Differential Encoding Techniques

Applied to Speech Signals

BY

Constantinos S. Xydeas

D.E. ("VRANAS" Higher School of Electronics, Greece).

M.Sc. (Loughborough University of Technology, 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 1978.

Supervisor: Raymond Steele, B.Sc., Ph.D., C.Eng., M.I.E.E.
Department of Electronic and Electrical Engineering.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Dr. Raymond Steele whom I am privileged to have had as supervisor. His keen enthusiasm for work has been a constant source of encouragement to me, and his broad knowledge in the field of Speech Digitization provided the technical guidance and valuable criticisms which greatly influenced the course of this research. I am thankful for his considered quidance in the manuscript of the thesis, and very grateful for his enthusiastic support of my Research Fellow appointment in the Department of Electronic and Electrical Engineering of Loughborough University of Technology.

I would like to thank Dr. N.S. Jayant and Dr. D.J. Goodman of Bell Laboratories, U.S.A., for their valuable comments and suggestions on the subject of Dynamic Ratio Quantization. I am also thankful to Professor M.N. Faruqui of the Institute of Technology, Kharagpur, India, for his suggestions and for the many fruitful discussions we had on Adaptive Speech Digitizers and Low-Bit Rate Coding of Speech Signals. I also wish to thank my colleagues Dr. F. Sakane and Dr. J. Daley for the numerous stimulating discussions on Processing and Modelling of Speech Signal.

I would like to thank Professor J.W.R. Griffiths, Head of the Electronic and Electrical Engineering Department of Loughborough University of Technology for providing all the research facilities. I am thankful to the Staff and Student members of this Department whose assistance was valuable to my work.

Thanks are also due to Mrs. B. Wright for her forbearance in typing this thesis.

Finally I express my deepest gratitude to my parents for their moral and financial support and to my wife, Helen, whose love and confidence in me, has made my study possible.

To her I dedicate this thesis.

List of Principal Symbols

$X(t)$	:	input speech waveform
$u(t)$	:	Vocal tract impulse response
$\left[X(t)\right]^c$	:	Cepstrum of $X(t)$
a_j	:	coefficients of a Linear predictor
β_n	:	excitation pulses
$\{X_n\}$	:	sequence of input samples
$\{\hat{X}_n\}$	:	sequence of recovered samples
$\{L_n\}$	:	sequence of binary words at the output of the Encoder
$\{L_n'\}$	:	received L_n
ξ_i	:	decision levels in a zero memory quantizer
$K(X), K^{-1}(X)$	:	zero memory non-linearity and its inverse
σ_x^2	:	variance of the input signal
σ_e^2	:	mean squared quantization error
$AL(X)$	:	characteristic of an A-Law compander
$ML(X)$	:	characteristic of an μ-Law compander
δ_n	:	step size of an adaptive quantizer at the nth instant
Y_n	:	nth sample at the output of the Local Decoder in a Differential Encoder
e_n	:	difference sample between X_n and Y_n
Ψ	:	autocorrelation function of the input speech signal
G	:	autocorrelation vector in LPC analysis
R	:	autocorrelation vector in LPC analysis
ρ_i	:	ith autocorrelation coefficient
$H(S)$	:	Entropy of the source
$\bar{L}$	:	average length of the Coding Procedure

f_c	:	frequency band of the input signal
f_p	:	sampling frequency in a Delta Modulator
A_{opt}	:	Optimum prediction coefficient's vector
g and g_o	:	optimizing constants in sequentially updating prediction algorithms
PF	:	fixed section of a predictor
PA	:	adaptive section of a predictor
$\{S_i\}$	:	ith pitch sequence of input samples
$\{\hat{S}_i\}$	:	the decoded sequence of $\{S_i\}$
$\{n_i\}$	:	sequence of noise samples
λ	:	a constant used in the formation of the PSFOD and PSDPE difference sequences
β_{ik}	:	prediction coefficient in the pitch loop of the PSFOD system
$Q_{(j)}$, $T_{(j)}$	:	the thresholds and output levels of an adaptive quantizer
$M_{(j)}$	:	time invariant expansion - contraction coefficients
f_k	:	Non Linear Transform's TR output sample
F_k	:	Non Linear Element's output sample
$\{en_k\}$	:	envelope samples added to the input samples $\{X_k\}$
snr_v	:	signal-to-noise ratio of the DRQ
$\{U_k\}$	:	sequence at the output of the Envelope-DRQ's Local Decoder
$\{V_k\}$	:	sequence of samples obtained by adding $\{en_k\}$ to $\{X_k\}$

CONTENTS

SYNOPSIS

The increasing use of digital communication systems has produced a continuous search for efficient methods of speech encoding.

This thesis describes investigations of novel differential encoding systems. Initially Linear First Order DPCM systems employing a simple delayed encoding algorithm are examined. The systems detect an overload condition in the encoder, and through a simple algorithm reduce the overload noise at the expense of some increase in the quantization (granular) noise. The signal-to-noise ratio (snr) performance of such a codec has 1 to 2 dB's advantage compared to the First Order Linear DPCM system.

In order to obtain a large improvement in snr the high correlation between successive pitch periods as well as the correlation between successive samples in the voiced speech waveform is exploited. A system called "Pitch Synchronous First Order DPCM" (PSFOD) has been developed. Here the difference sequence formed between the samples of the input sequence in the current pitch period and the samples of the stored decoded sequence from the previous pitch period are encoded. This difference sequence has a smaller dynamic range than the original input speech sequence enabling a quantizer with better resolution to be used for the same transmission bit rate. The snr is increased by 6 dB compared with the peak snr of a First Order DPCM codec.

A development of the PSFOD system called a Pitch Synchronous Differential Predictive Encoding system (PSDPE) is next investigated.

The principle of its operation is to predict the next sample in the voiced-speech waveform, and form the prediction error which is then subtracted from the corresponding decoded prediction error in the previous pitch period. The difference is then encoded and transmitted. The improvement in snr is approximately 8 dB compared to an ADPCM codec, when the PSDPE system uses an adaptive PCM encoder. The snr of the system increases further when the efficiency of the predictors used improve. However, the performance of a predictor in any differential system is closely related to the quantizer used. The better the quantization the more information is available to the predictor and the better the prediction of the incoming speech samples. This leads automatically to the investigation in techniques of efficient quantization. A novel adaptive quantization technique called Dynamic Ratio quantizer (DRQ) is then considered and its theory presented. The quantizer uses an adaptive non-linear element which transforms the input samples of any amplitude to samples within a defined amplitude range. A fixed uniform quantizer quantizes the transformed signal. The snr for this quantizer is almost constant over a range of input power limited in practice by the dynamic range of the adaptive non-linear element, and it is 2 to 3 dB's better than the snr of a One Word Memory adaptive quantizer.

Digital computer simulation techniques have been used widely in the above investigations and provide the necessary experimental flexibility. Their use is described in the text.

CHAPTER I

DIGITAL SPEECH COMMUNICATIONS - ORGANIZATION OF THESIS

1.1 INTRODUCTION.

Man can communicate to his fellows subtle changes in his mood, emotions, likes, dislikes, belief, disbelief, basic wants, appetites, and so forth by facial and body movements, the so-called body language. But this method of communication is useless in conveying intellectual arguments. Even the best "body-talker" would be hard pressed to explain Pythagoras theorem! To communicate intellectually and with precision we need to speak. Speech is not just the making of complex sounds but the development of language, a set of rules for relating a number of sounds into messages which the listener can interpret without ambiguity. The English language like many others achieves this if used carefully.

Speech involves the production of sound waves. Consequently it cannot be conveyed in an acoustical mode over quite moderate distances, like two hundred meters, without disturbing others and losing privacy. Over larger distances, the human voice becomes inadequate while acoustical amplification of the speech will generally be unacceptable in modern society. We don't appreciate high level noise, and that is what other peoples amplified conversation is. As a result, to communicate over long distances we must resort to electrical techniques. Acoustical-electrical and electrical-acoustical transducers are used. The former transforms the speech into an electrical format while the latter is used by the recipient at the distance point to reconvert

the electrical signal back into its acoustic form. Over long distances the electrical signal representing the speech will have to be repeatedly amplified. These amplifications will introduce noise, and the communication channel, be it line or radio link, will introduce a number of different forms of distortion. To reduce these distortions digital communications have been used. Here the electrical signal at the output of the transducer (microphone) is encoded into a digital form prior to transmission. Digital repeaters are placed in the transmission channel, and with careful design the digital signal emerging at the end of the channel is nearly identical to the one which entered. The received digital signal is decoded back to an analogue one which is analogous to the original sound pressure of the speech at the transmitting end of the channel, and it is then passed through the output transducer (the loudspeaker) to give the recovered speech. The quality of the speech is generally only degraded by the noise generated in the encoding process, which can be kept small.

In this chapter we briefly consider the answers to the question "why digitally encode speech signals?" and we proceed with the motivation for the research work described in this thesis. The chapter ends by illustrating the organization of the remainder of the thesis and the contributions which we believe are original.

1.2 DIGITAL SPEECH COMMUNICATIONS.

Digital coding of speech was proposed more than three decades ago, but its realization and the exploitation for the benefit of society took place only after the beginning of the transistor era.

Since then, numerous digital facilities have been introduced into the telecommunication networks. In recent years the telephone industries around the world have made huge investments in digital transmission systems for Junction communications and more can be expected when the local subscriber networks are digitized. Military and Law enforcement organizations have employed digital techniques in their communication systems and many of their existing analogue systems will probably be replaced by digital ones, in the future.

We pause at this point to answer, in an itemised format, the pertinent question: "why bother to digitize speech signals?"[1-5]

1) Digital encoding enables transmission of information over long distances to be achieved without degradation of the speech quality. This occurs because digital signals are regenerated i.e. retimed and reshaped, at repeaters placed along the transmission path and at the terminal station. The transmission quality therefore is almost independent of distance and network topology.

2) Digital processing allows the principle of time division multiplexing (TDM) to be applied in a very simple and economic way to telephone transmission lines and switching devices. In comparison with the frequency division multiplexing (FDM) technique in analogue transmission systems, where complex filters are required, the multiplexing function in TDM is accomplished with economic digital circuitry. Furthermore, switching of digital information is easily done with digital building blocks leading to all-electronic exchanges which eliminate the problems of analogue cross-talk and mechanical switching.

3) When multiplexed, digital signals increase the channel capacity in certain existing media. For example, on inter-exchange junction circuits cable pairs originally intended for single telephone channels can carry 30 telephone conversations in digital coded format.

4) Different transmission media and switching equipment are easily interconnected by means of relatively cheap interface equipment with little or no signal impairments.

5) Different types of signals encoded to a uniform digital format, can be transmitted over the same communication system. Consequently, speech signals can be handled together with other signals such as video, computer data, facsimile data, news dispatches, etc.

6) Digital speech signals are suitable for processing by digital computers and thus complex signal processing, not easily accomplished otherwise, can be achieved. Information in a digital format can be encripted and hence secrecy, especially important in military communications, is obtained.

7) In digital systems the required transmitter power is much less than that of analogue ones and the transmission reliability is much higher. These factors make the digital techniques more suitable for satellite and computer-controlled communications.

8) In extremely difficult transmission paths where the noise exceeds the signal level, digital systems can still extract the information by introducing high redundancy into the transmitted codes.

The information can also be extracted from the noise-corrupted signal by means of adaptive digital processing methods based on the statistics of the signal's source.[6]

9) Large Scale Integration techniques (LSI) employed in the realization of digital circuits can result in cheap and very compact equipment.

10) Digitization of speech offers the possibility of voice communication with computers. Recently much of the research effort is directed in two important areas of speech processing, namely recognition and synthesis. Computer recognition of digitized speech commands would enable the user to interact with the computer via a speech digitization terminal. Also the computer following speech synthesis procedures, would be able to generate digital speech data which would be retrieved to the user via the same terminal.

All the above ten points recommend digitization of speech and provide the motives for studying new speech digitization techniques. Two goals have to be achieved when designing a digital coding method. An efficient digitizer should possess: firstly data rate compression characteristics resulting in smaller transmission bandwidth requirements while maintaining the quality of the digitized speech. Secondly, low implementation cost, although this can on occasions be warned, for example, in some types of military communication systems. In general these two requirements oppose each other. That is, large bit rate compression and good quality speech is usually achieved by highly complex and costly digitizers. When the bandwidth allocated for digital speech transmission is fixed, the challenge always exists

for producing improved perceptual quality for less cost, i.e. efficient speech digitizers.

There is another long-term motive for studying improved speech digitization algorithms. Voice is a compressible source as indicated from the following two facts: i) high quality speech can be transmitted in digital format at a rate of 64 kbits/sec. ii) intelligible speech can also be transmitted with only 1000 bits/sec. Consequently, digital speech can be thought as a highly variable rate source and this could be used to increase the flexibility of a communications network under fluctuating traffic conditions. That is, when the incoming digital speech data begins to congest the network, the transmission bit rate from the various speech sources could be reduced while retaining speech intelligibility. This suggests that Programmable Real Time Signal Processor (PRTSP) terminals could be used to implement a variety of speech digitization algorithms. When the user wants high quality speech, digitization is performed by the proper high bit rate algorithm while if the network is too full a busy signal is returned as an indication for the user to lower his demand and employ a different speech algorithm with compressed transmission bandwidth characteristics. The goals to be achieved by a speech algorithm employed in a PRTSP terminal are the same with those previously discussed, with the only exception of having the implementation cost of the PRTSP terminal fixed.

There have been two main trends in digitizing speech algorithms (both are discussed in Chapter II) i) Modeling of the human vocal apparatus where an Analysis procedure estimates the model parameters. These parameters constitute the speech digitized data and are

transmitted. ii) Direct digital translation of the speech waveform.

Digitization algorithms of the first category are rather complex but offer large bit-rate-compression. Their transmission bit rate is of the order of 1000 to 8000 bits/sec. Bit rates higher than 8 kbits/sec. are usually produced by algorithms of the second category. These direct waveform encoding techniques are of great importance in digital speech communications because of their simplicity and little cost when compared to the Modeling techniques, and because of the high quality reproduced speech (at output bit rates above 20 kbits/sec.).

The research work presented in this thesis is focused on waveform encoding techniques. In particular we investigate new methods for differentially encoding speech signals. The proposed encoding algorithms are relatively simple and efficient in maintaining the quality of the speech and show good bit-rate compression characteristics.

1.3 ORGANIZATION OF THESIS.

We outline briefly each of the following chapters in this thesis.

Chapter II is a review chapter of digital coding techniques applied to speech signals. The reason to include this chapter is two-fold:

i) To acquaint the non-specialized reader with the existing speech digitization techniques, and to compare them.

ii) To provide all the necessary background knowledge and establish the framework for the investigations which follow.

The survey begins with a brief presentation of the basic "Modeling" or as they are better known, "Analysis-Synthesis" techniques. In this section we include the fundamental characteristics of speech

production and perception which are important in the development and understanding of Analysis-Synthesis techniques and which are quite useful in producing efficient waveform encoding algorithms. We then proceed by examining in depth Waveform Coding techniques. Special emphasis is given to analysing, comparing and assessing the performance of Differentially encoding systems such as Differential Pulse Code Modulation (DPCM) and Delta Modulation (DM). The essential element of all digitization algorithms namely: the Quantizer, is also discussed in details.

Chapter III describes the hardware and software development of a minicomputer based speech processing system which enables the storage of several minutes of speech material on digital magnetic tape. The speech is then processed by encoding algorithms derived in the computer, and the resulting digital data is converted into analogue form for subjective evaluation. The description of the system includes the basic computer controlled input-output hardware and software functions. It is written with the purpose of serving as a reference guide for future system users. Readers may omit this chapter without losing the continuity of the thesis.

In the early stages of the research we concentrated on the various possibilities for improving the performance of DPCM encoders. In Chapter IV we examine, through computer simulations, the effect of Delayed Encoding when applied to DPCM. In particular, while trying to keep the complexity of the resulting systems small, we introduce and examine DPCM systems employing simple Delayed Encoding algorithms. Computer simulation results obtained from these systems when speech is used as the input signal, are presented.

In Chapter V we take a closer look, through computer simulations, of a typical Adaptive DPCM system employing Jayant's adaptive quantizer and adaptive predictors. Then in order to obtain a digitization system with superior performance we introduce the concept of pitch synchron ous differential processing of speech signals. Two novel systems are described, the Pitch Synchron ous First Order DPCM (PSFOD) and the Pitch Synchron ous Differential Predictive Coder (PSDPE). Both of the systems exploit the waveform similarity between successive pitch periods of voiced speech, as well as the correlation between successive input samples. At the end of this chapter the importance of the quantization element in Differentially encoding systems is discussed. We conclude that the performance of Differential encoders, and further, the estimation efficiency of the predictors used by them, depends upon the performance of the quantizer. This leads our investigations into techniques of efficient quantization.

Chapter VI begins by discussing existing adaptive quantization schemes and generalizing their adaptation approach. Then a novel instantaneously adaptive non-linear ratio quantizer called the Dynamic Ratio Quantizer (DRQ) is proposed. A detailed mathematical analysis of the basic DRQ scheme is presented. An improved version of the DRQ called the Envelope - DRQ is then described. The performance of the DRQ systems is illustrated by means of computer simulations and signal-to-noise ratio (snr) results for First Order Markov process and speech input signals. The snr results are compared with our informal subjective listening experiments. The chapter ends by describing the simplicity of implementation of the DRQ quantizer.

Finally, in Chapter VII the main results reported in the Thesis are analysed and criticized. Some suggestions are also made relating to further work.

The over-all arrangement of the Thesis is illustrated in Figure 1.1.

1.4 SUMMARY OF MAIN RESULTS.

The main results presented in this Thesis are outlined as follows: First in Chapter IV we show that by incorporating simple Delayed encoding algorithms into DPCM encoders, an increase of only 1 dB in peak signal-to-noise ratio and a small increase of dynamic range is obtained. Consequently, unless the Delayed algorithm is very complex the snr advantage of such a system compared to DPCM is rather limited.

In Chapter V we introduce and develop the Pitch Synchronous First Order DPCM (PSFOD) and Pitch Synchronous Differential Predictive Coder (PSDPE) systems. Both of them show modest complexity and excellent encoding performance when compared with DPCM. The computer simulation results show an snr advantage of 6 dB's for the PSFOD and 8 dB's for the PSDPE systems (3 bits/sample quantization) over the First Order DPCM and Adaptive DPCM respectively.

Because we realized that the main limitation in the performance of Differentially Encoding Systems is their quantizers, we introduced the DRQ quantization technique. By utilizing non-linear elements, a fixed quantizer and simple prediction, a closed-loop adaptive quantizer emerged having a high constant snr over a wide dynamic range. The DRQ computer simulation shown an improvement compared

to Bell Laboratories One word memory APCM system in both snr and subjective experiments. The Envelope-DRQ scheme operating at transmission bit rates as low as 10 to 15 kbits/sec. has a subjective performance similar to that of Adaptive-DM.

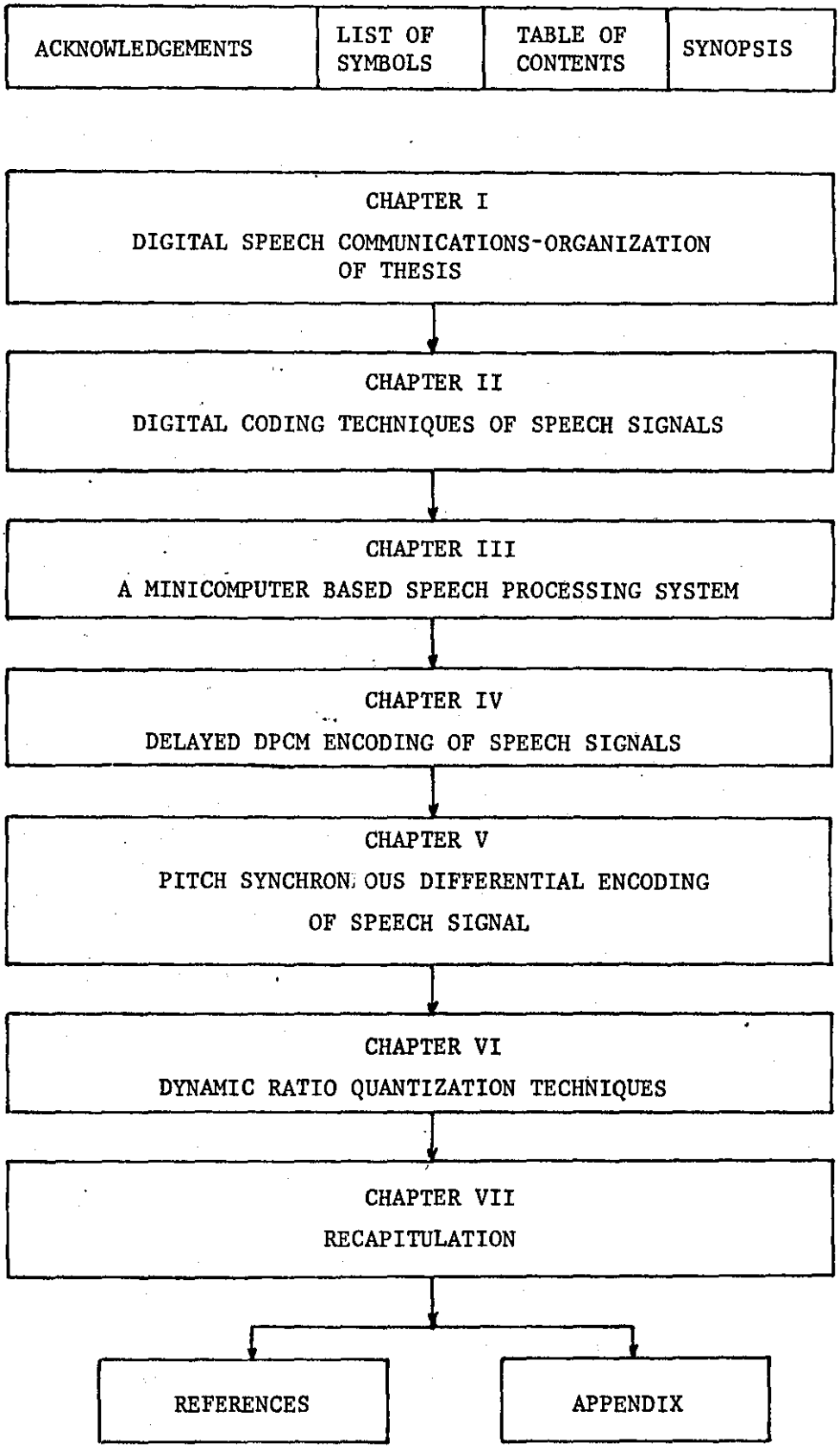


FIGURE 1.1 - Thesis Lay-out.

CHAPTER II

DIGITAL CODING TECHNIQUES OF SPEECH SIGNALS

2.1 INTRODUCTION.

Digital coding of speech signals can be broadly classified into two categories, namely: Synthesis-Analysis (vocoder) coding and waveform coding. The concepts used in these two methods are very different.

In the Synthesis-Analysis systems (described in detail in Section 2.2), a theoretical model of the speech production mechanism is considered and its parameters are derived from the actual speech signal. These parameters are digitally encoded and transmitted. At the receiver they are decoded and used to control a speech synthesizer which corresponds to the model used in the analyser. Provided that the perceptually significant parameters of the speech are extracted and transmitted, the synthesized signal perceived by the human ear approximately resembles the original speech signal. Thus during the Analysis procedure the speech is reduced to its essential features and all the redundant constituents which do not effect human perception are removed. Consequently a great saving in transmission bandwidth is achieved. On the other hand the synthesis, analysis processing operations are complex, resulting in expensive equipment.

In waveform encoding systems, an attempt is made to preserve the waveform of the original speech signal. In such a coding system the speech waveform is sampled and each sample is encoded and transmitted. At the receiver the speech signal is reproduced from

the decoded samples. The way in which the input samples are encoded at the transmitter may depend upon the previous samples or parameters derived from the previous samples, so that advantage can be taken of the speech waveform characteristics. Waveform coding systems tends to be much more simple and therefore inexpensive compared to the Vocoder type systems. Because of this, they are of considerable interest and importance and their applications varies from mobile radio and scatter links to commercial wire circuits.

Although the emphasis in this chapter, from section 2.3 onwards, is given to the coding systems of the latter category, the better known Analysis-Synthesis coding systems are also discussed to present a complete review of digital coding techniques applied to speech signals.

2.2 ANALYSIS-SYNTHESIS CODING TECHNIQUES (VOCODERS).

The main task in the design of a vocoder system is to determine the basic characteristics of speech production and perception and to incorporate these into the system. Ideally the characteristics are described in terms of few independent parameters which can serve as the information-bearing signals.

Basically the vocoding procedure can be divided into two parts, namely: analysis and synthesis. The analysis process is carried out at the transmitting end where quantities describing the vocal excitation and the vocal transmission parameters are extracted from the speech signal.

The receiver using this information attempts to synthesize a signal that sounds like the original speech. The idea is schematized

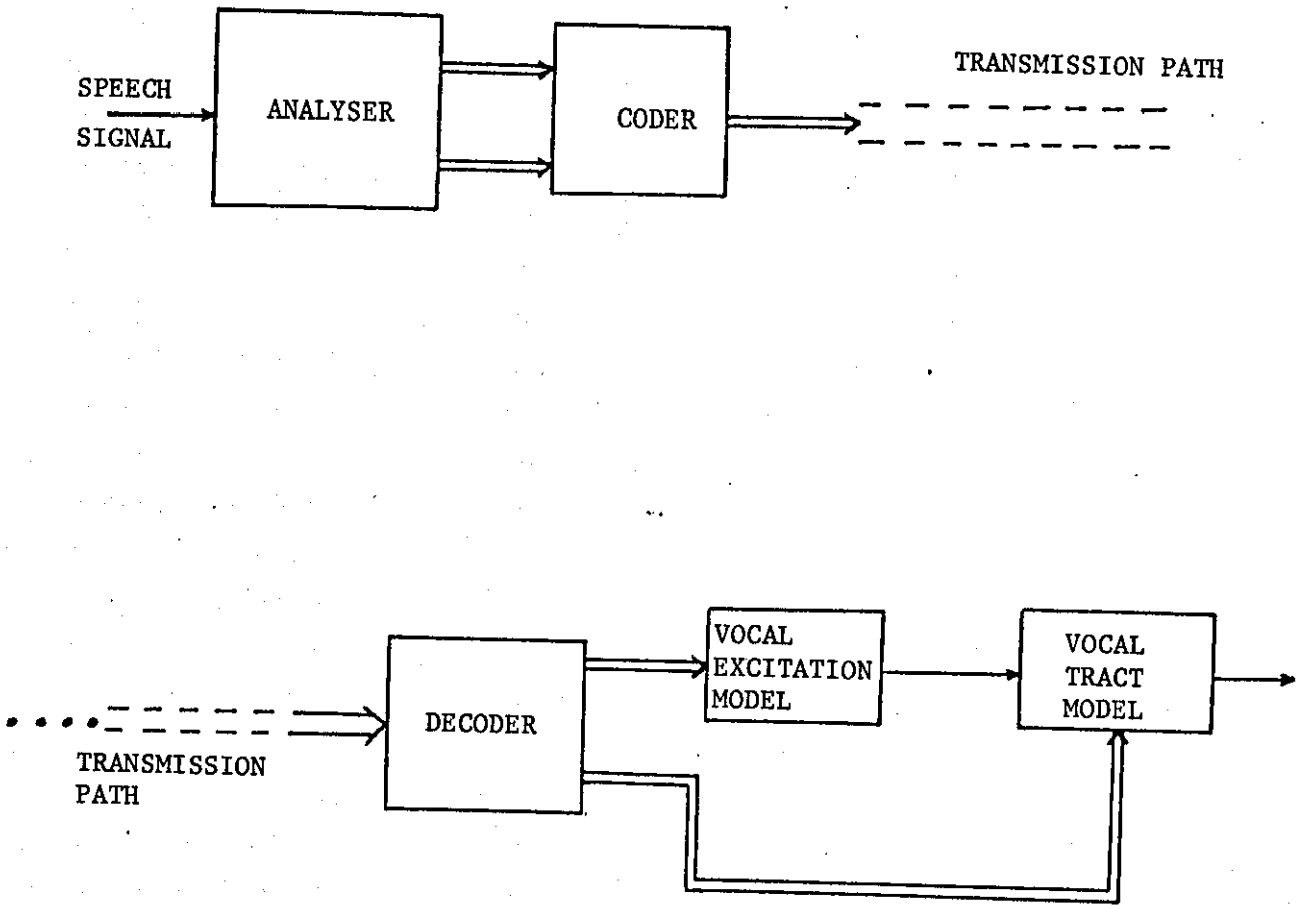


FIGURE 2.1 - Generalized Block Diagram of an Analysis-Synthesis System.

in Figure 2.1. In an ideal system both analysis and synthesis procedures will be accurate models of the speech production mechanism. It is worthwhile therefore to discuss briefly the subject of speech production and perception, before considering the various vocoding systems.

In articulatory terms the speech sounds are produced by exciting the vocal tract. The vocal tract is an acoustical tube which for an average man is approximately 17 cm long. It is terminated by the lips at one end and by the vocal cords constriction at the top of the trachea at the other end. The frequency response of such an acoustical tube shows resonant peaks (called the formants) corresponding to different multiples of the acoustic quarter wavelength. Assuming that the tube is 17.4 cm long and its diameter is constant across its length, then the resonant energy peaks will have frequencies of F_1 = 500 Hz, F_2 = 1500 Hz, F_3 = 2500 Hz, etc. The cross-sectional area of the vocal tract is controlled however by the articulators, i.e. the lips, jaw, tongue, and velum, and it may vary from zero to 20 cm^2. Consequently, the resonances are not fixed at 1000 Hz. intervals but can sweep higher or lower according to the vocal tract's shape. For example, in the sound /ah/ as in "father" the back part of the tongue is pushed towards the wall of the throat and in the front part of the mouth, the opening of the acoustical tube is increased. The effect of changing the shape of the vocal tract in this way is to raise the frequency of the first formant F_1 by several hundred Hz while the frequency of the second formant F_2 is slightly lowered. On the other hand if the tongue is moved forwards, as in the sound /ee/ of "heed", and the size of the tube at the front just behind

the teeth is much smaller than that at the back of the tube, F_1 drops sharply down to as low as 200 or 250 Hz and F_2 increases to as much as 2200 to 2300 Hz.

The vocal tract may also be acoustically coupled with the nasal cavity depending upon the position of the velum. In general, nasal coupling can substantially influence the character of a sound radiated from the mouth.

The source of energy for the speech production lies in the thoracic and abdominal musculatures. Air is drawn into the lungs by enlarging the chest cavity and its pressure is increased by contracting the rib cage. The vocal cords which form a constriction to the air flow are then forced in a oscillation producing quasi-periodic pulses of air and exciting the vocal tract. As the articulators can change the geometry and therefore the acoustical characteristics of the vocal tract, the spectrum of the quasi-periodic excitation is shaped accordingly and the various sounds are produced. (e.g. vowels, nasals, and glides). The rate of the vocal cord vibration, i.e. the rate of the air pulses excitation source is termed as the "pitch" frequency.

Another kind of vocal excitation is created by a turbulant flow of air through constricted spaces in the vocal tract, resulting to "unvoiced" sounds. (e.g. fricatives and plosives).

Although the process of speech production is well understood (see works of Flanagan (7) and Fant (8)), relatively little is known about perception of speech by the human auditory system. Despite the remarkable discovery by Von Bekesy[9] that the cochlea in the inner ear is capable of performing frequency analysis, many questions

remained unanswered. For example how voiced sounds are separated from unvoiced sounds, since the frequency analysis performed by the cochlea is insufficiently sensitive to distinguish between the periodically pitched power spectrum of a voiced waveform and the continuous spectrum of a non-periodic unvoiced noise like signal. Other unexplained phenomena are the binaural hearing (i.e. the ability to accurately locate the positions of a sound source) and the cocktail party effect (i.e. the ability to listen to a particular person in an extremely noisy environment).

At present the only reliable factors that the vocoder designer can rely on are: the preservation of the speech power spectral envelope and the preservation of the voicing information. Then the resynthesized speech will probably sound satisfactory.

Some well known vocoding methods are discussed below.

2.2.1. Channel Vocoders.

The first vocoding system was invented by Dudley[10] and it is known as the Spectrum Channel Vocoder. The system incorporates the two important features of speech production and perception mentioned previously,

i) recognizes that the perception of speech signals depends upon the preservation of the shape of the short-time amplitude spectrum (i.e. preservation of the magnitude of the short-time Fourier Transform disregarding the phase).

ii) recognizes that the vocal tract excitation can be a broad spectrum random signal (unvoiced mode).

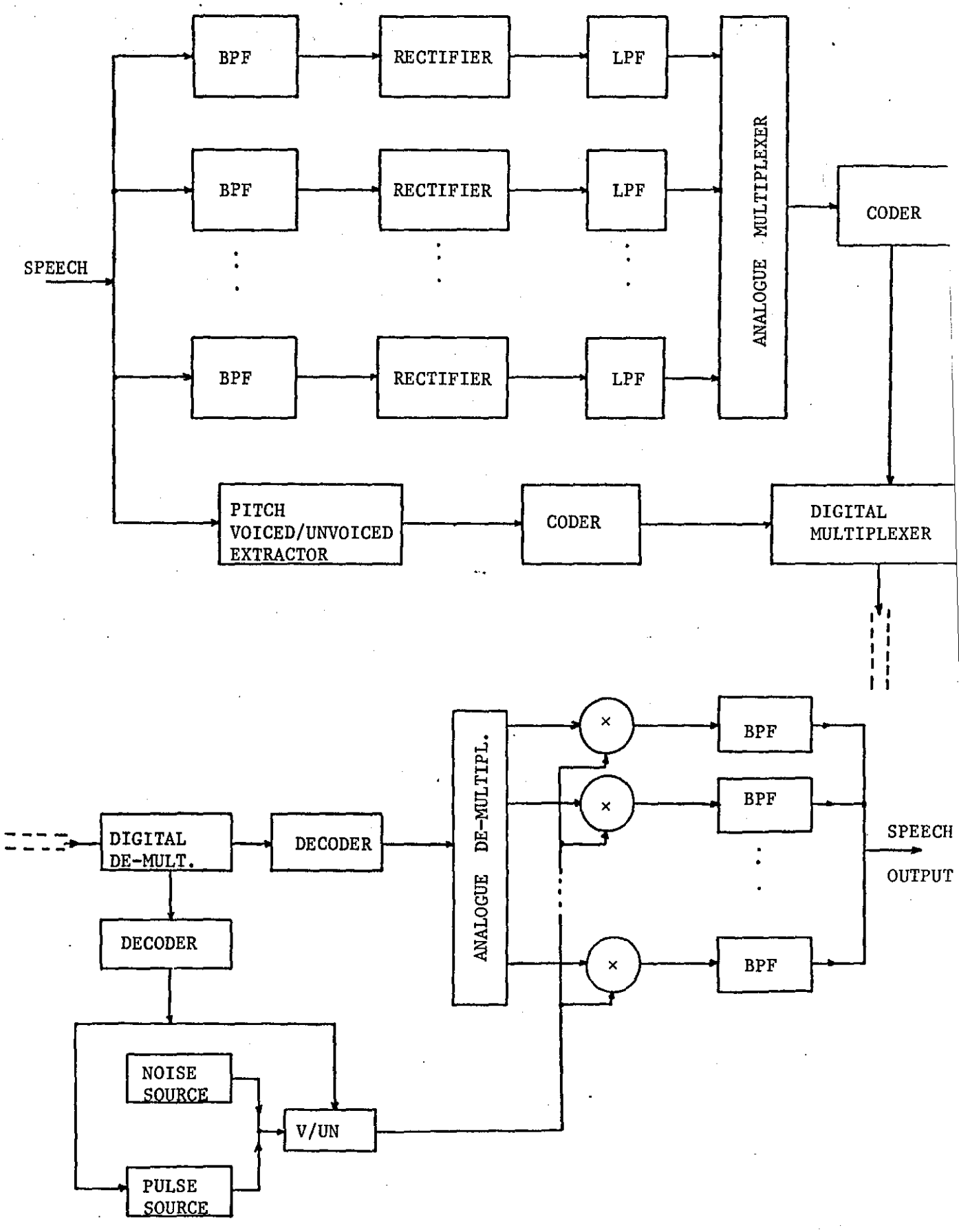


FIGURE 2.2 - A Typical Channel Vocoder.

The synthesizer of a channel vocoder (see Figure 2.2) is represented by a bank of band-pass filters connected in parallel. This arrangement models an estimate of the discrete power spectrum of the speech signal which is to be synthesized. The envelope of the power spectrum is controlled with variable attenuators at the input of each filter. At the transmitting end the input speech signal is analysed by a similar bank of band-pass filters and the measured power in each channel is used to control the inputs to the corresponding synthesizer filters. With regard to the vocal excitation a decision is made by the analyser as to whether the speech is voiced or unvoiced, and if voiced, the pitch period is measured and sent with the voiced/unvoiced information to the receiver end.

The bit rate needed to transmit the channel information depends upon the number of channels, the rate at which they are sampled, and the way in which the signal in each channel is encoded. It may vary over a wide range as can the quality of the resultant speech. In general, an overall transmission bit rate of 2400 to 9600 bits/sec. is adequate for the channel vocoder while the quality of the synthesized speech is monotonically related to the bit rate.

Although the intelligibility of the synthesized speech may be high, there is a perceptible degradation of the speech naturalness and quality. The factors responsible for this are:

i) The discrete representation of the amplitude spectrum is not a particularly efficient method of preserving all the perceptual important spectral details. This lack of high spectral resolutions is imposed by the number, bandwidth, and spacing of the filters.

ii) The large dynamic range of the spectrum may not be covered

due to practical limitations.

iii) The voiced/unvoiced decisions and the accurate pitch extraction is a difficult task and errors can occur. Furthermore, the voiced sounds are synthesized using quasi-periodic pulses whose characteristics can be different from those of the actual glottal pulses.

However, the spectrum channel vocoder can be improved in several ways. The amplitude spectrum can be better measured by careful filter design or by employing digital techniques such as Fast Fourier Transforms. Also, for the important voiced/unvoiced decisions sophisticated techniques can be used such as Cepstrum or Linear prediction so that the pitch period is extracted accurately.

One method to avoid the difficulties of voicing decision and pitch extraction is that employed in the Voiced Excited Channel Vocoder.[11,12] Here a low frequency narrow band section of the original speech is encoded and transmitted, in addition to the vocoder channels. At the receiving end this baseband signal is processed by a non-linear distortion element which flattens and broadens the signal's power spectrum without affecting its periodicity, if any. This flattened and broadened signal is used as the synthesizer's excitation and because it is derived as a subband of the speech signal, it inherently contains the required voicing information. In practical implementations the baseband signal can occupy the range of 250 Hz to 940 Hz while the range from 940 Hz to 3650 Hz is covered by a number of vocoder channels.

The speech quality obtained from such a system is definitely better than that of the spectrum channel vocoder although the transmission bandwidth is increased.

2.2.2. Homomorphic Vocoders.

The term homomorphic processing is generally used in systems in which a complex signal is transformed into a form where the principle of linear filtering can be easily applied. The idea is schemitized in Figure 2.3 where F and F^{-1} are inverse functions and L is a linear time invariant operation. In this system the output of F can be processed in a straightforward manner using linear techniques, while it will be difficult to produce Y(t) by a direct operation (ϕ) on the X(t), input signal.

The homomorphic vocoder[13] shown in Figure 2.4 is based on the observation that the speech waveform X(t) can be modelled as the convolution of the vocal tract impulse response u(t) and the vocal excitation e(t), i.e. $X(t) = u(t) * e(t)$. Consequently these components can be deconvolved in order to obtain two slow time varying (i.e. low transmission bit rate) signals which can then drive the synthesizer at the receiving end.

Specifically during analysis (Figure 2.4a) the input speech signal (X(t) is Hamming windowed (point A) and Discrete Fourier transformed (DFT) so that the signal at point B is the product of the DFT's of u(t) and e(t). Then the log. magnitude is taken resulting in a signal at point C that is the sum of the log. magnitudes of the DFT's of u(t) and e(t). By applying the Inverse Discrete Fourier Transform, a signal $[X(t)]^c$, called the cepstrum is obtained (point D), which is the sum of the cepstra of the excitation and the vocal tract impulse response, i.e.

$$[X(t)]^c = [u(t)]^c + [e(t)]^c.$$

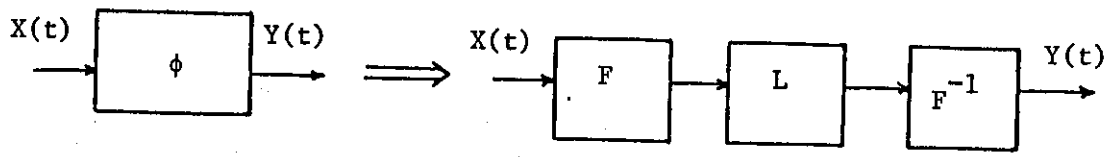
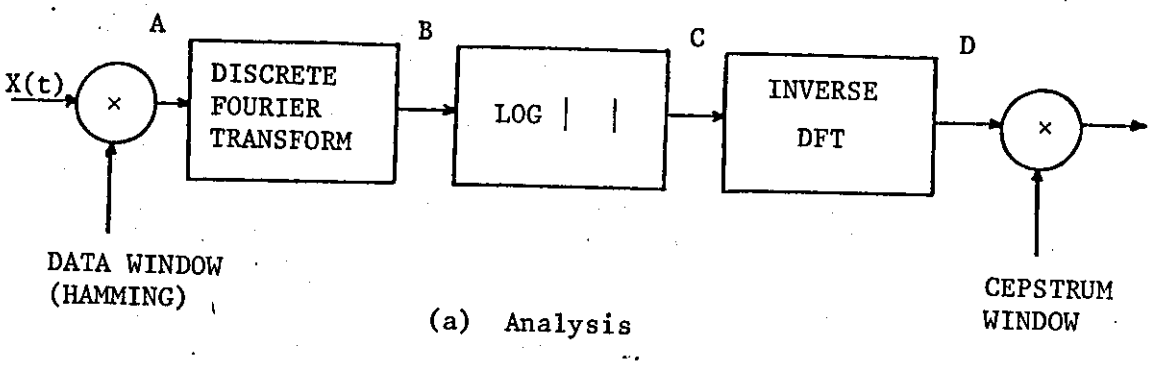
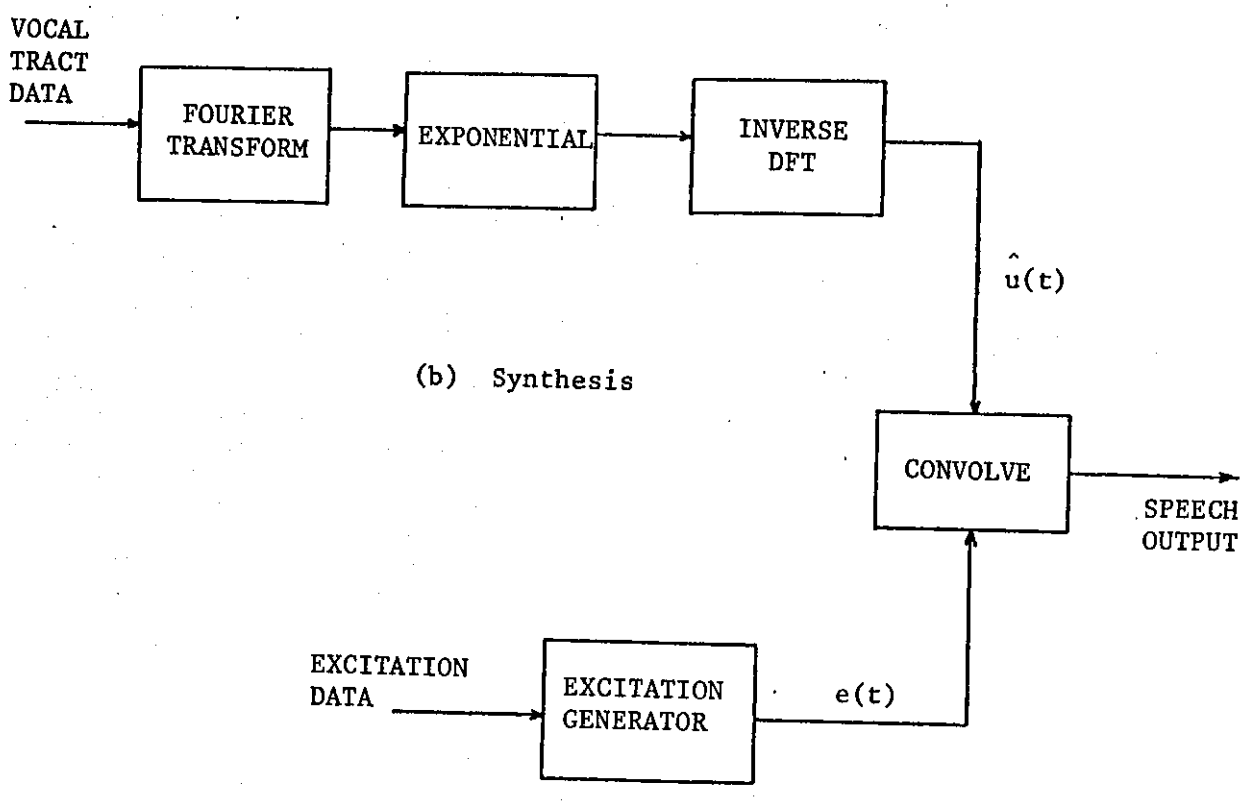


FIGURE 2.3 - A Homomorphic Processing System.



(a) Analysis



(b) Synthesis

FIGURE 2.4 - The Homomorphic Vocoder.

The vocal excitation and impulse response can then easily be separated from the $[X(t)]^c$ time function with a proper time window, and transmitted. This separation is achieved because the cepstrum of the vocal tract impulse response (the impulse response of the vocal tract last for approximately 20 to 30 msec.) becomes a sequence whose duration is much less than the pitch period. On the other hand the effect of the DFT, log. magnitude, and inverse DFT operations on the quasi-periodic vocal excitations component of the speech signal X(t), is to produce a time waveform with pulses spaced apart by the pitch period. Consequently, the initial part of the cepstrum ($L[X(t)]^c$) represents the properties of the vocal tract impulse response, while the subsequent part ($H[X(t)]^c$) provides the excitations information.

The synthesizer after receiving $L[X(t)]^c$ inverses all operations which have been applied on the input signal during the analysis, (i.e. $L[X(t)]^c$ is Fourier Transformed, Exponentiated, and Inverse Fourier Transformed) and an approximation of the vocal tract impulse response $\hat{u}(t)$ is obtained.

Finally, synthesized speech is produced by convolving $\hat{u}(t)$ with the output of an excitation generator controlled by $H[X(t)]^c$.

Good quality natural speech is obtained at the output of the synthesizer when the transmission rate is 7800 bits/sec. By applying predictive encoding to transmit the homomorphic vocoder parameters, the transmission bit rate can be reduced to 4000 bits/sec. with a slight impairment in speech quality.

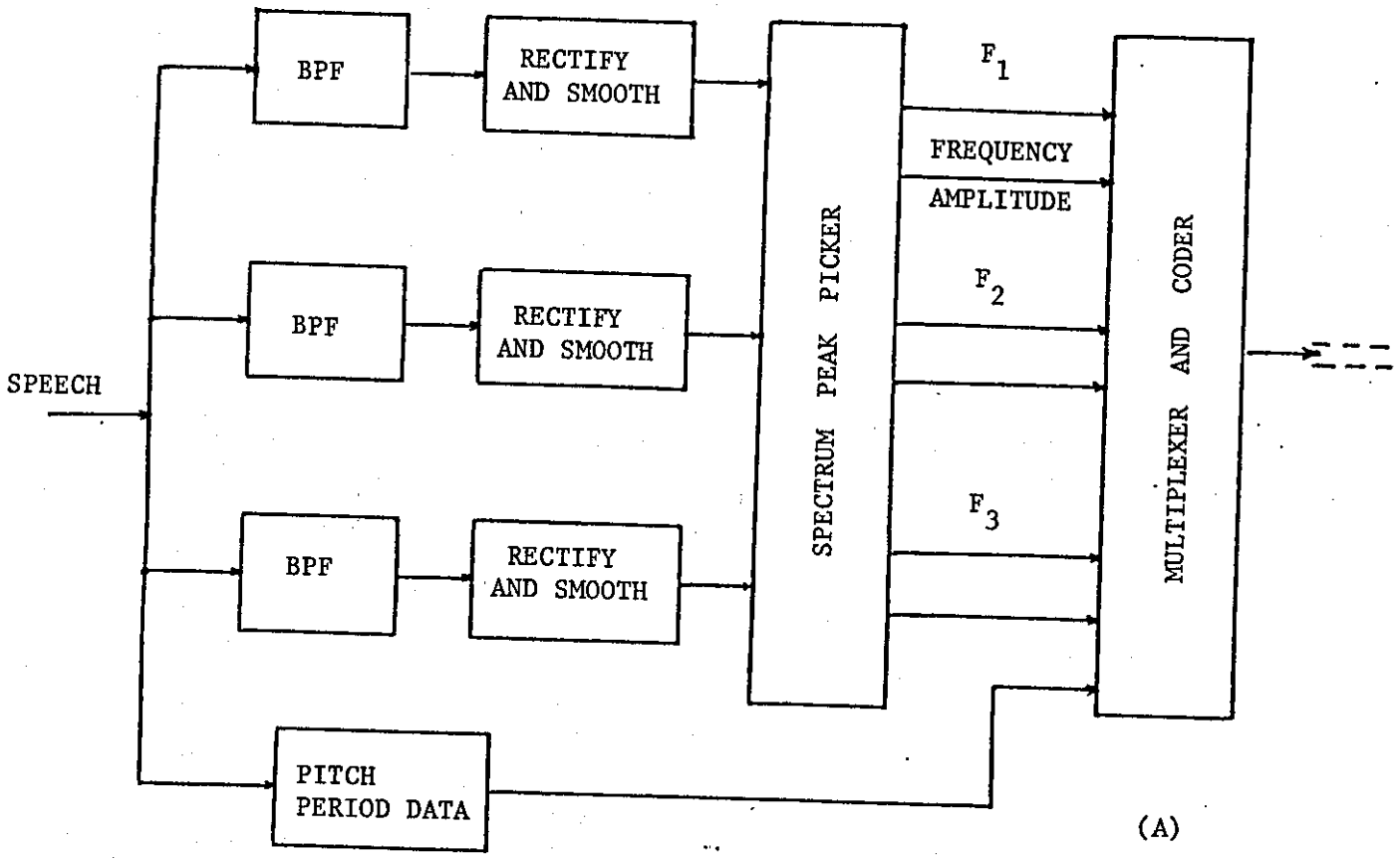
2.2.3. Formant Vocoders.

In the previously mentioned channel vocoders the short term amplitude spectrum of speech is effectively sampled, coded and transmitted to the synthesizer together with the vocal excitation information. However, such detailed representation of the amplitude spectrum is unnecessary as its adjacent values are highly correlated. In addition its shape can be defined by only specifying the frequencies and the spectral amplitudes of the formants. It is possible, therefore to achieve band savings in excess of that obtained in a Channel Vocoder, by transmitting to the synthesizer only the Formant and the vocal excitation data. Vocoding systems which base their operating procedure on the above principle are known as Formant Vocoders.

Generally, the Formant Vocoders are divided into two groups depending upon the synthesizer's structure, i.e. the synthesizer is implemented in a "cascade" or in a "parallel" form.

In the parallel form, Figure 2.5a, the formant characteristics obtained during analysis, are used by the synthesizer to control three variable resonant filters which represents the first three speech formants. Having adjusted the response of the filters according to the formant characteristics, their input is excited by a noise source or a pulse generator, and their outputs are combined to produce unvoiced or voiced speech, respectively.

In the serial form, the transmitted coding parameters are the complex frequencies of the poles and zeros of the vocal tract function (that is an equivalent way of defining the formant frequencies and amplitudes) and the excitation information. The simplified



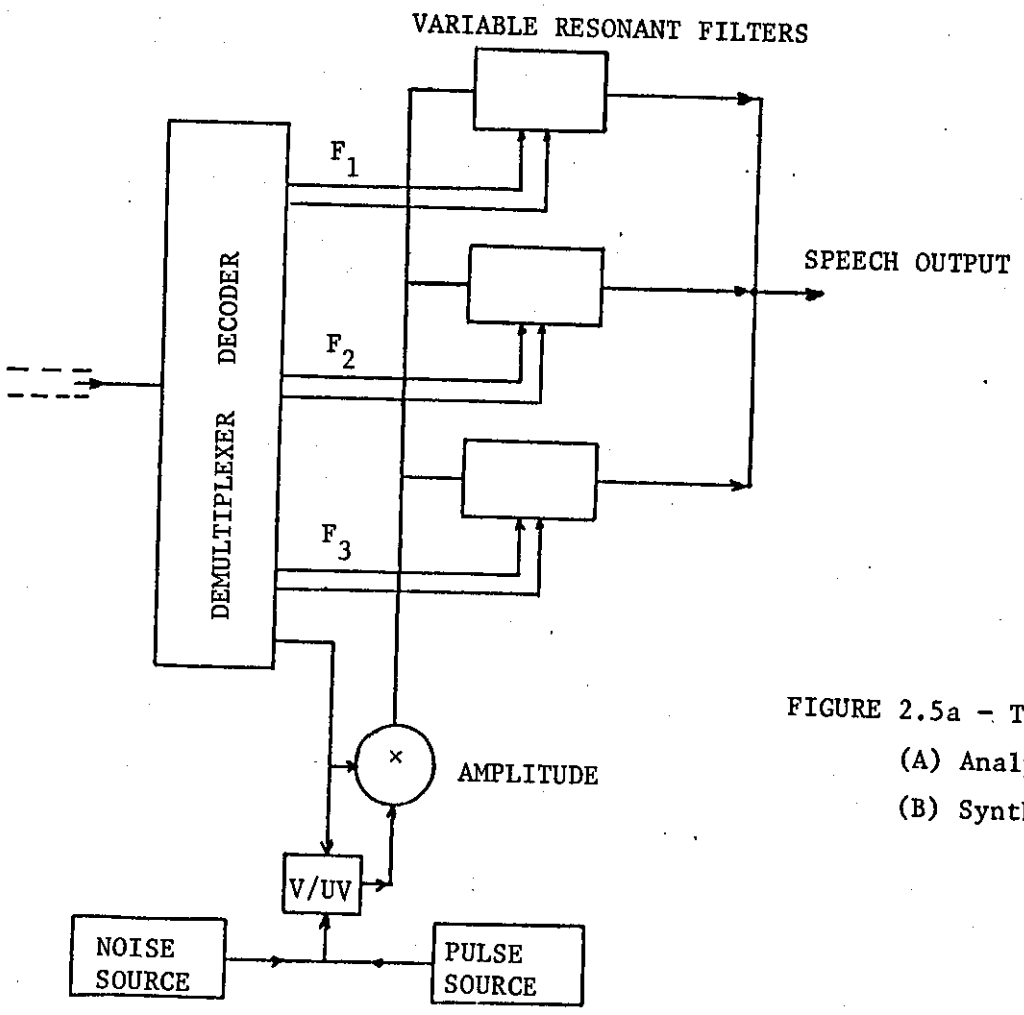


FIGURE 2.5a - The Formant Vocoder.
(A) Analyser.
(B) Synthesizer.

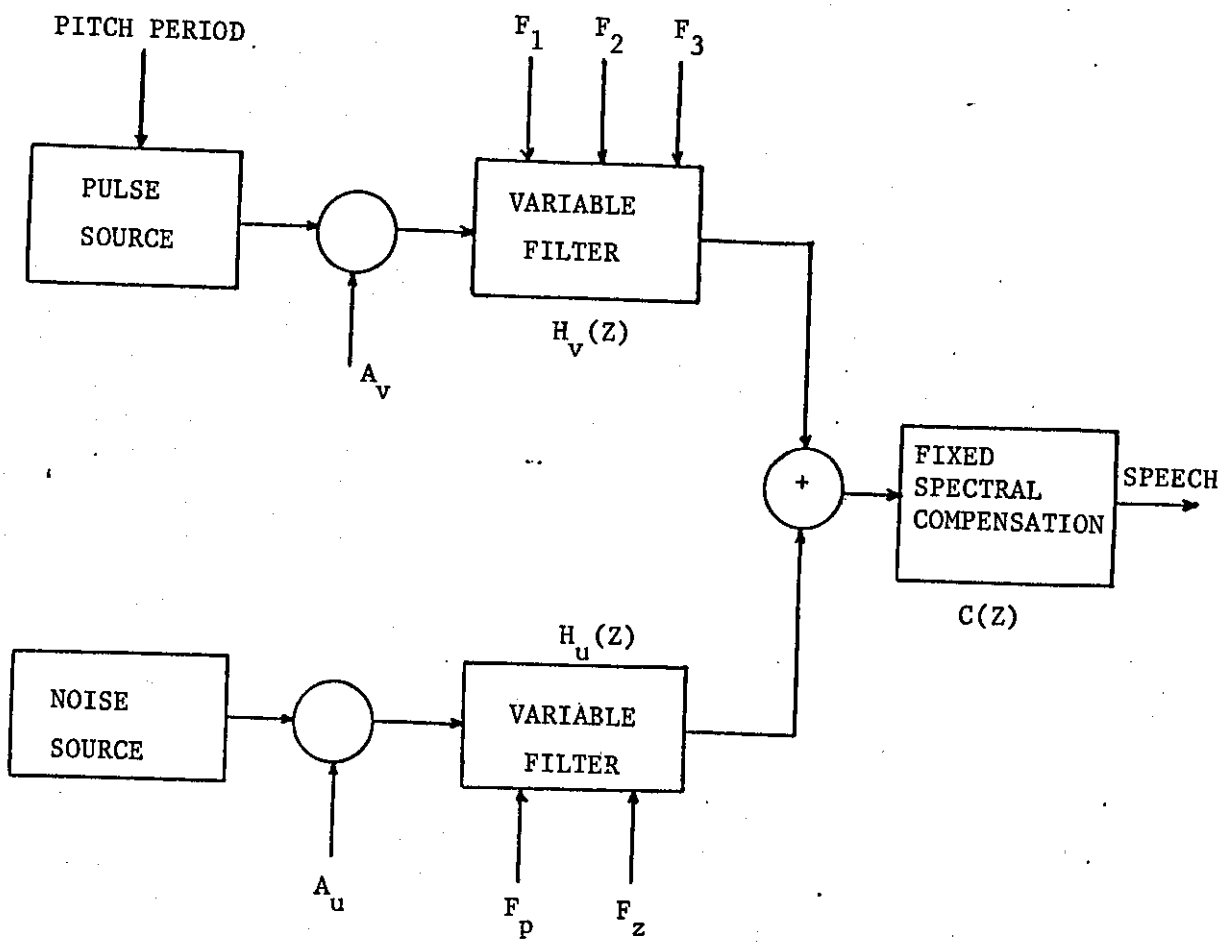


FIGURE 2.5b - Serial Form Forment Synthesizer.

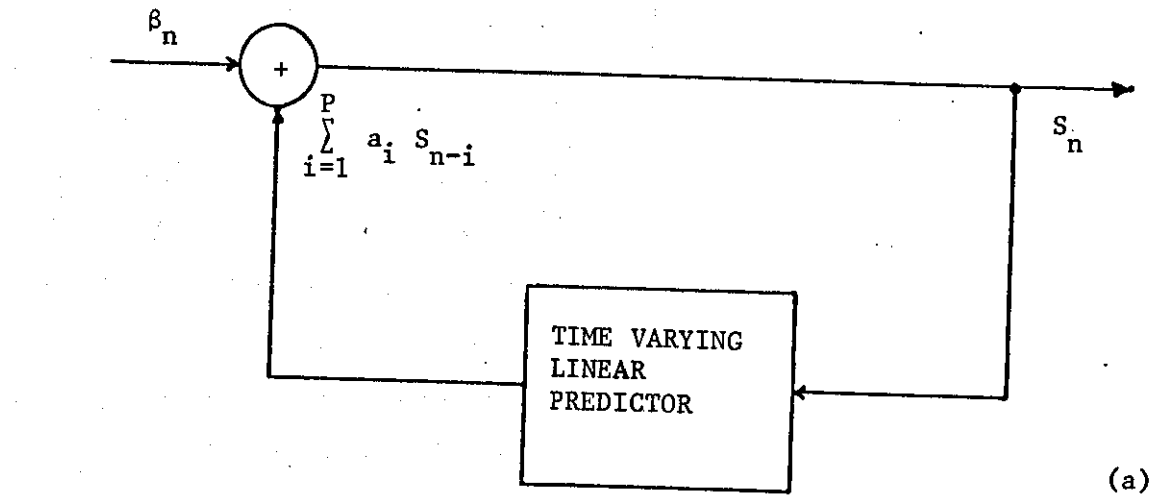
schematic diagram of this synthesizer is shown in Figure 2.5b. In the upper signal processing path the pitch pulses, whose amplitude is controlled by Av, are fed into a L pole time varying digital filter Hv(z). ($L \geq 3$, when $L > 3$ only the first three poles are variable). In the lower path the noise signal, whose amplitude range is controlled by Au, is filtered with a one pole, one zero time varying digital filter Hu(z). The output of Hu(z) presenting unvoiced speech components is added to the voiced components of the output of Hv(z).

The resulting signal is spectrally compensated by a two pole (situated on the real axis) digital filter C(z) which simulates the effect of any vocal tract nasal coupling.

The performance of a formant vocoder depends upon the analysis method used to obtain the formant and voiced/unvoiced information. The most direct method of identifying the formants is to use a large filter bank (as that of the earlier channel vocoder) and pick the frequencies at which the filter output is the highest. Modern formant vocoders tend to employ digital analysis techniques such as Discrete Fourier Transform followed by a peak peaking procedure[14], homomorphic filtering, or inverse linear filtering[15].

2.2.4. Linear Prediction Coding (LPC) Vocoders.

The analysis employed in the Linear Prediction Coding, LPC, vocoder is a time domain technique and avoids the formant location difficulties of the frequency domain formant analysis, where formants seem to disappear during certain sounds or seem to increase their number during others.



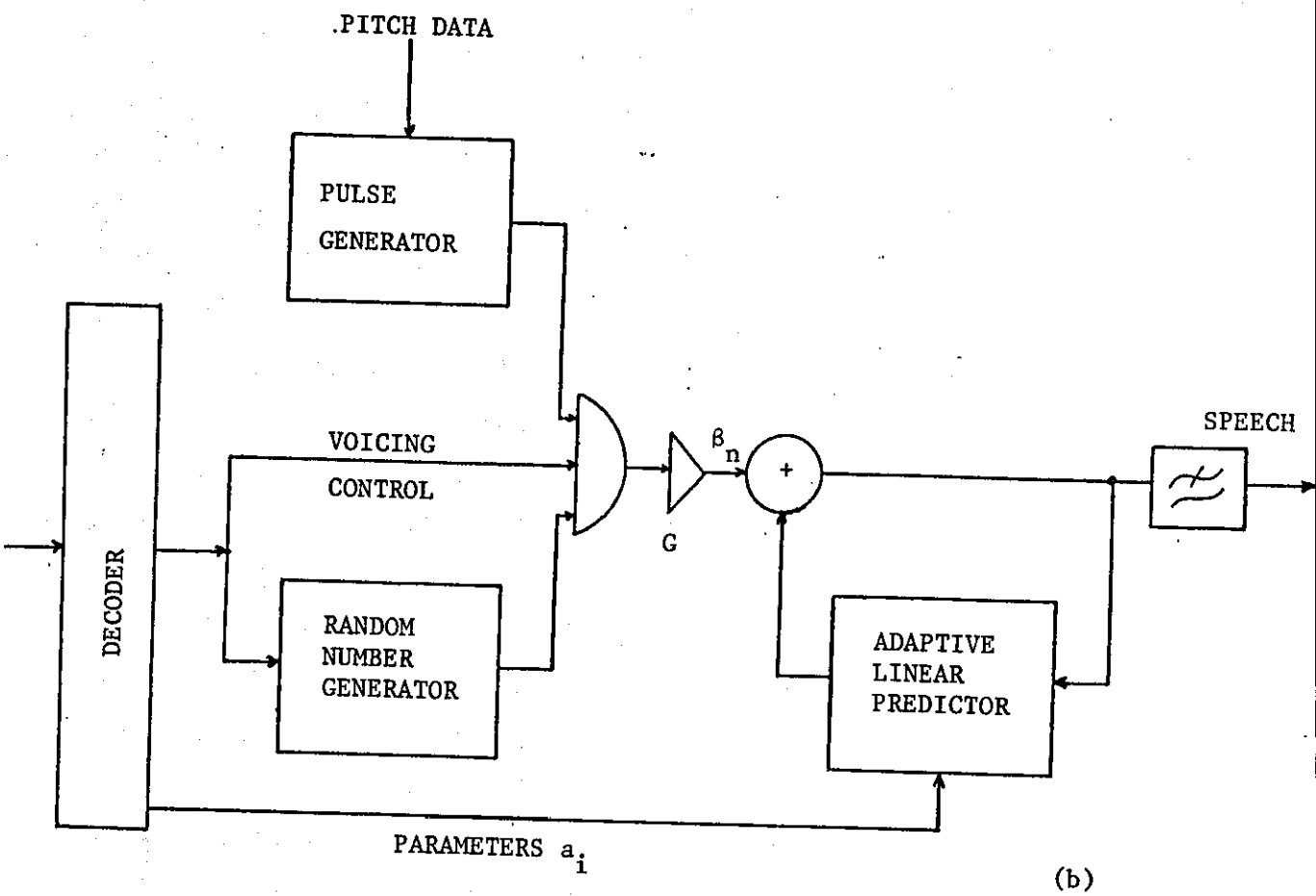


FIGURE 2.6 - (a) Linear Prediction model of Speech Production.
(b) Linear Prediction Synthesizer.

The basic idea in the LPC vocoders is that speech can be produced using an adaptive Pth order Linear digital filter, as shown in Figure 2.6a. This accounts for the vocal tract characteristics, the radiation characteristics and the pulse shape of the vocal excitation. The model proposed by Atal[16] is an all pole approximation of the shape of the original speech spectrum.

The transfer function H(z) of this recursive Pth order digital filter is given by:

$$H(z) = \frac{1}{1 - \sum_{i=1}^{P} a_i z^{-i}} \qquad (2.1.)$$

where P = 2L and L specifies the number of formants needed to characterize the speech amplitude spectrum. The complex roots of the denominator in Equation (2.1.) specifies the formants (and their bandwidths) of the modelled speech spectrum. When a voiced or unvoiced sound is to be produced, the filter H(z) is excited by a quasi-periodic or a random impulse signal, respectively. The difference equation applied to the model is of the form:

$$S_n = \sum_{i=1}^{P} a_i S_{n-i} + \beta_n \qquad (2.2.)$$

where S_n are the speech samples and β_n are the excitation impulses. During voiced speech β_n is zero except for one sample at the beginning of every pitch period. Consequently for all time, except for the start of a pitch period, Equation (2.2.) takes the form of the linear prediction formula:

$$S_n = \sum_{i=1}^{P} a_i S_{n-i} \qquad (2.3.)$$

The analysis procedure involves the determination of the Linear prediction coefficients a_i which, together with the extracted excitation data are transmitted to the synthesizer whose arrangement is shown in Figure 2.6b. The synthesizer's task is to produce a sequence of speech samples $\hat{S}_n$ such that the error e_n between $\hat{S}_n$ and the original speech samples S_n, i.e.

$$\begin{aligned} e_n &= S_n - \hat{S}_n \\ &= S_n - \sum_{i=1}^{P} a_i \hat{S}_{n-i} \end{aligned} \tag{2.4.}$$

is a minimum.

The prediction coefficients can be chosen to minimize the mean square error $E(e_n^2)$ averaged over all n. This is the classical Wiener filtering procedure in parameter estimation theory and $E(e_n^2)$ can be put into the form:

$$E(e_n^2) = E\left[S_n - \sum_{i=1}^{P} a_i \hat{S}_{n-i}\right]^2 \tag{2.5.}$$

To obtain the optimum a_i coefficients, Equation (2.5.) is differentiated with respect to a_j, $j = 1,2,\ldots,P$ and the result is set to zero producing a set of P linear Equations. In matrix notation the Pth order linear Equations system can be written as:

$$\Phi A = \Psi \tag{2.6.}$$

where Φ is the cross covariance matrix whose ϕ_{ij} element is

$$\phi_{ij} = E(\hat{S}_i \hat{S}_j), \text{ depends on } |i-j|$$

and Ψ is the autocovariance vector whose ith element ψ_i is

$$\psi_i = E(\hat{S}_o \hat{S}_i), \text{ depends on i.}$$

Although Φ is a symmetric and positive finite matrix, the optimal solution of Equation (2.6.) with respect to A involves, in implementation terms a rather difficult matrix inversion operation Φ^{-1}. Various methods have been employed to obtain solutions.[17,18] Markel[19] minimized the mean square error $E(e_n^2)$ using the autocorrelation method. This approach to the LPC solution provides

i) a Toeplitz matrix Φ which can be inverted with less computations,

ii) insures stability for infinite word length arithmetic while Atal's method does not always yield a stable synthesizer.

However, the autocorrelation method requires windowing of the input speech data which is unnecessary in the autocovariance method.

Adaptive iterative gradient techniques can also be applied to determine the LPC a_i coefficients. Examples of these techniques[20,21] are the Stochastic Approximation and the simplified Kelman filter sequential algorithms whose a_i solutions are sub-optimal but their implementation is simple.

Another time domain technique of speech Analysis and Synthesis proposed by Itakura and Saito[22] makes use of the Partial Correlation (PARCOR) coefficients. This method differs from the Linear prediction one of Atal and Hanauer in that a Lattice structure predictor is used rather than the canonical form of Equation (2.3.). The predictor's coefficients are optimized sequentially within one sampling period so that the error e_n of Equation (2.4.) is minimum. It has been shown[20] that the Lattice predictor is much less sensitive to parameter variations than the Linear predictive structure. Also in a non-stationary environment, the rate of convergence of the

PARCOR coefficients towards the optimum value is faster than that of the coefficients in the canonical Linear predictors.

The predictive analysis methods discussed so far assume an all pole speech signal. On the other hand it is generally recognized that zeros are included in the speech production (in nasal and unvoiced sounds) and that the H(z) transfer function should contain appropriately placed zeros as well as poles. As these zeros can be assumed to lie within the unit circle of the z plane, it is possible to approximate each zero to any desired accuracy by a set of multiple poles. At the same time it is difficult to access human perception sensitivities to errors in modelling different sounds. Nevertheless the LPC vocoder with all-pole model does produce synthesized speech which has gained a wide acceptance for its perceptual quality.

Scagliola[23] proposed a model, incorporating zeros and poles whose parameters are determined by an interative technique (using gradient optimization). A possible drawback of this system is that, whereas the all-pole model becomes more accurate as the order P of the predictor is increased, there is no systematic rule for defining the number of zeros and poles used in the pole-zero model. Perhaps a more severe restriction of the linear predictive analysis is the lack of a model for the excitation source that is, the use of Equation (2.3.) instead of Equation (2.2.) in the formulation of the LPC solution.

So far, most of the research in LPC has been focused on the modelling of the vocal tract so that the vocal excitation difficulties which are present in the channel vocoder remain with the LPC vocoder.

Specifically, the quality of the synthesized speech is critically dependent on the accurate estimation of the voice-unvoiced parameter and the pitch period. If the analyser incorrectly identifies a voiced sound to be unvoiced and vice-versa, an unpleasant harsh sound and "buzziness" occur in the synthesized speech. On the other hand, errors in the estimation of the correct pitch period of the analysed sound produces an unnatural speech sound. These effects can degrade substanially the quality of the synthesized speech even when the analyser for 95% of the time estimates accurately the excitation parameters. Many algorithms have been developed to determine the pitch period and provide voicing decision[24 to 28], and all of them suffer in one way or another from lack of robustness, i.e. they are sensitive to acoustic background noise, the type of microphone used and speaker variations. However, in spite of these difficulties the LPC vocoder produces good quality speech and usually operates at transmission bit rates between 2.5 and 4 kbits/sec.

A comparison between the basic vocoder techniques would be an appropriate end for this Analysis-Synthesis coding section. Unfortunately as these vocoders are still under development only a few observations will be made:

i) Neither the pitch nor the parameter quantization problem have been extensively examined in the homomorphic vocoder. The rapid development of the Charged-Coupled-Devices, (CCD), and their application in implementing the Discrete Fourier Transform efficiently, could substantially improve this vocoder.

ii) The channel vocoder, according to J.S.R.U. listening

experiments, is considered as good as the LPC vocoder[29]. Others[30] believe that channel vocoders have a slightly greater intelligibility than LPC and that they are more robust under difficult conditions caused by background acoustic noise and channel errors. If both systems are to be implemented digitally, LPC appear at present to be ahead in terms of cost and complexity. This is because the channel vocoder requires 3 to 5 times the computations needed by the LPC system. This cost situation could be altered in future with the development of CCD's techniques, which appear to be applicable to channel vocoders.

2.3. WAVEFORM CODING TECHNIQUES.

In waveform coding the transmitted digital information directly represents the analogue speech waveform, and at the receiver a decoding process attempts to reconstruct the original speech signal as accurately as possible. This is in contrast with the vocoding techniques where the essential characteristics of the excitation and the vocal tract functions are described by a few parameters which are then transmitted to the speech synthesizer at the receiving end.

In nearly all the Waveform coding systems, the analogue speech signal is quantized in both time and amplitude. Quantization in time means that the analogue signal is sampled at certain instants and the transmitted data is related only to these samples. On the other hand amplitude quantization means that the continuous amplitude range of the input samples is replaced by a set of finite number of discrete amplitude levels. This inherently introduces an error

in the amplitude of the samples, known as quantization noise. For clarity and simplicity the terms "sampling" and "quantization" will be used throughout the thesis, corresponding to quantization in time and amplitude respectively.

A generalized block diagram of a Waveform Coding System (or Codec) is illustrated in Figure 2.7. At the transmitter, the band-limited analogue speech signal X(t) is sampled at a rate greater or equal the Nyquist rate (i.e. $2f_{max}$ where f_{max} is the higher frequency present in X(t)) to produce a sequence of samples $\{X_n\}$, $n = 1,2,\ldots,\infty$. The goal of the Encoding technique is to accurately represent the $\{X_n\}$ sequence with a minimum number of bits per sample. The Encoding process must be reversible so that a close approximation $\{\hat{X}_n\}$ of the original sampled speech $\{X_n\}$ can be obtained from the Decoding process.

Consider the operation of the Codec at the Nth sampling instant. The input sample X_n is processed by the encoding algorithm to yield a sample $f(X_n)$ which can be directly related to previous input samples X_{n-i}, $i = 1,2,\ldots,m$, or to parameters derived from the statistical properties of $\{X_n\}$. $f(X_n)$ is then quantized and the resulting discrete amplitude level $\hat{f}(X_n)$ at the output of the quantizer and encoder is converted to a P-bit binary word. The L_n binary word of P bits corresponding to the $\hat{f}(x_n)$ sample is transmitted, and may be corrupted by additive noise, dispersion and non-linearities exsisting in the transmission path. The received L_n' word is binary decoded into a discrete sample $f'(X_n)$ which is used by the decoding algorithm to produce the $\hat{X}_n$ sample. In the absence of binary transmission errors $\hat{X}_n$ is a close approximation of the input speech sample X_n.

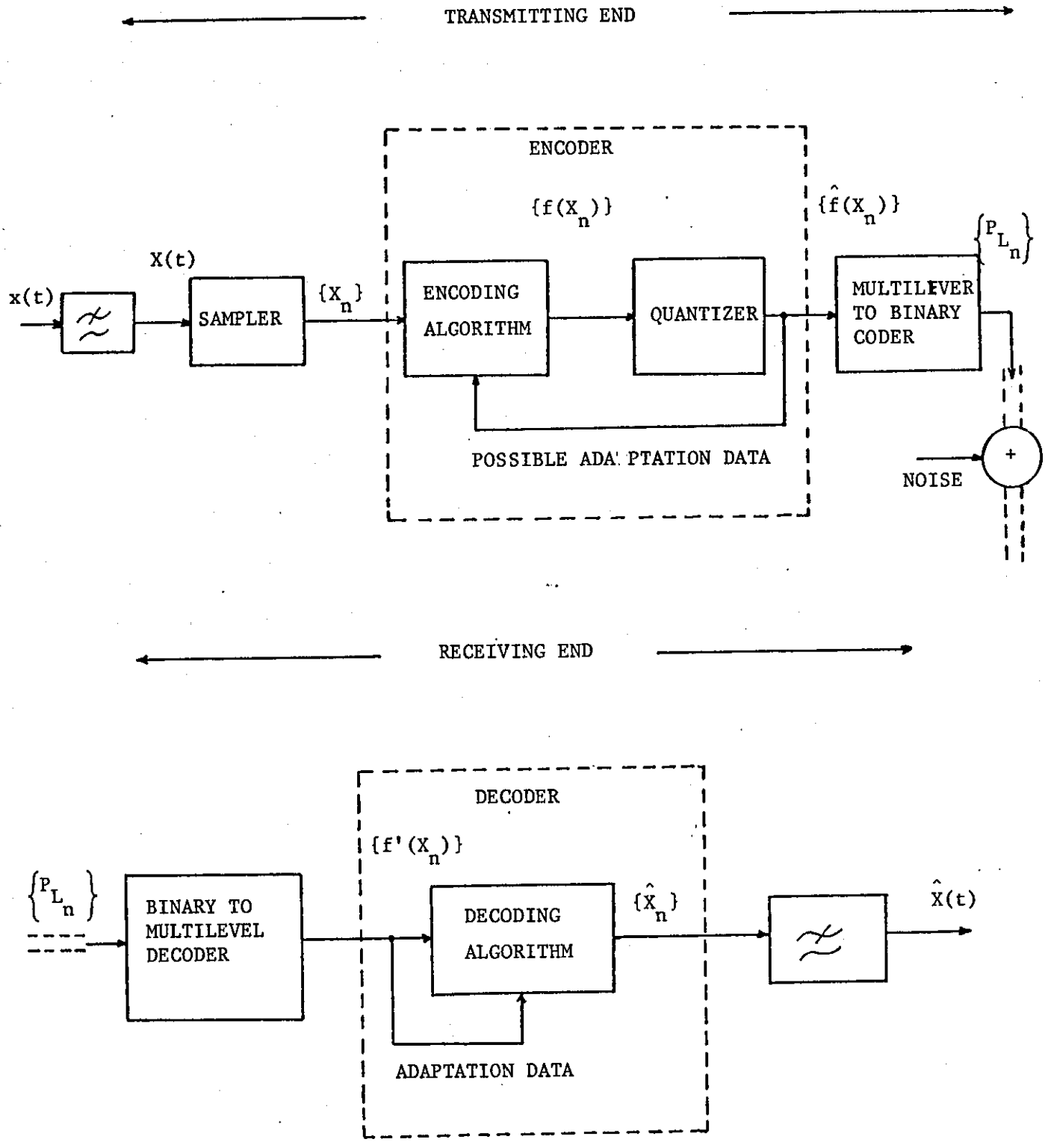


FIGURE 2.7 - Generalized Block Diagram of a Digital Waveform Encoding System.

The above encoding-decoding procedure applied to the input sequence $\{X_n\}$ results in a decoded sequence $\{\hat{X}_n\}$. The final operation in order to recover the analogue approximation $\hat{X}(t)$ of the original speech signal $X(t)$ at the sending-end, is the interpolation of the $\hat{X}_n$ samples by a low-pass filter. Assuming that the distortion in the reconstructed signal $\hat{X}(t)$ due to the channel is negligible, i.e. $L_n = L'_n$ the performance of the system depends upon the encoder's quantization noise. That is to say, for a given number of bits per sample available for transmission, the codec operates efficiently if the quantization noise is a minimum, i.e. the signal-to-noise ratio of the encoding process is a maximum.

Having introduced the basic principles and ideas behind the waveform encoding of speech signals, a fairly broad spectrum of waveform encoders will now be discussed.

2.3.1. Pulse Code Modulation (PCM) Coding.

The significance of Pulse Code Modulation is that, historically, it is the first method (due to Reeves[5]) converting analogue speech signals into a digital form, and that it is still widely used in digital speech transmission systems.

The processes involved in a PCM codec described in great details by Cattermole[1] are as follows:

The input speech signal $X(t)$ is band limited to exclude any frequencies greater than f_{max}. This signal is sampled at a rate W equal or greater than the Nyquist rate $2f_{max}$, so that a perfect reconstruction of the analogue signal $X(t)$ is ensured with an

appropriate filter procedure. The samples so produced are then quantized into the nearest of 2^P levels and a P bit word is assigned to them prior to transmission. The overall transmission rate of the system is 2WP bits/sec. At the receiving end the binary words are decoded back into amplitude levels which are then low-pass filtered (with W as the cut-off frequency) to reproduce the analogue decoded signal $\hat{X}(t)$.

2.3.1.1. Time invariant quantizers.

The quantizer is the element which determines in PCM the accuracy of the approximation of the recovered signal $\hat{X}(t)$ to the input signal X(t), assuming no transmission bit errors. In its simplest form it is called the zero-memory or memoryless quantizer. A zero-memory quantizer accepts analogue samples and imposes amplitude restrictions on them so that each analogue sample is forced, i.e. quantized to the nearest of a finite set of amplitude levels. Consequently the value of the quantized sample is independent of earlier analogue samples applied to it.

