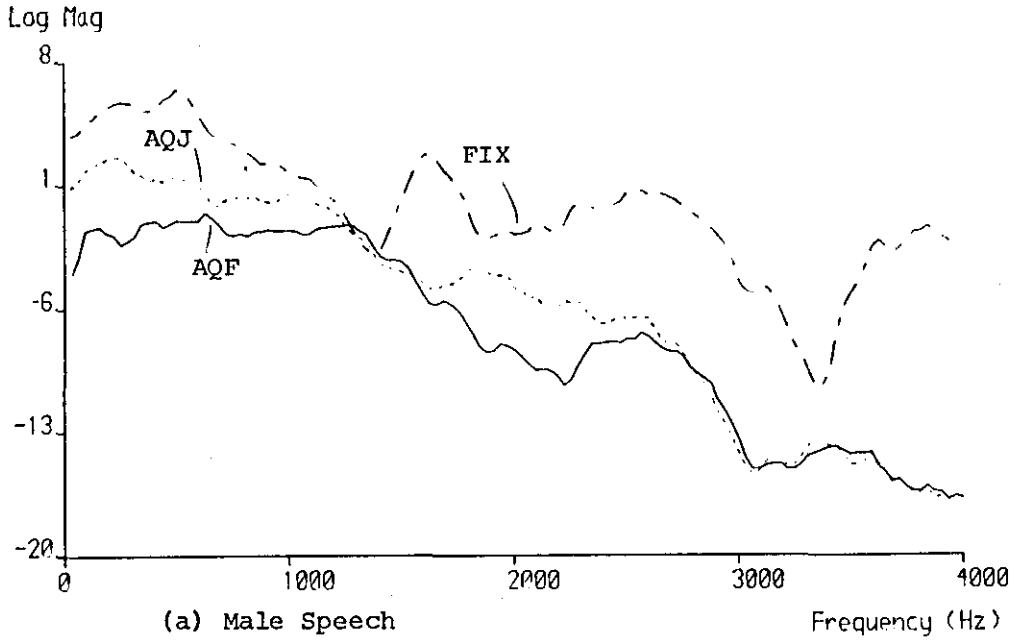


(a) SBC

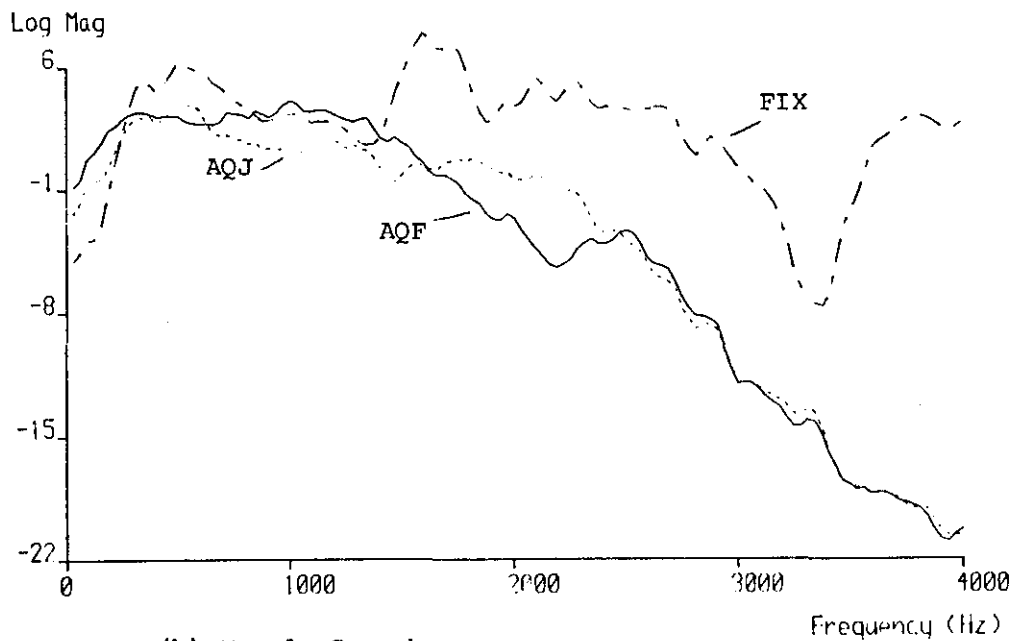


(b) TSBC

Fig. 6.27 Segmental SNR Performance of SBC and TSBC Schemes Employing Adaptive Bit Allocation with (i) AQF (ii) AQJ



(a) Male Speech



(b) Female Speech

Fig. 6.28 Output Noise Spectra for SBC Schemes (16 Bands, A=128)
(i) Fixed Bit Allocation with AQJ
(ii) Adaptive Bit Allocation with AQF
(iii) Adaptive Bit Allocation with AQJ

(a) fixed bit allocation with AQJ (b) adaptive bit allocation with AQF and (c) adaptive bit allocation with AQJ. The parameters are updated every 128 samples. Much of the advantage of AQF over AQJ occurs in the low frequency region as can be seen from the figure. Also, this performance advantage appears to be greater for male than for female speech. A whole set of results was obtained for all combinations of coder parameters. The general observation is that the use of AQJ leads to a drop in SNR in all cases (except the 32 band TSBC) and an increased 'burbly' distortion in the synthesised speech, similar to that obtained with ATC at low bit rates.

The application of parallel bit allocation is examined in relation to the 4, 8 and 16 band SBCs for which the QMF analysis stage incurs delays equal to 46.5 103.5 and 112.5 samples, respectively. To enable the use of the fast Fourier transform (FFT) for the frequency analysis, the blocksize of the discrete Fourier transforms used is chosen to be a power of two, nearest to the filter delay i.e. 64 for the 4 band coder and 128 for the 8 and 16 band cases. This gives a delay due to quantization and bit allocation, of about 17, 24 and 5 samples, respectively.

To observe the estimation accuracy of the DFT, the variances of the actual sub-band signals (derived from the outputs of the QMF analysis bank) are compared with the variances estimated by the DFT, for a number of contiguous blocks of the the input signal. This is shown in figure 6.29 for the 8 band SBC. It can be seen that the estimation of the sub-band signal variance is reasonably accurate for the low frequency bands. For the higher frequency regions however, the DFT performs badly, failing utterly to produce reasonable estimates in many cases.

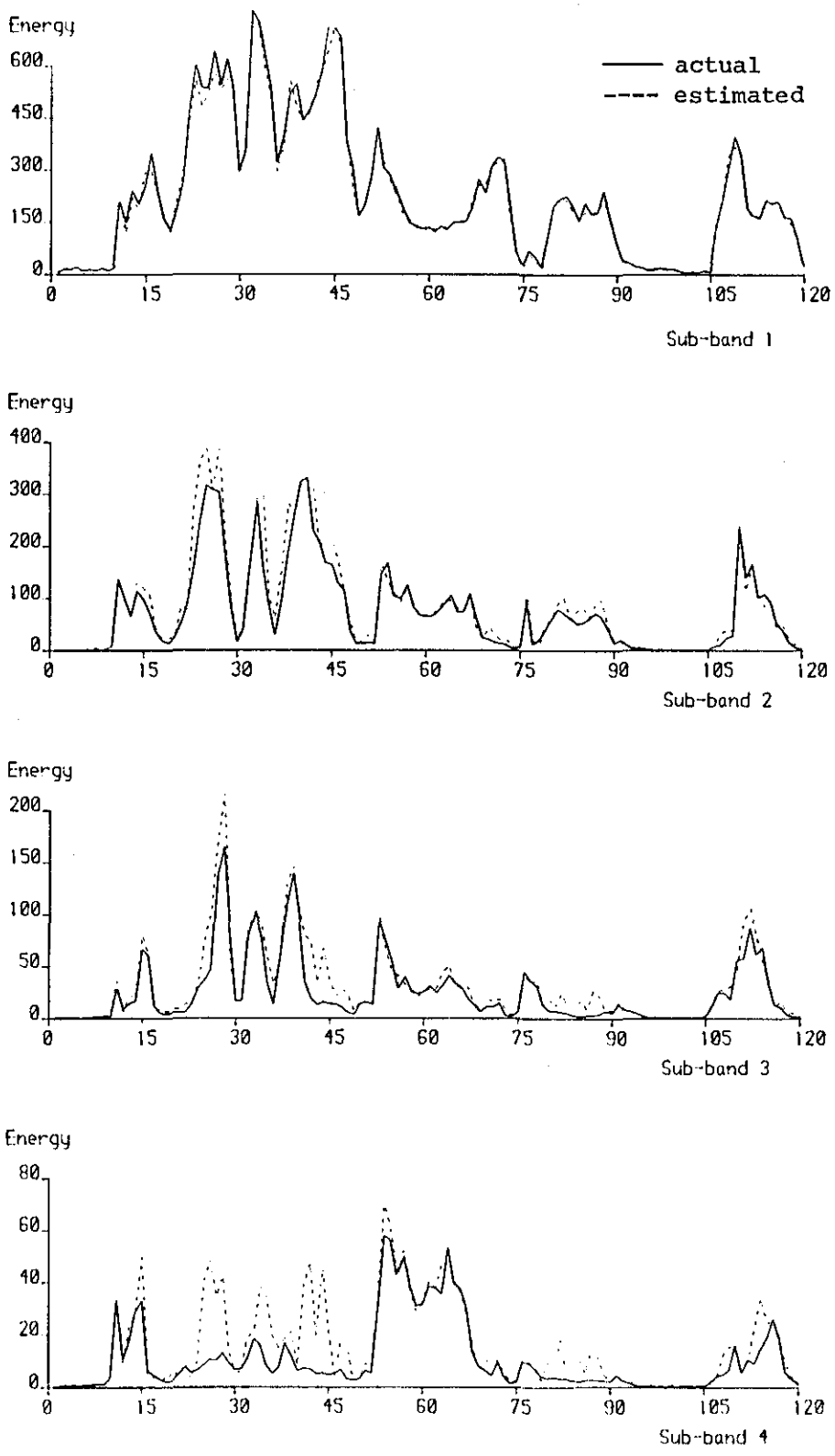
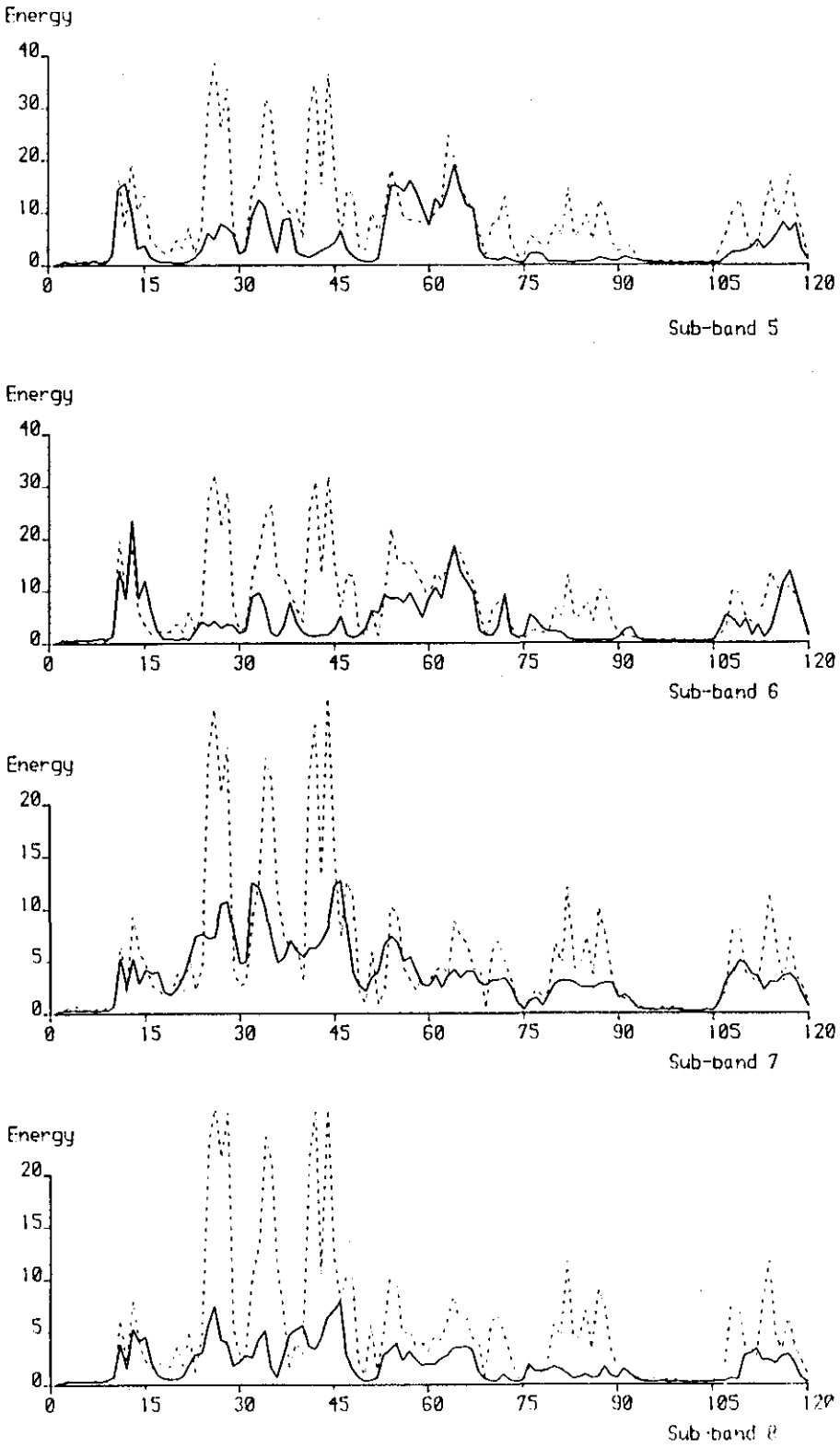


Fig. 6.29 Estimation of Sub-band Signals' Variances using the DFT (8 Band SBC)


Energy
40
30
20
10
0
0
15
30
45
60
75
90
105
120
Sub-band 5
Energy
40
30
20
10
0
0
15
30
45
60
75
90
105
120
Sub-band 6
Energy
20
15
10
5
0
0
15
30
45
60
75
90
105
120
Sub-band 7
Energy
20
15
10
5
0
0
15
30
45
60
75
90
105
120
Sub-band 8

This poor estimate for the higher bands is due largely to the imperfections of the DFT model as a spectral analyser. The errors in the model are particularly emphasised at the high frequency region where signal energy is very low. Nevertheless, it is reasonable to assume that the relative energy distribution among the sub-bands would not be too drastically affected by these inaccuracies. Consequently, the bit allocation map obtained by this means would be fairly similar to the 'serial' method. Observations of the bit allocation maps for the two methods do reveal some minor differences, mostly occurring with bands which are assigned a small number of bits. These deviations from the optimal cases however, are sufficiently frequent to result in a drop in SNR performance. Figure 6.30 shows the SNR performance of the SBC using this method of bit allocation, compared to the two cases considered before. The same trend is shown by the output noise spectral plots (figure 6.31). Parallel bit allocation generally produces a higher level of output noise compared to the other two methods.