A n-level zero-memory quantizer is defined by a set of n-1 decision levels $\xi_1, \xi_2, \ldots, \xi_{n-1}$, and a set of n output levels $x_1, x_2, \ldots, x_n$. When the input sample X lies in the i'th quantization interval, it is quantized to a value x which is contained within the interval

$$\xi_{i-1} < x < \xi_i \quad .$$

The input-output characteristic of a zero-memory quantizer can assume differing symmetries about the zero level as shown in

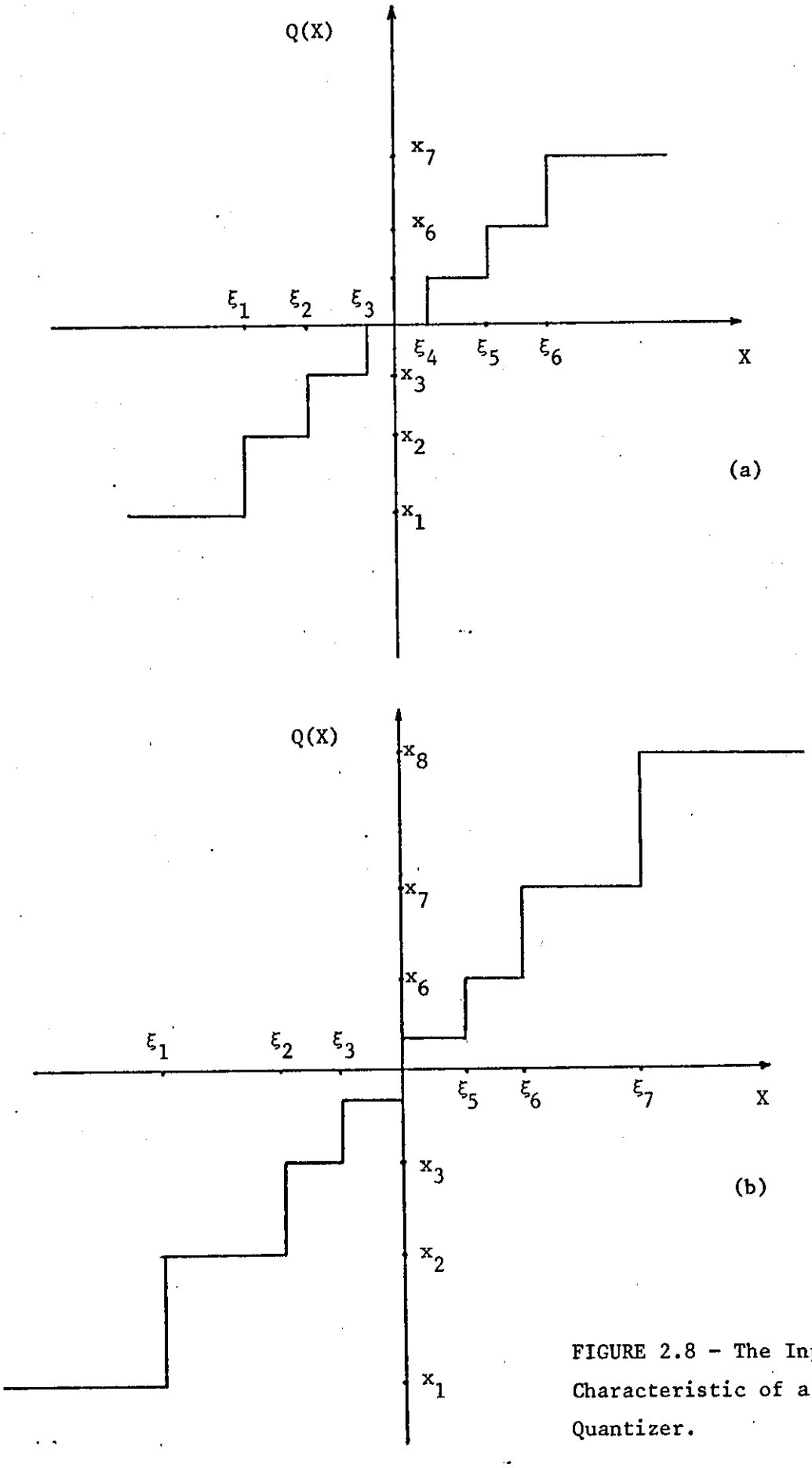


FIGURE 2.8 - The Input-Output Characteristic of a Time-Invariant Quantizer.

Figures 8a and 8b. They can be viewed as a stair-case approximation of x to the value of the input sample X. In the case where X lies within the amplitude range of the quantizer, i.e. $\xi_1 < X < \xi_{n-1}$, the quantization noise introduced is bounded and is sometimes known as granular noise. The noise is unbounded when the input sample lies outside the quantization range and it is described as peak or amplitude clipping noise. Obviously the overall noise is the sum of the peak clipping and granular quantization noise and the trade-off between their relative amounts is controlled by the values chosen for the ξ_1 and ξ_{n-1} decision levels.

For a uniform quantizer (i.e. the spacing δ between the quantization levels is constant) the mean-squared quantization noise is[1]

$$N^2 = \frac{\delta^2}{12} \tag{2.7.}$$

provided the amplitude distribution of the input signal X(t) falls within the range of the quantizer and δ is small compared to the variance of the signal.

The signal-to-noise ratio, snr, is often defined as the ratio of the rms value of the input signal $X(t)_{rms}$ to the rms value of the noise generated by the quantizer. Given that the amplitude range of the quantizer spans a width of eight times $X(t)_{rms}$, (say $\pm 4X(t)_{rms}$ which is a fairly good assumption for a zero mean Gaussian random variable) the step size δ is equal to

$$\delta = \frac{8X(t)_{rms}}{2^P} \tag{2.8.}$$

From Equations (2.7.) and (2.8.) the value of snr in dB is

$$\text{snr (dB)} = 10 \log_{10} \text{snr} = 6P - 7.2 \tag{2.9.}$$

Equation (2.9.) shows that the snr of a 2^P levels quantizer increases linearly with the number of bits P each quantized sample is coded. However the bandwidth of the transmitted bit stream also increases proportionally with P.

2.3.1.1a Optimum Quantizers.

In order to obtain a higher snr for a given number of bits per sample, the positioning of the levels of the quantizer have to be adjusted with respect to the probability density function (pdf) of the input signal. This is because in speech and in several other signals the occurrence of small amplitudes is more likely than large amplitudes. Consequently the optimum quantizer has non-uniform spacing of its quantization levels. As the probability of the input samples falling into the various intervals varies, so does their noise contribution. The non-uniform spacing of the quantization levels is equivalent to the scheme of a zero-memory nonlinearity K(X), called the compressor, followed by a uniform quantizer. The nonlinearity K(X) compresses the input samples in a manner dependent on their statistical properties. The compressed samples are then uniformly quantized. The approximation of the signal applied at the input of the compressor is obtained at the receiver by expanding the recovered samples with the inverse nonlinearity $K^{-1}(X)$. This nonlinear operation K(X) is monotonic and no signal distortion is introduced by the compression-expansion process. The overall scheme is known as companding.

Naturally, the question arises of how to select the best quantization characteristic for an input signal with a specific pdf.

This problem can be solved with two different approaches. The first[31 to 34] assumes a large number of quantization levels and leads to explicit solutions, the second[35,36] is a numerical procedure which makes no assumptions.

Panter and Dite[31] examined non-uniform quantization with the quantizing scale adapted to the pdf of the input signal and the mean square quantization error $\sigma_n^2 = E\left\{(X-x)^2\right\}$ kept to a minimum value. Their analysis is based on the assumption that the quantization is sufficiently fine and that the amplitude probability density function of the input signal is constant within the quantization intervals. Published results[31] show a significant improvement in snr over uniform quantizing when the ratio of the signal's peak-to-rms value is larger than four.

Smith[32] using the same assumptions derived the exponential companding law K(X) which produces a minimum error σ_n^2 for speech-line type signals having a Laplacian pdf.

In a theory of optimum quantization, Max[35] showed how to optimally choose the thresholds and quantization levels of a quantizer. In his analysis a priory knowledge of the pdf and the variance σ_x^2 of the input signal is required, and no assumption of fine quantization is made. His results include uniform and non-uniform optimum positioning of the quantizing levels, when the input signal is a zero mean, unit variance Gaussian random variable. Paez and Glisson[36] utilizing Max's technique derived the parameter's of uniform and non-uniform quantizers for signals with Laplacian and speech-like Gamma distribution.

For all three distributions the quantization noise $NU(\sigma_n^2)$ of

the non-uniform quantizer is clearly smaller than the noise $U(\sigma_n^2)$ of the uniform quantizer, when the number of quantization levels is large. In the case where the number of quantization levels is small and the input signal is Gaussian distributed[35], it seems that it is hardly worthwhile using non-uniform scaling as $NU(\sigma_n^2)$ and $U(\sigma_n^2)$ are remarkably similar. However, as the probability distribution of the input signal becomes closer to that of speech, $NU(\sigma_n^2)$ decreases rapidly over $U(\sigma_n^2)$ and this is clearly illustrated in the noise results given in Reference (36). Consequently for speech like signals, non-uniform scaling is advantageous in both fine or coarse quantization. Another reason in favour of non-uniform optimum quantizing is that intelligibility of speech depends substantially upon the low amplitude speech segments, and thus a non-uniform quantizer with its levels concentrated around zero will produce better subjective results than a uniform one.

2.3.1.1b Logarithmic Quantizers.

Although the optimal quantizers discussed previously provide an excellent snr for a particular variance of the input signal, their performance deteriorates rapidly as the signal's power deviates from its optimum value. This problem was recognized earlier by Cattermole[1] and Smith[32] in connection with the wide range of signal volumes encountered in the telephone systems, (the range can be easily 30 dB's) and two companding laws were devised namely the A-law (invented by Cattermole) and the μ-law. In both quantization techniques the obtained snr can be close to that of a uniform quantizer, but it remains relatively constant over a

wide range of input power. This means that for a specified dynamic range, these companded quantizers offer a reduction in the number of bits per sample required by a uniform quantizer to accommodate the same dynamic range of input signals. In both quantizers the input thresholds and the output levels are closely spaced for small ampitudes of the input signal and become progressively further apart as the input increases its amplitude. Consequently, in speech signals where the probability density function is unimodal and maximum at the origin, the frequently occurring small amplitudes will be more accurately quantized than the less probable large amplitudes. The A-law compander is described as:

$$AL(X) = \frac{AX}{1 + \log A} \qquad \text{for } 0 \leq X \leq 1/A \tag{2.10a}$$

$$= \frac{1 + \log AX}{1 + \log A} \qquad \text{for } 1/A \leq X \leq 1 \tag{2.10b}$$

where A, the compression parameter takes values close to 86 for a 7 bit speech quantizer.

On the other hand the μ-law is defined by

$$ML(X) = \text{sign}(X)\ \frac{V_o \log\left[1 + \frac{\mu|X|}{V_o}\right]}{\log(1 + \mu)} \tag{2.11.}$$

where V_o is equal to $V_o = L\sigma$, L is a loading factor and σ is the rms value of the input signal. A commonly used value for the compression parameter μ is 255. Equations (2.10.) shows that the A-law is a combination of a truly logarithmic curve employed for large amplitude signals, while for small amplitude signals the curve through the origin is linear. The μ-law, Equation (2.11.)

is not exactly linear or logarithmic anywhere but it is approximately linear or logarithmic for small and large amplitudes respectively. A comparison between μ-law and optimum quantization[36] shows that the optimum quantizer offers a maximum improvement of 4 dB's. However, the snr advantage of the optimum quantization is offset by its high idle channel noise and limited dynamic range so that in practice logarithmic quantization is always preferable.

2.3.1.2. Adaptive Quantizers.

In recent years the interest of many research workers has been directed towards quantization schemes capable of producing very wide dynamic range and better snr than the time-invariant logarithmic type quantizers. Several techniques have been proposed for the solution of the problem and they involve mainly time-varying adjustment (adaptation) of the quantizer's step size to the variance of the input signal.

In one of the earliest studies of time-varying quantizers[37] the range of the quantizer is made a function of the relative frequency of the maximum and minimum code levels generated inside a previous block of samples. A frequent generation of the maximum code level indicates that the variance of the input signal is larger than the quantizer's amplitude range which is then increased. The amplitude range is decreased if the minimum code level frequenly occurs.

In another study[38], the minimum noise power σ_n^2 quantizer is made adaptive to the statistics of the input signal. That is, the proposed quantizer estimates the probability distribution of the input signal at every sampling instant and performs a minimum mean

square error optimum quantization based on the estimated distribution.

The adaptive quantization technique investigated by Stroh[39] and Noll[40] recognizes the non-stationary nature of most real signals, like speech, and makes the reasonable assumption that the power of the input signal may vary relatively slowly with time. This time-varying quantizer involves the computation of a running maximum likelihood estimate $\hat{\sigma}_x^2$ of the input power σ_x^2 from the preceeding k input samples, followed by the normalization of the input sample by the square root of the estimate and finally the quantization of the resulting ratio, The purpose of the normalizing procedure is to produce a zero mean unit variance signal (this depends upon the accuracy of the estimate of the input power) which can then be quantized by an optimum quantizer matched to the signal's probability density function. It seems therefore that ideally when $\hat{\sigma}_x^2 \simeq \sigma_x^2$, the quantizer will produce a high snr independent of the power variations of the input signal. Noll examined the performance of this technique applied specifically to speech signals and the following two $\hat{\sigma}_x^2$ estimation methods were considered:

i) In the so-called "forward estimation", speech segments of k samples are assumed to be stationary and $\hat{\sigma}_x^2$ is given by

$$\hat{\sigma}_x^2 = \frac{1}{k} \sum_{i=1}^{k} X_i^2 \qquad (2.12.)$$

where X_i are the input speech samples. There is a dependence of the probability distribution of the resulting ratio upon the value of k. As k increases the probability distribution of the signal to be quantized changes from Gaussian ($k \lesssim 128$) to Laplacian ($k > 512$).

ii) The second method called the "backward estimation" calculates at each sampling instant the variance of the input signal using the preceeding k_1 quantized samples. Thus the normalizing factor at the n'th sampling instant is:

$$\hat{\sigma}_{x(n)} = \sqrt{\frac{a_1}{k_1} \sum_{i=1}^{k_1} \hat{X}^2_{n-i}} \qquad (2.13.)$$

where the "$\hat{}$" above the X^2_{n-i} symbol indicates quantized samples and a_1 is optimized to provide an unbiased estimator. The possibility of weighting the $\hat{X}_{n-i}$ samples of Equation (2.13.) provides marginal improvement. Stroh has shown that for a band limited stationary zero mean Gaussian input signal as the learning period k_1 increases the obtained snr tends asymptotically to a maximum value. However, k_1 must be such that the power of the signal is fairly constant during these samples. The snr advantage of the above variance estimating quantization technique over a logarithmic quantizer is in average 3 to 5 dB's.

Another efficient way of matching the quantizer's step size to the signal's variance is the "One Word Memory" adaptive quantization suggested by Flanagan, studied by Jayant[41] and developed in the laboratory by Jayant and Cummiskey[42]. The strategy of the step size adaptation is simple and can be illustrated as follows: Consider, at the n'th sampling instant, the step size of a P bit uniform quantizer to be δ_n and its output level x_n, i.e.

$$x_n = H_n \frac{\delta_n}{2}, \quad H_n = 1,3,5,\ldots. 2^P-1, \quad P \geq 2 \qquad (2.14.)$$

At each sampling instant the step size δ is multiplied by a fixed expansion-compression coefficient which is determined from the quantizer's previous output level. Thus at the (n+1)th instant the value of the step size δ (called sometimes the state variable) is:

$$\delta_{n+1} = \delta_n M_i\left(|H_n|\right) \tag{2.15.}$$

where M_i is one of i fixed coefficients corresponding to the quantizer's output levels. When P is even the number of coefficients is $\frac{P}{2}$ while for P odd there are $\frac{(P+1)}{2}$ coefficients. For a Gaussian input signal and with the multipliers appropriately defined to maximize the snr, the step size δ is for most of the time approximately that of an optimum fixed quantizer. When the values of the multiplying coefficients are not optimized the performance of the quantizer is still good with a relatively small snr loss. The only basic rule the M_i coefficients must follow is the assignment of values less, but close to unity, for coefficients corresponding to the inner quantization levels. Values between 1 and 2.5 are used for the outer levels of the quantizer. With this strategy the rate at which the step size δ is increasing is greater than its rate of decrease and the occurrence of possible subjectively serious overload errors is minimized.

The values of the multiplicative coefficients as derived by Jayant, are applicable to stationary uncorrelated input sequences and his approach does not clarify the "static" and "dynamic" behaviour of the quantizer. In the static operation the amplitude range of the quantizer matches the σ_x value of the incoming input sequence,

and the M_i coefficients must be such that the step size δ tends to its optimum value. On the other hand, the dynamic behaviour of the quantizer is related to the speed the step size δ can adapt to sudden large changes of the input's volume, and depends upon how close or far from unity are the M_i values of the inner and outer quantization levels, respectively.

Goodman and Gersho[43] in a statistically based, rigorously defined analysis, examine both the static and dynamic performance of this quantizer, and define the required coefficients for the best compromise between the ability of the quantizer to respond to sudden variation of the input power, and its steady state accuracy.

2.3.1.3. Dithered Quantization.

Before going into Differentially encoding systems, the technique of dithered quantization applied to speech signals is now considered. When in a fixed level quantizer used in PCM encoding the number of bits per sample is less than six, the quantization noise tends to be signal-dependent and perceptually annoying.

Jayant and Rabiner[44], and Wood and Turner[45] have shown that a "whitened" and thus less objectionable quantization noise pattern is obtained by dithering, while the snr is unchanged. The normal procedure of dithering is to add a pseudo-random noise sequence to the speech samples prior to quantization, and subsequently subtract at the decoder the pseudo-random samples from the decoded samples. The result is an almost white quantization error waveform. Subjective tests show that the dithered speech is perceptually

preferable but less intelligible at low bit rates ($P < 4$). Specifically, dithered quantization noise seems to mask consonant sounds more than a straight-forward quantization error.

Chen and Turner[46] suggested that since the variance of the noise with or without the dithering technique is essentially the same, the poor intelligibility at low bit rates is due to the irregular effect the dither has on the zero crossings of the speech signal. Dither can, in fact, move the position, or eliminate, or introduce new zero-crossing in the signal. From a number of schemes they propose for dithered quantization with preserved zero crossings, two of them exhibited a 1 bit advantage compared to PCM encoding with a normal fixed quantizer. Finally, dither can be applied successfully only to fixed level quantizers, as adaptive quantization techniques and especially instantaneous ones, tend to produce a signal-independent error pattern.

2.3.2. Differentially Coding Systems.

As mentioned earlier the quantizer of a PCM system operates directly on the $\{X_i\}$ samples of the input signal $X(t)$. In Differentially coding systems the error samples $\{e_i\}$ formed as the difference between the input $\{X_i\}$ samples and their estimates $\{Y_i\}$, are quantized. The reason for the formation of the error sequence $\{e_i\}$ before quantization is that in many signals, including speech, there is a strong correlation between adjacent samples and hence redundancy which is reduced by forming the error sequence $\{e_i\}$. Thus by decorrelating and then quantizing the resulting signal, Differential encoding systems are generally

more efficient when compared to PCM and provide higher snr at a given transmission bit rate.

To illustrate, in general, the advantage of differential encoding over the straight-forward quantization, consider M input samples to be PCM encoded and transmitted with a total of M·N bits. Consider also the same M samples to be Differentially encoded so M e_i error samples are quantized with N_1 bits/sample accuracy and transmitted together with N_2 bits of information related to the $\{Y_i\}$ estimation procedure parameters. As the correlation between the input speech samples is usually high, the variance of the error sequence is much smaller than that of the original speech samples and the bits per sample needed to describe, with the same accuracy as in PCM, the e_i samples are less than N, i.e. $N_1 < N$. Generally, $N_2 << N_1$ and therefore $M \cdot N > (M \cdot N_1 + N_2)$. Thus the main characteristic and objective of Differential encoders is the considerably smaller amplitude range of the error sequence, when compared with the input signal.

The method which is usually used to obtain the Y_i samples is Linear Prediction,[47] (see section 2.3.2.1.) where the estimates of the X_i input samples are formed as the weighted linear combination of some previous input samples. Linear interpolation can also be employed as an accurate estimation procedure but it is rather complex to implement and when used in feedback Differential systems looses its advantages over Linear Prediction.[48]

2.3.2.1. Differential Pulse Code Modulation (DPCM).

Differential Pulse Code Modulation systems are based on an invention by Cutler[49]. He proposed the quantization of the differences between successive Nyquist samples instead of the quantization of the input samples as in the case of PCM. Shortly after Cutler, Oliver[50], Harrisson[51] and Kretzmer[52] realized that the linear prediction theory was applicable to DPCM. They proposed predictive DPCM encoding of television signals. Since then, considerable effort has been expended in the development and understanding of DPCM systems applied to speech encoding.[53-59]

At the present although it is well known that DPCM is a more efficient way of encoding speech signals than PCM, the latter is employed almost exclusively in all the commercial digital transmission systems. This is due to two reasons:

i) At the beginning of the sixties PCM was established as a viable method of digital communications while DPCM was still being investigated.

ii) At that time the Compromise Predictors had not been developed and the dependence of the DPCM performance upon the statistics of the input signal appeared to be a serious weakness, particularly in the case of the Telecommunication networks which have to convey signals other than speech. When the long-term statistics of the input signal are different than those used in the design of the DPCM, the system may lose its encoding advantage over PCM unless a Compromise Predictor is employed.

The block diagram of the DPCM codec is illustrated in Figure 2.9, and its operation can be briefly described as follows.

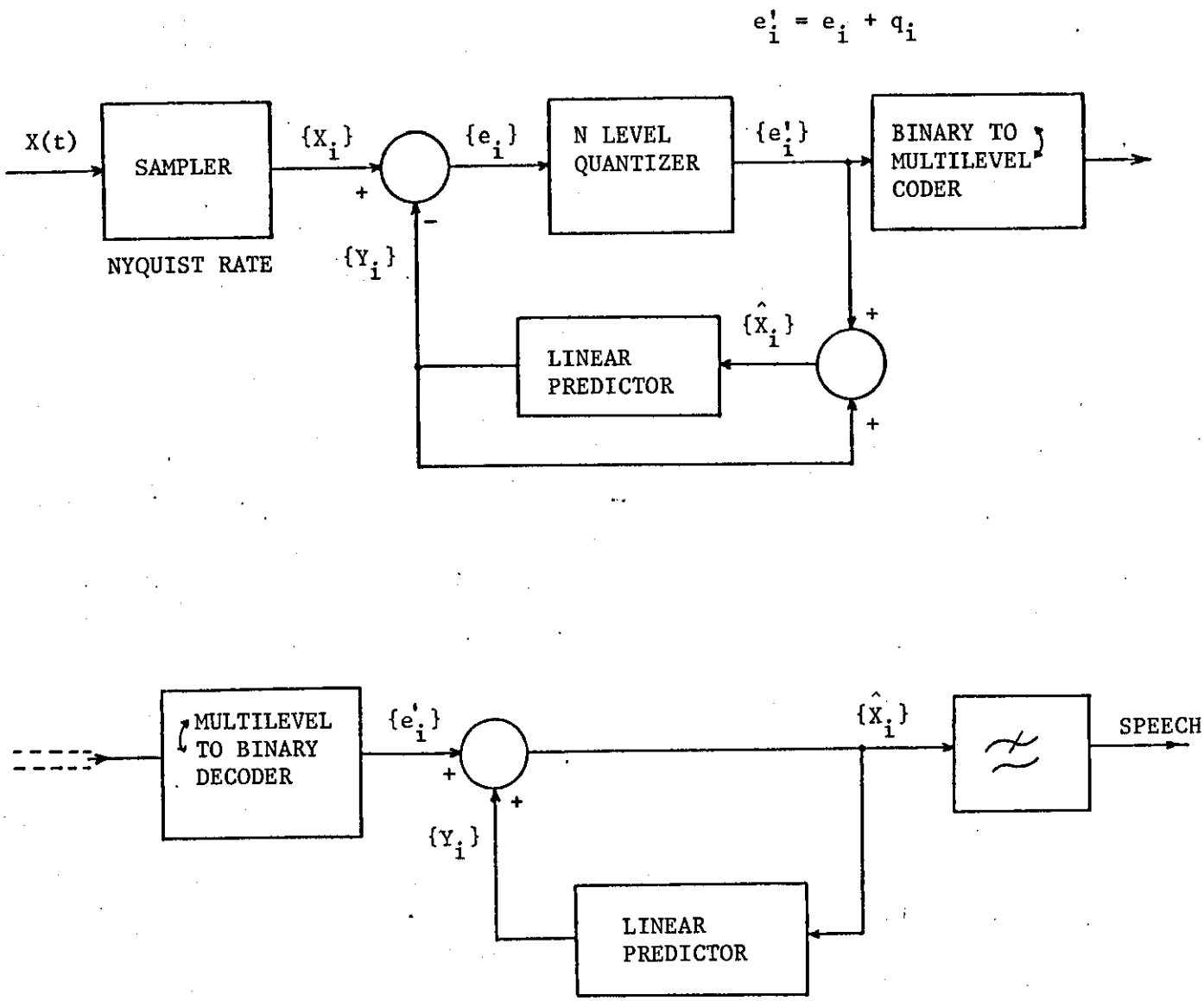


FIGURE 2.9 - The DPCM Codec.

The band limited analogue speech signal X(t) is sampled at the Nyquist rate to produce a sequence of samples $\{X_i\}$, $i=1,2,\ldots \infty$. At the same time the Linear Predictor in the feedback loop of the encoder, based on previous decoded speech samples, provides a sequence $\{Y_i\}$ of predicted samples. Each estimate Y_i is subtracted from the input samples and an error sequence $\{e_i\}$ is produced whose ith element is

$$e_i = X_i - Y_i \tag{2.16}$$

The error samples are quantized to produce $\{e_i'\} = \{e_i\} + \{q_i\}$ where q_i is the noise introduced at the ith instant by the quantization process. The samples at the output of the quantizer are then binary coded and transmitted as well as locally decoded in the feedback loop of the encoder.

The quantizer is included inside this predictive closed loop system so the quantization noise associated with the reconstructed sequence $\{\hat{X}_i\}$ is the same with that of the error sequence $\{e_i'\}$ i.e. $\{q_i\}$. This can be easily seen from the following Equations, applicable at the ith sampling instant.

$$\hat{X}_i = e_i' + Y_i \tag{2.17}$$

$$e_i' = e_i + q_i \tag{2.18}$$

$$e_i = X_i - Y_i$$

where by combining them the ith decoded speech sample is equal to

$$\hat{X}_i = X_i + q_i \tag{2.19}$$

On the other hand, when the quantizer is placed outside the

feedback loop, there is an accumulation of quantization noise at the output of the decoder.

The linear predictor employed in the local decoder, uses the previous n decoded speech samples to estimate the next incoming input sample, and Y_i is equal to:

$$Y_i = \sum_{j=1}^{n} a_j \hat{X}_{i-j} \tag{2.20}$$

The performance of a such predictor and its success in accurately predicting the incoming speech samples depends upon the values of the a_j coefficients of Equation (2.20). To determine the optimum (in a minimum mean squared error sense) set of the a_j coefficients we proceed as follows:

Using Equations (2.19), (2.20) and (2.16), the error sample e_i is equal to:

$$e_i = X_i - \sum_{j=1}^{n} a_j X_{i-j} + \sum_{j=1}^{n} a_j q_{i-j} \tag{2.21}$$

If we assume the autocorrelation of the noise samples and the cross-correlation of the noise and the input samples are both very small, the variance of the $\{e_i\}$ sequence can be expressed as:

$$\sigma_e^2 \simeq E\left[\left(X_i - \sum_{j=1}^{n} a_j X_{i-j}\right)^2\right] + E\left[q_i\right] \sum_{j=1}^{n} a_j^2 \tag{2.22}$$

When the quantization noise introduced by the system is small, the second term in Equation (2.22) is negligible and the magnitude of σ_e^2 depends on the ability of the predictor to minimize the squared difference of the first term. However, as previously mentioned, the advantage of the DPCM over PCM is due to σ_e^2 being

smaller than the variance of the input speech samples σ_x^2 and consequently the a_j prediction coefficients must be selected to minimize σ^2, where

$$\sigma^2 = E\left[\left(X_i - \sum_{j=1}^{n} a_j X_{i-j}\right)^2\right] \tag{2.23}$$

This is accomplished by expanding Equation (2.23) which becomes:

$$\sigma^2 = E\left[X_i^2\right] - 2\sum_{j=1}^{n} a_j E\left[X_i X_{i-j}\right] + \sum_{j=1}^{n}\sum_{\ell=1}^{n} a_j a_\ell E\left[X_{i-j} X_{i-\ell}\right] \tag{2.24}$$

In matrix notation Equation (2.24) is written as:

$$\sigma^2 = \sigma_x^2 - 2A^T G + A^T RA \tag{2.25}$$

where

$$A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \qquad G = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_n \end{bmatrix} \qquad R = \begin{bmatrix} \psi_o & \psi_1 & & \psi_{n-1} \\ \psi_1 & \psi_o & & \psi_{n-2} \\ \psi_2 & & \ddots & \psi_{n-3} \\ \vdots & & & \\ \psi_{n-1} & - - - - - & - & \psi_o \end{bmatrix}$$

and the elements of G and R are the values of the autocorrelation function Ψ of the input sequence $\{X_i\}$ i.e. $\psi_{(|i-j|)} = E(X_i X_j)$. The optimum set of prediction coefficients A_{opt}. which provide the minimum value of σ^2, is found by taking the derivative of σ^2 (in Equation 2.25) with respect to A and equating the result to zero.

$$\left.\frac{\partial \sigma^2}{\partial A}\right|_{A = A_{opt}} = 0$$

or $\qquad 2G + 2AR = 0$

and solving the latter Equation the optimum vector A is equal to:

$$A_{opt} = R^{-1} G \tag{2.26}$$

Using Equations (2.25) and (2.26) the minimum value of σ^2 can be obtained, i.e.

$$\begin{aligned} \sigma^2_{(min)} &= \sigma^2_x - G^T R^{-1} G \\ &= \sigma^2_x - A^T_{opt} G \end{aligned} \tag{2.27}$$

which is also the variance σ^2_e of the error sequence $\{e_i\}$ (under the assumption of fine quantization.) Notice that the value of σ^2_e is not constant or monotonically reduced as the number n of the prediction coefficients increases. This is because speech is not perfectly predictable from its past samples and so as n becomes large $\sigma^2_{e(min)}$ approaches a finite, non-zero value.

In practice the long-term autocorrelation function of the speech signal is measured and the a_j coefficients are calculated from Equation (2.26). By doing so, the predictor is matched, in an average sense, to the long-term spectrum of the speech signal. Such a predictor is relatively simple to implement and is known as time-invariant or fixed spectrum predictor.

Let us now consider the case of the simplest predictor, i.e. n = 1. Equation (2.26) defines the optimum predictor coefficient a_1,

$$a_1 = \frac{\psi_{(1)}}{\psi_{(o)}} = \frac{E[X_i\ X_{i-1}]}{E[X_i^2]} = \rho_1 \tag{2.28}$$

which is equal to the first normalized correlation coefficient ρ_1 of the input samples $\{X_i\}$. In this case the variance of $\{e_i\}$ is given by substituting Equation (2.28) into Equation (2.27)

$$\sigma_e^2 = \sigma_x^2 - A^T_{op}\, G$$

$$= \sigma_x^2 - \frac{[\psi_{(1)}]^2}{\psi_{(o)}}$$

$$= \sigma_x^2 (1 - \rho_1^2) \qquad (2.29)$$

Equation (2.29) illustrates a significant property of the optimized DPCM encoder. That is, as ρ_1 is less than one, $\sigma_e^2 < \sigma_x^2$ and DPCM holds always an advantage over PCM. On the other hand, if a_1 is equal to one ($\neq \rho_1$ in Equation 2.28) which is the case of an ideal integrator, the performance of DPCM is better than that of PCM only if $\rho_1 > .5$. This can be shown using Equation (2.25) with $n = 1$,

$$\sigma_e^2 = \sigma_x^2 - 2A^T\, G + A^T\, RA$$

$$= \sigma_x^2 - 2\,\psi(1) + \psi(o)$$

$$= 2\sigma_x^2 - 2\psi(1)$$

$$= \sigma_x^2\, 2(1 - \rho_1) \qquad (2.30)$$

and iff $\rho_1 \leq 0.5$, $\sigma_e^2 \geq \sigma_x^2$ and consequently DPCM loses its advantage over PCM.

The signal-to-noise ratio of a DPCM system can be simply expressed by

$$snr_D = \frac{E(X_i^2)}{E(q_i^2)} = \frac{\sigma_x^2}{\sigma_e^2} \; Q_{(N)} \tag{2.31}$$

$$Q_{(N)} = \frac{\sigma_q^2}{\sigma_e^2} \tag{2.31b}$$

where $Q_{(N)}$ is the ratio of the quantizing noise power σ_q^2 to the quantizer input power σ_e^2, and can be thought as the normalized quantizing noise power. The quantity $\frac{\sigma_x^2}{\sigma_e^2}$ represents the amount by which the power of the input signal can be reduced by linear prediction.

For a first order DPCM system (n = 1) employing an optimum leaky or ideal integrator, $\frac{\sigma_x^2}{\sigma_e^2}$ is given by the Equations (2.29) and (2.30) respectively and the snr becomes:

$$snr_D = \frac{1}{(1 - \rho_1^2)} \; Q_{(N)} \quad , \quad a_1 = \rho_1 \tag{2.32}$$

$$snr_D = \frac{1}{2(1 - \rho_1)} \; Q_{(N)} \quad , \quad a_1 = 1 \tag{2.33}$$

Comparison of Equations (2.32) and (2.33) shows a slight snr advantage of the $a_1 = \rho_1$ optimum case over the $a_1 = 1$ non-optimum one. Another advantage of the optimum system is the exponentially decaying effect of digital channel transmission

errors in contrast with the error accumulation which occurs in the $a_1 = 1$ case.

Equations (2.32), (2.33) apply only when the quantizing noise power is small compared to the signal's power. The derivation of the exact signal-to-noise ratio formula of a first order DPCM system, where the quantizing noise in the feedback loop is also taken into consideration, is given by Gish[60] and O'Neal[59], as:

$$snr_D = \frac{1 - \rho_1^2 Q_{(N)}}{(1 - \rho_1)^2 Q_{(N)}} \tag{2.34}$$

Notice that for small values of Q(N), Equation (34) takes the form of Equation (2.32) which is frequently used as a good approximation of the DPCM snr.

Having in mind that the snr of a PCM system is given by $Q(N)^{-1}$, the quantity $\frac{\sigma_x^2}{\sigma_e^2}$ also represents the signal to noise ratio improvement factor of a DPCM system over PCM. Consequently, Equation (2.31) can be expressed in desibels as:

$$snr_D = SNI - 10 \log_{10} Q_{(N)} \tag{2.35}$$

where the signal-to-noise improvement, SNI is equal to:

$$SNI = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2}$$

and when $\sigma_x^2 = 1$

$$SNI = - 10 \log_{10} \sigma_e^2 \tag{2.36}$$

In the specific case of a DPCM system, employing the μ-law quantizer, the values of $Q_{(N)}$ can be approximately represented[61]

by:

$$10 \log_{10} Q(N) = + 8.5 - 6.02N \quad \text{for } \mu = 100$$

$$10 \log_{10} Q(N) = + 10.1 - 6.02N \quad \text{for } \mu = 255 \qquad (2.37)$$

and thus the signal-to-noise ratio of this system can be expressed as:

$$snr_D = - 8.5 + 6.02N + SNI \quad \text{for } \mu = 100$$

$$snr_D = - 10.1 + 6.02N + SNI \quad \text{for } \mu = 255 \qquad (2.38)$$

The exact value of the normalized noise power $Q_{(N)}$ of an N level quantizer is difficult to be calculated. $Q_{(N)}$ depends on N, the structure of the quantizer, and the probability density function (pdf) of the quantizer input error sequence $\{e_i\}$. When a first order Markov process defined as

$$X_i = a X_{i-1} + S_i \quad , \qquad i = 1,2,\ldots$$

where $\{S_i\}$ is a sequence of zero mean random numbers and $a < 1$, is encoded by a First Order DPCM encoder the pdf of $\{e_i\}$ is the convolution of the pdf's of the two independent random variables S_i and aq_{i-1} [59].

This complication however, can be avoided when the pdf of the $\{e_i\}$ sequence is assumed to be identical to that of the input sequence $\{X_i\}$ and this leads to a good estimate of $Q_{(N)}$. The $Q_{(N)}$ values of optimum quantizers have been tabulated in (35) and (36) for input sequences with Gaussian, Laplacian and Gamma pdf respectively.

2.3.2.1a. Adaptive Differential Pulse Code Modulation (ADPCM).

Having discussed the optimum predictor and the snr performance of the DPCM system, it is clear that a priori knowledge of the statistics of the input samples is required for an efficient system design. This is because, given the input statistics, a predictor can be obtained which minimizes the variance of the samples to be quantized while an optimum quantizer will produce minimum quantizing noise. However, only a small amount of a priori knowledge of the speech statistics is known and in addition these statistics change with the time due to different speakers and to variations in the speech sounds. Consequently adaptive predictors and quantizers, which are able to follow the statistical variations in the input signal, can be used to increase the encoding efficiency of a DPCM system. The resulting codecs with adaptive quantizers and/or adaptive predictors are known as ADPCM systems. First, a few adaptive prediction methods are considered.

A. Adaptive predictors.

Adaptive predictors in contrast with the fixed spectrum ones, change the values of their a_j coefficients according to short-term variations of the spectral properties of the speech signal.

One way of updating a_j is to measure the short term autocorrelation function in blocks (BL) of buffered speech samples and then estimate the coefficient vector A from Equation (2.26). The a_j coefficients are therefore periodically updated at time intervals equal to the duration of BL. In order to determine the short time autocorrelation function, the input or locally decoded speech samples can be used, resulting to two estimation schemes,

prefixed by the terms "Forward" and "Backward". In the Forward scheme, which produces better prediction accuracy than the Backward one, the values of a_j are required to be transmitted to the receiver in addition to the quantized e_i samples. This does not consume extensive channel capacity as the coefficients tolerate coarse quantization and slow updating. A detailed comparitive review of the snr performance of various DPCM and ADPCM systems is given by Noll[62].

Another approach in updating the a_j coefficients is obtained using sequentially adapting estimation techniques such as gradient search methods, and the Kalman filter algorithms. In these techniques the coefficient adaptation is made at every Nyquist sampling instant. Also, the estimates of the coefficients are obtained from data which is available in both the encoder and decoder at the transmitter and receiver respectively, and therefore a separate a_j transmission procedure is unnecessary. Cummiskey,[55] in his ADPCM studies, employed with success the steepest descent gradient algorithm where each coefficient is updated according to:

$$a_{k+1}(j) = a_k(j) - c \frac{\partial [f(e_k)]}{\partial a_k(j)} \tag{2.39}$$

where k = kth sampling instant, $f(e_k)$ is a function of the prediction error e_k and c is a function of the $\hat{X}_k$ sequence. In his work, the e_k sgn(e_k), and e_k^2 error functions are resulting in the following updating Equations:

$$a_{k+1}(j) = a_k(j) - c_1 \frac{\text{sgn}(e_k') \hat{X}_{k-j}}{\sum_{i=1}^{n} |\hat{X}_{k-i}|} \tag{2.40}$$

and

$$a_{k+1}(j) = a_k(j) - c_2 \frac{e'_k \hat{X}_{k-j}}{\sum_{i=1}^{n} \hat{X}^2_{k-i}} \tag{2.41}$$

where c_1 and c_2 are optimizing constants.

More recently Gibson, Jones and Melsa[63,64] proposed and examined the performance of ADPCM systems with predictors updated by the Stochastic Approximation method and the Kalman algorithms. The Stochastic Approximation predictor is similar to that of Equation (2.41) and is characterized by the following Equation:

$$a_{k+1}(j) = a_k(j) + g \frac{e'_k \hat{X}_{k-j}}{M + \frac{1}{n} \sum_{i=1}^{n} \hat{X}^2_{k-i}} \tag{2.42}$$

where the constant g controls the adaptation rate of the algorithm. The denominator of the second term behaves as an automatic gain control which tends to equalize the adaptation rate of the algorithm as the mean square of the speech varies. Thus when the mean square value of the input signal increases the second term in Equation (2.42) decreases. In this way, overcorrections of the a_j coefficients are avoided and wild oscillations of the estimates are prevented. The constant M is a bias term introduced to compensate for the low values of $\hat{X}_k$ during periods of silence.

The estimation of the a_j coefficients using the Kalman filter procedure, is more accurate than the previous algorithm of Equation (2.42) but it is also more complicated. The adaptation of the prediction coefficients is described, in a vector form as:

$$A_{k+1} = A_k + K_k \, e'_k \, \bar{\hat{X}}_{k-1} \tag{2.43}$$

$$K_k = \frac{\bar{Va}_{k-1}}{V + \bar{\hat{X}}^T_{k-1} \, \bar{Va}_{k-1} \, \bar{\hat{X}}_{k-1}} \tag{2.44}$$

$$\bar{Va}_{k+1} = \left[I - K_k \, \bar{\hat{X}}_k \right] \bar{Va}_k \tag{2.45}$$

where A is the a_j vector, $j = 1,2,\ldots,n$, $\bar{\hat{X}}_{k-1} = \left[\hat{X}_{k-1}, \hat{X}_{k-2}, \ldots \hat{X}_{k-n}\right]^T$, $\bar{Va}_k$ is the error variance in a_j and represents the accuracy of the estimates of the coefficients. One can find many mathematically elegant derivations of the Kalman filter[65,66] but basically the algorithm of Equations (2.43), (2.44) and (2.45) can be simply considered as a sequential minimization of the square of the prediction error e_k. Furthermore it is reasonable to make the K_k variable proportional to the error variance Va_k since this would cause the a_j coefficients to receive larger corrections for larger errors. The term $\bar{\hat{X}}^T_{k-1} \, \bar{Va}_{k-1} \, \bar{\hat{X}}_{k-1}$ is included as a normalizing function while the V constant provides a lower bound to the value of K_k. In fact if $\bar{Va}_{k-1}$ is made equal to I then Equation (2.43) becomes identical to Equation (2.42). The main conclusion which can be drawn from the computer simulation results[63] are:

i) The snr advantage of the ADPCM system using the Kalman predictor is only 0.3 dB over the ADPCM which employs the stochastic approximation predictor. Thus in an actual hardware implementation of a such encoder operating with output bit rates between 12 and 24 Kbits/sec. (i.e. with a number of quantization levels between 3 and 8),

the considerably simpler stochastic approximation predictor should be used.

ii) This snr advantage increases with the decrease of the quantization noise and consequently the poor performance of the quantizer limits the estimation accuracy of the Kalman predictor. Because of this, the minimum number of quantization levels which produces any acceptable speech quality was found to be five which corresponds to a transmission rate of 18.6 Kbits/sec. Systems using three or four level quantizers exhibited considerable granular noise, poor prediction accuracy, and they were neglected. An attempt to lower the transmission bit rate to 16 Kbits/sec. by switching alternatively the quantization process between a 3 and a 4 level quantizer resulted in a worst encoding performance than the 4 levels system.

The last prediction scheme to be mentioned in this section is a rather sophisticated one used by Atal and Schroeder in their Adaptive Predictive Coding system[67]. They achieve better prediction of the speech waveform than the methods previously discussed by exploiting the quasi-periodic nature of the speech wave, in addition to a Linear Prediction modelling of the speech process. The block diagram of the system is shown in Figure 2.10, and its prediction process can be described as follows. A predictor of the form $F_1(z) = Bz^{-m}$ where B is an amplitude variable and m is the pitch period length variable, removes the redundancy due to waveform similarities which exist between pitch periods. This is simply done by delaying the speech waveform by one pitch period and forming a difference signal $e_1(n)$ between successive pitch periods. The m

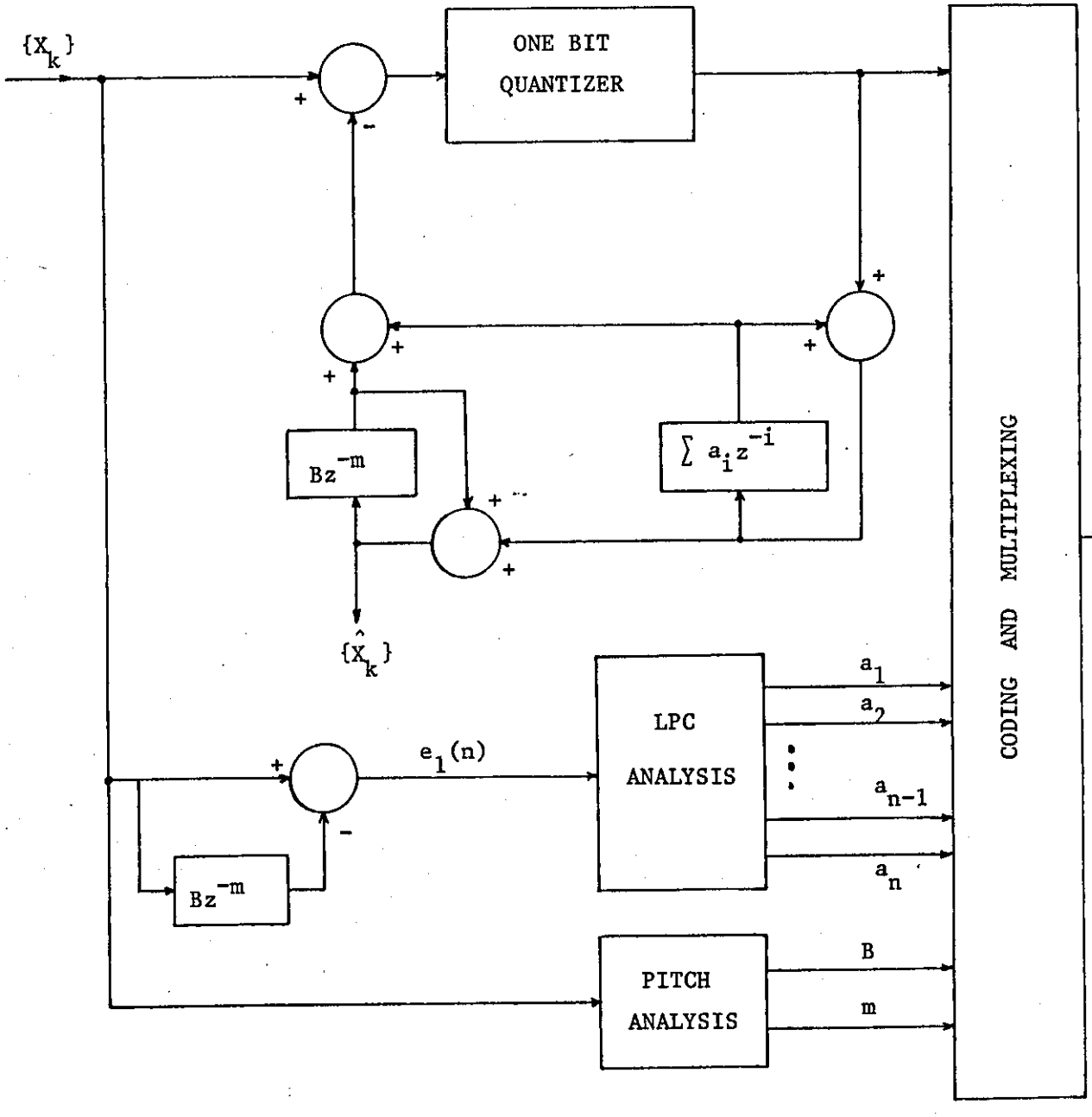


FIGURE 2.10 - The Adaptive Predictive Coding System.

variable is automatically extracted using a correlation pitch extraction procedure where the maximum value of the normalized correlation coefficient is detected.

A $F_2(z) = \sum_{i=1}^{n} a_i z^{-i}$ predictor which models the spectral envelope of the speech signal is then used to remove any format information from the $e_1(n)$ difference signal. In this way a second difference signal $e_2(n)$ is produced which is quantized by a one bit adaptive quantizer and transmitted together with B, m, and a_i's to the receiver. At the receiving end an inverse procedure using $F_1(z)$, $F_2(z)$ and the quantized samples $\hat{e}_2(n)$, produces an approximation of the original speech waveform.

This system can achieve very large snr gains over PCM. However, the large amount of computations required to determine its parameters together with its complexity, limits its application for real-time communications.

B. Compromise Predictors.

Having referred to fixed and adaptive predictors designed according to the statistics of a specific input signal, the possibility of producing a predictor which performs well when predicting several different types of input signals will be briefly considered. Such a predictor is known as the "compromise" predictor and it is required when different types of signals are transmitted in a Telecommunication network. In this case a DPCM system employing a predictor designed matched to a X(t) input signal, could loose its advantage over PCM when a statistically different signal Y(t) is encoded.

O'Neal and Stroh[61] studied four cases of compromise predictor optimization applied to two signals, X(t) and Y(t). Assuming that the autocorrelation functions of X(t) and Y(t) are respectively $\Psi_{X(i)}$ and $\Psi_{Y(i)}$, the mean squared value of the resulting error sequence in the DPCM encoder will be according to Equation (2.25)

$$\sigma_1^2 = \sigma_x^2 - 2A^T G_x + A^T R_x A \tag{2.46a}$$

$$\sigma_2^2 = \sigma_y^2 - 2A^T G_y + A^T R_y A \tag{2.46b}$$

The predictor coefficients a_j are then optimized with respect to one of the next four criteria:

(1) $b\sigma_1^2 + c\sigma_2^2$ is minimized where b and c are the time percentages of occurrance of the X(t) and Y(t) signals respectively.

(2) σ_1^2 or σ_2^2 is minimized under the constraint that $\sigma_1^2 = \sigma_{2\ min.}^2$ i.e. the snr advantage of the encoder for both signals will be equal over PCM.

(3) The constraint becomes $\sigma_1^2/\sigma_{1\ min}^2 = \sigma_2^2/\sigma_{2\ min}^2$ which means that the obtained error variance in the encoder will be greater than $\sigma_{1\ min}^2$ or $\sigma_{2\ min}^2$ by the same amount.

(4) Finally σ_1^2 or σ_2^2 is minimized while the other is kept to a constant value.

The results[61] show that a DPCM system employing a compromize predictor is an advantageous over PCM even when statistically different signals are encoded by the system. However because of the constraints imposed in the optimization procedure, the snr of

such an encoder is not as good as the one obtained by DPCM when it is optimized for one specific signal.

C. Adaptive Quantizers.

Quantization is the other important operation which determines the encoding performance of a DPCM system. All types of time-invariant and time-variant quantizers can be used in a DPCM codec. In fact, during recent years, many systems have been proposed combining fixed and adaptive predictors and quantizers. Noll in his ADPCM studies[62] obtained the best snr performance from an encoder employing a 12 coefficient block adaptive Forward estimation predictor and a Forward estimation optimum Gamma quantizer. In the ADPCM system of Gibson and others[63] the quantizer used together with the sequentially adaptive Kalman predictor was a Jayant's adaptive quantizer with its levels spaced optimally for a Laplacian probability density function input. However, as already mentioned, the acceptable performance of this ADPCM encoder is limited to transmission bit rates > 18.6 Kbits/sec., despite the high efficiency of Jayant's adaptation procedure.

The objective in the design of a good ADPCM quantizer is to adapt successfully to both the long term syllabic variation as well as to the short term pitch variations of the speech waveform. One way in realizing such a quantizer will be of course the use of pitch information so that the quantizer's amplitude range is properly increased when a local maximum is detected in the voiced speech waveform shortly after a pitch pulse. This scheme would undoubtedly perform well but the cost and the complexity makes, at the present, its implementation unjustified. Cohn and Melsa[68] in their ADPCM

encoder proposed a much simpler alternative, the Pitch Compensating Quantizer (PCQ). Here the algorithm used to compute the quantizer's adaptive state variable δ_n operates in two modes, that is, an envelope detector is used for the syllabic adaptation while a Jayant loop is used for the pitch compensation. The long term syllabic variations of $\{e_i\}$ are tracked by a scaled average of the magnitude of $\{e_i\}$ or $\{\hat{X}_i\}$. This is because the envelopes of $\{e_i\}$ and $\{X_i\}$ tend to vary proportionally, and either of these sequences can be used in order to obtain an acceptable estimate of the long term syllabic variations in $\{e_i\}$. In voiced sounds, and particularly when the pitch peaks occur the quantizer detects a possible pitch pulse with its outermost levels specially set at values higher than normal. When the output of the quantizer corresponds to one of those outermost levels, the adaptation algorithm of the step size reacts as if the sequence of the samples to be quantized is related to a pitch pulse, and the quantization step size is significantly increased. Now, because the outermost levels of the quantizer can occur in instants other than those of pitch pulses, the quantization step size δ_n is permitted to rapidly decay back to its long term average value after a sudden "pitch" expansion. When a false pitch pulse is detected, the quantizer is mismatched from the amplitude range of the signal only for a few samples with no serious deterioration of its performance. Finally, in this particular scheme, the set of output and threshold quantizing levels were not chosen according to some known probability density function as in references (62,63) but a random computer simulated search was used to determine the quantization characteristic which produces minimum

quantizing noise.

Qureshi and Forney[69] observed that the adaptive quantizer was the most important element in their ADPCM encoder. Moreover, the subjective quality with fast quantizer adaptation seemed to be limited by granular noise rather than overload distortion. A slower quantizer adaptation strategy with the capability of rapid expansion upon detection of overload was therefore required. In an attempt to produce an easily implemented PCQ quantizer, they proposed a similar scheme which uses two Jayant's adaptive loops: one for syllabic adaptations and another for pitch compensation. The adaptation of the step size δ_n is therefore accomplished according to the Equation:

$$\delta_n = a_n . b_n . c$$

where c is a normalizing constant, a_n is the output sample from Jayant's loop that tracks the syllabic variations in the input speech signal, and b_n the output sample from the second pitch compensating Jayant's adaptation loop.

2.3.2.1b. Entropy Encoding applied to DPCM.

Suppose that a source S outputs statistically independent symbols S_i, $i = 1,2,\ldots,q$, and the probability associated with S_i are p_i, $i = 1,2,\ldots,q$. The Entropy of the above source is defined[70] as:

$$H(S) = \sum_{i=1}^{q} p_i \log \frac{1}{p_i} \qquad (2.47)$$

Now, 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:

$$\bar{L} = \sum_{i=1}^{q} p_i \ell_i \tag{2.48}$$

and the following important property of the Entropy can be proved

$$H(S) \leq \bar{L} \tag{2.49}$$

Equation (2.49) shows the Entropy of the source to be the lower bound of the codeword average length. This means that the best coding procedure, where codeword 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 $\frac{H(S)}{\bar{L}} = E$ is defined as the Efficiency of the coding procedure, while (1-E) is the Redundancy.

Entropy Encoding is a variable-length coding procedure applied at the output of an Encoder to assign short codewords to high probable output quantization levels and longer codewords to less probable ones. In this way the average transmitted codeword length could be approximately equal to the Entropy of the signal at the output of the quantizer. Much of the redundancy in the speech waveform is eliminated when it is encoded by a DPCM encoder. Additional coding of the DPCM output using Entropy encoding can result into a further snr improvement at a given transmission bit rate.

O'Neal[71] compared the performance of a DPCM system with

entropy coding to a simple DPCM arrangement. The first system employed a uniform quantizer while the second used a fixed optimum Max quantizer. The results of this theoretical study shows that for a large number of quantization levels and when the quantizer input signal has a Laplacian pdf, the entropy coding could provide a further snr improvement of 5.6 dB's. The difficulties of practically implementing this technique are also mentioned in the paper. Variable length codes imply the use of a buffer which necessitates a buffer management scheme to handle initial synchronization, underflow and overflow. The codes must have good synchronization and reconvergence properties in the presence of a channel error.

Entropy encoding techniques were used in both the ADPCM systems of Melsa[68] and Queshi[69]. One reason for this was the design objective of an output transmission bit rate of 9.6 and 16 Kbits/sec. at a sampling rate of 6.4 kHz. This leaves 1.5 bits to encode each sample in the first case and 2.5 bits in the second. Furthermore, even if three levels are to be used in the quantizer, a fixed length coding procedure would require 1.58 bits/sample and an acceptable 9.6 Kbits/sample encoder cannot be obtained. On the other hand, with variable length codes, five quantization levels would result to an average of 1.48 bits/sample while a bit rate of 2.5 bits/sample could easily accommodate 7 or 9 quantization levels. Another reason is the use of the Pitch compensating quantizer in these ADPCM systems. The addition of the outermost pitch compensating quantization levels, which occurs 1% or 2% of the time, can be quite costly in terms of transmission bit rate. Specifically, in a fixed length coding the addition of these two levels in a three level

quantizer increases the required bits/sample from 1.58 to 2.33, i.e. a 47% increase, while with entropy coding the numbers are 1.25 and 1.37 bits/sample respectively.

The ADPCM system in reference (68) makes use of variable input fixed output codes. In this coding technique each codeword has a fixed length but may represent a different number of quantization output levels. The coder accepts the quantization output levels and waits until a fixed length message is formed, which is then transmitted. The main property of the technique is its resistance to channel error. This is because all the bit sequences in the channel are with the same length and thus loss of the word synchronization due to channel error is avoided. Such errors cannot be allowed to accumulate since that would cause the receiver buffer to eventually overflow or underflow. Qureshi's[69] ADPCM system employs a variable input variable output coding technique. The scheme is showing good synchronization properties due to a strategy employed to insert or delete code-words at the receiver after the occurrence of channel errors.

2.3.2.2. Delta Modulation (DM).

Most of the power in speech resides in its lower frequencies and consequently when sampling at the Nyquist rate considerable oversampling frequently occurs. DPCM encoders exploit the high correlation of the "over-sampled" speech by various sophisticated forms of predictors and quantizers, as previously described.

It is natural to presume that the relative complexity in DPCM encoders could be avoided by a further increase in the

correlation of the input speech samples, i.e. by increasing the sampling rate. Simpler forms of prediction than those used in DPCM would then result. Oversampling would also remove the necessity of using multi-level quantizers in the encoder. Consequently, we could consider Differential encoders which highly oversample the input signal and incorporate a one bit quantizer together with a simple predictor in the feedback loop. Such encoders, known as Delta Modulation encoders or just Delta Modulators, combine low complexity with good waveform tracking properties. A thorough examination of Delta Modulation encoding techniques is given by Steele[2].

The simplest form of DM is the Linear Delta Modulator (LDM) of Figure 2.11 where the input signal X(t) band limited to f_c, is sampled at a frequency f_p which is much higher than the Nyquist frequency, to produce the input sequence $\{X_k\}$. An error sequence is formed as:

$$e_r = X_r - Y_{r-1} \tag{2.50}$$

which is then quantized by a two level quantizer $\pm \delta$ (the value of δ is constant). The Local decoder forms Y_r, the prediction of X_r, by simply integrating the output of the quantizer, i.e.

$$Y_r = Y_{r-1} + a\,\delta\,b_r \tag{2.51}$$

where $b_i = \mathrm{sgn}(e_i)$ and $a = 1$ for an ideal integrator or $a < 1$ for a leaky one. The output of the quantizer $\pm \delta$ is then transmitted as a one bit word. The decoder at the receiving end is identical to the Local decoder at the encoder, and the recovered signal, $\hat{X}(t)$ is obtained by passing $\{Y_k\}$ through a Low-pass filter

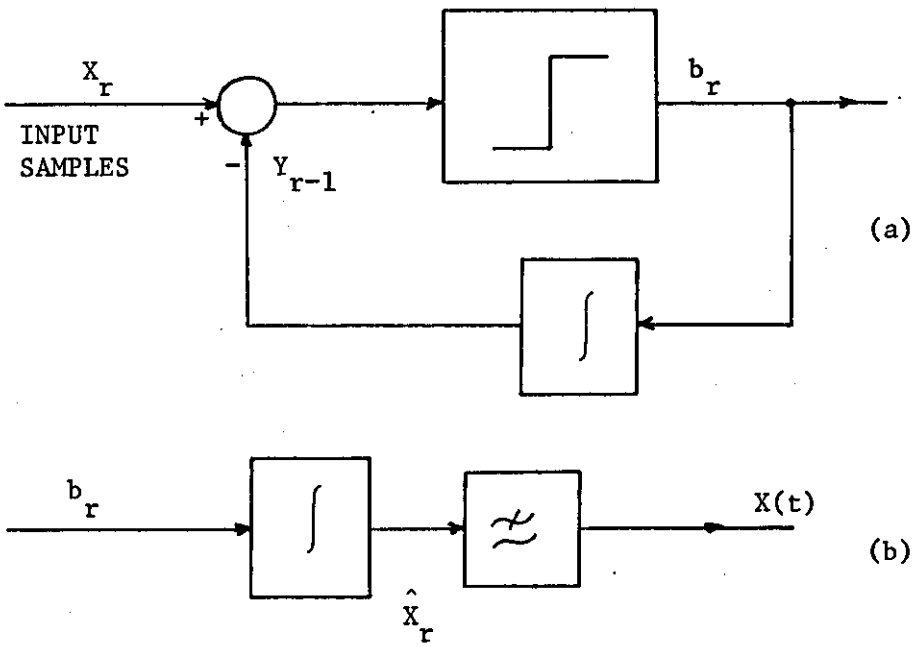


FIGURE 2.11 - The Linear Delta Modulator
(a) Encoder
(b) Decoder

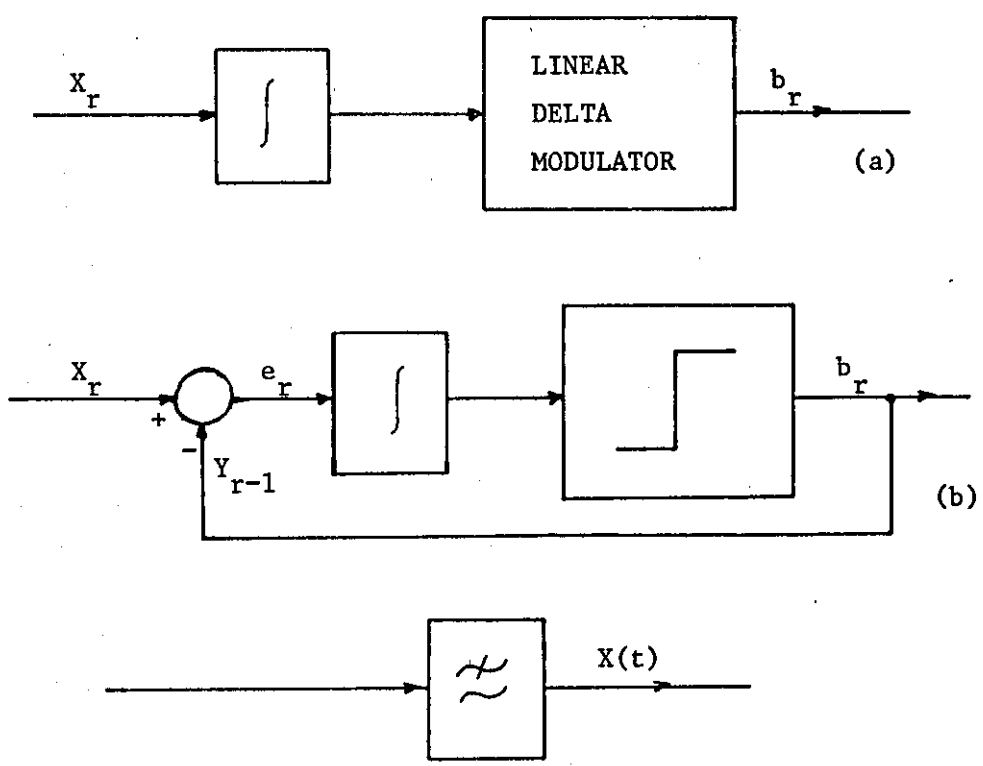


FIGURE 2.12 - (a) The Delta Sigma Modulator (DSM)
(b) The Simplified Form of DSM

having a cut-off frequency f_c which removes the out-of band noise.

The rate of change in the values in the $\{Y_k\}$ sequence, namely ∇Y_k, is an important characteristic of the encoder. This is because it determines the ability of Y_k to adapt in sudden amplitude changes of X_k and therefore to follow effectively the input signal with a minimum quantization error. Obviously ∇Y_k depends upon the δf_p product. When δf_p is such that Y_k is correctly tracking the input sequence with an error $< \delta$ the noise introduced from the encoding procedure is called "granular" noise or quantization noise. However it is possible for a slope overload situation to arise when the feedback sequence $\{Y_k\}$ is not able to track the input signal. "Overload" noise is then produced which is larger than the granular noise. For a sinusoidal input signal, $E_s \sin 2\pi f_s t$, the necessary condition to avoid slope overload is:

$$E_s \, 2\pi \, f_s \leq \delta f_p \tag{2.52}$$

and the maximum amplitude E_m of the sinusoidal input signal which does not overload the encoder is:

$$E_m = \frac{\delta f_p}{2\pi \, f_s} \tag{2.53}$$

Now, to calculate the signal-to-noise ratio of the LDM we use the following empirical expression[72] for the quantization noise power σ_n^2,

$$\sigma_n^2 = K \frac{f_c \, \delta^2}{f_p} \tag{2.54}$$

and therefore:

$$snr = \frac{\sigma_x^2}{\sigma_n^2} = \frac{1}{K} \frac{f_p}{f_c} \frac{\sigma_x^2}{\delta^2} \tag{2.55}$$

where σ_x^2 and f_c are the variance and the frequency band of the input signal, and K is an empirical constant. Noting that for a sinusoid $\sigma_x^2 = \frac{E_s^2}{2}$ and using Equation (2.55) the peak snr, namely $\hat{snr}$, is:

$$\hat{snr} = \frac{1}{8\pi^2 K} \frac{f_p^3}{f_c f_s^2} \tag{2.56}$$

Although the accuracy of Equation (2.56) depends upon K, the value of which varies with f_p/f_c and δ, it shows the important property that the snr in LDM varies proportionally with the cube of the transmission bit rate.

The calculation of an accurate snr formula in LDM is a difficult problem to solve and the attempts which have been made are complicated[73 to 79]. The usual approach is to calculate the granular and overload noise separately and add them to obtain the total noise expression.

To improve the performance of a Linear DM, double integration, i.e. the combination of two integrators in series, can be used in the feedback loop of the encoder. The idea behind this modification is to allow the prediction samples Y_r to respond faster in the amplitude changes of the input signal. At the output of a double integrator the rate of change in Y_r is porportional to the second derivative of the input signal. Thus for a $E_s \sin 2\pi f_s t$ input, the rate of change in Y_r is $E_s (2\pi f_s)^2$ and therefore the overload condition is specified by:

$$E_s (2\pi f_s)^2 = \delta f_p \tag{2.57}$$

When Equations (2.57) and (2.52) are compared, we see that the quantization step sizes which overload the single and double integration encoders are δ and $\frac{\delta}{2\pi f_s}$ respectively. Consequently double integration offers the advantage of allowing a considerably smaller step size to be used without overloading the encoder which automatically leads to a reduction of the granular noise. It can be shown that the peak snr in the case of a double integration DM is

$$\text{snr} = \frac{1}{8\pi^2 K_d} \frac{f_p^5}{f_s^2 f_{c_2}^3} \tag{2.58}$$

where f_{c_2} is the break frequency of the second integrator and K_d an empirical constant. The double integration DM shows an improvement of 5 to 10 dB's over the single integration LDM when f_s = 800 Hz and $f_p \geq 12 f_{c_2}$. However, the fast response of the double integration predictor can cause instabilities in the encoding of speech signals and this is the main disadvantage of the scheme. This problem is solved using Delayed encoding techniques[80] where the encoder is allowed to look-ahead into the input signal and properly slow down very fast adaptations in Y_r.

One characteristic in the performance of LDM encoders is their dependence on the frequency of the input signal, as shown in Equation (2.52). Now, before going to Adaptive DM, we briefly consider the Delta Sigma Modulation (DSM) encoder which overcomes the above frequency limitation. Here an additional integrator is used in the front of the encoder as shown in Figure 2.12a. Because of the relationship:

$$\int X_r \, dX_r - \int Y_r \, dY_r = \int e_r \, de_r \tag{2.59}$$

the encoder can be reduced to the simpler form of Figure 2.11b, which employs only one integrator located prior to the quantizer.

When a signal $E_s \sin 2\pi f_s t$ is applied to the input integrator of the arrangement in Figure 2.12a, the LDM which follows is presented with an $-\frac{E_s}{2\pi f_s} \cos 2\pi f_s t$ signal whose maximum slope is E_s. Consequently the overload expression for DSM is described by:

$$E_s = \delta f_p \tag{2.60}$$

and clearly is independent of the frequency of the input signal. Using Equation (2.60) and applying a similar argument with those in the LDM, we can find the peak signal-to-noise ratio of the DSM to be:

$$\hat{snr} = \frac{3}{8\pi^2 K_s} \frac{f_p^3}{f_c^3} \tag{2.61}$$

Observe that in DSM, as in LDM, $\hat{snr}$ is proportional to the cube of the sampling frequency f_p.

2.3.2.2a. Adaptive Delta Modulation (ADM).

When the input signal is stationary, the f_p and δ parameters could be arranged for the LDM to provide a reasonable $\hat{snr}$. However, the non-stationary nature of speech signals suggests the need for some form of adaptation of the feedback signal Y_r, and as f_p is usually fixed, δ is made to adapt its magnitude to the statistical variations of the input signal. In this way, the variable step size δ results in a high snr for a wide range of input power.

The first ADM system called High Information Delta Modulator (HIDM) was proposed by Winkler[81] and it is shown in Figure (2.13). Its adaptation strategy is based on the observation that a possible overload condition is revealed at the output of the encoder by a sequence of identical bits. At the same time, alternative polarity bits indicate that a smaller step size should be used. Specifically, the step adaptation algorithmis formulated as:

a) the step size δ is doubled if the current and previous two binary outputs are of the same polarity,

b) if the last two output bits are of opposite polarities, then δ is halved,

c) in all the other cases the step size δ is kept unaltered.

The HIDM encoder has a similar peak snr but an improved dynamic range compared with LDM, and its adaptation algorithm is better suited for encoding TV signals rather than speech signals.

Many other systems followed[82 to 88] which also make significant changes in δ every sampling instant by observing the patterns of a few consecutive bits at the output of the encoder. Such ADM systems are known as Instantaneously Companded Delta Modulators (ICDM). A typical example of an ICDM speech encoder is the one proposed by Jayant[86]. Its step size adaptation rule is closely related with that of Jayant's multi-level adaptive quantizer[41]. In the latter, as we already mentioned in section 2.3.1.2, multiplicative coefficients are assigned to the quantization levels so the step size δ for the (n+1)th sampling instant is equal to δ_n multiplied by the $M_{(j)}$ coefficient which corresponds to the output of the

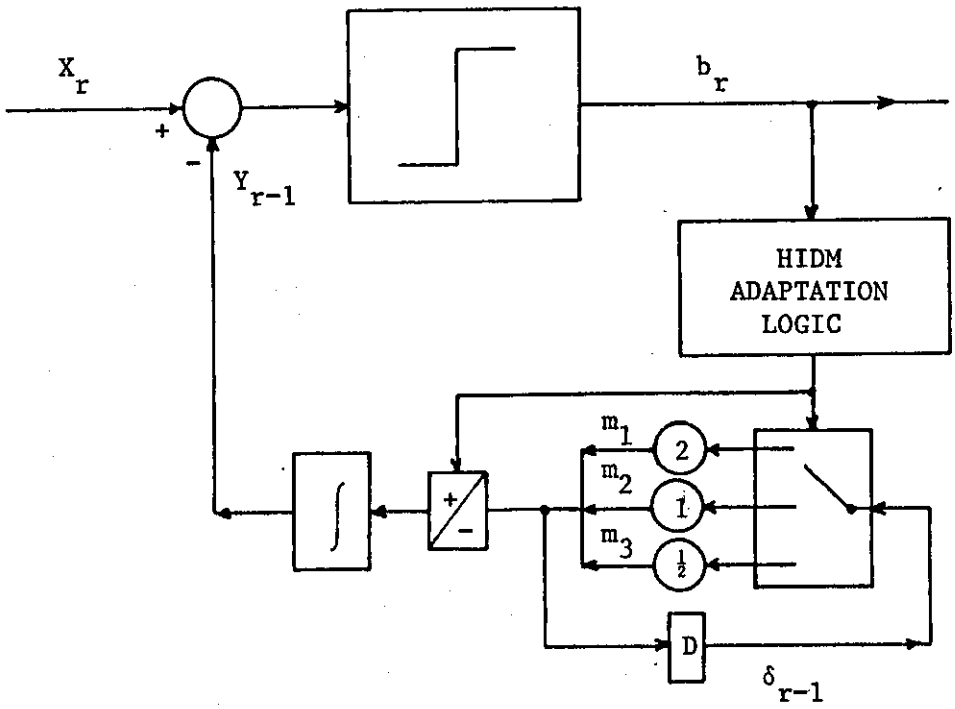


FIGURE 2.13 - The High Information Delta Modulator.

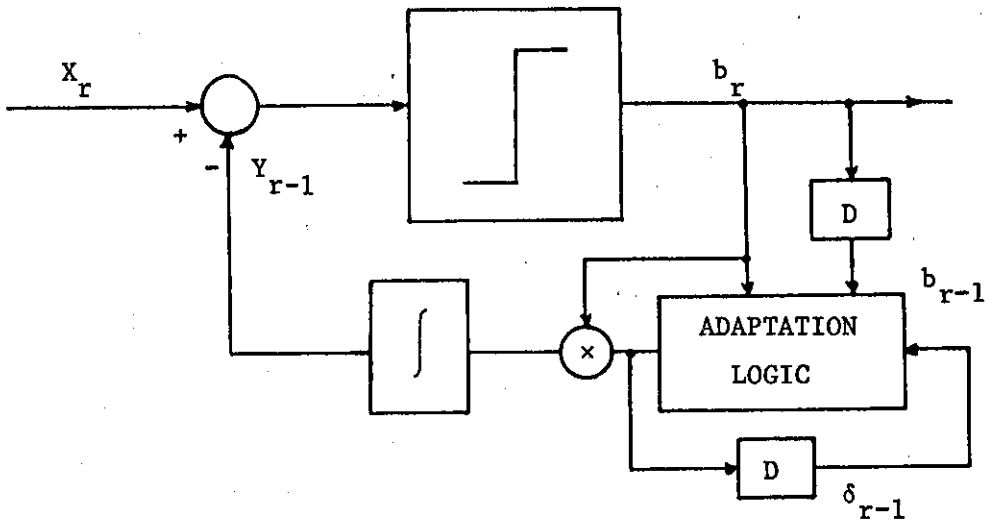


FIGURE 2.14 - Jayant's Adaptive Delta Modulator.

quantizer at the nth instant. If the quantizer is to reduce its number of levels to two, the adaptation algorithm fails because only one $M_{(j)}$ coefficient can be assigned to the two levels, say M_1, and δ will continuously increase or decrease its size when $M_1 > 1$ or $M_1 < 1$.

Thus the only way to make the adaptation stable and the step size δ to track the input signal is to employ two coefficients M_1 and M_2 ($M_1 > 1$, $M_2 < 1$) while the decision of which one coefficient is to be used at each sampling instant, is made by observing two consecutive bits at the output of the quantizer. Two identical bits indicate the use of M_1 so δ is increased and for two bits with opposite polarities M_2 is used to decrease the step size. Therefore δ is expressed as:

$$\delta_r = \delta_{r-1} \cdot M^{b_r\, b_{r-1}} \tag{2.62}$$

where $1 < M_{opt} < 2$ and $M_1 = M$, $M_2 = \frac{1}{M}$. The encoder is shown in Figure (2.14). Y_r, the feedback signal, is again formed by an integration as:

$$Y_r = Y_{r-1} + a\, \delta_r \cdot b_r \tag{2.63}$$

Jayant's ADM achieves an impressive 10 dB's snr advantage over LDM when both systems are encoding speech with an output bit rate of 60 Kbits/sec.

The other alternative to instantaneously companded algorithms in adapting δ, are the Syllabically Companded (SC) techniques. In a such scheme the quantizers step size δ varies at a much slower rate than the instantaneous variations in the speech signal. The

typical adaptation time constant is about 5 to 10 msec. and consequently δ approximately follows the variations of the speech envelope. The main advantages of such a long-term average adaptation technique, are observed in the presence of channel errors where the encoders show good converging properties and therefore stability.

The Continuous Delta Modulation (CDM)[89] is one of a few, rather typical, Syllabically Companded ADM systems which we are to consider. In the CDM encoder (Figure 2.15) the envelope of the band limited speech signal X(t) (f_{c_1} = 300 Hz, f_{c_2} = 3200 Hz) is extracted through a series combination of differential, rectification and low-pass filtering. The Envelope information EN is added to X(t) so EN resides in the lower band of the resulting signal. It is possible therefore to Delta Modulate this signal and extract the Envelope information in the feedback loop of the CDM encoder using a 100 Hz Low-pass filter. The output of the filter controls the magnitude of the step size δ which now varies slowly with EN.

The SC ADM system of Tomozawa and Kaneko[90] shows that the same slow adaptation in δ can be achieved without the addition of any signal at the input of the encoder. In their scheme (Figure 2.16) the syllabic information is directly obtained from the decoded signal inside the Local decoder and δ is scaled accordingly.

The SC ADM[91] of Brolin and Brown follows a slightly different approach, and the envelope signal is not extracted from the encoder's feedback loop. Instead the system (see Figure 2.17) is composed of two individual DM encoders. The Envelope signal is extracted from

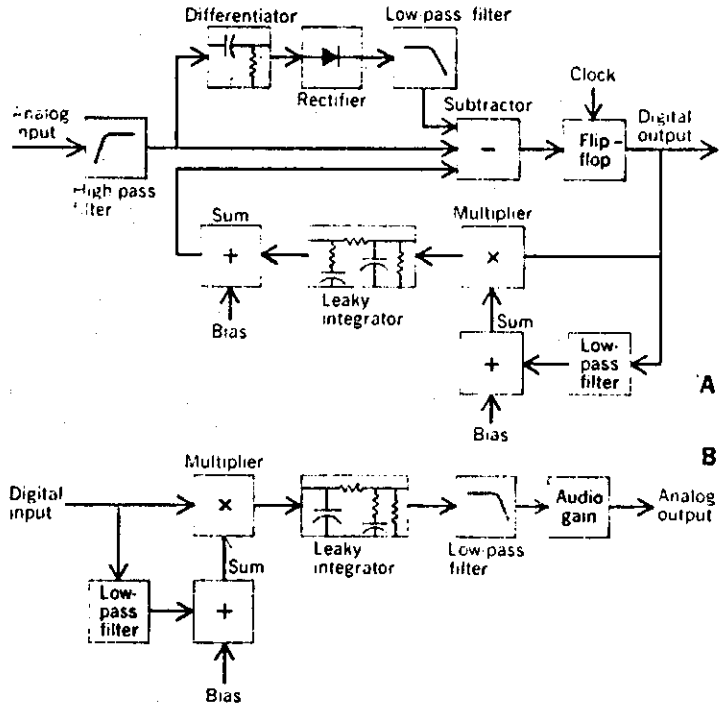


FIGURE 2.15 - The Continuous Delta Modulator (after (108))

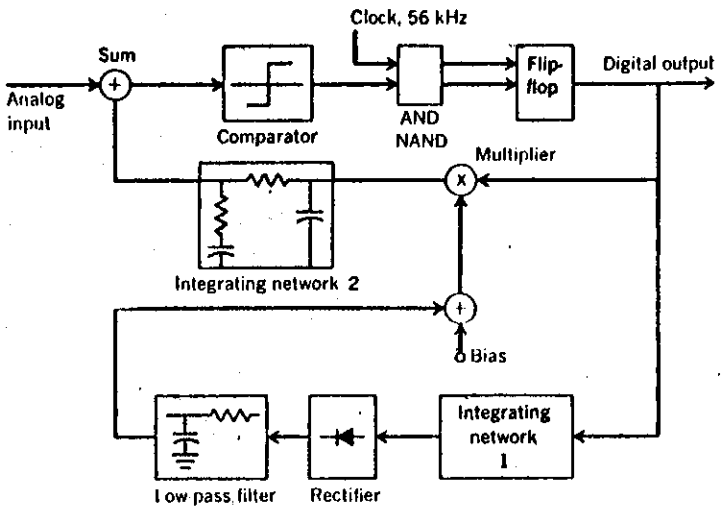


FIGURE 2.16 - Tomozawa's and Kaneko's ADM. (after (108))

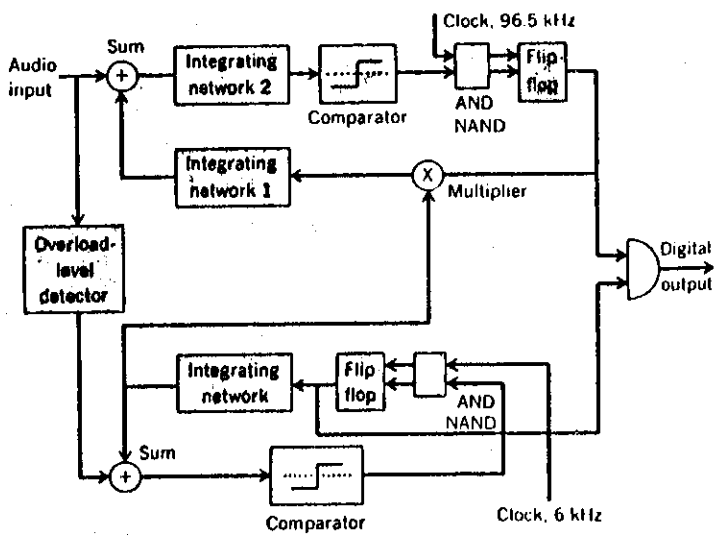


FIGURE 2.17 - Brolin's and Brown's ADM. (after (108))

the input speech and it is encoded with one encoder say DM1 whose decoded output controls the step size δ of the second coder DM2. DM2 is used to encode the speech signal and its binary output is multiplexed with the output of DM1 and transmitted. By doing so the overall transmission rate is not seriously increased as the Envelope signal is composed of very low frequencies and DM2 operates at low clock rates.

An interesting Syllabically Companded ADM is the Digitally Controlled Delta Modulation[92] DCDM where no Envelope detection is required. Instead, a logic detects the presence of four consecutive bits of the same polarity and outputs a pulse to a RC network with a 10 msec time constant. The slow varying signal at the output of the RC controls the value of the step size δ. The performance of this system is satisfactory when working with medium or high output bit rates. However, at bit rates below 16 Kbits/sec. its performance deteriorates considerably as the correlation in the input samples is reduced to a point where decision for scaling δ based on observations at the output bit stream are not particularly useful. In contrast systems like CDM which continuously detect and use the speech Envelope in their adaptation, seem to perform much better at rates below 16 Kbits/sec.

Finally, we mention two Delta Sigma ADM systems successfully used to encode speech signals whose high frequencies are pre-emphasized. The first one shown in Figure 2.18 is called Syllabically Companded All Logic Encoder, SCALE[93a] and its step size adaptation procedure is very similar to one employed in the DCDM system.

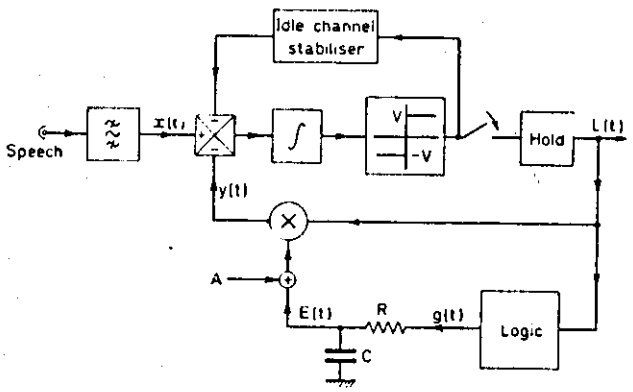


FIGURE 2.18 - The SCALE Encoder
(after (2))

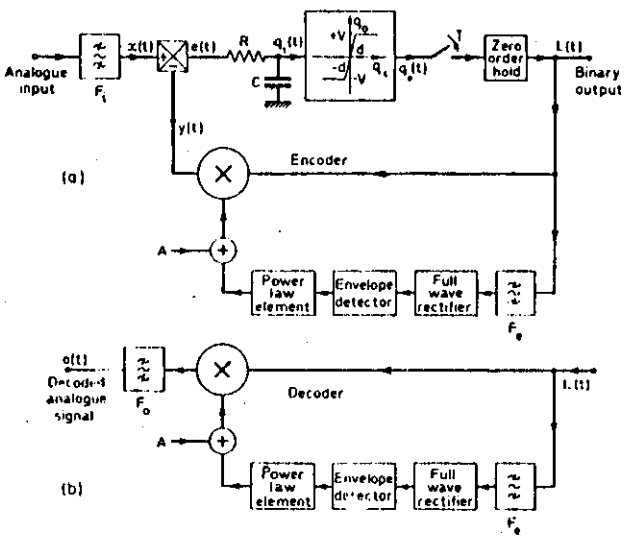


FIGURE 2.19 - The Syllabically Companded DSM
(after (2))

The second Delta Sigma adaptive encoder is known as Syllabically Companded Delta-Sigma Modulation system, SCDSM[93b] and it is shown in Figure 2.19. The Y(t) feedback signal in the SCDSM encoder is scaled according to envelope information extracted from the binary sequence at the output of the quantizer.

2.3.3. Linear Transform Coding.

As the name indicates, Linear Transform Coding (LTC) schemes are based on linear transformation techniques. They have been extensively used in image digitalization rather than speech, but very recently an adaptive LTC scheme was employed successfully in Low-bit rate (16 Kbits/sec.) encoding of speech signals.

A LTC system is shown in Figure 2.20 and operates as follows: A block of N successive input samples X_i, $i = 1,2,\ldots,N$ is processed by the Linear Transform LT to produce a block of N, P_i samples, $i = 1,2,\ldots,N$. These samples are quantized by a set of N quantizers Q_i, $i = 1,2,\ldots,N$ (as shown in Figure 2.20) whose output samples P_1' are binary encoded and transmitted. Assuming that no channel-errors occur during transmission, the recovered P_i' samples at the receiver are processed through an Inverse Linear Transformation ILT to yield an approximation $\hat{X}_i$, $i = 1,2,\ldots,N$, of the N original speech samples. It is obvious from the above description that LT and ILT are the important elements of the system. Consequently a discussion on Linear Transformations is to follow.

Consider an Nth dimensional vector $X = (X_1, X_2, \ldots, X_N)$ whose components are successive speech samples. Let us also assume a Nth dimensional orthonormal vector space A_r, $r = 1,2,\ldots,N$, whose

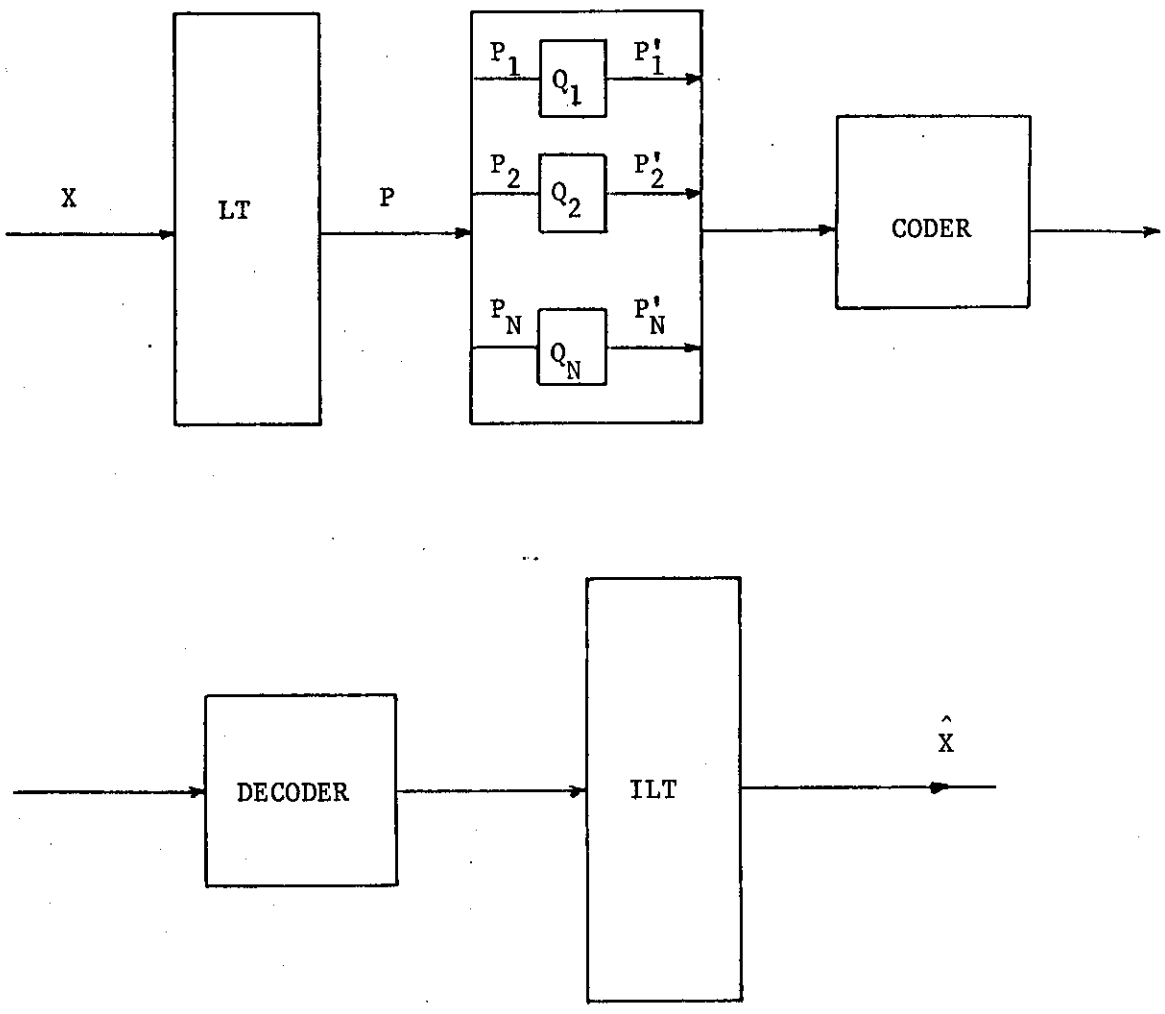


FIGURE 2.20 - Transform Coding.

$$A_m \cdot A_n = \begin{cases} 0 , & m \neq n \\ 1 , & m = n \end{cases} \tag{2.64}$$

The vector X can be expressed as

$$X = \sum_{i=1}^{N} A_i P_i \tag{2.65}$$

where P_j is the component of X along the A_j dimension.

Now because of Equation (2.64), the P_j signal component in the transform domain is:

$$X \cdot A_j = \sum_{i=1}^{N} A_i \cdot A_j P_i$$

$$= P_j \tag{2.66}$$

The last two Equations are in fact employed by LTC systems. The LT operation of Figure 2.20 corresponds to Equation (2.66) solved for all j's, while the ILT operation uses Equation (2.65) and produces the Nth dimensional vector $\hat{X}$ of the $\hat{X}_i$, $i = 1,2,\ldots,N$ recovered speech samples as:

$$\hat{X} = \sum_{i=1}^{N} A_i P_i' \tag{2.67}$$

The success of LTC in reducing the transmission bit rate when encoding speech signals, resides in the fact that the variances of the P_i coefficients are different for the various coefficients. This means that the number of bits assigned for the quantization of P_i can vary with i so that the overall average transmission bit rate is reduced when compared with conventional quantization schemes.

At this point it is natural to ask the following two questions:

a) how to select the optimum N dimensional orthogonal space A, and

b) how to assign in an optimum way the number of bits representing each coefficient, i.e. how to define the optimum number of levels used for each of the N quantizers.

With regard to the second question, it has been shown[94] that in the case of optimum bit assignment R_i, the number of bits needed for the quantization of P_i, is given by:

$$R_i = R_{av} + \frac{1}{2} \log_2 \frac{\sigma_i^2}{\left[\prod_{j=1}^{N} \sigma_j^2\right]^{1/N}} \quad \text{bits/sample} \tag{2.68}$$

where R_{av} is the average transmission bit rate of the LTC and σ_i^2 is variance of the P_i coefficient.

To answer the first question, we have to define a space A which provides minimum distortion D in LTC. A convenient measure of D is defined as:

$$D = \frac{1}{N} \sum_{i=1}^{N} \langle e_i^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} D_i \tag{2.69}$$

where e_i is the mean-square error in the ith sample, and in an optimum bits/sample assignment case, D is given[95] by:

$$D = 2^k \cdot 2^{-2R_{av}} \left[\prod_{j=1}^{N} \sigma_j^2\right]^{1/N} \tag{2.70}$$

k is a constant.

Now for any N dimensional orthonormal space A we have[96]:

$$\prod_{j=1}^{N} \lambda_j \leq \prod_{j=1}^{N} \sigma_j^2 \tag{2.71}$$

where λ_i is the ith eigenvalue of the speech covariance matrix.

From Equations (2.70) and (2.71) we see that the optimum space A should satisfy the following relationship

$$\sigma_j^2 = \lambda_j \tag{2.72}$$

The space which shows the above property is known as the Karhunen-Loeve A_{KL} space and is a special set of orthogonal basis vectors composed of the eigenvectors of the speech covariance matrix. These eigenvectors $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$ are ordered into a sequence such that a $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ monotoneous decrease of the corresponding eigenvalues is obtained.

Thus the Karhunen-Loeve (KL) transform offers the best transform performance in LTC. It removes large amounts of redundancy from the input samples which leads to small values of σ_i^2 and to better quantization of P_i. In fact the P_i coefficients are linearly uncorrelated and the differences between the σ_i^2 variances are proportional to the increase in the correlation of the input samples. The KL transform suffers however, from two serious disadvantages, a) A_{KL} is signal dependent, and the computation of the A_i vectors is not a simple task, b) no fast algorithms are available for the computation of the P_i coefficients. In contrast other orthogonal spaces such as Discrete Fourier (DF)[97], Discrete Cosine (DC)[98], Walsh-Hadamard (WH)[99], and Discrete Slant (DS)[100], are not

optimum, but they are signal-independent and have fast computational algorithms.

Campanella and Robinson[101] examined the DF, WH, and KL transforms in LTC coding of speech signals. For N = 16, using Log-quantizers and when R_i, $i = 1,2,\ldots,N$ is calculated from the long-term statistics of the speech signal, their computer simulation results indicate an approximate gain over Log-PCM of 9 dBs, 6 dBs, and 3 dBs for the KLT, DFT, and WHT schemes respectively.

P. Noll[96] modelled speech by a tenth order Markov process whose first ten autocorrelation coefficients are equal to the first ten long-term autocorrelation coefficients of speech. Then by using the formula

$$G_{LTC} \simeq \frac{\sigma_x^2}{\left[\prod_{i=1}^{N} \sigma_i^2\right]^{1/N}} \tag{2.73}$$

which defines the gain of LTC over PCM, he obtained very similar results with Campanella. In particular with N = 16 the gain over PCM for the KLT, DCT, DFT, DST, WHT, Linear Transform Coding systems are 8.0, 7.8, 6.0, 4.5, 2.3 dB's respectively. Furthermore, the G_{LTC} results for various values of N show the WHT and the DST to be tru ly suboptimum transform for speech, with no substantial improvement in the gain for large value of N. For example, when N = 128, the gain G_{LTC} of WHT and DST is only 3 and 4 dBs, while the gain for KLT is 9.5 dBs, DCT is marginally inferior and DFT is about 2 dBs worse.

Finally, Frangoulis and Turner[102] examined the perceptual

effect of encoding and transmitting a limited number of coefficients of a N = 64 WHT scheme. Their system employed the same number of quantization levels in quantizing P_i and showed that very good quality speech can be recovered by transmitting only 8 dominant transform coefficients with an average bit rate of 17.55 Kbits/sec. These dominant coefficients are found from the probability density function of the Hadamard coefficients.

Adaptive LTC.

Adaptive LTC systems achieve an improved encoding performance over the previously mentioned non-adaptive ones. There are three elements in LTC which can be made to adapt to the statistical variations of the input speech signal.

a) The amplitude range of the N quantizers used to quantize the P_i coefficients. It can vary proportionally to the variance of the input signal. That is, N adaptive quantizers can be used to compensate for the changing levels of speech sound.

b) The number of bits R_i assigned for the quantization of each coefficient. R_i can vary according to the short-term statistics of speech, by recalculation of its value for each input block of samples.

c) The orthogonal vectors of the A_{KL} space. When the KL transform is employed in the system, the A_i vectors can be updated by calculating the covariance matrix i) for different speech sounds, ii) for each block of N input speech samples.

Only a few speech adaptive LTC systems have been proposed[96,103].

Modena[103] employed adaptive quantizers in his LTC scheme. Noll[96] showed that an AQF-LTC system, using feedforward variance estimation techniques for the adaptation of N Laplacian quantizers, provides an additional 4 dB gain over non-adaptive Log-LTC. He also proposed a fully Adaptive Discrete Cosine-LTC system where the quantization as well as the bit assignment procedures are adaptive. The choice of the DC transform is based on its nearly optimum performances and its independence to signal statistics.

The Adaptive DC-LTC system shows a 4 dB improvement over AQF-LTC and at 16 Kbits/sec. produces better quality speech than a 16 Kbits/sec. ADPCM system.

2.3.4. Other Waveform Coding Systems.

The speech encoding systems mentioned so far belong to one of the four basic waveform coding techniques, i.e. PCM, DPCM, DM and LTC. However, other systems have been developed which combine characteristics from the above techniques and new strategies specially conceived for Low-bit rate encoding of speech.

An example of a such strategy is the interruption/reiteration technique used to exploit the quasi-periodic nature of voiced speech. In its simplest form the encoding of the input signal is interrupted at a constant rate and the transmitted binary data corresponds only to segments of the speech waveform. At the receiver, the decoder reconstructs these segments while a reiteration procedure attempts to restore the signal's continuity by repeating the decoded parts of the waveform. Although the intelligibility of the produced speech can be as high as 85% its quality is very poor. This is

mainly due to the constant interruption/reiteration rate which results considerable distortion in the speech fundamental frequency.

The obvious way to improve the quality of the speech is to incorporate in the system a pitch synchron ous interruption procedure, and three such systems have been proposed[104,105,106]. The most sophisticated is the Speech-Reitetion DM developed by Baskaran[105] which provides acceptable quality speech at a transmission bit rate of 10 Kbits/sec. The encoder used in the system is an Adaptive DM which encodes every other pitch period of the voiced speech waveform. Its adaptation strategy exploits the presence of the Pitch Extractor Circuits, PEC, (which controls the interruption process during voiced sounds) and allows the quantization step size to increase at the beginning of each pitch period by ten times its minimum value and to exponentially decrease afterwards with a time constant of about 8 to 10 msec. When unvoiced speech is detected by the PEC, the interruption of the low amplitude high frequency speech waveform is performed randomly in order to avoid a line spectrum occurring in the decoded signal, while the DM encoder behaves as a LDM. The binary information transmitted to the receiver includes, in addition to speech data, synchronizing data for voiced/unvoiced decisions and pitch period lengths. The receiver decodes the voiced/unvoiced segments of speech while the synchronization bits are used by the reiteration procedure to reform the original speech.

Another coding technique to mention in this section is the Sub-Band Coding (SBC). In SBC the speech spectrum is first

partitioned into frequency sub-bands according to perceptual criteria (for example, equal Articulation Index for the sub-bands) and then each sub-band is sampled at a different sampling rate and digitally encoded. Furthermore, in some SBC systems, the sub-bands are Low-pass translated before encoding so the benefits of encoding Low-frequency signals are obtained. The SBC techniques have also the advantage of restricting the quantization noise in discrete frequency bands and therefore masking of various frequency ranges by quantization noise produced from different frequency range signals, is avoided. This leads to perceptually less annoying quantization noise and consequently to good quality speech at transmission bit rates as low as 16 Kbits/sec.

The last system to mention is the 4.8 Kbits/sec., 1 bit PCM developed by Wilkinson[107]. Although the encoder employs a two level quantizer together with a ADM, it is basically acting as a two level adaptive quantizer. The input signal is channelled into two separate paths. In the upper path the signal is sampled at the Nyquist rate of 4.4 K samples/sec. and the polarity of the resulting samples is obtained with a two level quantizer. The speech signal in the lower path is full wave rectified and its envelope is obtained with a 5 mS RC circuit. This low frequency envelope signal is encoded by an ADM whose output bits (400 bits/sec.) are multiplexed with those at the output of the quantizer and transmitted. The receiver after de-multiplexing uses the polarity and envelope data to control a Pulse Amplitude Modulator whose output is an approximation of original speech.

CHAPTER III

THE H.P. 2100A MINICOMPUTER BASED SPEECH PROCESSING SYSTEM

3.1 INTRODUCTION.

The signal-to-noise ratio (snr) measurement is accepted by many research workers (62,68) as a meaningfull method of evaluating the performance of an encoding system. This is because snr is related to the subjective quality of the decoded signal provided that the transmitted bit rate is higher than approximately 16 kbits/sec. In this thesis the snr criteria is used extensively in the computer simulation studies, and various systems are evaluated by encoding speech segments of duration of 1.5 to 2. seconds. Although these durations are often adequate, there are occasions when longer intervals of speech signals are required in order to highlight the wide variety of the signal's characteristics. To achieve this, the H.P.2100A computer speech processing system was developed.

In this system the computer is interfaced to the external analogue speech signals by means of an Analogue-to-Digital (ADC) and a Digital-to-Analogue (DAC) converters. The combination of this hardware with two H.P.7970E Magnetic tape units, enables digitized speech of up to 10 minutes duration to be stored. The stored speech is used as the source material in the various codec simulations. The decoded date is also stored on magnetic tape and is subsequently removed through the DAC to the loudspeaker.

In developing the system's software special emphasis was given to the production of a computer operating system built on a modular

basis with basic routines. Using these routines transfers between the computer and the Magnetic tape or ADC - DAC peripherals, and manipulations of speech signals can be handled by any person having a knowledge of basic Fortran programming. Hence the system is not only a convenient and powerful tool for the author's own research but should also be useful to other research workers.

Sections 3.2 and 3.3 deal with the hardware and software realization of the speech processing system respectively. In section 3.4 the parts of the present system that could benefit from modifications are discussed and suggestions are made for some possible additions.

3.2 HARDWARE DESCRIPTION OF THE COMPUTER INTERFACE WITH THE ADC, DAC PERIPHERALS.

The Electrical Engineering Department's H.P.2100A computer is a 24k memory, 16 bit word, compact data processor. Standard features include memory parity generation and checking, memory and input/output protections for executive systems, extended arithmetic capability and power fail interrupt with automatic restart. Optional features include two channel Direct-Memory-Access (DMA, see sections 3.2.1., 3.2.2.) and multiplexed input/output.

Interfacing of peripheral devices is accomplished by plug-in interface cards. The external device is connected by a channel in a form of cable through which data and control signals pass to an interface card, which in turn plugs into one of the computer's input/output slots. Each slot is assigned a fixed address, and the computer can communicate with a specific external device on the

basis of its address. The address is termed as the "Select Code". The computer mainframe can accommodate up to 14 interface cards, expandable to a total of 45 when an input/output Expander is used. All the input/output channels are buffered and bi-directional and are serviced through a multilevel priority interrupt structure, as described in the subsequent section.

3.2.1. Input/Output Data Transfer.

In an input/output operation, data is transferred between the computer memory and an external device through the A,B registers or the DMA hardware as shown in Figure 3.1.

The commands required in the program for the communication between the computer and the external device are simply the start device (control set), the device busy (flag clear), the device operation completed (flag set), and the stop device (control clear). A general block diagram of the computer interface with an external device is illustrated in Figure 3.2. The data receivers and drivers are used for buffering purposes.

A. Input data transfer.

The control of the input operation is achieved through a program which has been previously inserted into the computer. To connect a particular peripheral to the computer the program addresses the interface card associated with this peripheral. The program instruction STC X, C i.e. Set Control, Clear Flag initiates the input of the 16 bits of data from the input device. The instruction sets the Control F.F. and resets the Flag F.F. In addition to that it sets the Command F.F. which applies a Command signal to

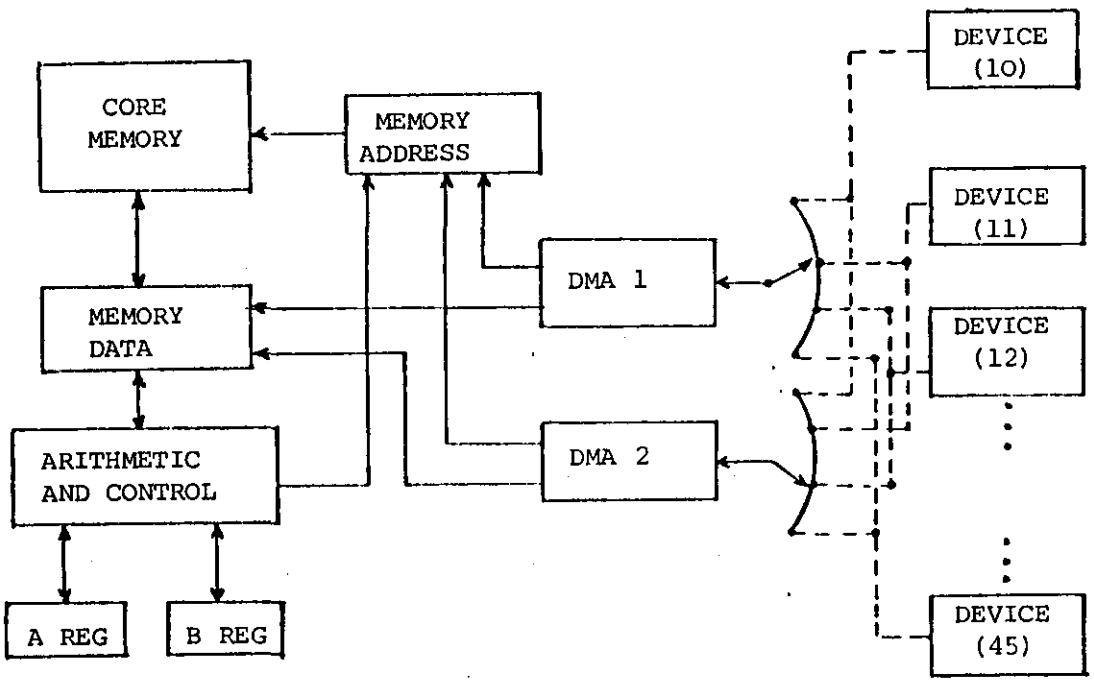


FIGURE 3.1 - Input/Output Data Transfers.

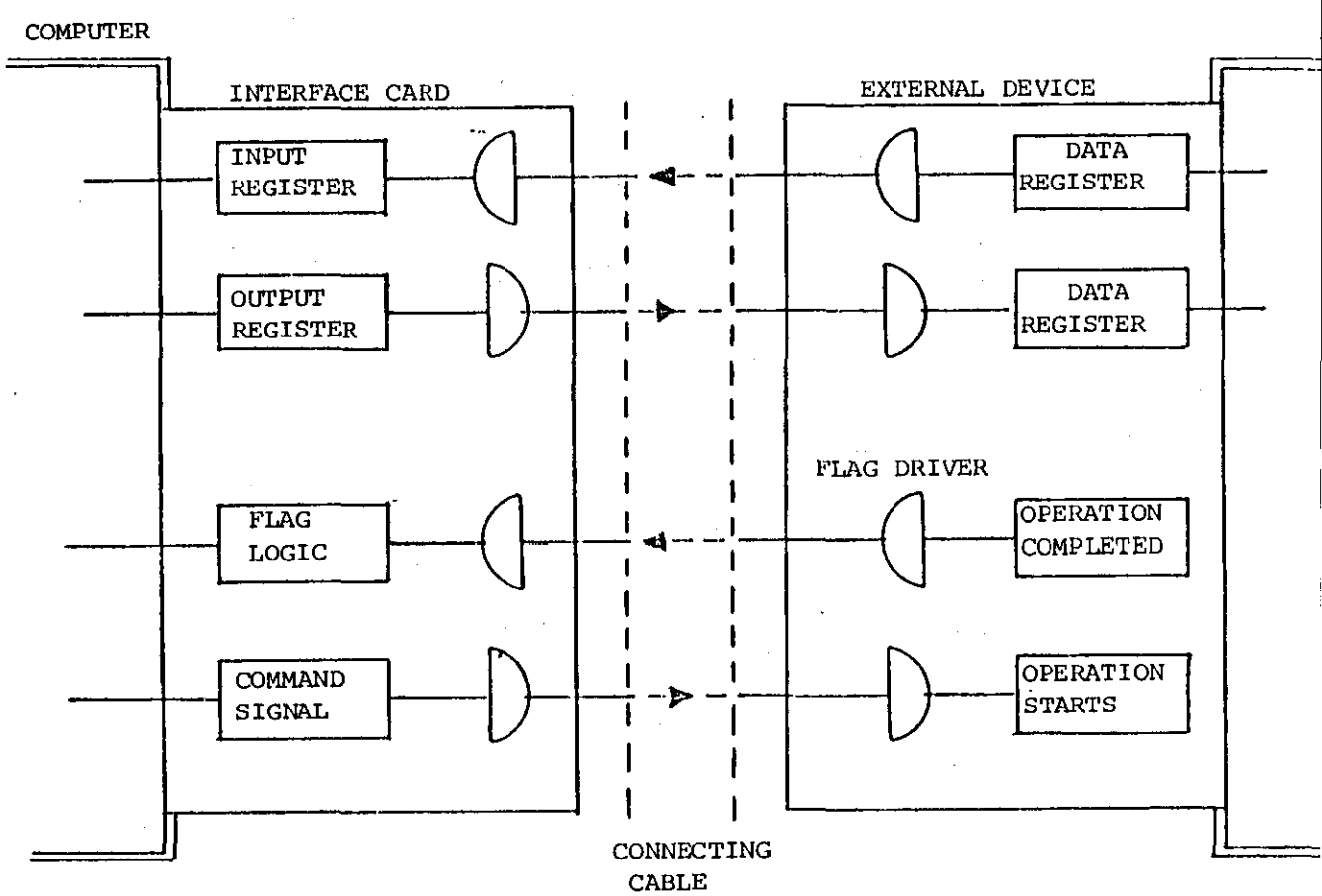


FIGURE 3.2 - Interfacing the Computer with an External Peripheral.

the device initiating its operation (see Figure 3.4.). The data bits 0 to 15 are placed into the interface register and the Command F.F. is reset after a data flag signal is applied to the interface by the external device (see Figure 3.3.). This signal also informs the control logic of the interface card that the input data is available to the computer, by setting the Flag F.F. As a next step the interface is to interrupt the computer which is to accept the input data. Provided that the interrupt conditions are met i.e.

a) the interrupt system in the computer is on,

b) no higher priority interrupts for other interface cards are requested,

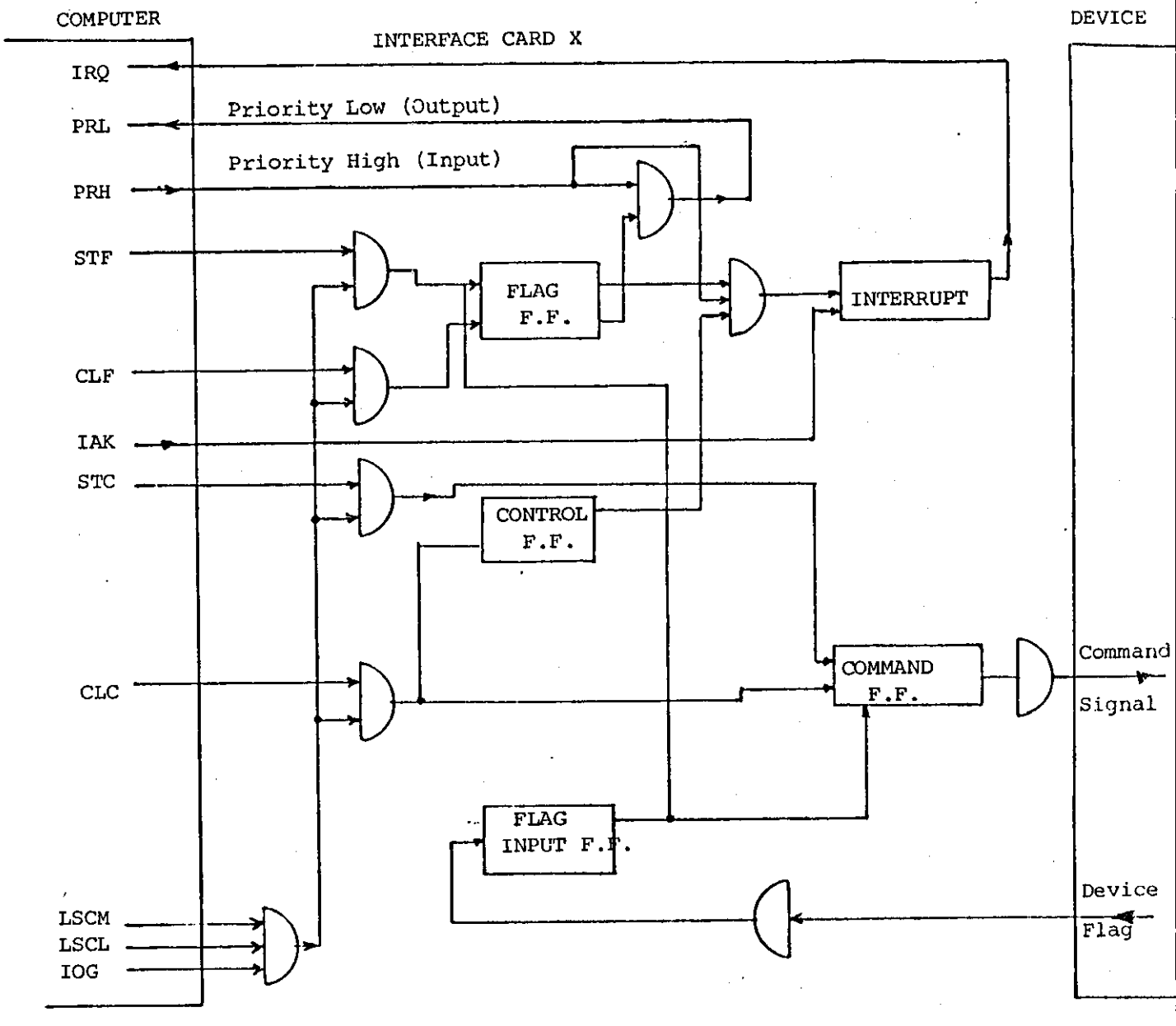
c) the Control and Flag F.F. are set (see Figure 3.4.), an interrupt signal IRQ to the program control is generated.

This causes the current computer program to suspend its operations and control is transferred to a service input subroutine which includes the LIA or LIB instruction for loading the data into the A or B register (Figure 3.1.)

Specifically, the LIA or LIB instruction addressed to the select code of the X interface card (Figure 3.3.) enables the address LSCM, LSCL and the IOG, IOI lines and the data is transferred into the computer via the IOBI lines.

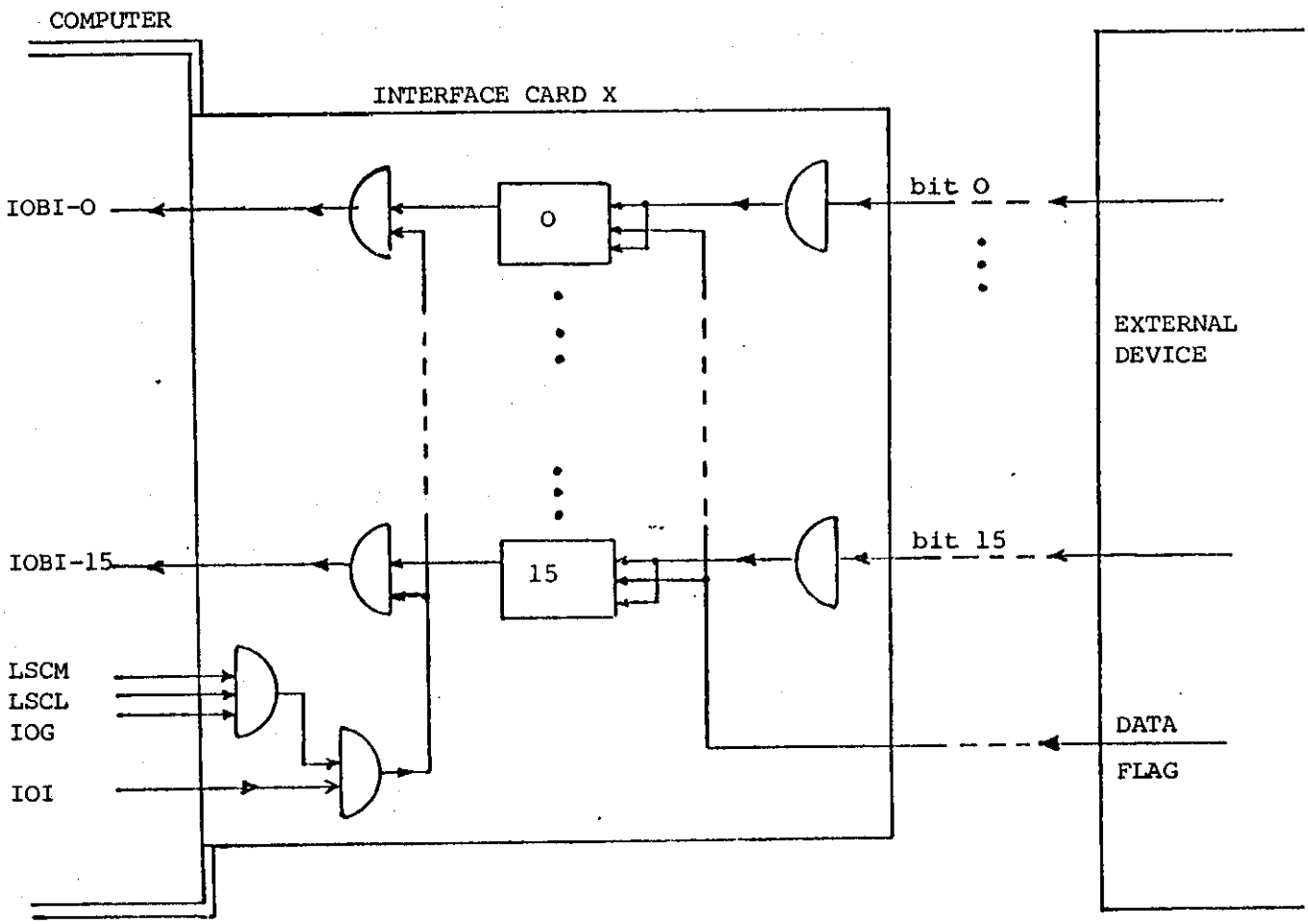
B. Output data transfer.

An output operation similarly is initiated with a programmed output instruction OTA X (or OTB X). The address lines LSCM, LSCL and the IOO, IOG lines of the X interface card are enabled and the 16 bits data after transferred from the A or B register via the



IRQ	interrupt request.
PRL	inhibits transfers for devices with lower priority.
STF,CLF	set flag, clear flag.
IAK	reset interrupt flip-flop.
STC,CLC	set control, clear control.

FIGURE 3.4 - Commands to the Control Logic of the Interface Card.



LCSM, LSCL' are the address lines enabled by the computer when the X interface card is selected for input operation.

The IOI,IOG lines are enabled when an input programmed instruction LIA or LIB is issued to the X interface card.

FIGURE 3.3 - Input Data Transfer.

IOBO lines into the interface buffer is available to the device (see Figure 3.5.). The Set Control, Clear Flag STC, XC instruction which follows, sets the Command F.F. This issues a Command signal to inform the external device that the data is available for transfer. The computer program is suspended by an interrupt when a "done" device flag is returned to the interface card. Control then is transferred to a service subroutine where further OTA X, STC X, C instructions for additional data transfers can be issued.

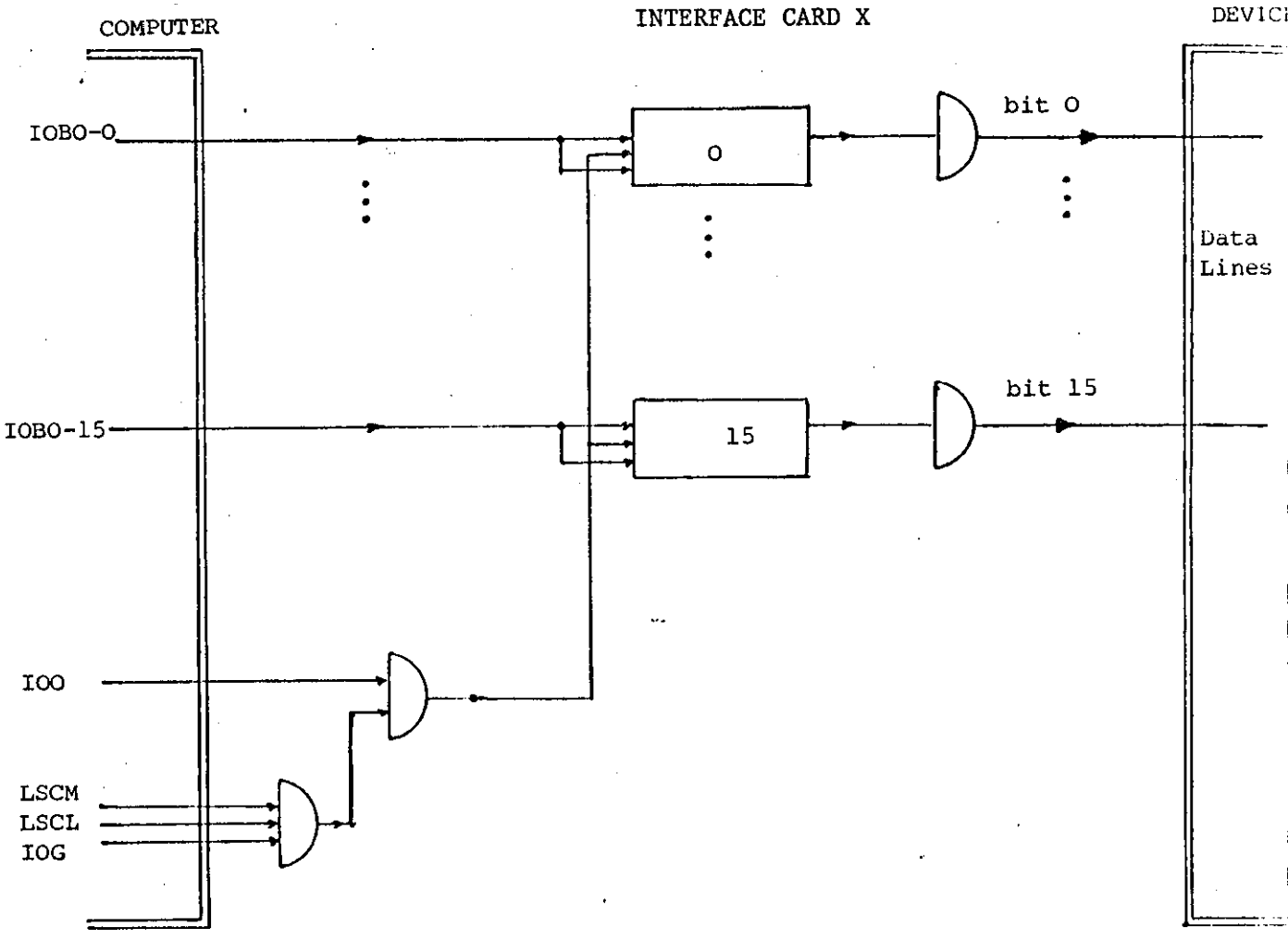
In the case where the Direct Memory Access option is used in an input/output operation, the data is transferred directly between the memory and the high speed peripheral via the interface cards, without the Arithmetic and Control logic and A and B registers of Figure 3.1 being required. By this method a transfer rate of data up to 1.020.400 16 bits words per second is achieved.

Finally the input/output priority given by the computer to the various external devices is established along a "line", where the priority given by the computer to communicate with a particular peripheral decreases progressively down the line. A device in the process of transferring data essentially breaks the line disabling all the devices with lower priority.

C. Input operation.

Considerations will now be given to the transference of speech signals into the computer.

The analogue speech signal after being sampled and held as shown in Figure 3.6., is converted into digital form by the ADC. The 10 bit data words at the output of the ADC device are inverted by the "driver" NAND gates and the logic used in the interface



OTA
OTB } Output Data Commands.

The IOO,IOG Lines are enabled when an OTA or OTB is us ed to the X interface card.

FIGURE 3.5 - Output Data Transfers.

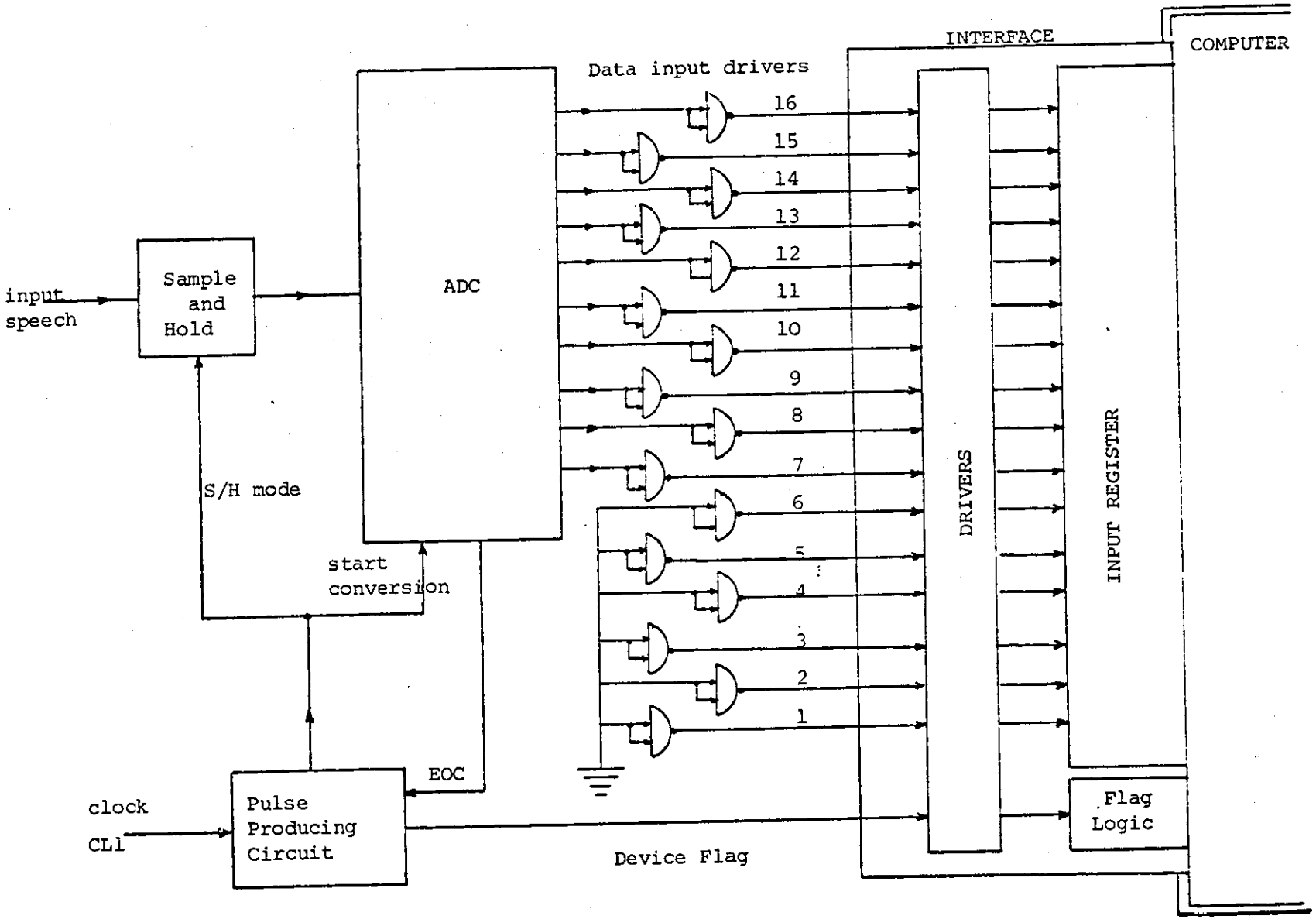


FIGURE 3.6 - Transfer of Speech Data into the Computer.

card is ground true logic and thus the computer accepts the data in the same state as it appears at the input of the NAND gates. In order to match the 16 bit computer word with the 10 bits ADC output, the six less significant input lines of the interface card are made zero.

Let us suppose that a 10 bits digitized speech sample appears at the input lines of the number 22B interface card. An input operation starts by programming a STC 22B, C instruction as described in the previous section. The Clear Flag portion of the instruction resets the Flag F.F. of the interface card to prevent any interrupt signal from being sent to the computer before the ADC device has transferred the data into the interface input register. The interface card is now able to accept the speech data on receipt of the response-in Flag pulse. This pulse is related to the clock waveform CL1 whose frequency is the sampling frequency of the speech signal, as follows:

From the positive going edges of the CL1 waveform positive true pulses of 4 μsec. duration are produced. Those pulses are used as the mode control signals in the Sample and Hold device and also as the "start conversion" signal of the ADC. When the ADC starts its operation, the Sample and Hold device is already in the hold mode and the correct conversion is performed. At the end of the conversion time the ADC produces an End-of-Conversion signal (EOC) which is shaped as a pulse of 1.5 μsec. duration. This pulse forms the response-in Flag signal which enters the speech data into the input interface register and sets-up interrupt request for service. The computer responds to the interface card with an

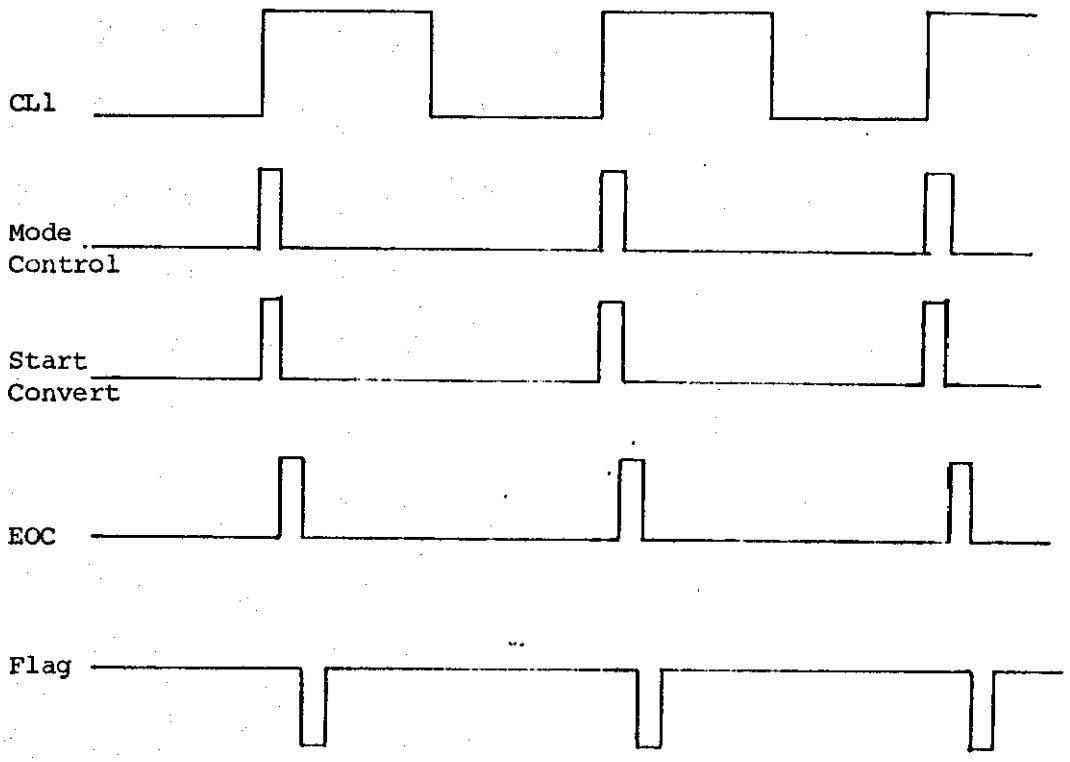


FIGURE 3.7 - Timing Diagram of Input Operation.

input instruction LIA 22B or LIB 22B that enters the speech data into the computer, and waits for the next response-in Flag pulse indicating that further data is ready for input.

The rate at which the computer accepts data is determined by the frequency of CL1 as it is shown in the timing diagram of Figure 3.7.

D. Output operation.

To transfer data from the computer's A or B registers into the interface card output storage register, an output instruction OTA 22B or OTB 22B is programmed. From the 16 bit word at the output lines of the interface card, the 10 most significant bits represent the speech data. These bits are inverted by the data output drivers and fed into the input of a D-flip-flop buffer as shown in Figure 3.8. The next instruction to follow in the program is a STC 22B, C i.e. a Set Control. Clear Flag one which prepares the interrupt logic of the interface card to suspend to computer program when device Flag is received. A device Flag pulse then

a) clocks the D-buffer and the 10 data bits are presented to the DAC device and

b) sets the Flag F.F. of the interface card so an interrupt to the computer's program occurs. In this way control is transferred to a service subroutine for issue of further OTA and STC, C instructions.

The device Flag pulses are of duration 1.5 μsec and they are obtained from the positive-going edges of the CL1 clock waveform. Consequently the rate with which the data bits are presented to the DAC device is equal to sampling rate of the speech waveform.

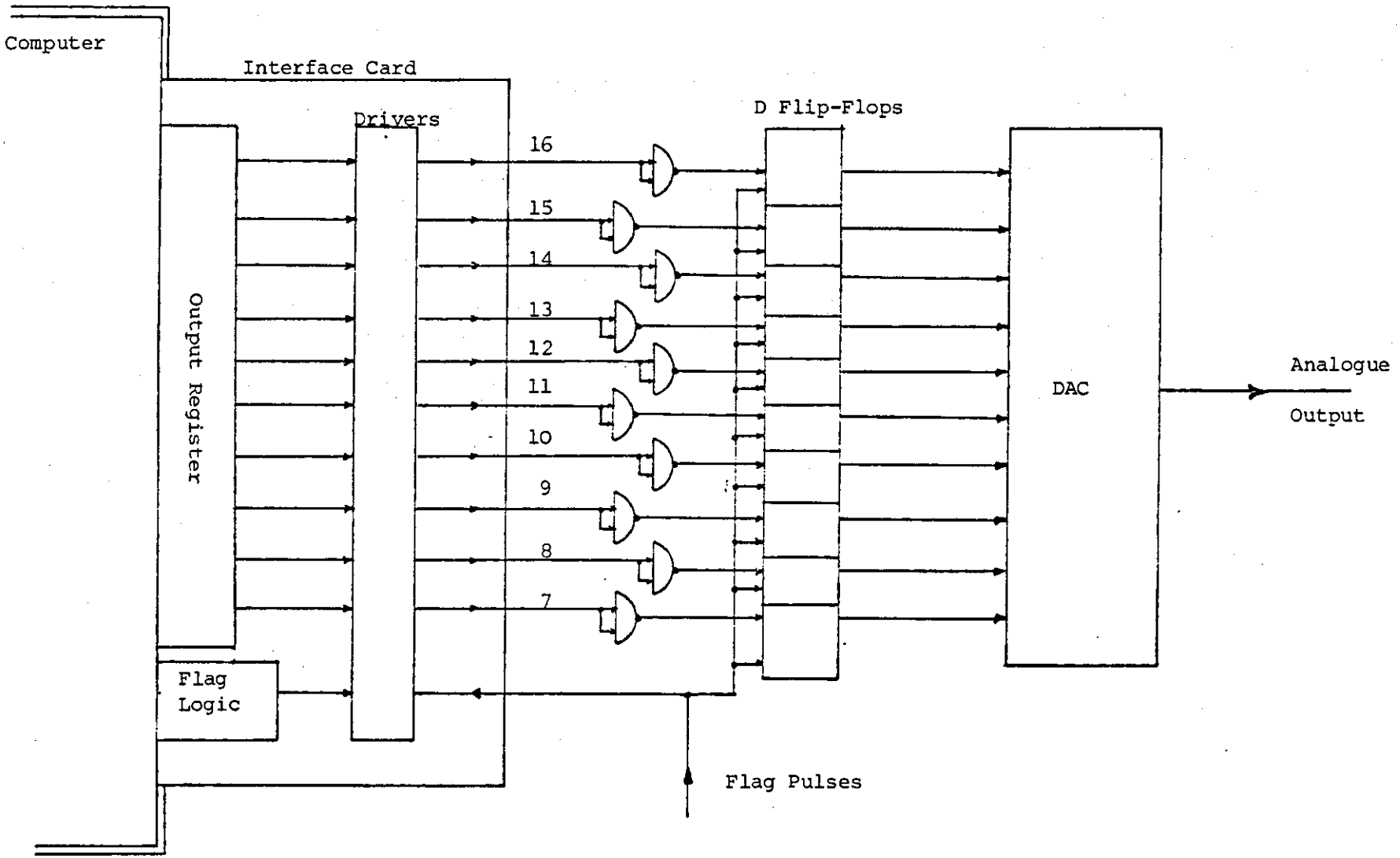


FIGURE 3.8 - Transfer of Speech Data to the DAC Device.

3.3 DESCRIPTION OF THE SOFTWARE CREATED TO SUPPORT THE SPEECH PROCESSING SYSTEM.

The computer operating system used in connection with the H.P.2100A speech processing system is called the Basic Control System (BCS). BCS is a paper tape based system which provides an efficient loading, linking and input-output control capability for relocatable programs produced by the HP Assembler or HP Fortran. The Basic Control System is modular and has two distinct parts, namely: the input/output subroutines and the relocating loader.

The input/output software package consists of an HP input/output control subroutine (IOC) and the BCS driver subroutines which controls the peripheral devices. When the program is written in Assembler the input/output operations are specified by a symbolic calling sequence. In Fortran programs the requests for "READ" or "WRITE" are translated by the compiler and with the aid of the subroutine "FORMATER", the proper calling sequence is produced.

When the user requests an input/output operation using a logical unit reference number, the IOC subroutine finds the logical unit entry in the equipment table (a memory table created at BCS configuration time) which contains the addresses of the drivers and the physical channel number of the external devices. The IOC directs the request to the proper driver, and the input/output operation is initiated. The BCS driver transfers control back to the main program which continues operation until the input/output device completes a single operation. At that time an interrupt request is generated which causes the transfer of the control back to the BCS driver. The data is now transferred between the

device and the specified memory buffer and the input/output device is commanded to another operation. This process continues until all the data has been transferred, when a "completed operation" status is produced by IOC and it is checked by the main program.

The task of the relocating loader is to load and link into the memory the object code programs (i.e. machine language program) produced by the HP Assembler or Fortran compilers. The loader has the ability to assemble the main program as a set of subroutines which are linked together through program entry points and external reference instructions. This allows design and test of each of the subroutines separately and execution of all in one program. The loader also allows the program to be designed without concern of page* boundaries, as indirect addressing is produced automatically. The indirect addressing occurs when a memory location in which the instruction is referred, is not on the same page with the instruction. An optional feature of the loader allows the production of absolute** paper tape version of a relocatable program plus the BCS and those library subroutines that were referenced in the main program. The process of generating the absolute program is such that core memory allocated normally to the loader may be occupied by the program instructions.

The standard Hewlett Packard software package which produces an absolute version of the BCS is called the Prepare Control System. During the construction of the absolute BCS the relationship among input/output channel number, drivers, interrupt entry points in

---

* The computer memory is logically divided into pages of 1024 words each.

** An absolute program can be loaded directly into core memory.

the drivers and unit reference numbers, is established.

The input/output devices included in the Basic Control System configured to be used with the speech processing system are, a teletype, a photoreader, a punch, and two magnetic tape units.

3.3.1. Speech data handling subroutines.

In order to transfer speech signals into or out of the computer, routines supporting the ADC and DAC peripherals are required. There are two possible modes of operation between those two external devices and the speech data processing, namely "Synchronous" or "On line operation" and "Asynchronous" or "Off line operation".

In the first mode relatively uncomplicated speech data processing can be performed synchronously with the incoming input speech signal. This is provided that the time required for the data processing and, or, the time necessary to obtain an analogue output through the DAC, is less than one sampling period of the input signal. The advantage of this method is that there is no need for extensive data storage. Also, complicated processing requirements outside the real time capabilities of the computer, can in principle, be handled by means of an FM tape recorder which slows down the input data rate.

However the "On line operation" appears to be inconvenient for the following reasons:

(1) The processing time for each input sample may be different and it depends on the number and type of operations required by each sample. Consequently when the computer is operating in an on-line mode, the rate at which the samples are fed to the computer is

dependent on the longest processing time required by a particular sample(s). Off-line operations is not bound by this restriction and hence the processing time is faster.

(2) The need of using the same input material more than once in various experiments creates problems. Two sampled waveforms produced from the same analogue speech material in two separate computer runs, cannot be identical, due to differences of the starting point, the slight changes in the sampling frequency and the amplifiers gain. Supposing that the signal-to-noise ratios of two different encoding methods are to be compared with this slightly different input data, then the validity of the snr results may be suspect.

(3) For every experiment a laborious procedure has to be followed. This means that the speech input level has to be adjusted so that the signal occupies the quantization range of the ADC and produces a maximum snr. The d.c. drifts of the amplifiers have to be compensated correctly, the sampling frequency has to be adjusted, etc.

Because of these disadvantages the speech handling routines were designed for "Off line operation". Using these routines the speech material is stored permanently on digital magnetic tapes, and when required it is transferred directly into the computer's core memory. After processing the speech data it can be stored again on the magnetic tape, from where it can be transferred through the computer's core memory into the DAC peripheral for listening evaluation.

In a such mode of operation the computer's core memory is

occupied with the operating system, the main program instructions, and the "storage buffers" which are needed for the data transfers between computer and peripherals as described in details later.

The transfer of the speech data between the core memory, the ADC, DAC, and magnetic tape units is done by using the Direct Memory Access option (DMA). This option is employed because Direct Memory Access has the capability of handling data extremely fast with minimum programming requirements. It is therefore useful to describe, in general terms, the operation and the programming considerations of the Direct Memory Access before presenting the speech handling routines.

In order for the DMA to operate it must be programmed to know

a) the direction of data transfer,

b) where in the memory the data is to be placed or removed,

c) which input/output channel is to be used for the data transfer, and

d) what is the amount of the data to be transferred.

This information is given by means of three control words which must be addressed directly to the DMA hardware. Specifically:

Control word 1 (CW1) identifies the input/output channel to be used and provides the options of STC (set the Control flip-flop) or no STC at the end of each DMA cycle, and CLC (Clear Control flip-flop) or no CLC at the end of each block transfer for the particular input/output channel under consideration.

Control word 2 (CW2) provides the starting memory address for the data block to be transferred, and defines whether the data is to go into, or out of the memory.

Control Word 3 (CW3) defines the number of data words to be transferred into, or out of the memory.

For the initialization of the DMA channel 1, the CW1 control word is loaded into a Service Select Register, in the DMA circuitry, with an OTA6 instruction. A programmed CLC2 instruction clears a Register Load Control Flip-Flop and activates the Memory Address register and an input/output Flip-Flop. Then the CW2 word is stored into DMA by an OTA2 instruction. An STC2 instruction prepares a Word Count register to receive the CW3 word which is then outputted to the DMA hardware by another OTA2 programmed instruction. The last step is to activate the DMA channel with an STC6 instruction. For initializing the second DMA channel, the select code 2 has to be replaced by 3 and the select code 6 by 7.

Once the DMA operation is initiated no additional programming steps are required until the end of the transfer of the data block is reached. Then if the interrupt system is enabled, an input/output interrupt to the DMA channel address 6 or 7 occurs. The interrupt location normally contains a jump to a completion routine instruction (JSB) and the program control is forced to this routine, the contents of which varies according to the specific application. When the interrupt system is disabled it is possible to check for completion of a block transfer by testing the status of the flag in the select code 6 or 7 depending upon the DMA channel used.

A generalized block diagram of the DMA hardware is shown in Figure 3.9. Under the program instructions cards 1 and 2 perform the switching functions to connect the DMA channels with any peripheral device controlled by the computer. The timing logic

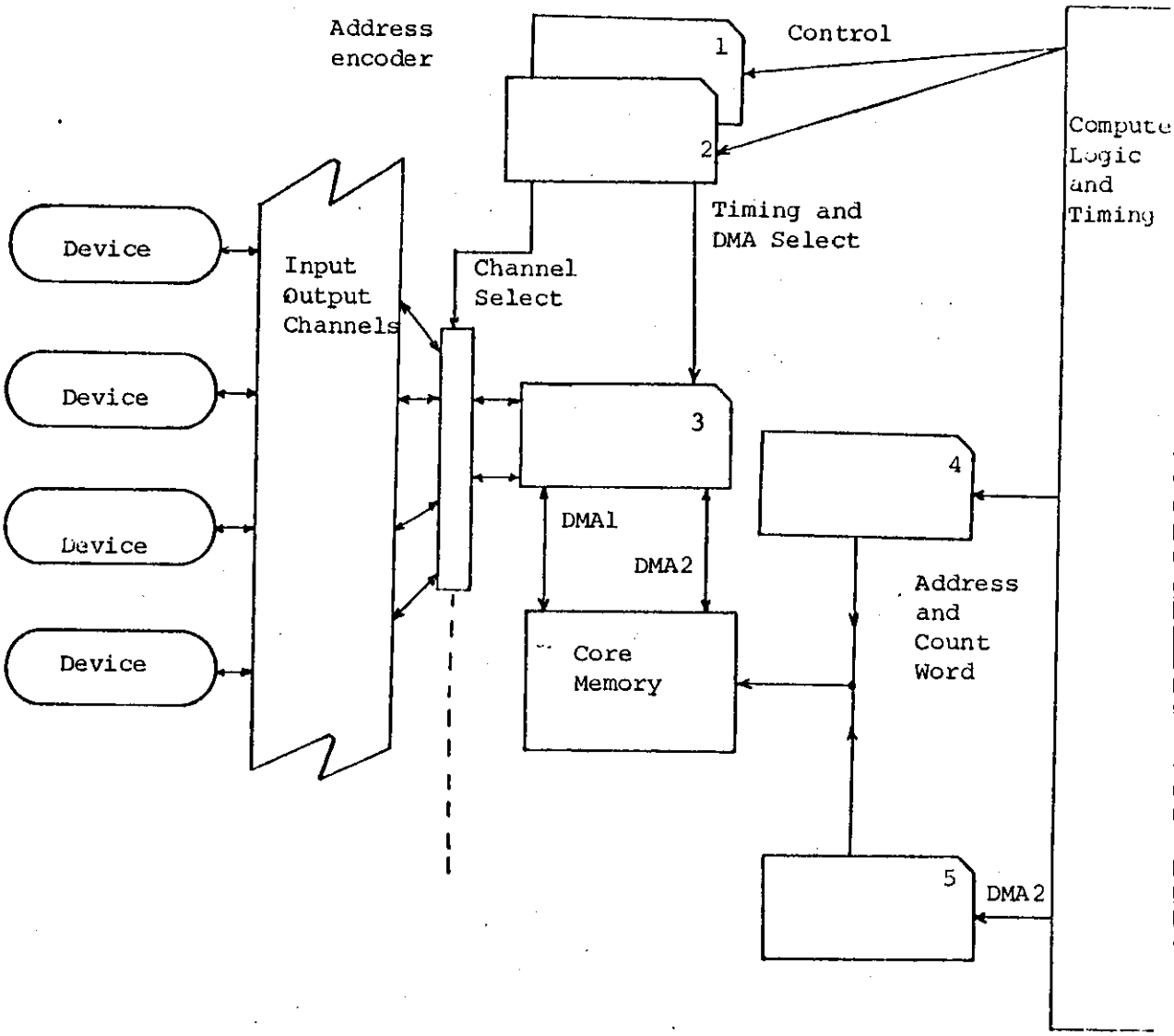


FIGURE 3.9 - DMA Hardware.

in these cards enables the DMA not to interact with the central processor operations. A priority interrupt logic is also included in these cards. Card 3 contains a storage register and logic for storing data while the starting memory address of the data with its length and the direction of the transfer is stored in cards 4 and 5. A word count register in these cards determines for the DMA controller of card 1, when the block transfer is completed.

Having considered the operation and the software requirements of the Direct Memory Access Option, the speech handling routines are discussed next. All the routines are in the form of subroutines and are called from the Fortran main program. The subroutines are written in Assembler language and their object version is loaded and linked with the main program using the Relocatable loader of the BCS operating system.

Specifically, in the Fortran program the Assembler written subroutine is called by the statement CALL X (a_1,a_2,...,a_n) where X is the name of the subroutine, and a's are the actual arguments. As a result of this call in the main program, the following calling sequence is generated by the Fortran Compiler.

```
JSB    X       transfer control to subroutine X
DEF    *n+1    define return location
DEF    a1      define address of a1
 :      :
 :      :
DEF    an      define address of an
```

The words from the locations listed in this calling sequence are then accessed and transferred to the subprogram under the supervision

of the .ENTR Fortran library subroutine. The .ENTR subroutine moves the addresses of the arguments into a reserved area within the Assembly Language subprogram, performs all the testing to determine if the locations given in the calling sequence are direct or indirect reference, and finally sets the correct return address in the entry point of the subprogram.

The software which provides the interface between the Assembler subroutine and the Fortran program is always written as follows

```
      NAM    X       define subroutine's name
      ENT    X       define entry point to the subroutine
      EXT    .ENTR   designates the name of the subroutine .ENTR
                     referenced inside X
a     BSS    n       reserve n words of storage for the addresses
                     of the arguments
X     NOP            entry point location
      JSB    .ENTR   jump to .ENTR
      DEF    a       define the first location of the area used
                     to store the argument's addresses
       .
       .
       .             main program of the subroutines
       .
       .
       .
      JMP    X,I     jump indirectly to the return location in
                     the main program
      END
```

All the following subroutines are available in a library file under the name of SPS. All the subroutine arguments are integers.

MAC 1 (ISTOR, NIT, NOD)

This subroutine transfers a record of data from the magnetic tape into the computer memory. The subroutine should be called every time a block of speech data stored into the magnetic tape is required to be processed by the computer simulating an encoding system. The DMA option is used for the data transfer.

The arguments that have to be specified are:

ISTOR , defines the address of the first element of an array declared in the main program and used as a storage buffer for the blocks of data to be encoded.

NIT , defines the select number of the magnetic tape unit from which the data is to be transferred.

NOD , defines the length of the data block which is to be transferred into the computer memory.

A simple flow chart of the MAC 1 subroutine is shown in Figure 3.10.

MAC 2 (ISTOR, NID, NOD)

This subroutine reads a certain block of data from the computer memory and writes the data into a record on the magnetic tape. The subroutine is called in the main program when a block of decoded (i.e. processed) speech samples is required to be stored back into the magnetic tape.

The data transfer is again under DMA control. The arguments to be specified when calling the subroutine are:

ISTOR , provides the address of the first element of the memory storage buffer where the decoded data is kept.

NIT , provides the number of the magnetic tape unit where the data is to be stored.

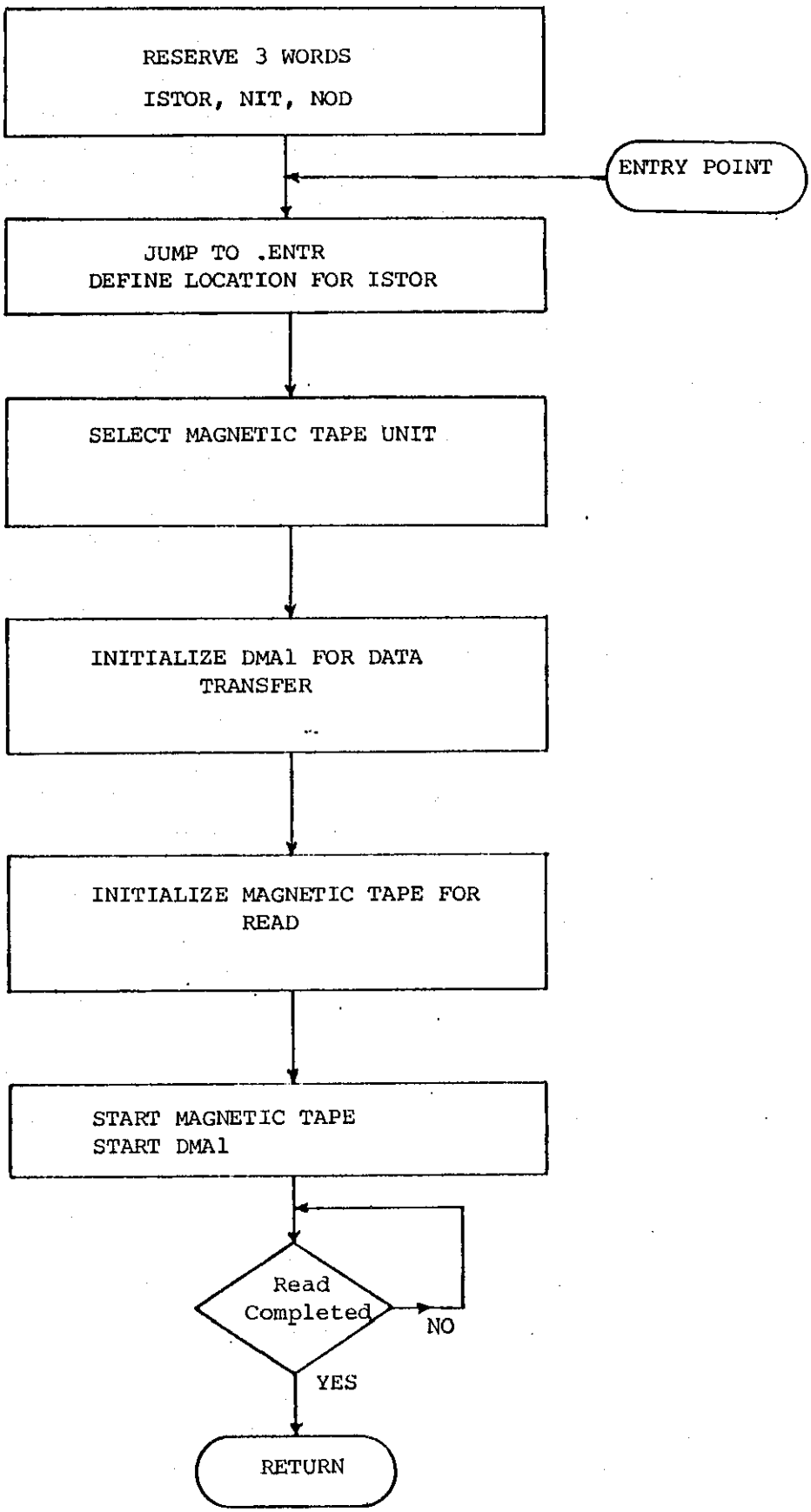


FIGURE 3.10 - Transfer of Data from Magnetic Tape into the Computer.

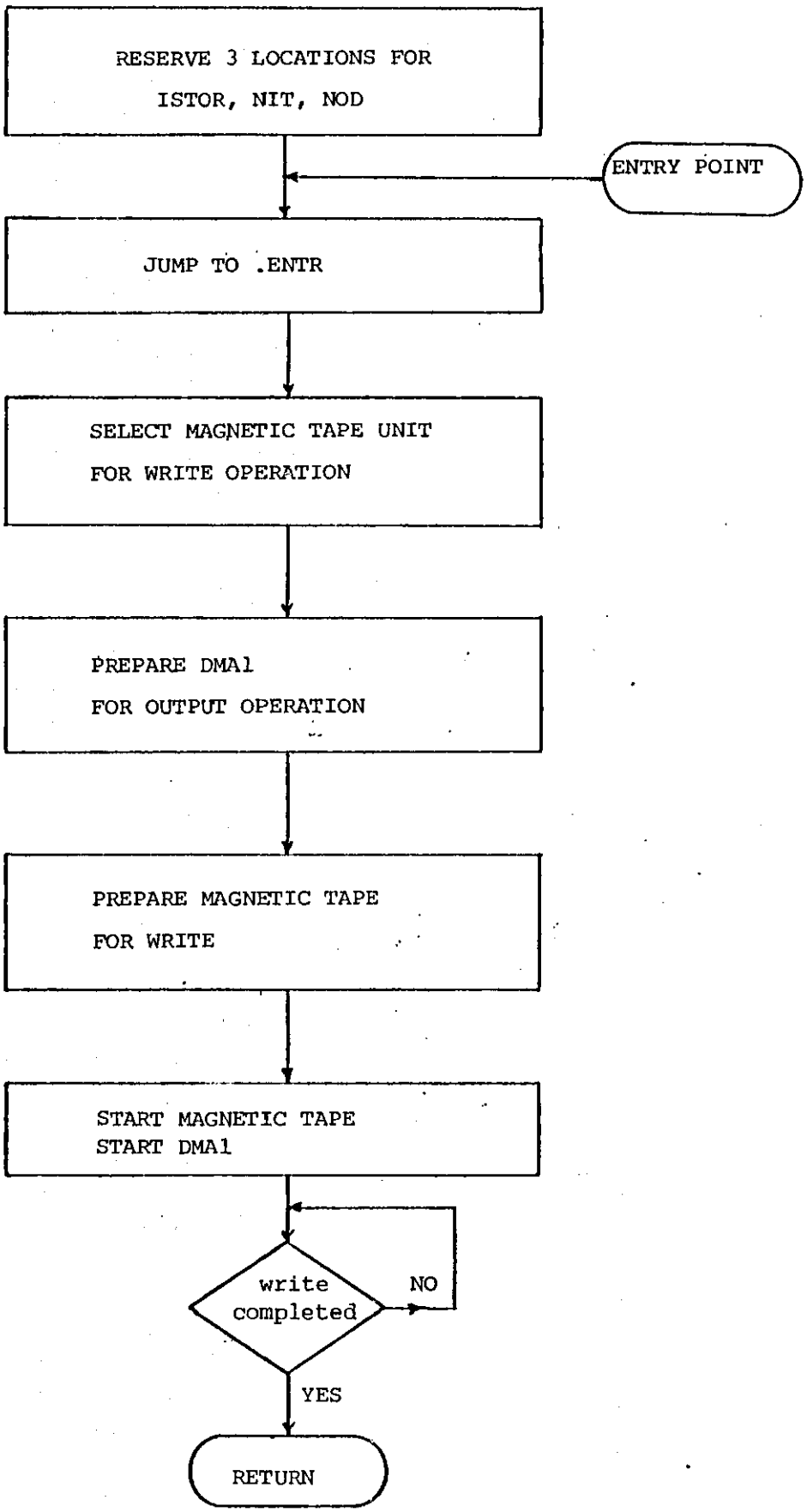


FIGURE 3.11 - Transfer of Data from Computer to Magnetic Tape.

NOD , provides the length of the data record to be written on the magnetic tape.

The flow chart of this subroutine is shown on Figure 3.11. We mention that versions of the MAC1 and MAC2 subroutines are also available employing a standard H.P. magnetic tape driver software package.

COMMD (NIT, COMD)

This subroutine writes file marks and moves the magnetic tape to any required position. The subroutine is called whenever the magnetic tape has to be positioned to a specific record of a file, for a possible read or write operation. The commands given to the magnetic tape using COMMD subroutine include: write file mark, gap and file mark, gap, forward-space record, backspace record, forward-space file, backspace file, rewind, rewind/off-line. The arguments used in the subroutine are defined as follows:

NIT , defines the number of the magnetic tape unit where the command is to be directed.

COMD , provides the code number of the command to be executed.

Figure 3.1 2. shows the flow chart of the COMMD subroutine.

INPT(ISTOR, I1, I2, I3)

The subroutine transfers the incoming speech data from the Analogue-to-Digital converter, into the computer memory and hence into the magnetic tape. INPT is called whenever new speech sentences are to be recorded on the digital magnetic tape.

Both DMA channels are employed in the data transfer. DMA channel 1 is responsible for the transfer of the data blocks between

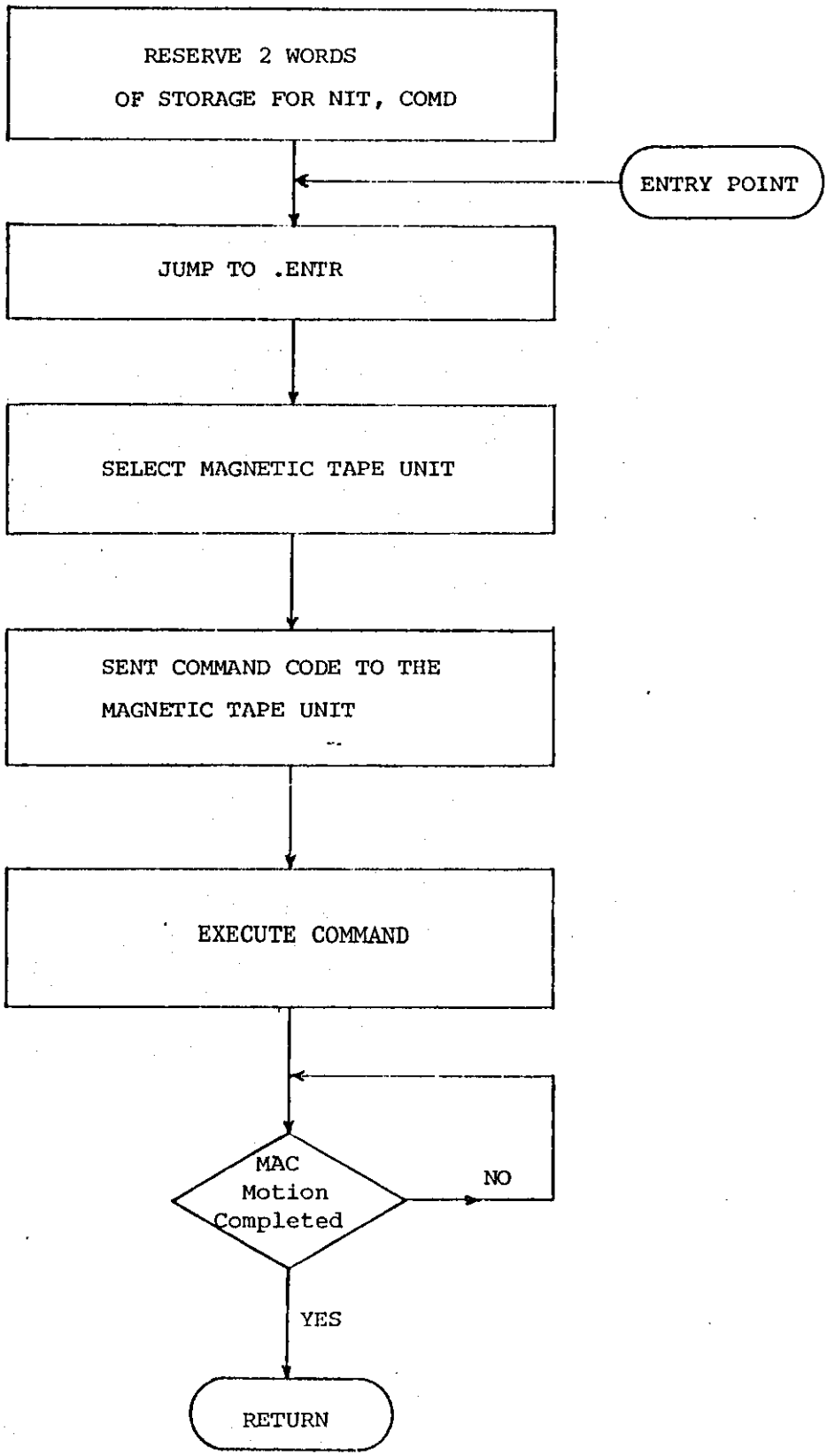


FIGURE 3.12 - The COMMD Subroutine.

the ADC external device (channel 22) and the buffer storage in the core memory, while DMA channel 2 is responsible for the transfer of data from the memory buffer into the magnetic tape (channels 20, 21).

Although only one buffer is used in the memory for serving the data transfer between the ADC device and the magnetic tape, the subroutine is designed in such a way that the whole operation is continuous. This continuous storage of speech data into the magnetic tape is achieved as channel 1 of the DMA is working with the relatively slow clock rate of the ADC device while the DMA channel 2 is working with the much greater speed at which the magnetic tape unit is accepting data. Thus at a given instant of time, where the N'th block of data is inputted, DMA 2 is working ahead of DMA1 moving the data of the N-1 block from the buffer into the magnetic tape. DMA1 operating at a slower speed is behind storing the N'th block of data into the memory buffer.