The higher noise level is also audible subjectively. The synthesised speech produced by parallel bit allocation methods contains even more of the 'burbly' distortion noted for schemes employing AQJ, and the quality is quite significantly worse than the conventional SBC.

6.7 SUMMARY AND CONCLUSION

The area of frequency domain speech coding has been examined in some detail in this chapter, with particular emphasis on sub-band and adaptive transform coding. While these techniques are generally able to

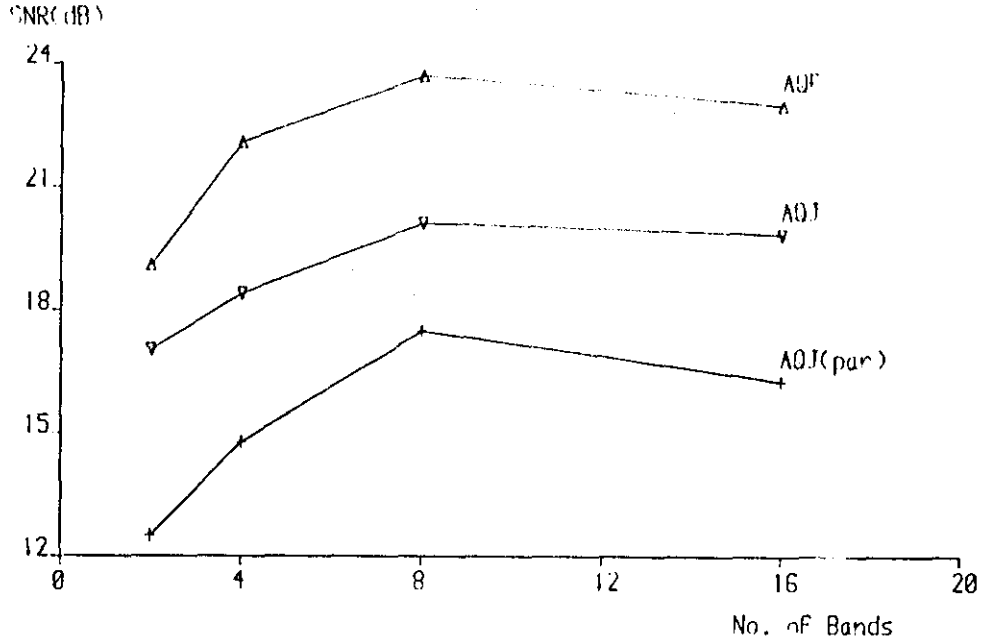
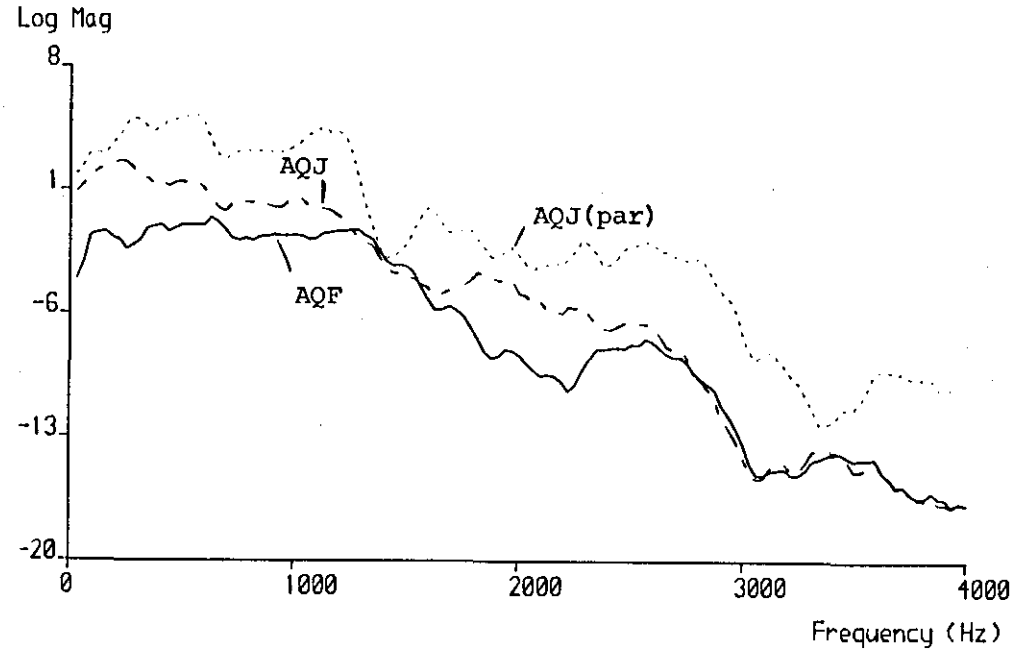


Fig. 6.30 Segmental SNR Performance of SBCs Employing Adaptive Bit Allocation with (i) AQF (ii) AQJ and (iii) AQJ using Parallel Bit Allocation

offer improvements in performance over simple time domain coders at the same bit rate, they are usually also comparatively much more complex and normally incur fairly long coding delays. These drawbacks may well render them unsuitable for many applications.

The sub-band coder, in particular has received an enormous amount of interest in recent years as a viable means of achieving good quality speech at low to medium bit rates with a complexity that is acceptable. [147-160]. Much of this interest has been due to the development of quadrature mirror filters which are able to achieve frequency band splitting without the aliasing problems that have dogged earlier designs using band-pass filters. Realisation of the SBC in hardware has also been eased considerably with the introduction of polyphase filter designs. However, for SBCs with a large number of bands, the accumulated delay due to the QMF tree, plus the delay incurred by



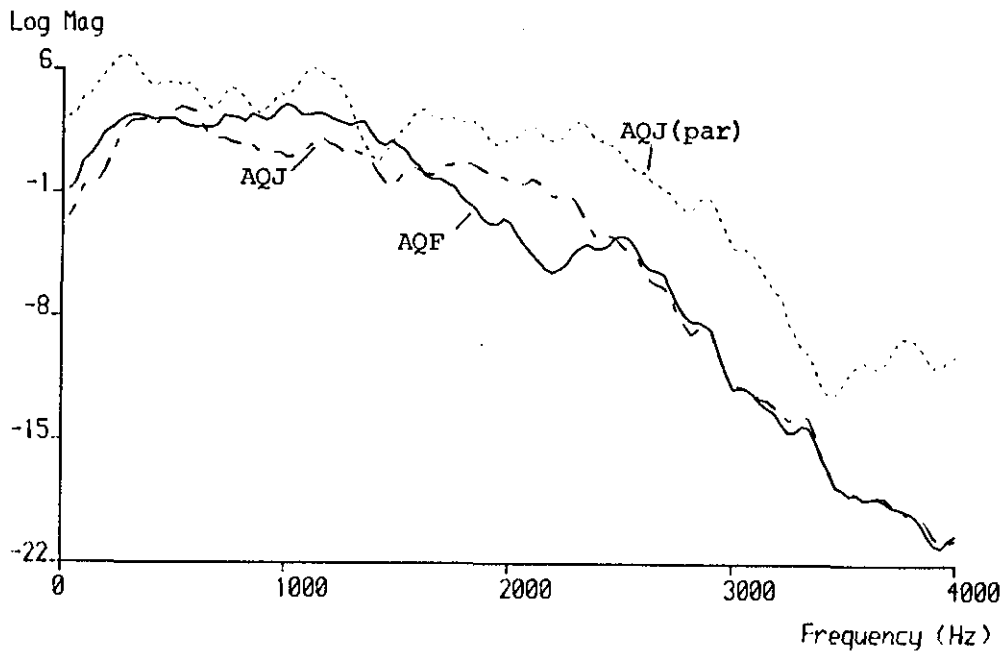


Fig. 6.31 Output Noise Spectra for SBCs employing Adaptive Bit Allocation with
(i) AQF
(ii) AQJ
(iii) AQJ using Parallel Bit Allocation

forward adaptive bit allocation and quantization may still prove to be unacceptable for some applications.

In an effort to minimise this undesirable delay and the complexity involved in the filtering process, a novel approach to split-band coding is proposed and described[201]. This was found to provide comparable performance to the sub-band coder, but with much reduced complexity and delay. Moreover, the approach promises greater flexibility and control of the various parameters involved so that a whole range of different coder designs are available to meet the requirements of a variety of applications.

A number of techniques for further reducing the delay and complexity associated with split-band coding schemes have also been presented and examined. The use of backward quantization together with forward transmission of bit allocation patterns using a simple form of vector quantization, has resulted in a significant reduction of side information bit rate for coders employing a large number of bands. The problem of the one-bit backward quantizer adaptation for the high frequency bands was overcome by transmitting the sign information of the signal and obtaining the magnitude adaptation from a scaled version of a high energy reference band. A further parallel method of adaptive bit allocation was proposed for use with the QMF tree-structured sub-band coder. This is able to halve the delay associated with the conventional method and would be attractive for applications where coding delay although permitted, must be controlled to a suitably low value (to avoid the use of echo cancellers in transmission networks, for instance[10]). Experimental results unfortunately indicate a drop in performance when backward adaptive quantization is employed.

One interesting consequence of the proposed transform-based approach to split-band speech coding is the ability to achieve 32 band frequency resolution (fine frequency resolution becomes increasingly essential for obtaining good quality speech at low bit rates), without the unacceptable delay and complexity of the conventional sub-band coder, thus bringing wide-band analysis split-band schemes a step closer to the ultimate narrow-band coding analysis provided by the adaptive transform coder.

CHAPTER SEVEN RECAPITULATION AND CONCLUSION

This final chapter provides a brief recapitulation of the work described in the last four chapters of the thesis, together with the main results obtained. Suggestions for possible areas and directions of future research are also given.

7.1 RECAPITULATION

A fairly wide cross-section of current digital speech coding techniques has been investigated in detail in the course of this research. The underlying aim of the exercise is, of course an attempt to search for new and more efficient methods of speech coding which are able to provide improvements over existing methods, in terms of straight forward bit rate reduction, enhancement of decoded speech quality, increased robustness to transmission errors or a decrease in coder complexity. This frequently involves slight modifications to existing algorithms, although on occassions, an entirely new approach may be undertaken. Much of the work done is geared for applications at a transmission bit rate of 16 Kbps, but the performance of coders at higher or lower rates is also considered where it is relevant to the context.

Chapter three begins with a survey of current prediction techniques used in ADPCM speech coding. Both fixed and adaptive prediction were considered, and for the latter, both forward and backward adaptive algorithms were examined. Generally, forward prediction is superior to

backward adaptation in terms of error minimisation and signal processing requirements. However, the dual penalty of coding delay and additional side information transmission associated with forward methods can be a serious disadvantage in certain applications. Consequently, much of the research effort is concerned with backward adaptation techniques, which do not have these problems and are therefore more attractive in many cases. Backward prediction algorithms normally involve some form of steepest descent or gradient technique of adaptation, where the predictor coefficients are sequentially modified based on past information, in an attempt to minimise the prediction error. Such methods are traditionally based on a transversal filter structure and are characterised by the general predictor update equation,

$$a_k(n+1) = a_k(n) + g(n)\hat{e}(n)\hat{x}(n-k) \qquad (7.1)$$

where $\{a_k, k=1,2,...p\}$ are the p predictor coefficients at the nth instant, g(n) is a gain constant and $\hat{e}(n)$ and $\hat{x}(n)$ are the quantized prediction residual and decoded signal sample, respectively. An example of this technique is the conventional stochastic approximation predictor (SAP) which uses a fixed gain, g(n), optimised from long-term characteristics for the adaptation. It is known however, that the gain constant should ideally be able to adapt to the short-term signal statistics, taking on a large value for quicker adaptation during signal transitions and a smaller value during steady-state periods of short-term stationarity. At the same time, the use of the most recent decoded samples in the adaptation of the higher predictor coefficients (not permitted by (7.1)) could possibly lead to better prediction accuracy, since it ensures that the latest available information is always utilised in the update process. Two modifications to the basic SAP algorithm were proposed in an attempt to produce more efficient

adaptation. The first introduces a simple switched gain term, which can take on a number of different values depending on the directions of predictor adaptation at previous instants. Successive adaptations in one direction indicate a possible signal transition - the predictor coefficients need to change quickly, and g(n) is set to a high value. On the other hand, adaptations of opposite polarity imply signal stationarity, and therefore, a smaller value of g is desirable. The second modification seeks to provide some inter-relation between the predictor coefficients and to allow the adaptation of higher coefficients to be affected by the most recent decoded signal samples. Although there was evidence of improved performance during signal transitions as a consequence of these modifications, this was not sufficiently significant to produce conclusive results in terms of SNR gains. Further experimentation suggests that the scope for improving prediction efficiency based on modifying the SAP algorithm is very limited, due to the relative insensitivity of predictor performance to a wide range of changes in the amount of adaptation used.

This led to a move away from sequential techniques to the development of the backward block adaptive (BBA) predictor, in which predictor adaptation is based on an optimally determined block of previous decoded samples. This was found to perform better than the gradient methods generally, and to compare well with the efficient sequential lattice algorithm, in terms of both SNR and subjective speech quality. The better performance is particularly significant for signals with a high unvoiced content and frequent magnitude transitions, where an improvement over the gradient methods of as much as 6 dB was recorded. At the same time, the BBA predictor promises greater robustness to

transmission errors (since a block, as opposed to a sequential adaptation is involved) and lower complexity in terms of signal processing requirements, compared to the sequential methods. The use of the BBA predictor is therefore recommended for ADPCM applications.