The rate at which DMA2 transfers the speech data depends upon the block size used in the operation. The greater the size of the data block written onto the magnetic tape, the faster the magnetic tape accepts the data, and therefore the speech waveform can be sampled at a higher rate, if so desired.

Figure 3.13. illustrates the flow chart of the subroutine INPT. The arguments which have to be specified in the main program are defined as follows:

ISTOR , is the location of the first element of the buffer, used in the subroutine. This buffer is declared as an array in the main Fortran program.

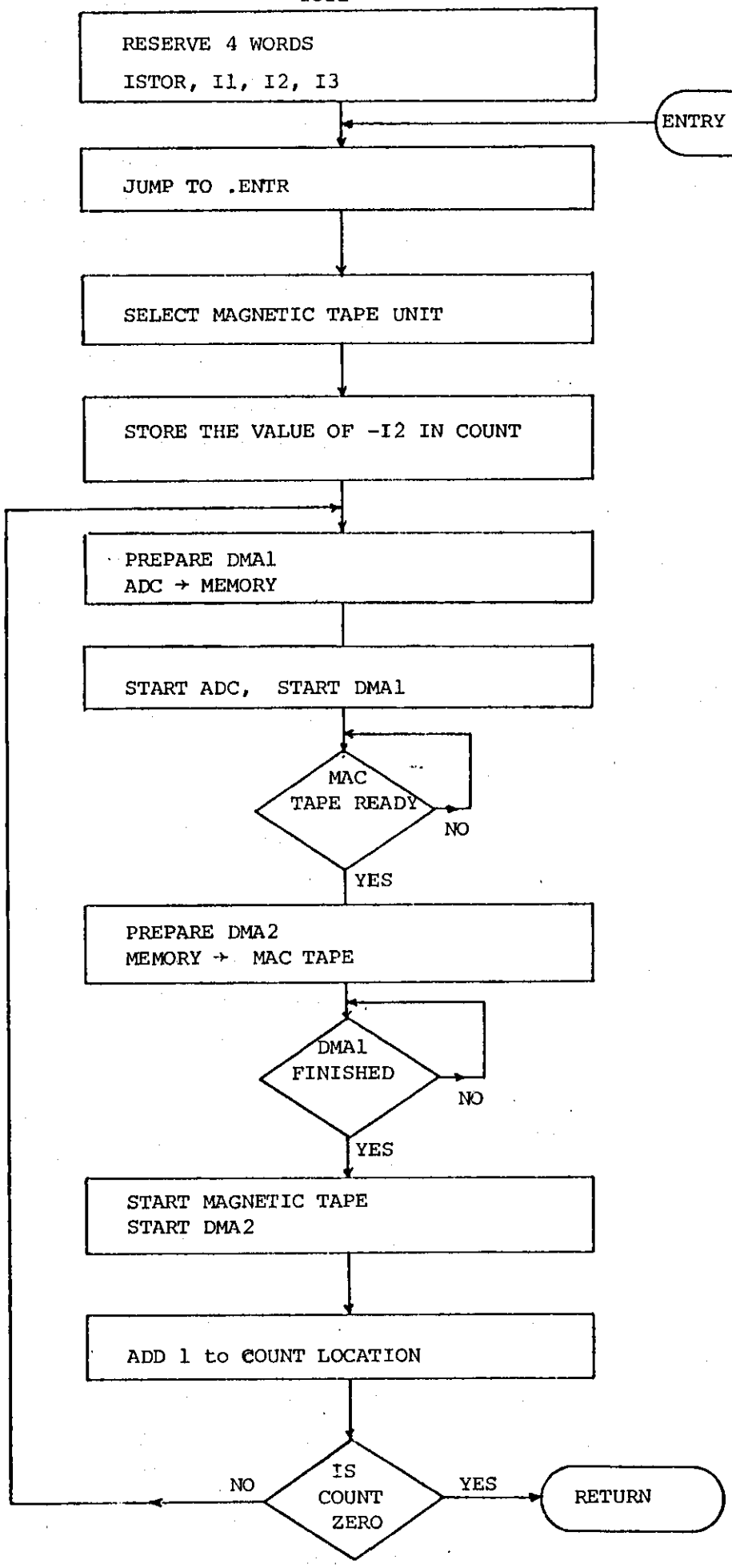


FIGURE 3.13 - Input Operation.

I1 , is the select number of the magnetic tape unit where the data is to be stored.

I2 , is the number of the blocks of data to be stored.

I3 , is the size of the data blocks (in words) used in the transfer operation.

OUTP (ISTOR, I1, I2, I3)

This subroutine transfers recorded speech data from the digital magnetic tape into the computer memory and then outputs the data to the Digital-to-Analogue converter device. OUTP is called in the main program whenever the decoded and stored speech samples are to be outputted through the DAC to a loudspeaker. The transfer is accomplished in blocks as the DMA option is employed for this purpose. DMA channel 2 moves the data between the magnetic tape peripheral and a buffer in the computer memory, while DMA channel number 1 reads from the memory buffer and outputs the data to the DAC peripheral.

The speech waveform at the output of the DAC is continuous as DMA1 operates at a slower rate than DMA2. At a given instant of time DMA2 is filling the memory buffer with a block of the speech data taken from the tape and DMA1 is operating in the same block but some words behind, reading and moving the data to the DAC device. The transfer rate of DMA1 is equal to the rate the speech waveform is sampled in the input operation. The rate of operation in DMA2 depends upon the size of the data blocks. An effective rate of transference of 54.4 k bytes/second is achieved when the block size is equal to 5050 characters.

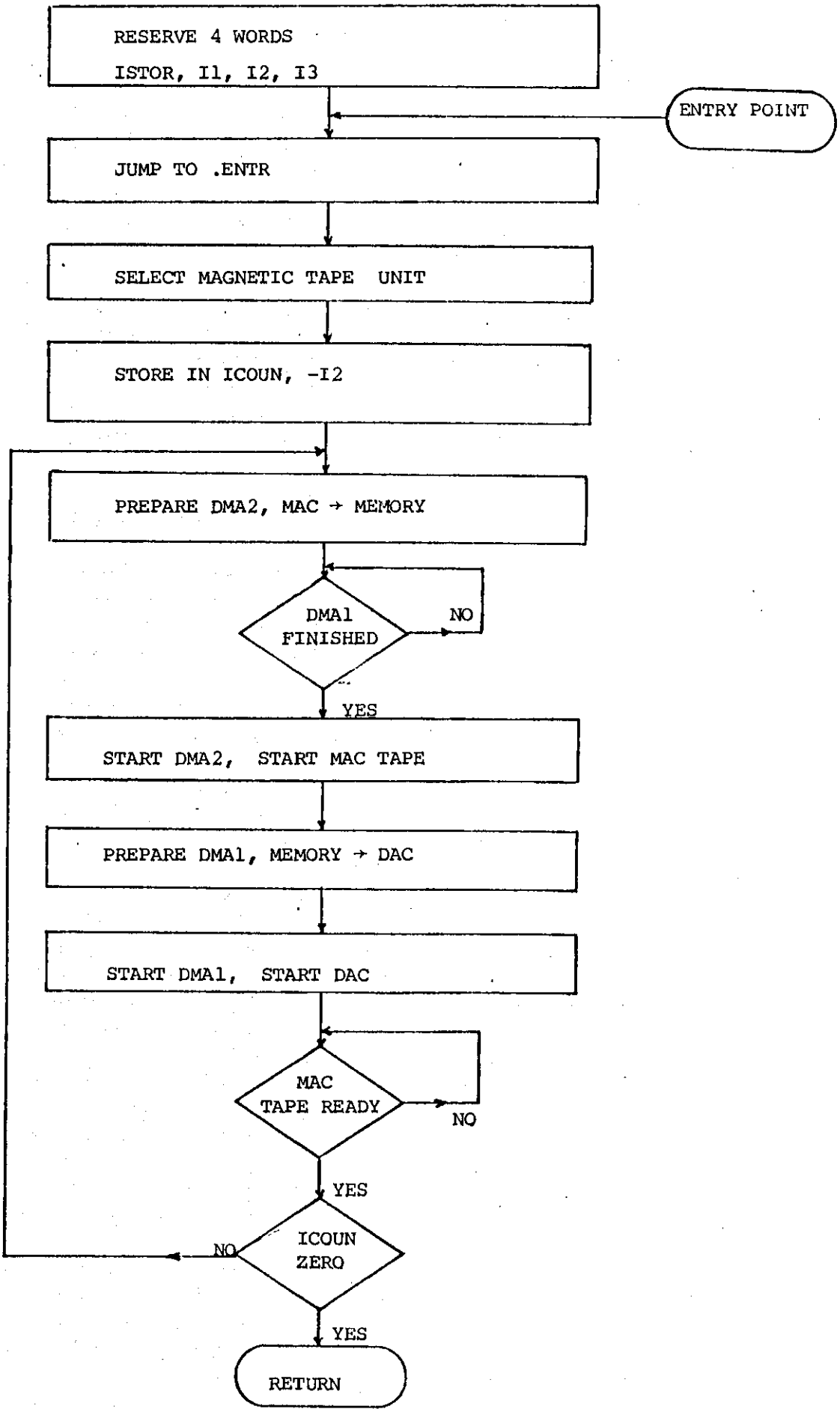


FIGURE 3.14 - Output Operation.

The arguments to be specified when the subroutine is called are:

ISTOR , defines the address of the first element of the buffer in memory. This buffer is declared as an array in the main program.

I1 , defines the code number of the selected digital magnetic unit.

I2 , defines the number of data blocks to be transferred in the output operation.

I3 , defines the size of the used data blocks.

The flow chart of the OUTP subroutine is shown in Figure 3.14.

An absolute program under the name "ABS IN/OUT" which combines both INPT and OUTP subroutines is also available. This program can be stored in and run separately by the computer, without using the BCS operating system. The origin of the program when it is loaded in the memory with the standard Basic-Binary-Loader is equal to 2.

The input-output operation in the INPT-OUTP subroutines or the ABS IN/OUT program can also be accomplished by using a two memory buffer strategy. The program design in this case is rather straightforward. In an input operation for example, one DMA channel, say number 1, transfers the data from ADC into a buffer (ABUF), while DMA channel 2 removes the previous received block of data from the second buffer (BBUF) into the magnetic tape. Thus DMA channels 1 and 2 are transferring data into and out of the memory, switching their operations between the buffers ABUF and BBUF in such way that the recording of the speech data on the tape is continuous.

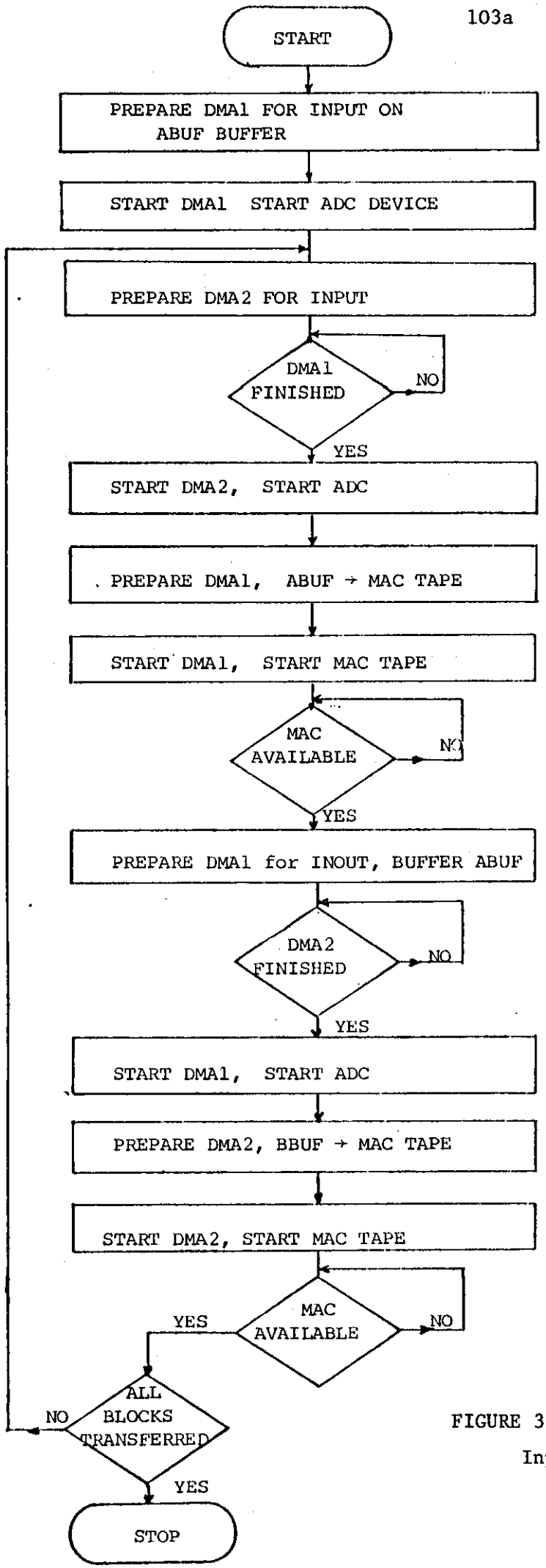


FIGURE 3.15 - Two Buffers, Input Operation.

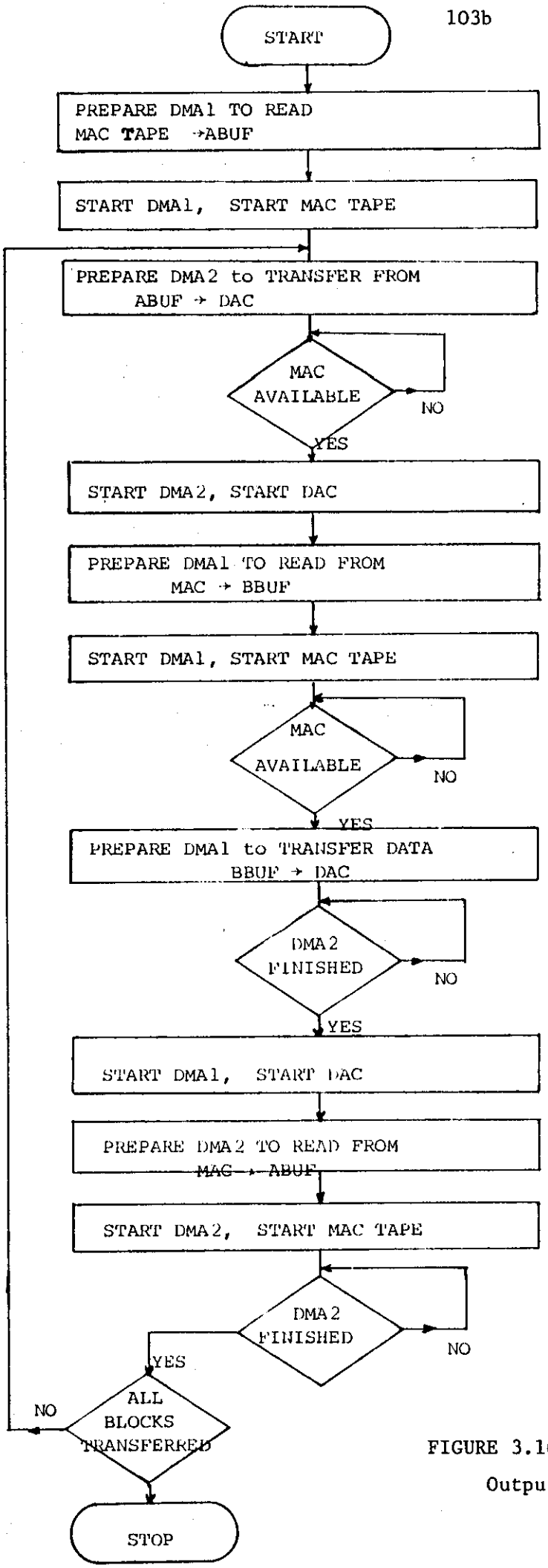


FIGURE 3.16 - Two Buffers, Output Operation.

The two buffer strategy is clearly illustrated for both input and output transfers in the flow charts of Figures 3.15. and 3.16. An absolute program designed, with two buffers in the memory, for input and output operation, is available under the name "ABS 2 IN/OUT".

3.4 DISCUSSION.

Using the hardware and the software interface between the computer and the ADC/DAC peripherals described in the earlier parts of this chapter, complicated processing of speech and subjective tests of the resulting signals can be performed. The basic subroutines which drive the magnetic tape units and the ADC/DAC devices allow the storage of speech material with durations of up to several minutes. Also, there are no limitations in the processing time of the speech signals as the system is designed to work in an "off line mode".

These characteristics of the speech processing system, namely its ability to store and use huge amounts of input signal data, and its handling of extremely complex manipulations on speech signals makes it very useful as a research tool in the field of speech communication.

As a consequence the system has been used extensively not only by the author but also by a number of speech research workers. However, as experience has been gained, it has been found that there are a number of possible additions and alterations that would be beneficial to the system. These are summarized as follows:

(1) The BCS operating system is a paper tape based system and thus considerably slow in loading and linking programs in the computer memory. Also the various compilers are based on paper tape and

this makes the compilation of the programs a time comsuming operation.

In order to save valuable computer time, there is a need for a magnetic tape based operating system which acts as a simple vehicle for quickly loading into the memory software programs such as compilers, the BCS relocatable loader, absolute programs, etc. Such a system is created by transferring software programs from paper tape into magnetic tape. As these are in the magnetic tape environment, the programs can be loaded into core automatically by a supervisory program that operates in response to the users requests.

(2) The number of peripherals used in the system could be extended. It was found that a plotter added to the system would certainly be an improvement. In many experiments the requirement of immediate comparison between the original waveform and the decoded one would be satisfied by a plotter. A line printer could also increase the speed with which the system can list results. Both peripherals, the plotter and the line printer, are available in the Electrical Engineering Department, and only necessary software needs to be developed.

(3) The library of the speech processing system can be extended by simply adding new routines which fit into the general software structure. For example, welcome additions would be subroutines for generating and printing spectograms and subroutines for computing fast autocorrelation functions.

(4) The hardware in the input interface can be modified so that additional information, such as pitch and voiced-unvoiced indications, could be stored in the unused six least significant bits of the computer words.

CHAPTER IV

DELAYED DPCM ENCODING OF SPEECH SIGNALS

4.1 INTRODUCTION

The performance of differential encoders can be significantly improved by anticipating future signal values and modifying accordingly the output of the quantizer. This requires the speech to be sampled and delayed by a few clock periods. Then the delayed samples are encoded using information related to previously encoded samples and knowledge of the future speech samples.

Figure (4.1.) shows a differential encoder employing the Delayed Encoding (DE) technique. The sampled input signal is delayed by an (m-1) stages shift register and when the X_n sample is to be encoded the $X_n, X_{n+1}, X_{n+2}, \ldots, X_{n+m-1}$ samples are presented to the modified quantization algorithm. This enables the quantization strategy to change from a "fixed" to a "sequentially searching" one. To clarify this, we note that for both adaptive and non-adaptive types of normal differential encoders a single decision at the nth instant is made in order to determine the output quantization level. This decision depends upon X_n and the previously decoded samples $\hat{X}_{n-1}, \hat{X}_{n-2}, \ldots$. In contrast, the quantization strategy of a delayed encoder ensures that the incoming input samples $X_{n+1}, \ldots, X_{n+m-1}$ are also used by the quantization process. The quantization algorithm searches for the best $\{L_n^m\} = L_n, L_{n+1}, \ldots, L_{n+m-1}$ sequence of quantized outputs which minimize a certain error criterion f(e). e is defined as the error difference between the $X_n, X_{n+1}, \ldots, X_{n+m-1}$ sequence of

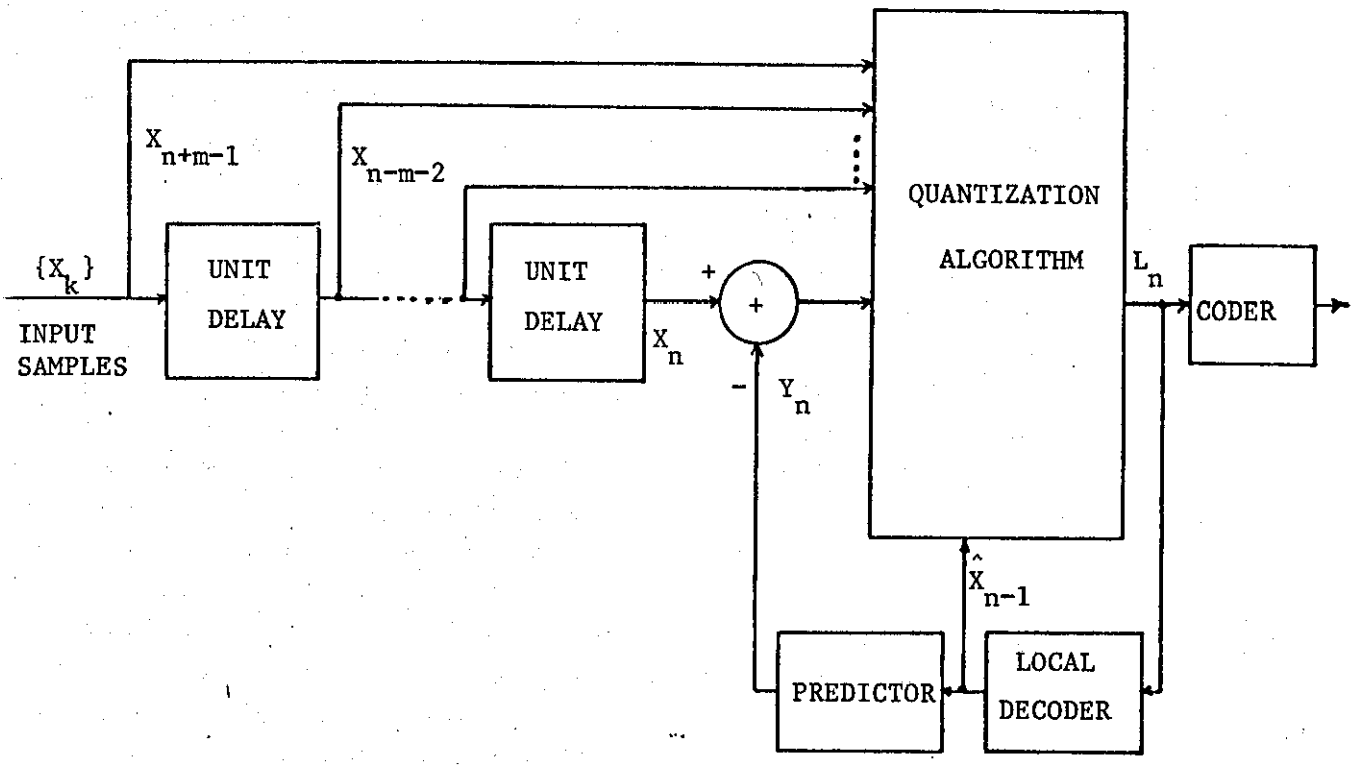


FIGURE 4.1 - Delayed Differential Encoder.

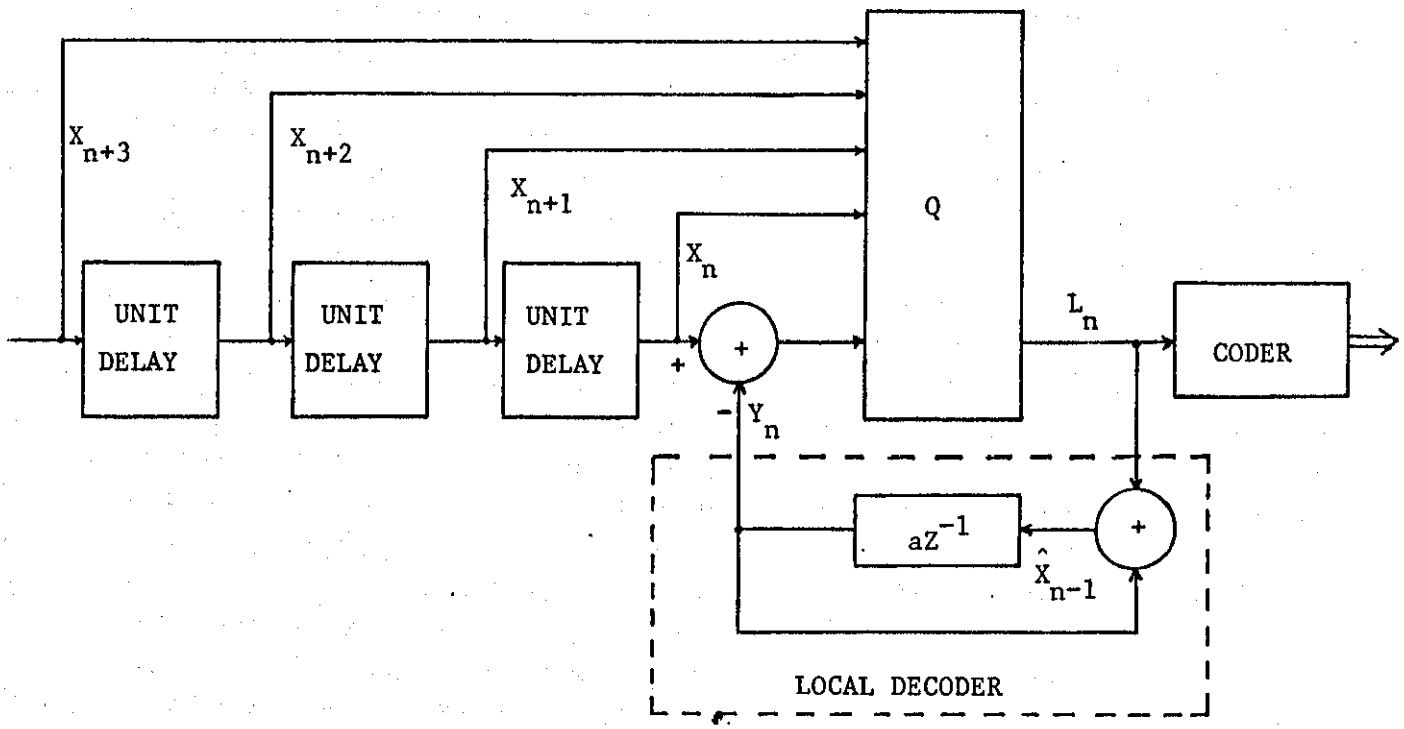


FIGURE 4.2 - Delayed First Order DPCM.

input samples and the $\hat{X}_n, \hat{X}_{n+1}, \ldots, \hat{X}_{n+m-1}$ path* of decoded samples obtained using $\{L_n^m\}$. Then the first sample L_n of the optimum $\{L_n^m\}$ is coded and transmitted, and the procedure is repeated for the encoding of the next sample X_{n+1} while the algorithm is searching through the $\{L_{n+1}^m\}$ sequences.

Delayed encoding has been mainly used with Delta Modulation rather than with Differential Pulse Code Modulation systems. The reason for the preference of DM can be easily seen when considering the number of possible $\{L_n^m\}$ sequences to be checked by the encoder in order to find the one with the minimum f(e). In DM, L_n assumes a binary value and the number of the $\{L_n^m\}$ sequences is equal to 2^m while in a DPCM case with a P level quantizer this number is increased to P^m.

Newton(121) showed that the signal-to-noise ratio of a Linear Delta Modulator can be improved by 2 dB using Delayed Encoding. Cutler(80) demonstrated that delayed encoding can be used as a stablizer in Adaptive-DM; fast adaptation algorithms causing instabilities in DM have been used successfully and offered good encoding performance only when they have been combined with Delayed Encoding.

Zetterberg and Uddenfeldt(122) employed Delayed Encoding in the well known ADM codec whose step size δ_n is updated according to $\delta_n = f(L_{n-1}, L_{n-2}, \ldots, L_{n-k}) \cdot \delta_{n-1}$ where $f(\cdot)$ is a function of the

---

* It can be thought that over the n,n+1,n+2,...,n+m-1 sampling instants a tree is formed having P^m different branches or paths, each of them corresponding to a sequence of $\hat{X}_n, \hat{X}_{n+1}, \ldots, \hat{X}_{n+m-1}$ decoded samples. P is the number of quantization levels used by the encoder.

last k binary values L_i. The computer simulation results of this DE-ADM system using speech-like signals as input have indicated a few dB's snr improvement over the conventional ADM scheme.

Koubanistas[123] proposed the use of the Viterbi algorithm in order to reduce the search time for the optimum $\{L_n^m\}$. He showed that the number of calculations in estimating the error sequence e can be reduced approximately by a factor m.

The Delayed Encoded High Information DM implemented at Loughborough[124] for encoding speech with an output bit rate of 32 kbits/sec., demonstrated that the few dB's advantage of the system over the HIDM provide a noticeable improvement in subjective performance. However the system also showed another inherent feature of the Delayed Encoding, that is its high implementation complexity and considerable cost.

Finally Anderson[125] combined the sequential search of Delayed Encoding together with a modified prediction algorithm in a DPCM system which produced a snr advantage of several dB's over DPCM. In order to simplify the Delayed Encoding search procedure, and make the system practical, he used the so-called M search algorithm which amounts to a highly truncated Viterbi approach.

4.2 THE FIRST ORDER DELAYED DPCM ENCODER

Suppose that speech is band limited to 3.4 kHz, sampled at 8 kHz and it is to be encoded by the Delayed First Order DPCM of Figure (4.2.) The term "First Order" represents the use of only one prediction coefficient in the local decoder. We consider the system to operate at transmission bit rates of 24 kbits/sec or

32 kbits/sec., and consequently the number of quantization levels P will be 8 and 16 respectively. Let us also assume that the number of sampling period delay units used is 3, i.e. m-1 = 3.

Having m fixed to the value of 4 and knowing P, we find the number of possible sequences $\{L_n^m\} = L_n, L_{n+1}, L_{n+2}, L_{n+3}$ to be 8^4 or 16^4 for the two transmission bit rates. Therefore at the nth sampling instant the encoder decodes 8^4 or 16^4 $\{L_n^m\}$ sequences in order to form the corresponding $\{\hat{X}_n^m\} = \hat{X}_n, \hat{X}_{n+1}, \hat{X}_{n+2}, \hat{X}_{n+3}$ paths and then defines the error function f(e) for each path. The error criterion normally used is the summation of the squared errors between the input and the decoded speech samples, i.e.

$$f(e)_n = \sum_{j=0}^{m-1} e_{n+j}^2 \qquad (4.1.)$$

where $e_n = X_n - \hat{X}_n$. The encoder applies Equation (4.1.) for all paths and keeps the path whose f(e) value is a minimum. The first quantization output L_n of this path is then binary coded and transmitted.

The above procedure illustrates the complexity of the system and suggests that its implementation is impractical. In order to reduce the number of calculations required to determine f(e) the Viterbi algorithm can be applied. In such a case it is easy to show[123] that the error function $f^k(e)_{n+1}$ of the kth path at the (n+1) instant is equal to:

$$f^k(e)_{n+1} = f^k(e)_n - e_n^2 + e_{n+m}^2 \qquad (4.2.)$$

where $f^k(e)_n$ and e_n^2 are known from the previous sampling period and e_{n+m}^2 has only to be determined.

Another alternative in simplifying the search procedure is the M-algorithm used in (125), where the algorithm pursues, at every sampling instant only, a limited number of M paths.

To summarize, the basic concept of Delayed Encoding has been discussed, and the "search path" approach of Delayed Encoding proposed and used mainly with Delta Modulation, has been described. It has also been indicated why this technique is not a practical one when applied to DPCM encoders.

At the beginning of the research program described in this thesis, it was felt that Delayed Encoding could be used with DPCM, provided a simplified Delayed Encoding technique, different from the one mentioned above, was used. Consequently our investigations were focused on simple Delayed Encoding algorithms which modify the output samples of a normal DPCM quantizer according to some information of the future input signal values. Two such delayed DPCM algorithms were developed and are presented in the following sections.

4.3 DELAYED FIRST ORDER DPCM. SCHEME 1.

The amplitude of voiced speech waveforms assumes large values at the beginning of the pitch periods and it decreases in an exponential-like way until the arrival of the next pitch period. When the power of the input speech signal increases, the encoder overloads first the large amplitude parts, i.e. at the beginning of the pitch period, and then the rest of the speech waveform. If the First order DPCM encoder is not to be overloaded at all, then the amplitude range of the encoder's quantizer must be large enough to accommodate the high amplitude error samples which occur with

the pitch pulses. In such a case however, the remaining part of the error waveform will be quantized with a rather large quantization step size and this increases the amount of subjectively annoying granular noise produced in the encoding process. Consequently the encoder is allowed to operate slightly overloaded at the time the pitch pulses occur while the rest of the speech waveform is encoded with good accuracy. In fact, it is observed that the maximum signal-to-noise ratio value is obtained (in a First Order DPCM), when the encoder is operating in this slight overload condition.

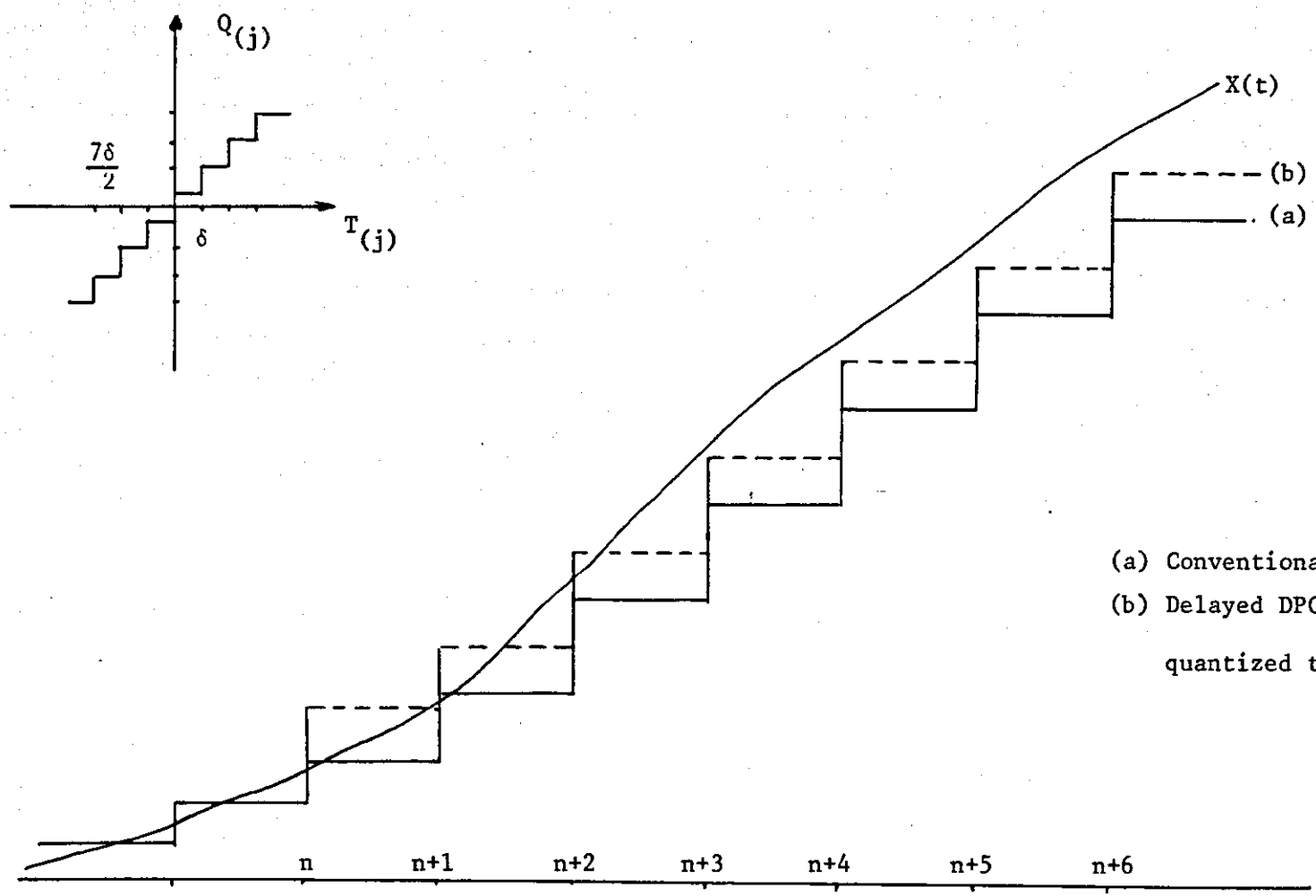
There are possibly two ways in order to improve the performance of the encoder, i.e. to reduce this overload noise while keeping granular noise low.

1) To use an efficient adaptive quantizer instead of a fixed one.

2) To employ a form of Delayed encoding.

The first solution is very effective in improving the encoding performance of a DPCM system and we present in a subsequent chapter of this thesis novel efficient adaptive quantization techniques. We examine here the second approach, that of Delayed Encoding, but before going into the description of Scheme 1 we briefly answer the question of how the encoder can reduce the above mentioned noise using delayed encoding.

Suppose that a DPCM encoder is operating on a arbitrary sampled input signal X(t) and that at the nth sampling instant the error sample E_n, which is well inside the amplitude range of the fixed quantizer, is quantized to the nearest output level of $\frac{3\delta}{2}$, as shown in Figure (4.3.). Suppose also that the next X_{n+2}, X_{n+3},....,X_{n+6} input samples are overloading the encoder. By employing Delayed



(a) Conventional DPCM

(b) Delayed DPCM when E_n is quantized to $\frac{7\delta}{2}$.

FIGURE 4.3.

Encoding with the DPCM, i.e. by making available to the encoder the X_{n+1}, $X_{n+2},\ldots,X_{n+6}$ input samples, the encoder can sense the incoming overload condition and modify appropriately its nth quantization output in order to reduce the overload noise. For example, if E_n is quantized to the maximum output quantization level of $\frac{7\delta}{2}$, instead of $\frac{3\delta}{2}$, the overload noise over the n+2, n+3,..., n+6 sampling instants is considerably reduced. Consequently in the presence of future overload, the Delayed encoding encoder quantizes E_n to a different value than the usual nearest quantization level, resulting in additional granular noise shown in Figure (4.3.). However, as this noise is added towards the direction of the magnitude of the subsequent input samples in overload, the overload noise which in a normal DPCM is produced during the encoding of the $X_{n+2},\ldots,X_{n+6}$ input samples, is reduced. In this way by increasing the noise at the nth sampling instant the quantization distortion is reduced for many consecutive sampling periods and therefore the overall encoding noise is decreased.

Scheme 1 of Delayed Encoding is effectively operating in the same way and modifies the nth quantization output in order to reduce incoming overload distortion. Further, the quantized value assigned to E_n is decided from a single "looking ahead" observation rather than a multi-path search procedure.

4.3.1. Operation of Scheme 1.

The system representation of the Delayed DPCM encoder of Scheme 1, using an ideal integrator, i.e. a = 1, is shown in Figure (4.4.). Suppose the speech sample X_n is presented at the

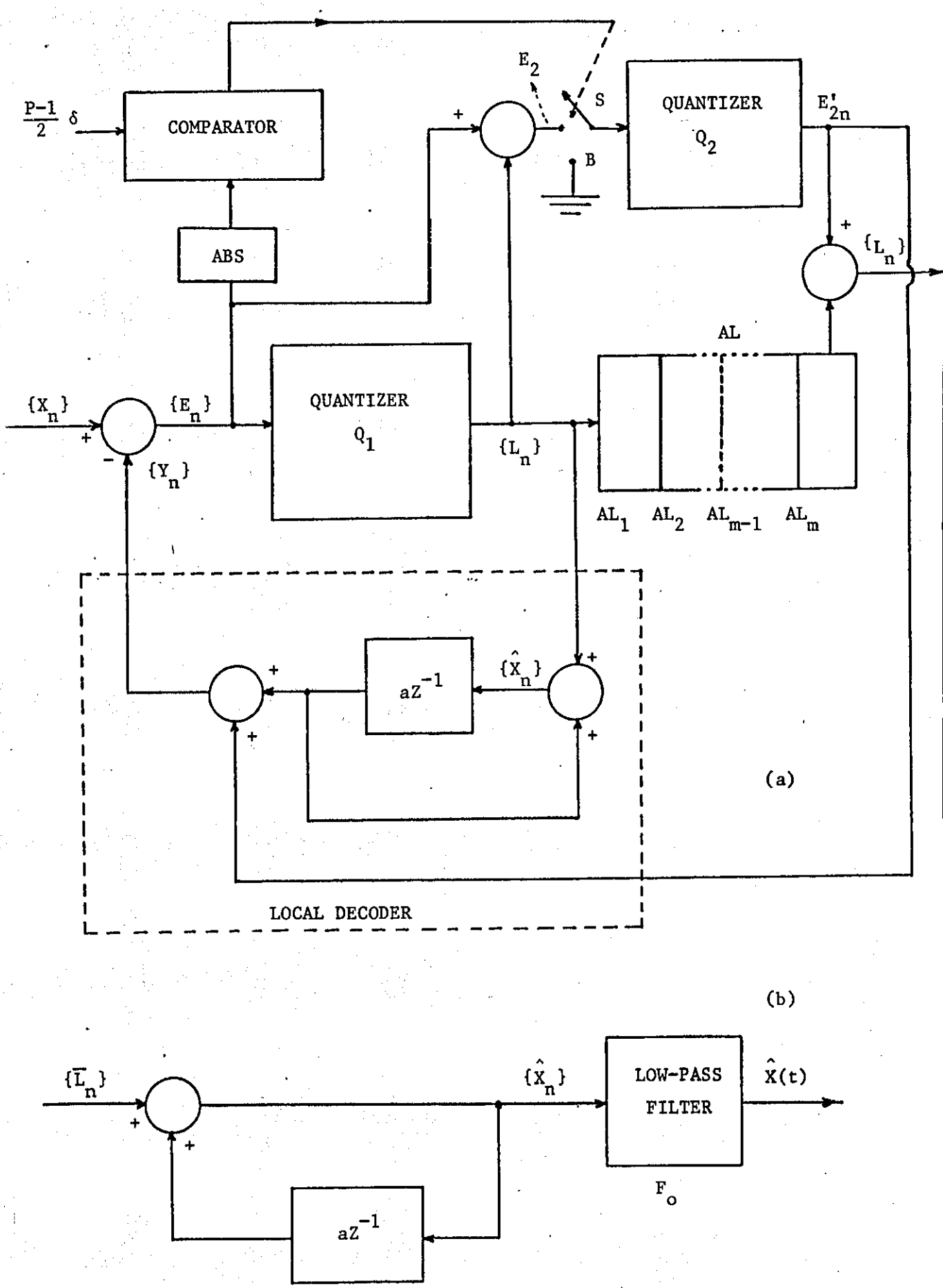


FIGURE 4.4 - Delayed DPCM Codec, Scheme 1.
(a) Encoder, (b) Decoder.

input of the encoder at the nth sampling instant. The feedback sample Y_n, a prediction of X_n, is subtracted from X_n to yield an error sample E_n. E_n is quantized by the Q_1 P level quantizer whose step size is δ and whose output quantization levels L and thresholds T are defined as:

$$L_{(j)} = \pm \left(\frac{1}{2} + j\right)\delta \quad , \quad T_{(j)} = \pm j\delta \tag{4.3a}$$

where $$j = 0,1,\ldots \left(\frac{P}{2} - 1\right) , \quad j = 1,2,\ldots \left(\frac{P}{2} - 1\right) \tag{4.3b}$$

The sample L_n produced at the output of Q_1 is fed to both the local decoder and an m-sampling periods delay-register AL. At the nth instant, AL contains L_n in AL_1, L_{n-1} in $AL_2,\ldots,L_{n-m}$ in AL_m. When the delayed L_{n-m} quantized sample is to be coded and transmitted the $X_{n-m-1},\ldots,X_n$ input samples have been already encoded and consequently the m-delay units register make it possible to examine the speech signal m-1 sampling periods ahead and observe whether the encoder is in overload. If the encoder is overloaded at the nth sampling instant, then before coding and transmitting L_{n-m} we modify its value in order to reduce this overload noise.

In particular, the absolute value (ABS) of the error sample is compared with the maximum output level $\frac{P-1}{2}\delta$ of Q_1. When the absolute value of E_n is less than $\frac{P-1}{2}\delta$, the encoder behaves as a normal First Order DPCM. That is, the output of the comparator which controls the switch S forces this switch to position B and E'_{2n} assumes a zero value. E'_{2n} is added to the L_{n-m} quantized value stored in AL_m and consequently L_{n-m} is coded and transmitted without being modified.

Let us assume now that the absolute value of E_n is larger than $\frac{P-1}{2}\delta$. In this case the switch S connects point A to the input of the quantizer Q_2. The quantization step size of Q_2 is the same as that of Q_1, i.e. δ but its output levels E_2' and thresholds T_2 are defined as:

$$E_2'(j) = 0 \pm j\delta \quad , \qquad T_2(j) = \pm \left(\frac{1}{2} + j\right) \tag{4.4a}$$

where $$j = 1,2,\ldots \left(\frac{P}{2} - 1\right), \quad j = 0,1,\ldots \left(\frac{P}{2} - 2\right) \tag{4.4b}$$

This quantization characteristic of Q_2 ensures that the components in the transmitted sequence $\{\overline{L}_n\}$ are members of the Q_1 output quantization levels set. The difference E_{2n} between the error sample E_n and $L_n = \frac{P-1}{2}\delta$ is quantized by Q_2 to produce E_{2n}'. This sample is then added to the L_{n-m} quantization output stored in AL_m and the resulting $\overline{L}_{n-m}$ sample is coded and transmitted. Of course if $\overline{L}_{n-m}$ is larger than the maximum output level of Q_1 then $\overline{L}_{n-m}$ is made equal to $\frac{P-1}{2}\delta$.

The receiver recovers an approximation of the input signal by presenting the $\{\overline{L}_n\}$ received sequence of samples to the decoder shown in Figure (4.4b).

Now, for the local decoder to operate exactly as the decoder in the receiving end, the value of Y_n has to be adjusted in order to compensate for the addition of E_{2n}' into L_{n-m}. When a = 1 the $\hat{X}_n$ sample is given at the decoder by

$$\hat{X}_n = \sum_{i=1}^{n} \overline{L}_i \tag{4.5a}$$

while Y_n of the local decoder is given by

$$Y_n = \sum_{i=1}^{n-1} L_i \tag{4.5b}$$

Equations (4.5.) illustrates that the addition of E'_{2n} to L_{n-m} increases the value of the received $\hat{X}_n$ by E'_{n2} and therefore E'_{n2} is also added to Y_n of the local decoder as it is shown in Figure (4.4a).

The adjustment of Y_n, when a leaky integrator ($a < 1$) is used in the local decoder, is made in a slightly different way. The nth decoded sample at the receiver is now given by

$$\hat{X}_n = \sum_{i=0}^{n-1} a^i \bar{L}_{n-i} \tag{4.6a}$$

while the local decoder's Y_n sample is equal to:

$$Y_n = \sum_{i=1}^{n-1} a^i L_{n-i} \tag{4.6b}$$

We observe from Equations (4.6.) that the effect in $\hat{X}_n$ of adding E'_{2n} into the L_{n-m} quantization output is not constant but it is decreasing (as expected because of the leaky integrator) and therefore E'_{n2} cannot be directly added to Y_n as in Figure 4.4a. However Equation (4.6b) can also be written as:

$$Y_n = \sum_{i=1}^{m} a^i L_{n-i} + a^{m+1} \hat{X}_{n-m-1} \tag{4.7.}$$

with the first m term of Equation (4.6b) retained while the remaining terms are substituted by $a^{m+1} \hat{X}_{n-m-1}$. To clarify the equivalence between Equations (4.6b) and (4.7.) we consider Equation (4.6b) with $n = 10$ say, i.e.

$$Y_{10} = aL_9 + a^2L_8 + a^3L_7 + a^4L_6 + a^5L_5 + a^6L_4 + a^7L_3 + a^8L_2 + a^9L_1.$$

Now if we assume that $m = 4$ the last Equation can take the form

$$Y_{10} = aL_9 + a^2L_8 + a^3L_7 + a^4L_6 + a^5\hat{X}_5 \tag{4.7a}$$

because $\hat{X}_5 = L_5 + aL_4 + a^2L_3 + a^3L_2 + a^4L_1$.

It can be seen that Equation (4.7a) is also obtained from Equation (4.7) when n = 10 and m = 4.

By using Equation (4.7.) in the design of the Local Decoder the L_{n-m} quantization output is available at the nth sampling instant. So L_{n-m} can be modified by the E'_{n2} sample and the decaying effect of the leaky integrator (a < 1) can also be taken into consideration when the Y_n feedback sample is formed, see Figure 4.5.

After binary coding, the $\{\overline{L}_n\}$ samples are transmitted to the receiver. Equation (4.6a) describes the decoder whose arrangement is shown in Figure 4.4b for both a = 1 and a < 1 cases. Assuming that the transmission channel is error-free, $\{\overline{L}_n\}$ is recovered and decoded to produce the $\{\hat{X}_n\}$ sequence of samples. The high frequency out-of-band quantization noise is rejected after passing the $\{\hat{X}_n\}$ sequence through a low pass filter F_o and the original speech together with the in-band quantization noise emergies at the receiver output as $\hat{X}(t)$.

As we have seen, the decision of adding zero or a certain amplitude value E'_{2n} into the L_{n-m} sample depends upon the outcome of the comparison between E_n and $\frac{(P-1)}{2}\delta$. We also mentioned that the overload noise associated with the $\hat{X}_n, \hat{X}_{n+1}, \ldots, \hat{X}_{n+r}$ samples is reduced at the expense of adding some granular noise in the $\hat{X}_{n-m}$ sample. This means that the Delayed encoding algorithm will reduce the overall quantization noise only if r > 1. Consequently we add to the decision characteristics of the comparator (see Figure 4.4a) which controls the action of switch S, the constraint that the point A is connected to the input of the Q_2 quantizer iff a certain number IO

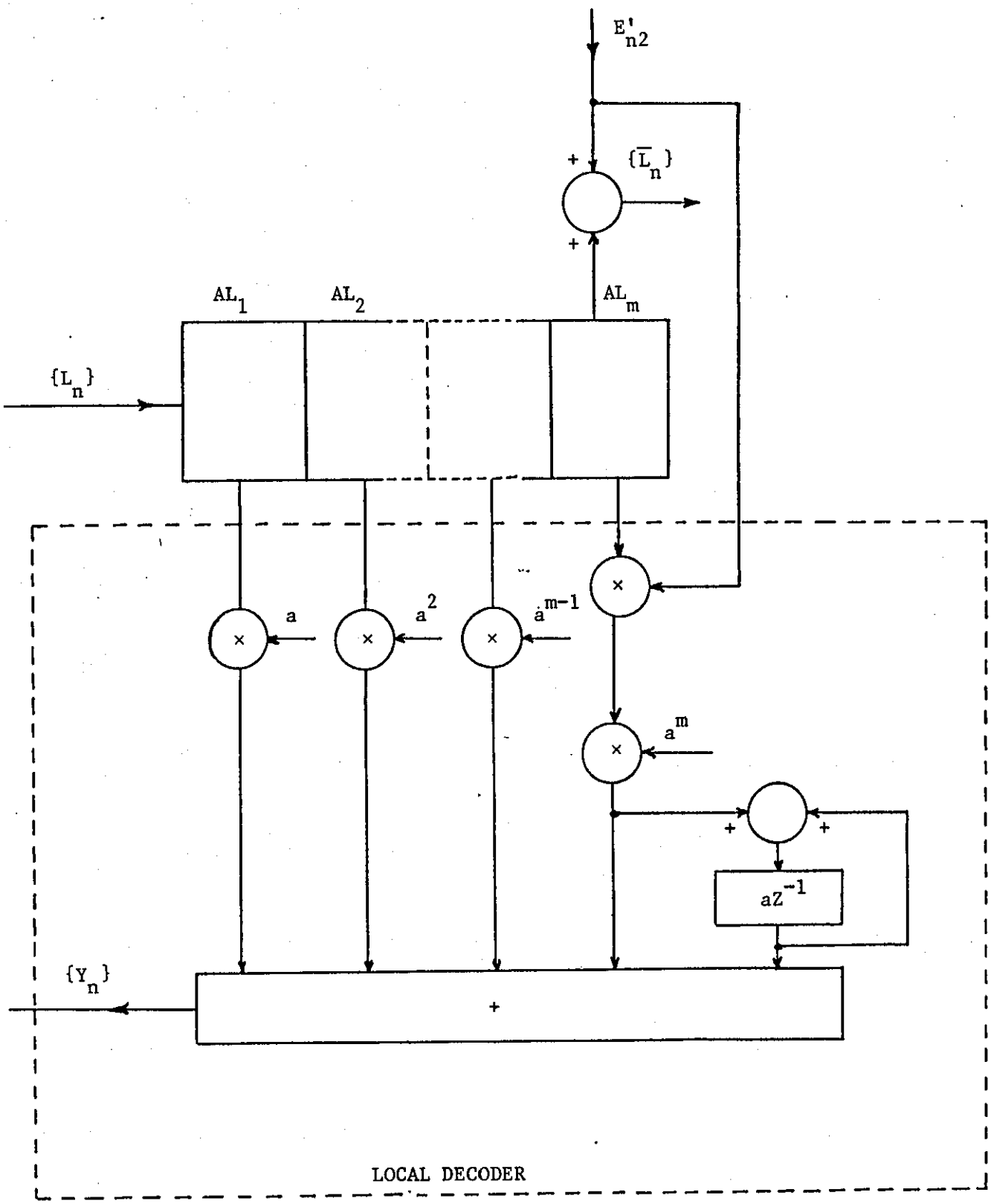


FIGURE 4.5 - The Local Decoder when a < 1.

of successive E_n samples are larger than $\frac{(P-1)}{2}\delta$. The values of IO and the number of delay units m in AL were determined from computer simulation experiments.

To complete this section and show how the algorithm of Scheme 1 improves the performance of a DPCM system in the presence of overload, we refer to Figure 4.6. where an arbitrary signal X(t) is encoded by a First Order DPCM and a Delayed Scheme 1 DPCM system. Figure 4.6, shows the increase of the granular noise prior to overload and the overall reduction of the encoding distortion. It also illustrates that this granular noise assumes the form of peak distortion while the encoder tends to track the input signal closely for most of the time.

4.3.2. Computer Simulation Outline.

The Delayed DPCM encoder of Scheme 1, presented in the previous section, plus a first order DPCM encoder have been simulated on the HP 2100A computer-based speech processing system described in Chapter III. The input data used in simulation experiments are segments of continuous speech, band-limited to 3.4 kHz, sampled at the rate of 8 kHz/sec. and stored on a digital magnetic tape. The speech material, spoken by a male, is from a RSRE(C), Christchurch standard voice tape.

The overall simulation procedure is indicated in the diagram of Figure 4.7. Most of the programming is written in HP Fortran II Language, except the subroutines which transfer data between the computer memory on the magnetic tapes. Having as reference the diagram of Figure 4.7, this rather general computer simulation

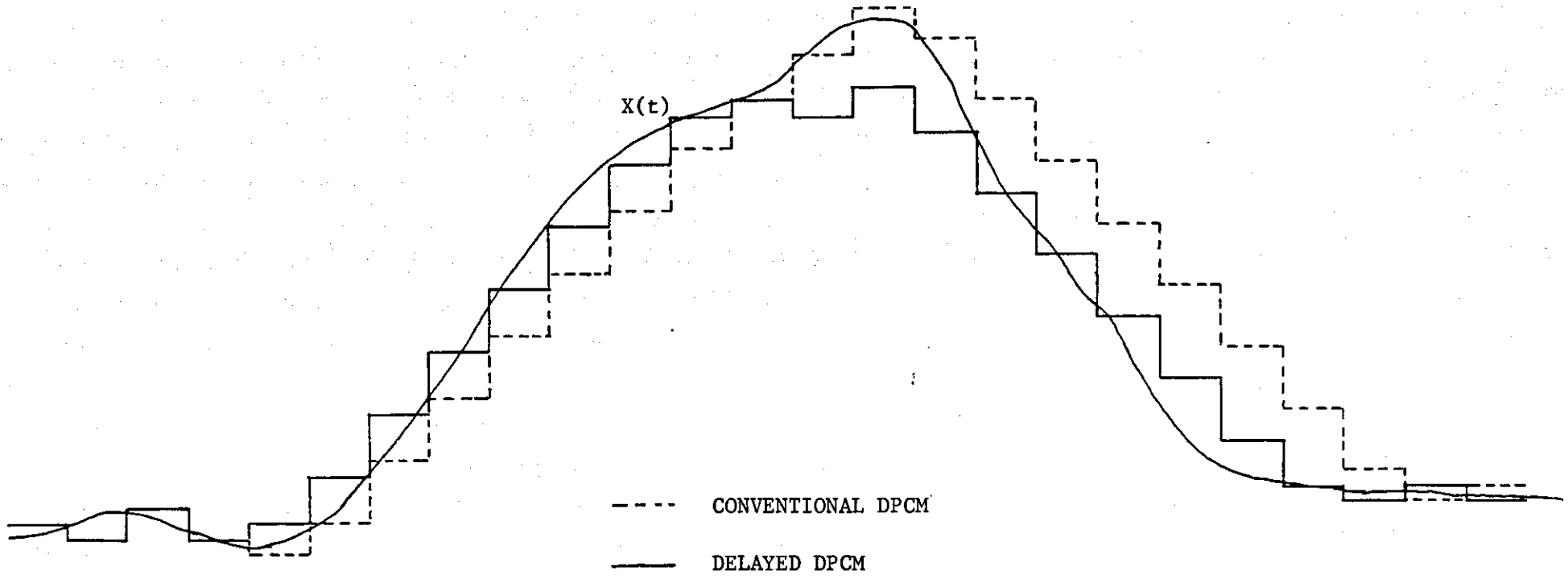


FIGURE 4.6.

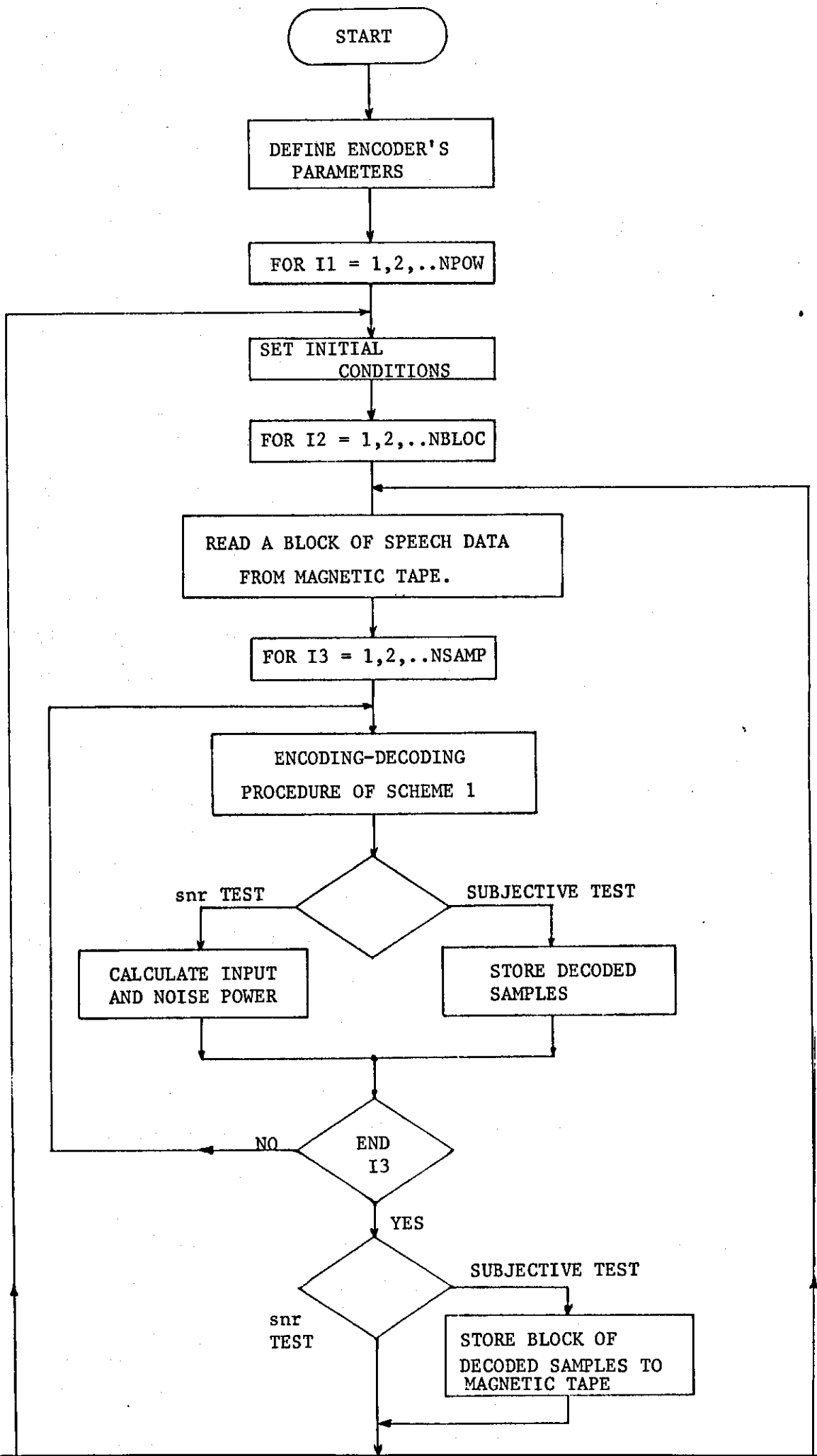
procedure is outlined first, which can be used for testing the performance of many waveform encoders by simply changing the "Encoding-Decoding procedure" part of it.

After the starting point, the computer program issues statements asking for various encoder parameters, such as the value of the quantization step size δ, the number m of delay units, and the value of IO. Furthermore, because the speech data is stored on the magnetic tape in blocks of 5000 samples, we specify the number NBLOC of data blocks to be processed by the encoder. Other information given to the computer include whether the experiment is to provide signal-to-noise ratio (snr) measurements, or whether the processed speech is to be stored back into another magnetic tape for further subjective tests, (ST).

Also, in the case of snr measurements, a set of NPOW power factors is given to the program so that the input speech data is scaled into different power level before being encoded.

After all the above parameters are made available to the computer, the procedure enters the "power points" Do Loop I1 which is executed NPOW times. The function of the I1 Loop is to present NPOW times a specific amount of input speech data into the Scheme 1 encoder, while each time the input speech assumes a different power value.

Now the initial conditions of the encoder are set. For example, memory locations assigned to store the power values of the input and quantization noise signals are set to zero, the digital filters used in the procedure are reset etc. Then a I2 Do Loop follows which allows NBLOC input data blocks to be transferred from the digital


START
DEFINE ENCODER'S PARAMETERS
FOR I1 = 1,2,..NPOW
SET INITIAL CONDITIONS
FOR I2 = 1,2,..NBLOC
READ A BLOCK OF SPEECH DATA FROM MAGNETIC TAPE.
FOR I3 = 1,2,..NSAMP
ENCODING-DECODING PROCEDURE OF SCHEME 1
snr TEST
SUBJECTIVE TEST
CALCULATE INPUT AND NOISE POWER
STORE DECODED SAMPLES
NO
END I3
YES
SUBJECTIVE TEST
snr TEST
STORE BLOCK OF DECODED SAMPLES TO MAGNETIC TAPE

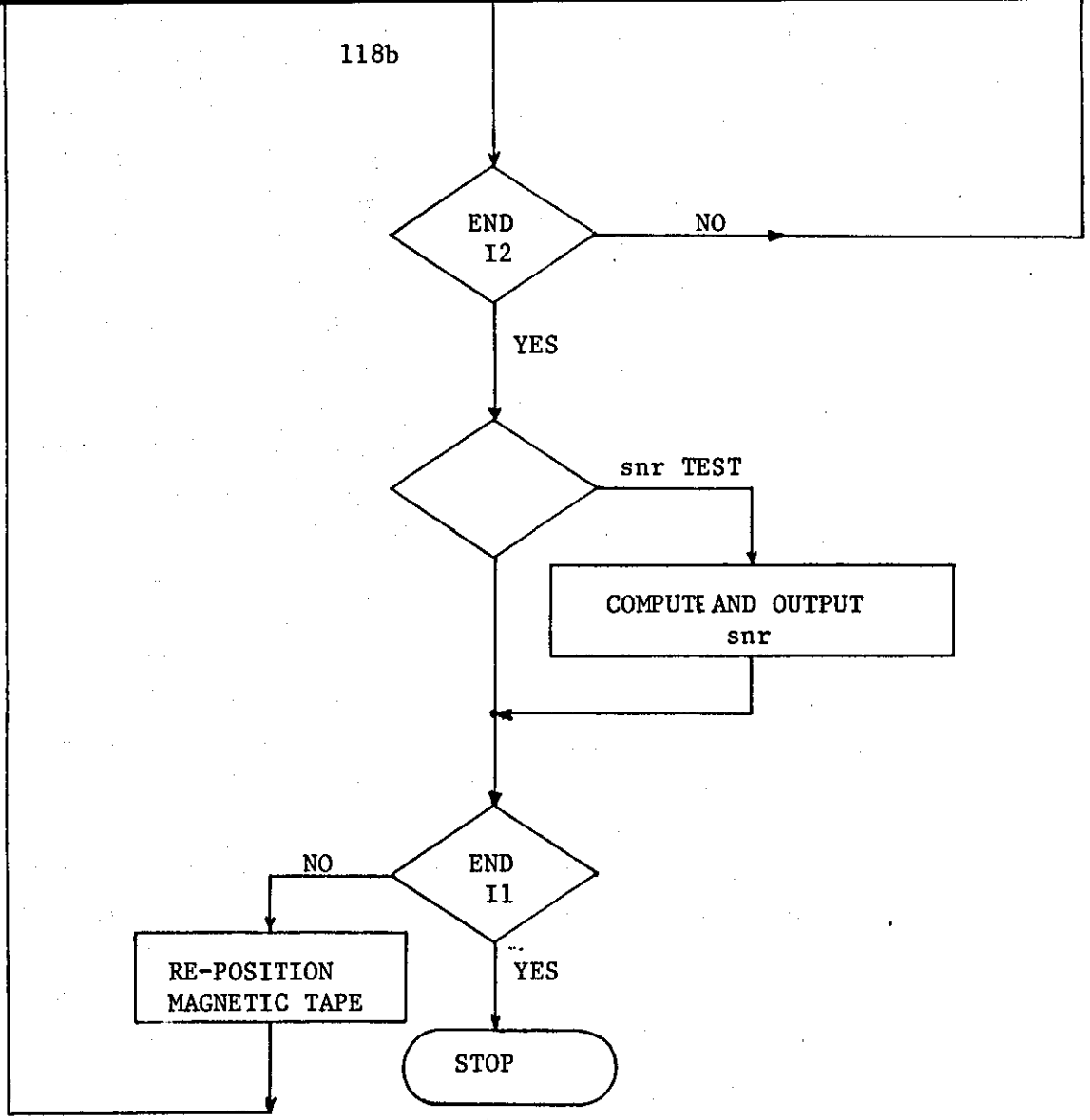


FIGURE 4.7 - Overall Simulation Procedure.

Magnetic tape to the memory buffer in order to be processed by the encoder. The data is already scaled to a power value specified in the I1 Do Loop. The control parameter I3 of the Do Loop which follows, varies from one to NSAMP, where NSAMP is the number of samples stored in each data block. These samples are sequentially presented to the Encoding-Decoding procedure of Scheme 1 and for each input sample X_k a decoded $\hat{X}_k$ is obtained. At this point the program checks if a (snr) or a (ST) instruction has been entered at the beginning of the procedure. If a snr test is to be carried out, the power of the input and the quantization noise samples are calculated. In the case of an (ST) test, the decoded samples are stored to a memory buffer.

Once the programs comes out the I3 Do Loop, it checks again for a (snr) or a (ST) command. When the encoder is to be tested subjectively, i.e. (ST) is true, the block of decoded samples stored in the memory buffer is transferred back into a second magnetic tape. Then the program returns at the beginning of I2 Loop. The same also happens when the signal-to-noise ratio is to be calculated. Consequently the process of transferring a block of input samples from the magnetic tape to the computer memory and to encode the samples using the system of Scheme 1 is continued until the program comes out of the I2 Loop.

A further (snr) or (ST) test, follows. If (snr) is true the signal-to-noise ratio is computed. Then the program returns to the starting point of the I1 Loop, after re-positioning the magnetic tapes back at the beginning of the speech segment. When the I1 Loop is completed the program stops while a set of snr values for different

input power levels or several minutes of decoded speech data, stored on digital magnetic tape, are available.

The simulation procedure described so far has been used to evaluate not only the performances of the Scheme 1 and the normal DPCM systems but also the performance of other schemes examined in this chapter. The part of the procedure which we are to consider next, is the calculation of the signal power σ_x^2 and the quantization noise power σ_e^2 required to determine the signal-to-noise ratio.

The input speech power, σ_x^2 is calculated by averaging the signal power over the length of the speech segment used in the simulation experiment. That is,

$$\sigma_x^2 = \frac{1}{N} \sum_{i=1}^{N} X_i^2 \qquad (4.8.)$$

where X_i is the ith sample in a sequence of N band-limited input samples.

The in-band noise power σ_e^2 can be calculated in two ways.

a) by first passing the $\{\hat{X}_i\}$ sequence of decoded samples through a low-pass digital filter which rejects the out-of-band quantization noise. Then after compensating for any delays introduced from the filtering process, the error is formed between the original input samples X_i and the samples $\tilde{X}_i$ at the output of filter, i.e.

$$e_i = X_i - \tilde{X}_i \qquad (4.9.)$$

and the noise power is equal to

$$\sigma_e^2 = \frac{1}{N} \sum_{i=1}^{N} e_i^2 \qquad (4.10.)$$

b) by calculating the error signal $\{e_i\} = \{X_i - \hat{X}_i\}$ between the input and the decoded speech samples and low-pass filtering $\{e_i\}$ to obtain a sequence $\{\tilde{e}_i\}$ of band-limited error samples. The noise power is then formed as:

$$\sigma_e^2 = \frac{1}{N} \sum_{i=1}^{N} \tilde{e}_i^2 \qquad (4.11.)$$

There is a difference between these two methods and can be simply analysed as follows.

In the second method the error samples, used to form the noise power are estimated as:

$$\tilde{e}_i = H\left[X_i - \hat{X}_i\right] \qquad (4.12.)$$

where $H[\cdot]$ represents a band limited process. Assuming that the digital filter used is a linear non-recursive one, $H[\cdot]$ is a linear operation and consequently

$$\begin{aligned} \tilde{e}_i &= H\left[X_i\right] - H\left[\hat{X}_i\right] \\ &= H\left[X_i\right] - \tilde{\hat{X}}_i \end{aligned} \qquad (4.12a)$$

Comparing Equations (4.9.) and (4.12a) we notice they differ in that in method (b) the already band-limited input sample is filtered again. The magnitude of the difference between $\tilde{e}_i$ and e_i is zero when an ideal (rectangular-like response) low pass filter is used or is very small if the filter has a sharp cut-off characteristic.

It has been decided in our simulations to use the second method for computing σ_e^2 while the filter $H[\cdot]$ is a recursive one. The reason for this choice is that the precise estimation and

compensation of the delay of the filter, required in the first method, is only achieved with a non-recursive type filter, which is usually of large length (typically 256 coefficients). In contrast, the recursive type filter which can be used in the first method with no delay compensation required, employs a limited number of coefficients and can rapidly process the error samples. The difference between the two methods is of the order of .1 dB's. The design details of the digital filter used in the simulations are presented in Appendix A.

After calculating σ_x^2 and σ_e^2, the signal-to-noise ratio is given by:

$$\text{snr} \triangleq 10.\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) \quad \text{(4.13.)}$$

We end this section by showing in Figure 4.8 a simplified flow-chart of the simulation procedure for Scheme 1. The encoder employs an ideal integrator a = 1 in the feedback loop.

The input speech sample X enters the AH shift register which is used to delay the input samples by the same number of sampling periods as the AL register delays the L_n quantized samples. In this way, when the decoded sample $\hat{X}_{n-m}$ is obtained, the corresponding X_{n-m} input sample is taken from AH and the correct error sample $(X_{n-m} - \hat{X}_{n-m})$ is formed. The formation of the X1 = X - XN difference follows, where XN is the feedback sample in the Local DPCM Decoder. The error sample X1 is quantized by the fixed quantizer Q1 and its output sample Y1 is fed to the m stage shift register AL.

The next step in the program is to compare the absolute value of X1 with the maximum output quantization level of Q1. If X2 is

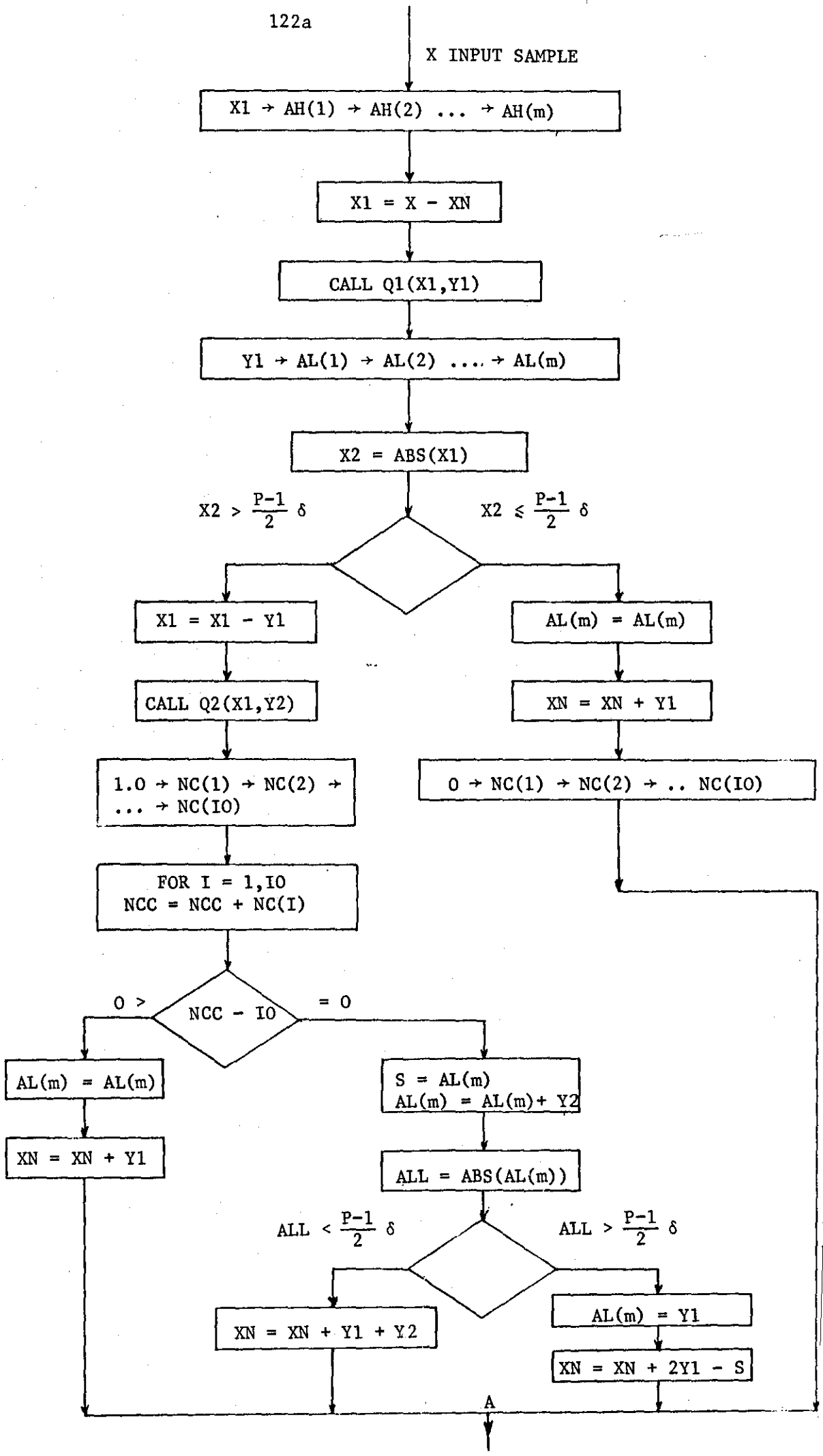


FIGURE 4.8 - Part of Scheme 1 Simulation Procedure.

less than this output level the sample stored in AL(m) remains unchanged and a zero is fed into the IO units shift register NC which is used for detection of successive input samples in overload. Then before going into the σ_x^2 and σ_e^2 calculations the feedback sample XN is formed as the sum of its previous value plus Y1. However, when X2 is larger than the maximum output level of Q1 the difference between X1 and Y1 is formed and quantized by Q2, whose output is Y2. The value of unity is then inserted into the NC register and the zeros and/or ones contained in NC are added to form a NCC sum. When NCC is less than IO it means that there are fewer or no samples in overload than the pre-defined IO number and therefore no action is taken to modify the value of AL(m). In contrast if NCC is equal to IO, AL(m) is modified by adding in it the value of Y2. Before forming the feedback signal XN the magnitude of AL(m) is examined. When AL(m) is less than the maximum output of Q1, XN is equal to the summation of its previous value plus the values of Y1 and Y2. In the case, however, when AL(m) = S is larger than $\frac{(P-1)}{2}\delta$, the value of AL(m) is restricted to Y1 and XN is equal to its previous value plus twice the value of Y1 minus S.

Finally, as shown in the flow chart of Figure 4.8, all the above described separate program paths are merged to the reference level A. The program continues with the calculation of σ_x^2 and σ_e^2.

4.3.3. Encoding of Speech Signals - Results.

The overload distortion reduction advantage of the Scheme 1 Delayed DPCM over the First Order DPCM system was shown in Figure 4.6 when a arbitrary signal X(t) was encoded. Now we refer to Figures 4.9 and 4.10, in order to discuss the performance of the Scheme 1 Delayed algorithm when encoding speech signals.

A section of voiced waveform having duration of approximately one pitch period is shown in Figure 4.9. Curve (a) is the original input waveform while Curve (b) is the decoded one produced from a 3 bits/sample First Order DPCM encoder operating in a slightly overload condition. The overload is present at the beginning of the pitch period where the amplitude value of the speech signal changes significantly between sampling periods and the feedback signal $\{Y_n\}$ of the encoder is unable to follow these fast amplitude variations.

In Figure 4.10 the same segment of the input speech waveform is shown by Curve (a) while its decoded version, produced by a 3bits/sample Scheme 1 Delayed First Order DPCM, is shown in Curve (c). The modification of the quantized error sample values before transmission, due to the Delayed encoding algorithm, changes the rate with which the amplitude of the decoded waveform varies. (See Figures 4.9, 4.10). This change in the slope of the decoded waveform has two effects, i) the so produced decoded speech waveform is a better approximation of the input speech signal than the decoded waveform of a normal First Order DPCM. The Delayed encoding algorithm is effectively limiting the overload noise at the cost of some peak distortion. ii) Scheme 1 tends to preserve the zero crossing of

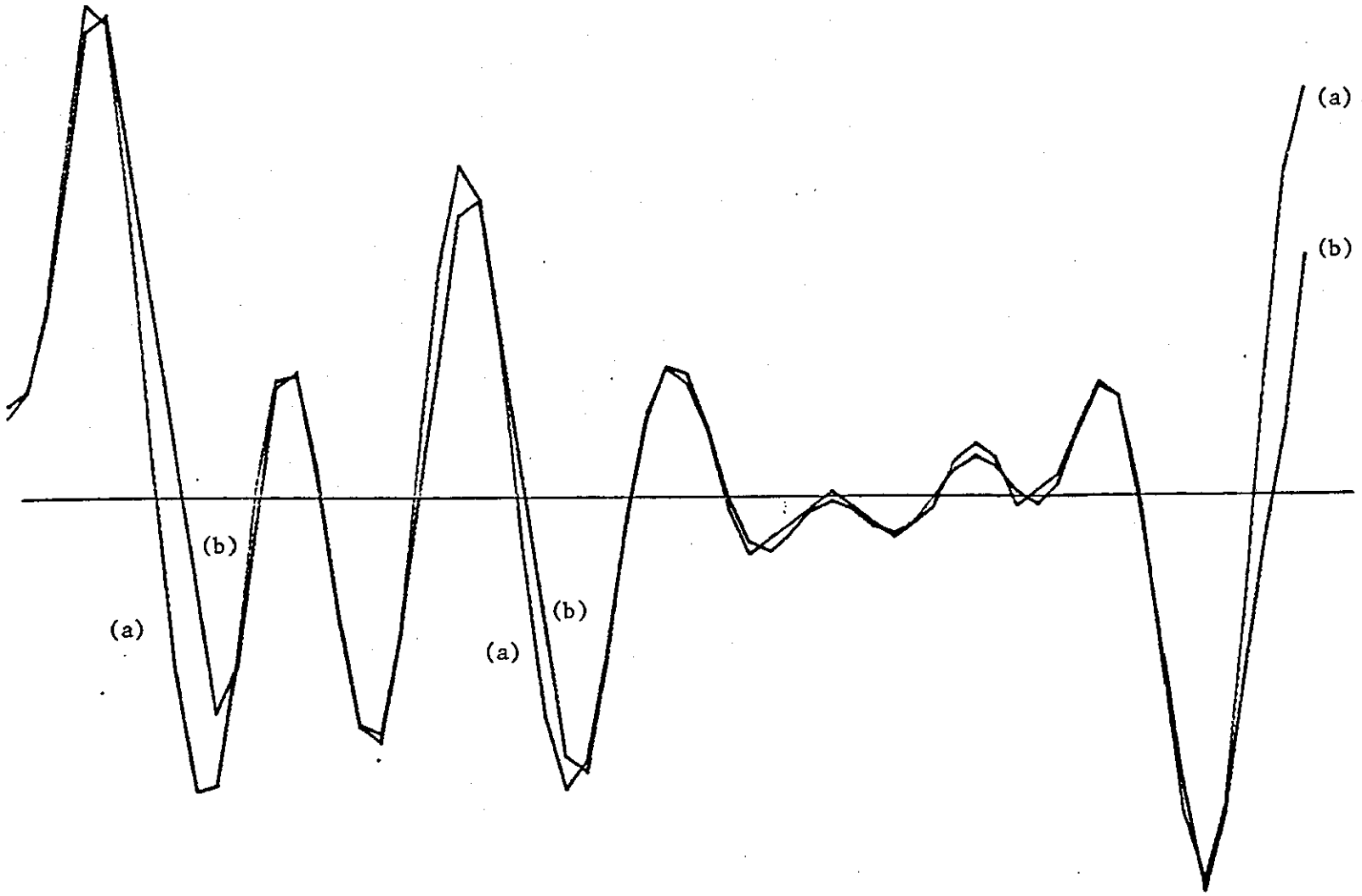


FIGURE 4.9 - (a) Original Speech Waveform
(b) Decoded Speech Waveform from a Overloaded DPCM Encoder.

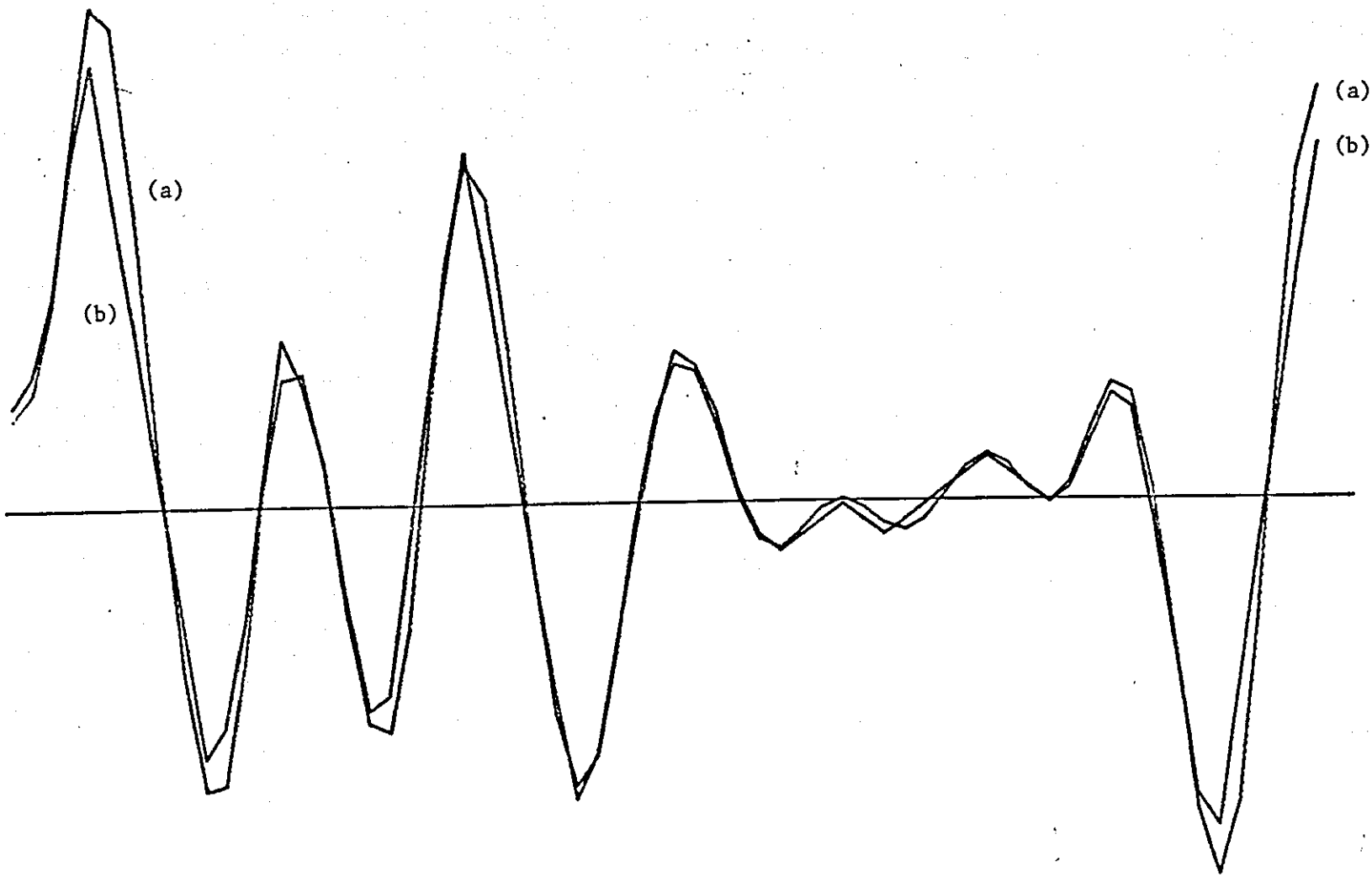


FIGURE 4.10 - (a) Speech Waveform
(b) Decoded Speech Waveform from a Scheme 1 DPCM.

the speech waveform which are otherwise shifted from their original position due to overload noise. According to Linklider[126] and Morris[127], the intelligibility in speech depends significantly upon the preservation of its zero crossings. Thus the Delayed algorithm of Scheme 1 should enhance the intelligibility of DPCM encoded speech signals in the presence of slope overload.

Having discussed some encoding properties of the Scheme 1 algorithm when encoding voiced speech signals, we proceed with the signal-to-noise ratio performance comparison between the First Order and Delayed Scheme 1 DPCM systems. The snr versus input power curves presented in Figures 4.11 and 4.12, were obtained through computer simulation experiments using the programming procedure discribed in section 4.2.2.

The input signal to the encoder is a segment of continuous speech of duration of 2.5 seconds, band limited to 3.4 kHz and sampled at the rate of 8 kHz. The value of a for both systems is equal to 0.85.

Figure 4.11 illustrates the signal-to-noise ratio performance of 4 bits/sample systems. Curve (a) represents the snr of the First Order DPCM system. Curve (b) is obtained from the Delayed Scheme 1 DPCM Scheme when the number of delay units m in the AL register is equal to 3. The value IO of the consecutive samples in overload necessary to activate the Delayed algorithm is equal to 2. Curve (c) is also obtained from the Scheme 1 system but the values for m and IO are four and two respectively. It was found that the "best" value for m is approximately equal to the average number of successive samples in overload and depends upon the rate the input signal is

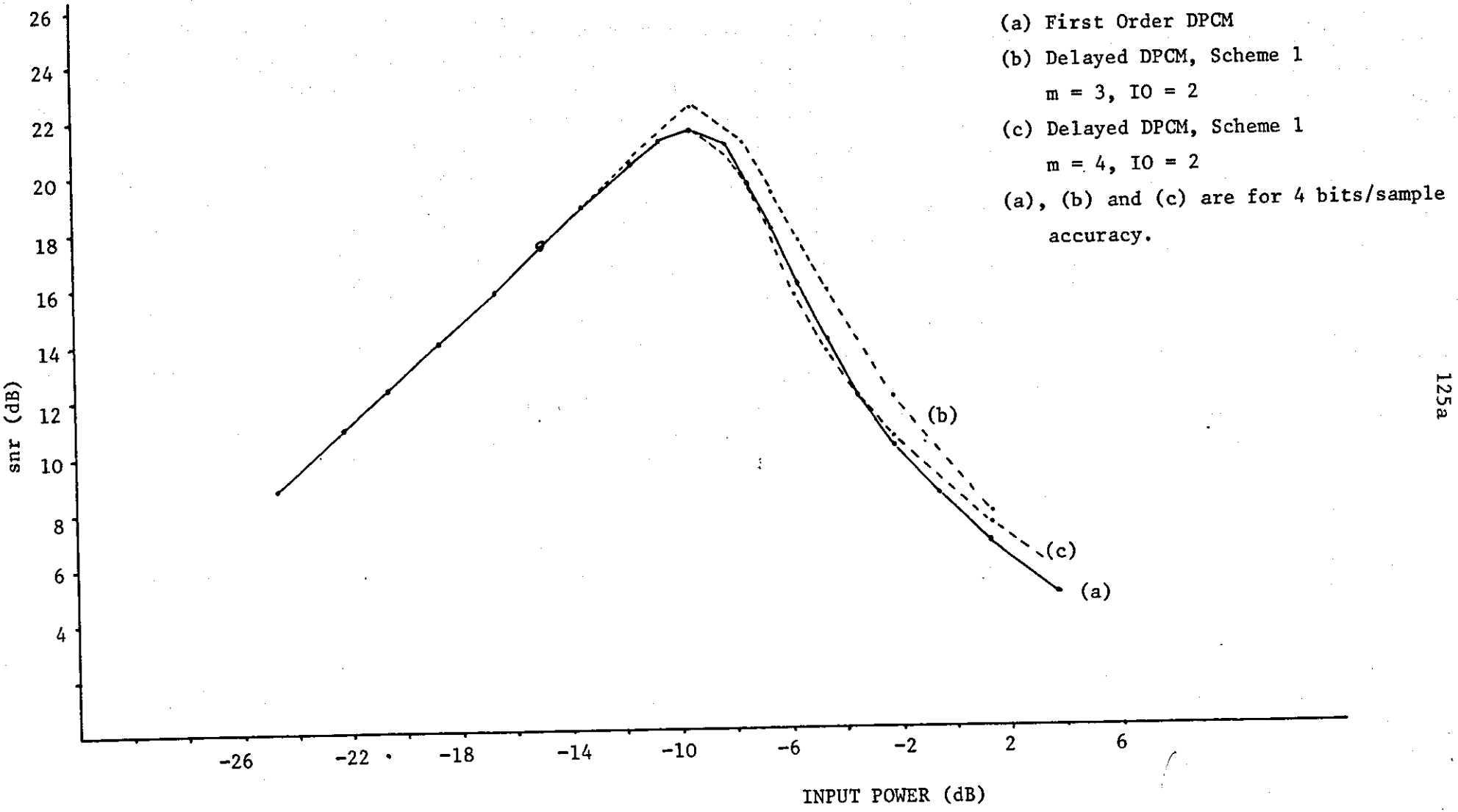


FIGURE 4.11 - snr as a Function of Input Power.

sampled. As the number of successive samples in overload is decreasing, the amount of the granular noise introduced by the algorithm becomes comparable to the reduction achieved in overload noise and the advantage of the Delayed algorithm decreases.

From computer simulation experiments the best value of m was found to be 3. When the value of m was larger than 3, the snr of the system deteriorated rapidly and assumed values lower than the values obtained from the DPCM encoder. On the other hand when m was smaller than 3, the snr improvement was marginal. Also simulations of the system with m = 3 and IO = 1 showed no improvement.

The value of the peak snr advantage of the Delayed Scheme 1 system over the First Order DPCM, is 1 dB increasing to 2dB when severe overload occurs. We found similar improvements in snr with the 3 bits/sample encoders. This is shown in Figure 4.12 where Curve (a) corresponds to the DPCM and Curve (b) to the Delayed DPCM of Scheme 1 with m = 3 and IO = 2. Again, the value of a in both encoders is equal to 0.85.

4.4 DELAYED DPCM, SCHEME 2.

We have seen that the Scheme 1 Delayed algorithm improves the encoding performance of a DPCM system in the presence of slope overload. Its main element, an m-stages shift register delays the samples at the output of the quantizer by m-1 sampling periods. Thus the encoder measures the amount of slope overload (if any) m-1 sampling periods ahead, having as reference in time the quantized error sample stored in the last stage of the shift register. In the presence of overload the value of this sample is appropriately

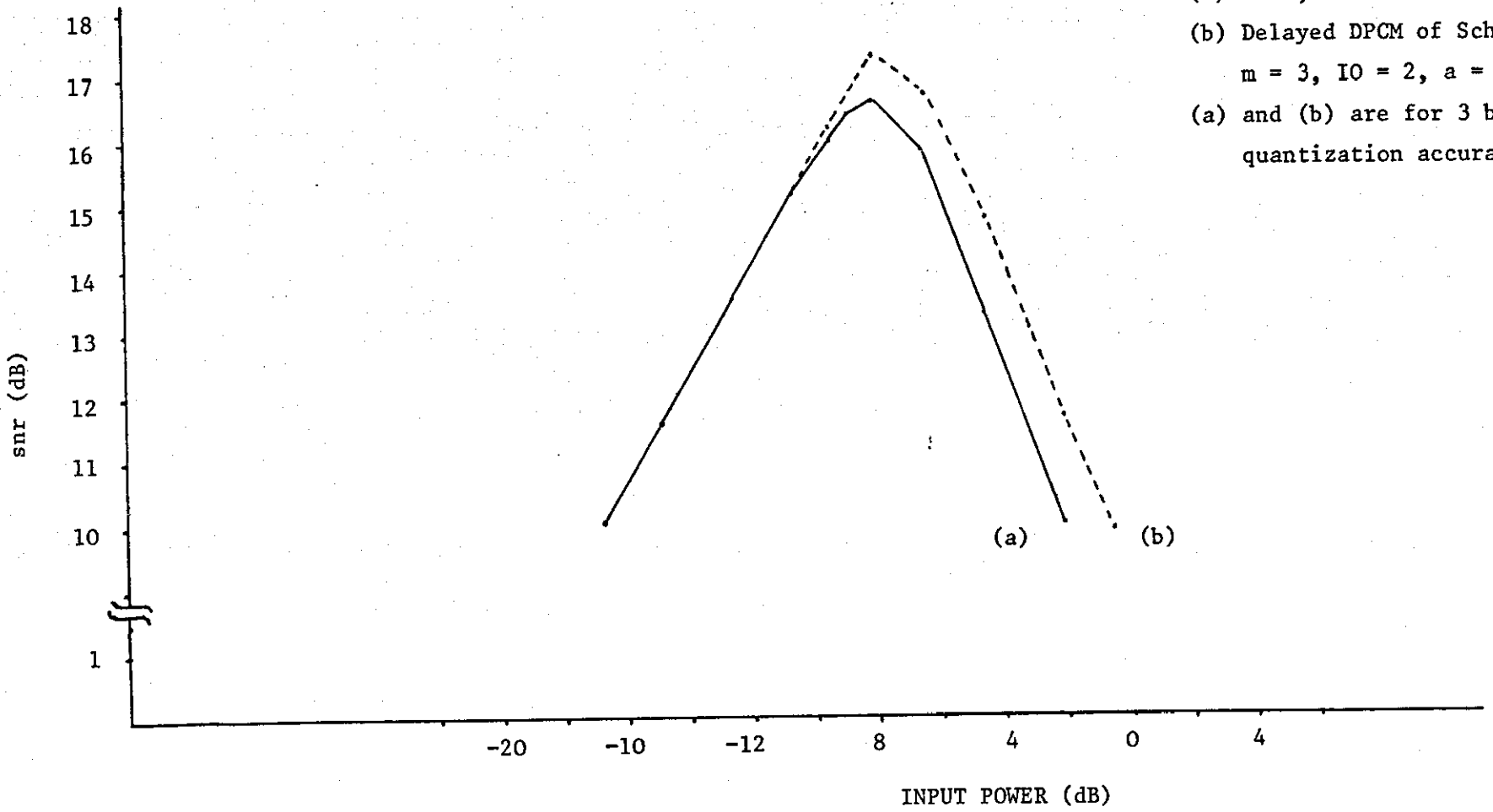

(a) DPCM, a = 0.85
(b) Delayed DPCM of Scheme 1
m = 3, IO = 2, a = 0.85
(a) and (b) are for 3 bits/sample
quantization accuracy.
snr (dB)
18
17
16
15
14
13
12
11
10
1
(a)
(b)
-20
-10
-12
8
4
0
4
INPUT POWER (dB)

FIGURE 4.12.

modified and then transmitted. Although the Scheme 1 system is much simpler when compared with the multipath-search Delayed encoding procedures described in section 4.1, it still requires the use of two different quantizers Q_1 and Q_2.

The next step in our Delayed encoding investigations was to simplify further the algorithm of Scheme 1. That is, to develop a simpler Delayed DPCM system which could provide better or similar results when compared to Scheme 1. The Delayed DPCM of Scheme 2 is such a system. There are two basic differences between the two schemes:

i) In Scheme 2 the slope overload condition is not measured as in the case of Scheme 1, but it is detected by observing the number of successive maximum values at the output of the DPCM quantizer. This is because when the encoder is in slope overload, the output of the quantizer assumes its maximum value for several consecutive sampling periods.

ii) In Scheme 2 the modification of the DPCM error samples is achieved by multiplying their values with a constant coefficient instead of adding to them the quantized value of the slope overload distortion, as in the case of Scheme 1.

Except for the use of only one quantizer in Scheme 2, there is another advantage. That is, when the amplitude of the input signal varies in such a way that the encoder is overloaded for many sampling periods, the Scheme 2 algorithm tracks the input signal better than Scheme 1. This is because when overload is detected, the samples to be transmitted are multiplied at every sampling instant with a constant COEF > 1 coefficient before being

fed into the Local Decoder. Thus the rate of increase of the encoder's Y_n feedback signal is COEF times larger than rate of increase of Y_n in Scheme 1.

4.4.1. Operation of Scheme 2.

The block diagram of the Delayed DPCM Scheme 2 system is illustrated in Figure 4.13. Suppose that $\{X_n\}$is the sequence of input speech samples and that at the nth sampling instant Y_n is the sample produced at the output of the Local Decoder. Y_n is subtracted from the input speech sample X_n to form an error sample E_n. This error sample is quantized by the Q_1 quantizer having a size δ and an input-output characteristic defined by Equations (4.3a), (4.3b) in section 4.3.1. The sample L_n at the output of the quantizer is then fed into a m stage shift register AL.

As mentioned in the previous section an overload condition is detected from the number OV of consecutive samples stored in AL and whose amplitude is that of the outer quantizer levels of Q_1. Thus the "logic" in Figure (4.13) accepts samples from OV stages of the AL register, starting from AL_1, and examines if these samples are of maximum magnitude. When this is true it means that an overload condition is detected and the logic forces the S1 switch in to the A position so that the sample in AL_m is multiplied by the COEF coefficient. When the overload test is proved negative, S1 is switched to the B position and the value of the sample stored in AL_m is multiplied by 1.0.

The two points to note in the Scheme 2 system (Figure 4.13) are:

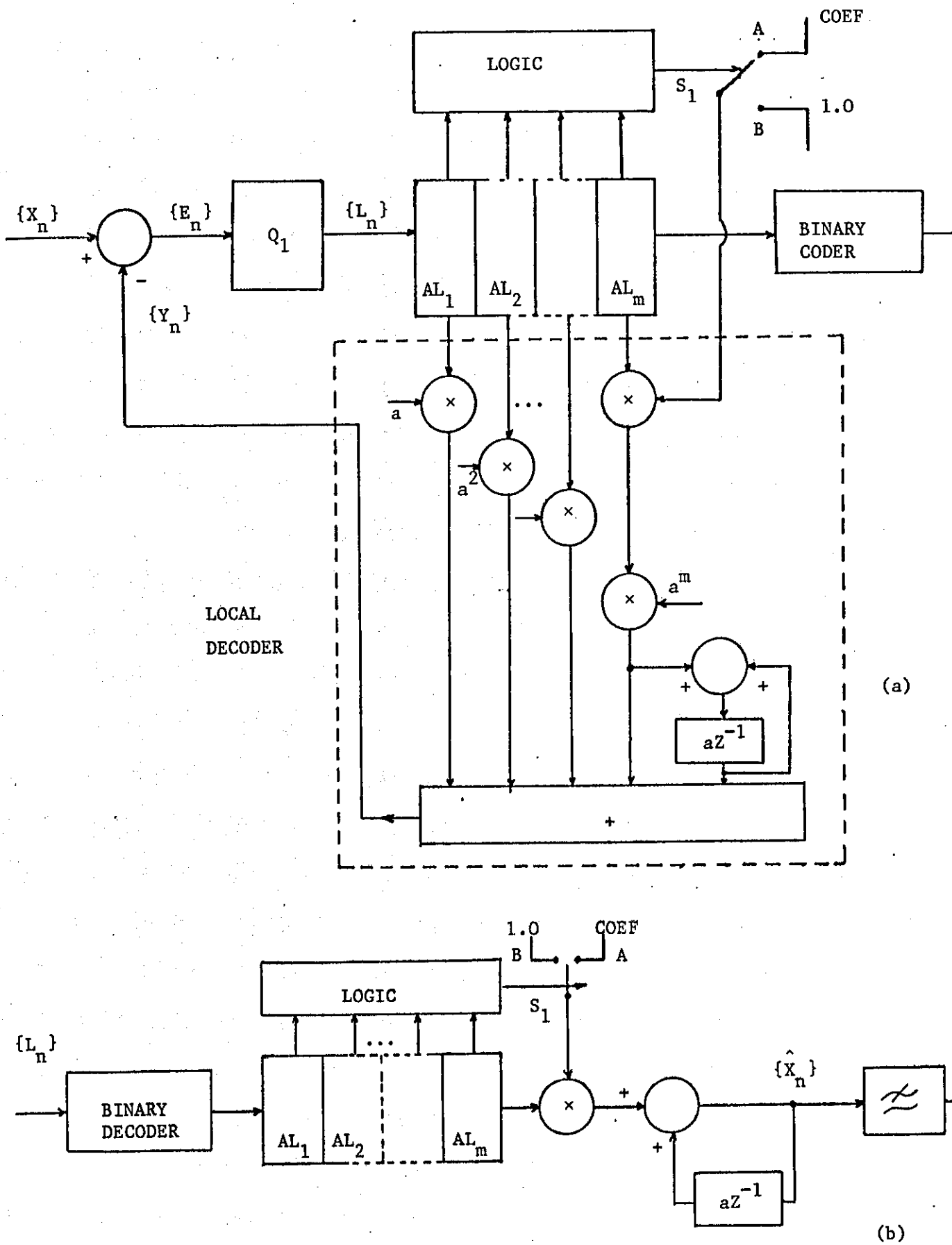


FIGURE 4.13 - The Scheme 2 Delayed DPCM system.
(a) Encoder (b) Decoder.

i) The Local decoder, in contrast with that of Scheme 1, assumes the form shown in Figure 4.13a for both the a = 1 or a < 1 cases. This is because of the way the magnitude of the sample stored in AL_m is modified, i.e. it is multiplied by COEF. Even with a = 1 we cannot apply the decoding Equation (4.6b) and employ the normal DPCM Local decoder shown in Figure 4.4a. Instead the

$$Y_n = \sum_{i=1}^{n-m} a^i L_{n-i} + a^{m+1} \hat{X}_{n-m-1}$$

Equation is used, because only then can the sample stored in AL_m be directly accessed and multiplied by COEF before being used to form the Y_n sample.

ii) The sample to be transmitted is not multiplied by COEF. This means that the set of amplitude values transmitted to the receiver is finite and is defined by the Q_1 quantizer.

The $\{L_n\}$ samples coming out of the AL register are binary coded and transmitted. Assuming an error-free transmission channel, the binary words are received and decoded back to $\{L_n\}$ sequence of samples.

Because of point (ii) mentioned above, the Decoder at the receiving end includes an $\overline{AL}$ m stage shift register and the same "logic" as the one employed by the encoder. In this way, after a delay of m sampling periods, the sample transmitted from AL_m is stored in $\overline{AL}_m$ while the sample in AL_1 is now stored in $\overline{AL}_1$. Consequently, the OV samples available to the logic are the same as those used in the encoding procedure. The logic can test for an overload condition and if found switch S1 to position A, if not S1 remains to position B.

After the sample in $\overline{AL}_m$ multiplied by 1.0 or COEF, it is fed to a normal DPCM decoder to produce the $\{\hat{X}_n\}$ sequence of samples, which is a close approximation of the original input sequence $\{X_n\}$. Finally the $\{\hat{X}_n\}$ samples are low-pass filtered in order to reject the out-of-band quantization noise and to obtain the $\hat{X}(t)$ recovered signal.

4.4.2. Outline of Computer Simulations - Results.

The Scheme 2 Delayed DPCM system has also been simulated on the HP 2100A computer based speech processing system. The input speech data was the same as that used in the Scheme 1 simulations, that is, continuous speech band limited to 3.4 kHz and sampled at the frequency of 8 kHz.

In this section only the Encoding-Decoding simulation is described as the rest of the program has been discussed in section 4.3.2. A flow chart of the Encoding-Decoding procedure is shown in Figure 4.14. At the nth sampling instant the input sample X is fed to the m stages AH shift register. AH is used in the program to compensate for the delay caused by the AL register so that the correct differences between input samples and decoded samples are used in the signal-to-noise ratio calculations. The error sample X1 is then formed as the difference between X and XN · a where XN is the decoded sample at the previous sampling instant. X1 is quantized by the uniform fixed quantizer Q12 subroutine which, except for the quantized output sample Y1, provides an IND variable in its output. The value of IND is equal to unity iff Y1 is the largest magnitude quantization level, otherwise IND is equal to zero. The samples stored in the AL register

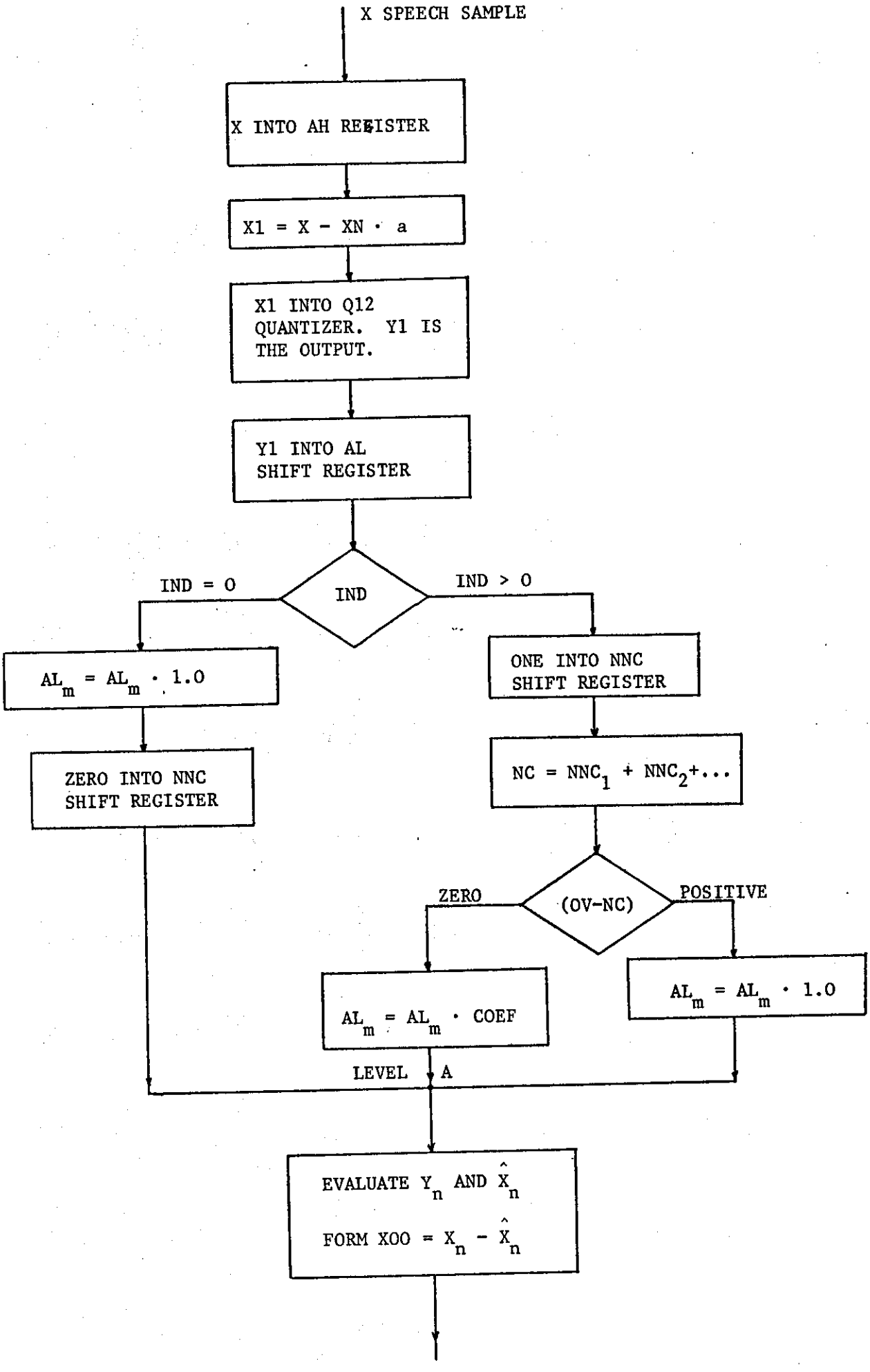


FIGURE 4.14 - Encoding-Decoding Procedure, Scheme 2.

are advanced by one stage and Y1 is fed to the first stage AL_1.

At this point the program examines the value of IND. If IND is zero, that is Y1 is not one of the outermost quantization levels, the sample stored in AL_m remains unchanged. Furthermore, the NNC is clocked once and a zero is stored in its first stage NNC_1. The number of stages in the NNC shift register is equal to OV, i.e. the number of successive maximum Y1 quantization outputs required for an overload detection. In the program, the NNC shift register is used as a part of the "logic" which controls the S1 switch of Figure (4.14). The program then goes to reference level A.

In the case where the value of IND is unity, the NNC shift register is clocked again while "1" is stored in its first stage NNC_1. The contents of the first OV stages of NNC are then added to to give the number NC. If NC is less than OV then the sample stored in AL_m remains the same. If, however, NC is equal to zero then the sample used in the calculations of the Local decoder is equal to that stored in AL_m times COEF.

The reference level A follows in the program where the two separate paths of IND = 0 or IND > 0 merge. Then the following samples are calculated:

i) the Y_n sample of the Local decoder using Equation (4.7.),

ii) the decoded sample $\hat{X}_n$ in the Local decoder,

iii) the decoded sample $\hat{X}_n$ produced from the decoder in the receiving end. (Figure 4.13b).

Finally the last part of the program before going into snr calculations is to form the error samples between the original speech samples and the decoded ones.

We examined the snr performance of the Scheme 2 system using the above programming procedure. The parameters to be determined for the best system's operation are:

i) m, the length of the AL and $\overline{AL}$ registers.

ii) OV, the number of successive maximum magnitude quantization outputs required for the detection of an overload condition.

iii) COEF, the constant which may multiply the samples stored in AL_m and $\overline{AL}_m$. Our approach in determining the best set of parameters was to vary the m and OV and for each combination of m and OV to examine the snr for various values of COEF.

It was observed that when OV assumed values larger than three the signal-to-noise ratio measurements, in the region of the peak snr ($\hat{snr}$) were the same with the snr values obtained from a First Order DPCM. Only when the input signal severely overloaded the encoder were the Scheme 2 snr values better than those for the DPCM system. It was also found that large values of m resulted in a decrease in the systems' snr performance.

For each m, OV set of values, the snr of the encoder improved by increasing the value of COEF starting from unity. The best snr measurements were obtained with COEF between 1.3 and 1.4. Then, a further increase in the value of COEF resulted in a considerable reduction in peak snr and to much lower snr values than those obtained from First Order DPCM.

Figure (4.15) illustrates the snr performance of First Order DPCM and Scheme 2 Delayed DPCM. Both systems were operated with a quantization accuracy of 4 bits per sample, i.e. at a transmission bit rate equal to 32 kbits/sec. Curve (a) is obtained from the First

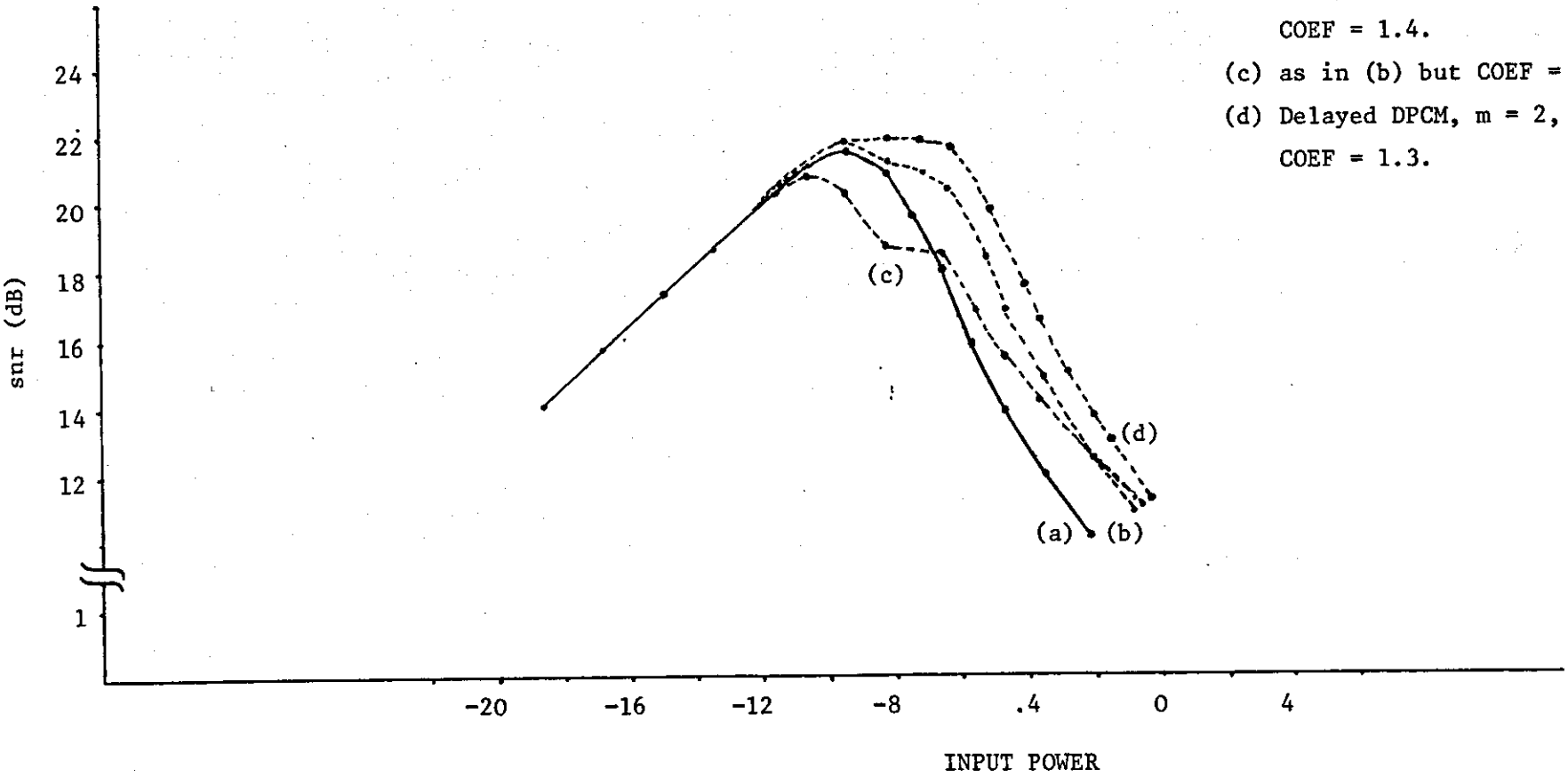


FIGURE 4.15 - snr Performance of First Order DPCM and Scheme 2 Delayed DPCM, 4 bits/sample.

Order DPCM with a = 0.85. Curve (b) indicates the encoding performance of the Scheme 2 system with m = 3, OV = 2 and COEF = 1.4. When the value of COEF is changed to 1.7, while the values for m and OV remain the same, the snr measurements of curve (c) are obtained. From the last two curves we notice the loss of about 1.5 dBs in peak snr because of the increase in COEF beyond the value of 1.4 which, in the m = 3, OV = 2 case, is the optimum one. The snr measurements of curve (d) were obtained from the Delayed DPCM system with m = 2, OV = 2, COEF = 1.3 and represent the best performance Scheme 2 could offer.

As it is shown, from curves (a) and (d), the snr's for both the First Order and Delayed DPCM systems are the same for small values of input power where no overload occurs. When the First Order DPCM shows its peak snr, the Scheme 2 snr is marginally better, i.e. by about .4 dBs. As the power of the input signal increases further the snr of Scheme 2 remains constant while the First Order DPCM is overloaded and its snr is decreasing. However, the constant snr versus input power characteristic is not maintained,

and the snr values of curve (d) starts to decrease with the same rate as in the case of the normal DPCM. For the input power value where this decrease in snr starts to occur, Scheme 2 shows an advantage, over the First Order DPCM, of about 4.5 dBs.

When Scheme 2 employed a 3 bits/sample quantizer, the best m, OV, COEF coefficients found were the same with those in the 4 bits/sample experiments.

Figure (4.16) indicates the 3 bits/sample snr performance of

i) a First Order DPCM with a = 0.85, in curve (a),

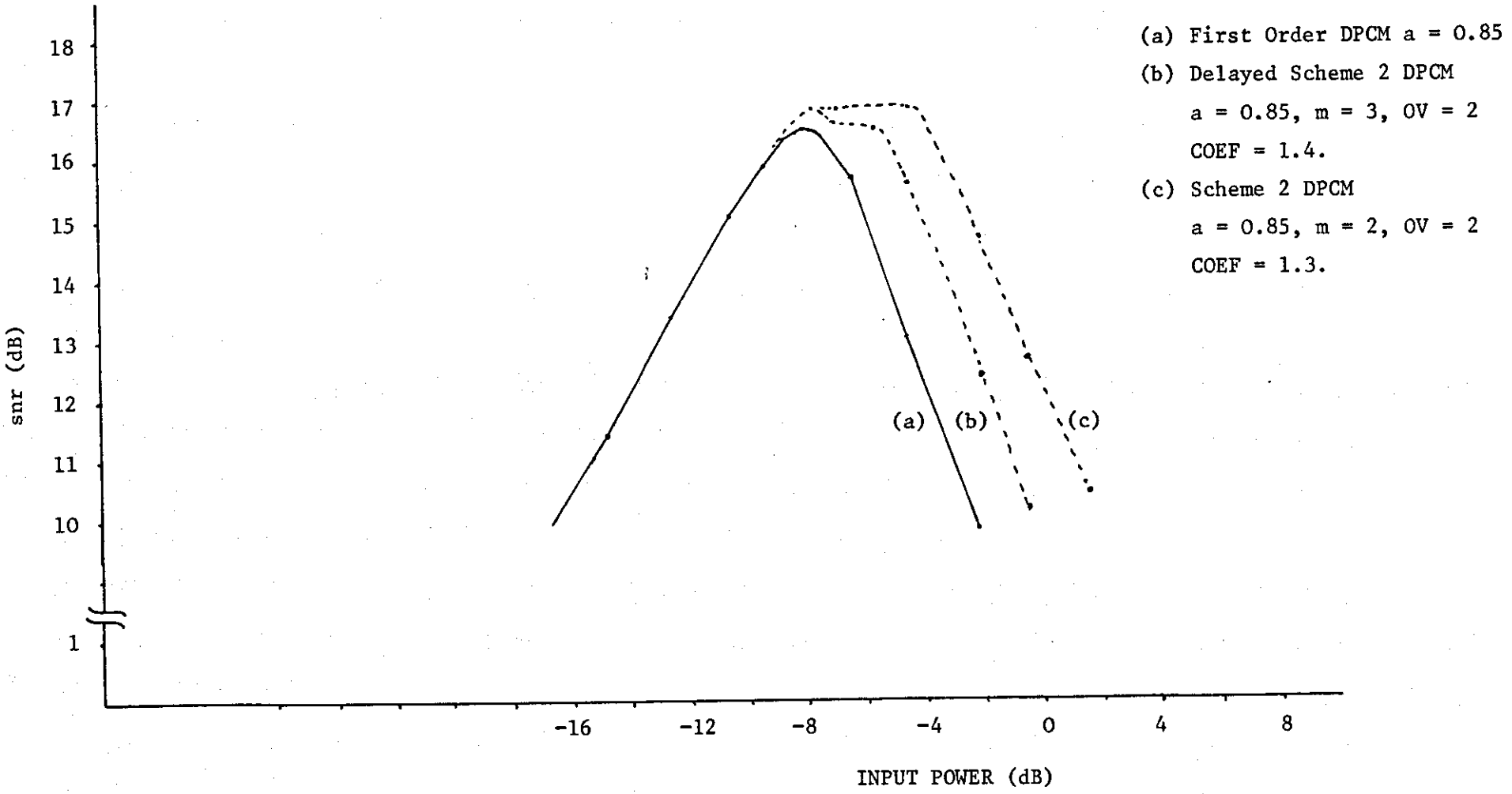

(a) First Order DPCM a = 0.85
(b) Delayed Scheme 2 DPCM
a = 0.85, m = 3, OV = 2
COEF = 1.4.
(c) Scheme 2 DPCM
a = 0.85, m = 2, OV = 2
COEF = 1.3.
snr (dB)
18
17
16
15
14
13
12
11
10
1
(a)
(b)
(c)
-16
-12
-8
-4
0
4
8
INPUT POWER (dB)

FIGURE 4.16.

ii) a Scheme 2 Delayed DPCM with a = 0.85, m = 3, OV = 2, COEF = 1.4, in curve (b),

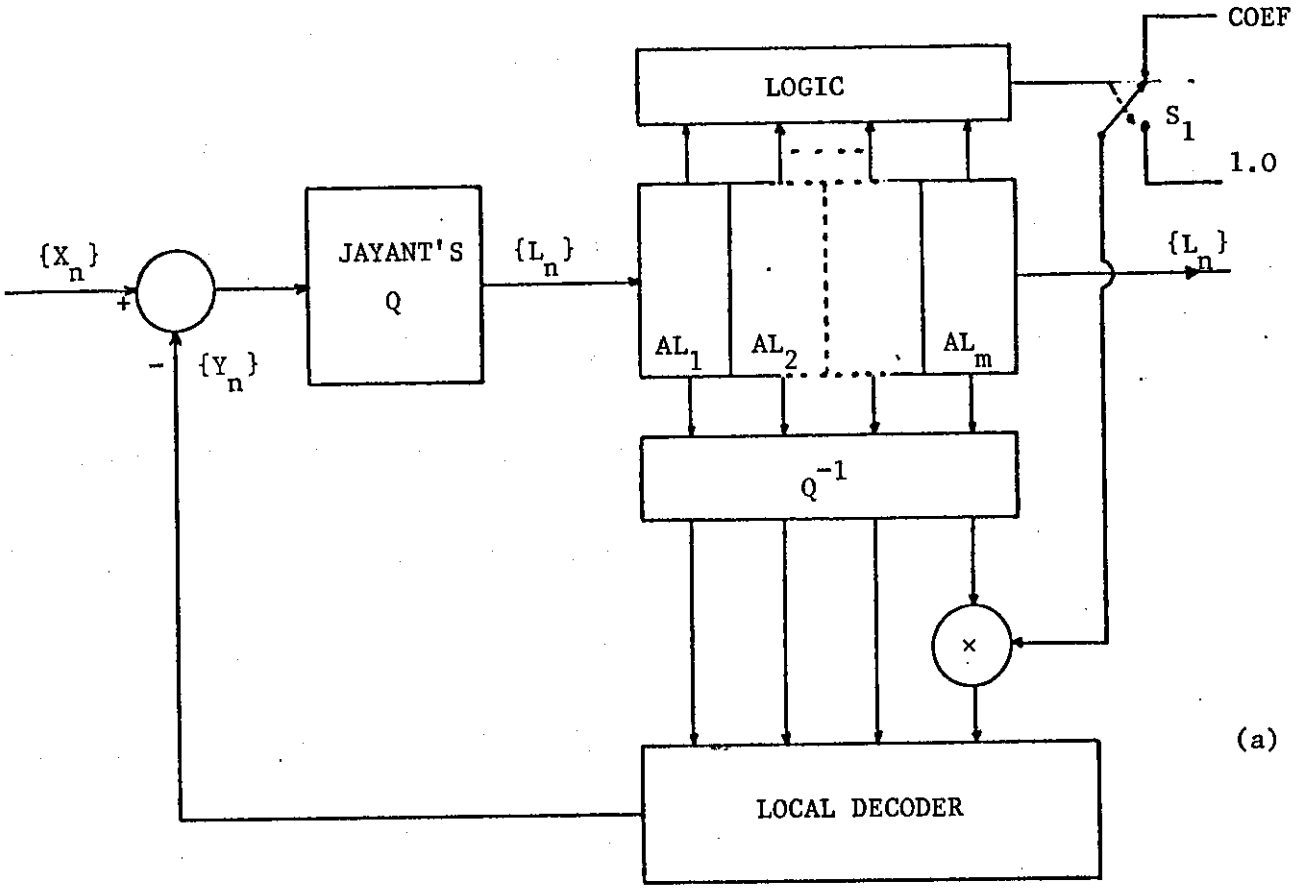
iii) a Scheme 2 Delayed DPCM with a = 0.85, m = 2, OV = 2, COEF = 1.3, in curve (c).

From Figure (4.16) we notice that the snr advantage of the Delayed DPCM when compared to the First Order DPCM, is similar for both the 3 and 4 bits/samples cases.

In order to observe the effect of combining in a DPCM configuration the Scheme 2 delayed algorithm with an adaptive quantizer, we substituted in the Encoding-Decoding procedure of Firgure (4.14) the fixed Q12 uniform quantizer with a Jayant's adaptive quantizer[41]. The block diagram of this Delayed Adaptive DPCM system is shown in Figure (4.17) while its snr behaviour together with that of an adaptive-DPCM and a First Order DPCM is shown in Figure (4.18).

Curve (a) corresponds to a 3 bits/sample First Order DPCM with a = 0.85, while curve (b) is obtained from an Adaptive-DPCM using a Jayant's quantizer with a ratio of maximum to minimum step size $\frac{\delta max}{\delta min} = 128$ and a = 0.85. Curve (c) shows the snr of a Delayed Scheme 2 ADPCM with a = 0.85, $\frac{\delta max}{\delta min} = 128$, m = 3, OV = 2, and COEF = 1.4. Keeping in the latter system, the same coefficient values except m = 2 and COEF = 1.3, curve (d) is obtained. The points to be noticed from Figure (4.18) are:

i) When the fixed quantizer is substituted with the Jayant's quantizer, the resulting ADPCM offers not only an extended Dynamic Range but also improves the peak snr by approximately 2 dBs. (see curves (a) and (b).



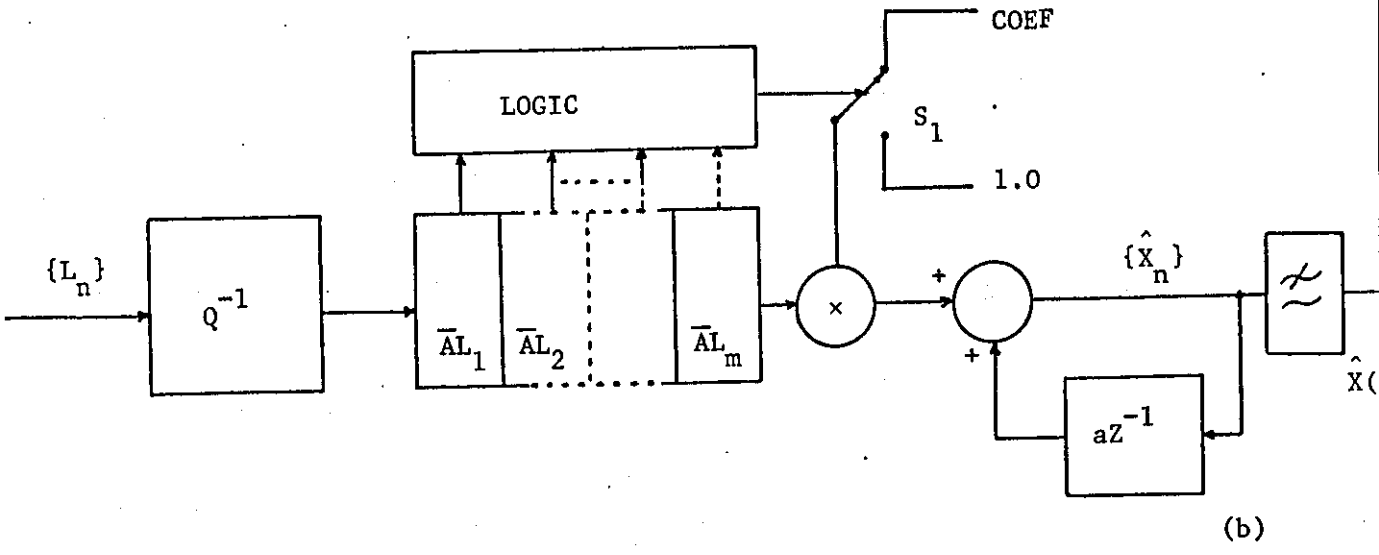
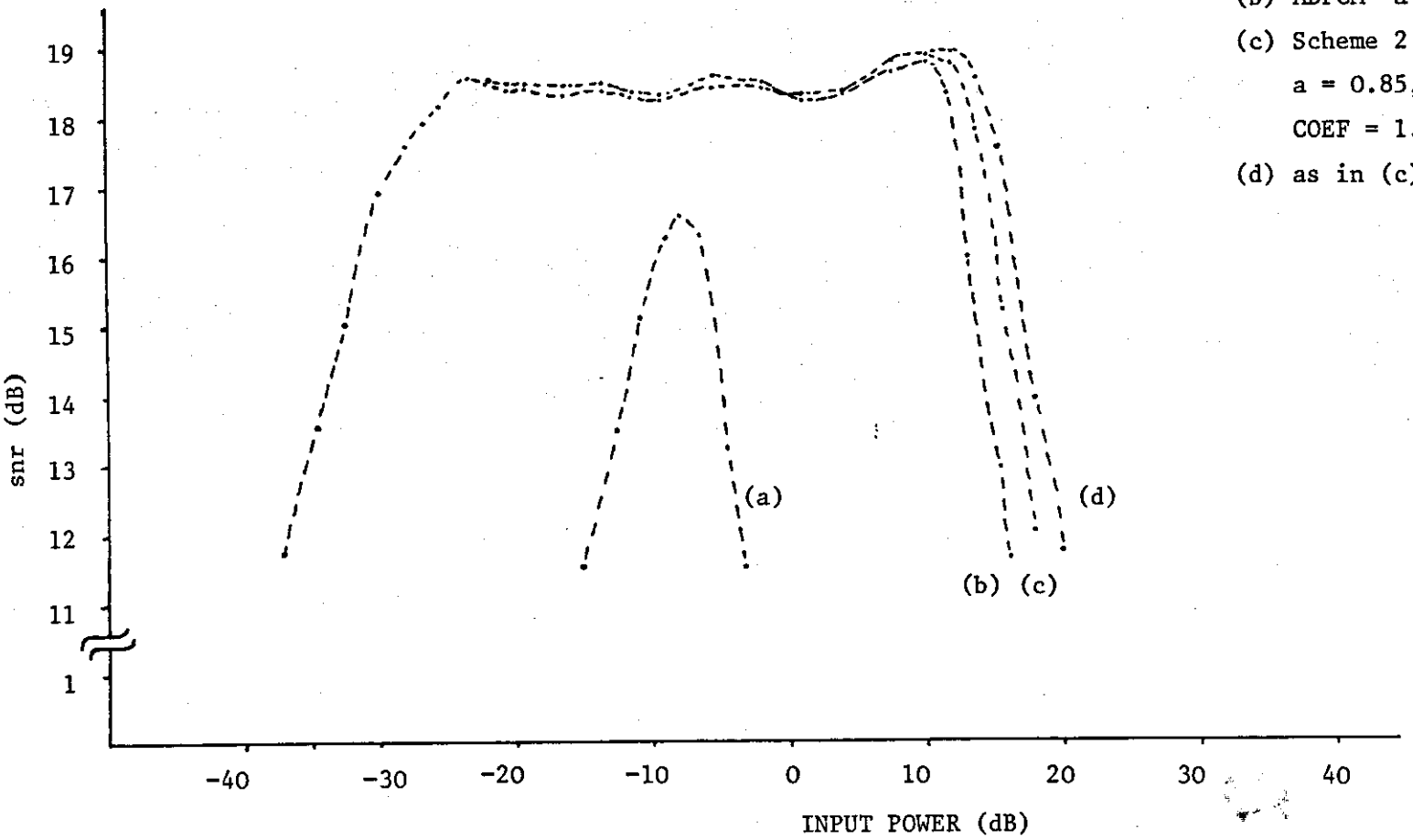


FIGURE 4.17 - Delayed ADPCM, Scheme 2.
(a) Encoder (b) Decoder.



(a) DPCM a = 0.85

(b) ADPCM a = 0.85

(c) Scheme 2 ADPCM
a = 0.85, m = 3, OV = 2,
COEF = 1.4.

(d) as in (c) but m = 2, COEF = 1.3.

FIGURE 4.18.

ii) The peak snr of the Delayed ADPCM system is marginally better when compared to ADPCM. The only improvement occurs in the Dynamic range which is further extended by 2 to 3 dBs (see curves (b) and (d)).

4.5 DISCUSSION

In this chapter we introduced the concept of Delayed encoding and described how this technique could be applied to improve the performance of a Differential encoder. The Delayed multi-path search technique was discussed which can offer the best Delayed encoding improvement because, at each sampling period the optimum encoding path which minimizes a certain error criterion is chosen. This multi-path search algorithm is complex and it is used only with Delta Modulators where the number of possible paths is minimum. Even in this case however, the complexity and cost of implementation is considerable while the advantage obtained in $\hat{snr}$ over the conventional DM system is only about 3 dBs.

Because of this we decided to search for and examine the performance of Delayed encoding methods involving only "single" look-ahead decision, simple to implement algorithms. We developed two such Delayed encoding algorithms and after combining them with DPCM systems we evaluated their snr for various values of input power. First the Scheme 1 system showed a peak snr ($\hat{snr}$) advantage of about 1 dBs when compared to a First Order DPCM. It was found that this improvement remained the same when the systems used 3 or 4 bit quantizers, i.e. their transmission bit rate was 24 or 32 kbits/sec.

Then in an attempt to simplify further the Delayed algorithm, we produced the Scheme 2 Delayed encoder. Its peak snr was only 0.4 dBs better than the $\hat{snr}$ of the First Order DPCM system. However, Scheme 2 system performs better than the Scheme 1 system because of its companding properties. That is, the snr produced from the Scheme 2 encoder remains constant and equal to the $\hat{snr}$ for values of input power where the DPCM and the Scheme 1 DPCM system were overloaded and therefore their snr was considerably reduced. This constant snr region is not extended as in the case of an ADPCM encoder and the snr starts decreasing in value. The reason for this limitation in obtaining constant snr over a large range of input power variations, is the fixed maximum rate with which the feedback samples Y_n can vary their magnitude, i.e. COEF times the maximum magnitude sample at the output of the fixed quantizer, (a = 1). This suggests that a larger dynamic range could be obtained when the coefficient COEF is not constant but adaptive, so the rate of increase in Y_n is not fixed. Thinking along these lines we modified the Scheme 2 algorithm and the procedure which made COEF adaptive was as follows: iff overload is detected for OV consecutive samples then the value of COEF of the nth sampling instant is equal to $\hat{COEF}$, the value of COEF at the n-1 sampling instant times AVA were AVA > 1 is a constant. If an overload condition is not detected or less than OV successive samples are detected in overload, then the value of COEF is equal to $\overline{COEF}$ where $\overline{COEF}$ is a constant.

The adaptation procedure is apparent from Figure 4.14. AL_m is equal to $AL_m \cdot$ COEF where COEF is equal to its previous value $\hat{COEF}$

times AVA, only if (OV-NC) is zero. In all the other cases, i.e. when IND = 0 or when (OV-NC) is positive, the value of AL_m remains the same and in addition COEF assumes its $\overline{COEF}$ value.

Computer simulations of this algorithm showed only a slight increase in the dynamic range when compared with the Scheme 2 system. Furthermore the algorithm proved to be very sensitive to the selection of the values of AVA and $\overline{COEF}$ and frequently developed instabilities. Another modification of the above scheme that is to multiply $\hat{COEF}$ by AVA1 instead of AVA (AVA < 1) when the samples stored in AL_1 and AL_2 are of opposite sign, i.e. when slope overload is over-corrected, failed to produce the extended dynamic range.

Thus, at the end of the Chapter IV computer simulation experiments, we felt that simplified Delayed encoding algorithms could not offer considerable improvement to DPCM systems. Consequently in order to design an efficient DPCM system for encoding speech signals our investigations were directed on the other two important elements of Differential Encoding, that is, the predictor employed in the feedback loop and the quantizer. We started examining first, the "prediction problem" as applied to DPCM and this is the topic of the next chapter.

CHAPTER V

PITCH SYNCHRONOUS DIFFERENTIAL ENCODING OF SPEECH SIGNALS

5.1 INTRODUCTION

In the previous chapter we show the limitation of practical Delayed DPCM encoders to produce a substantial snr advantage when compared to conventional DPCM. It was believed however, that an efficient encoder, whose design constituted the objective of our investigations, would have the form of a Differential encoder employing multi-level quantization and operating at sampling rates above but near the Nyquist rate. Consequently we returned to the conventional DPCM system in order to examine possible modifications which could produce an improved coder.

The operation and the analysis of the DPCM system together with discussions and criticisms regarding various proposed Adaptive-DPCM encoders, have already been presented in Section (2.3.2) of Chapter II. The flow diagram of the codec is however, for the reader's convenience, again illustrated in Figure 5.1. The two main elements which define the encoding performance of the system are the quantizer and the predictor.

The predictor employed in the feedback loop of the encoder have been firstly examined (see Figure 5.1a). In general, the function of a such predictor is to predict the current input sample X_i say, from a weighted combination of recent decoded speech samples:- $\hat{X}_{i-1}, \hat{X}_{i-2}, \ldots, \hat{X}_{i-N}$. Thus the predicted sample Y_i at the output of a Linear predictor is:

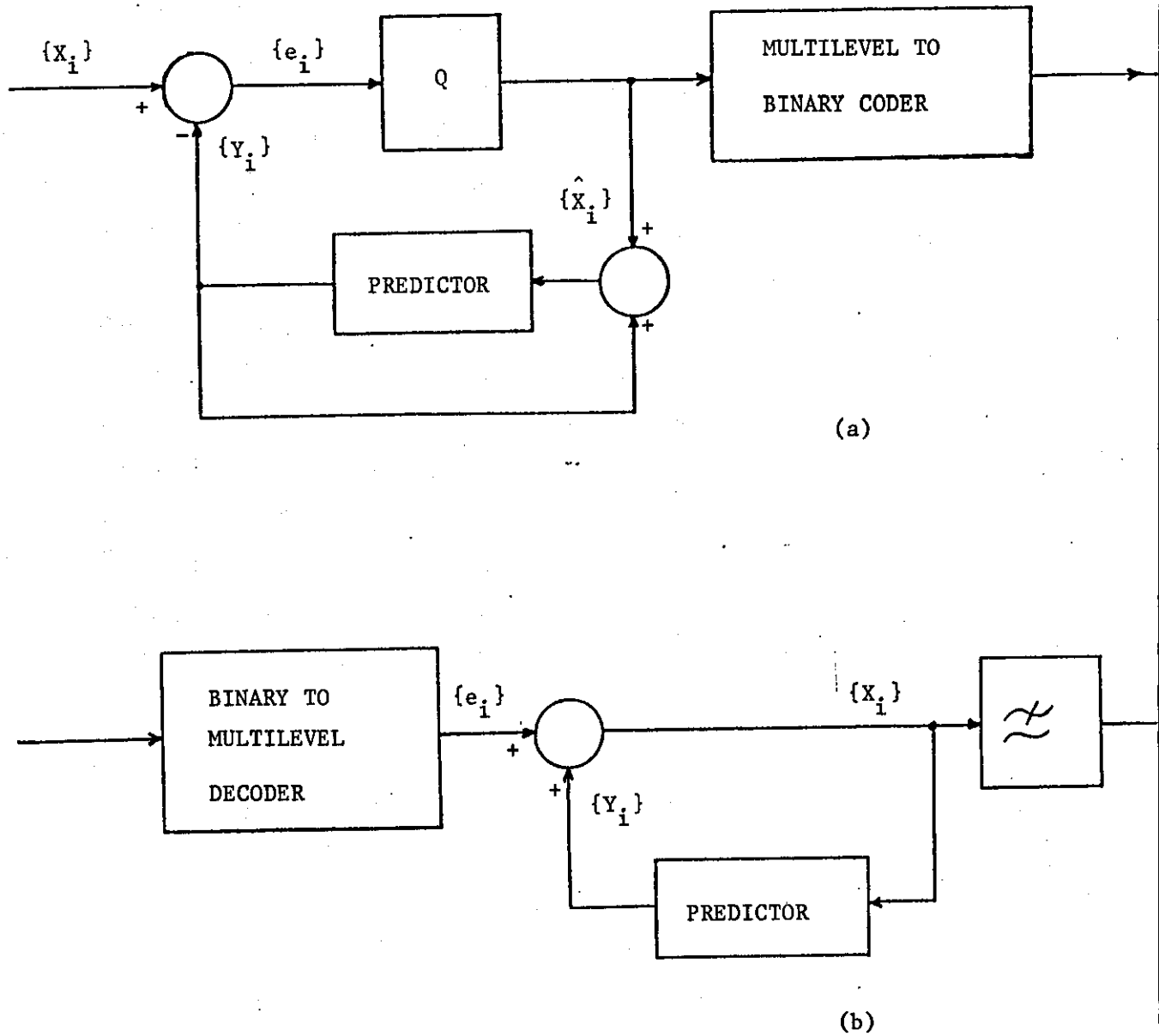


FIGURE 5.1 - The DPCM Codec.

(a) Encoder

(b) Decoder

$$Y_i = \sum_{k=1}^{N} a_k \hat{X}_{i-k} \tag{5.1}$$

The so formed Y_i is subtracted from the input sample X_i and the resulting error sample e_i is then quantized, binary coded and transmitted.

Intuitively, the smaller the prediction error e_i the more accurately it can be represented by a fixed number of quantization levels, which means the smaller the noise q_i, produced from the quantization process. q_i however, also represents the quantization noise of the DPCM encoder as can be seen from the following Equations:

$$e_i = X_i - Y_i \tag{5.2}$$

$$e_i' = e_i + q_i \tag{5.3}$$

$$\hat{X}_i = Y_i + e_i' = X_i - e_i + e_i'$$

$$= X_i + q_i \tag{5.4}$$

Consequently for efficient encoding, accurate prediction of the input samples is a pre-requisite. The same conclusions can be reached by observing the snr Equation as applied to DPCM, i.e.

$$snr_D = \frac{\sigma_x^2}{\sigma_q^2} = \left(\frac{\sigma_x^2}{\sigma_e^2}\right)\left(\frac{\sigma_e^2}{\sigma_q^2}\right) \tag{5.5}$$

where $\sigma_x^2 = E(X_i^2)$, $\sigma_e^2 = E(e_i^2)$, $\sigma_q^2 = E(q_i^2)$ and $E(\cdot)$ is the expected value of $(\cdot)$.

Equation (5.5) can be expressed in decibels as:

$$snr_D = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2} + 10 \log_{10} \frac{\sigma_e^2}{\sigma_q^2}$$

$$= snr(imp) + snr(pcm) \tag{5.6}$$

Equation (5.6) indicates that the signal-to-noise ratio of a DPCM encoder, is the summation of the snr produced by the Q quantizer, i.e. snr(pcm) plus an improvement term snr(imp) which is inversely proportional to the average power σ_e^2 of the prediction error. Thus the smaller the prediction error the larger the value of the improvement term, and the larger the value of the snr_D.

Following this brief analysis which shows the importance of an efficient predictor, we examine the "prediction problem" and the possible types of predictors which can be applied to DPCM. Among the several "paths" opened to research on the subject of DPCM predictors, we will provide the reasons which led us to pitch synchroneous type of prediction and thus to Pitch Synchroneous Differential Encoding of speech signals. Two pitch synchroneous differential encoding systems will then be presented which show significant performance improvement over conventional DPCM and ADPCM codecs.

5.2 THE "PREDICTION PROBLEM"

In order to understand "prediction" as used in DPCM three known techniques have been examined. Their estimation accuracy have been observed, through computer simulations, when:

i) original speech samples were used as their input signal,

ii) the predictors were included in ADPCM systems and decoded speech samples formed their input signal.

As a result of these experiments and having in mind the existing work on DPCM prediction techniques[62,63], we found some questions yet to be answered on this subject and further decided the type of predictor which is probably best suited for Differential encoding applications.

5.2.1. Prediction Techniques.

This section presents the various prediction methods which can be applied to DPCM encoding of speech signals. Equation (5.1) represents a Linear predictor and it is the one usually employed in DPCM. It is easy to see from this Equation, that the accuracy of the predictor in estimating the input samples depends upon the selection of the proper a_k weighting coefficients. In Chapter III, section (2.3.2.1) we derived the optimum values of the predictor's coefficients a_k for a stationary input signal. These were obtained by setting the partial derivatives of the error power function, with respect to the a_k's, equal to zero. The Equation which define the optimum coefficients is of the form:

$$\begin{bmatrix} E(X_i\ X_{i-1}) \\ \vdots \\ E(X_i\ X_{i-j}) \\ \vdots \\ E(X_i\ X_{i-N}) \end{bmatrix} = \begin{bmatrix} E(X_{i-1}\ X_{i-1}) & \cdots & E(X_{i-1}\ X_{i-N}) \\ E(X_{i-1}\ X_{i-2}) & E(X_{i-2}\ X_{i-2}) & \cdots \\ \vdots & \vdots & \vdots \\ E(X_{i-1}\ X_{i-N}) & & E(X_{i-N}\ X_{i-N}) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}$$

while the optimum coefficient's vector A is given by:

$$A_{opt} = R^{-1} \cdot G \tag{5.7}$$

where R and G are the N·N autocorrelation matrix and the Nth order autocorrelation vector respectively. (see Equation (2.26)).

In the case where the predictor is operating on speech samples, there are several methods to define the a_k coefficients. The first one to mention, measures the long-term autocorrelations function, i.e., an average autocorrelation function obtained from many speech sentences which are sampled at the same rate. Then the a_k coefficients are calculated using Equation (5.7). In this way the predictor is designed to match the long-term statistics of the speech. Because the a_k coefficients are fixed such predictor is known as a fixed spectrum or a time invariant one.

Speech however is not a stationary signal with fixed statistics. Intuitively we expect that an adaptive predictor which can follow the statistical variations of the speech signal would perform better than a fixed predictor. There are basically two techniques in updating the a_k coefficients of an adaptive predictor, the "block"

adaptation and the "sequential" adaption techniques.

a) In the "block" adaptation method the short-term autocorrelation function of a "block" of speech samples is measured.

The coefficient's vector A is then obtained from Equation (5.7). Usually the length of the segment of speech whose statistics are measured is larger than the expected maximum pitch period. The procedure is repeated every 4 or 5 mS. and the prediction coefficients are kept constant for this time interval until new values are calculated from the next segment of samples. An overlapping between the "analysis" segments is also allowed.

When applying the "block" adaptation method in a DPCM system, the analysis which measures the speech statistics can be performed, i) on the incoming original speech samples and, ii) on the decoded speech samples. In the first case the estimation procedure is said to be a "Forward" one while in the second case we have the "Backward" adaptation procedure.

Between these two methods the "Forward" method is the more accurate because the "analysis" is performed on a segment of speech samples and the a_k coefficients are used by the DPCM encoder to encode the same segment of speech samples. Consequently, successive speech segments are encoded while the predictor is using the optimum a_k coefficients for each segment.

In the Backwards method the speech samples used in the "analysis" procedure are the decoded ones, and the obtained a_k coefficients are employed in the encoding of the next segment of input samples. Thus during the encoding of a block of speech samples, the a_k coefficients used are the optimum ones for the previous decoded

segment. The method operates well when the statistics of the speech signal vary slowly and when the analysis segment is not larger than that of the Forward method.

The advantage of the Backward method is that no separate channel is required for the transmission of the a_k coefficients as these are calculated from already decoded samples which are also available at the receiver. In contrast, the optimal prediction coefficients of the Forward method, are transmitted to the receiver, together with the speech information, so both the predictors at the transmitting and receiving ends are operating employing the same vector A.

b) The second technique in updating the a_k coefficients of an adaptive predictor is the "sequential" one where the coefficients are re-calculated at every sampling instant. The information used in updating the a_k's is available in both the transmitter and receiver, without the need of transmitting separate information except the output of the DPCM quantizer. The penalties in saving transmission bandwidth are:

i) the coefficient's adaptation is performed using the quantized value e_i' of the DPCM error sample. Consequently the look-ahead to compute the optimal a_i's for the X_i input sample is not allowed, as in the case of the Forward block prediction procedure.

ii) the error sequence $\{e_i'\}$ contains additive, not necessarily uncorrelated, quantization noise which can cause the a_k coefficients determined from this sequence to be biased away from their optimal value and to fluctuate even for a stationary sound. The rougher

the quantization the more the degradation in the estimation accuracy of the predictor.

One simple and rather popular method in sequentially updating the Linear predictor's coefficients is the steepest descent gradient search technique[117] which minimizes the mean squared prediction error. In this method, at the k+1 sampling instant, the jth a_j coefficient assumes its new value according to:

$$a_{k+1}(j) = a_k(j) - g \frac{\partial e_k'^2}{\partial a_k(j)} \tag{5.8}$$

where $$\frac{\partial e_k^2}{\partial a_k(j)} = -2\hat{X}_{k-j}\, e_k' \tag{5.9}$$

i.e. the prediction coefficients are updated in a direction opposite to the gradients given by Equation (5.9).

From Equations (5.8) and (5.9) the updating algorithm is therefore

$$a_{k+1}(j) = a_k(j) + g\, \hat{X}_{k-j}\, e_k' \tag{5.10}$$

The convergence of Equation (5.10) towards the optimal prediction coefficients A_{opt}, as defined in Equation (5.7), can be easily proved as follows.

Let us assume that the optimal set of coefficients A_{opt} was used by the predictor and the associated error sample is $\underline{e}_k$. The difference between $\underline{e}_k$ and e_k is formed, where e_k is the prediction error at the kth sampling instant with the predictor using the a_k's derived from Equation (5.10). i.e.

$$\underline{e}_k - e_k = (A_{opt} - A_k)^T \bar{\hat{X}}_k$$

or $$e_k = -(A_{opt} - A_k)^T \bar{\hat{X}}_k + \underline{e}_k \tag{5.11}$$

$\bar{\hat{X}}_k$ is a column vector whose elements are the N previous decoded speech samples $\hat{X}_{k-1}, \hat{X}_{k-2}, \ldots, \hat{X}_{k-N}$.

Next a difference vector γ is defined as

$$\gamma = A_{opt} - A_k \tag{5.12}$$

and from Equations (5.10), (5.11) and (5.12) we have

$$\begin{aligned}\gamma_{k+1} &= \gamma_k + g\,\bar{\hat{X}}_k\, e_k \\ &= \gamma_k + g\left[-(A_{opt} - A_k)^T\,\bar{\hat{X}}_k + \underline{e}_k\right] \cdot \bar{\hat{X}}_k \\ &= \gamma_k - g\left[(\gamma_k^T\,\bar{\hat{X}}_k).\bar{\hat{X}}_k - \underline{e}_k\,\bar{\hat{X}}_k\right] \end{aligned} \tag{5.13}$$

The convergence of A_k forwards A_{opt} becomes clear by taking the sum of the squares for all the vector components in Equation (5.13), i.e.

$$\begin{aligned}||\gamma_{k+1}||^2 = ||\gamma_k||^2 &- 2g\left[(\gamma_k^T\,\bar{\hat{X}}_k)^2 - \underline{e}_k\,\gamma_k^T\,\bar{\hat{X}}_k\right] + \\ &+ g^2\left[(\gamma_k^T\,\bar{\hat{X}}_k)^2 - 2\underline{e}_k\,\gamma_k\,\bar{\hat{X}}_k + \underline{e}_k^2\right]\bar{\hat{X}}_k{}^2\end{aligned}$$

When the value of g is sufficiently small the last Equation becomes:

$$||\gamma_{k+1}||^2 \simeq ||\gamma_k||^2 - 2g\left[(\gamma_k^T\,\bar{\hat{X}}_k)^2 - \underline{e}_k\,\gamma_k^T\,\bar{\hat{X}}_k\right] \tag{5.14}$$

and if $|\underline{e}_k|$ is small compared to the $|\gamma_k^T\,\bar{\hat{X}}_k|$ i.e. $|\underline{e}_k|$ is small compared to the $|e_k|$ error, we have

$$||\gamma_{k+1}||^2 = ||\gamma_k||^2 - 2g\,(\gamma_k^T\,\bar{\hat{X}}_k)^2 \tag{5.15}$$

By the Schwartz inequality and because $g << \frac{1}{2}$ we could say that

$$||\gamma_{k+1}||^2 < ||\gamma_k||^2 \tag{5.16}$$

Hence when $|\underline{e}_k| << |\gamma_k^T \bar{\hat{X}}_k|$, the vector A adjusts its components towards the value of A_{opt}. When the A_{opt} solution is approached by A, the process is slowed down and $||\gamma_{k+1}||^2$ can be larger than $||\gamma_k||^2$, provided that $|\underline{e}_k| > |\gamma_k^T \bar{\hat{X}}_k|$ and $\text{sgn}(\underline{e}_k) = \text{sgn}(\gamma_k^T \bar{\hat{X}}_k)$ (see Equation (5.14)).

The adaptation algorithm of Equation (5.10) assumes that g is a small constant ($g << 1$) and this limits its performance since e_k and $\hat{X}_{k-j}$ are related to the overall signal level. Thus g is made inversely proportional to the speech power and Equation (5.10) takes the form:

$$a_{k+1}(j) = a_k(j) + \left(\frac{g_o}{M + \frac{1}{n} \sum_{i=1}^{N} \hat{X}_{k-i}^2} \right) \hat{X}_{k-j}\, e'_k \tag{5.17}$$

The denominator of the term inside the brackets behaves as an automatic gain control which tends to equalize the adaptation rate of the algorithm to a mean square value computed over the N past decoded samples. Thus as the power of the speech increases the second term of Equation (5.17) is reduced and overcorrections of the a_k coefficients are avoided preventing the occurrence of a large prediction error. M is a constant and a bias term added to reduce the value of the term in brackets, during silent intervals, and prevent possible oscillations. The term g_o is an optimizing constant ($g_o << 1$).

Another method of sequentially adapting the coefficients of a Linear predictor is the modified Kalman filter procedure[20]. This adaptation algorithm is certainly more complex than that of Equation (5.17) but, it is also more accurate in estimating the speech signal. The a_k coefficients are updated as follows:

$$A_{k+1} = A_k + K_{(k)}\, e'_{(k)} \tag{5.18a}$$

$$K_{(k)} = \frac{\overline{Va}_{k-1}\, \bar{\hat{X}}_{k-1}}{\bar{\hat{X}}^T_{k-1}\, \overline{Va}_{k-1}\, \bar{\hat{X}}_{k-1} + V} \tag{5.18b}$$

$$\overline{Va}_{k+1} = \left[I - K_{(k)}\, \bar{\hat{X}}_k\right] \overline{Va}_k \tag{5.18c}$$

where $\bar{\hat{X}}_k = \left[\hat{X}_{k-1}, \hat{X}_{k-2}, \ldots, \hat{X}_{k-N}\right]^T$ and V is a bias term similar to M of Equation (5.17). The $\bar{\hat{X}}^T_{k-1}\, \overline{Va}_{k-1}\, \bar{\hat{X}}_{k-1}$ quantity acts as an automatic gain control and limits the coefficients from being overcorrected when the amplitude of the speech signal is large. $\overline{Va}_{k-1}$ is proportional to the estimation error obtained from the algorithm[20].

Until now we have discussed techniques for adapting the prediction coefficients to the varying statistics of the speech signal. The structure of the predictor was defined by Equation (5.1) i.e. a linear predictor has been assumed. The predictor, however, can take another form, that of a Lattice filter whose prediction coefficients b_i can also be updated using the above sequential techniques. The Lattice filter derived by Itacura and Saito[22] has been used primarily in vocoder type systems. The filter is

reproduced in Figure 5.2 together with its associate inverse filter. Its main characteristic is that the redundancy of the input signal is removed successively at each of the cascaded stages of the filter. Thus the b_i coefficient, at the ith stage, is optimized to minimize the e_{i+1} output and in this way the final output e_{n+1} is of minimal energy.

The sample e^k_{n+1} at the output of the filter at the kth sampling instant is given by

$$e^k_{n+1} = X_k - \sum_{i=1}^{n} b_i F_i \tag{5.19}$$

and consequently the output Y_k of a Lattice predictor is equal to

$$Y_k = \sum_{i=1}^{n} b_i F_i \tag{5.20}$$

When the Lattice predictor is to be used in a Differential encoder the F_i samples are replaced by their received version $\hat{F}_i$ and the prediction Equation takes the form of:

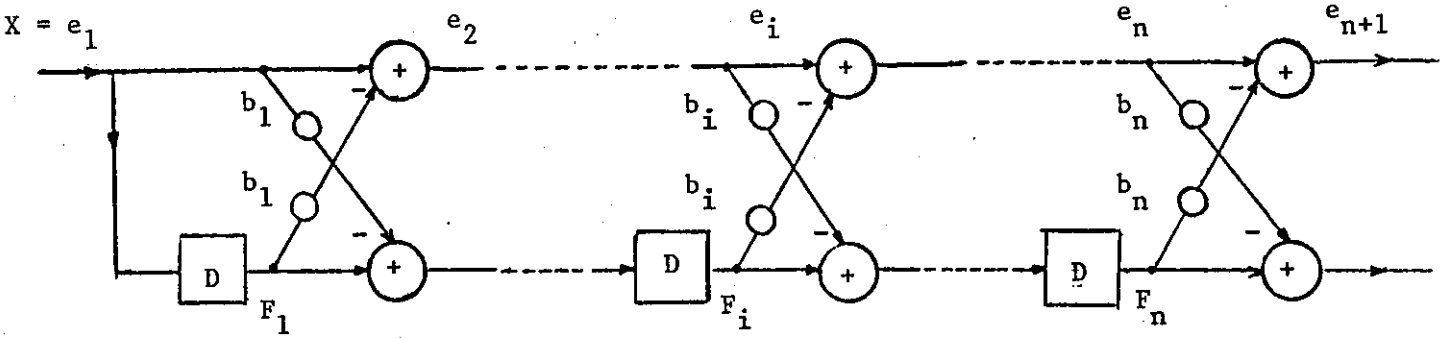
$$Y_k = \sum_{i=1}^{n} b_i \hat{F}_i \tag{5.21}$$

A Differential encoder employing the Lattice predictor is shown in Figure 5.3.

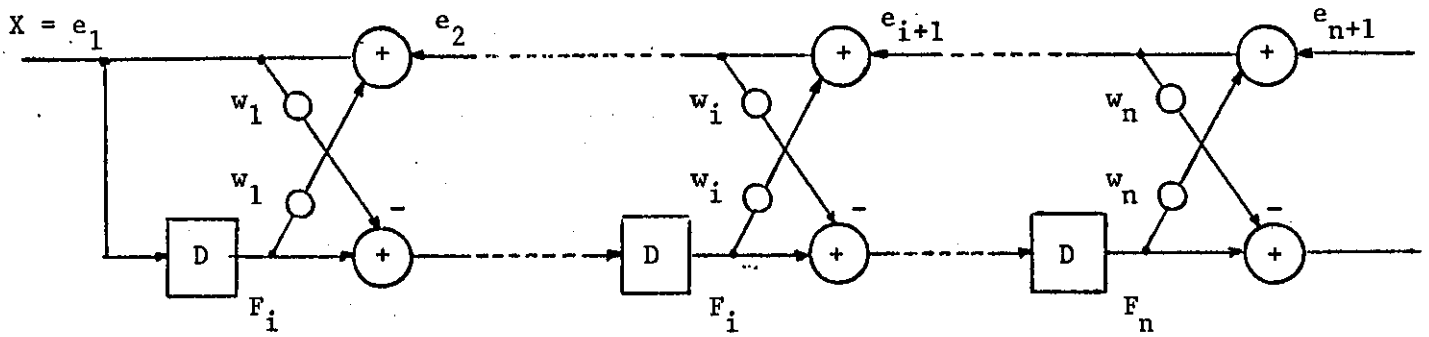
5.2.2. Estimation Performance of Three Prediction Methods.

We now consider the performance of computer simulated Linear predictors which employ three different methods to determine the a_k coefficients.

a) The block adaptation method where the short term



(a)



(b)

FIGURE 5.2 - (a) The Maximum likelihood Vocoder Filter, (b) Its Inverse Filter.

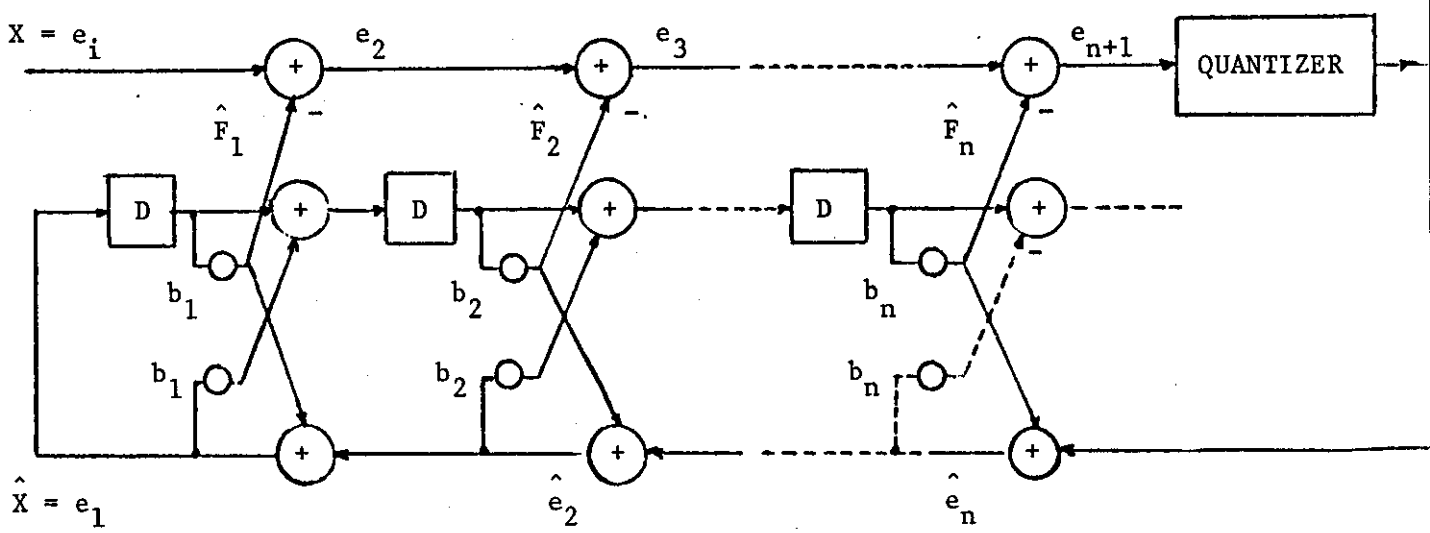


FIGURE 5.3 - A Differential Encoder using the Lattice Predictor.

autocorrelation function of successive segments of speech samples is measured. The optimum coefficients are then obtained from Equation (5.7). The segment for measuring the autocorrelation function was kept constant and contained 120 speech samples. An overlapping of the order of 60 samples between segments was also allowed. Thus the A vector was updated every 60 samples.

b) The sequential adaptation method of Equation (5.17) which is also called the "Stochastic Approximation" algorithm. Because the original speech samples are used as the input signal, $\hat{X}_{k-j}$ and e'_k were substituted by X_{k-j} and e_k respectively. The values used, during the computer experiments, for the M and g_o constants were 100 and 10^{-4}.

c) The time-invariant method where the prediction coefficients are fixed and calculated from the long-term autocorrelation function given by McDonald[53].