The second part of chapter three considers some pitch adaptive coding schemes, with the aim of simplifying the powerful adaptive predictive coder (APC) to an easily implementable level of complexity without sacrificing too much of its signal compression ability. A relatively simple one-tap pitch predictor based on the average magnitude difference function (AMDF) algorithm was used together with a fixed vocal tract predictor for this simplified APC scheme. Although SNR was high for periodic segments of the speech signal when the pitch is correctly detected, the coder was too dependant on pitch extraction accuracy (not provided by the simple one-tap predictor) to give consistently reliable performance. The use of adaptive vocal tract prediction does not help either, as it appears to affect the periodic structure of the residual signal in a way which interferes with the signal compression process provided by the pitch loop. Further tests showed that the general performance of this simplified APC system is no better than the much simpler ADPCM with adaptive prediction.

Chapter four examines the effectiveness of incorporating noise shaping features into differential coders. The principle of noise shaping is the manipulation of the relative output noise distribution in the frequency domain, to produce a reduced perception of noise in the decoded speech. Two methods of achieving noise shaping in ADPCM systems were introduced and investigated - one employs quantization noise feedback via a noise shaping filter on the ADPCM structure, while the

other performs the shaping of the noise spectrum externally, i.e. using pre- and post-filtering on the differential system. Both forward and backward methods of adaptation were examined. In the forward case, the coder adaptation parameters were all optimised from the input speech samples and transmitted as side information to the receiver. Noise shaping is thus also considered as forward adaptive, since the coefficients of the noise feedback filter are derived from those of the predictor. For this case, the pre-/post-filtering approach was found to yield better SNR and subjective speech quality, due largely to the favourable interaction of the predictor and quantizer under the coarse (2-bit) quantization condition. The improvement however, was obtained at a slight expense of an increase in coding delay and transmission bit rate, since an additional set of adaptive pre-filter coefficients has to be computed and transmitted. This penalty can be avoided if the pre-filter is fixed. In this case, although a drop in speech quality was noticeable, the general performance is still comparable to, or better than the more complicated adaptive noise-feedback scheme. We conclude that for applications in which coarse quantization is employed, the need for a relatively complex noise-feedback filter is unwarranted, since a fixed pre-/post-filtering arrangement (or pre-emphasis) is adequate to provide the available improvement in subjective performance.

As noted previously, the necessity of delay and side information may render forward adaptive methods unsuitable for certain applications. Consequently, our investigation into noise shaping also involved fully backward adaptive techniques which do not suffer from these drawbacks. The same two methods of noise shaping as in the forward case were examined, with the important difference that all adaptation -

prediction, pre-filtering and quantization are performed in a backward mode. The BBA predictor developed in chapter three was used for this purpose. Experiments indicate that significant improvements in decoded speech quality are obtainable from both methods over conventional ADPCM. In particular, the backward adaptive pre-/post-filtering configuration proposed was able to exploit advantageously the quantizer-predictor interaction in the system to yield extremely good quality speech, which at 16 Kbps, is comparable to that obtained from 7-bit log PCM.

Adaptive quantization techniques are considered in chapter five, where emphasis is placed on the backward adaptation algorithms suitable for use in ADPCM systems. Undoubtedly the best known adaptive quantization technique is the one-word memory algorithm (AQJ) developed by Jayant. However, although its efficiency has been widely recognised, it is nonetheless limited in its ability to respond quickly to rapid signal transitions, such as that encountered in the ADPCM prediction residual. This residual signal consists typically of a randomly varying waveform punctuated by large magnitude spikes at the positions of the excitation (or pitch) pulses. One proposal for improving the AQJ is the pitch compensating quantizer (PCQ) requiring variable rate coding methods, which are quite unacceptable for many applications. A different approach to this problem, which does not attempt to alter the basic ADPCM configuration in any way, was proposed and developed in this chapter. This method consists of applying correction to the ADPCM decoded speech samples at the receiver based on observations of the received quantizer output sequences. The appropriate correction factors used are obtained from long-term statistics, derived from relating the quantizer output sequences to the corresponding input samples at the

transmitter. Experiments on ADPCM systems employing different methods of prediction indicate a general reduction in the coder output noise level across the frequency spectrum, due to the correction, with substantial suppression of high frequency noise. This noise reduction is reflected in higher SNR values and reduced background hiss in the decoded speech.

Chapter six is concerned with frequency domain coding. The adaptive transform coder (ATC) and the sub-band coder (SBC) were both simulated and examined. Generally, the decoded speech at 16 Kbps produced by these powerful frequency domain coders is of a high quality. However, the associated complexity is often also much greater than time domain coders such as ADPCM. Moreover, some coding delay is invariably required in these systems and this can be quite substantial in many cases, such as in the tree-structured filter-bank implementation of the SBC. In order to control the amount of delay and the level of complexity associated with the SBC and ATC, a new approach to frequency domain coding was developed and evaluated. This is essentially a split-band coding scheme similar to the SBC, except that instead of a filter-bank analysis, a discrete transformation approach is used to perform the partitioning of the input signal into frequency sub-bands. These sub-band signals are then coded in the usual way - using dynamic bit allocation and forward adaptive quantization (AQF). This transform-based split-band coder (TSBC) was found to provide comparable performance to the SBC, but with much reduced complexity and coding delay. In addition, the approach employed allows greater flexibility in the design of a coding system, as the various parameters involved are easily modified to yield the optimal trade-off between performance,

delay and complexity for a given application and environment.

Much of the superior performance of frequency domain coders (the split-band techniques in particular) lies in the use of preferential encoding i.e. the adaptive assignment of bits for coding each frequency component or band in accordance with some minimum distortion criterion. The adaptation parameters therefore needs to be communicated periodically to the receiver as side information. The amount of this side information is a function of the number of frequency bands employed and can be quite considerable when the number of bands is large. One method of reducing the side information for SBC and TSBC schemes proposed in the chapter utilises a simple form of vector quantization to transmit the bit allocation information to the receiver, while the sub-band signals are quantized using AQJ. By this means, the side information can be kept to a suitably small proportion of total available bit rate so that coding efficiency is not impaired. However, the use of the AQJ instead of the more efficient AQF leads to a perceptible degradation in the subjective decoded speech quality.

A further effort to reduce the total delay in the SBC makes use of parallel bit allocation. Unlike the conventional method where the bit assignment process is performed on blocks of the sub-band signals emerging from the analysis filter bank, this proposal determines the bit allocation pattern corresponding to a given block of the input signal during the time delay incurred by the propagation of the signal through the filter bank. Together with the use of backward quantization for the sub-band signals, this method avoids the delay due to the adaptive bit allocation procedure, without forgoing the advantages and flexibility of preferential encoding. The actual bit assignment pattern is computed

from the discrete Fourier transform (DFT) of the appropriate block of input signal, in parallel with the split-band analysis. However, although the bit allocation patterns produced by this means appear to reflect the frequency composition of the input signal rather well, preliminary observations indicate that the slight deviation from the optimum (serial) allocation is sufficient to result in a drop in coder performance.

It has not been possible, during the course of this research into various speech coding techniques, to cover each area investigated with a completeness or thoroughness that would be desirable. Nevertheless, it is believed, the main lines of investigation in each area have been pursued with sufficient depth, although a not insignificant amount of follow-up research remains to be done. Some suggestions for further investigation continuing from the present work are given in the following.

7.2 SUGGESTIONS FOR FURTHER RESEARCH

While the ADPCM configuration continues to attract interest in speech coding applications, recent trends have indicated that much of this is concerned with practical aspects of the coder, especially with regard to its performance in a less than ideal transmission environment. Telecommunication organisations such as CCITT, which has favoured the ADPCM configuration during its recent standard-setting exercise for 32 Kbps coding, are particularly interested in the capability of the coder to withstand errors, up to rates of 10^{-3}. Consequently, the BBA predictor developed in chapter three for ADPCM applications must also be tested in a noisy transmission environment to assess its robustness to

transmission errors. While the predictor can be expected to be more robust than the conventional gradient algorithms by intuitive reasoning (since the block method of adaptation employed provides some 'smoothing' effects), proper evaluative tests must be conducted nonetheless, before conclusions can be drawn.

The interest in the exploitation of pitch periodicity in speech signals to effect signal compression has not waned over the years, although a simple yet effective solution to accurate pitch detection remains as elusive as ever. Due to the wide variations in pitch frequency encountered in speech signals, the use of pitch adaptive methods of redundancy removal in differential schemes has been fraught with difficulties. In fact, the pitch predictor has been dispensed with recently by one researcher working on differential coding systems, on the ground that its contribution to efficiency is far outweighed by the many problems connected with its use. Nevertheless, considering that a great proportion of normal speech is quasi-periodic voiced sounds, the exploitation of pitch information will continue to have an appeal. There is much scope for further research in this direction - not merely in attempting to produce novel methods of pitch prediction, but more importantly, to arrive at an algorithm which can be applied to differential schemes without excessive complexity and which is able to maintain an acceptable level of performance during the occurrence of pitch errors without leading to instability in the system.

The work on noise shaping covered in chapter four is perhaps more complete than the other chapters. Once again however, it is necessary to consider the performance of the various systems proposed, in the presence of transmission errors. While the backward pre-/post-filtering

method of noise shaping provides the best performance in terms of speech quality, it is possibly also the least robust system of all, due to the use of backward adaptation for the predictor, pre-filter and quantizer. Experiments must be conducted to determine the error performance of this system, and remedial measures applied where necessary. One problem associated with all backward adaptation algorithms is the danger of divergence. For adaptive prediction, this occurs when the predictor at the receiver fails to track the predictor at the transmitter, due to the accumulated effects of transmission errors in the latter. A possible method of checking this divergence is to re-synchronise the predictors at both ends periodically - setting the coefficients to certain fixed pre-determined values and then allowing adaptation to proceed. Also, all the systems employing the AQJ will need to have it replaced by the robust version which incorporates a leakage factor to dissipate the effects of errors. In the same way, some form of subdued prediction might also be helpful in improving predictor error performance.

The quantizer correction procedure described in chapter five provides a new approach to noise reduction in ADPCM-AQJ systems which leaves the basic differential coder structure undisturbed. However, because the set of correction factors used were obtained from long-term statistics, it would be a sub-optimum compromise for the short-term, being too large for some cases and too small for others. What is required is obviously a set of variable correction factors which is able to adapt according to the short-term requirements of the signal. To avoid the need for side information, this adaptation must preferably evolve in a backward mode, based on previously decoded samples. A useful first step might be to link the magnitudes of the correction factors to the local signal power.

The energy in the vicinity of a pitch pulse is always higher than average and this could be used to control the variation of the gain term.

In the realm of frequency domain coding, modifications to existing techniques for the purpose of obtaining improved performance are usually rather limited. While the sub-band coder has received much attention as a viable means of speech coding in recent years, a great proportion of the interest it generates has been focussed on secondary issues, such as the use of more efficient methods of performing the bit allocation, and coding the side information. The same is true of the adaptive transform coder. Apart from the highly complicated vocoder-driven strategy suggested for low bit rate applications, the ATC system has been virtually unchanged since its first appearance in the literature. It appears that for frequency domain coders such as these, where quality is already extremely good, more attention should perhaps be shown on the problems of complexity and delay. This has been done to some extent by the proposed transform-based split-band coding (TSBC) approach to frequency domain coding. More efforts are required in this direction however, to understand more fully the implications of this approach. The problems of delay and side information requirements associated with split-band coding schemes is also a useful area for further study. The parallel method of bit assignment suggested in chapter six has not been investigated to sufficient depth owing to limitations of time. This is useful in controlling the delay of the SBC while maintaining the flexibility and advantages of adaptive allocation of bits, and would certainly merit further attention. Limited experiments performed have indicated that the use of AQJ for coding the sub-band signals has led to

a 'burbly' distortion in the decoded speech. More investigations are needed to study the cause of this distortion and to develop remedial measures if possible. Shaping of the output noise spectrum to improve perceptual performance is also worthy of some consideration.

For the ATC, the use of a smaller size transform holds much attraction in terms of coder simplification and practical implementability. There is a lot of scope for research in this area, and recent results have indicated that the degradation introduced by the use of small transforms may be overcome. For the TSBC as well, the use of a smaller initial transform can be useful in reducing both complexity and delay.

7.3 CLOSING REMARKS

The work presented in this thesis is but a tiny corner in the vast and rapidly expanding field of speech coding research. While the underlying goal of any speech coding system is likely to remain unchanged with time, further breakthroughs in digital technology may lead to a re-ordering of the relative importance of different factors pertaining to coder design. Complexity, in particular, will be expected to become an increasingly less important consideration as hardware capabilities continue to surge ahead unabated. This could usher in a new generation of coder algorithms based on exhaustive iterative or search techniques (presently too complex for implementation) which will be able to provide good quality speech at low bit rates. Also, continuing research on the development of a more accurate and comprehensive model of speech production could, in the not too distant future, allow the full potential of source coding to be realised without sacrificing speech quality.

Nevertheless, the time and frequency domain algorithms for speech coding examined in this thesis will continue to be of important relevance in many areas of digital speech communication. It is our hope that the efforts expended in this research work have resulted in a contribution in some small measure to the vast pool of current knowledge in the subject.

APPENDIX A

Durbin's Recursive Solution for the Autocorrelation Equation

The autocorrelation method of solving for the predictor coefficients is given by the set of normal equations[33],

$$\sum_{k=1}^{p} a_k R(|i-k|) = R(i) \qquad ; 1 \leq i \leq p \tag{A.1}$$

where p = order of predictor

$R(i)$ = ith shift autocorrelation

a_k = kth predictor coefficient

Durbin's method involves solving the recursive relations given by the following set of equations[33,221]:

$$E(0) = R(0) \tag{A.2}$$

$$k_i = - \frac{R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)}{E(i-1)} \qquad ; 1 \leq i \leq p \tag{A.3}$$

$$a_i^{(i)} = k_i \tag{A.4}$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i \, a_{i-j}^{(i-1)} \qquad ; 1 \leq j \leq i-1 \tag{A.5}$$

$$E(i) = (1 - k_i^2) E(i-1) \tag{A.6}$$

The optimum predictor coefficients $\{a_k, k=1,2...p\}$ are obtained as,

$$a_i = -a_i^{(p)} \qquad ; 1 \leqslant i \leqslant p \qquad (A.7)$$

and the reflection coefficients are given by the k_i's.