Method number one showed the best modelling of the vocal tract characteristics and consequently produced the smaller error $E(X_i - Y_i)$ between the input speech samples and their estimates. The signal-to-noise ratio in dBs, defined as

$$snr = 10 \log_{10} \frac{E(X_i^2)}{E\left[(X_i - Y_i)^2\right]} \qquad (5.22)$$

was found to be of the order of 19 dBs. It was observed that the accuracy of obtaining a set of prediction coefficients 'A' which closely modelled the vocal tract characteristics, depends upon

the position of the excitation pulse inside the analysis segment.

The Stochastic Approximation algorithm produced a maximum snr of approximately 13 dBs. It was found that the performance of the algorithm was dependent upon the power of the input signal, unlike the block adaptation method where the snr of 19 dBs was obtained for any input power. From the simulation it became evident that the reason for not producing a constant snr over a wide range of input power values, is the bias M in the denominator of the gain factor (Equation 5.17). For very small values of the input power, M becomes considerably larger than $\frac{1}{n} \sum_{j=1}^{N} X_{k-j}^2$. Thus the gain factor inside the brackets in Equation (5.17) is more or less constant (<< 1) and not-varying with the amplitude variations of the input, which decreases considerably the estimation accuracy of the predictor. Consequently the 13 dBs mentioned above is the value of the peak snr obtained from the predictor. The rate the snr decreases from its peak value was found to depend on the number of samples used to form the normalized power term added to M. In Equation (5.17) this number is equal to N, i.e. the order of the predictor. However, if N is substituted by another variable, say N_2, then it was observed that:-

i) By making the value of N_2 equal to the length of the average pitch period expected in the input speech signal, the dynamic range of the algorithm is reduced while its peak snr increases.

ii) When the value of N_2 is reduced and it is considerably smaller than the average pitch period, for example, $N_2 = 12$, then the value of the peak snr decreases while the dynamic range of the

algorithm increases.

The fixed coefficients predictor provided the smallest snr compared to the other two methods. The snr of a fixed one coefficient predictor (a_1 = 0.85) was found to be of the order of 8.5 dBs. When the number of prediction coefficients increased to four, the snr showed variations with different speech sentences used as the input signal. The maximum snr obtained for a fourth order fixed predictor was of the order of 11.5 dBs. The use of higher order predictors showed a small snr advantage. Figure 5.4 illustrates the behaviour of these three predictors when operating on a segment of voiced speech, shown in 5.4a. The error waveform between the original signal and the predicted one, when the predictor employs the block adaptation method, is shown in Figure 5.4b. The waveform in 5.4c corresponds to the error produced from the Stochastic Approximation algorithm, and finally the last error waveform in 5.4d is produced from a fixed single coefficient predictor with a_1 = 0.85.

All the three prediction techniques were then successively employed in an adaptive DPCM encoder whose adaptive uniform quantizer followed Jayant's adaptation procedure[41]. The signal-to-noise ratio values for 2, 3 and 4 bits per sample quantization accuracy are illustrated in Figure 5.5, when the input signal is 2.2 seconds of continuous speech band-limited at 3.4 kHz and sampled at the frequency of 8kHz.

The ADPCM-FW system using the Forward Block adaptation prediction technique found to provide the higher snr, compared to the other systems, for all the 2, 3 and 4 bits per sample

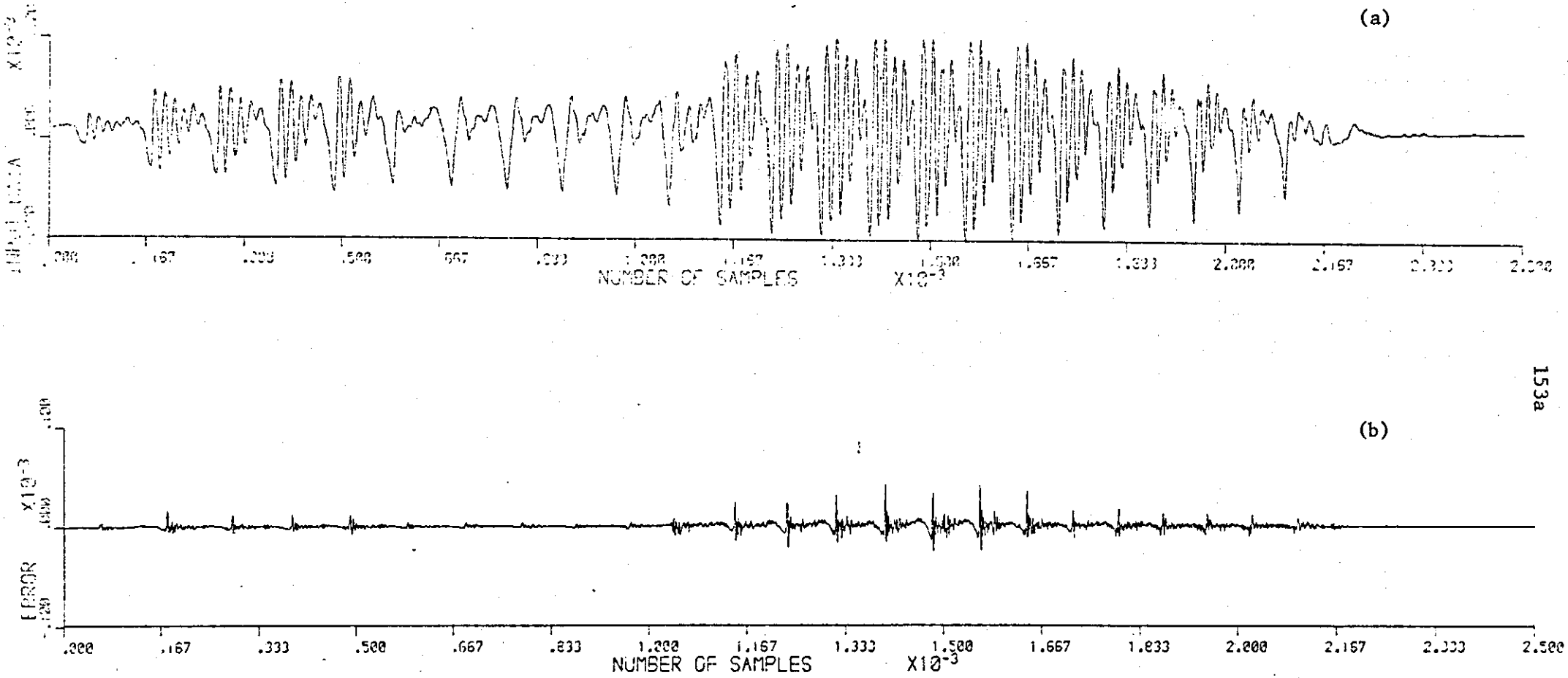


FIGURE 5.4 - (a) Input Speech Waveform.

(b) Error Waveform using a Block Adaptive Predictor.

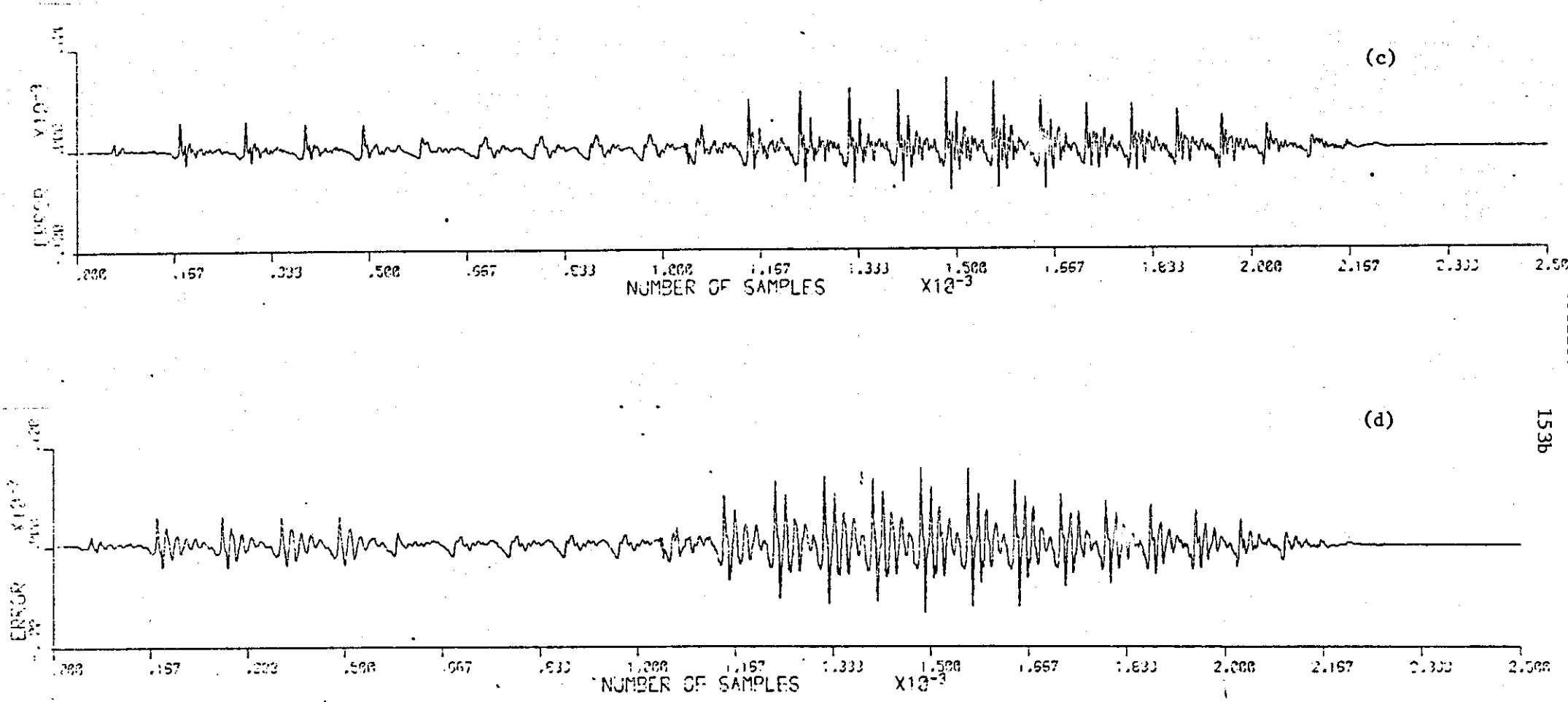


FIGURE 5.4 - (c) Error Waveform using a Stochastic Appr. Predictor.
(d) Error Waveform using a Fixed First Order Predictor $\alpha = 0.85$.

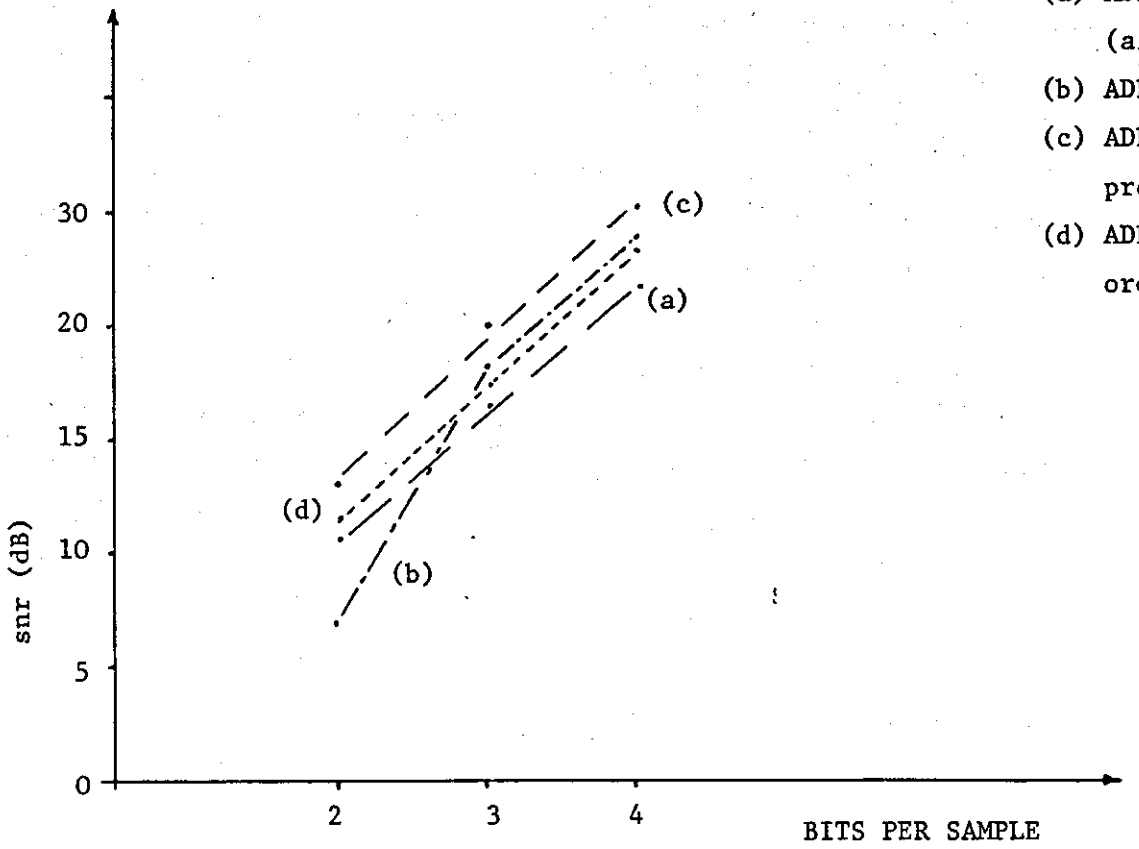
quantization accuracy. The improvement of this system (see curve c) over the conventional ADPCM encoder employing a fixed one-coefficient predictor (see curve a) is of the order of 2 to 3 dBs. When the order of the fixed predictor increases to four, the signal-to-noise ratio of the ADPCM also increases approximately by 1.5 dBs. (curve b) in the case of 3 and 4 bits per samples quantization. When however the quantization bits are reduced to two, resulting to a considerable increase of the quantization noise, the encoder shows signs of instability and low snr values are obtained.

The stochastic approximation predictor having N = 4, when used in ADPCM increased the stability and therefore the snr of the two bits per sample encoder (see curve d). This however occurred only at the transmission bit rate of 16 Kbits/sec. and for the higher bit rates, i.e. 24 or 32 Kbits/sec. the snr performance of the system showed to be equal or even lower to that of the fixed four order predictor ADPCM encoder.

5.2.3. Discussion.

From the simulations of the ADPCM systems employing the previously described prediction techniques, the following points were evident:

i) The ADPCM-FW encoder using the block adaptive linear predictor provided the higher snr values at low and high transmission bit rates. The prediction algorithm having the ability to look-ahead when defining the prediction coefficients, showed good stability properties at low output bit rates. However the prediction coefficients have to be transmitted separately to the receiver, and



(a) ADPCM, fixed one coefficient predictor ($a_1 = 0.85$)

(b) ADPCM, fixed 4th order predictor

(c) ADPCM, block adaptation 8th order predictor

(d) ADPCM, stochastic approximation 4th order predictor

FIGURE 5.5 - The snr Performance of ADPCM using 2,3 and 4 bits/sample Quantization Accuracy.

this means of course that an increase of the encoder's transmission channel bandwidth is required. Consequently, for the same output transmission bit rate the snr advantage of the ADPCM-FW system over the systems employing a Stochastic approximation predictor or a fixed one, is less than that shown in Figure 5.5.

ii) The ADPCM-ST encoder employing the Stochastic approximation predictor shows a better snr performance than a fixed predictor ADPCM system, ADPCM-FX, only at the low transmission bit rate of 16 kHz per second. Operating at output bit rates of 24 and 32 Kbits/sec. the much simpler ADPCM-FX encoder produced the same or higher snr than that of a ADPCM-ST system. The simulations also showed that the Stochastic approximation adaptation algorithm required different values for the optimizing constant g_o, for different number of quantization levels employed in the encoder. It was found that for coarse quantization the value of g_o should be smaller than that used in fine quantization cases. This is because the larger the quantization noise q_i, the larger the fluctuations of the e_i prediction coefficients around their optimum values and the easier for the algorithm to diverge. Consequently the value of the gain constant g_o should be reduced.

iii) The performance of the ADPCM-FX system was found to be acceptable only at the transmission bit rates of 24 and 32 Kbits/sec. The decoded speech data distorted from the 2 bits per sample quantization, found to cause instabilities for an N > 1 fixed predictor designed to match the long-term statistics of speech.

As mentioned in section 5.2, the purpose of the above computer

simulations was to understand prediction as applied to DPCM. Thus at the end of these computer simulation experiments and bearing in mind i) the above three points and ii) recent work on DPCM prediction by others[62,63,68], it was thought that further research on the subject could be possibly directed along the following lines:

1) The examination of how the a_k coefficients of the ADPCM-FW system could be encoded with a minimum number of bits per coefficient so its snr advantage over the ADPCM-ST and ADPCM-FX system will be as close as possible to that shown in Figure 5.5.

2) The improvement of the Stochastic approximation sequentially adaptive predictor. The limitation of the algorithm to produce a constant snr over a wide range of input powers suggests that the constant bits M of Equation (5.17) should be replaced by a variable quantity so that the gain factor in updating the a_k's is independent from power variations.

3) The Modified Kalman filter prediction procedure when applied to ADPCM[63] showed a small (0.3 dBs) improvement over the ADPCM-ST system, both operating with an output bit rate of 18.4 Kbits/sec. The use of the complete Kalman filter prediction procedure could perhaps enhance this snr improvement.

4) Chen[109] employed the Lattice predictor in ADM, updating its coefficients sequentially. From his subjective tests at transmission bit rates of 8 and 10 Kbits/sec. it appeared that the algorithm was sensitive to the quantization noise produced by the adaptive two level quantizer. Perhaps the Lattice predictor could be successfully used in ADPCM encoder operating at higher bit rates.

5) Except for these four possibilities two other cases were also considered. In particular, the predictor instead of being a fixed or an adaptive one, it could be one which combines fixed and adaptive parts. If a such PR predictor has its fixed part PF equal to $PF = \sum_{i=1}^{n_1} f_i z^{-i}$ while its adaptive section PA is $PA = \sum_{j=1}^{n_2} v_j z^{-j}$, then

$$PR = PF + PA \tag{5.23}$$

The block diagram of a DPCM with a such predictor is shown in Figure 5.6.

6) Another scheme considered, was to use two separate predictors PR_1 and PR_2 in a DPCM system, as shown in Figure 5.7. The prediction characteristics of PR_1 and PR_2 should be different since different sequences of samples, i.e. $\{X_k\}$ and $\{e_{1k}\}$ are presented to their inputs. It can be considered that

i) the first predictor PR_1 attempts to remove a certain type of redundancy from the input signal while PR_2 removes the same type of redundancy from the resulting $\{e_{1k}\}$ sequence.

ii) The second predictor attempts to remove another type of redundancy present in the input signal.

The Equations which describes the encoder of Figure 5.7 at the kth sampling instant, are written as follows:

$$e_{1k} = X_k - \sum_{i=1}^{n_1} a_{1i} \hat{X}_{k-i} \tag{5.24}$$

$$e_{2k} = e_{1k} - \sum_{j=1}^{n_2} a_{2j} e'_{1(k-j)} \tag{5.25}$$

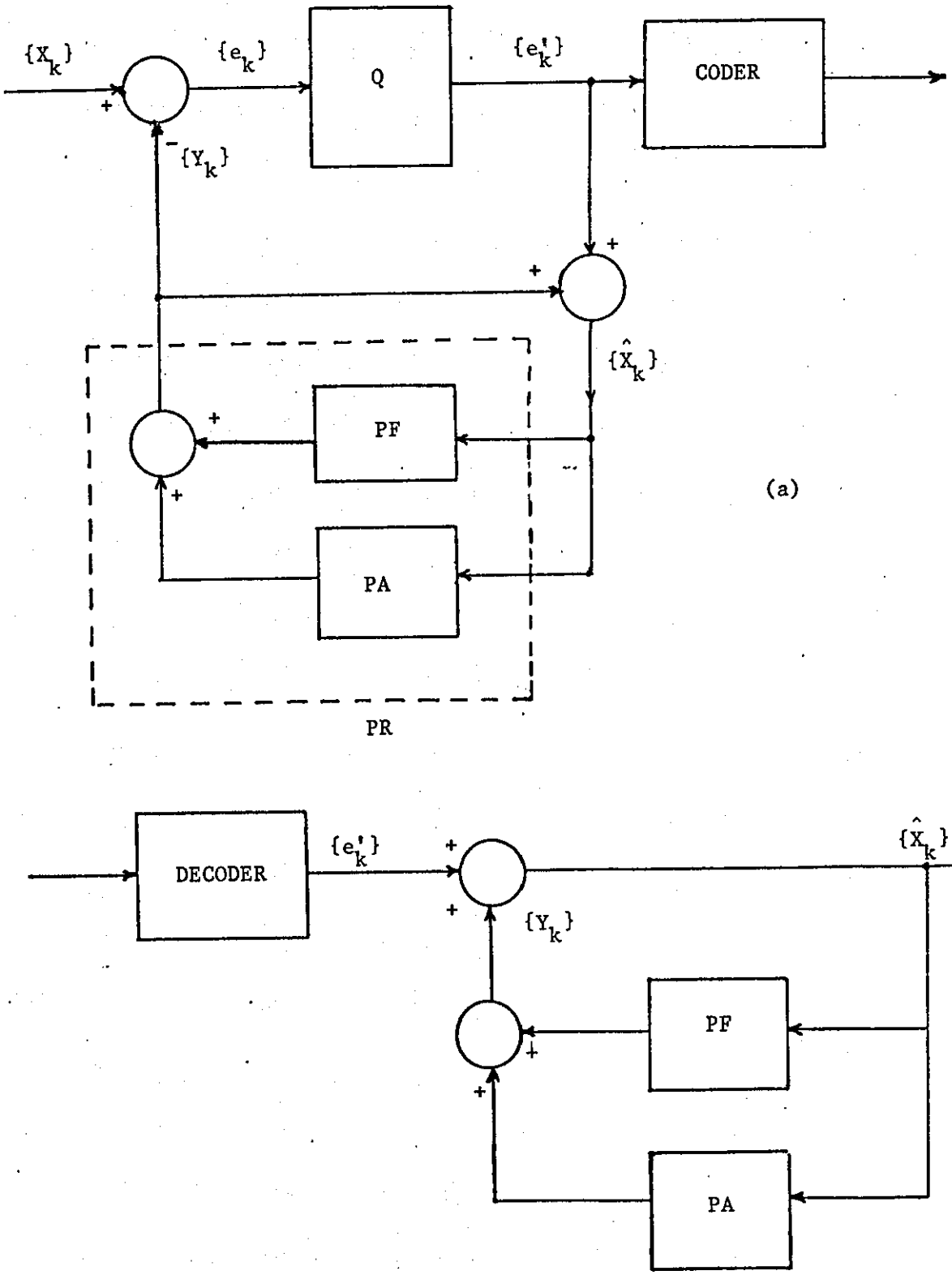


FIGURE 5.6 - A DPCM using a Fixed and Adaptive Predictor.

(a) Encoder

(b) Decoder

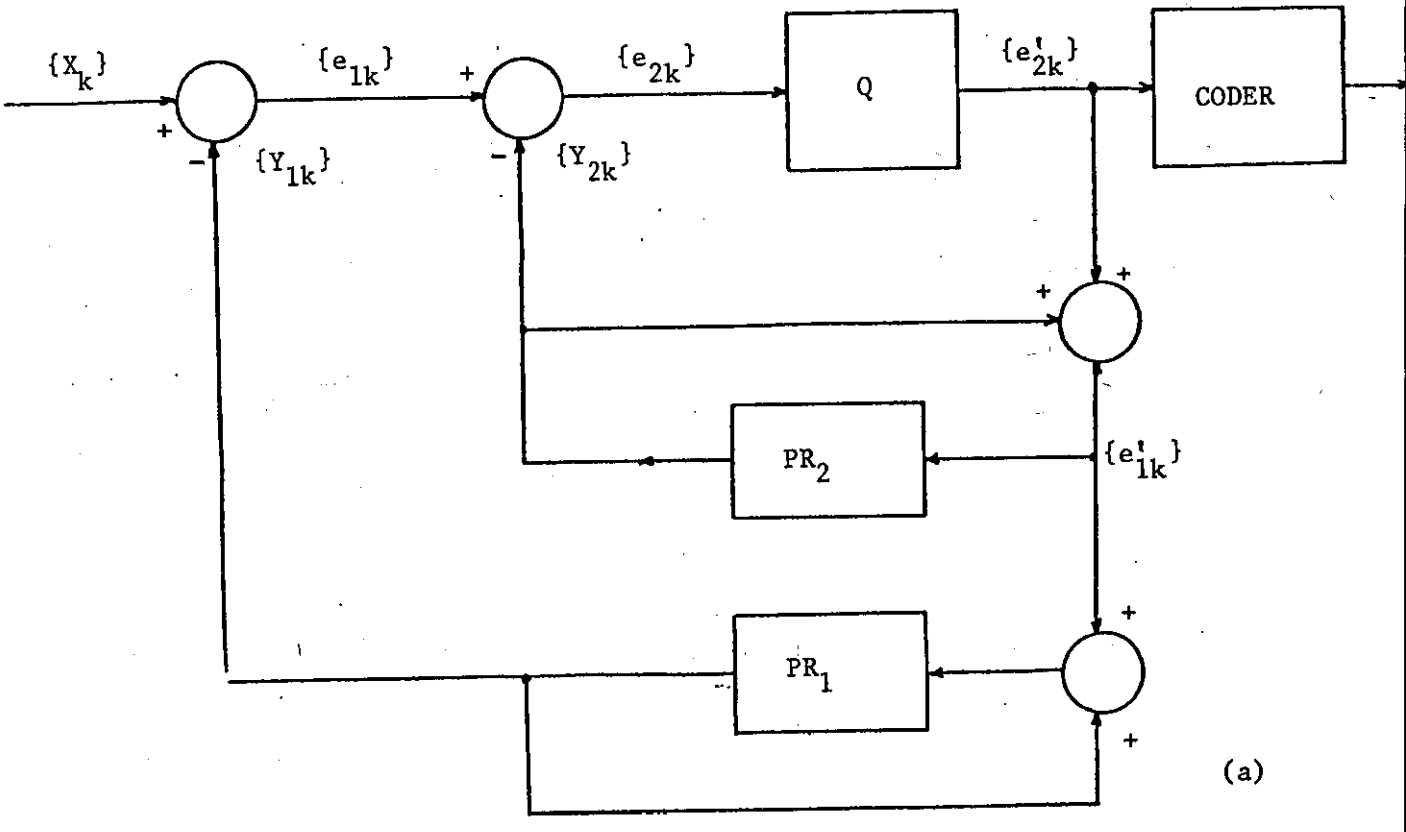


FIGURE 5.7 - A DPCM Codec using Two Separate Predictors.

(a) Encoder

(b) Decoder

$$e'_{2k} = e_{2k} + q_k \tag{5.26}$$

$$\begin{aligned} e'_{1k} &= e'_{2k} + Y_{2k} = e_{2k} + q_k + Y_{2k} = \\ &= (e_{1k} - Y_{2k}) + q_k + Y_{2k} \\ &= e_{1k} + q_k \end{aligned} \tag{5.27}$$

$$\begin{aligned} \hat{X}_k &= e'_{1k} + Y_{1k} = e_{1k} + q_k + Y_{1k} = \\ &= (X_k - Y_{1k}) + q_k + Y_{1k} = \\ &= X_k + q_k \end{aligned} \tag{5.28}$$

The various schemes for possible further DPCM-prediction investigations are illustrated in Figure 5.8. From these alternatives it was decided to examine that which combines two separate predictors in a close-loop DPCM configuration. The reasons for this choice can be explained as follows.

In all the other schemes where a single predictor is used, the predictor models the characteristics of the vocal tract. Now, it has been widely accepted that the source of excitation and the vocal tract system are independent. It is this source-vocal tract independence which allows us to consider that the speech is obtained by exciting a filter, representing the vocal tract, with the excitation signal. Consequently when a DPCM predictor models the vocal tract systems and removes from the input speech signal redundancy to form the $\{e_k\}$ error signal, it is expected that the excitation signal would be present in the $\{e_k\}$ waveform. Indeed, in the case of voiced speech the excitation information appears in

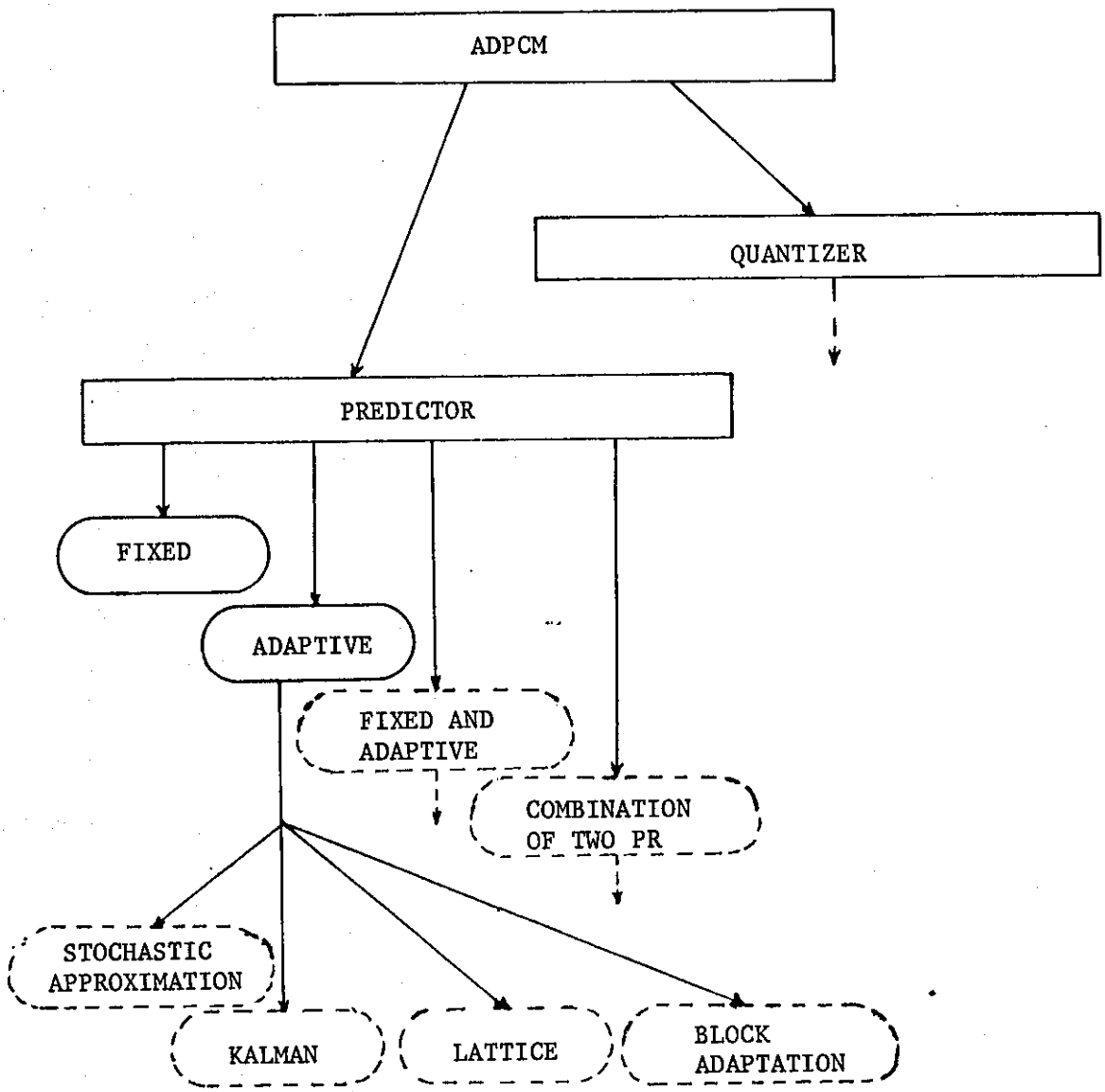


FIGURE 5.8 - The "Prediction Problem" Research Directions Considered.

the error waveform having the form of high amplitude pulses. This can be seen in the waveform of Figure 5.4.

Such an error signal is however, not the proper one to be the input signal of the encoder's quantizer. This is because if the quantizer is a fixed one, its amplitude range should cover all the high amplitude excitation pulses of the error waveform, thus producing excessive granular noise during the quantization of the rest of $\{e_k\}$. On the other hand, an adaptive quantizer should be able to increase its step size rapidly when an excitation pulse occurs while during the remaining pitch period its amplitude range should optimally cover the slowly decreasing speech waveform. This is rather difficult to achieve since the faster the quantizer responds to sudden changes in the amplitude of the input signal, the larger the amount of granular noise produced when the signal varies relatively slowly. Consequently, the excitation information must somehow be removed before the quantization of the error waveform. This is achieved when a second predictor is used in the feedback loop of the DPCM system (as shown in Figure 5.7) which removes the excitation pulses.

In the following sections two such Pitch Synchronous DPCM systems are proposed. Their performance is examined and compared to that of conventional DPCM and ADPCM systems.

5.3 PITCH SYNCHRONOUS FIRST ORDER DPCM SYSTEM.

It has been concluded in the previous section that it is advantageous if, in Differential encoding of speech signals, the error signal presented to the quantizer is free from the excitation

pulses which normally appear in DPCM systems. This is because by eliminating these relatively high amplitude pitch pulses, the error signal to be quantized has both smaller variance and dynamic range compared to the error signal $\{e_i\}$ of a DPCM encoder. Also, it was shown in the introduction of the present chapter that the main objective of Differentially encoding systems is to reduce the variance of the error signal which is subsequently quantized.

The question therefore arises of how the pitch information can be removed from the voiced speech signal. The answer to this question becomes apparent when observing the section of the voiced speech waveform shown in Figure 5.4a. It is easy to see that voiced speech is a quasi-periodic signal, i.e. there is a similarity between successive pitch periods. If we therefore form the difference between adjacent pitch periods, the resulting signal $e_1(t)$ (or $\{e_{1k}\}$, in a sampled form) will be free of excitation pulses while its amplitude range will be greatly reduced compared to that of the voiced speech. This signal can subsequently be encoded by a DPCM encoder which further exploits the correlation between the successive samples of the $\{e_{1k}\}$ sequence presented in its input. Thus a second difference sequence of samples $\{e_{2k}\}$ is produced whose variance is even smaller to that of the $\{e_{1k}\}$ sequence. When $\{e_{2k}\}$ is quantized the produced quantization noise is considerably smaller compared to the quantization noise of a DPCM system operating directly on the original signal. Consequently for the same decoded signal-to-noise ratio the number of bits per code word used for the encoding of the speech signal can be significantly reduced.

Based on the above concept the Pitch Synchronous First Order DPCM codec (PSFOD) has been developed[110]. The system is a Pitch Synchronous one since the $\{e_{1k}\}$ difference sequence is formed on a pitch period basis. $\{e_{1k}\}$ is encoded by a First Order DPCM encoder.

5.3.1. Operation of the PSFOD System.

The block diagram of the system is shown in Figure 5.9. The input speech signal X(t) is band limited and sampled to produce a sequence of samples $\{X_k\}$. Suppose that $\{X_k\}$ is a sequence of voiced samples. Let the sequence $\{S_1\}$ be the speech samples in the first pitch period of the voiced speech while sequences $\{S_2\}$, $\{S_3\}$,.... contain the samples of subsequent pitch periods. The feedback sequence $\{S'_k\}$ is initially zero and because the input speech is voiced switch SW_1 is in position 1. The first input sequence $\{S_1\} = s_1, s_2, s_3, \ldots$ is thus inverted, i.e. $\{e_1\} = -\{S_1\}$ and encoded by the First Order DPCM encoder to yield the binary sequence $\{L_1\}$, which is transmitted and also Locally decoded. In the Local decoder and also in the decoder at the receiving end (in the absence of transmission error) the decoded sequence $\{d_1\}$ is equal to

$$\{d_1\} = -\{S_1\} + \{n_1\}$$

where $\{n_1\}$ is the quantization noise generated by the DPCM encoder. Upon inverting the $\{d_1\}$ sequence, the input sequence $\{S_1\}$ is recovered as

$$\{S'_1\} = \{S_1\} - \{n_1\}$$

which is also inserted into the feedback buffer.

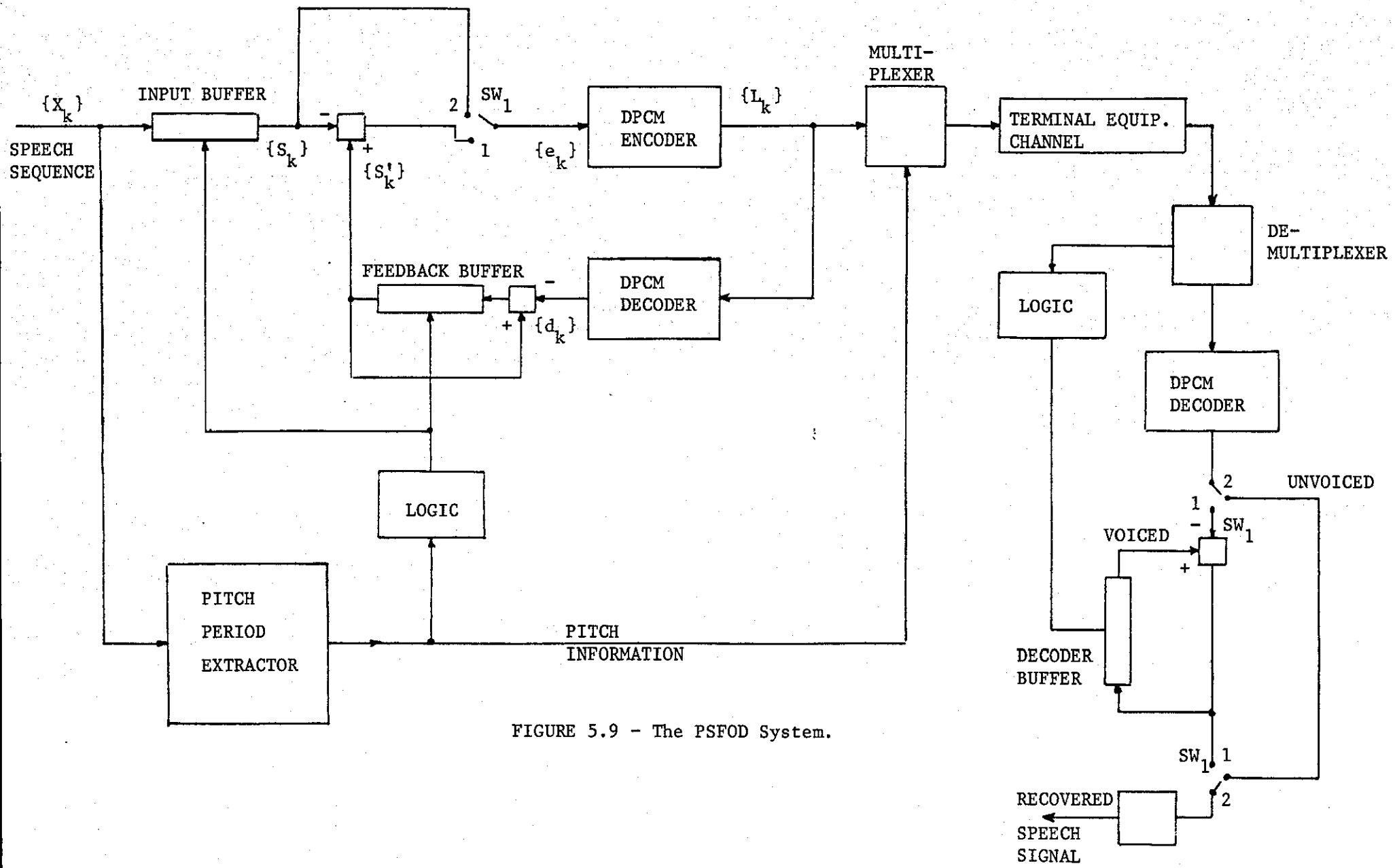


FIGURE 5.9 - The PSFOD System.

When the next input sequence $\{S_2\}$ comes, the difference sequence

$$\{e_2\} = \{S_1'\} - \{S_2\}$$

is formed and encoded by the DPCM encoder to provide $\{L_2\}$. The Local decoder and the receiver decodes $\{L_2\}$ as

$$\{d_2\} = \{e_2\} + \{n_2\}$$

and the input $\{S_2\}$ sequence is recovered by subtracting $\{d_2\}$ from the previous decoded $\{S_1'\}$ sequence stored in the feedback and decoder buffers, i.e.

$$\{S_2'\} = \{S_1'\} - \{e_2\} - \{n_2\} = \{S_1'\} - \{S_1'\} + \{S_2\} - \{n_2\} =$$

$$= \{S_2\} - \{n_2\} .$$

The new decoded sequence of input speech samples is placed in the feedback and decoder buffers of the Local decoder and receiver, in order to be used in forming the next difference sequence $\{e_3\}$. The process is repeated for the subsequent pitch segments and in general, during the encoding of the kth input sequence, the following sequences are formed:

$$\{e_k\} \text{ where } e_{ki} = S'_{(k-1)i} - S_{ki} \tag{5.29}$$

$$\{d_k\} \text{ where } d_{ki} = e_{ki} + n_{ki} \tag{5.30}$$

$$\{S_k'\} = \{S'_{k-1}\} - \{d_k\} \qquad \text{where}$$

$$S'_{ki} = S'_{(k-1)i} - d_{ki} = S_{ki} - n_{ki} \tag{5.31}$$

and i is the ith sample of the kth sequence of samples.

From Equations (5.30) and (5.31) it can be seen that the noise produced by the PSFOD system during the encoding of the kth sequence of speech samples is the noise of the First Order DPCM when encoding the difference sequence $\{e_k\}$. Because the variance of $\{e_k\}$ is considerably smaller than that of the input sequence $\{S_k\}$ the encoding performance of the system is enhanced compared to DPCM.

The operation of the system described so far, applies only for the encoding of voiced speech sounds. When unvoiced speech occurs switch SW_1 is moved to position 2, and the unvoiced speech samples are fed directly to the DPCM encoder. This is because the variance of unvoiced speech is much smaller than that of voiced (approximately 20 dB's or more) and comparable to the variance of the $\{e_k\}$ sequences formed during the voiced mode of operation. Consequently the quantization range of the DPCM encoder is suitable for encoding the $\{e_k\}$ samples, when the input signal is both voiced and unvoiced speech.

The structure of the system when forming the difference sequences $\{e_k\}$ is a closed loop one, i.e. the transmitted binary sequences $\{L_i\}$ are locally decoded. Thus the recovered sequence $\{S'_{k-1}\}$ is used to form $\{e_k\} = \{S'_{k-1}\} - \{S_k\}$, and not the actual input sequence $\{S_{k-1}\}$, i.e. $\{e_k\} = \{S_{k-1}\} - \{S_k\}$ as it happens in the case of an open loop system. The reason for using closed loop structure is to avoid the accumulation of quantization noise during encoding. Specifically if $\{e_k\}$ is formed as $\{S_{(k-1)}\} - \{S_k\}$ then it is easy to show that the recovered speech samples are equal to:

$$S'_{ki} = S_{ki} - \sum_{p=1}^{k} n_{pi} \qquad i = 1,2,3,.... \qquad (5.32)$$

and not $S'_{ki} = S_{ki} - n_{ki}$ $\qquad i = 1,2,3,....$

5.3.1.1. Formation of the difference sequences.

It has been seen that voiced speech sounds, which occur substantially more often than unvoiced sounds, are processed by PSFOD system in a pitch period basis to form the low variance difference sequence $\{e_k\}$. Adjacent pitch periods however are generally of slightly different duration and consequently the number of samples in adjacent pitch sequences $\{S_k\}$ differ. We will now take into consideration this fact and present the rules of forming the difference sequences with a minimal variance.

Suppose that the sequence $\{S_a\}$ has already been encoded as

$$\{S'_a\} = \{S_a\} + \{n_a\} .$$

The stylized Figure 5.10 shows $\{S'_a\}$ and the next sequence $\{S_b\}$ which is about to be encoded and transmitted. Let the number of samples M in $\{S'_a\}$ be greater than the number of samples N in $\{S_b\}$ i.e. $T_1 > T_2$ where T_1 and T_2 is their duration respectively. If we simply form the difference sequence

$$(a'_1 - b_1),(a'_2 - b_2),... (a'_N - b_N)$$

we find that the initial $(a'_1 - b_1),(a'_2 - b_2) \ldots (a'_{N-\lambda} - b_{N-\lambda})$ samples of this sequence are smaller than the final ones, i.e. $(a'_{N-\lambda+1} - b_{N-\lambda+1}) \ldots (a'_N - b_N)$ which tend to have large amplitude values. This is because the amplitude of the speech samples is usually decreasing after the occurrance of a pitch pulse. Thus

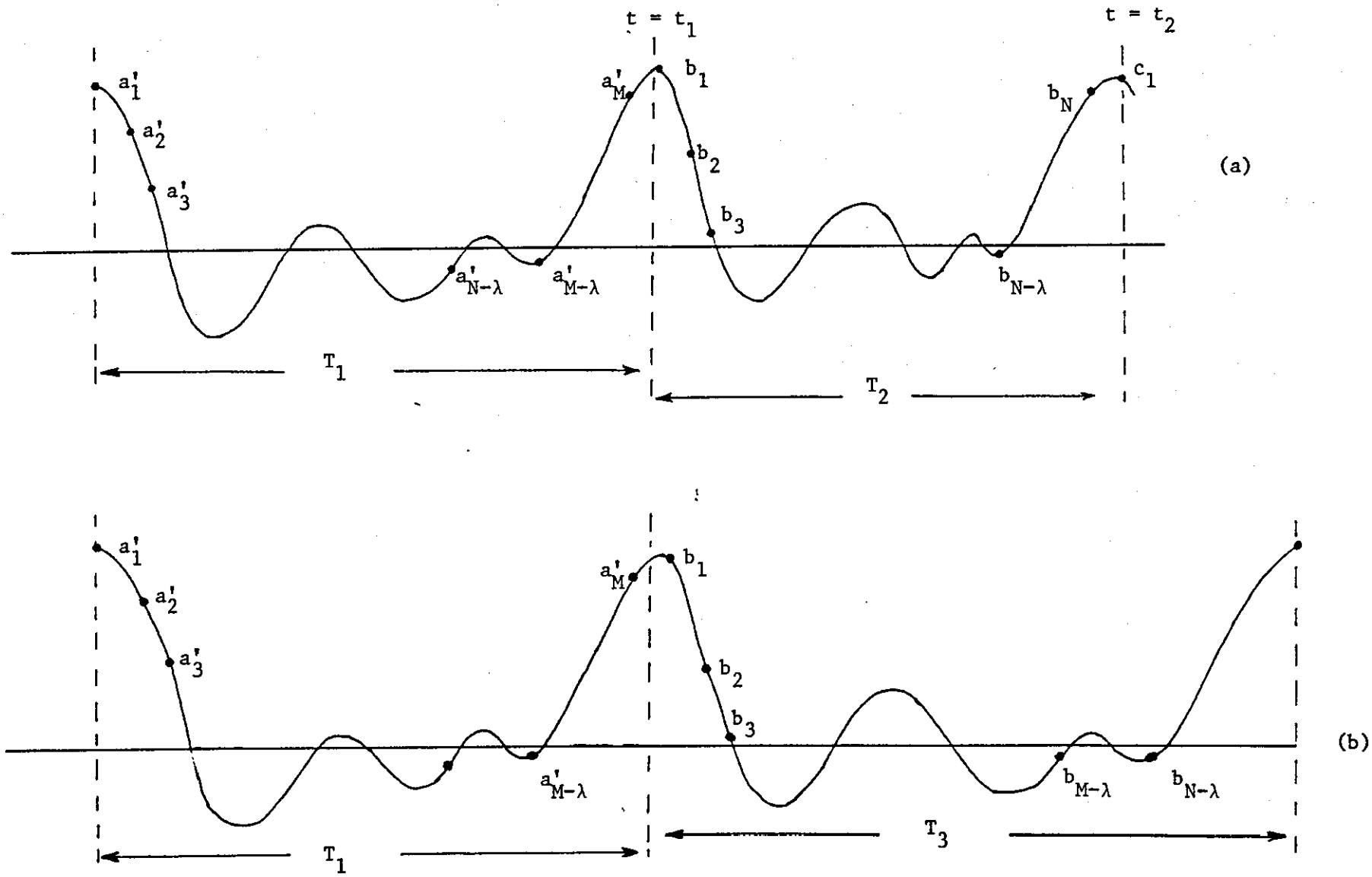


FIGURE 5.10 - Adjacent Pitch Sequences of Different Duration.

while the magnitude of the $a'_{N-\lambda+1},\ldots, a'_N$ samples is still quite small, the magnitude of the $b_{N-\lambda+1},\ldots b_N$ samples is large (due to the pitch pulse occurring at time $t = t_2$) and so is the amplitude of the formed difference samples. Because it is required all the difference samples to have a small amplitude range, the following algorithm is used.

i) The difference $(a'_1 - b_1),\ldots,(a'_{N-\lambda} - b_{N-\lambda})$ is formed.

ii) The values close to a'_M and b_N have similar magnitude dictating that $(a'_{M-\lambda} - b_{N-\lambda}),\ldots(a'_M - b_N)$ should be formed.

From the N difference samples obtained using the above rules, the receiver can recover the N samples of the $\{S_b\}$ sequence. Consequently the samples $a'_{N-\lambda+1}$ to $a'_{M-\lambda-1}$ are rejected and when $M > N$

$$\{e_b\} = (a'_1 - b_1),\ldots,(a'_{N-\lambda} - b_{N-\lambda})(a'_{M-\lambda+1} - b_{N-\lambda+1}),\ldots,(a'_M - b_N) \tag{5.33}$$

Part (b) of Figure 5.10 shows the case where the pitch sequence $\{S_b\}$ to be processed by the system has a duration T_3 and $T_3 > T_1$. The difference sequence $\{e_b\}$ is formed as follows:

i) $(a'_1 - b_1),(a'_2 - b_2),\ldots,(a'_{M-\lambda} - b_{M-\lambda})$,

ii) retain the relatively small amplitude samples

$b_{M-\lambda+1}$ to $b_{N-\lambda}$,

iii) $(a'_{M-\lambda+1} - b_{N-\lambda+1}),\ldots,(a'_M - b_N)$.

Thus when $M < N$

$$\{e_b\} = (a'_1 - b_1),\ldots(a'_{M-\lambda} - b_{M-\lambda}),\ b_{M-\lambda+1},\ldots,b_{N-\lambda},(a'_{M-\lambda+1} - b_{N-\lambda+1}),\ldots$$

$$\ldots(a'_M - b_N) \qquad (5.34)$$

The "logic" block in Figure 5.9 controls the formation of the correct difference sequences according to the above rules.

5.3.1.2. Synchronizing procedure.

When describing the operation of the PSFOD system, we assumed that the encoder and the receiver knows if the speech to be processed is voiced or unvoiced, and if voiced, knows the duration T_k of the successive $\{S_k\}$ sequences. This information is used by the "logic" block in Figure 5.9 which controls i) the position of the SW_1 switch and ii) the feedback and decoder buffers so that difference sequences $\{e_k\}$ having small amplitude range are formed.

The detection of voiced or unvoiced sounds and the measurement of the duration of the $\{S_k\}$ sequences is performed by the "Pitch Extractor" block in Figure 5.9. Because the PSFOD encoder and specifically its "logic" has to know prior to the encoding of a certain speech segment the voiced/unvoiced and T_k information related to this segment, the input speech is delayed AD seconds by the "input buffer". In this way the Pitch Period Extractor works in time ahead of the encoder following the input buffer, and the correct pitch information is provided to the "logic" of the system. The amount of the delay AD introduced by the input buffer is discussed at the end of this section.

At the transmitter the Pitch Period Extractor after examining the input speech signal, provides the necessary information to be

used by the encoding procedure. The question arises of how this data can be conveyed to the decoder at the receiving end, so that it knows i) when a pitch sequence $\{S_k\}$ commences and ii) when a transition from a voiced sound to an unvoiced one and vice-versa, occurs.

The following method can be used to achieve this:
A synchronizing word B composed of b bits is multiplexed with the data stream $\{L_k\}$ which emerges at the output of the DPCM encoder. The code-word B is transmitted every ρT seconds where T is the sampling period and ρT is less than the minimum expected pitch period. At the receiver the B code-words are demultiplexed from the received data stream and they inform the "logic" of the receiver when a sequence $\{S_k\}$ starts or that it has lasted for more than ρT seconds. To clarify this we refer to Figure 5.11 where part (b) shows the B code-words formed every ρT seconds and also the information which corresponds to the b bits of each code-word. For example the B_2 code-word contains the information that $\mu_1 T$ seconds back in time the pitch sequence $\{S_a\}$ starts. As the pitch sequence has not ended when B_3 occur, B_3 contains all zeros indicating that the start of the next pitch sequence is to be defined in a subsequent code-word. B_4 contains this information, i.e. μ_2 and indicates that $\mu_2 T$ seconds back in time from the instant B_4 occurs, the $\{S_b\}$ pitch sequence starts.

Now we have to take into consideration the delay AD introduced by the "input buffer". Let us assume that the B_i code-word is obtained at the $t = n_i T$ instant, for example B_2 corresponds to $t = n_2 T$ seconds. B_2 is multiplexed with the $\{L_k\}$ $k = 1,2,\ldots.$ binary data obtained at the output of the DPCM encoder, but the

section of $\{L_k\}$ next to the B_2 code-word is not the encoded version of the speech waveform starting at $t = n_2T$ and onwards. In fact, it represents another section of the speech waveform back in time. This is because the input speech signal has been delayed by the "input buffer" before being encoded. If the delay introduced by the input buffer is equal to $AD = 5\rho T$ then the encoded version of the speech waveform starting at time $t = nT$ is the one placed next to the B_2 binary code-word as shown in Figure 5.11. Therefore when the "logic" receives at the time instant of $t = nT$ the B_2 synchronizing code-word, the time the $\{S_a\}$ pitch sequence commences is defined as $t = nT + (5\rho - \mu_1)T$ seconds.

The AD seconds delay of the encoded speech at the transmitting end, is particularly useful at the receiver end because it allows the "logic" in the decoder to examine B_i code-words related to speech waveform not yet received, and to decide when a transition occurs from a voiced sound to an unvoiced sound and vice-versa. As mentioned, the exact location of this transition is important for the positioning of the SW_1 switch which controls the voiced/unvoiced mode of the decoder's operation. The "logic", at the receiving end, detects these voiced/unvoiced transitions as follows.

Let us suppose that an unvoiced sound is followed by a voiced one as shown in Figure 5.11. The received code-words at $t = n_oT$ and before, i.e. $B_1, B_o, \ldots$ contain b zero bits due to unvoiced speech. At time $t = nT$ the received code-word B_2 contain the binary equivalent of μ_1 and thus informs the "logic" that a unvoiced to voice transition is to occur at time $t = nT + (5\rho - \mu_1)T$ seconds.

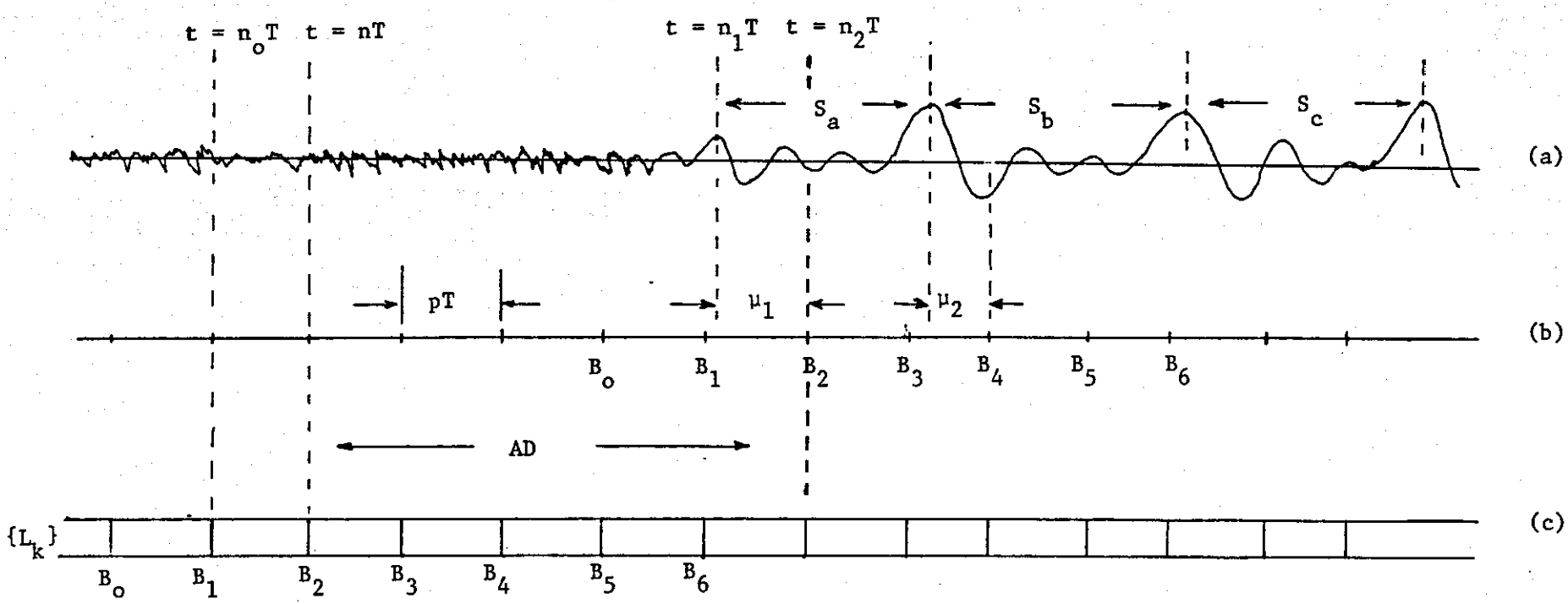


FIGURE 5.11 - Unvoiced to Voiced Transition.

In the case where a voiced sound is followed by an unvoiced one, as illustrated in Figure 5.12, the decoder's "logic" locates the transition point when more than N (for example N = 4) zero code-words B_i have been received. In particular, B_2 received at $t = nT$, informs the "logic" that the $\{S_c\}$ pitch sequence ends and another starts at time $t = nT + (5\rho - \mu_1)T$. Then the zero B_3, B_4, B_5 and B_6 follow which suggests to the "logic" that at the time $t = n_1T$ a voiced to unvoiced change in the speech waveform occurs instead of the start of a new pitch sequence, as it was assumed at the time instant of $t = nT$. Consequently the logic upon receiving B_6 arranges so that at time $t = n_1T$ the SW_1 switch is moved to position number 2.

It is now obvious that the amount of the delay AD introduced by the input buffer, depends upon N, i.e. the number zero B_i code-words required by the logic to detect an voiced to unvoiced transition. If we assume that the minimum expected pitch period is greater than 3 msec. then $\rho T = 3$msec., and if the maximum expected pitch period is 12 msec., $N = 4$ and $AD = (N+1)\rho T = 15$ msec.

When multiplexing the B code-words with the $\{L_k\}$ binary data stream at the output of the quantizer, the overall transmission bit rate of the system is not considerably increased. Suppose that $\rho T = 3$ msec. and the rate the speech is sampled is 8 kHz, i.e. a sampling period of 125 μsec, then 24 input samples are contained within the ρT time interval. Assuming that the quantizer of the PSFOD uses 8 quantization levels, that is, each of 24 quantized samples is represented by 3 bits, a total of 72 bits is obtained inside ρT. Now, the number of bits in the B code-words

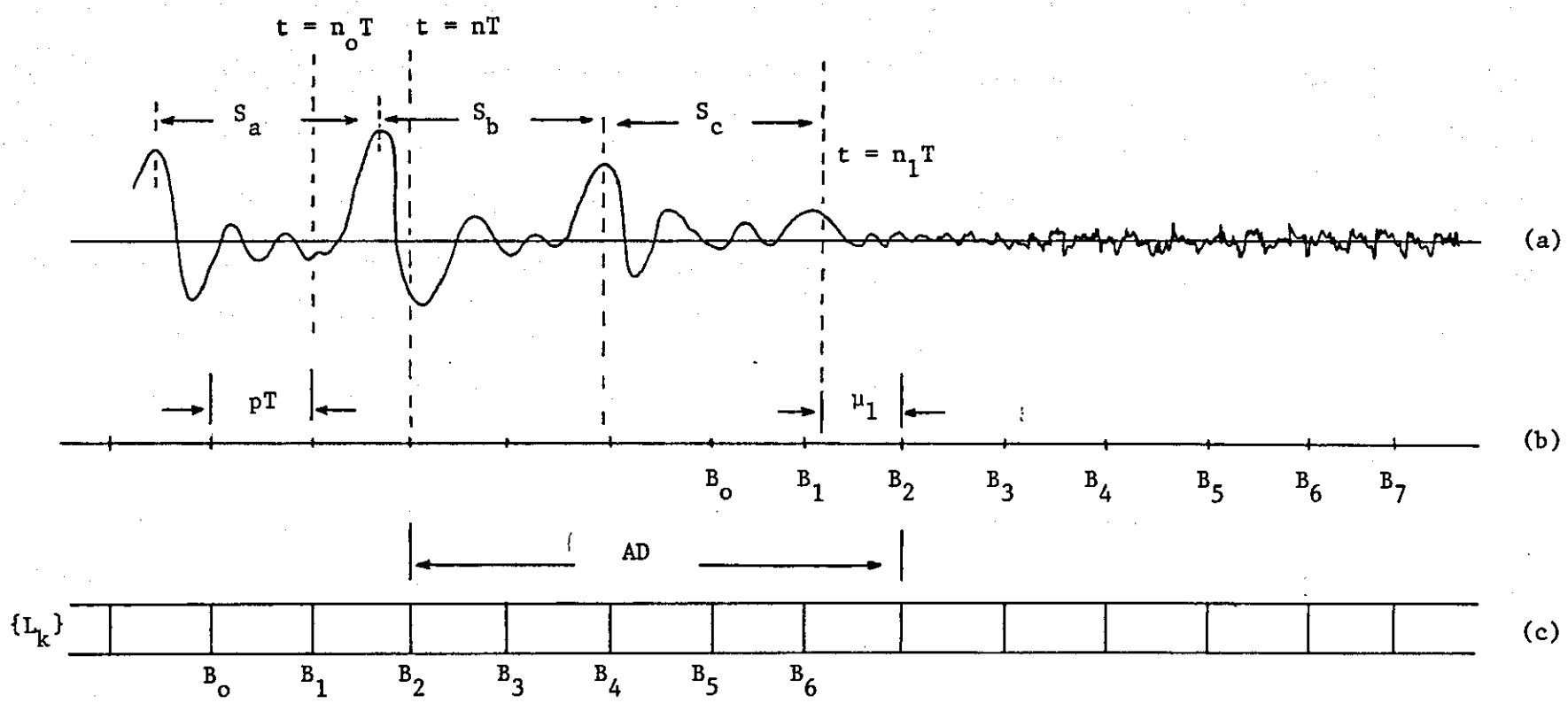


FIGURE 5.12 - Voiced to Unvoiced Transition.

depends upon the number of samples present within ρT and 5 bits are adequate for the 24 samples. Thus every 72 bits, at the output of the PSFOD quantizer, 5 more bits are added. This leads to an increase of the transmission bit rate by approximately 1.6 Kbits/sec., i.e. while the bit rate of a conventional 3 bits DPCM is 24 Kbits/sec. the bit rate of a 3 bits PSFOD system is approximately 25.6 Kbits/sec.

However, i) this extra number of bits per second required to be transmitted to the receiver side can be reduced. This is achieved when a differential version of the previously discussed synchronizing process is used. Specifically the code-word B_2, in Figure 5.12 contains the binary code of $\mu_1 - \mu_o$ instead of only μ_1, and as μ_o is known, μ_1 can be found. As the variation between adjacent pitch periods is slow the difference between adjacent μ values is small and therefore the number of b bits per code-word is reduced.

ii) The superior performance of the PSFOD system over DPCM, offsets by far this small increase in transmission bit rate.

5.3.2. Outline of Computer Simulations.

The programming simulation procedure of the PSFOD codec is rather complicated, and therefore only the basic outline of the simulation procedure is presented here.

The input speech data to the PSFOD system was first analysed. The unvoiced/voiced information together with the number of the pitch sequences contained in each voiced speech section and the number of samples contained in each of these sequences, were

stored on a digital magnetic tape. Consequently all the information provided by the "Pitch Extractor" in Figure 5.9 to control the voiced/unvoiced mode of operation and form the correct difference sequences $\{e_k\}$, was available to the PSFOD encoder and decoder.

A generalized diagram of the PSFOD simulation procedure is shown in Figure 5.13. Before discussing this procedure, the meaning of a few parameters which are read from the Magnetic Tape prior to the start of the speech encoding, will be given.

a) The variable NVAUS indicates the number of voiced/unvoiced sections in the speech signal to be encoded by the system.

b) The numbers of samples contained in each of the pitch sequences detected in the whole speech data, are stored in the NPIT(J) array in a continuous manner. For example the first element in this array NPIT(1) contains the length of the first pitch sequence detected in the input speech data, NPIT(2) contains the length of the second pitch sequence, etc.

c) MV(J) is an array which in its first element MV(1) contains the number of pitch sequences detected during the first voiced section of the speech data, in its second element MV(2) the number of pitch sequences of the second voiced section, etc.

d) MU(J) is an array which contains in its elements the numbers of speech samples in the unvoiced sections of the speech data.

The procedure starts with a DO Loop statement which determines the number of times the encoding of the input signal having different power values is to be performed. The setting of initial conditions for variables like NVA = 1, the reset of filters and counters used

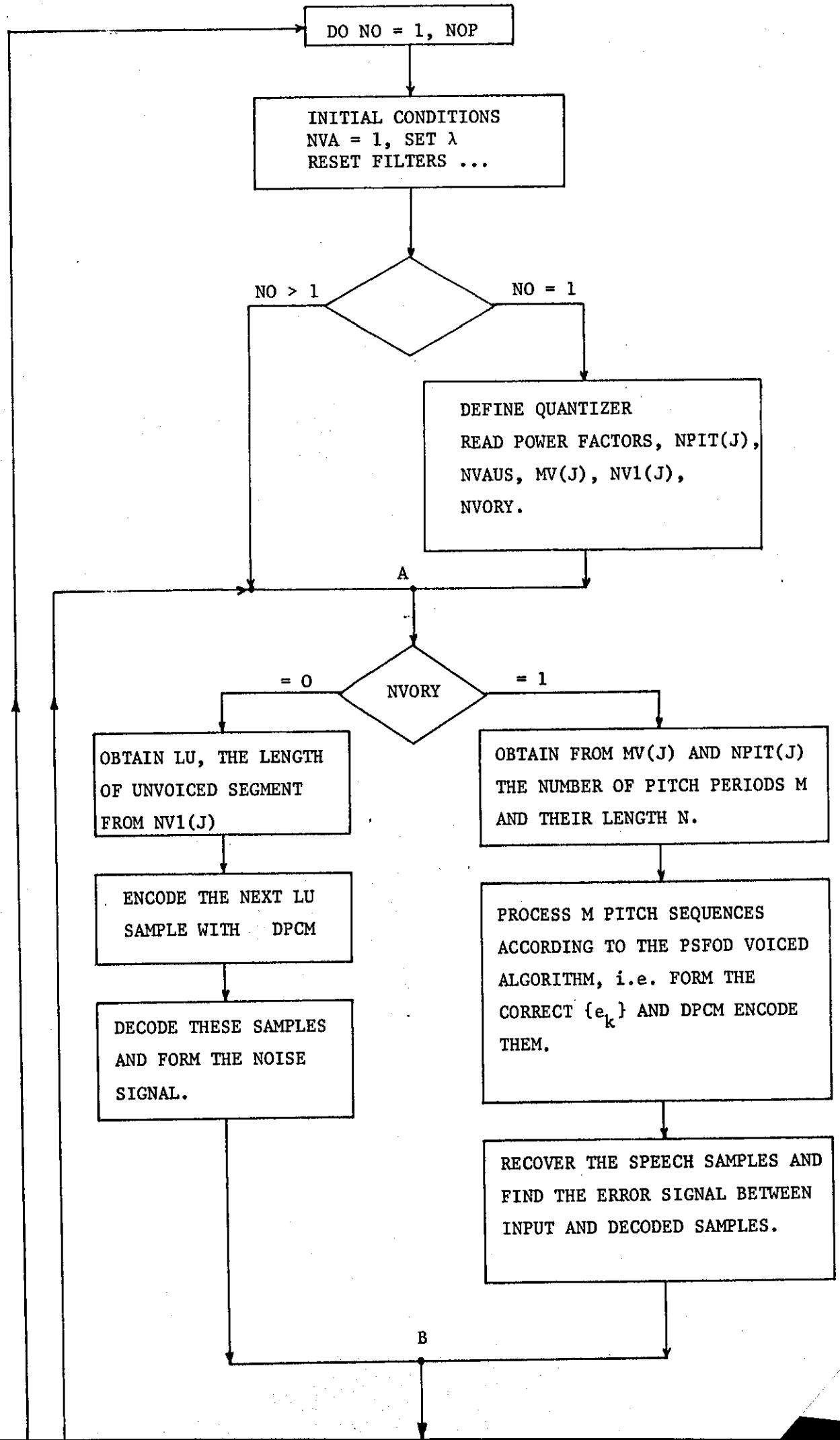
for the snr calculations etc., then follows. The program examines the NO variable of the above main DO Loop. If NO is greater than one the program goes to reference level A of Figure 5.13. If however, NO is equal to one, it means that the encoder is to process the speech data for the first time and some further information is required by the program. Specifically the structure of the quantizer employed in the First Order DPCM is defined by providing the step size δ, the number of quantization levels and the adaptation coefficients, if any. NOP multiplicative coefficients are also given to the program which are used to scale the input speech into various power levels. Finally the NPIT(J), MU(J), MV(J) arrays and the NVAUS, NVORY variables are read from the magnetic tape unit. NVORY indicates whether the next segment of speech samples to be encoded are voiced or unvoiced.

The program then goes to reference level A where the value of the NVORY variable is examined. If NVORY is equal to 1, the incoming speech to be encoded is voiced, while a value of 0 indicates that the speech is unvoiced. Let us assume that NVORY is zero, and the program follows the path which encodes segments of unvoiced speech samples. The length LU of the unvoiced segment is obtained from the MU(J) array and the next LU samples of the input speech data are fed into the input of a First Order DPCM encoder. The encoded speech samples are then decoded and the noise sequence between the original input samples and the decoded ones is formed. The power of this noise sequence is also measured and it is used in the snr calculations when the input speech data has been processed by the PSFOD system. The program then goes to reference level B.

If however, NVORY is equal to one, the speech samples to be encoded are voiced. The number of pitch sequencies M and the length of each sequence is then obtained from the MV(J) and NPIT(J) arrays. The program having all the necessary information related to the pitch sequences, processes the next M $\{S_k\}$ sequences of input samples according to the PSFOD voiced encoding procedure described in the previous section. That is, the correct $\{e_k\}$ sequences are formed which are then encoded by the First order DPCM encoder used previously to encode the unvoiced speech samples. The PSFOD decoding procedure then follows and the recovered $\{S_k'\}$ sequences are obtained. As in the case of unvoiced speech samples, the noise sequence between the original speech samples and the decoded ones is formed and its power measured. The program then goes to reference level B.

The value of NVAUS, equal to the number of voiced or unvoiced segments in the input speech data, is then compared with the value stored in the NVA counter which counts the number of voiced or unvoiced speech segments already processed by the system. If NVA is equal to NVAUS it means that the whole input speech data has been processed and the program proceeds to the snr calculations using the already measured values of the power of the input and the quantization noise sequences. Then after providing an snr output the program returns to the starting point of the main DO Loop (if NO < NOP) to process the input speech again, scaled however to a different power level. When NO = NOP the program stops.

When the value of NVA is smaller than that of NVAUS it means


DO NO = 1, NOP
INITIAL CONDITIONS
NVA = 1, SET λ
RESET FILTERS ...
NO > 1
NO = 1
DEFINE QUANTIZER
READ POWER FACTORS, NPIT(J),
NVAUS, MV(J), NV1(J),
NVORY.
A
NVORY
= 0
= 1
OBTAIN LU, THE LENGTH
OF UNVOICED SEGMENT
FROM NV1(J)
ENCODE THE NEXT LU
SAMPLE WITH DPCM
DECODE THESE SAMPLES
AND FORM THE NOISE
SIGNAL.
OBTAIN FROM MV(J) AND NPIT(J)
THE NUMBER OF PITCH PERIODS M
AND THEIR LENGTH N.
PROCESS M PITCH SEQUENCES
ACCORDING TO THE PSFOD VOICED
ALGORITHM, i.e. FORM THE
CORRECT {e_k} AND DPCM ENCODE
THEM.
RECOVER THE SPEECH SAMPLES AND
FIND THE ERROR SIGNAL BETWEEN
INPUT AND DECODED SAMPLES.
B

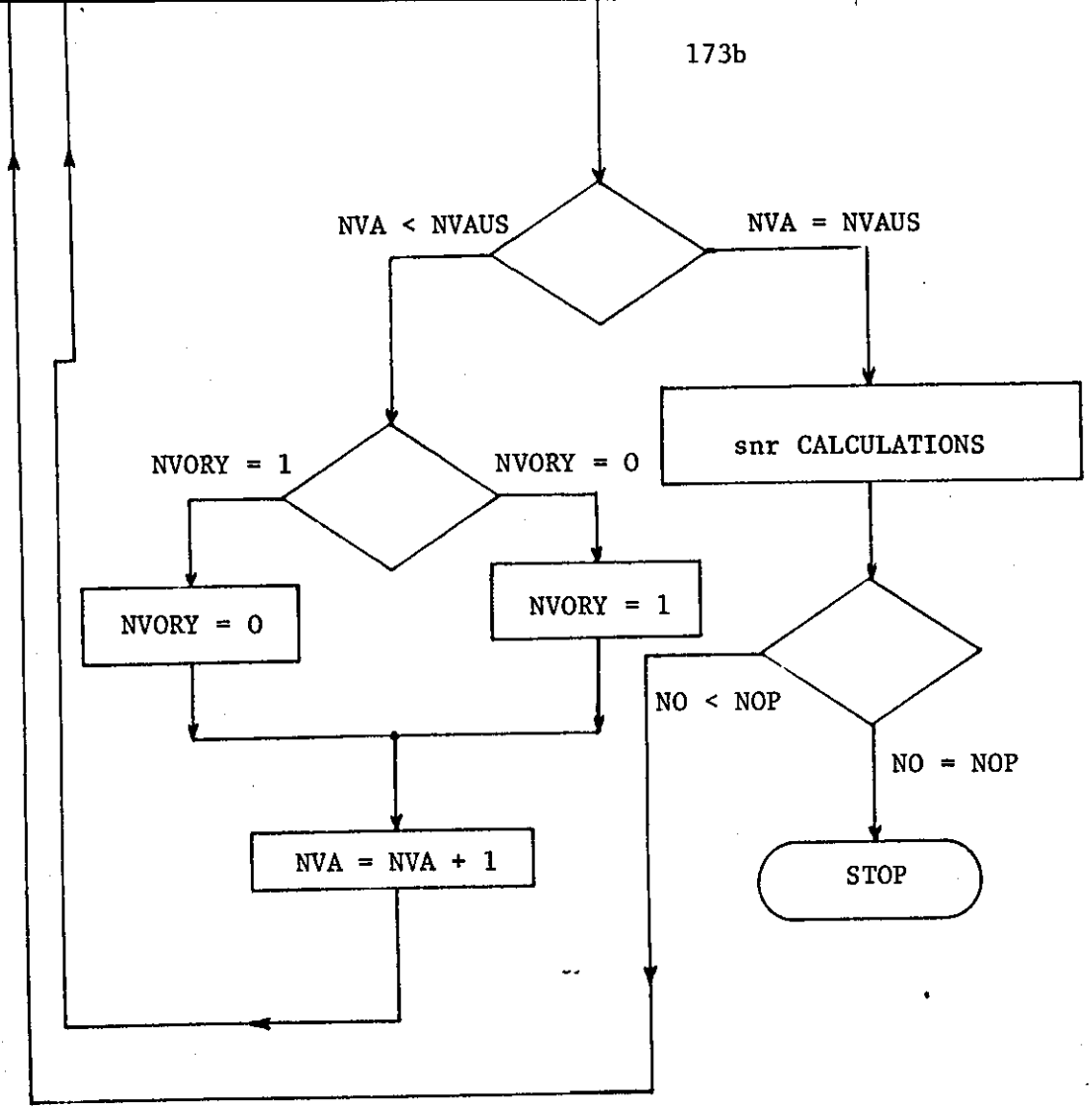
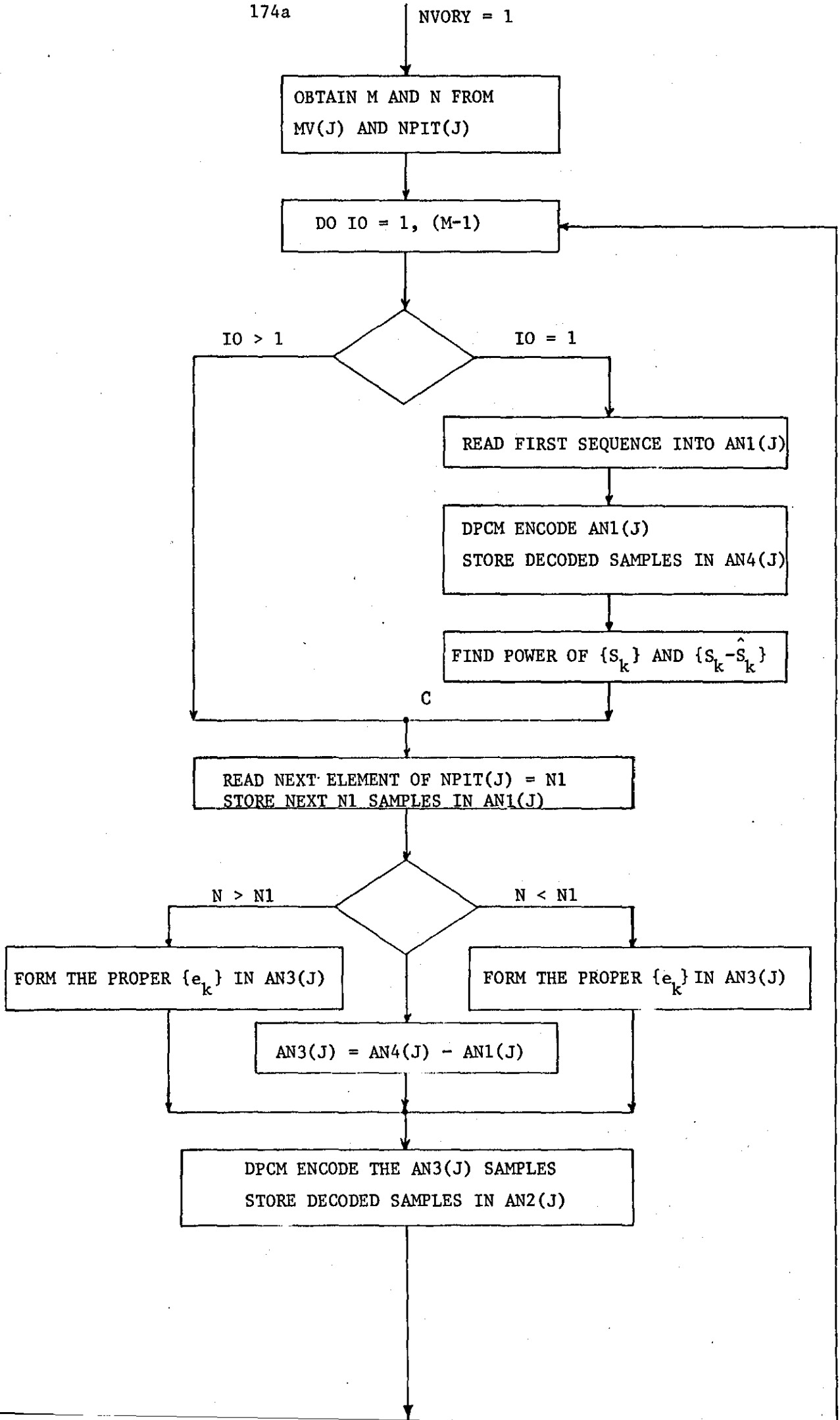


FIGURE 5.13 - A Generalized Diagram of the PSFOD Simulation Procedure.

that further speech segments have to be processed by the system. Thus the value of NVORY changes from 1 to 0 or from 0 to 1 as voiced and unvoiced sounds are considered by the program to succeed each other (silence is considered by the program as an unvoiced section). The value of NVA is increased by one and the program goes back to reference level A.

The parts of the PSFOD simulation procedure which are important, and require further explanation, are the DPCM encoder and the voiced pitch sequences processing algorithm. As the simulation procedure of the First Order DPCM System having a uniform fixed or an adaptive quantizer has already been presented in section 4.3.2. of Chapter IV, only the pitch sequences encoding algorithm need be considered here.

Figure 5.14 illustrates the block diagram of the PSFOD simulation procedure for encoding voiced speech segments. When NVORY = 1, the values for M and N are obtained from the MV(J), NPIT(J) arrays respectively. The next M pitch sequences are then processed by the part of the program which starts with the IO. Do Loop (see Figure 5.14). If IO is equal to one it means the first pitch sequences of the voiced segment is to be processed. Thus the next N speech samples are stored in an AN1(J) array and are then fed to the input of the First Order DPCM encoder. The decoded samples obtained at the output of the DPCM decoder are stored in an AN4(J) array while the power of the input samples in AN1(J) and of the noise samples (AN1(J) - AN4(J)) is measured and stored. Reference level C follows in the program which is also the point where the simulation procedure goes when IO > 1. The


NVORY = 1
OBTAIN M AND N FROM MV(J) AND NPIT(J)
DO IO = 1, (M-1)
IO > 1
IO = 1
READ FIRST SEQUENCE INTO AN1(J)
DPCM ENCODE AN1(J)
STORE DECODED SAMPLES IN AN4(J)
FIND POWER OF {Sk} AND {Sk-Ŝk}
C
READ NEXT ELEMENT OF NPIT(J) = N1
STORE NEXT N1 SAMPLES IN AN1(J)
N > N1
N < N1
FORM THE PROPER {ek} IN AN3(J)
FORM THE PROPER {ek} IN AN3(J)
AN3(J) = AN4(J) - AN1(J)
DPCM ENCODE THE AN3(J) SAMPLES
STORE DECODED SAMPLES IN AN2(J)

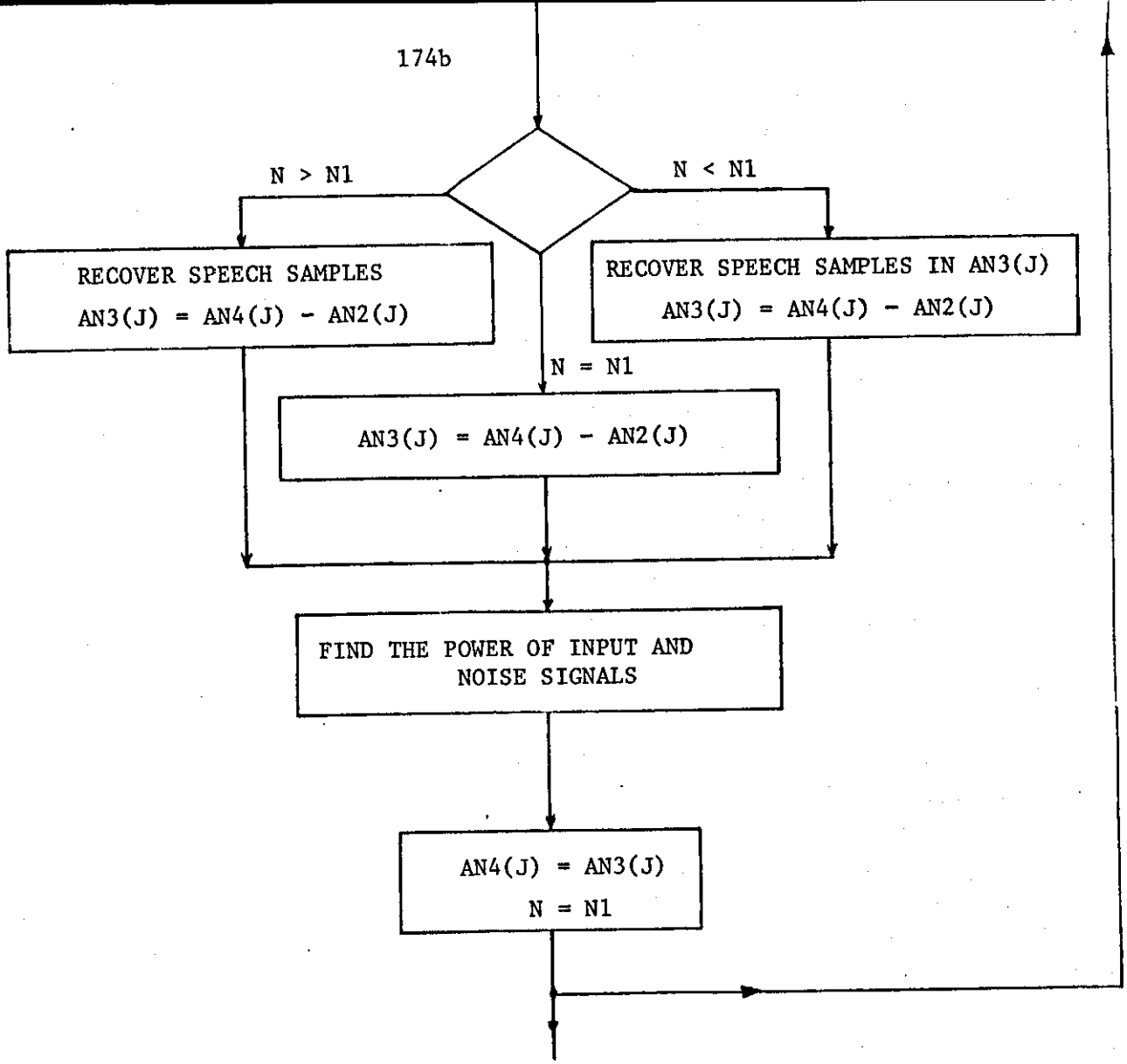


FIGURE 5.14 - "Voiced" Part of the PSFOD Simulation Procedure.

next element of the NPIT(J) array, which is the length of the next pitch sequence, is then made equal to N1. Knowing the length of the next pitch sequence, the following N1 samples are stored into AN1(J). The program compares the lengths N and N1 of the present input pitch sequence and the previous decoded one, and forms the correct $\{e_k\}$ difference sequence according to Equations (5.33), (5.34) of the PSFOD operation section. $\{e_k\}$ is stored in AN3(J) and also encoded by the DPCM system. The DPCM decoding procedure then follows and the decoded difference sequence is stored in AN2(J). The program then proceeds to form the recovered $\{S'_k\}$ sequence after comparing N and N1. $\{S'_k\}$ is obtained by taking the proper differences between samples of the AN4(J) and AN2(J) arrays, and it is stored in AN3(J). The power of the input sequence in AN1(J) and also the power of the noise sequence (AN1(J) - AN3(J)), is also measured and stored. Finally the contents of AN3(J) array, i.e., the decoded samples, are transferred to the AN4(J) array and also N is made equal to N1. If IO is less than (M-1) the program goes to the start of the IO Do Loop, otherwise it goes to the reference level B in Figure 5.13.

5.3.3. Experimental Procedure - Results.

The Pitch Synchroneous First Order DPCM system was simulated on a Hewlett Packard 2100A computer. The input data used in the simulation experiments was short sentences, spoken by a male, band-limited to 3.4 kHz and sampled at the rate of 8 kHz. The power of the speech data was set to various levels and the signal at each level was processed by the PSFOD codec. In order to

compare the PSFOD's performance with that of a DPCM, the same input speech signal was encoded by a DPCM codec. The signal-to-noise ratio was used in the experiments as a reasonable performance measure for the simulated systems. The procedure of calculating the snr has already been discussed in Chapter IV, section 4.3.2 while the actual snr formula is defined in Equation 4.13.

In the simulation experiments the DPCM codec employed in the PSFOD to encode-decode the difference sequences $\{e_k\}$ and the unvoiced speech samples, used fixed or adaptive quantizers and fixed predictors. Consequently, having in mind the various DPCM encoders, the performance of the following PSFOD systems was investigated:

i) the PSFOD-LI system, where the DPCM encoder uses a fixed uniform (Linear) quantizer, and an Ideal integrator in its feedback loop,

ii) the PSFOD-AI system, where the quantizer used in the DPCM encoder is Jayant's Adaptive quantizer and the predictor is an Ideal integrator,

iii) the PSFOD-AF system where the DPCM quantizer is Jayant's Adaptive quantizer and the predictor is a linear, Fixed coefficient predictor.

The adaptive quantizer used in (ii) and (iii) updated its step size according to Jayant's adaptation algorithm. Specifically the current quantization step size δ_r is related to the previous step size δ_{r-1} by:

$$\delta_r = \delta_{r-1} \cdot H_{(r-1)}$$

$H_{(\cdot)}$ is a function whose value depends on the modulus of the quantizations output level at the (r-1)th instant. The values of the $H_{(\cdot)}$ function are tabulated in Table 5.1, for quantizers having 8 and 16 quantization levels.

The graph of the snr against input signal power for the PSFOD-LI system is shown in Figure 5.15. The number of quantization levels used in the fixed quantizer is 8. Curve (a) is obtained from the PSFOD-LI system while curve (c) is for a First Order DPCM encoder using a fixed 8 level uniform quantizer and an ideal integrator. When comparing curves (a) and (c) a significant increase of encoding performance is noticed. The peak snr of the Pitch Synchroneous system is approximately 6 dB's higher than the peak snr of the First Order DPCM codec, while their transmission bit rates are 25.6 Kbits and 24 Kbits per second. Also for a snr of 10 dB's the dynamic range of the PSFOD-LI and the DPCM systems are 19 and 5.5 dB's respectively. When the quantization accuracy was increased in both systems to 4 bits/sample, the peak snr advantage of the PSFOD system over the DPCM remained the same, i.e. 6 dB's.

Simulation experiments were also carried out in order to answer the question of "how the accuracy of the Pitch Extractor influence the PSFOD encoding performance". The Pitch Extractor seems to be an important element of the system since it provides the input data to the "logic" which controls the formation of difference sequences having a minimum amplitude range. Furthermore, the smaller the amplitude range of these sequences is, the higher the obtained snr from the PSFOD system.

The operation of a low performance Pitch Extractor was

TABLE 5.1.

$H_{(\cdot)}$	Quantizer's o/p	8 levels	16 levels
H_1	$\frac{\delta_k}{2}$	0.875	0.9
H_2	$\frac{3\delta_k}{2}$	0.875	0.9
H_3	$\frac{5\delta_k}{2}$	1.25	0.9
H_4	$\frac{7\delta_k}{2}$	2.0	0.9
H_5	$\frac{9\delta_k}{2}$		1.20
H_6	$\frac{11\delta_k}{2}$		1.60
H_7	$\frac{13\delta_k}{2}$		2.0
H_8	$\frac{15\delta_k}{2}$		2.4

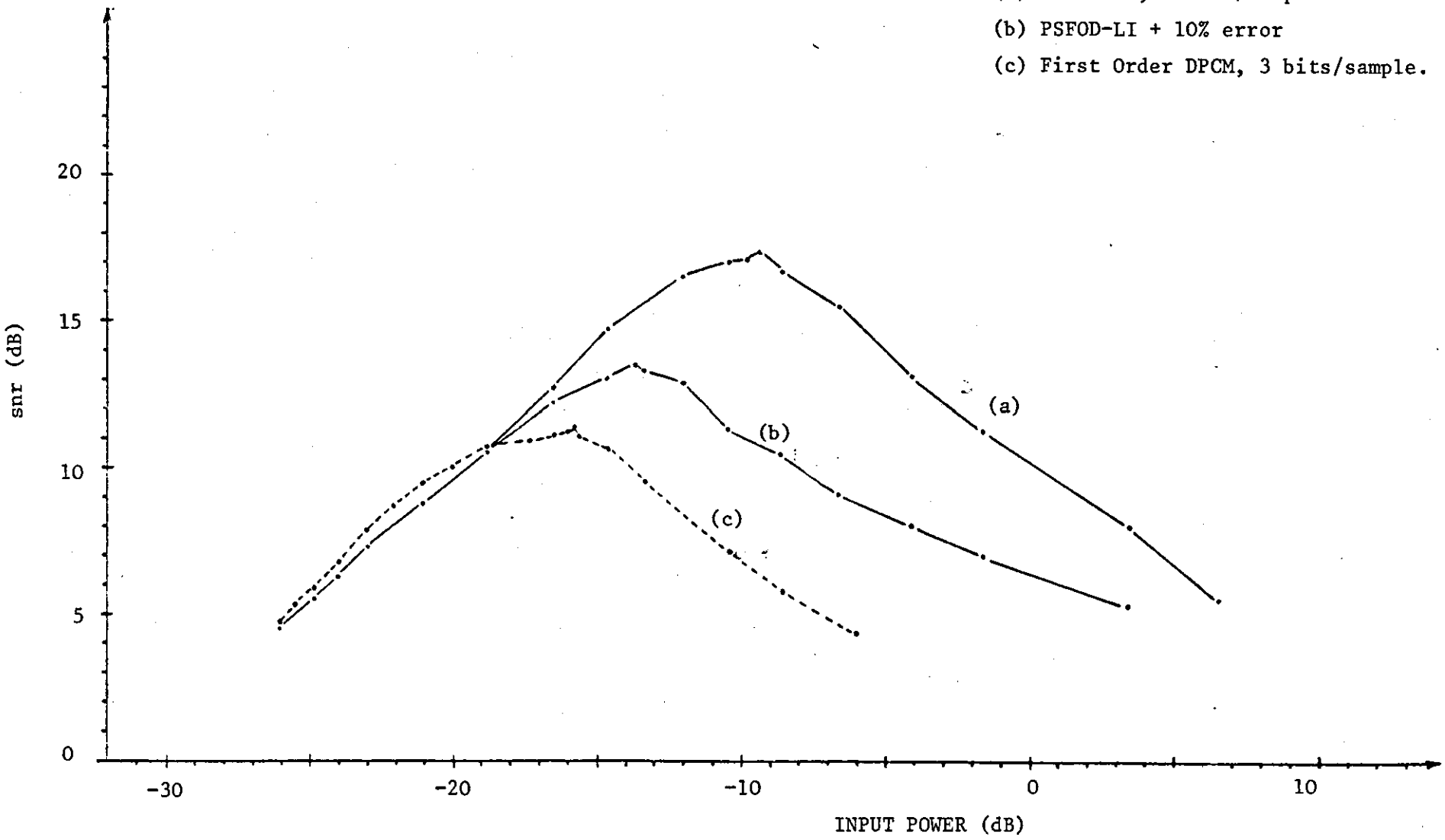


FIGURE 5.15 - snr Performance of the PSFOD-LI and DPCM Systems.

simulated by selecting 10% of the pitch periods of the input speech data in a random basis and subjecting those selected to a substantial 10% error in locating the correct pitch period. In fact, in many instants, this 10% error resulted in the definition of pitch sequences whose first sample was of opposite polarity to the first sample of pitch periods selected by a peak detection procedure. This pitch period definition provides the largest possible samples when forming the $\{e_k\}$ difference sequence. However as it is shown in curve (b) of Figure 5.15 and despite these large Pitch Extraction" errors, the peak snr of the PSFOD-LI system is still significantly above the peak snr of the DPCM codec, curve (c). Note that the snr performance of the PSFOD-LI codec which corresponds to curve (a), is obtained using an near optimum pitch extractor based on peak detection.

During this first set of PSFOD-LI experiments it was noticed that the encoding accuracy of the first pitch sequence in each voiced section significantly effected the snr. This is because the larger the encoding noise during the processing of the first pitch sequence, the larger is the amplitude range of the following difference sequences $\{e_k\}$ and consequently the lower the overall obtained snr. As a variation of the above PSFOD-LI system the programming procedure was modified so that while the input speech data was encoded with a 3 bits per sample accuracy, the encoding of every first pitch sequence was performed using 4 bits per sample. The result of this 3 bits/sample PSFOD-LI scheme, which switches into a 4 bits/sample mode when encoding $\{S_1\}$ of every voiced sound, was to obtain an additional 1.3 dB improvement in peak snr.

The second set of simulation experiments involved the PSFOD-AI system which used a ADPCM having a Jayant's adaptive quantizer and an ideal integrator. The ratio of the quantizer's maximum step size to the minimum step size was $\frac{\delta_{max}}{\delta_{min}} = 128$. When the quantizer in the system used 8 quantization levels the snr performance of the codec is shown in Figure 5.16. Curve (a) is for the PSFOD-AI system while curve (c) is for a ADPCM encoder using Jayant's adaptive quantizer and an ideal integrator in its feedback loop. Observe from curves (a) and (c) that the improvement in signal-to-noise ratio is approximately of 8 dB's over a wide dynamic range. The PSFOD-AI system presents the ADPCM encoder with a signal having a smaller and more constant dynamic range than that of the original speech signal and this results in the 8 dB's advantage shown in Figure 5.16. It is only in the region of -35 dB's in input power that the snr peaks for the conventional ADPCM encoder and the advantage of the PSFOD-AI over the ADPCM is reduced to approximately 7 dB's. Curve (b) is for the PSFOD-AI system when the pitch extractor is in error for 10% of the pitch periods selected in a random basis. The magnitude of this error is again equal to 10% of the correct pitch duration. It can be seen from curves (b) and (c) that the PSFOD-AI has still a snr gain of 5 dB's over the isolated ADPCM encoder.

Figure 5.17 illustrates the variations of snr against the input signal power, obtained from the above two systems when the quantization accuracy of 4 bits/sample. The improvement in the snr of the PSFOD-AI system when the input power causes the isolated ADPCM to have its peak snr, is again 6 dB's. Over a substantial

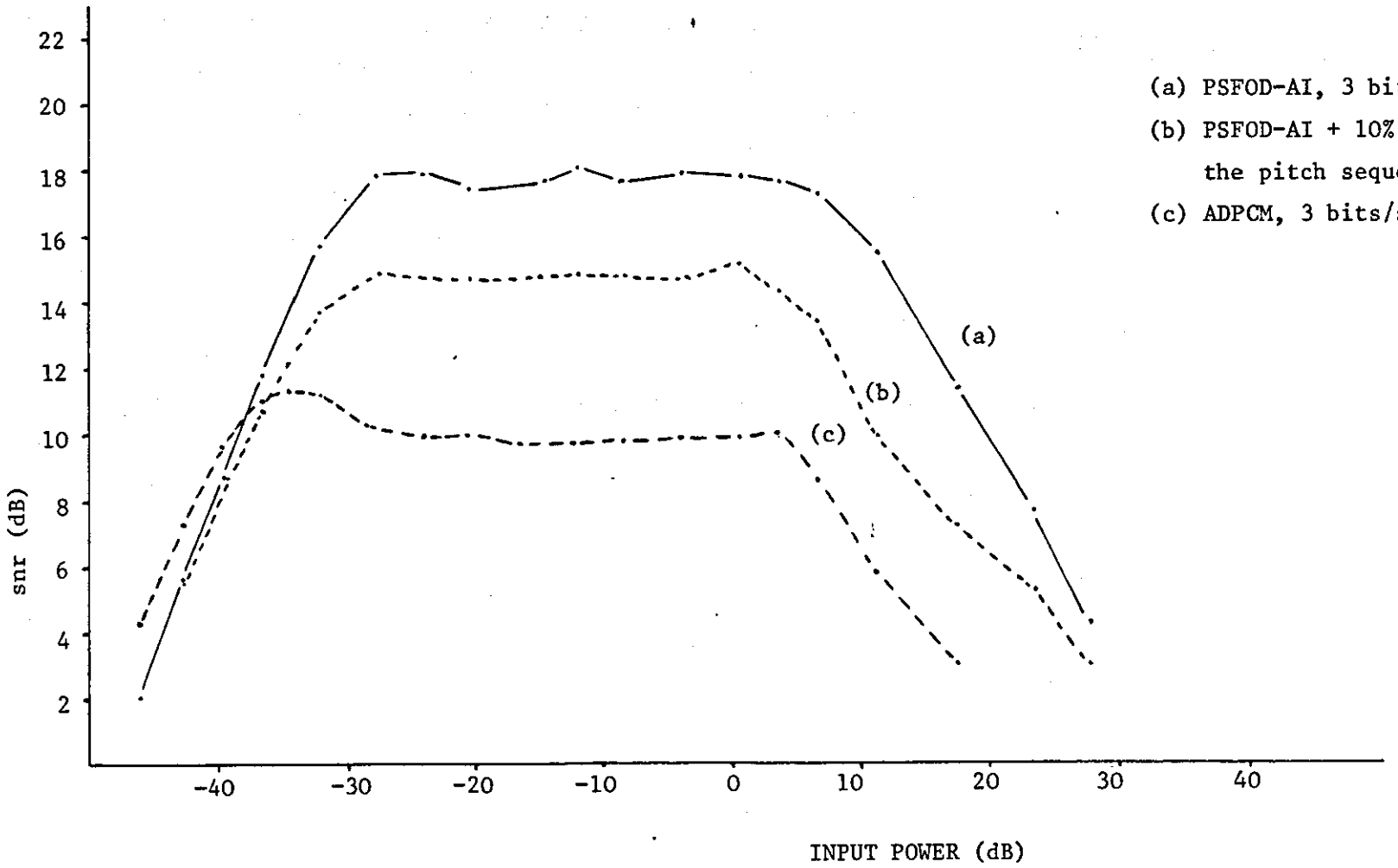


FIGURE 5.16 - snr Performance of a 3 bits/sample PSFOD-AI System.

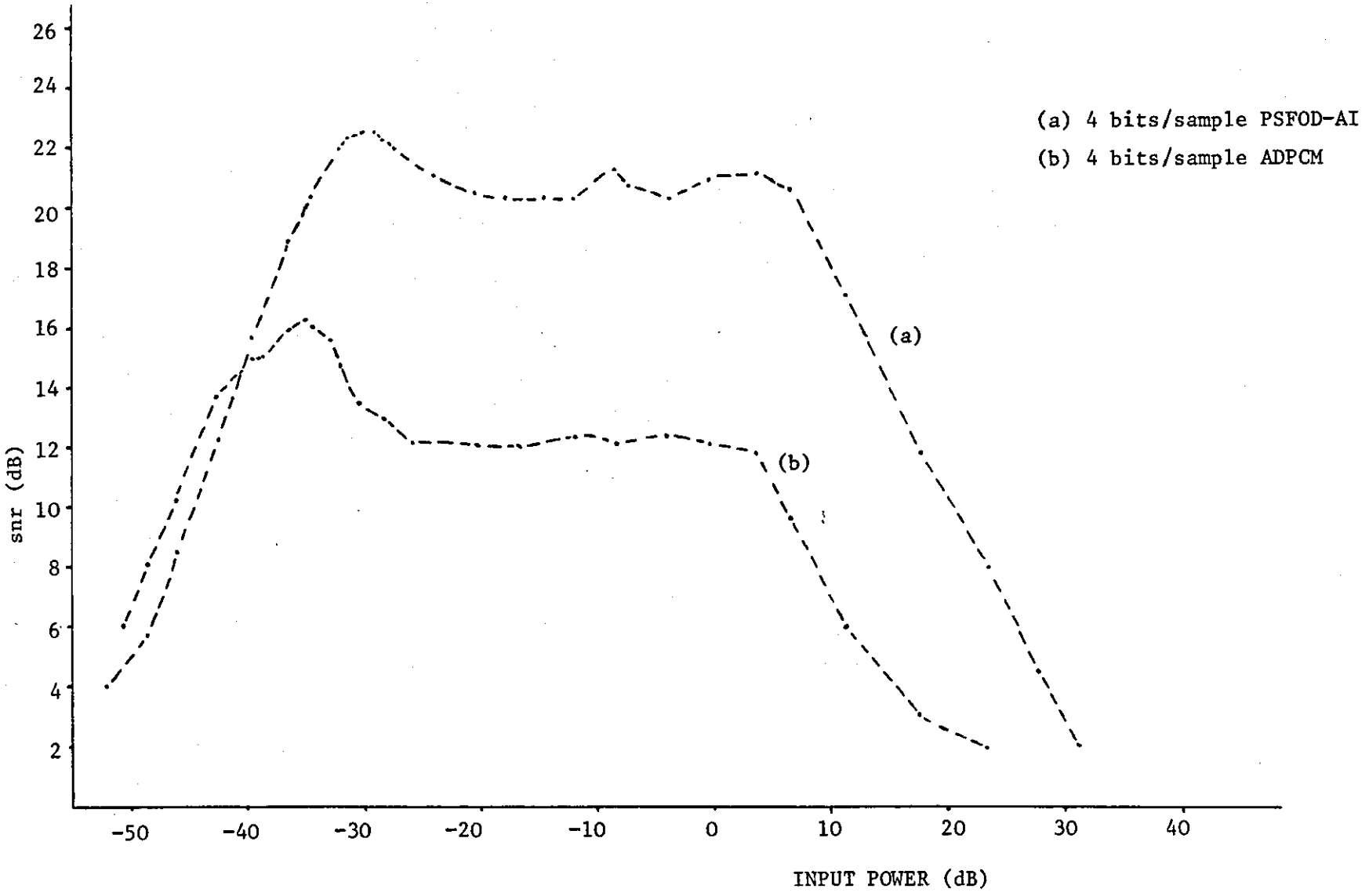


FIGURE 5.17 - snr as a Function of Input Power.

dynamic range the PSFOD-AI system maintains a 8 to 9 dB's advantage.

The other element of the ADPCM encoder which could improve the PSFOD performance is the predictor. When the ideal integrator in the feedback loop of the DPCM encoder was substituted by a fixed coefficient predictor, the resulting PSFOD-AF provided the snr against input power graph shown in Figure 5.18. Jayant's adaptive quantizer had 8 quantization levels and the fixed predictor contained only one coefficient $a_1 = 0.55$. Curve (a) corresponds to the PSFOD-AI system while curve (b) is for the PSFOD-AF codec. The two curves show that for a wide range of input power levels the snr of the PSFOD-AF having one coefficient is approximately 1 dB better than the snr of the PSFOD-AI system. The system was also tested when the ADPCM used a higher order fixed coefficient predictor. The prediction coefficients were defined by the long-term autocorrelation speech values given by McDonald[53]. It seemed however that the statistics of the input signal used in the experiments were not matched to the predictors coefficients. Thus when a 4th order fixed predictor was employed in the ADPCM the snr performance of the PSFOD-AF system was considerably reduced.

The snr of the PSFOD system has been discussed so far when the DPCM encoder, which processes the $\{e_k\}$ sequences and unvoiced speech samples, uses a fixed or adaptive quantizer and an ideal or fixed coefficient predictor. The best encoding performance observed was that of the one coefficient PSFOD-AF system. In order to increase further the snr of the codec., the possibility of applying prediction in the main pitch loop of the system was also considered.

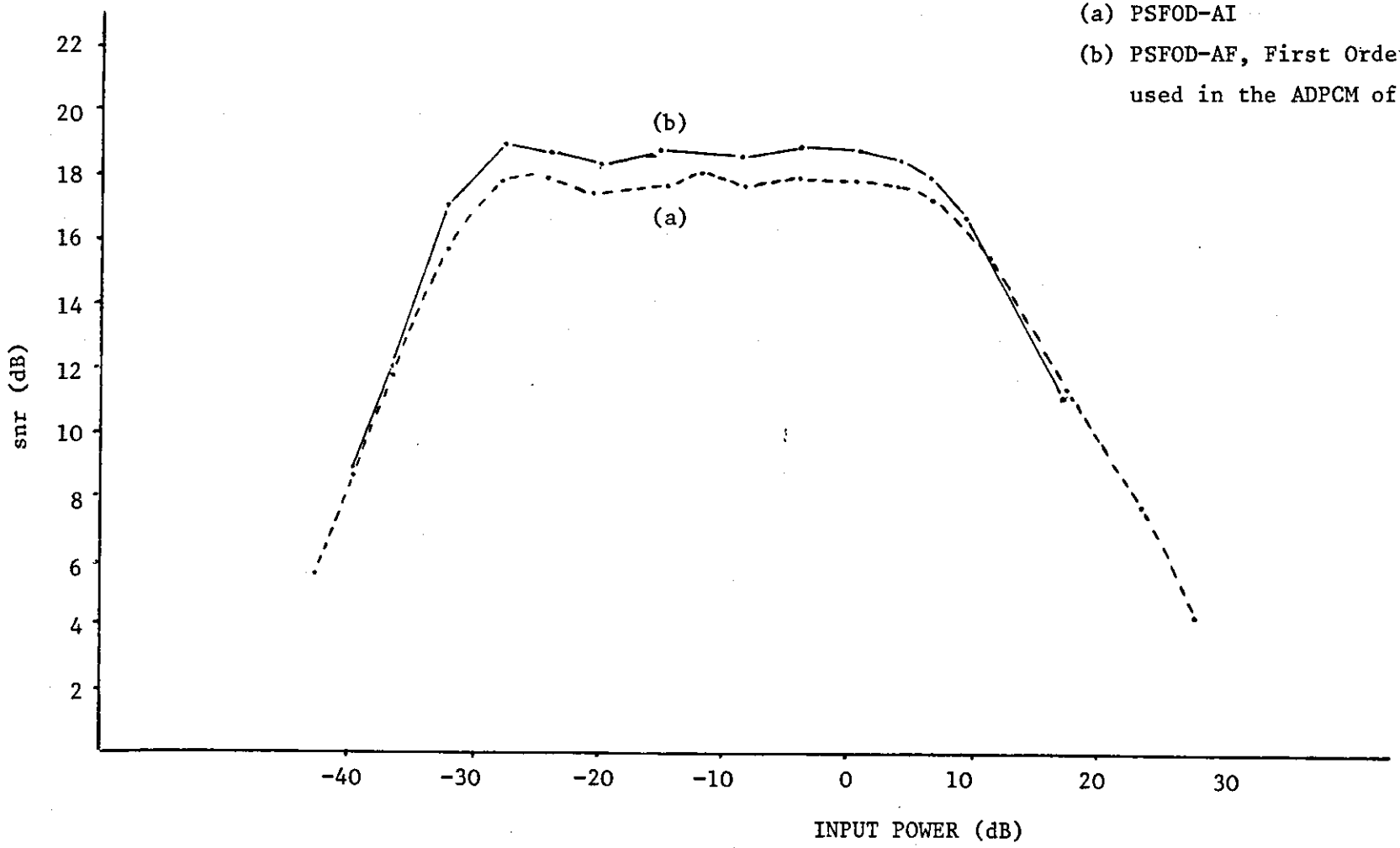


FIGURE 5.18 - snr Performance for 3 bits per sample PSFOD-AI and PSFOD-AF Systems.

It was observed that a slight difference in the amplitude range of adjacent pitch periods existed, due to slow variation in the power of the speech signal during voiced sounds. Consequently instead of forming the difference sequence $\{e_k\}$ as $\{S'_{k-1}\} - \{S_k\}$, the already decoded sequence $\{S'_{k-1}\}$ can be multiplied by a coefficient β_{1k} which scales the samples of $\{S'_{k-1}\}$ in order to reduce the amplitude of the $\{e_k\}$ sequence. The constraint imposed in defining β_{1k} is that all the information used in its calculation must be available to the receiver without transmitting any additional data.

Two such prediction schemes were considered. The first one operates as follows: During the processing of the kth input sequence $\{S_k\}$, $\{e_k\}$ was formed as

$$\{e_k\} = \{S'_{k-1}\} \cdot \beta_{1k} - \{S_k\} \tag{5.35}$$

where

$$\beta_{1k} = \frac{\frac{1}{N}\sum_{i=1}^{N} \left| S'_{(k-1)i} \right|}{\frac{1}{N_1}\sum_{i=1}^{N_1} \left| S'_{(k-2)i} \right|} = \frac{B}{A} \tag{5.36}$$

and N, N_1 are the number of samples in the $\{S'_{k-1}\}$ and $\{S'_{k-2}\}$ sequences respectively. Thus when the power of the voiced speech signal is slowly increasing, $\beta_{1k} > 1$ because $B > A$ and the amplitude range of the $\{S'_{k-1}\}$ sequence is increased after multiplied by β_{1k}.

In this way the power of $\{S'_{k-1}\}$ approaches further that of the $\{S_k\}$ sequence and the amplitude range of $\{e_k\}$ is reduced. In the case where the power of the voiced speech is slowly decreasing $B < A$, $\beta_{1k} < 1$ and the amplitude range of $\{S'_{k-1}\}$ is reduced in

order to further approximate the following $\{S_k\}$ pitch sequence.

Simulation experiments of the above pitch loop prediction technique were carried out for the PSFOD-AI and PSFOD-AF systems. The obtained snr against input power curves show no improvement when compared with the snr curves of these two systems with $\beta_{1k} = 1$. As a result the following prediction method was developed.

Suppose that $\{e_k\} = \{S'_{k-i}\}\ \beta_{1k} - \{S_k\}$ and that β_{1k} is required to minimize

$$\frac{1}{N} \sum_{i=1}^{N} \left(S'_{(k-1)i}\ \beta_{1k} - S_{ki} \right)^2 = \varepsilon \qquad (5.37)$$

where N is the number of samples in the $\{S_k\}$ pitch sequence. It is evident from Equation (5.37) that ε is a function of β_{1k} and to minimize ε we must have

$$\frac{d\varepsilon}{d\beta_{1k}} = 0$$

and since $\frac{1}{N}$ is a constant

$$\frac{d\left[\sum_{i=1}^{N} \left(S'_{(k-1)i}\ \beta_{1k} - S_{ki} \right)^2\right]}{d\beta_{1k}} = 0$$

If we expand the summation term and take its derivative we have

$$\sum_{i=1}^{N} \left(-2\ S'_{(k-1)i} \cdot S_{ki} + 2\beta_{1k}\ S^2_{(k-1)i} \right) = 0$$

or

$$\beta_{1k} \sum_{i=1}^{N} S'^2_{(k-1)i} = \sum_{i=1}^{N} S_{(k-1)i}\ S_{ki}$$

and

$$\beta_{1k} = \frac{\sum_{i=1}^{N} S'_{(k-1)i}\ S_{ki}}{\sum_{i=1}^{N} S'^2_{(k-1)i}} \qquad (5.38)$$

Now because it is required β_{1k} to be calculated from samples already known to the receiver and since $\{S_k\}$ is not known to the receiver, Equation (5.38) was modified as

$$\beta_{1k} = \frac{\sum_{i=1}^{N} S'_{(k-2)i} \, S'_{(k-1)i}}{\sum_{i=1}^{N} S'^{2}_{(k-2)}} \qquad (5.39)$$

The simulation of the PSFOD-AI system having the above prediction algorithm in its pitch loop, provided the snr curves shown in Figure 5.19. Curve (a) is for the PSFOD-AI without prediction system and curve (b) is for the system which forms the difference sequences according to Equations (5.35) and (5.39). Notice that the prediction improves the overall snr performance of the system but not substantially.

In order to find the maximum snr advantage when pitch prediction is included in the system, Equation (5.38) has been also used in the simulations, while assuming that the receiver knew the values of β_{1k}. It was observed that the pitch prediction resulted a maximum of 1 dB gain in snr throughout the dynamic range of the encoder. Similar snr observations were made when pitch prediction was applied to the PSFOD-AF system.

Finally we mention that the gains in snr of the Pitch Synchroneous systems over DPCM, were observed when processing many short segments of speech of duration of about 2 seconds. It was noticed that the actual values of the peak snr obtained for various speech segments could differ by 3 or 4 dBs, but the actual snr advantage of the PSFOD over the DPCM was always of the order shown in Figures 5.15,

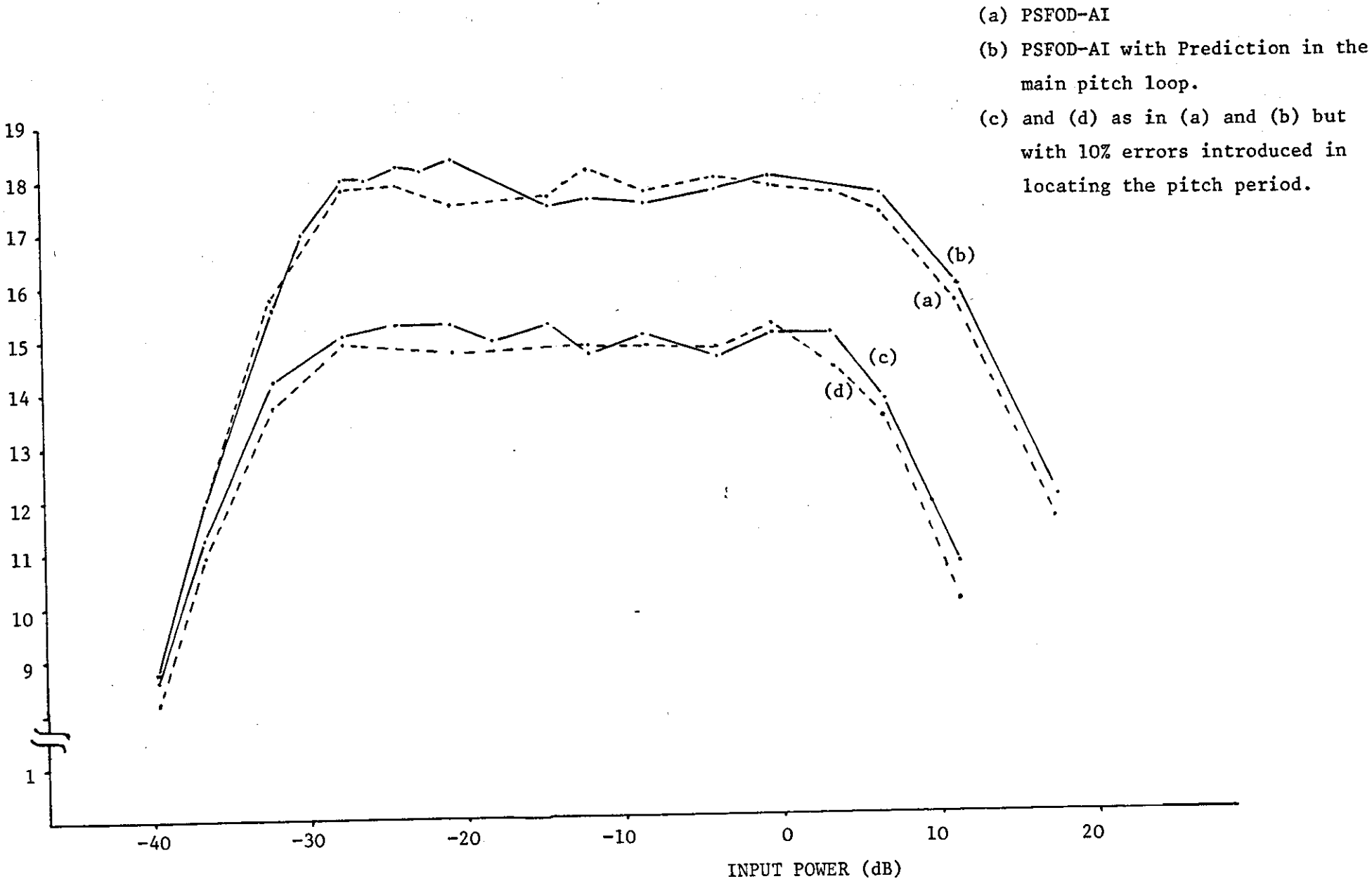


FIGURE 5.19 - snr as a Function of Inpur Power, 3 bits/sample.

5.16, 5.17 and 5.18. The input speech material for which the results in these Figures were obtained was "I decided that" recorded on a digital magnetic tape in a rather noisy laboratory environment.

5.3.4. Note on Publication[110].

A paper entitled "Pitch Synchronous First Order Linear DPCM System", in co-authorship with Dr. R. Steele (thesis supervisor), has been published in Electronic Letters of I.E.E., Vol.12, number 4, February 1976. The paper is a brief version of sections 5.3.1, 5.3.1.1, 5.3.1.2 and presents the snr against input power results obtained from the PSFOD-LI system.

5.4 PITCH SYNCHRONOUS DIFFERENTIAL PREDICTIVE ENCODING SYSTEM.

The PSFOD system, presented in the previous section, has a substantial snr gain over DPCM and ADPCM systems. Most of this gain is due to the pitch synchroneous processing of the speech signal, and only a fraction of it is contributed by the prediction. Specifically, the PSFOD-AI system has an snr advantage over isolated ADPCM of approximately 8 dBs, while the addition of a fixed coefficient predictor in the feedback loop of the PSFOD's, ADPCM encoder gives an increase of only 1 dB. Furthermore, it was shown that the introduction of prediction in the system's pitch loop gave a marginal improvement in snr. The reasons for the poor performance of the predictors are:

i) in the case of prediction in the outer pitch loop, the prediction coefficient β_{1k} is calculated using the previous decoded

pitch sequences $\{S'_{k-1}\}$ and $\{S'_{k-2}\}$, instead of $\{S_k\}$ and $\{S'_{k-1}\}$ (see section 5.3.3). This arrangement is used to avoid transmitting data corresponding to β_{1k}.

ii) The coefficients of the time-invariant ADPCM predictor used in the PSFOD-AI system, were not matched to the long term statistics of the speech signal. It was observed that the correlation of $\{e_k\}$ sequences presented to the ADPCM encoder was considerably reduced compared with that of the input speech samples. This made the task of predicting the incoming e_{ki} samples, difficult.

To overcome the prediction difficulties present in the PSFOD system, a second pitch synchroneous system called, "Pitch Synchroneous Differential Predictive Encoding System" (PSDPE), was developed.[128] The system, like PSFOD, reduces the dynamic range of voiced speech to a value similar to that of unvoiced speech. Thus the signal produced from the PSDPE differential processing algorithm is encoded with a much improved accuracy because its dynamic range is smaller than that of the input speech.

The principle of operation of the PSDPE system can be described as follows. Suppose that $S_{j(i-1)}$ and S_{ji} are the (i-1)th and ith speech samples of the jth pitch sequence, as shown in Figure 5.20. Let us also assume that S_{ji} is the speech sample to be encoded by the system. Then the predicted value of S_{ji} is obtained from the past $S_{j(i-1)}$, $S_{j(i-2)}$,... samples. The prediction error e_{ji} is formed between the actual speech sample and the predicted one. The same difference procedure is also applied to the corresponding samples of the (j-1)th pitch sequence, i.e. $S_{(j-1)i}$ is predicted from the $S_{(j-1)(i-1)}$, $S_{(j-1)(i-2)}$... samples and the $e_{(j-1)i}$

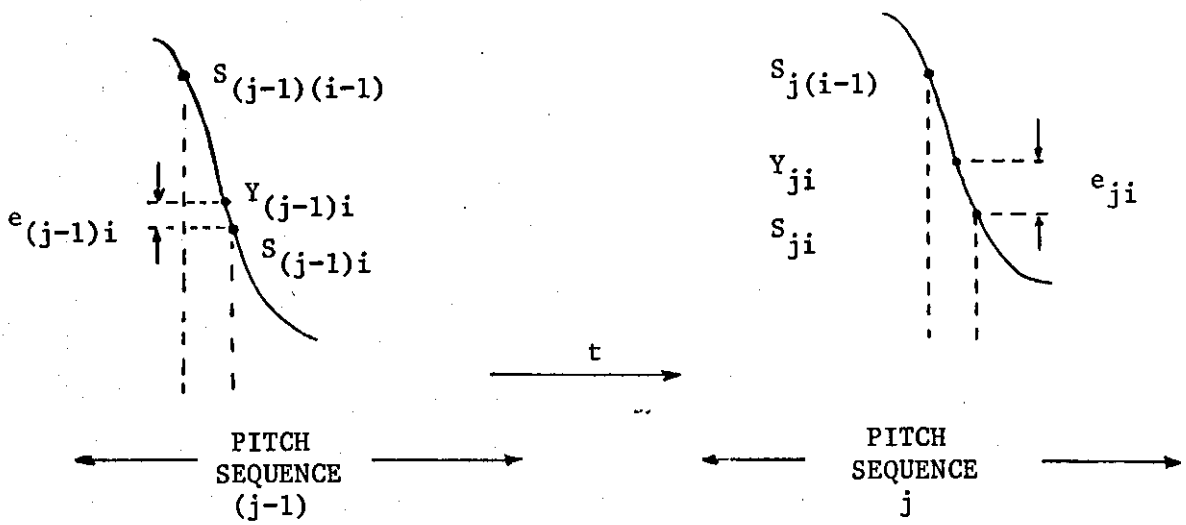

$S_{(j-1)(i-1)}$
$Y_{(j-1)i}$
$e_{(j-1)i}$
$S_{(j-1)i}$
t
$S_{j(i-1)}$
Y_{ji}
S_{ji}
e_{ji}
PITCH SEQUENCE (j-1)
PITCH SEQUENCE j

FIGURE 5.20.

prediction error is formed. After the calculation of the two error samples corresponding to adjacent pitch sequences, their difference is formed which is encoded into a binary form and transmitted. The decoder at the receiver recovers the S_{ji} input speech sample plus quantization noise associated with the encoded small amplitude difference sample.

Thus the PSDPE processing of voiced speech samples involves the formation of three differences:

a) between the input sample S_{ji} to be encoded and its predicted value Y_{ji},

b) between the sample $S_{(j-1)i}$ of the previous pitch sequence and its predicted value $Y_{(j-1)i}$, and

c) between the error samples obtained from (a) and (b).

The point to notice is that, unlike the predictor in the feedback loop of the PSFOD's DPCM encoder which used as its input the low-correlation e'_{ki} samples, the predictor in the PSDPE system operates on the correlated speech samples and therefore an improved prediction accuracy and snr performance is expected.

Before we present the block diagram of the PSDPE codec and describe its operation, we emphasize the fact that there is no need to specify the pitch period of the voiced speech with the accuracy required in Analysis-Synthesis coding techniques. By pitch we mean the similarities of the voiced waveform, measured between major-peaks of the signal. If peaks other than the maximum peak in the voiced speech oscillations are used as a measure of pitch period, the performance is virtually unaffected.

5.4.1. Operation of the PSDPE System.

For simplicity the operation of the PSDPE system is described when the predictor used to predict the S_{ji} and $S_{(j-1)i}$ samples is a first order one with a coefficient of unity, i.e. the predicted sample is equal to the previous one.

The block diagram of the PSDPE codec is shown in Figure 5.21. Suppose the input speech signal is sampled and $\{X_k\}$ is the sequence of voiced sampled presented to the input of the PSDPE encoder. Suppose also that $\{X_k\}$ contains the $\{S_k\}$ pitch sequences, where $k = 1,2,3,...$ and S_{ki} is the ith component of the kth pitch sequence.

When encoding voiced speech samples, the switch SW_1 is in position 1. Just prior to the instant where the first sample S_{11} of the first pitch sequence $\{S_1\}$ is removed from the Input Buffer and encoded, the Feedback Buffer is reset as it is also the integrator in the PSDPE feedback loop. Also, during the encoding of the first pitch sequence, switch SW_2 is open and the $\{S_1'\}$ and $\{U_1\}$ sequences are zero. Consequently when $\{S_1\}$ is processed, the sequence $\{E_1\}$ presented to the encoder is $\{-S_1\}$, i.e. $-S_{1i}$, $i = 1,2,....M$. $\{E_1\}$ is encoded to a binary sequence $\{L_1\}$ which is transmitted and also locally decoded to give the $\{R_1\}$ sequence. The samples contained in $\{R_1\}$ are $R_{1i} = -S_{1i} + n_{1i}$, $i = 1,2,.....M$, where n_{1i} is the quantization noise associated with the encoding of the E_{1i} sample. Because SW_2 is open and the U_{1i} samples are zero, a sequence $\{\hat{S}_1\}$ $\hat{S}_{1i} = S_{1i} - n_{1i}$, $i = 1,2,....M$ is obtained. This sequence is the decoded one and it is also produced at the receiver as shown in Figure 5.21b. During the encoding of $\{S_1\}$, the sequence $\{V_1\}$ is also formed whose components are $V_{1i} = \hat{S}_{1i} - \hat{S}_{1(i-1)}$,

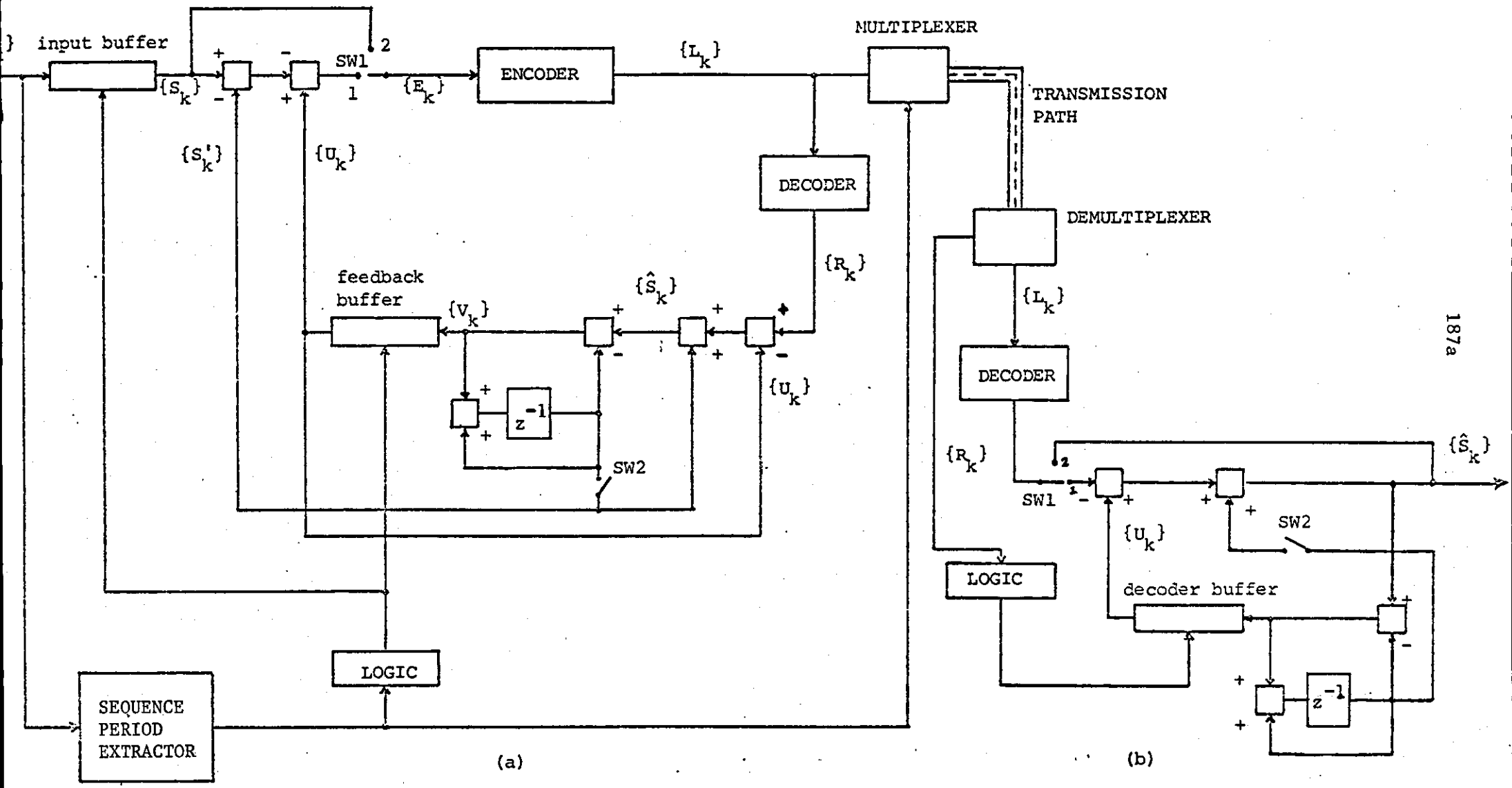


FIGURE 5.21 - (a) Encoder

(b) Decoder

$i = 1,2,3,....M$ and $\hat{S}_{10}$ is zero. $\{V_1\}$ is stored in the Feedback Buffer.

When the first pitch sequence has been encoded and decoded, the next sequence $\{S_2\}$ is removed one sample at a time from the Input Buffer. The SW_2 switch is now closed and remains in that condition for as long as voiced speech prevails. For the first S_{21} sample, S'_{21} is zero, U_{21} is equal to $\hat{S}_{11}$ and thus the error sample E_{21} is equal to $(\hat{S}_{11} - S_{21})$. E_{21} is encoded to produce a L_{21} binary word which is transmitted and locally decoded to yield $R_{21} = E_{21} + n_{21}$ where n_{21} is the noise sample associated with the encoding of E_{21}. The difference $U_{21} - R_{21} = \hat{S}_{11} - \hat{S}_{11} + S_{21} - n_{21}$ is then formed which is the recovered value of the input S_{21} sample, i.e. $\hat{S}_{21} = S_{21} - n_{21}$. $\hat{S}_{21}$ is then placed in the feedback buffer as the output of the integrator in the feedback loop is zero when S_{k1}, $k = 1,2,....$ is encoded. $V_{21} = \hat{S}_{21}$ is stored in the integrator.

When the second sample S_{22} is removed from the Input Buffer $S'_{22} = \hat{S}_{21}$, $U_{22} = \hat{S}_{12} - \hat{S}_{11}$ and the error sample E_{22} is equal to

$$E_{22} = U_{22} - (S_{22} - \hat{S}_{21}) = (\hat{S}_{12} - \hat{S}_{11}) - (S_{22} - \hat{S}_{21})$$

E_{22} is encoded to a binary form and transmitted as well as locally decoded to $R_{22} = E_{22} + n_{22}$. Then the difference between U_{22} and R_{22} is formed as:

$$U_{22} - R_{22} = (\hat{S}_{12} - \hat{S}_{11}) - (\hat{S}_{12} - \hat{S}_{11}) + (S_{22} - \hat{S}_{21}) - n_{22}$$

while the S_{22} sample is recovered as:

$$\hat{S}_{22} = U_{22} - R_{22} + S'_{22} = S_{22} - n_{22}$$

The sample stored in the integrator is then subtracted from $\hat{S}_{22}$ and the second component of $\{V_2\}$ namely V_{22} is

$$V_{22} = \hat{S}_{22} - \hat{S}_{21} .$$

For the remainder of the second pitch period, k = 2, and for the succeeding pitch periods, the PSDPE sequences and their components at the ith sampling instant are:

$$\left.\begin{aligned}
&\{S_k\}, \quad S_{ki} \\
&\{S'_k\}, \quad S'_{ki} = \hat{S}_{k(i-1)} \\
&\{U_k\}, \quad U_{ki} = V_{(k-1)i} = \hat{S}_{(k-1)i} - \hat{S}_{(k-1)(i-1)} \\
&\{E_{ki}\}, \quad E_{ki} = \left[\hat{S}_{(k-1)i} - \hat{S}_{(k-1)(i-1)}\right] - \left[S_{ki} - \hat{S}_{k(i-1)}\right] \\
&\{R_k\}, \quad R_{ki} = E_{ki} + n_{ki} \\
&\{\hat{S}_k\}, \quad \hat{S}_{ki} = S_{ki} - n_{ki} \\
&\{V_k\}, \quad V_{ki} = \hat{S}_{ki} - \hat{S}_{k(i-1)}
\end{aligned}\right\} \qquad (5.40)$$

When the input speech is unvoiced and therefore the correlation between samples is low, SW_1 switch is moved to position 2 and the speech samples are fed directly to the input of the encoder (see Figure 5.21). Because the dynamic range of unvoiced speech is substantially lower than that of voiced speech and similar to the dynamic range of the $\{E_k\}$ sequences, the amplitude range of the quantizer used by the encoder is the same during the encoding of both voiced or unvoiced speech samples. The unvoiced samples after encoded ino a binary form are transmitted. The decoder at the receiving end recovers the $\{R_k\}$ sequence of samples which contains the unvoiced input samples plus the associated noise produced

during encoding. A channel free of transmission errors is assumed.

As in the case of the PSFOD system, the objective of the PSDPE encoder is to reduce the dynamic range of the $\{E_k\}$ sequence of samples. To achieve this we acknowledge that adjacent pitch periods are generally of different lengths. This may result in the two bracketed terms in Equation 5.40 of being so different that E_{ki} overloads the encoder and large values of noise samples n_{ki} are produced. Consequently when forming the error sequence $\{E_k\}$, we apply similar rules with those presented in section 5.3.1.1.

Suppose that the locally decoded (j-1) pitch sequence has P components, i.e.

$$\{S_{(j-1)}\} = \hat{S}_{(j-1)1}, \hat{S}_{(j-1)2}, \ldots, \hat{S}_{(j-1)(N-\lambda)}, \ldots, \hat{S}_{(j-1)(P-\lambda+1)}, \ldots$$

$$\ldots \hat{S}_{(j-1)(P-1)}, \hat{S}_{(j-1)P}$$

and the next pitch sequence to be decoded has N components, i.e.

$$\{S_j\} = S_{j1}, S_{j2}, \ldots, S_{j(N-\lambda)}, S_{j(N-\lambda+1)}, \ldots, S_{j(N-1)}, S_{jN}$$

where λ is a constant ($\lambda << N,P$) and $P > N$.

In order to produce $\{\hat{S}_j\}$ at the receiver we encode $\{E_j\}$, where E_{ji} is given in Equation (5.40). As $P > N$ only N encoded components of $\{E_j\}$ have to be transmitted. The question arises, which N components of the $\{S_{j-1}\}$ sequence to use. If E_{ji} is formed as

$$E_{ji} = \left[\hat{S}_{(j-1)i} - \hat{S}_{(j-1)(i-1)}\right] - \left[S_{ji} - \hat{S}_{j(i-1)}\right] \quad i = 1,2,\ldots.N$$

then large values of E_{ji} result for i close to N, due to components in $\{\hat{S}_{j-1}\}$ being usually much smaller than those in $\{S_j\}$. This is because the pitch sequences are defined as the duration between samples which are large values in the voiced speech waveform following the closing of glottis, i.e. they correspond closely to the peak of the envelope of the voiced speech waveform.

Consequently $\{E_j\}$ is formed using the following samples from the jth and (j-1)th pitch sequences.

a) the first $(N-\lambda)$ components of $\{E_j\}$ are

$$\left[\hat{S}_{(j-1)1} - 0\right] - \left[S_{j1} - 0\right], \ldots, \left[\hat{S}_{(j-1)(N-\lambda)} - \hat{S}_{(j-1)(N-\lambda-1)}\right] - \left[S_{j(N-\lambda)} - \hat{S}_{j(N-\lambda-1)}\right]$$

b) the

$$\left[\hat{S}_{(j-1)(N-\lambda+1)} - \hat{S}_{(j-1)(N-\lambda)}\right] \cdots, \left[\hat{S}_{(j-1)(P-\lambda)} - \hat{S}_{(j-1)(P-\lambda-1)}\right]$$

$\{U_j\}$ components are not used.

c) the last λ components of $\{E_j\}$ are formed using the last $\lambda+1$ samples of $\{\hat{S}_{j-1}\}$ and $\{S_j\}$, i.e.

$$\left[\hat{S}_{(j-1)(P-\lambda+1)} - \hat{S}_{(j-1)(P-\lambda)}\right] - \left[S_{j(N-\lambda+1)} - \hat{S}_{j(N-\lambda)}\right], \ldots, \left[\hat{S}_{(j-1)P} - \hat{S}_{(j-1)(P-1)}\right] - \left[S_{jN} - \hat{S}_{j(N-1)}\right].$$

In the case where N > P, i.e. the duration of the pitch sequence to be encoded is larger than the duration of the previous decoded pitch sequence, $\{E_j\}$ is formed as follows:

a) its first P-λ components are:

$$\left[\hat{S}_{(j-1)1} - 0\right] - \left[S_{j1} - 0\right], \ldots, \left[\hat{S}_{(j-1)(P-\lambda)} - \hat{S}_{(j-1)(P-\lambda-1)}\right] - \left[S_{j(P-\lambda)} - \hat{S}_{j(P-\lambda-1)}\right]$$

b) $S_{j(P-\lambda+1)}, \ldots, S_{j(N-\lambda)}$ are the next components of $\{E_j\}$.

c) finally the last λ components of the error sequence are:

$$\left[\hat{S}_{(j-1)(P-\lambda+1)} - \hat{S}_{(j-1)(P-\lambda)}\right] - \left[S_{j(N-\lambda+1)} - \hat{S}_{j(N-\lambda)}\right], \ldots, \left[\hat{S}_{(j-1)P} - \hat{S}_{(j-1)(P-1)}\right] -$$

$$- \left[S_{jN} - \hat{S}_{j(N-1)}\right] .$$

Obviously in order for the PSDPE system to form the above error sequences, the voiced/unvoiced information and the duration of $\{S_k\}$ $k = 1,2,\ldots$ is required. This information is obtained from the Sequence Pitch Extractor (SPE) included in the system. The data at the output of the SPE, related to a certain segment of speech, is available to the PSDPE encoder and decoder before the encoding of the speech segment. This is because the input speech is delayed in the Input Buffer while the SPE extracts from the speech the necessary information and sends it to the "logic". The function of the "logic" is to control the SW_1 and SW_2 switches plus the Feedback Buffer, so that the rules of obtaining $\{E_k\}$ sequences having a minimum amplitude range are applied.

The same information is necessary for the "logic" at the receiving end to recover $\{\hat{S}_k\}$. The method of conveying the SPE data to the receiving end is the same as described in the synchronizing procedure of the PSFOD system. (section 5.3.1.2). Code-words B_i which contain information related with the duration of the pitch sequences, are multiplexed with $\{L_k\}$ and transmitted every ρT seconds. Upon receiving these B_i code-words, the decoder's "logic" is able to precisely calculate the duration of the incoming pitch sequences. The speech waveform transitions from a voiced sound to an unvoiced one are defined after the logic receives a certain number of zero B_i code-words.

5.4.2. Outline of the Simulation Procedure.

As in the case of the PSFOD computer simulations, the input speech data was first analysed and the obtained voiced/unvoiced information was stored on a magnetic tape. The PSDPE program could access this data from the following arrays and variables.

NVAUS : is the total number of voiced or unvoiced segments in the input speech waveform.

NPIT(J) : contains the duration of each pitch sequence occurred in the input speech waveform.

MV(J) : contains the number of pitch sequences inside every voiced segment of the input speech.

MU(J) : contains the number of samples inside every unvoiced segment of the input speech.

The general structure of the PSDPE programming procedure is similar with that of the PSFOD codec, already presented in section

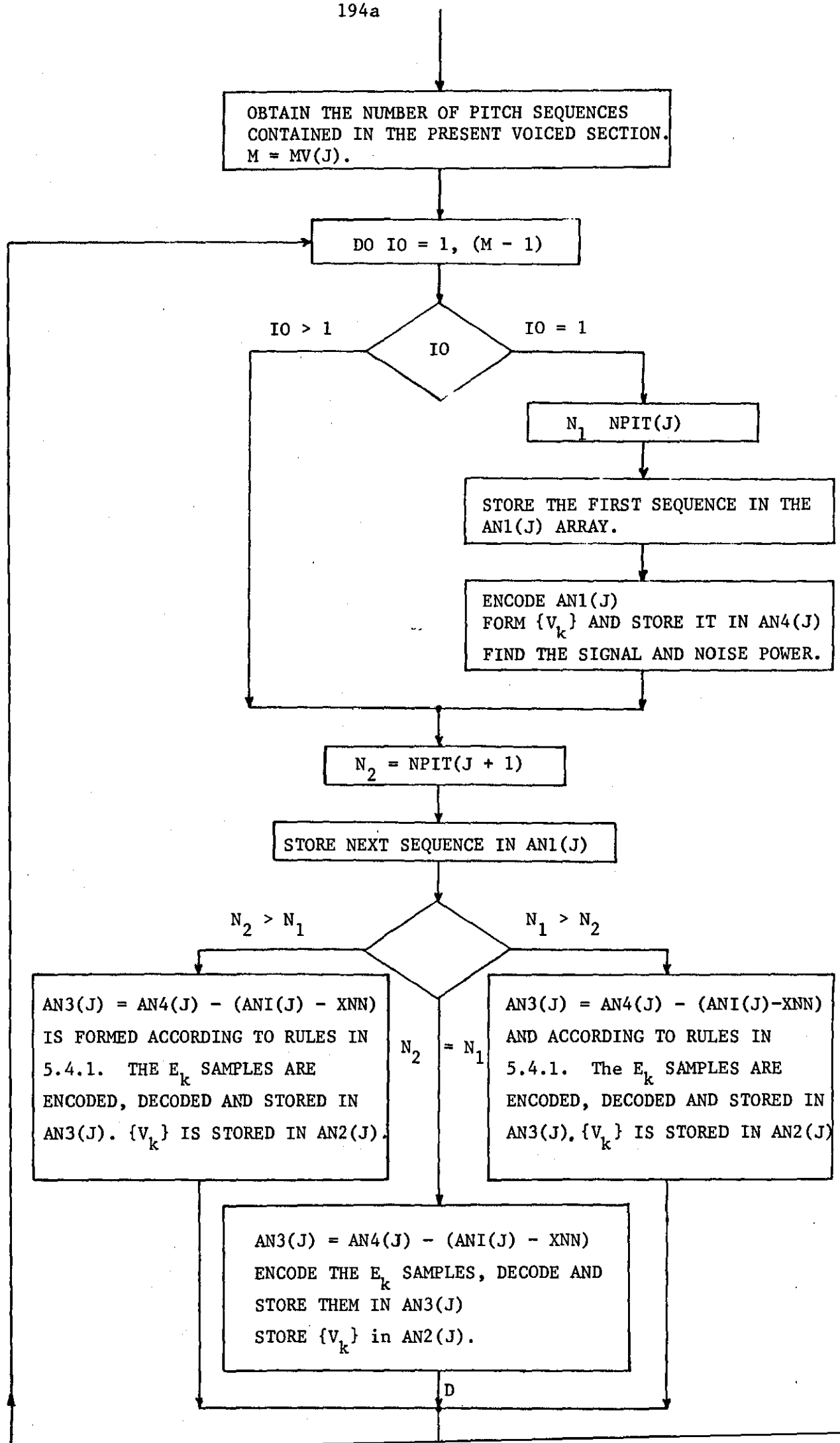
5.3.2. What will be outlined in this section is the part of PSDPE simulation procedure which processes the voiced segments of the input speech. The block diagram of the "voiced" part of the PSDPE program is shown in Figure 5.22.

Once the program decides that the next speech segment is a voiced one, the number M of pitch sequences is obtained from the MV(J) array, knowing M the program goes to an IO Do Loop with IO = 1,...,(M-1). The value of IO is then examined. If IO is equal to one, i.e. the first pitch sequence of the voiced segment is to be encoded, its length N_1 is obtained from NPIT(J) and the following N_1 input samples are stored in AN1(J). The samples are then removed one by one from AN1(J), encoded and decoded. The $\{V_1\}$ sequence is formed as $V_{1i} = (\hat{S}_{1i} - \hat{S}_{1(i-1)})$ where $\hat{S}_{10} = 0$, and stored in the AN4(J) array. At the same time the input speech power and the power of the noise associated with the decoded samples, is calculated.

After processing the first pitch sequence the procedure reads from NPIT(J) the length N_2 of the second pitch sequence, and this is also the point where the program transfers its operation if IO > 1. The next N_2 input samples are stored in AN1(J) and N_1, N_2 are compared. If $N_1 = N_2$, $\{E_k\}$ is sequentially formed according to:

$$E_{ki} = \left[\hat{S}_{(k-1)i} - \hat{S}_{(k-1)(i-1)}\right] - \left[S_{ki} - \hat{S}_{k(i-1)}\right] \quad i = 1,2,....N_1$$

The E_{ki} values are stored in AN3(J) and then encoded and decoded. The decoded speech samples are stored back to AN3(J). For example, let us consider the procedure during the nth sampling instant of the kth pitch sequence. E_{kn} is formed according to the above equation


OBTAIN THE NUMBER OF PITCH SEQUENCES CONTAINED IN THE PRESENT VOICED SECTION. M = MV(J).
DO IO = 1, (M - 1)
IO > 1
IO
IO = 1
N1 NPIT(J)
STORE THE FIRST SEQUENCE IN THE AN1(J) ARRAY.
ENCODE AN1(J)
FORM {Vk} AND STORE IT IN AN4(J)
FIND THE SIGNAL AND NOISE POWER.
N2 = NPIT(J + 1)
STORE NEXT SEQUENCE IN AN1(J)
N2 > N1
N1 > N2
AN3(J) = AN4(J) - (ANI(J) - XNN) IS FORMED ACCORDING TO RULES IN 5.4.1. THE Ek SAMPLES ARE ENCODED, DECODED AND STORED IN AN3(J). {Vk} IS STORED IN AN2(J).
N2 = N1
AN3(J) = AN4(J) - (ANI(J)-XNN) AND ACCORDING TO RULES IN 5.4.1. The Ek SAMPLES ARE ENCODED, DECODED AND STORED IN AN3(J). {Vk} IS STORED IN AN2(J)
AN3(J) = AN4(J) - (ANI(J) - XNN)
ENCODE THE Ek SAMPLES, DECODE AND STORE THEM IN AN3(J)
STORE {Vk} in AN2(J).
D

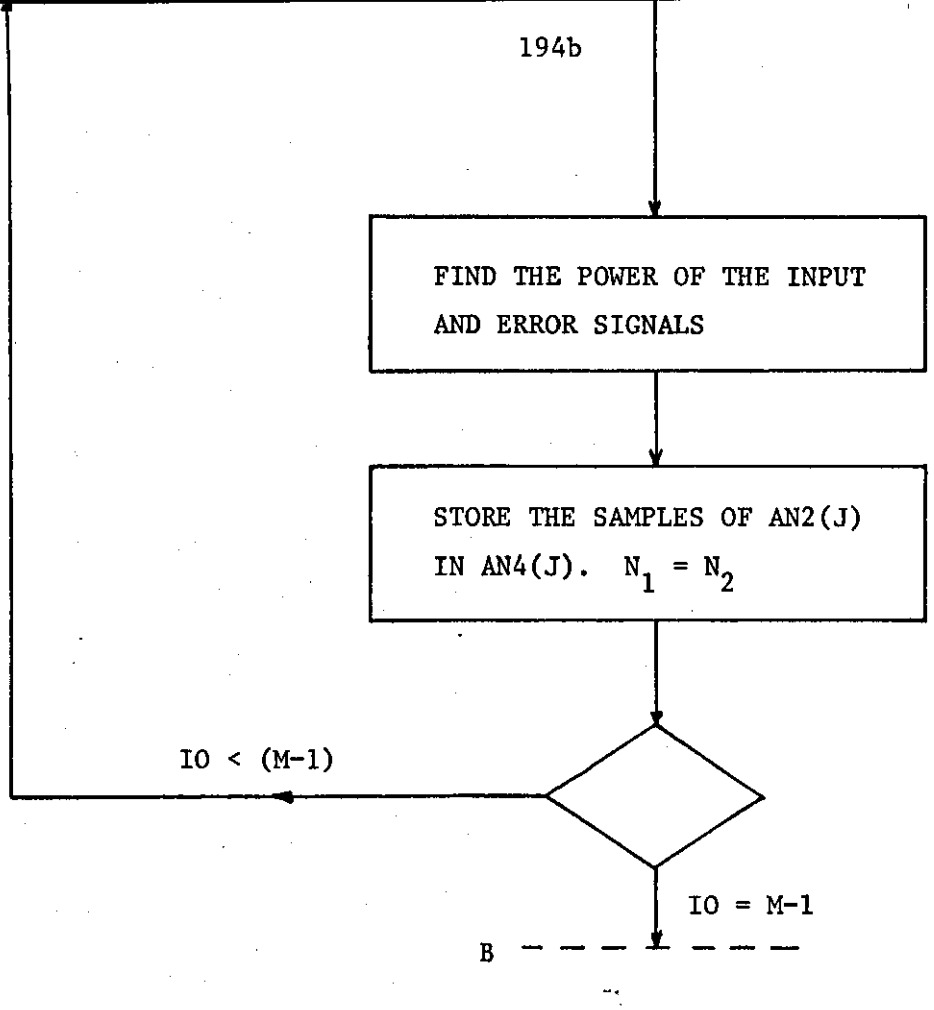


FIGURE 5.22 - Flowchart of the "Voiced" part of the PSDPE Simulation Procedure.

and stored in AN3(n). Then E_{kn} is encoded, decoded and the S_{kn} speech sample is recovered as $\hat{S}_{kn}$. $\hat{S}_{kn}$ is stored back in AN3(n) since E_{kn} is not of further use. The V_{kn} sample, $V_{kn} = \hat{S}_{kn} - \hat{S}_{k(n-1)}$, is then formed and stored in AN2(n).

After processing all the samples in the input pitch sequence, the reference point D follows which is the common point of all the three $N_1 = N_2$, $N_1 > N_2$, $N_1 < N_2$, programming paths.

If, on the other hand, $N_1 \neq N_2$ the samples of the kth pitch sequence are sequentially removed from AN1(J) and form the E_{ki} samples according to the rules described in section 5.4.1. Again the final values stored in AN3(J) are the recovered input samples $\hat{S}_{ki}$ while the $\{V_k\}$ sequence is stored in AN2(J). When all the input samples are processed and the procedure goes to reference level D, the power of the input samples and the power of the noise associated with the decoded input samples is measured in order to be used later, when the snr of the codec is calculated. Before the program returns to the beginning of the IO Do Loop, to process further pitch sequences, AN4(J) and N_1 are made equal to AN2(J) and N_2 respectively. When IO = M-1 the program proceeds to a reference level B and the remaining simulation procedure is the same with that shown in Figure 5.13 in the PSFOD section.

5.4.3. Experimental Procedure, Results.

The PSDPE system was simulated on a Hewlett Packard 2100 A computer. The input speech was the same as that used in the PSFOD experiments. That is, short sentences of speech (minimum duration of 1.5 seconds) band limited to 3.4 kHz and sampled at 8 kHz per

second. After the processing of the input speech the snr of the PSDPE was calculated and compared to that of an ADPCM system using the same number of quantization levels. To determine the snr produced by the PSDPE and ADPCM encoders, the noise signal was formed as the difference signal between the input samples and the corresponding decoded samples which had been filtered by an 8 order Butterworth recursive filter having a cut-off frequency of 3.4 kHz. The snr calculation procedure is described in Chapter IV, section 4.3.2.

It was decided that the encoder employed in the PSDPE system, to encode the error sequences $\{E_k\}$ and the unvoiced speech samples, would be an adaptive quantizer. This is because the correlation of these samples, was found to be much lower than that of voiced speech and the use of an ADPCM encoder could actually reduce the performance of the system. The adaptive quantizer used in the simulations was Jayant's adaptive quantizer, described in the PSFOD section and whose adaptation coefficients for 3 and 4 bits per sample quantization are given in Table 5.1.

In section 5.4.1, the operation of the PSDPE-AI system was described whose error samples E_{ki} are formed according to Equation (5.40). Instead of taking the $\hat{S}_{(k-1)(i-1)}$ and $\hat{S}_{k(i-1)}$ decoded samples as being the predicted values of $\hat{S}_{(k-1)i}$ and S_{ki}, a linear predictor can be used whence $\hat{S}_{(k-1)i}$ and S_{ki} are predicted as a weighted combination of the previous decoded samples. This system called PSDPE-AF, has also been examined.

Figure 5.23 shows graphs of snr against input power. Curve (a) is for a 3 bits quantization PSFOD-AI system while curve (b)

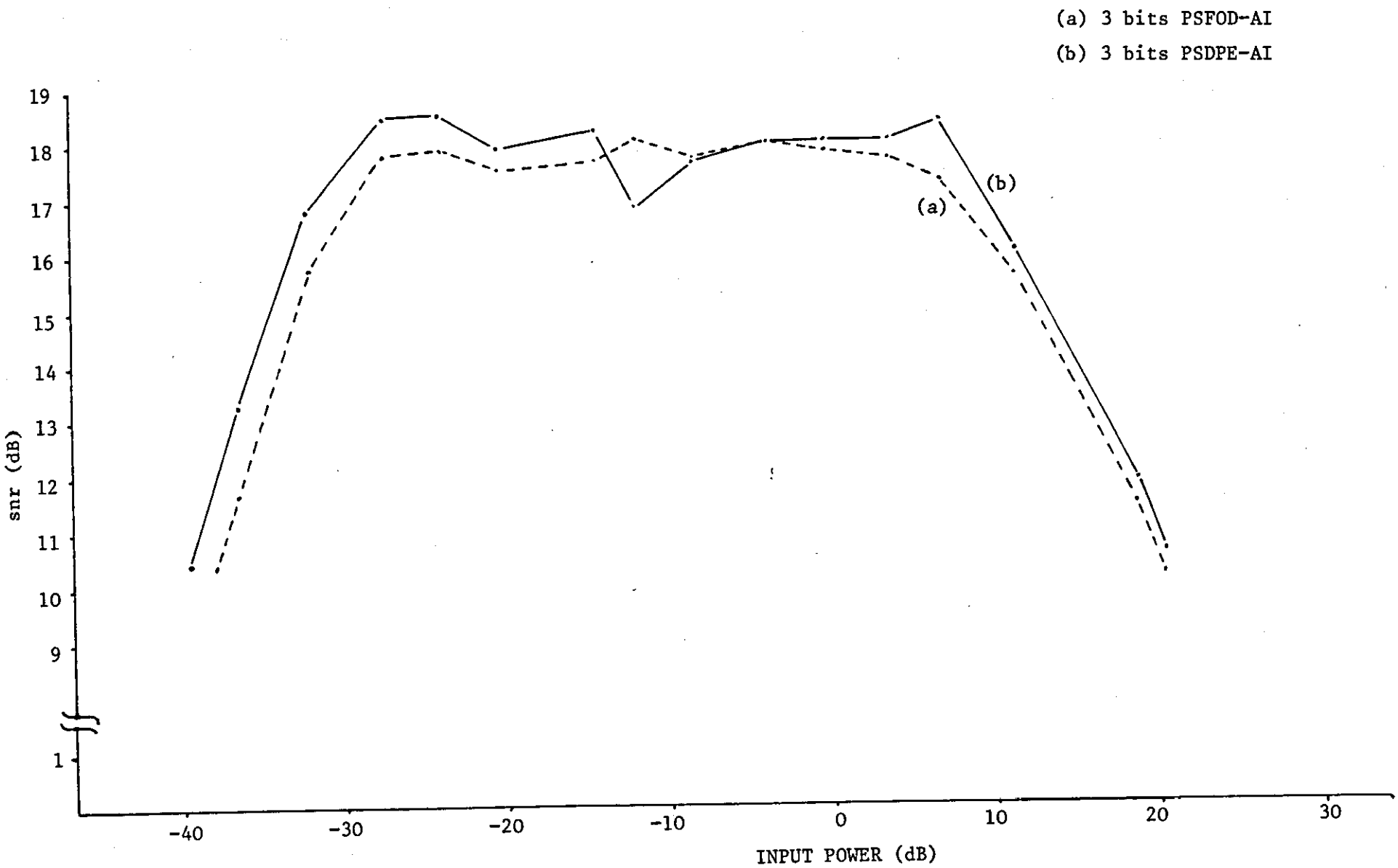


FIGURE 5.23 - snr as a Function of Input Power.

corresponds to a 3 bits quantization PSDPE-AI codec. It is shown that the latter system has an advantage of approximately 0.5 dBs over PSFOD-AI. However, this snr advantage is increased when prediction is used in the PSDPE encoder. Figure 5.24, curve (b) is for the PSDPE-AI system. When $\{E_k\}$ is formed as:

$$E_{ki} = \left[\hat{S}_{k-1)i} - a_1 \hat{S}_{(k-1)(i-1)}\right] - \left[S_{ki} - a_1 \hat{S}_{k(i-1)}\right]$$

$i = 1,2,...N$, i.e. a first order fixed predictor is used, curve (a) is obtained. Consequently a first order predictor provides a further 2 dBs improvement over curve (b). Note that when a fixed coefficient predictor is used in the PSFOD system an additional snr of 1 dB is obtained. Curve (e) is for the PSDPE-AF system having a fixed two coefficient predictor, and it shows an snr increase of approximately 0.6 dBs over the one coefficient prediction case of (a) curve. Further increase in the order of the predictor resulted to considerably reduced snr values, probably because the coefficients were not well matched to the statistics of the input signal. In the same Figure, curve (c) is for the PSDPE-AI system when 10% of the pitch sequences selected in a random basis, were subjected to a 10% error in locating the correct pitch period. In this particular experiment after introducing the errors, the pitch sequences were allowed to commence at any amplitude level and this resulted in the loss in snr shown between curves (b) and (c). However, if a large percentage error occurs in locating the first positive peak of a pitch period and a nearby peak is used instead, the loss in the snr performance of the system is smaller than that shown by curve (c).