APPENDIX B

Derivation of the Update Equation for the Modified SAP Algorithm (SAPM)

The general SAP update equation is given by,

$$a_k(n+1) = a_k(n) + \gamma_k(n) \qquad (B.1)$$

where,

$$\gamma_k(n) = g e_k(n)\hat{x}(n-k) \qquad (B.2)$$

The residuals $e_k(n)$ are given by,

$$e_1(n) = \hat{e}(n) = \hat{x}(n) - \sum_{k=1}^{p} a_k \hat{x}(n-k)$$

$$e_m(n) = e_{m-1}(n) - \gamma_{m-1}(n)\hat{x}(n-[m-1]) \qquad ; m \geqslant 2 \qquad (B.3)$$

Thus all residuals can be expressed in terms of $e_1(n)$,

$$\begin{aligned}
e_2(n) &= e_1(n) - \gamma_1(n)\hat{x}(n-1) \\
e_3(n) &= e_2(n) - \gamma_2(n)\hat{x}(n-2) \\
&= e_1(n) - \gamma_1(n)\hat{x}(n-1) - \gamma_2(n)\hat{x}(n-2) \qquad (B.4) \\
&\;\;\vdots \qquad\qquad \vdots \\
e_k(n) &= e_{k-1}(n) - \gamma_{k-1}(n)\hat{x}(n-k+1) \\
&= e_1(n) - \gamma_1(n)\hat{x}(n-1) - \gamma_2(n)\hat{x}(n-2) - \ldots e_{k-1}(n)\hat{x}(n-k+1)
\end{aligned}$$

The correction terms, $\gamma_k(n)$, which are functions of $e_k(n)$ can also be expressed in terms of $e_1(n)$:

$$\begin{aligned}
\gamma_1(n) &= ge_1(n)\hat{x}(n-1) \\
\gamma_2(n) &= ge_2(n)\hat{x}(n-2) \\
&= g[e_1(n) - \gamma_1(n)\hat{x}(n-1)]\hat{x}(n-2) \\
&= g[e_1(n) - ge_1(n)\hat{x}^2(n-1)]\hat{x}(n-2) \\
&= ge_1(n)[1 - g\hat{x}^2(n-1)]\hat{x}(n-2) \\
\gamma_3(n) &= ge_3(n)\hat{x}(n-3) \\
&= g[e_1(n) - \gamma_1(n)\hat{x}(n-1) - \gamma_2(n)\hat{x}(n-2)]\hat{x}(n-3) \\
&= ge_1(n)[1 - g\hat{x}^2(n-1)][1 - g\hat{x}^2(n-2)]\hat{x}(n-3) \\
&\;\;\vdots \qquad\qquad \vdots \\
\gamma_k(n) &= ge_1(n)[1 - g\hat{x}^2(n-1)][1 - g\hat{x}^2(n-2)]..... \\
&\quad ...[1 - g\hat{x}^2(n-k+1)]\hat{x}(n-k)
\end{aligned} \tag{B.5}$$

Or more generally, as $e_1(n) = \hat{e}(n)$, the update equation is given by,

$$a_k(n+1) = a_k(n) + g\hat{e}(n)\hat{x}(n-k).F(k) \tag{B.6}$$

where,

$$F(k) = 1 \qquad ; k = 1$$

$$= \prod_{m=1}^{k-1}[1 - g\hat{x}^2(n-m)] \qquad ; 1 < k \leq p$$

APPENDIX C

Computational Requirements of Adaptive Prediction Algorithms

An estimate of the complexity of each adaptive prediction algorithm is presented. This complexity is measured only in terms of the number of multiplications required, with a division considered computationally equivalent to two multiplications. It must be emphasised however, that the accuracy of the following analysis is necessarily limited for the sake of simplicity. In many instances, the amount of computation may be reduced at the expense of increased storage. A complexity measure based solely on multiplications alone is thus incomplete, although it does provide a useful indication of the relative complexity among the various algorithms[225].

(1) Forward Block Adaptive (FBA) Predictor

The computation of the predictor coefficients may be divided into two parts: (i) the calculation of the autocorrelation function over the block of N samples, and (ii) the solution of the normal equations using Durbin's recursion.

(i) Autocorrelation Calculations

Considering a block of N signal samples $\{x_n\}$,the autocorrelation values, R(n) required for a pth order predictor are given as:

$$R(i) = \sum_{n=0}^{N-1-i} x(n)x(n+i) \qquad ; i \geq 0 \qquad (C.1)$$

This requires,

$$N + (N-1) + (N-2) + \ldots\ldots(N-p) \text{ multiplications}$$

i.e.

$$(p+1)N - \sum_{j=1}^{p} j$$

$$= \underline{(p+1)N - p(p+1)/2} \text{ multiplications.}$$

(ii) Durbin's Recursion

Durbin's recursive solution of the autocorrelation equation is given in Appendix A[33,221]. The computational requirement for each step is considered as follows for the first 3 stages,

For i=1, (A.3) to (A.6) are given as:

$k_1 = R(1)/E(0)$; 1 division or 2 multiplications

$a_1^{(1)} = k_1$; 0 multiplications

$E(1) = (1 - k_1^2)E(0)$; 2 multiplications

i=2,

$$k_2 = -\frac{R(2) + a_1^{(1)} R(1)}{E(1)} \qquad ; 3 \text{ multiplications}$$

$a_1^{(2)} = a_1^{(1)}(1 + k_2)$; 1 multiplication

$E(2) = (1 - k_2^2)E(1)$; 2 multiplications

i=3,

$$k_3 = -\frac{R(3) + a_1^{(2)}R(2) + a_2^{(2)}R(1)}{E(2)} \quad ; \text{ 4 multiplications}$$

$$a_1^{(3)} = a_1^{(2)} + k_3 a_1^{(2)} \quad ; \text{ 2 multiplications}$$

$$a_2^{(3)} = a_2^{(2)} + k_3 a_1^{(2)} \quad ; \text{ 2 multiplications}$$

$$E(3) = (1 - k_3^2)E(2) \quad ; \text{ 2 multiplications}$$

Considering a pth order predictor, the number of multiplications for each step is given by,

Step 1 (eqn. (A.3)) :

$$2 + 3 + 4 + \ldots\ldots(p+1)$$

$$= p + \sum_{j=1}^{p} j$$

$$= \underline{p + p(p+1)/2}$$

Step 2 (eqn. (A.5)) :

$$0 + 1 + 2 + \ldots\ldots\ldots(p-1)$$

$$= \underline{p(p+1)/2 - p}$$

Step 3 (eqn. (A.6)) :

$$2 + 2 + 2 + \ldots\ldots\ldots(\text{ p terms})$$

$$= \underline{2p}$$

Thus Durbin's recursion requires :

$$p + p(p+1)/2 + p(p+1)/2 - p + 2p$$

$$= \underline{p(p+3)} \text{ multiplications}$$

Hence, for the pth order FBA predictor, the total amount of computation required using a blocksize of N is given by,

$$(p+1)N - p(p+1)/2 + p(p+3)$$

$$= \underline{(p+1)N + p(p+5)/2} \text{ multiplications}$$

(2) Backward Block Adaptive (BBA) Prediction Algorithm

For the BBA predictor, the autocorrelation values can be updated sequentially as new samples arrive. Considering a block of N signal samples $\{x_n\}$, the autocorrelation values required for a pth order predictor are obtained as,

$$\begin{aligned}
R(0) &= x_1^2 + x_2^2 + x_3^2 + \ldots\ldots\ldots\ldots\ldots x_N^2 \\
R(1) &= x_1x_2 + x_2x_3 + x_3x_4 + \ldots\ldots\ldots\ldots x_{N-1}x_N \\
R(2) &= x_1x_3 + x_2x_4 + x_3x_5 + \ldots\ldots\ldots\ldots x_{N-2}x_N \\
&\vdots \\
R(p) &= x_1x_{p+1} + x_2x_{p+2} + \ldots\ldots\ldots\ldots\ldots x_{N-p}x_N
\end{aligned} \qquad \text{(C.2)}$$

AT the next instant, these are updated as,

$$\begin{aligned}
R'(0) &= x_2^2 + x_3^2 + x_4^2 + \ldots\ldots\ldots\ldots\ldots x_{N+1}^2 \\
R'(1) &= x_2x_3 + x_3x_4 + \ldots\ldots\ldots\ldots\ldots x_Nx_{N+1} \\
&\vdots \\
R'(p) &= x_2x_{p+2} + x_3x_{p+3} + \ldots\ldots\ldots\ldots x_{N+1-p}x_{N+1}
\end{aligned} \qquad \text{(C.3)}$$

$$\text{i.e. } R'(p) = R(p) - x_1x_{p+1} + x_{N+1-p}x_{N+1} \qquad \text{(C.4)}$$

Hence, at each time instant, each of the (p+1) autocorrelation values is updated by discarding the least recent element in the block and adding the contribution from the latest sample. This requires (p+1)N multiplications for a block of N samples.

For the BBA predictor, the autocorrelation is calculated once for a block of N samples, and Durbin's recursion is performed N/M times. This gives the total amount of computation per block of N samples as

$$(p+1)N + p(p+3)N/M$$

$$= \underline{N[\ p+1 + p(p+3)/M\ ]} \text{ multiplications.}$$

(3) Stochastic Approximation Predictor (SAP)

The update equation for the SAP algorithm[75] is given (from (3.30) and (3.31)) by:

$$a_k(n+1) = a_k(n) + \frac{\hat{Ge}(n)\hat{x}(n-k)}{\gamma + 1/p \sum_{j=1}^{p} \hat{x}^2(n-j)} \qquad (C.5)$$

The normalisation term in the denominator is a moving average variance estimator. At the nth instant, it is given by,

$$NORM(n) = 1/p\ [\hat{x}^2(n-1) + \hat{x}^2(n-2) + ...\hat{x}^2(n-p)] \qquad (C.6)$$

and for the (n+1)th instant, it is,

$$NORM(n+1) = 1/p\ [\hat{x}^2(n) + \hat{x}^2(n-1) + ...\hat{x}^2(n-p+1)] \qquad (C.7)$$

From (C.6) and (C.7), it can be seen that, at each sampling instant, the latest $\hat{x}^2$ term is included and the least recent term discarded. Thus, one multiplication and one division (by p) is required. Alternatively, the division by p can be avoided by scaling G and γ appropriately. Hence, the normalisation requires one multiplication. The gain term involves one division by the normalising factor, or equivalently, two multiplications. The $g\hat{e}(n)$ term appears in the update of all the coefficients and needs to be computed once only. So, the computation of all these requires 3 multiplications. Finally, the update procedure of (C.5) needs one more multiplication per coefficient, hence giving for the SAP algorithm, a total requirement of (p+4) multiplications per sample or (p+4)N multiplications per block of N samples.

(4) Modified SAP algorithm (SAPM)

The update equation is given by,

$$\begin{bmatrix} a_1(n+1) \\ a_2(n+1) \\ \vdots \\ a_p(n+1) \end{bmatrix} = \begin{bmatrix} a_1(n) \\ a_2(n) \\ \vdots \\ a_p(n) \end{bmatrix} + g\hat{e}(n) \begin{bmatrix} \hat{x}(n-1) \\ \left[1-g\hat{x}^2(n-1)\right]\hat{x}(n-2) \\ \vdots \\ \left[1-g\hat{x}^2(n-1)\right]\left[1-g\hat{x}^2(n-2)\right]\ldots\hat{x}(n-p) \end{bmatrix} \qquad \text{(C.8)}$$

The term g is computed in the same way as SAP, requiring 3 multiplications. Because of the different gains for the different coefficients, the computational load of SAPM depends on the predictor order p. The requirements are tabulated as follows:

Coefficient	Effective Gain	Additional multiplication
a_1	$\hat{ge}(n)\hat{x}(n-1)$	2
a_2	$\hat{ge}(n)\hat{x}(n-2)[1-g\hat{x}^2(n-1)]$	4
a_3	$\hat{ge}(n)\hat{x}(n-3)[1-g\hat{x}^2(n-1)][1-g\hat{x}^2(n-2)]$	4
.	.	
.	.	
a_p	$\hat{ge}(n)\hat{x}(n-p)[1-g\hat{x}^2(n-1)]...[1-g\hat{x}^2(n-p+1)]$	4

The sucessive coefficients are updated by an accumulated product so the gain term is also successively accumulated, e.g. $\hat{ge}(n)$ for a_1, $\hat{ge}(n)[1-g\hat{x}^2(n-1)]$ for a_2, etc. The total requirements are: $3 + 2 + 4(p-1) = 4p+1$ multiplications per sampling instant or $\underline{(4p+1)N}$ multiplications per block of N samples.

(5) Adaptive Gain SAP (SAPA)

The SAPA variations are similar to SAP in its computational requirements. The different values of g used can be stored in fixed memory and retrieved for use when required.

(6) Fast Converging SAP (FSAP)

The update equation of FSAP[226] is given from (3.64) as:

$$a_k(n+1) = a_k(n) + 1/2\ \beta(1-q)G_f(n) + q(a_k(n) - a_k(n-1)) \qquad (C.9)$$

The term $1/2\ \beta(1-q)G_f(n)$ is similar to the SAP adaptations (β and q are fixed constants) and therefore, requires (p+4) multiplications per sample. An additional multiplication per coefficient is due to the term

$q(a_k(n)-a_k(n-1))$ giving a total of (2p+4) multiplications per sample or (2p+4)N multiplications per block of N samples.

(7) Adaptive Lattice Predictor (LAT)

The so-called direct method of adaptation[200] proceeds as given by (3.50) to (3.52):

$$C_m(n) = (1-\gamma)C_m(n-1) - 2\gamma f_m(n)b_m(n-1) \qquad (C.10)$$

$$D_m(n) = (1-\gamma)D_m(n-1) + \gamma[f_m^2(n) + b_m^2(n)] \qquad (C.11)$$

and

$$k_{m+1}(n) = -C_m(n)/D_m(n) \qquad (C.12)$$

At each instant, (C.10) requires, 3 multiplications,

(C.11) requires 4 multiplications,

and (C.12) requires 1 division or equivalently, 2 multiplications. This gives a total of 9p multiplications per sample. However, if γ is made a power of 2, then some multiplications can be reduced to simple shift operations. In this case, the requirements become:

(C.10) - 1,

(C.11) - 2,

and (C.12) - 2 multiplications,

giving a total of 5pN multiplications per block of N samples.

(8) Adaptive Lattice - Sign-Product Method (LAT-SP)

The sign product method[200] of adapting the lattice predictor is governed by (3.53) - (3.54):

$$k_m(n) = \sin [(\pi/2)k_m'(n)] \qquad (C.13)$$

where,

$$k_{m+1}'(n+1) = (1-\gamma)k_{m+1}'(n) - \gamma \mathrm{sgn}\{f_m(n)\}.\mathrm{sgn}\{b_m(n)\}$$

In this case, no multiplications are involved, if γ is chosen to be a power of 2, and (C.14) is implemented by means of a look-up table.

APPENDIX D

Computation of Autocorrelation Function for Backward Block Adaptive Predictor

---

It is shown in Appendix C(2) that the autocorrelation function for the BBA predictor may be calculated sequentially by adding the most recent contribution and discarding the least recent. Since the computation of the predictor coefficients is performed only once every M samples, the contribution of each newly decoded sample can be accumulated in partial sums (of M samples) to avoid excessive memory demand. For example, the zero-shift autocorrelation for a block of N samples is given by:

$$R(0) = x_1^2 + x_2^2 + \ldots\ldots x_M^2 + \ldots\ldots + x_N^2 \qquad (D.1)$$

After M sampling instants, its updated value is:

$$R'(0) = x_{M+1}^2 + x_{M+2}^2 + \ldots\ldots\ldots\ldots + x_{M+N}^2 \qquad (D.2)$$

Thus,

$$R'(0) = R(0) - \sum_{i=1}^{M} x_i^2 + \sum_{j=N+1}^{N+M} x_j^2 \qquad (D.3)$$

The terms involved in the summation need not be stored individually but can be accumulated as each decoded sample arrives. For instance, if N=256 and M=32, then instead of 256 memory locations, only N/M = 8 are required, each storing the accumulated products of 32 samples as indicated below:

1	2	3	4	5	6	7	8
1-32	33-64	65-96	97-128	129-160	161-192	193-224	225-256

The same method may be used for the other autocorrelation values, so that the total memory requirements is approximately given as (p+1)N/M+p.

APPENDIX E

Proof of Constraint on Quantization Noise Spectrum

Proof of constraint:

$$\frac{1}{f_s} \int_0^{f_s} \log \Gamma(f)\,df = 0 \qquad (E.1)$$

given,

$$\Gamma(f) = \left| \frac{1 - F(e^{2\pi jfT})}{1 - P(e^{2\pi jfT})} \right|^2 \qquad (E.2)$$

where F and P are linear filters given by the general form G(z) in the z domain,

$$G(z) = \sum_{k=1}^{m} g_k z^{-k} \qquad (E.3)$$

f_s is the sampling frequency and T the sampling period. The roots of both (1-F) and (1-P) are assumed to be inside the unit circle[81].

Consider the function (1-F), which is expressed in z transform notation as:

$$1 - F(z) = 1 - \sum_{k=1}^{m} b_k z^{-k}$$

$$= 1 - (b_1 z^{-1} + b_2 z^{-2} + \ldots\ldots b_m z^{-m}) \qquad (E.4)$$

(E.4) is a polynomial in z^{-1} which can be factorised to give,

$$1 - F(z) = (1 - z_1 z^{-1})(1 - z_2 z^{-1}) \ldots . (1 - z_m z^{-1}) \qquad (E.5)$$

i.e.

$$1 - F(z) = \prod_{k=1}^{m}(1 - z_k z^{-1}) \qquad (E.6)$$

where z_k is the kth root of [1-F(z)].

Taking logarithm of (E.6) converts the product term on the r.h.s. to a summation,

$$\log\{1 - F(z)\} = \sum_{k=1}^{m} \log\{1 - z_k z^{-1}\} \qquad (E.7)$$

Since the roots of (1-F) lies inside the unit circle, $|z_k| < 1$ and each term in the r.h.s. of (E.7) can be expanded in a logarithmic series, as a polynomial function of z^{-1}. Hence,

$$\begin{aligned}
\log(1 - z_1 z^{-1}) &= -z_1 z^{-1} + 1/2\, z_1^2 z^{-2} - 1/3\, z_1^3 z^{-3} + \ldots \\
\log(1 - z_2 z^{-1}) &= -z_2 z^{-1} + 1/2\, z_2^2 z^{-2} - 1/3\, z_2^3 z^{-3} + \ldots \\
&\vdots \qquad\qquad \vdots \\
\log(1 - z_m z^{-1}) &= -z_m z^{-1} + 1/2\, z_m^2 z^{-2} - 1/3\, z_m^3 z^{-3} + \ldots
\end{aligned} \qquad (E.8)$$

Summing up similar terms in the expansion gives,

$$\begin{aligned}
\sum_{k=1}^{m} \log(1 - z_k z^{-1}) = &- (z_1 + z_2 + z_3 + \ldots z_m) z^{-1} \\
&+ 1/2(z_1^2 + z_2^2 + z_3^2 + \ldots z_m^2) z^{-2} \\
&\vdots \\
&(-1)^m 1/m(z_1^m + z_2^m + z_3^m \ldots z_m^m) z^{-m}
\end{aligned}$$

$$= \sum_{n=1}^{\infty} c_n z^{-n} \qquad \text{(E.9)}$$

$$\text{where } c_n = z_1^n + z_2^n + \ldots . z_m^n = \sum_{k=1}^{m} z_k^n \qquad \text{(E.10)}$$

Therefore,

$$\log \{1 - F(z)\} = \sum_{n=1}^{\infty} c_n z^{-n} = \sum_{n=1}^{\infty} c_n e^{-2\pi jfTn} \qquad \text{(E.11)}$$

The integral of $\log[1-F(z)]$ over the frequency range from 0 to f_s is then given by,

$$\int_0^{f_s} \{\log \{1 - F(e^{2\pi jfT})\}df = \sum_{n=1}^{\infty} c_n \int_0^{f_s} e^{-2\pi jfTn} df = 0 \qquad \text{(E.12)}$$

Since P is of the same form as F, the same result holds for (1-P), thus,

$$\frac{1}{f_s} \int_0^{f_s} \log\Gamma(f)df = 0$$

APPENDIX F

Aliasing Cancellation Property of Quandrature Mirror Filter Bank[145,147]

Let $X1(e^{j\omega})$ and $X2(e^{j\omega})$ be the Fourier transforms of the lower and upper sub-band signals, respectively before decimation and $X(e^{j\omega})$ be the

transform of x(n). Then,

$$X1(e^{j\omega}) = X(e^{j\omega})\ H1(e^{j\omega}) \tag{F.1}$$

and

$$X2(e^{j\omega}) = X(e^{j\omega})\ H2(e^{j\omega}) \tag{F.2}$$

where $H1(e^{j\omega})$ and $H2(e^{j\omega})$ are the Fourier transforms of h1(n) and h2(n), respectively. After decimation, the lower and upper sub-band signals may be defined as $Y1(e^{j\omega})$ and $Y2(e^{j\omega})$, respectively and can be expressed as,

$$Y1(e^{j\omega}) = 1/2\ [X1(e^{j\omega/2}) + X1(e^{j(\omega+2\pi)/2})] \tag{F.3}$$

and

$$Y2(e^{j\omega}) = 1/2\ [X2(e^{j\omega/2}) + X2(e^{j(\omega+2\pi)/2})] \tag{F.4}$$

Letting $U1(e^{j\omega})$ and $U2(e^{j\omega})$ be the interpolated lower and upper sub-band signals in the receiver, and ignoring effects of quantization, we get,

$$U1(e^{j\omega}) = 2\ Y1(e^{j2\omega})\ H1(e^{j\omega}) \tag{F.5}$$

and

$$U2(e^{j\omega}) = -2\ Y2(e^{j2\omega})\ H2(e^{j\omega}) \tag{F.6}$$

Finally, the output signal $\hat{X}(e^{j\omega})$, the transform of $\hat{x}(n)$ in figure 6.4(a), can be expressed as,

$$\hat{X}(e^{j\omega}) = U1(e^{j\omega}) + U2(e^{j\omega}) \tag{F.7}$$

Combining equations (F.1) to (F.7) gives the input to output relation

$$\hat{X}(e^{j\omega}) = X(e^{j\omega})[H1^2(e^{j\omega}) - H2^2(e^{j\omega})]$$

$$+ X(e^{j(\omega+\pi)})[H1(e^{j\omega})H1(e^{j(\omega+\pi)}) - H2(e^{j\omega})H2(e^{j(\omega+\pi)})] \quad (F.8)$$

The first term of the r.h.s. of (F.8) expresses the desired component of $\hat{X}(e^{j\omega})$ and the second term expresses the undesired aliasing component. The cancellation of this aliasing component can be observed by transforming equantion (6.3) to get,

$$H1(e^{j\omega}) = H2(e^{j(\omega+\pi)}) \quad (F.9)$$

and applying this condition to (F.8). It can be easily verified that the second term cancels, leaving

$$\hat{X}(e^{j\omega}) = X(e^{j\omega})[H1^2(e^{j\omega}) - H1^2(e^{j(\omega+\pi)})] \quad (F.10)$$

From the symmetry property in equation (6.2), it can be shown that the frequency response of $H1(e^{j\omega})$ can be expressed in the form,

$$H1(e^{j\omega}) = |H1(e^{j\omega})| \, e^{j\omega(T-1)/2} \quad (F.11)$$

Recalling that N is even, and applying this condition to (F.10), leads to the expression,

$$\hat{X}(e^{j\omega}) = X(e^{j\omega})[|H1(e^{j\omega})|^2 + |H1(e^{j(\omega+\pi)})|^2] \, e^{j\omega(T-1)} \quad (F.12)$$

In the above expression, the term $e^{j\omega(T-1)}$ implies that there is a T-1 sample delay between $\hat{x}(n)$ and $x(n)$. Furthermore, it can be seen from (F.12) that if $\hat{x}(n)$ is to be a (delayed) replica of $x(n)$, then $H1(e^{j\omega})$ must satisfy the requirement that,

$$|H1(e^{j\omega})|^2 + |H2(e^{j(\omega+\pi)})|^2 = 1 \qquad \text{(F.13)}$$

or equivalently,

$$|H1(e^{j\omega})|^2 + |H2(e^{j\omega})|^2 = 1 \qquad \text{(F.14)}$$

APPENDIX G

Computational Requirements of the Tree-structured Quadrature Mirror Filter Bank Sub-band Coder

---

Consider the filtering of N samples through the QMF filter bank[145], employing T tap filters. Using polyphase implementations[147], the tree structure of figure G.1 results. Considering the first stage, the N input samples x(n) are divided into 2 signals of N/2 samples each, containing the odd and even values of x(n) respectively. The odd samples are filtered by a T/2 tap filter containing the odd filter coefficients of the original QMF and the even samples by a T/2 tap filter containing the even coefficients. The sums and differences of the two filter outputs are taken to produce the (decimated) upper and lower band signals, respectively.

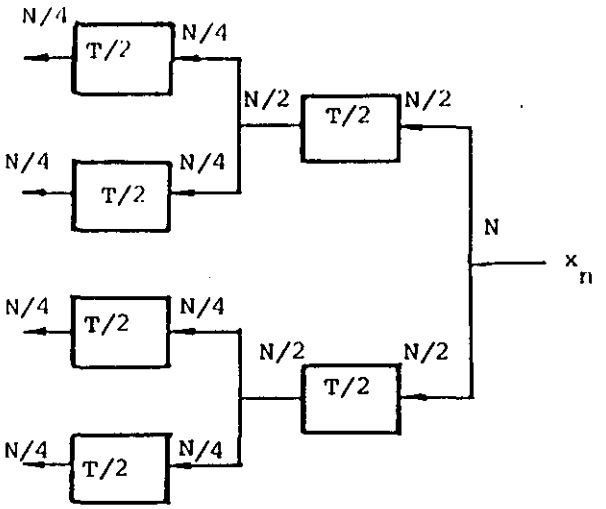


Fig. G.1

Polyphase Implementation of Sub-band Coder

For the first stage, the filtering of N samples of x(n) involves twice N/2 x T/2 = NT/2 multiplications and additions. Also, the N/2 outputs from the two filters must be added and subtracted, giving 2 x N/2 = N further additions. So, for the first stage, the computational requirements are: NT/2 multiplications and NT/2 + N additions.

For subsequent stages, the amount of computation remains the same as is clear from figure G.1, so that a b band sub-band coder requires: NT/2 $\log_2 b$ multiplications and N(T/2 + 1) $\log_2 b$ additions. Each sample therfore requires, T/2 $\log_2 b$ multiplications and (T/2 + 1) $\log_2 b$ additions.

APPENDIX H

Computational Requirements of the Transform-based Split Band Coder

---

For an N point DCT, the amount of computation is given by, $3N/2(\log_2 N-1)$

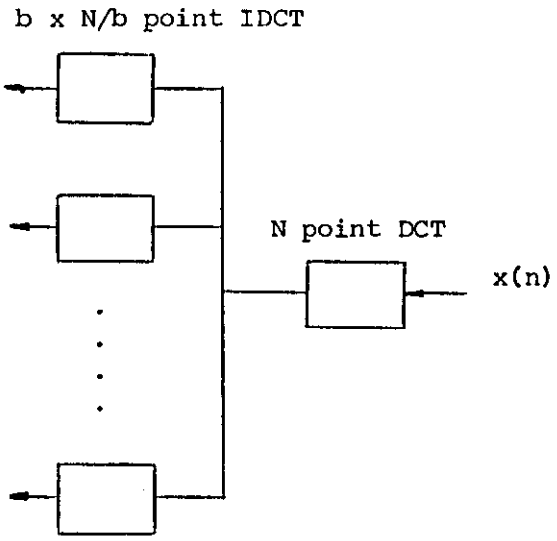


Fig. H.1 Transform-based Split-band Coder

+2 additions and $N\log_2 N-3N/2+4$ multiplications[236]. The b band split-band coder of figure H.1 requires 1 N point and b N/b point transforms, thus requiring,

$$3N/2\ (\log_2 N - 1) + 2 + b[3N/2b\ (\log_2 (N/b) - 1) + 2]$$

$$= 3N\log_2 N - 3N - 3N/2\ \log_2 b + 2(b + 1) \text{ additions.}$$

and

$$N\log_2 N - 3N/2 + 4 + b[N/b\ \log_2 (N/b) - 3N/2b + 4]$$

$$= 2N\log_2 N - 3N - N\log_2 b + 4(b + 1) \text{ multiplications.}$$

Therefore, for each sample, the requirements are,

$$3\log_2 N - 3 - 3/2\ \log_2 b + 2(b + 1)/N \text{ additions}$$

and

$$2\log_2 N - 3 - \log_2 b + 4(b + 1)/N \text{ multiplications.}$$

REFERENCES

1 C. Cherry, On Human Communication - MIT Press (3rd Edition), Cambridge, Massachusetts 1978.

2 J.L. Flanagan, Speech Analysis, Synthesis and Perception - Second Edition, Springer-Verlag, New York, 1972.

3 J.L. Flanagan, "Voices of Men and Machines" - J. Acoust. Soc. America, vol.51, pp.1375-1387, Mar. 1972.

4 J.F. Young, Information Theory - Butterworth, London 1971.

5 R. Paget, Human Speech: Some Observations, Experiments, and Conclusions as to the Nature, Origin, Purpose and Possible Improvement of Human Speech - Harcourt, New York 1930.

6 J. Reusch and W. Kees, Non-verbal Communication - University of California Press, Berkeley and Los Angeles, 1964.

7 L. Hogben, "The Wonderful World of Communication" - Macdonald, London 1969.

8 B.P. Lathi, Communication Systems - John Wiley & Sons, Inc., 1968.

9 K.W. Cattermole, Principles of Pulse Code Modulation - Illiffe, London 1973.

10 X. Maitre & T. Aoyama, "Speech Coding Activities within CCITT: Status and Trends" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.954-959, May 1982.

11 C.S. Xydeas, "Differential Encoding Techniques Applied to Speech Signals" - Ph.D. Thesis, Loughborough University of Technology, Dec. 1978.

12 J.L. Flanagan, M.R. Shroeder, B.S. Atal, R.E. Crochiere, N.S. Jayant & J.M. Tribolet, "Speech Coding" - IEEE Trans. on Commun., vol. COM-27, pp.710-737, April 1979.

13 R.E. Crochiere & J.L. Flanagan, "Current Perspectives in Digital Speech" - IEEE Communications Magazine, pp.32-40, Jan. 1983.

14 PRIME Computer, PRIMOS Commands Reference Guide, Prime Computer Inc., Mass. USA 1979.

15 PRIME, The Fortran Programmer's Guide, Prime Computer Inc., Mass. USA 1979.

16 PRIME, Primos Subroutines Reference Guide PDR3621, Prime Computer Inc., Mass. USA 1979.

17 GINO-F User Manual - Computer Aided Design Centre.

18 GINOGRAF User Manual - Computer Aided Design Centre.

19 J.D. Gibson, "Adaptive Prediction in Speech Differential Encoding Systems" - Proc. of the IEEE vol.68, no.4, pp.488-525, April 1980.

20 P. Noll, "Adaptive Quantizing in Speech Coding Systems" - Proc. Int. Zurich Seminar on Digital Comm., pp.B3.1-b3.6, 1974.

21 NAG Fortran Library Manual (Mark 8) - Numerical Algorithms Group, 1981.

22 Hewlett Packard Operating Manual, Fourier Analyzer System 5451A.

23 B. Gold, "Digital Speech Networks" - Proc. of the IEEE, vol.65, no. 12, pp.1636-1658, Dec. 1977.

24 R.W. Schafer & L.R. Rabiner, "Digital Representations of Speech Signals" - Proc. IEEE, vol.63, pp.662-677, April 1975.

25 H. Dudley, "The Vocoder" - Bell Labs Rec. vol.18, pp.122-126, Dec. 1939.

26 R. Steele, "Parametric Representation of Speech Signals" - Monograph 7, Loughborough University of Technology, 1976.

27 J.N. Holmes, "The JSRU Channel Vocoder" - IEE Proc., vol.127, Pt. F, no.1, Feb. 1980.

28 B. Gold & C.M. Rader, "The Channel Vocoder" - IEEE Trans. Audio Electroacoust., vol.AU-15, no.4, pp.148-160, Dec. 1967.

29 C.P. Smith, "Voice Communication Method using Pattern Matching for Data Compression" - J. Acoust. Soc. America, vol.35, p.805, 1963.

30 A.V. Oppenheim, "A Speech Analysis-synthesis System based on Homomorphic Filtering" - Jour. Acoust. Soc. America, vol.45, no.3, pp.243-248, June 1976.

31 A.M. Noll, "Cepstrum Pitch Determination" - J. Acoust. Soc. America vol.41, pp.293-309, Feb. 1967.

32 J.D. Markel & A.H. Gray, Linear Prediction of Speech - New York: Springer-Verlag, 1976.

33 J. Makhoul, "Linear Prediction: A Tutorial Review" - Proc. of the IEEE vol.63, pp.561-580, April 1975.

34 M.R. Sambur, "An Efficient Linear Prediction Vocoder" - Bell Syst. Tech. Jour., vol.54, no.10, pp.1693-1723, Dec. 1975.

35 B.S. Atal & S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave" - J. Acoust. Soc. America, vol.50 no.2, pp.637-655, Aug. 1971.

36 B.S. Atal & N. David, "On Synthesizing Natural-sounding Speech by Linear Prediction" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp.44-47, 1979.

37 N.S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers" - Proc. of the IEEE, vol.62, pp.611-632, May 1974.

38 H. Morikawa & H. Fujisaki, "Adaptive Analysis of Speech based on a Pole-Zero Representation" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-30, no.1, pp.77-87, Feb. 1982.

39 B.S. Atal & M.R. Shroeder, "Linear Prediction Analysis of Speech based on a Pole-zero Model" - J. Acoust. Soc. America, vol.58, no.1, p.S96, Fall 1975.

40 A.H. Reeves, French Patent 852183, 1938.

41 R.W. Stroh, Optimum and Adaptive DPCM - Ph.D Dissertation, Polytechnic Inst. of Brooklyn, New York 1970.

42 B. Smith, "Instantaneous Companding of Quantized Signals" - Bell Syst. Tech. Jour., vol.36, pp.653-709, May 1957.

43 J. Max, "Quantizing for Minimum Distortion" - IRE Trans. Information Theory, vol.6, pp.16-21, Mar. 1960.

44 P.F. Panter & W. Dite, "Quantization Distortion in Pulse-count Modulation with Non-uniform Spacing of Levels" - IRE Proc. vol.39, pp.44-48, Jan. 1951.

45 M.D. Paez & T.H. Glisson, "Minimum Mean-squared-error Quantization in Speech PCM and DPCM Systems" - IEEE Trans. Commun., vol.COM-20, pp.225-230, April 1972.

46 N.S. Jayant, "Step-size Transmitting Differential Coders for Mobile Telephony" - Bell Syst. Tech. Jour., vol.54, pp.1557-1581, Nov. 1975.

47 P. Noll, "A Comparative Study of Various Quantization Schemes for Speech Encoding" - Bell Syst. Tech. Jour. vol.54, pp.1597-1614, Nov. 1975.

48 J. Huang & P. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables" - IEEE Trans. Commun. Syst., vol.CS-11, pp.289-296, Sept. 1963.

49 N.S. Jayant, "Adaptive Quantization with a One-word Memory" - Bell Syst. Tech. Jour., vol.52, pp.1119-1144, Sept. 1973.

50 P. Castellino, "Bit Rate Reduction by Automatic Adaptation of Quantizer Step-size in DPCM Systems" - Proc. 1974 Int. Zurich Seminar on Digital Commun., p.B6(1).

51 R.M. Wilkinson, "An Adaptive Pulse-code Modulator for Speech" - Proc. IEEE Int. Conf. Comm., Montreal, Canada, pp.1-11 to 1-15, June 1971.

52 C.S. Xydeas, M.N. Faruqui & R. Steele, "Envelope Dynamic Ratio Quantizer" - IEEE Trans. on Commun., vol.COM-28, no.5, pp.720-728, May 1980.

53 C.S. Xydeas & R. Steele, "Dynamic Ratio Quantizer" - IEE Proc. vol. 125, no.1, Jan. 1978.

54 R.E. Crochiere, "A Mid-rise/Mid-tread Quantizer Switch for Improved Idle Channel Performance in Adaptive Coders" - Bell Syst. Tech. Jour. vol.57, no.8, pp.2953-2955, Oct. 1978.

55 C.C. Evci, "Prediction Techniques Applied to Differential Pulse Code Modulation Systems for Encoding Speech Signals" - Ph.D. Thesis, Loughborough University of Technology, 1982.

56 P. Elias, "Predictive Coding - Part I" and "Predictive Coding - Part II" - IRE Trans. Inform. Theory, vol.IT-1, pp.16-33, Mar. 1955.

57 B.M. Oliver, "Efficient Coding" - Bell Syst. Tech. Jour., vol.31, pp.724-750, July 1952.

58 J.B. O'Neal, Jr., "Signal-to-quantizing-noise Ratios for Differential PCM" - IEEE Trans. on Commun., vol.COM-19, pp.568-569, Aug. 1971.

59 J.B. O'Neal, Jr. & R.W. Stroh, "Differential PCM for Speech and Data Signals" - IEEE Trans. on Commun., vol.COM-20, pp.900-912, Oct. 1972.

60 P. Noll, "Non-adaptive and Adaptive DPCM of Speech Signals" - Overdruk uit Polytech, Tijdschr. Ed. Electrotech/Electron. (The Netherlands) no.19, 1972.

61 C.C. Cutler, "Differential Quantization of Communications" - U.S. Patent 2 605 361, July 29, 1952.

62 R.A. McDonald, "Signal-to-noise and Idle Channel Performance of DPCM Systems - Particular Application to Voice Signals" - Bell Syst. Tech. Jour., vol.45, pp.1123-1151, Sept. 1966.

63 R. Steele, "Linear Predictors and Differential PCM for Speech Signals" - Monograph 2, Loughborough University of Technology, 1977.

64 P. Cummiskey, N.S. Jayant & J.L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech" - Bell Syst. Tech. Jour. vol. 52, pp.1105-1118, Sept. 1973.

65 J.D. Gibson, "Unified Development of Algorithms used for Linear Predictive Coding of Speech Signals" - Comput. Elec. Eng., vol.3, pp.75-91, 1976.

66 D.L. Cohn & J.L. Melsa, "A Pitch Compensating Quantizer" - Conf. Rec. IEEE Int. Conf. Acoustics, Speech and Signal Processing, pp.258-261, 1976.

67 S.U.H. Qureshi & G.D. Forney, "A 9.6/16 Kbps Speech Digitizer" - Proc. IEEE Int. Conf. on Commun., pp.30-31 to 30-36, June 1975.

68 J.D. Gibson, "Comparisons and Analyses of Forward and Backward Adaptive Prediction in ADPCM" - Conf. Rec. Nat. Telecommunications Conf., pp.19.2.1-19.2.5, 1978.

69 J.D. Gibson, "Sequentially Adaptive Backward Prediction in ADPCM Speech Coders" - IEEE Trans. on Commun., vol.COM-26, pp.145-150, Jan. 1978.

70 J.D. Gibson, "Sequential Filtering of Quantization Noise in Differential Encoding Systems" - Proc. Int. Conf. Cybernetics and Society, pp.685-689, 1976.

71 J.D. Gibson & V.P. Berglund, "Kalman Backward Adaptive Predictor Coefficient Identification in ADPCM with PCQ" - IEEE Trans. on Commun., vol.COM-28, no.3, MAr. 1980.

72 J.D. Gibson, J.L. Melsa & S.K. Jones, "Digital Speech Analysis using Sequential Estimation Techniques" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-23, no.4, pp.362-369, Aug. 1975.

73 D.L. Cohn & J.L. Melsa, "The Residual Encoder - An Improved ADPCM System for Speech Digitization" - IEEE Trans. on Commun., vol. COM-23, no.9, pp.935-941, Sept. 1975.

74 C.C. Evci, R. Steele & C.S. Xydeas, "Sequential Gradient Estimation Predictor for Speech Signals" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Washington D.C., pp.723-726, 1979.

75 J.D. Gibson, S.K. Jones & J.L. Melsa, "Sequentially Adaptive Prediction and Coding of Speech Signals" - IEEE Trans. on Commun., vol.COM-22, no.11, pp.1789-1797, Nov. 1974.

76 P. Cummiskey, Adaptive DPCM for Speech Processing - Ph.D Dissertation, Newark College of Engineering, Newark, N.J. 1973.

77 B. Friedlander, "Lattice Filters for Adaptive Processing" - Proc. of the IEEE vol.70, no.8, pp.829-867, Aug. 1982.

78 J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-25, no.5, pp.423-428, Oct. 1978.

79 J. Makhoul & L.K. Cossell, "Adaptive Lattice Analysis of Speech" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-29, no.3, pp.654-658, June 1981.

80 B.S. Atal & M.R. Shroeder, "Predictive Coding of Speech Signals" - Bell Syst. Tech. Jour. vol.49, pp.1973-1986, Oct. 1970.

81 B.S. Atal & M.R. Shroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-27, pp.247-254, June 1979.

82 B.S. Atal, "Predictive Coding of Speech at Low Bit Rates" - IEEE Trans. on Commun., vol.COM-30, no.4, pp.600-614, April 1982.

83 A.J. Goldberg & H.L. Shaffer, "A Real-time Adaptive Predictive Coder using Small Computers" - IEEE Trans. on Commun. vol.COM-23, no.12, pp.1443-1451, Dec. 1975.

84 L.R. Rabiner, M.J. Cheng, A.E. Rosenberg and C.A. McGonegal, "A Comparative Study of Several Pitch Detection Algorithms" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-24, pp.399-418, Oct.1976.

85 M.J. Ross, H.L. Shaffer, A. Cohen, R. Freudberg and H.J. Manley, "Average Magnitude Difference Function Pitch Extractor" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-22, pp.353-362, Oct. 1974.

86 N.S. Jayant, "Pitch-adaptive DPCM Coding of Speech with Two-bit Quantization and Fixed Spectrum Prediction" - Bell Syst. Tech. Jour., vol.56, pp.439-454, MAr. 1977.

87 C.S. Xydeas & R. Steele, "Pitch Synchronous Differential Predictive Encoding System" - Electronics Letters, vol.12, no.5, July 1976.

88 C.S. Xydeas & R. Steele, "Pitch Synchronous First Order DPCM System" - Electronics Letters, vol.12, no.4, Feb. 1976.

89 R. Steele, Delta Modulation Systems - Pentech Press, London, 1975.

90 F. deJager, "Delta Modulation, a Method of PCM Transmission using the 1-unit Code" - Philips Res. Rept., pp.442-466, 1952.

91 J.E. Abate, "Linear and Adaptive Delta Modulation" - Proc. of the IEEE, vol.55, pp.298-308, Mar. 1967.

92 H. Van De Weg, "Quantization Noise of a Single Integration Delta Modulation System with an N-digit Code" - Philips Res. Rep., pp.367-385, Oct. 1953.

93 D.J. Goodman, "Delta Modulation Granular Quantization Noise" - Bell Syst. Tech. Jour., pp.1197-1218, May-June 1969.

94 E.N. Prontanotarios, "Slope-overload Noise in Differential PCM

Systems" - Bell Syst. Tech. Jour., pp.2118-2161, 1967.

95 L.J. Greenstein, "Slope Overload Noise in Linear Delta Modulators with Gaussian Inputs" - Bell Syst. Tech. Jour., pp.387-422, Mar. 1973.

96 J.B. O'Neal, Jr., "Delta Modulator Quantizing Noise: Analytical and Computer Simulation Results for Gaussian and Television Input Signals" - Bell Syst. Tech. Jour., pp.117-142, Jan. 1966.

97 R. Steele, "SNR Formula for Linear Delta Modulation with Band-Limited Flat and RC-Shaped Gaussian Signals" - IEEE Trans. on Commun., vol.COM-28, no.12, pp.1977-1984, Dec. 1980.

98 P.T. Nielsen, "On the Stability of a Double Integration Delta Modulator" - IEEE Trans. Commun. Technol., vol.COM-19, pp.364-366, June 1971.

99 C.C. Cutler, "Delayed Encoding: Stabilizer for Adaptive Coders" - IEEE Trans. on Commun., vol.COM-19, pp.898-907, Dec. 1971.

100 L.H. Zetterberg & J. Uddenfeldt, "Adaptive Delta Modulation with Delayed Decision" - IEEE Trans. on Commun., vol.COM-22, no.9, Sept. 1974.

101 J.E. Flood & M.J. Hawksford, "Adaptive Delta-Sigma Modulator Using Pulse Grouping Techniques" - Joint Conf. on Digital Processing of Signals in Commun., Loughborough University of Technology, U.K., April 1972.

102 M.R. Winkler, "High Information Delta Modulation" - in IEEE Int. Conv. Rec., pt.8, pp.260-265, Mar. 1963.

103 N.S. Jayant, "Adaptive Delta Modulation with a One-bit Memory" - Bell Syst. Tech. Jour., pp.321-342, Mar. 1970.

104 A.T. Kyaw & R. Steele, "Constant Factor DM" - Electronics Letters vol.9, no.4, pp.96-97, Feb. 1973.

105 S.J. Brolin & J.M. Brown, "Companded Delta Modulator for Telephony" - IEEE Trans. Commun. Technol., vol.COM-16, pp.157-162, Feb. 1968.

106 J.A. Greefkes, "A Digitally Controlled Delta Codec for Speech Transmission" - Proc. IEEE Int. Conf. on Comm. pp.7-33-7-48, 1970.

107 J.A. Greefkes & F. DeJager, "Continuous Delta Modulation" - Philips Res. Rep., vol.23/2, pp.233-246, 1968.

108 V.R. Dhadesugoor, C. Ziegler & D.L. Shilling, "Delta Modulator in Packet Voice Networks" - IEEE Trans. on Commun., vol.COM-28, no.1, pp.33-51, Jan. 1980.

109 A. Tomozowa & H. Kaneko, "Companded Delta Modulator for Telephone Transmission" - IEEE Trans. Commun. Techno., vol.COM-16, pp.149-157,

Feb. 1968.

110 P. Noll, "On Predictive Quantizing Schemes" - Bell Syst. Tech. Jour., vol.57, pp.1499-1532, May-June 1978.

111 H.A. Spang III and P.M. Schultheiss, "Reduction of Quantization Noise by use of Feedback" - IRE Trans. Commun. Syst., vol.CS-10, pp.373-380, Dec. 1962.

112 J. Makhoul & M. Berouti, "Adaptive Noise Spectral Shaping and Entropy Coding in Predictive Coding of Speech" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-27, pp.63-73, Feb. 1979.

113 J. Makhoul & M. Berouti, "High Quality Adaptive Predictive Coding of Speech" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Tulsa, Oklahoma, pp.303-306, 1978.

114 E.G. Kimme & F.F. Kuo, "Synthesis of Optimal Filters for a Feedback Quantization System" - IEEE Trans. Circuit Theory, vol.CT-10, pp.405-413, Sept. 1963.

115 M.R. Shroeder, B.S. Atal & J.L. Hall, "Optimising Digital Speech Coders by Exploiting Masking Properties of the Human Ear" - J. Acoust. Soc. America, vol.66, no.6, pp.1647-1652, Dec. 1979.

116 R.C. Brainard and J.C. Candy, "Direct-feedback Coders: Design and Performance with Television Signals" - Proc. of the IEEE, vol.57, pp. 776-786, May 1969.

117 J.V. Bodycomb & A.H. Haddad, "Some Properties of a Predictive Quantizing System" - IEEE Trans. on Commun., vol.COM-20, pp.682-684, Oct. 1970.

118 J.L. Melsa & R.B. Kolstad, "Kalman Filtering of Quantization Error in Digitally Processed Speech" - Conf. Rec. Int. Conf. Commun., pp.310-313, 1977.

119 R.B. Ash, Information Theory - John Wiley and Sons, New York 1967.

120 J.B. O'Neal, Jr., "Differential Pulse-code Modulation (PCM) with Entropy Coding" - IEEE Trans. Inform. Theory, vol.IT-22, pp.169-174, Mar. 1976.

121 K. Virupaksha & J.B. O'Neal, Jr., "Entropy-coded Adaptive Differential Pulse-code Modulation (DPCM) for Speech" - IEEE Trans. on Commun., vol.COM-22, pp.777-787, June 1974.

122 B.S. Atal & M.R. Shroeder, "Improved Quantizer for Adaptive Predictive Coding of Speech Signals at Low Bit Rates" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp.535-538, April 1980.

123 D.A. Huffman, "A Method for the Construction of Minimum-Redundancy

Codes" - IRE Proc. pp.1098-1101, Sept. 1952.

124 H.G. Fehn & P. Noll, "Multipath Search Coding of Stationary Signals with Applications to Speech" - IEEE Trans. on Commun., vol.COM-30, no.4, pp.687-701, April 1982.

125 A. Buzo, A.H. Gray, Jr., R.M. Gray & J.D. Markel, "Speech Coding based upon Vector Quantization" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp.15-18, April 1980.

126 A. Buzo, A.H. Gray, Jr., R.M. Gray & J.D. Markel, "Speech Coding based upon Vector Quantization" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-28, no.5, pp.562-574, Oct. 1980.

127 Y. Matsuyama & R.M. Gray, "Universal Tree Encoding for Speech" - IEEE Trans. on Inform. Theory, vol.IT-27, no.1, pp.31-40, Jan. 1981.

128 Y. Matsuyama & R.M. Gray, "Voice Coding and Tree Encoding Speech Compression Systems based upon Inverse Filter Matching" - IEEE Trans. on Commun., vol.COM-30, no.4, pp.711-720, April 1982.

129 Y. Linde, A. Buzo & R.M. Gray, "An Algorithm for Vector Quantization Design" - IEEE Trans. on Commun., vol.COM-28, no.1, pp.84-95, Jan. 1980.

130 T. Berger, Rate Distortion Theory - Englewood Cliffs, New Jersey, Prentice-Hall, 1971.

131 A.J. Viterbi, "Trellis Encoding of Memoryless Discrete Time Sources with a Fidelity Criterion" - IEEE Trans. Inform. Theory, vol.IT-20, no.3, pp.325-336, May 1974.

132 F. Jelinik, "Tree Encoding of Memoryless Time-discrete Sources with a Fidelity Criterion" - IEEE Trans. Inform. Theory, vol.IT-15, pp.584-590, Sept. 1969.

133 S.G. Wilson & S. Hussain, "Adaptive Tree Encoding of Speech at 8000 bps with a Frequency-weighted Error Criterion" - IEEE Trans. on Commun., vol.COM-27, pp.165-170, Jan. 1979.

134 F. Itakura & S. Saito, "On the Optimum Quantization of Feature Parameters in the PARCOR Speech Synthesizer" - in IEEE Int. Conf. on Speech Commun. and Process., pp.434-437, April 1972.

135 R. Viswanathan & J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-23, pp.209-231, June 1975.

136 B.H. Juang, D.Y. Wong & A.H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-30, no.2, pp.294-303, April 1982.

137 J.B. Anderson & J.B. Bodie, "Tree Encoding of Speech" - IEEE Trans. Inform. Theory, vol.IT-21, pp.379-387, July 1975.

138 N.S. Jayant & S.A. Christensen, "Tree Encoding of Speech using the (M,L) algorithm and Adaptive Quantization" - IEEE Trans. on Commun. vol.COM-26, pp.1376-1379, Sept. 1978.

139 L.C. Stewart, R.M. Gray & Y. Linde, "The Design of Trellis Waveform Coders" - IEEE Trans. on Commun., vol.COM-30, no.4, pp.702-709, April 1982.

140 J.M. Tribolet & R.E. Crochiere, "Frequency Domain Coding of Speech" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-27, no.5, pp.512-530, Oct. 1979.

141 R.E. Crochiere, S.A. Webber & J.L. Flanagan, "Digital Coding of Speech in Sub-bands" - Bell Syst. Tech. Jour. vol.55, pp.1069-1085, Oct. 1976.

142 R.E. Crochiere, "On the Design of Sub-band Coders for Low-bit Rate Speech Communication" - Bell Syst. Tech. Jour. vol.56, pp.747-770, May-June 1977.

143 M.H. Ackroyd, Digital Filters - Butterworths, London 1973.

144 R.E. Crochiere & L.R. Rabiner, "Interpolation and Decimation of Digital Signals - A Tutorial Review" - Proc. of the IEEE, vol.69, no.3, Mar. 1981.

145 D. Esteban & C. Galand, "Application of Quadrature Mirror Filters to Split Band Voice Coding Schemes" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Hartford, Conn., pp.191-195, May 1977.

146 R.E. Crochiere, R.V. Cox & J.D. Johnston, "Real-time Speech Coding" - IEEE Trans. on Commun., vol.COM-30, no.4, pp.621-633, April 1982.

147 R.E. Crochiere, "Digital Signal Processor: Sub-Band Coding" - Bell Syst. Tech. Jour. vol.60, no.7, pp.1632-1653, Sept. 1981.

148 D.S.K.I. Lee, C.S. Xydeas, S.N. Koh & S.J. Perkins, "64 Kbps Coding of 7 KHz Bandwidth Speech" - Colloquium on 'Digital Processing of Speech', IEE Savoy Place, London, pp.4/1-4/7, April 1983.

149 British Telecom Research Laboratories, R9.2.1, private communication.

150 R.E. Crochiere, "An Analysis of 16 Kbps Sub-band Coder Performance: Dynamic Range, Tandem Connections and Channel Errors" - Bell Syst. Tech. Jour. vol.57, pp.2927-2952, Oct. 1978.

151 R.E. Crochiere, "A Novel Approach for Implementing Pitch Prediction in Sub-band Coding" - Proc. IEEE Int. Conf. on Acoustics, Speech and

Signal Processing, Washington D.C., pp.526-529, 1979.

152 R.S. Cheung & R.L. Winslow, "High Quality 16 Kbps Voice Transmission: The Subband Coder Approach" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp.319-322, April 1980.

153 A.J. Barabell & R.E. Crochiere, "Sub-band Coder Design Incorporating Quadrature Mirror Filters and Pitch Prediction" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Washington D.C., pp.530-533, 1979.

154 C. Galand & D. Esteban, "16 Kbps Sub-Band Coder Incorporating Variable Overhead Information" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.1684-1687, May 1982.

155 A.J. Goldberg, R.L. Freudberg & R.S. Cheung, "High Quality 16 Kbps Voice Transmission" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp.244-246, Hartford, Conn., 1976.

156 V. Gupta & K. Virupaksha, "Performance Evaluation of Adaptive Quantizers for a 16 Kbps Sub-Band Coder" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp. 1688, May 1982.

157 C.D. Heron, R.E. Crochiere & R.V. Cox, "A 32-band Sub-band/Transform Coder Incorporating Vector Quantization for Dynamic Bit Allocation" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Boston, pp.1276-1279, April 1983.

158 J.D. Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp.291-294, April 1980.

159 D. Malah, R.E. Crochiere & R.V. Cox, "Performance of Transform and Subband Coding Systems Combined with Harmonic Scaling of Speech" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-29, no.2, pp.273-283, April 1981.

160 T.A. Ramstad, "Sub-Band Coder with a Simple Adaptive Bit-Allocation Algorithm - A Possible Candidate for Digital Mobile Telephony?" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.203-207, May 1982.

161 R. Zelinski & P. Noll, "Adaptive Transform Coding of Speech Signals" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-25, no.4, pp.299-309, Aug. 1977.

162 R. Zelinski & P. Noll, "Approaches to Adaptive Transform Speech Coding at Low Bit Rates" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-27, no.1, pp.89-95, Feb. 1979.

163 R.V. Cox & R.E. Crochiere, "Real-time Simulation of Adaptive Transform Coding" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-29, no. 2, pp.147-154, April 1981.

164 N. Ahmed, T. Natarajan & K.R. Rao, "Discrete Cosine Transform" - IEEE Trans. on Computers, pp.90-93, Jan. 1974.

165 J.M. Tribolet & R.E. Crochiere, "A Vocoder-driven Adaptation Strategy for Low-bit-rate Transform Coding of Speech" - Proc. Int. Conf. on Digital Signal Processing, Florence, Italy, pp.638-642, Sept. 1978.

166 S.J. Campanella & G.S. Robinson, "A Comparison of Orthogonal Transformations for Digital Speech Processing" - IEEE Trans. Commun. Technol., vol.COM-19, pt.1, pp.1045-1049, Dec. 1971.

167 J.L. Flanagan & R.M. Golden, "Phase Vocoder" - Bell Syst. Tech. Jour. vol.45, pp.1493-1509, 1966.

168 M.R. Portnoff, "Implementation of the Digital Phase Vocoder using the Fast Fourier Transform" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-24, no.3, pp.243-248, June 1976.

169 H. Gethoffer, "Polar Plane Block Quantization of Speech Signals Using Bit-pattern Matching Techniques" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Hartford, CT, pp.200-203, May 1977.

170 R. Viswanathan, W. Russell & J. Makhoul, "Voice-Excited LPC Coders for 9.6 Kbps Speech Transmission" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Washington D.C., pp.558-561, 1979.

171 C.K. Un & D.T. Magill, "The Residual-Excited Linear Prediction Vocoder with Transmission Rate Below 9.6 Kbps" - IEEE Trans. on Commun., vol.COM-23, no.12, pp.1466-1474, Dec. 1975.

172 D. Esteban, C. Galand, D. Mauduit & J. Menez, "9.6/7.2 Kbps Voice Excited Predictive Coder (VEPC)" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Tulsa, OK, pp.307-311, April 1978.

173 J. Makhoul & M. Berouti, "High-Frequency Regeneration in Speech Coding Systems" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Washington D.C., pp.428-431, 1979.

174 E.E. David, Jr., M.R. Shroeder, B.F. Logan, & A.J. Prestigiacomo, "Voice-Excited Vocoders for Practical Speech Bandwidth Reduction" - IRE Trans. Inform. Theory, vol.IT-8, pp.S101-S105, Sept. 1962.

175 M.D. Dankberg & D.Y. Wong,"Development of a 4.8-9.6 Kbps RELP Vocoder" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Washington D.C., pp.554-557, 1979.

176 H. Katterfeldt, "A DFT-based Residual-Excited Predictive Coder (RELP) for 4.8 and 9.6 Kbps" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Atlanta, pp.824-827, Mar. 1981.

177 C.K. Un & J.R. Lee, "On Spectral Flattening Techniques in Residual-Excited Linear Prediction Vocoding" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.216-219, May 1982.

178 H.E. Watkins, "Description of a Hybrid 7.2 Kbps Vocoder" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Washington D.C., pp.546-549, 1979.

179 B.M. Abzug, "Using the Prediction Residual to Improve LPC Synthesis for 9600 Bps Applications" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Atlanta, pp.812-815, Mar. 1981.

180 C.K. Un & W.Y. Sung, "A 4800 Bps LPC Vocoder with Improved Excitation" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp.142-145, April 1980.

181 C.J. Weinstein, "A Linear Predictive Vocoder with Voiced Excitation" - Proc. EASCON '75, pp.30A-30G, Washington D.C., Sept. 1975.

182 D. Malah & J.L. Flanagan, "Frequency Scaling of Speech Signals by Transform Techniques" - Bell Syst. Tech. Jour., vol.60, no.9, pp.2107-2156, Nov. 1981.

183 D. Malah, "Time-Domain Algorithms for Harmonic Bandwidth Reduction and Time Scaling of Speech Signals" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-27, no.2, pp.121-133, April 1979.

184 J.L. Flanagan & S.W. Christensen, "Technique for Frequency Division /Multiplication of Speech Signals" - J. Acoust. Soc. America, vol. 68, no.4, pp.1061-1068, Aug. 1980.

185 J.L. Melsa & A.K. Pande, "Mediumband Speech Encoding Using Time Domain Harmonic Scaling and Adaptive Residual Coding" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Atlanta, pp.603-606, Mar. 1981.

186 J.L. Flanagan & S.W. Christensen, "Computer Studies on Parametric Coding of Speech Spectra" - J. Acoust. Soc. America, vol.68, no.2, pp.420-430, Aug. 1980.

187 L.B. Almeida & J.M. Tribolet, "Harmonic Coding: A Low Bit-rate, Good-Quality Speech Coding Technique" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.1664-1667, May 1982.

188 P. Noll, "Effects of Channel Errors on the Signal-to-noise Performance of Speech-encoding Systems" - Bell Syst. Tech. Jour., vol.54, pp.1615-1636, Nov. 1975.

189 D.J. Goodman & A. Gersho, "Theory of an Adaptive Quantizer" - IEEE Trans. on Commun., vol.COM-22, pp. 1037-1045, Aug. 1974.

190 D.J. Goodman & R.M. Wilkinson, "A Robust Adaptive Quantizer" - IEEE Trans. on Commun. vol.COM-23, pp.1362-1365, Nov. 1975.

191 L.S. Moye, "Self-adaptive Filter Predictive-coding System" - Proc. Int. Zurich Seminar Inst. Syst. for Speech, Video and Data Comm., Paper No. F3, 1972.

192 R. Steele and D.J. Goodman, "Detection and Selective Smoothing of Transmission Errors in Linear PCM" - Bell Syst. Tech. Jour., vol.51, pp.399-409, Mar. 1977.

193 R. Steele & N.S. Jayant, "Statistical Block Protection Coding for DPCM-AQF Speech" - IEEE Trans. on Commun., vol.COM-28, no.11, pp.1899-1907, Nov. 1980.

194 R. Viswanathan, W. Russell & A. Higgins, "Noise-Channel Performance of 16 Kbps APC Coders" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Atlanta, pp.615-618, Mar. 1981.

195 D. Cointot, "A 32-Kbit/sec ADPCM Coder Robust to Channel Errors" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.964-967, May 1982.

196 Y. Yatsuzuka & H.G. Suyderhoud, "A 32 Kbps ADPCM Encoding with Variable Initially Large Leakage and Adaptive Dual Loop Predictors" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.976-979, May 1982.

197 M.L. Honig & D.G. Messerchmitt, "Comparison of Adaptive Prediction Algorithms in ADPCM" - IEEE Trans. on Commun., vol.COM-30, no.7, pp.1775-1785, July 1982.

198 P. Combescure, A. Le Guyader & M. Haghiri, "ADPCM Algorithms Applied to Wide-band Speech Encoding" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, May 1982.

199 A. Le Guyader, "Speech Signal Algorithms and Coding Techniques" - Translated from Technical Notice NT/LAS/TSS/18, CNET Lannion A, June 1980 (in French).

200 A. Le Guyader & A. Gilliore, "Comparison of Basic and Simplified Sequential Algorithms for the Computation of Lattice Filter Predictor Coefficients in ADPCM Coding of Speech" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.1676-1679, May 1982.

201 F.S. Yeoh & C.S. Xydeas, "A Transform Approach to Split-band Coding Schemes" - to be published in IEE Proc. on Commun., Radar and Signal Processing.

202 F.S. Yeoh & C.S. Xydeas, "Split-band Coding of Speech Signals Using a Transform Technique" - submitted to the International Conference on Communications (ICC '84) to be held in Amsterdam, Holland, May 1984.

203 C.S. Xydeas, "Embedding Data into Speech and Video Signals" - Proc. Mediterranean Electrotechnical Conf. MELECON '83, p.B11.10, May 1983.

204 R. Steele & D. Vitello, "Embedding Data in Speech using Scrambling Techniques" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.1801-1804, May 1982.

205 L.D. Davisson, "Rate Distortion Theory and Application" - Proc. of the IEEE, vol.60, pp.800-808, July 1972.

206 J.J. Dubnowski & R.E. Crochiere, "Variable Rate Coding of Speech" - Bell Syst. Tech. Jour.'vol.58, no.3, pp.577-600, Mar. 1979.

207 D.J. Quarmby, "Using the NEC 7720 in Speech Processing" - Colloquium on 'Digital Processing of Speech', IEE Savoy Place, London, pp.8/1-8/6, April 1983.

208 A.J. Goldberg, "Practical Implementations of Speech Waveform Coders for the Present Day and for the Mid 1980s" - J. Acoust. Soc. America, vol.66, no.6, pp.1653-1657, Dec. 1979.

209 M. Honda, N. Kitawaki & F. Itakura, "Adaptive Bit Allocation Scheme in Predictive Coding of Speech" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.1672-1675, May 1982.

210 T. Nishitani, S. Aikoh, T. Araseki, K. Ozawa & R. Maruta, "A 32 Kbps Toll Quality ADPCM Codec Using a Single Chip Signal Processor" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, May 1982.

211 J.B. O'Neal, Jr., "A Bound on Signal-to-quantizing Noise Ratios for Digital Encoding Systems" - Proc. of the IEEE, vol.55, pp.287-292, Mar.1967.

212 C.S. Xydeas, F.S. Yeoh & S.N. Koh, "Noise Spectral Shaping Applied to Coarse Quantization Differential Speech Coders" - Proc. Mediterranean Electrotechnical Conf. MELECON '83, p.C1.08, May 1983.

213 F.S. Yeoh & C.S. Xydeas, "Adaptive Noise Shaping in ADPCM at 16 Kbps" - Proceedings of the Second International Conference on New Systems and Services in Telecommunications, Liege, Belgium, Nov. 1983.

214 F.S. Yeoh & C.S. Xydeas, "Noise Reduction in ADPCM-AQJ Systems using Quantizer Correction at the Receiver" - Electronics Letters vol.19, no.11, pp.420-421, May 1983.

215 F.S. Yeoh & C.S. Xydeas, "Noise Shaping in Backward Adaptive ADPCM at 16 Kbps" - submitted to the Proc. IERE.

216 P. Combescure, A. Le Guyader & A. Gilliore, "Quality Evaluation

of 32 Kbit/s Coded Speech by Means of Degradation Category Ratings" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.988-991, May 1982.

217 W.R. Daumer, "Subjective Evaluation of Several Efficient Speech Coders" - IEEE Trans. on Commun., vol.COM-30, no.4, pp.655-662, April 1982.

218 N. Weiner, The Extrapolation, Interpolation and Smoothing of Stationary Time Series - Wiley, 1949.

219 D.K. Fadeev & V.N. Fadeeva, Computational Methods of Linear Algebra - San Francisco, Calif., Freeman 1963.

220 N. Levinson, "The Weiner RMS Error Criterion in Filter Design and Prediction" - J. Math. Phys., vol.25, no.4, pp.261-278, 1947.

221 J. Durbin, "The Fitting of Time-series Models" - Rev. Inst. Int. Statist., vol.28, no.3, pp.233-243, 1960.

222 R.J. Wang & S. Treitel, "The Determination of Digital Weiner Filters by Means of Gradient Methods" - Geophysics vol.38, no.2, pp.310-326, April 1973.

223 J.L. Melsa, et al, "Study of Sequential Estimation Methods for Speech Digitization" - Univ. of Notre Dame, Final Report, DCA Contract No. DCA 100-74-C-0037, June 1975.

224 C.C. Evci, R. Steele & C.S. Xydeas, "DPCM-AQF using Second-order Adaptive Predictors for Speech Signals" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-29, no.3, pp.337-341, June 1981.

225 C.S. Xydeas & C.C. Evci, "A Comparative Study of DPCM-AQF Speech Coders for Bit Rates of 16-32 Kbps" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.1680-1684, May 1982.

226 B. Farhang-Boroujeny & L.F. Turner, "Fast Converging Stochastic Gradient Algorithm" - IEE Proc., vol.128, Pt. F, no.5, pp.271-274, Oct. 1981.

227 B.S. Atal & J.R. Remde, "Split-band APC System for Low Bit-rate Encoding of Speech" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Atlanta, pp.599-602, Mar. 1981.

228 N.J. Miller, "Pitch Detection by Data Reduction" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-23, no.1, pp.72-79, Feb.1975.

229 M.M. Sondhi, "New Methods of Pitch Extraction" - IEEE Trans. Audio Electroacoust., vol.AU-16, pp.262-266, June 1968.

230 J.W. Mark & P.P. Dasiewicz, "Application of Iterative Algorithms to

Adaptive Predictive Coding" - Journal of Cybernetics, no. 7, pp.279-317, 1977.

231 R. Bastian, "Subjective Improvements in DPCM-AQ Performance based on Adaptive Noise Shaping" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-29, no.5, pp.1067-1071, Oct. 1981.

232 N. Dal Degan & C. Scagliola, "Optimal Noise Shaping in Predictive Coding of Speech" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp.539-542, April 1980.

233 B.J. McDermott & C. Scagliola, "The Perception of Spectrally Shaped Additive Noise in Speech" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.196-198, May 1982.

234 K.D. Kryter, "Methods for the Calculation and Use of the Articulation Index" - J. Acoust. Soc. America, vol.34, pp.1689-1697, 1972.

235 A. Habibi, "Survey of Adaptive Image Coding Techniques" - IEEE Trans. on Commun., vol.COM-25, no.11, pp.1275-1284, Nov. 1977.

236 W.H. Chen, C. Smith & S.C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform" - IEEE Trans. on Commun., vol. COM-25, no.9, pp.1004-1009, Sept. 1977.

237 J. Makhoul, "A Fast Cosine Transform in One and Two Dimensions" - IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-28, no.1, pp.27-34, Feb. 1980.

238 M.J. Narasimha & A.M. Peterson, "On the Computation of the Discrete Cosine Transform" - IEEE Trans. on Commun., vol.COM-26, no.6, pp.924-936, June 1978.

239 J.M. Tribolet & R.E. Crochiere, "An Analysis/Synthesis Framework for Transform Coding of Speech" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing. Washington D.C., pp.81-84, 1979.

240 J.M. Tribolet & R.E. Crochiere, "A Modified Adaptive Transform Scheme with Post-Processing Enhancement" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Denver, pp.336-339, April 1980.

241 K. Annamalaj & T. Fjallbrant, "An Adaptive Transform Coding System with Short Primary Blocklengths and Frequency Domain Quantization using Feedback Adaptation" - Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Paris, pp.1696-1699, May 1982.

242 J.M. Tribolet, P. Noll, B.J. McDermott & R.E. Crochiere, "A Comparison of the Performance of Four Low Bit Rate Speech Waveform Coders" - Bell Syst. Tech. Jour., vol.58, pp.699-712, Mar. 1979.

243 J.M. Tribolet, P. Noll, B.J. McDermott & R.E. Crochiere, "A Study of Complexity and Quality of Speech Waveform Coders" - Proc. IEEE

Int. Conf. on Acoustics, Speech and Signal Processing, Tulsa, OK, pp.1586-1590, April 1978.

244 G. Bertocci, B.W. Schoenherr & D.G. Messerchmitt, "An Approach to the Implementation of a Discrete Cosine Transform" - IEEE Trans. on Commun., vol.COM-30, no.4, pp.635-641, April 1982.