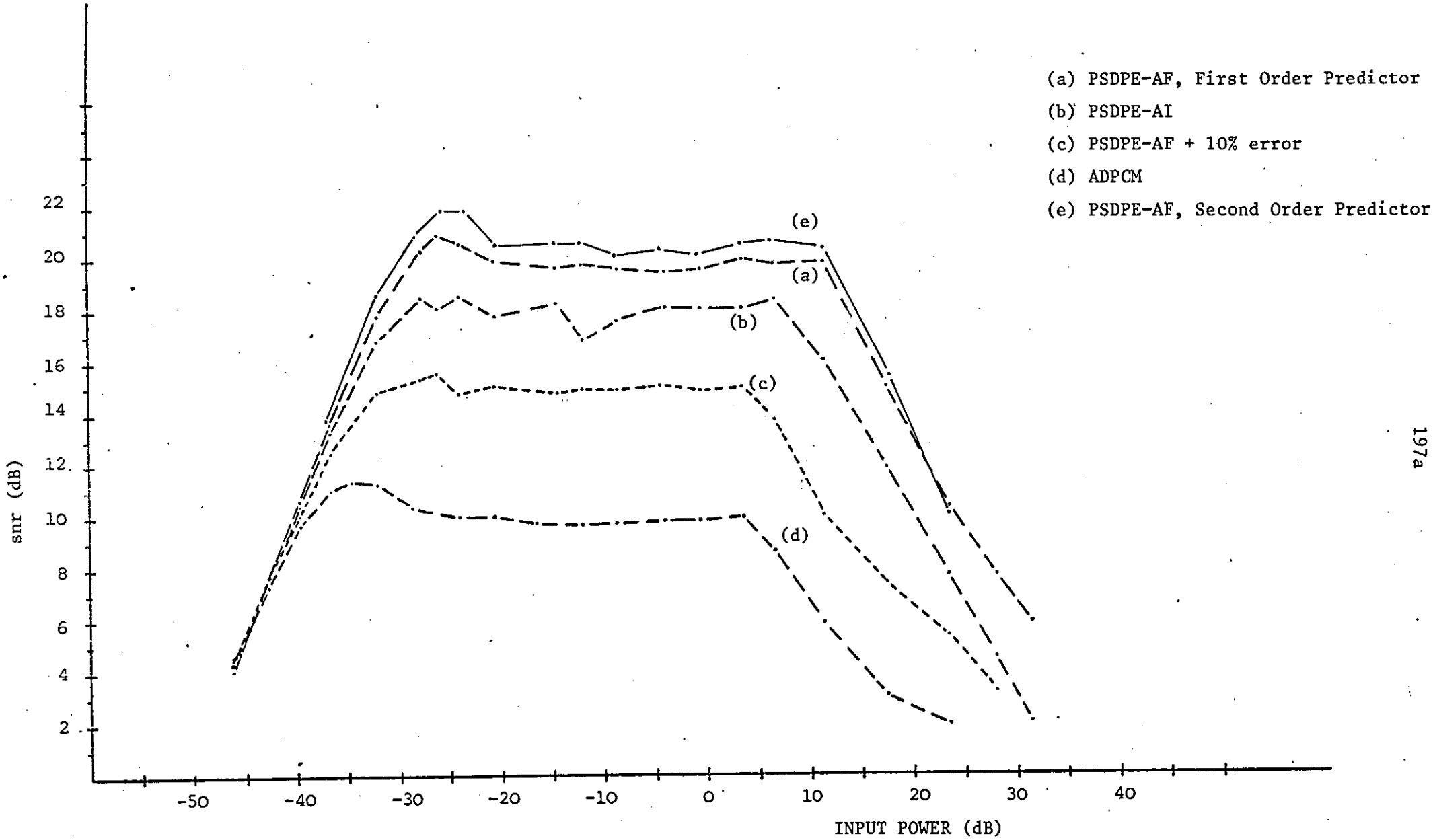


FIGURE 5.24 - snr as a Function of Input Power, 3 bits/sample.

During the PSDPE-AF experiments the simulation procedure was modified to observe the effect of using a first order ADPCM encoder, instead of APCM, when encoding unvoiced speech. It was found that the snr of this scheme was similar to the PSDPE-AF snr performance.

Finally when the E_{kj} samples are formed according to Equation (5.40), it is possible that the terms inside the brackets are of opposite sign and the magnitudes of the terms are added instead of subtracted causing large amplitude E_{ki} samples. In order to observe the effect of forming the E_{ki}'s samples so that E_{ki} is always smaller than the terms in the brackets, it was arranged that when these two terms were of opposite sign, the sign of one of them was inverted. The snr curve (b) of Figure 5.25 is for a 8 quantization levels PSDPE-AI system when the sign of the first term $\left[\hat{S}_{(k-1)i} - \hat{S}_{(k-1)(i-1)}\right]$ is inverted when necessary. Curve (a) is for the 8 levels PSDPE-AI codec. Figure 5.25 shows that a snr gain of approximately 1 dB is obtained from the above scheme. However, the information that one of the terms which form E_{ki} changed its sign, has to be conveyed to the receiver in order to recover the correct $\hat{S}_{ki}$ input sample. This means an increase in the transmission bit rate of the system. No further investigations were carried out for this scheme.

5.4.4. Note on Publications.

A paper entitled "Pitch Synchronous Differential Predictive Encoding System" in co-authorship with Dr. R. Steele, has been published in Electronic Letters of I.E.E., Vol.12, number 5, July 1976. This paper is an abridged version of the PSDPE-AI and

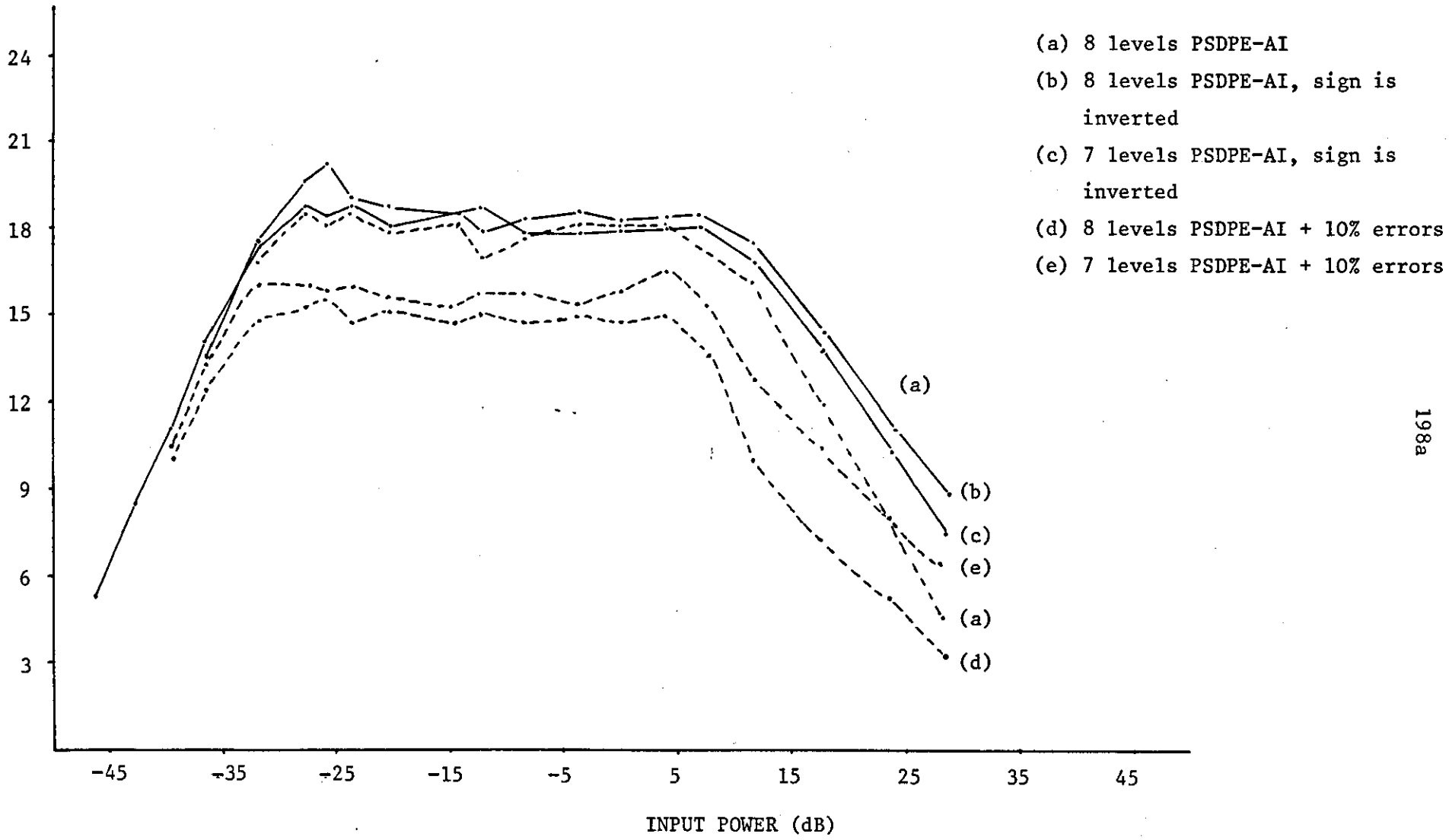


FIGURE 5.25 - snr as a Function of Input Power.

PSDPE-AF systems. Sections 5.3 and 5.4 of this chapter provided the material for two papers delivered by Dr. R. Steele.

1) C.S. Xydeas, R. Steele, "Pitch Synchronous Encoding methods of speech signals", I.E.E.E. International symposium on Information Theory, Ronneby, Sweden, 20-24 June 1977.

2) R. Steele, C.S. Xydeas, "Pitch Synchronous Encoding of Speech", I.E.R.E. Communication Group Colloquium on Digital Encoding of Speech, The Royal Institution, 22 Feb. 1977.

5.5 DISCUSSION.

At the beginning of this chapter the importance of an efficient predictor was emphasized when used in a Differential encoding system. The "prediction problem" as applies to DPCM systems was considered and the estimation accuracy of three prediction techniques examined when:

i) predicting input speech samples,

ii) used in the feedback loop of a DPCM System.

After presenting the results of the above investigations, various research directions were suggested which could produce an improved prediction scheme and thus an improved DPCM encoder. The most promising scheme using two different types of predictors in the DPCM feedback loop was examined. As a consequence, the PSFOD and PSDPE pitch synchroneous systems were developed and gave substantially improved encoding performance when compared to the ADPCM system.

We consider now the work (brought to our attention) on the "prediction problem" of other researchers and specifically on some of the prediction projects suggested in Figure 5.8. The Stochastic approximation and modified Kalman predictors when used in a DPCM and operating over a wide range of transmission bit rates have been examined by Gibson[111]. He concluded that only at bit rates between 16 and 20 Kbits/sec. these predictors had a definite advantage over long-term fixed predictors. In addition he showed that in the range of 12.8 to 32 Kbits/sec, transmission rates, the modified Kalman algorithm was always better (but for an improvement in snr of no more than 1.5 dBs) than the Stochastic approximation one. He acknowledged the fact that the prediction accuracy in both algorithms, depends upon the power of the input speech signal, but his work is not extended beyond this point.

Pirami and Scagniola[112] examined the DPCM encoder using a Kalman predictor with fixed prediction coefficients. His simulations demonstrate the need of adaptive coefficients which follow the variations of the vocal tract.

Evci[113] is working to improve the Stochastic approximation algorithm and make it independent from the input power. Furthermore he examines new sequentially adaptive prediction algorithms which converge fast to the vocal tract characteristic and are robust to quantization noise.

Research establishments in the States show an interest in Pitch Synchroneous differential type encoding systems.[114,115,116]. In particular we mention the work of Jayant[114] and Goldberg[115].

Jayant reported computer simulation results obtained from a Pitch-adaptive DPCM encoder (PA-DPCM), with a two-bit quantizer and a fixed spectrum predictor. The system is intended to operate at the transmission bit rate of 16 Kbits/sec. Although the system as described in (114) shows little in common with the PSDPE system, it can be shown that Jayant examined:

a) a two-bit PSDPE-AI encoder,

b) a two-bit PSDPE-AF encoder using three fixed prediction coefficients,

c) a two-bit PSDPE system where $E_{ki} = S_{ki} - \hat{S}_{(k-1)i}$, i.e. $\hat{S}_{(k-1)(i-1)}$ and $\hat{S}_{k(i-1)}$ in Equation (5.40) are zero.

Case (c) is not included in our PSDPE investigations.

The performance of the PA-DPCM system was examined using:

i) an Average Magnitude Difference Function (AMDF), and

ii) an Autocorrelation pitch extractor.

The simpler AMDF algorithm showed better snr values. The maximum signal-to-noise ratio gain of PA-DPCM over a non-pitch DPCM encoder is reported to be approximately 4 dBs which is considerably lower than the snr advantage obtained in our simulations. There are three possible explanations:

i) The fact that adjacent pitch periods are of different lengths was not considered in the PA-DPCM system. Thus when forming the error $\{E_k\}$ sequence, samples of large amplitudes can occur.

ii) The adaptive quantizer of the PA-DPCM encoder used as adaptation coefficients $H_1 = 0.95$ and $H_2 = 1.1$ giving a slow

adaptation rate for the quantization step size. This is because it has been assumed that the amplitude range of $\{E_k\}$ changes slowly. Consequently any sudden changes in the amplitude range of $\{E_k\}$, due for example to differing lengths in adjacent pitch sequences or to an error in the location of the correct pitch sequence, overloads the quantizer causing large quantization errors.

iii) The performance of the AMDF algorithm is lower than the nearly optimum pitch extraction technique used in our experiments.

Goldberg[115] examined the performance of a 16 Kbits/sec. Pitch Synchroneous system similar to PSFOD. His encoder employs two predictors. The first predictor estimates the sample to be encoded as a weighted value of the corresponding sample one pitch period before. A difference sequence is produced from this pitch loop prediction while a second linear predictor operating on this difference sequence further reduces the variance of the error signal. Both predictors are adaptive and their coefficients together with the pitch period information are separately transmitted to the receiving end. The system was evaluated using three different quantizers, i.e., Jayant's, Forney's and a Fixed frame quantizer. Goldberg concluded that at 16 Kbits/sec. and at low transmission error bit rates, the Pitch Synchron ous Differential system out performs the CVSD adaptive Delta Modulator. Only at high error bit rates (10^{-2}) does CVSD have a superior performance.

CHAPTER VI

DYNAMIC RATIO QUANTIZATION TECHNIQUES

6.1 INTRODUCTION.

The "prediction Problem" has been examined in the previous chapter, where it was shown that the performance of a DPCM system is determined by the predictor and the quantizer. It is the quantizer which is the subject of this chapter.

In section 5.1, Equation 5.6, i.e.

$$snr_D = snr(imp) + snr(PCM)$$

indicates that the snr of a DPCM system is the summation of the signal-to-noise ratio produced by the quantizer, snr (PCM), plus another term which depends upon the estimation accuracy of the predictor. The higher the snr of the quantizer, the higher is the snr of the DPCM system. Thus in order to improve the performance of a Differentially encoding system, we considered the problem of "how to design an efficient quantizer" with an improved snr compared to known quantization techniques.

Since adaptive quantizers provide a superior snr over non-adaptive, i.e. fixed quantizers, our investigations were focussed on methods of adaptive quantization.

In the first part of this chapter some well-known adaptive quantizers are discussed and a generalized model of an adaptive quantizer is presented. Then our solution to the "efficient quantization" problem is given. This is a novel quantization technique called Dynamic Ratio Quantization (DRQ). The theory of

Dynamic Ratio Quantization is presented and several DRQ quantizers are examined. Their performance is evaluated through computer simulations. A DRQ scheme called the Envelope Dynamic Ratio quantizer, Envelope-DRQ, is then examined in detail. The theory of the quantizer is presented together with computer simulation results which show an improvement compared to one word memory APCM system. Finally the simplicity of implementing the Envelope-DRQ is described.

6.2 ADAPTIVE QUANTIZATION TECHNIQUES.

A quantizer accepts analogue samples and imposes amplitude restriction on them such that each analogue sample is forced, i.e. quantized, to the nearest one of a finite number of available levels. These quantization levels need not be equi-spaced or time invariant.

Adaptive quantizers are time-variant, i.e. they have the ability to change the amplitude range of their quantization levels while maintaining the same number of levels. In this way the quantization noise, which is the difference between the quantized samples at the output of the quantizer and the analogue samples at its input, will vary as a function of the input samples. A desirable condition is to arrange for the quantization noise to be proportional to the power of the input analogue samples. This results in a constant signal-to-noise ratio as a function of input power.

As the quantizer is required to adapt to the variations in the input sequence of samples, it seems appropriate to use this sequence to control the adaptation system. Unfortunately this method necessitates the transmission of the adaptation information along with the binary

representation of the input samples. This multiplexed "side" information results in an undesirable increase in the bandwidth of the transmitted signal. A popular approach is to up-date the quantization characteristic as a function of the current and/or previous quantization levels, information which is available at the receiver.

In this section we present the main adaptive quantization techniques and discuss their limitations. In particular Jayant's[41] the One Word Memory adaptive procedure is described in detail while the Variance Estimating quantizer[39] is briefly considered. The adaptive Pitch Compensating quantizers of Cohn, Melsa[68] and Qureshi, Formey[69] are then presented as an extention of Jayant's work in an attempt to improve the quantizer's dynamic performance, while keeping its static performance satisfactory. A generalization of adaptive quantization follows and the concept of the DRQ quantization method is then discussed.

Throughout this chapter error free transmission channels are assumed. Consequently the various techniques[116] for modifying the adaptation algorithms of an adaptive quantizer in order to combat transmission errors are not described.

6.2.1. Jayant's Adaptation Procedure.

Consider the n-level uniform quantizer shown in Figure 6.1 whose thresholds $T_{(j)}$ and output quantization levels $Q_{(j)}$ are defined by

$$\begin{aligned} T_{(j)} &= \pm j\delta && j = 1,2,\ldots, (\tfrac{n}{2} - 1). \\ Q_{(j)} &= \pm (j + \tfrac{1}{2})\delta && j = 0,1,\ldots, (\tfrac{n}{2} - 1). \end{aligned} \tag{6.1}$$

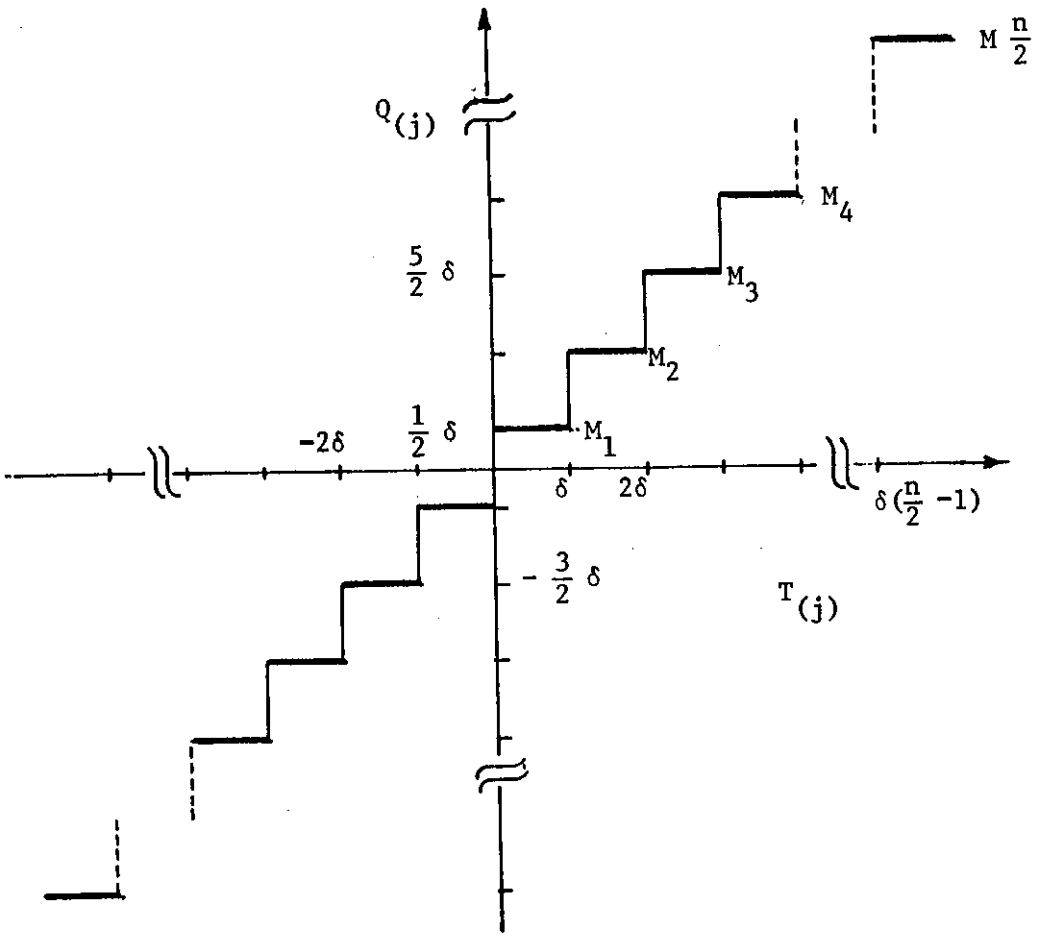


FIGURE 6.1 - n-level Quantizer whose Step Size δ is Adaptive.

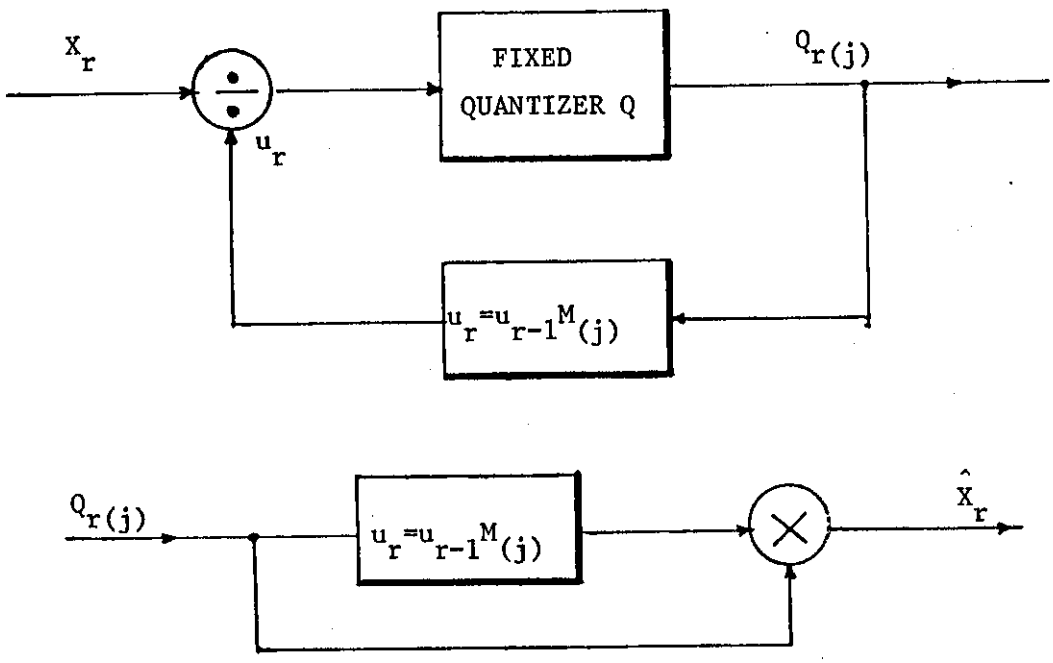


FIGURE 6.2 - Jayant's Adaptive Quantizer.

δ is an adaptable step size whose value at the (r+1)th sampling instant assumes the value of

$$\delta_{r+1} = \delta_r \cdot M_{(j)} \quad , \qquad j = 0,1,\ldots,(\frac{n}{2} - 1) \qquad (6.2)$$

δ_r is the value of the step size at the rth sampling instant while $M_{(j)}$ is a time-invariant expansion-contraction coefficient whose value depends on $Q_{(j)_r}$, i.e. the quantization output level at the rth instant. Equation (6.2) defines Jayant's adaptation algorithm.

When the values of the $M_{(j)}$ coefficients are properly selected, the quantizer has at each sampling instant a step-size which tends to be matched with the variance of the input to the quantizer sequence of samples. Thus the quantizer expands or contracts its amplitude quantization range according to the variance of the incoming input samples.

Alternatively Jayant's adaptive quantizer can be viewed as one which normalizes the input samples X_k with a state variable u_k, and uses a fixed range quantizer for the quantization of the resulting ratio. This representation is shown in Figure 6.2 and follows the general model of an adaptive quantizer which is described at the end of this section. In order to justify the model in Figure 6.2,let us consider the values which Equation (6.2) assigns to δ at four consecutive sampling instants k-1, k, k+1, k+2, given that an arbitrary sequence $\{X_k\}$ is quantized. Assume that the output of the quantizer at the (k-1)th sampling instant is:

$$Q_{k-1} = \frac{1}{2} \delta_{k-1}$$

and thus the next step size is equal to

$$\delta_k = \delta_{k-1} \cdot M_1$$

In the same way, if

$$Q_k = \frac{3}{2}\,\delta_k$$

the step size of the quantizer at the (k+1)th sample is

$$\delta_{k+1} = \delta_k \cdot M_2 = \delta_{k-1}\cdot M_1 \cdot M_2$$

and if $$Q_{k+1} = \frac{1}{2}\,\delta_{k+1}$$

then $$\delta_{k+2} = \delta_{k+1}\cdot M_1 = \delta_{k-1}\cdot M_1 \cdot M_2 \cdot M_1 =$$

$$= \delta_{k-1}\cdot u_{k+2} \qquad (6.3)$$

Consequently at the (k+2)th sampling instant the step size δ is equal to $\delta_{k-1}\cdot u_{k+2}$, and in general

$$\delta_r = \delta_{init}\cdot M_1^{m_1}\cdot M_2^{m_2}\cdot,\ldots,\ M^{m_{n/2}}$$

$$= \delta_{init}\cdot u_r \qquad (6.4)$$

where δ_{init} is the initial value of δ at time r = 0, m_i is the number of occurrence of $M_{(i)}$ $i = 1,2,\ldots,\frac{n}{2}$ and $r = m_1 + m_2 +,\ldots,m_{\frac{n}{2}}$. The value of the u_r variable depends upon the variance of the input samples $\{X_k\}$. Equation 6.4 leads to the quantizer shown in Figure 6.2 as quantization with a step size δ_r is equivalent to division of the input sample by u_r followed by a fixed step size δ_{init} quantizer.

It can be seen from the above procedure (Equation 6.3) that the state variable u_r is updated in the same way as δ_r in Equation (6.2). Therefore the value of the state variable at the rth sampling instant is equal to:

$$u_r = u_{r-1} \cdot M_{(j)}, \qquad j = 1,2,\ldots, \frac{n}{2} . \tag{6.5}$$

where $M_{(j)}$ is the time-invariant expansion-contraction coefficient whose value depends on $Q_{(j)_r}$.

The values of the $M_{(j)}$ coefficients can be optimized for a particular speech segment so that a maximum snr is obtained. The quantizer's performance is not in general critically dependent on the $M_{(j)}$ values. The basic requirement is that the rate of increase in the value of u_r should be larger than its rate of decrease. Thus when $M_{(j)} < 1$ the values of the coefficients are always close to unity while when $M_{(j)} > 1$, the coefficients can assume values much larger than unity. This is because the state variable u_r should respond rapidly to a sudden increase in the amplitude level of the input signal and hence avoid the $X_k >> u_k$ situation which results in large values of $\frac{X_k}{u_k}$ ratio and overload of the fixed quantizer. On the other hand when the variance of the input signal decreases slowly, a fast adaptation of u_r towards X_r can result in an over-reduction in the value of u_r and to an undesired $X_k >> u_k$ situation. To clarify the relationship between the values of the $M_{(j)}$ coefficients and the performance of the quantizer, let us firstly define the design objectives of Jayant's adaptation procedure.

Consider the ideal case where a unity variance signal $\sigma_x^2 = 1$ with known probability density function is to be quantized. The optimum δ_r or u_r, indicated by $\hat{\delta}_r$ or $\hat{u}_r$, in a minimum mean square quantizing sense, is obtained using Max's[35] method. Note that the optimum $\hat{u}_r$ has only to be properly scaled to ρu_r when the input signal is scaled with the constant ρ. If we now consider that the

power of the input signal is not constant but it is changing with time, then it is not possible for u_r to always assume the optimum value $\hat{u}_r$. Consequently two design objectives can be defined, one for the "static" mode and another for the "dynamic mode" of operation.

i) The "static" operation is referred to the case where the signal's rms value is ρ over a long sequence of input samples. In this case the values of the $M_{(j)}$ coefficients must be such that u_r approximates $\rho\hat{u}_r$.

ii) The "dynamic" mode of operation is related to the case where the signal's rms value changes from ρ_1 to ρ_2. The values of the $M_{(j)}$ coefficients are required to provide a fast "adaptation response", so that $u_r = \rho_1 \hat{u}_r$ assumes rapidly its new value, i.e. $u_r = \rho_2 \hat{u}_r$.

It can be shown[43] that in the static mode of operation the normalizing variable u_r will continue to expand or contract until a steady state $E[u_r] = \bar{u}_r$ is reached, where

$$\sum_{i=1}^{n/2} P[Q_{(i)}] \log_2 M_{(i)} = 0 \qquad (6.6)$$

$P[Q_{(i)}]$ is the probability of selecting the $M_{(i)}$ coefficient when the input sample to u_r ratio is between the (i-1)th and ith quantization thresholds, i.e. $P[Q_{(i)}] = P[T_{(i-1)} < X_r/\bar{u}_r \leq T_{(i)}]$. It is also established[43] that the steady state fluctuation of u_r is related to

$$R = \log_2 \left[\frac{\max M_{(i)}}{\min M_{(i)}} \right]$$

and the "adaptation response" is inversely related to R. The

closer the $\frac{\max M_{(i)}}{\min M_{(i)}}$ ratio is to unity, the narrower is the shape of the probability distribution of u_r around the value of $\hat{u}_r$ and thus the better the performance of the quantizer in the static mode of operation. However this leads to a long "adaptation response" and consequently to overload noise. Therefore the values of the $M_{(j)}$ coefficients must offer a compromise between the "static" and "dynamic" objectives so that the overall snr performance of the quantizer is satisfactory.

Figure 6.3 shows the values of the $M_{(j)}$ coefficients as quoted from reference (41). A more detailed diagram of Jayant's quantizer is shown in Figure 6.4.

6.2.2. The Variance Estimating Quantizer.

In the Variance Estimating Quantizer (VEQ), studied by Noll[40], Stroh[39] and Castelino[118], the input signal is normalized by the square root of a maximum likelihood estimate of its variance and the resulting ratio is quantized with a fixed quantizer.

The block diagram of VEQ is shown in Figure 6.5. The normalizing variable u_r is made proportional to a moving estimate of the decoded signal's standard deviation in order to obtain a unity variance ratio signal which can then be optimally quantized. Thus

$$u_r = \sigma'_{x_r} \tag{6.7}$$

where $\sigma'^2_{x_r}$ is an estimate of the variance of the input signal at the rth sampling instant.

The variance estimate is usually formed as

i) the average of the N previous decoded samples $\hat{X}_r$

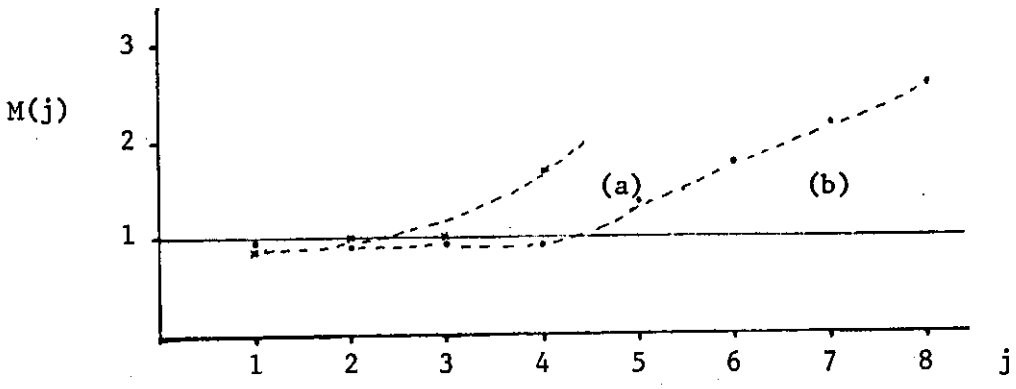


FIGURE 6.3 - The M(j) Function.
(a) 3 Bits PCM Quantizer
(b) 4 Bits DPCM Quantizer

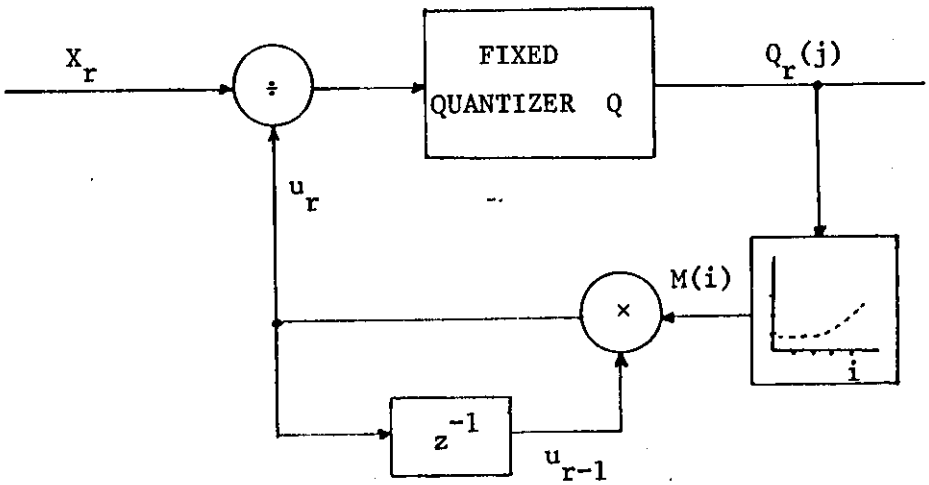


FIGURE 6.4 - Detailed Diagram of the One Word Memory Adaptive Quantizer.

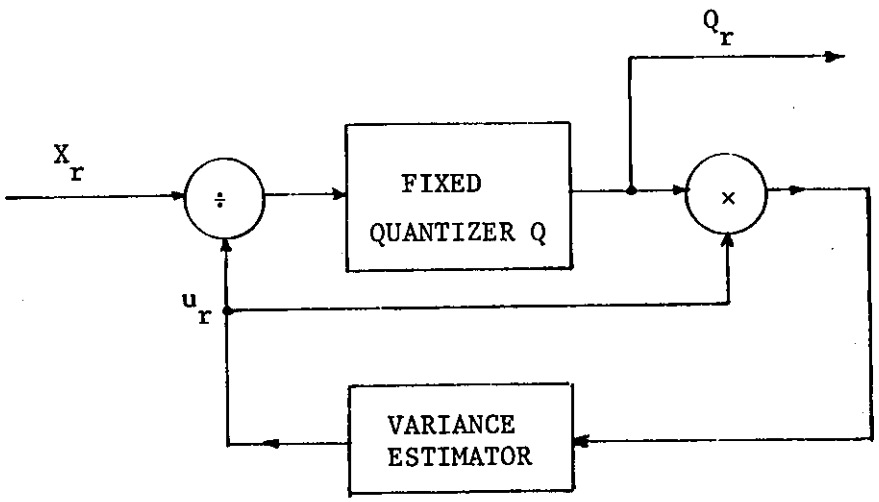


FIGURE 6.5 - The Variance Estimating Adaptive Quantizer.

$$u_r = a \frac{1}{N} \sum_{i=1}^{N} \hat{X}^2_{r-i} \tag{6.8}$$

where a is an optimizing constant.

ii) the exponential average of the previous N decoded samples

$$u_r = a \left[\sum_{i=1}^{N} (1-\gamma)\gamma^{i-1} \hat{X}^2_{r-i} \right]^{1/2} \tag{6.9}$$

where the effective memory of the variance estimator varies by changing the value of the constant γ.

It is easy to see that the VEQ adaptation algorithm is equivalent to Jayant's adaptation alogrithm. Let us consider the exponentially weighted average of Equation (6.9). It can be re-written in a recursive form as

$$u_r = \left[a^2(1-\gamma)\hat{X}^2_{r-1} + \gamma u^2_{r-1} \right]^{1/2} \tag{6.10}$$

Because now $\hat{X}_{r-1} = u_{r-1} \cdot Q_{(j)r-1}$, Equation (6.10) takes the form

$$u_r = u_{r-1} \left[a^2(1-\gamma)Q^2_{(j)r-1} + \gamma \right]^{1/2} \tag{6.11}$$

Clearly Equation (6.11) is the same with Equation (6.5) when

$$M_{(j)} = \left[a^2(1-\gamma)Q^2_{(j)r-1} + \gamma \right]^{1/2}$$

and consequently the behaviour of the Variance Estimating Quantizer is equivalent to that of Jayant's quantizer.

6.2.3. Pitch Compensating Quantizers.

Although the encoding efficiency of Jayant's quantizer is high when quantizing speech samples $\{X_k\}$ or the error samples $\{e_k\}$ in a

DPCM, its performance can be further improved. It was mentioned in Section 6.2.2. that the steady state fluctuations of u_r is proportional to

$$R = \log_2 \left[\frac{\max M_{(i)}}{\min M_{(i)}} \right]$$

and that the adaptation response of the algorithm is inversely related to R. Now in order to achieve a rapid increase in the amplitude range of the quantizer, required at the beginning of each pitch period where there is a sudden increase in the amplitude of the speech samples, the value of R must be large. This however leads to an increased amount of granular noise during the part of the waveform following the pitch pulses. Consequently an adaptive quantizer is required to adapt successfully to i) long term syllabic variations and ii) to short term pitch variations and unvoiced to voiced transitions of speech signals. Two similar quantization methods devised to meet the above requirement have been proposed and are referred to as pitch compensating quantizers.

In the first method of Cohn and Melsa[168] two u_r adaptive estimators are operating simultaneously. One is an Envelope estimator and computes u_r as a moving average of the magnitude of previous decoded $\hat{X}_k$ or $\hat{e}_k$ samples. The other is a Jayant's estimator whose $M_{(j)}$ coefficients are all smaller than unity except the coefficients which correspond to the outermost quantization levels. Note that the outermost quantization levels are set at values higher than normal and the value of $M_{(j)}$ for these levels are considerably larger than unity. For example in a 7 levels quantizer $M_{(1)} = 0.7$, $M_{(2)} = 0.8$, $M_{(3)} = 0.9$ and $M_{(4)} = 2.3$.

The value the normalizing variable assumes at each sampling instant is the largest value obtained from the two estimators. In this way when unvoiced speech is quantized,all but the outermost quantization levels are used and as a consequence the output of Jayant's estimator assumes values much smaller than those of the envelope estimator. Thus during the quantization of unvoiced speech where the average of previous $|\hat{X}_k|$ sampler is an acceptable estimate of the standard deviation of the input samples $\{X_k\}$, u_r is made equal to the output of the Envelope estimator.

When voiced sounds are quantized and in particular when pitch peaks occur in the speech waveform, the quantizer detects a possible pitch pulse with its outermost quantization levels. Because the values of $M_{(j)}$ corresponding to these quantization levels are large the output of the Jayant's estimator significantly increases to values much higher than those obtained from the Envelope estimator. Thus when a pitch pulse occurs, u_r rapidly assumes large values as required. If the outermost levels occur at instants other than those of the putch pulses, the $M_{(j)}$ coefficients allow for u_r to rapidly decay back to its long term average value.

The Equation for updating the above PCQ quantizer can therefore be written as follows:

$$u_r = \text{Max.}\left[au_{r-1}\cdot M_{(j)}, <|\hat{X}_k|> b\right] \tag{6.12}$$

where $\text{Max}[A,B]$ means the maximum value between A or B, a, b are optimizing coefficients and <•> indicates a time average.

The second Pitch Compensating Quantizer developed by Qureshi and Forney employs two Jayant's estimators, one for tracking the

syllabic variations of the input signal and another providing large values for u_r when the outermost quantization levels, set at values higher than normal, are used. Thus the quantization strategy is the same with the previous one except that the envelope estimator is substituted with a Jayant's estimator whose $M_{(j)}$ coefficients are near to unity and consequently its output follows the long-term syllabic variations of the input signal. If $U_r = \log_2 u_r$ the adaptation procedure of Qureshi's PCQ quantizer is defined as:

$$U_r = U'_r + V_r + U_{min.} \tag{6.13}$$

where U_{min} is a constant and the minimum value of U_r. U'_r is the logarithm to the base 2 of the output of the pitch compensating Jayant's estimator and is updated according to:

$$U'_r = \gamma U'_{r-1} + M_{1(j)_{r-1}} \tag{6.14}$$

$M_{1(j)}$ is a set of zero coefficients except for one which corresponds to the outermost levels of the quantizer, and $\gamma < 1$, causes U'_r to decay exponentially after the occurrence of an outermost quantization level.

Finally V_r is the $\log_2$ output of the second syllabic estimator and it is defined as:

$$V_r = \gamma_1 V_{r-1} + M_{2(j)_{r-1}} \tag{6.15}$$

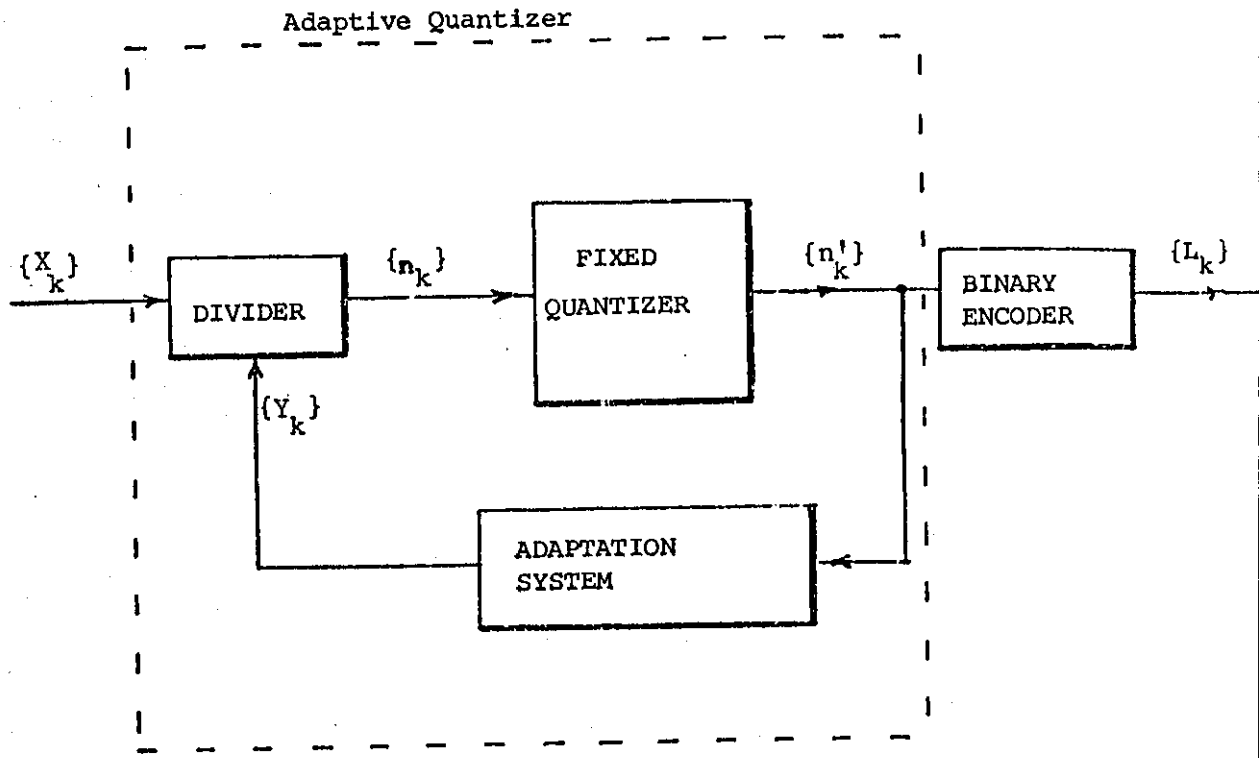
where $\gamma_1 < 1$, and the $M_{2(j)}$ coefficients are close to zero except for the outermost level which is zero.

6.2.4. A Generalized Adaptive Quantization Approach.

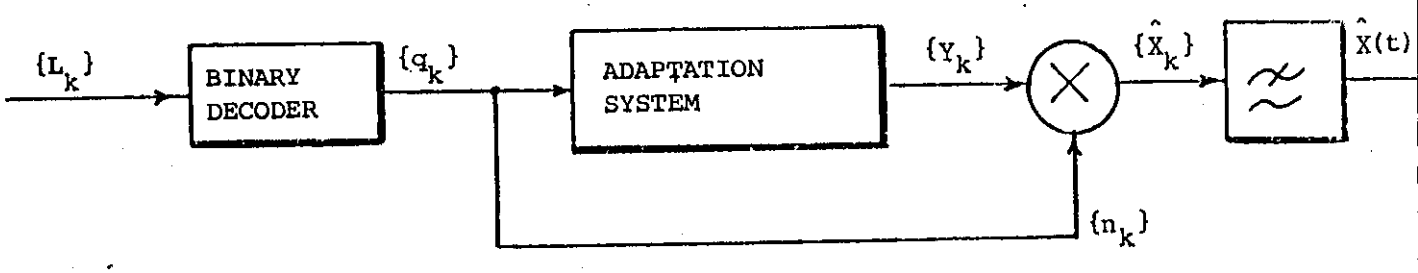
An adaptive quantizer is required to update its step size δ according to the amplitude variations in the input sequence of samples $\{X_k\}$ (or $\{e_k\}$, when the quantizer is used in DPCM). We have seen in the previous sections how δ is updated in four well known adaptive quantizers and we can now represent an adaptive quantizer, by a feedback system having a fixed quantizer in the forward path, an adaptation system in the feedback loop and a divider. The arrangement is shown in Figure 6.6. Observe that the output from the adaptive quantizer also comes from the fixed quantizer. The concept of a fixed quantizer is important because in constructing an adaptive quantizer a fixed quantizer would be used in the form of an analogue to digital converter (ADC).

The function of the adaptation system is to accept the quantized sequence $\{n'_k\}$ and produce a feedback sequence $\{Y_k\}$ which when divided into $\{X_k\}$ yields a sequence $\{n_k\}$. This normalized sequence $\{n_k\}$ is generally within the range of the fixed quantizer. In other words, no matter how great the amplitude variations of the components in the input sequence $\{X_k\}$ are, a sequence $\{Y_k\}$ is produced which ideally confines the components of $\{n_k\}$ within the range of the fixed quantizer.

When used in the PCM format of Figure 6.6, $\{n'_k\}$ is encoded into the binary sequence $\{L_k\}$ and transmitted. Assuming no transmission errors, $\{L_k\}$ is received and decoded back to $\{n'_k\}$. The receiver uses an identical adaptation system to produce $\{Y_k\}$ and by multiplying the components in $\{Y_k\}$ by those in $\{n'_k\}$ the sequence $\{\hat{X}_k\}$ is obtained. $\{X_k\}$ differs from the original sequence



(a)



(b)

FIGURE 6.6 - A Generalized Adaptive Quantizer.

$\{X_k\}$ by the effects of quantization noise.

The adaptation system attempts to produce a normalizing sequence $\{Y_k\}$ which enables the components in $\{n_k\} = \{X_k/Y_k\}$ to utilize the full range of the fixed quantizer. Further the characteristic of the input-output relationship of the quantizer can be arranged to match the pdf of $\{n_k\}$ in order to minimize the mean square quantization error. These objectives can be realized when the statistics of the input sequence are stationary. However, when the statistics of $\{X_k\}$ are non-stationary the pdf of the $\{n_k\}$ sequence varies, and the quantization noise propagates through a non-linear feedback system. Consequently an appropriate criterion is to design the adaptation system to output a sample Y_k which is a good prediction of X_k. This requires a faster adaptation time than achieved by most quantizers[40,41] which appear to be instantaneously adaptive in that they make changes at every sampling instant, but these changes in Y_k are generally slower than the maximum changes occurring in the signal. In fact the normalization of the components in $\{X_k\}$ is dependent on the envelope of $\{X_k\}$, rather than the instantaneous changes in this sequence.

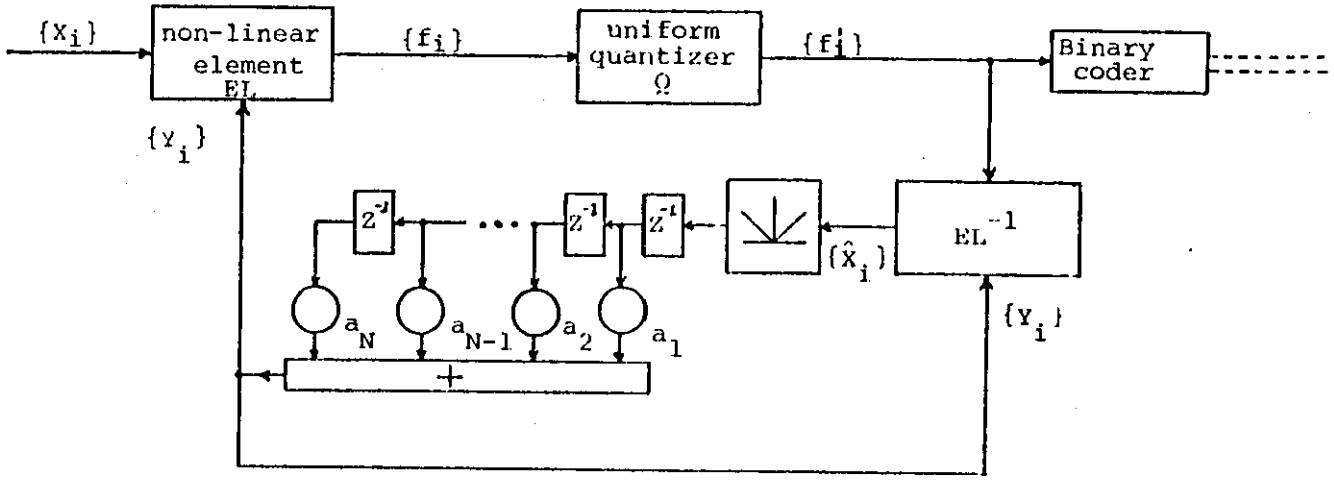
6.3 THE DYNAMIC RATIO QUANTIZER (DRQ).

The "slow adaptation" systems[40,41] tend to produce a unity variance $\{n_k\}$ sequence of samples which can then be quantized by a uniform or a non-uniform quantizer, the latter being designed to match the pdf of $\{n_k\}$. We examined the possibility of producing an adaptive quantizer which employs a much faster adaptation procedure and can reduce the variance of the sequence of samples presented in the input of the Fixed quantizer.

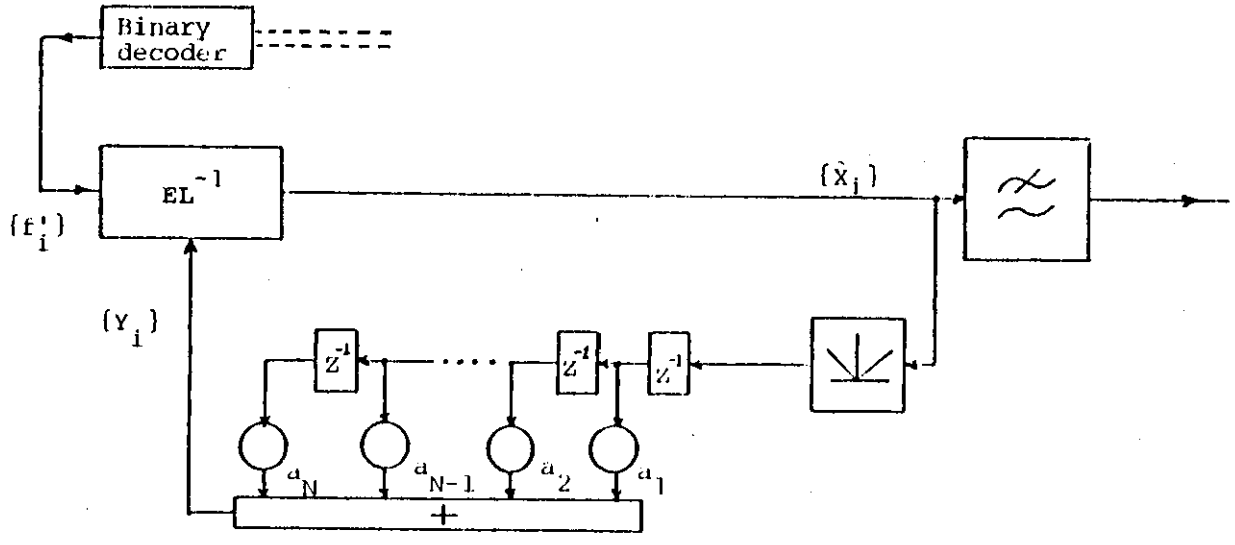
The Dynamic Ratio Quantizer[119] is a such system and uses the idea of making Y_k proportional to the prediction of X_k. If the prediction is good the components in $\{n_k\}$ are close to unity enabling the range of the quantizer to be small and thereby reducing the quantization noise. However, we may anticipate that over a long time interval $X_k > Y_k$ is as likely to occur as $X_k < Y_k$. In the former case, the ratio n_k can in principle extend from unity to a large number, while in the latter situation the ratio is confined between zero and unity. Consequently a non-linear function must be inserted between the output of the divider and the quantizer in order to restore the symmetry in $\{n_k\}$. This non-linear function should be ideally independent of the statistical properties of the input sequence $\{X_k\}$ and enable the snr of the adaptive quantizer to be substantially larger than that obtained with a fixed quantizer. We have not determined the optimum non-linear function, but the function used in the DRQ quantizer does achieve the above objectives.

6.3.1. Operation of the Dynamic Ratio Quantizer.

The block diagram of this instantaneously adaptive non-linear ratio quantizer is shown in Figure 6.7. A sample of absolute magnitude Y_k is produced from a transversal digital filter whose z-transform is H(z). This transversal filter can take various forms as described later in this section, or it can be an optimal or sub-optimal adaptive predictor whose design procedures have been discussed in the previous chapter. The feedback sample Y_k and the input sample X_k are connected to an instantaneously adaptive non-linear element EL whose output f_k is a function of the ratio X_k/Y_k.



(a)



(b)

FIGURE 6.7 - The Dynamic Ratio Quantizer.

f_k is quantized to f_k' by a uniform fixed quantizer.

f_k' is transmitted after binary encoding, and it is also locally processed by EL^{-1} with the aid of Y_k. The modulus of the decoded sample $\hat{X}_k$ is then applied to the transversal filter. The arrangement in Figure 6.7a, which accepts f_k' and produces $\hat{X}_k$ and Y_k, is called the local decoder.

The receiver accepts f_k' and processes it by the same local decoder as used at the encoder to produce $\hat{X}_k$. A low-pass filter is used to remove the out-of-bound quantization noise in the recovered sequence $\{\hat{X}_k\}$.

The non-linear element EL ensures that for widely varying input amplitudes its output is always within the amplitude range of the following uniform quantizer. As EL contains an adaptive non-linear transform TR we commence the detailed explanation of the DRQ by describing TR.

The purpose of this transform TR is

i) to restore the symmetry about unity in $\{n_k\}$,

ii) to transform input samples of any amplitude to samples whose amplitudes are defined within a certain range.

Let $\{X_i\}$ be a sequence of input samples where the current sample to be quantized is X_k. Suppose there is a sample available whose magnitude is Y_k and whose value approximates to X_k. The method of forming Y_k will be subsequently described.

We define TR, which accepts X_k and Y_k and produces f_k as follows:

$$f_k = \frac{Y_k \, \mathrm{sgn}(X_k)}{\sqrt{X_k^2 + Y_k^2}} \quad , \qquad \text{if} \quad |X_k| > Y_k \tag{6.16}$$

and

$$f_k = \frac{X_k}{\sqrt{X_k^2 + Y_k^2}} \quad , \quad \text{if} \quad |X_k| \leq Y_k \tag{6.17}$$

The transformed signal f_k tends asymptotically to zero when in Equation (6.16) $|X_k| >> Y_k$, or when in Equation (6.17), $|X_k| << Y_k$. The extremal values of f_k are $\pm 1/\sqrt{2}$ when $X_k = Y_k$. A sketch of f_k as a function of the ratio X_k/Y_k is shown in Figure 6.8.

To recover in input sample X_k from f_k, as a decoder would be required to do, the inverse adaptive non-linear transform TR^{-1} is employed, and is specified by

$$\hat{X}_k = \frac{Y_k\sqrt{1 - f_k^2}}{f_k} \quad , \quad \text{if} \quad |X_k| > Y_k \tag{6.18}$$

and

$$\hat{X}_k = \frac{Y_k f_k}{\sqrt{1 - f_k^2}} \quad , \quad \text{if} \quad |X_k| \leq Y_k \tag{6.19}$$

where $\hat{X}_k$ is the decoded value of X_k.

Having introduced TR and its inverse TR^{-1}, we now describe a monotonic instantaneously adaptive non-linear element EL, which confines any sample X_k to a fixed amplitude range, here $\pm 2/\sqrt{2}$. This element produces an output sample f_k according to:

$$f_k = \left[\frac{2}{\sqrt{2}} - \frac{Y_k}{\sqrt{X_k^2 + Y_k^2}}\right] \text{sgn}(X_k) \quad , \quad \text{if} \quad |X_k| > Y_k \tag{6.20}$$

and

$$f_k = \frac{X_k}{\sqrt{X_k^2 + Y_k^2}} \quad , \quad \text{if} \quad |X_k| \leq Y_k \tag{6.21}$$

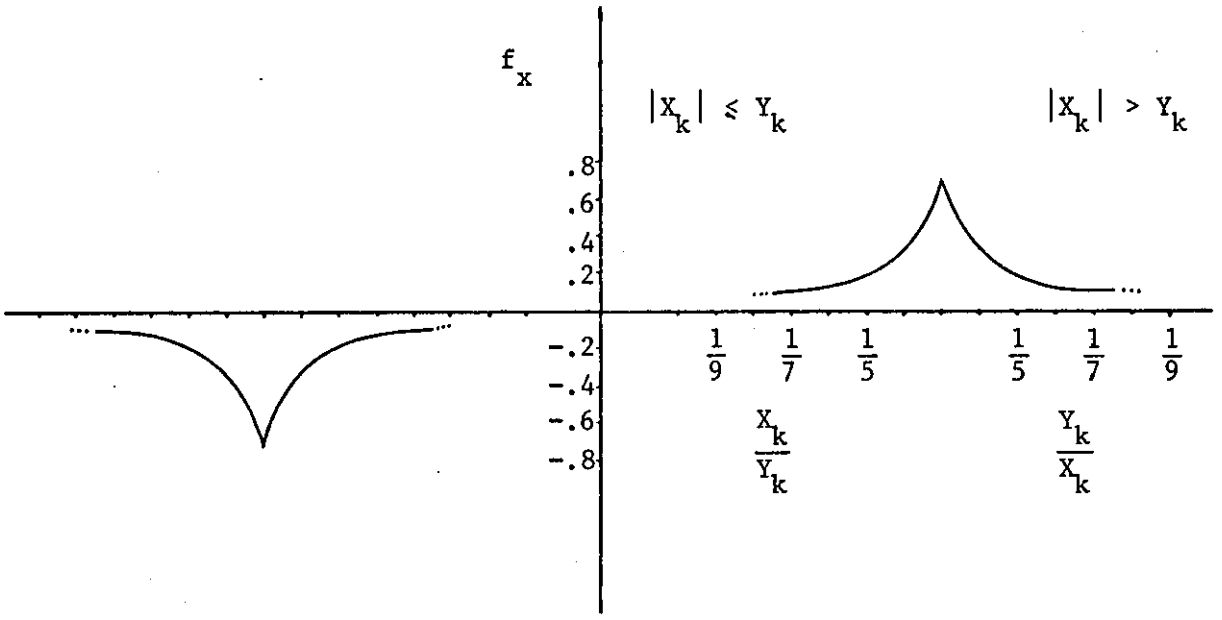


FIGURE 6.8 - The TR Characteristic.

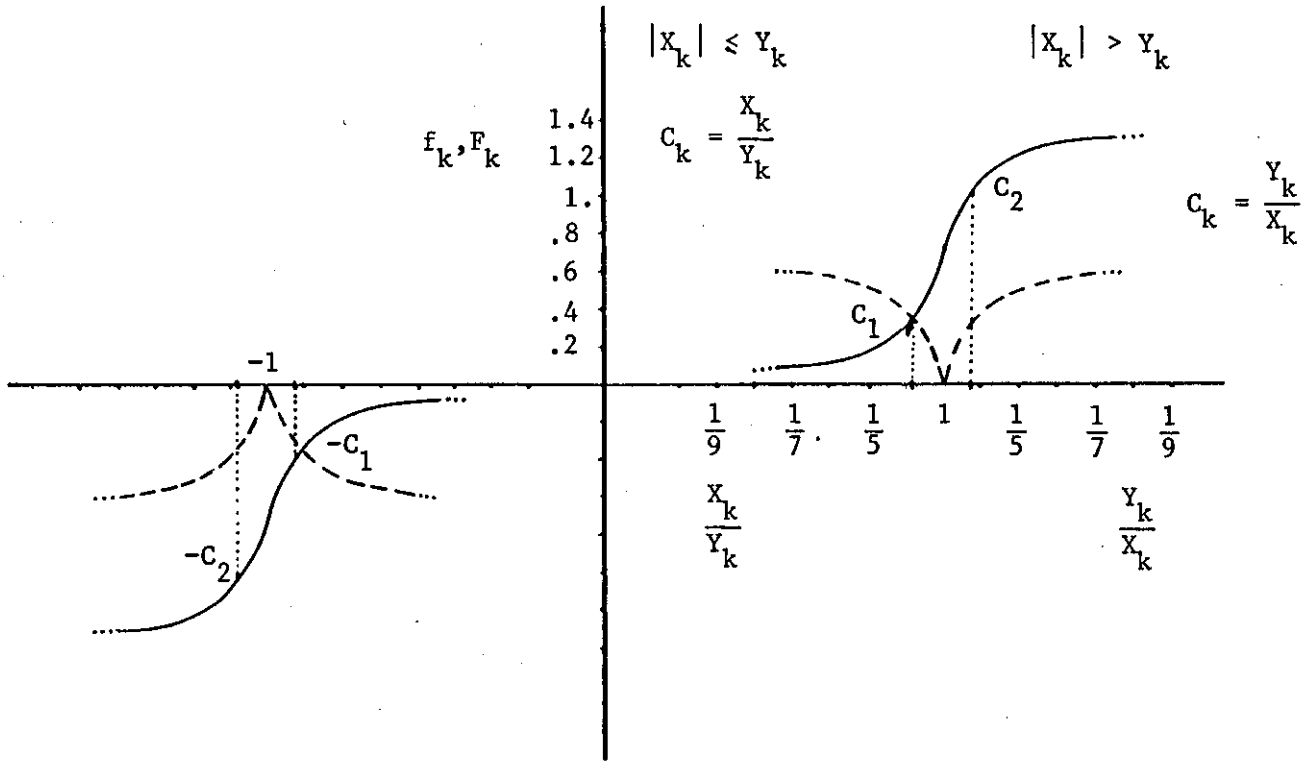


FIGURE 6.9 - The EL and MEL Characteristic.

Figure 6.9 shows f_k as a function of X_k/Y_k for EL.

After X_k is passed through the instantaneous adaptive non-linear element EL, its output sample f_k is quantized with a uniform quantizer to yield f_k' which is transmitted as a binary code word. After recovering f_k', the receiver compares it with $1/\sqrt{2}$. If $|f_k'| \leq 1/\sqrt{2}$, the encoder must have used Equation (6.21), i.e. $|X_k| \leq Y_k$. Consequently, the recovered sample $\hat{X}_k$ is produced from the following equation

$$\hat{X}_k = \frac{Y_k f_k'}{\sqrt{1 - (f_k')^2}} \tag{6.22}$$

If $|f_k'| > 1/\sqrt{2}$, then equation (6.20) must have been used at the encoder, i.e. $|X_k| > Y_k$. The decoder forms

$$f_k'' = \left[\frac{2}{\sqrt{2}} - |f_k'|\right] \operatorname{sgn}(f_k') \tag{6.23}$$

and thence performs

$$\hat{X}_k = \frac{Y_k \sqrt{1 - (f_k'')^2}}{f_k''} \tag{6.24}$$

The inverse instantaneously adaptive non-linear element EL^{-1} is represented by Equations (6.22), (6.23) and (6.24).

Suppose the ratio X_k/Y_k is confined to take values inside certain intervals $[-C_2, -C_1]$ or $[C_1, C_2]$ where the rate of change of the slope of the function shown in Figure 6.9 is relatively small. This confinement enables the quantization noise produced at the encoder to be expanded by only a small amount due to the inverse processing by EL^{-1} at the decoder.

The specific values of C_1 and C_2 will be defined later to produce the maximum snr. The segments of the curve in Figure 6.9 defined by C_1 and C_2 have their axes of symmetry at $X_k = Y_k$. Consequently in order to achieve the confinement of X_k/Y_k into these two zones, the system must maintain Y_k close to $|X_k|$. As an example, if the correlation coefficient of the input sequence $\{X_k\}$ is above 0.8 say, then a convenient choice of Y_k is the previous decoded sample $|\hat{X}_{k-1}|$.

6.3.2. Estimation of the DRQ snr.

Suppose that f_k is quantized to yield $f_k' = f_k + df_k$, where df_k is the quantization noise introduced from the uniform quantizer. f_k' is used at the decoder to produce $\hat{X}_k$ which is equal to, $\hat{X}_k = X_k + d\hat{X}_k$, where $d\hat{X}_k$ is the noise due to the use of f_k' instead of f_k in EL^{-1}.

Let us assume that $df_k << X_k$ and Y_k. In order to find the change $d\hat{X}_k$ in X_k due to the change df_k in f_k we proceed as follows:

Case when $|X_k| \leq Y_k$.

Differentiating Equation (6.19) with respect to f_k,

$$\frac{d\hat{X}_k}{df_k} = \frac{Y_k(\sqrt{1 - f_k^2} + f_k^2 / \sqrt{1 - f_k^2})}{1 - f_k^2}$$

$$= \frac{Y_k}{(1 - f_k^2)^{3/2}}$$

Substituting f_k from Equation (6.17)

$$d\hat{X}_k = \frac{(Y_k^2 + X_k^2)^{3/2}}{Y_k^2} \cdot df_k \qquad (6.25)$$

Let $C_k = X_k/Y_k$, $|X_k| \leq Y_k$.

Substituting C_k in Equation (6.25)

$$d\hat{X}_k = \frac{X_k(1 + C_k^2)^{3/2}}{C_k} \, df_k \qquad (6.26)$$

The value of signal-to-noise ratio in dBs is

$$snr = 10 \log_{10} \left[\frac{\sum_{i=1}^{N} X_i^2}{\sum_{i=1}^{N} \left[\frac{X_i (1 + C_i^2)^{3/2}}{C_i} \cdot df_i \right]^2} \right] \qquad (6.27)$$

Case when $|X_k| > Y_k$.

Proceeding as in the previous case, we differentiate $\hat{X}_k$ in Equation (6.18) with respect to f_k.

$$\frac{dX_k}{df_k} = \frac{Y_k \left[-\frac{1}{2} \cdot \frac{2f_k}{\sqrt{1 - f_k^2}} \cdot f_k - \sqrt{1 - f_k^2} \right]}{f_k^2}$$

$$= - \frac{Y_k}{f_k^2 \cdot \sqrt{1 - f_k^2}}$$

and substituting f_k from Equation (6.16), we have

$$d\hat{X}_k = \frac{-(X_k^2 + Y_k^2)^{3/2}}{X_k Y_k} \, df_k \qquad (6.27a)$$

Because of the symmetry of the characteristic shown in Figure 6.9, we define:

$$C_k = Y_k/X_k \ , \quad |X_k| > Y_k$$

Substituting C_k into Equation (6.27a),

$$d\hat{X}_k = - \frac{X_k(1 + C_k^2)^{3/2}}{C_k} \ df_k \tag{6.28}$$

Equations (6.26) and (6.28) have the same magnitudes but different signs.

The minus sign in Equation (6.28) means that when $|X_k| > Y_k$ the noise component $d\hat{X}_k$ in the recovered $\hat{X}_k$ signal is subtractive rather than additive as in Equation (6.26). In computing snr, it is the magnitude of $d\hat{X}_k$ which is important as $d\hat{X}_k$ is squared. Hence the snr for $|X_k| > Y_k$ is the same as for $|X_k| \leq Y_k$, i.e. Equation (6.27) is applicable for all $|X_k|/Y_k$ ratios.

We can now define the range $[C_1, C_2]$ about unity of the ratio variable C_k which maximize the snr of the DRQ quantizer. Note that the definition of C_k is different in the $|X_k| \leq Y_k$ and $|X_k| > Y_k$ cases.

It is seen from Equation (6.27) that the maximum snr occurs when $C_i^2/(1 + C_i^2)^3$ takes its maximum value. This is because a uniform quantizer is used resulting in $C_i^2/(1 + C_i^2)^3$ being independent of df_i^2.

Now the ratio in dBs of X_k^2 to $(d\hat{X}_k)^2$ is:

$$\begin{aligned} snr_k &= 10 \log_{10} \frac{C_k^2}{(1 + C_k^2)^3} \ \frac{1}{df_k^2} \\ &= 10 \log \frac{C_k^2}{(1 + C_k^2)^3} + 10 \log \frac{1}{df_k^2} \end{aligned} \tag{6.29}$$

The contribution of the first term in Equation (6.29) is near its maximum, if $0.4 \leq C_k \leq 1.0$. As C_k is reduced below 0.4, snr_k reduces rapidly. Consequently C_1 is set to 0.4 for $|X_k| \leq Y_k$.

For $|X_k| > Y_k$, C_k is defined as the Y_k/X_k ratio and it is again required to be ≥ 0.4. Consequently the value of C_2 is 0.4.

As the term $C_i^2/(1 + C_i^2)^3$ varies by approximately 1.5 dB for $C_i \geq 0.4$, we can to a first approximation, replace it by a constant L. As the df_i samples have zero mean and are statistically independent of X_i we can write Equation (6.27) as

$$snr \doteq - 10 \log_{10} L - 10 \log_{10} \sum_{i=1}^{N} df_i^2$$

snr is therefore independent of the input sequence $\{X_k\}$ and dependent on df_i^2, i.e. on the step size δ of the uniform quantizer. In practice the dynamic range of the input signal for constant snr will only be limited by the dynamic range of TR.

The snr given by Equation (6.27) was found to be within 0.1 dBs of the snr obtained by computer simulation of the Dynamic ratio quantizer when a Gauss Markov input source was used.

6.3.3. Modification of the Non-Linear Element EL, the Transversal Filter.

The success in reducing the quantization noise in the dynamic ratio quantizer DRQ depends on its ability to confine C_i to the intervals C_1, C_2 for most of the time. Although a uniform quantizer has been used following EL, it is better to concentrate the quantization levels in the intervals C_1, C_2, and this implies employing a non-linear fixed quantizer.

Alternatively the characteristic of EL can be adjusted to enable a fixed uniform quantizer to be employed.

This modified EL, called MEL, is required to produce an output F_k of zero rather than $\pm 1/\sqrt{2}$ when $|X_k|/Y_k$ is close to unity. TR, defined by Equations (6.16) and (6.17) and shown in Figure 6.8, is again used to yield:

$$F_k = \left| |f_k| - \frac{1}{\sqrt{2}} \right| \text{sgn } (f_k) \tag{6.30}$$

The output values F_k of the MEL are given in Table 6.1 for several C_i ratios. The MEL characteristic is illustrated by the dotted curve in Figure 6.9. Observe that if $C_i = C_1$ or C_2, MEL will produce the same F_k. This means that it is essential to inform the receiver whether $|X_k| > Y_k$ of $|X_k| \leq Y_k$. However, the range of the output signal from the MEL is half the range of the original EL. Consequently the quantizer range for the MEL is halved, and for the same quantizer step size as used with EL one less bit is required in the code word to specify the amplitude of the quantization level. However, the length of the transmitted code word is unchanged as one bit is required to inform the receiver of the status of the $|X_k|/Y_k$ ratio.

The transmitted code relating to the quantized sample F'_k, is recovered at the receiver and is used to produce the decoded value of f_k namely

$$\hat{F}_k = \left| |F'_k| - \frac{1}{\sqrt{2}} \right| \text{sgn } (F'_k) \tag{6.31}$$

By observing the status bit concerning the $|X_k|/Y_k$ ratio, the output sample $\hat{X}_k$ is recovered according to:

EL				MLE	
$\lvert X_k \rvert \leq Y_k$, $C_k = \frac{X_k}{Y_k}$		$\lvert X_k \rvert > Y_k$, $C_k = \frac{Y_k}{X_k}$		$\lvert X_k \rvert \leq Y_k$ or $\lvert X_k \rvert > Y_k$	
C_k	f_k	C_k	f_k	C_k	f_k
1	0.7071067	1	0.7071067	1	0
1/2	0.4472135	1/2	0.9669996	1/2	0.259893
1/3	0.3162277	1/3	1.0979857	1/3	0.390879
1/4	0.2428356	1/4	1.1716779	1/4	0.464571
1/5	0.1961161	1/5	1.2180974	1/5	0.510945
1/6	0.1643989	1/6	1.2498145	1/6	0.542707
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

TABLE 6.1.

$$\hat{X}_k = \frac{Y_k \sqrt{1 - (\hat{F}_k)^2}}{\hat{F}_k} \quad , \quad |X_k| > Y_k \tag{6.32}$$

$$\hat{X}_k = \frac{Y_k \hat{F}_k}{\sqrt{1 - (\hat{F}_k)^2}} \quad , \quad |X_k| \leq Y_k \tag{6.33}$$

Before presenting the computer simulation results obtained from the DRQ quantizer, the transversal filter used in the quantizer is considered. It has been shown that the output Y_k of the transversal filter is required to be an approximation of the input X_k. In fact the closer the approximation of the $\{Y_k\}$ sequence to $\{X_k\}$, the higher is the received snr. Consequently the filter acts as a predictor and the prediction techniques discussed in Chapter V can be applied. For simplicity the DRQ has been examined using the following two simple filters:

Form 1.

Y_k is equated to the weighted value of the magnitude of the previous decoded sample:

$$Y_k = W_1 \, |\hat{X}_{k-1}| \tag{6.34}$$

where W_1 is an optimizing constant.

The Z-transform H(Z) of the filter is:

$$H(Z) = W_1 \, Z^{-1} \tag{6.35}$$

Form 2.

Y_k is the average of the absolute values of the N previous decoded samples,

$$Y_k = \frac{W_2}{N} \sum_{i=1}^{N} |\hat{X}_{k-1}| \tag{6.36}$$

where W_2 is an optimizing constant.

$$H(Z) = \frac{W_2}{N} \sum_{i=0}^{N} z^{-i} \quad . \tag{6.37}$$

6.3.4. Computer Simulation Results.

The three DRQ schemes which are described in this section were simulated on a Modulo 1 computer and their performance was evaluated using a Gauss Markov process $\{X_k\}$ as their input. $\{X_k\}$ was generated by the Equation:

$$X_{k+1} = (1 - B^2)^{1/2} E_{k+1} + BX_k \tag{6.38}$$

where X_{k+1} and X_k are the input samples at the (k+1)th and Kth instants, E_{k+1} is a noise sample at the (k+1)th instant, and $B = 0.85$ is the correlation coefficient of the process. The sequence $\{E_k\}$ having a Normal distribution of unit variance truncated at 6.0 standard deviations was produced by a random number generator.

In agreement with other workers[62,68,39], we use snr as a measure of performance. The noise in DRQ was obtained by passing the difference between the input samples $\{X_k\}$ and corresponding decoded samples $\{\hat{X}_k\}$ through a low pass 8th order Butterworth recursive digital filter whose cut-off frequency was 3.4 kHz. The power of $\{X_k\}$ sequence containing 3000 samples was varied and for each level snr was computed. The three DRQ Schemes are:

Scheme 1.

The Scheme 1 DRQ uses EL, as defined in Equations (6.20) and

(6.21), and forms Y_k according to Equation (6.34). The graph of snr against input power for this Scheme is shown in Figure 6.10 curve (a). The transmitted code word has 4 bits/sample, and the range of the uniform quantizer used in the DRQ is ± 1.3147. Curve (b) shows the snr obtained from a uniform quantizer having 4 bits/sample.

Curves (a) of Figure 6.10 illustrates that the snr of the DRQ quantizer is constant over a range of input power which in practice is determined by the dynamic range of the adaptive non-linear Transform TR. However, the value of snr is approximately 3 dBs less than the peak snr of the uniform quantizer (curve b).

This 3 dB difference occurs as X_k/Y_k can on occasions have values well outside C_1 and C_2, even when the correlation coefficient of the input sequence is as high as .85. This mainly occurs when Y_k is close to zero, because even if the difference $X_k - Y_k$ is small (i.e. a high correlation coefficient), the ratio X_k/Y_k can be large. For example if $Y_k = .01$ and $X_k = 0.1$ the difference is small but the ratio is 10, and outside the C_1, C_2 ranges.

Scheme 2.

The DRQ uses MEL, and forms Y_k using Equation (6.36). As Y_k is a better approximation to $|X_k|$ compared to Y_k produced by the transverse filter arrangement of Form 1, Section 6.3.3, the C_i ratio is more frequently in the part of the MEL characteristic where its rate of change is small. The snr is therefore increased because (i) there is less noise produced by EL^{-1} and (ii) the effective range of the uniform quantizer is reduced which results in a smaller step size δ.

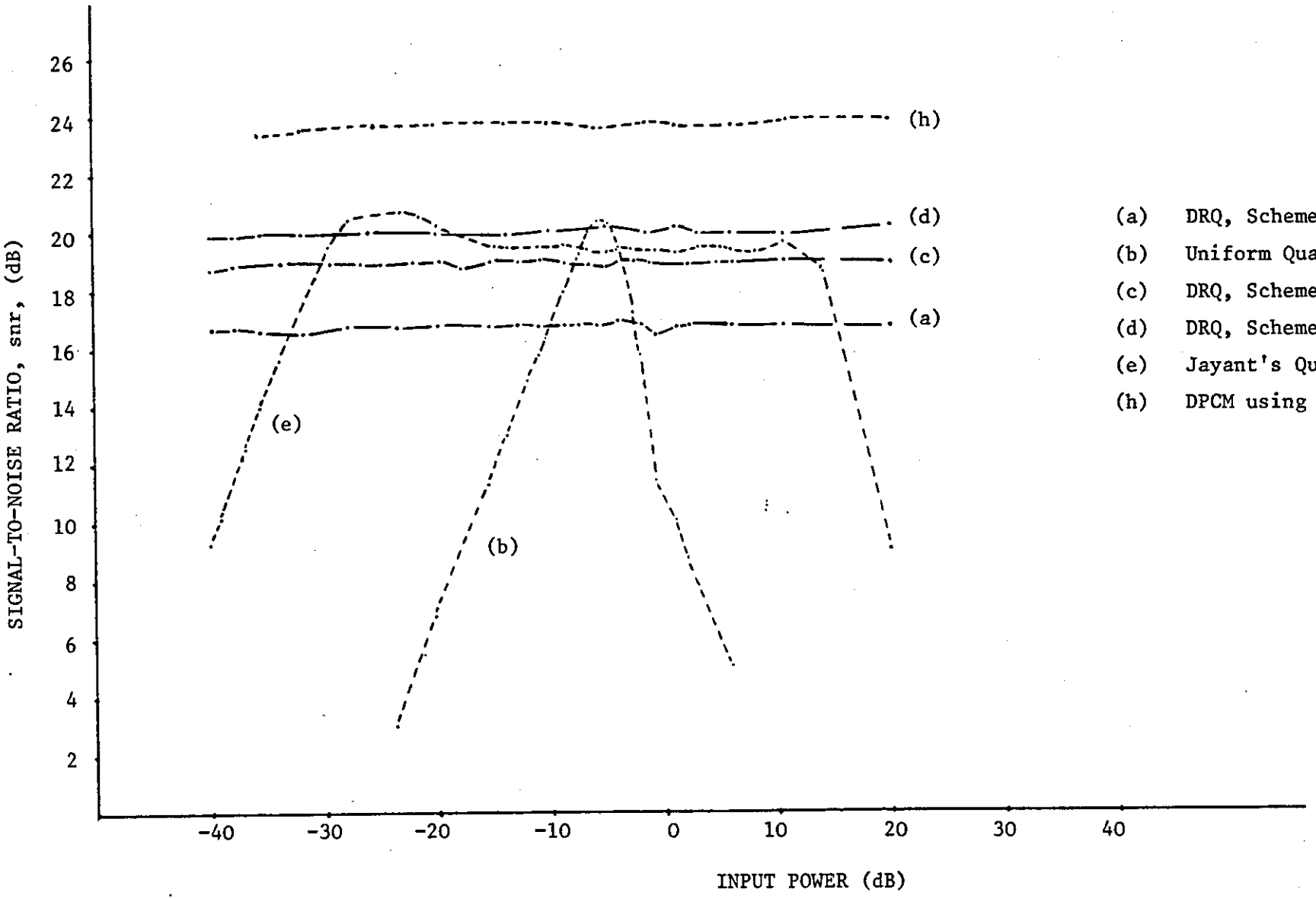


FIGURE 6.10 - snr as a Function of Input Signal Power, 4 bits/sample.

The performance of snr obtained for N = 8 in Equation (6.36), and a uniform quantizer having a range ± 0.399 is shown by curve (c); Figure 6.9. There is an improvement in snr of 2 dB over Scheme 1.

Scheme 3.

In order to increase the snr further, a positive constant S can be added to both X_k and Y_k such that the ratio $(X_k + S)/(Y_k + S)$ is closer to unity than X_k/Y_k. This ensures that the range of the quantizer can be reduced and this effect will increase the snr. However, as the DRQ has a flat snr versus input power characteristic, the quantization noise is proportional to the input power, so that by increasing X_k by S the quantization noise is also increased. These two opposing factors reduce the snr. If the value of S is such that

$$(S + X_k) > 0 \tag{6.39}$$

then the output from the MEL, namely F_k, will always be positive.

Consequently the range of the quantizer to accommodate the signal F_k is halved compared to when S = 0, but the number of levels remain the same, i.e. the spacing between adjacent levels δ is halved. This arrangement results in an improved snr compared to when S = 0, i.e. Scheme 2.

A constant value of S which satisfies Inequality (6.39) will restrict the dynamic range of the DRQ, and therefore S is made to adapt to the variance of the input samples. For the Kth instant,

$$S_k = \frac{W_3}{N} \sum_{i=1}^{N} |\hat{X}_{k-i}| \tag{6.40}$$

where W_3 is an optimizing constant such that

$$S_k + Y_k > 0 \quad . \tag{6.41}$$

The samples presented to MEL are:

$$X'_k = X_k + S_k$$

and

$$Y'_k = Y_k + S_k$$

where

$$Y_k = |X_{k-1}|$$

The transmitted word format for this Scheme is the same as in Scheme 2.

The snr for Scheme 3 is shown in Figure 6.10, curve (d), for transmitted code words having 4 bits. Curve (e) shows the snr for Jayant's one word memory adaptive quantizer for the same transmitted bit rate of 32 kBits/sec. and the same input signal. The snr's for DRQ, Scheme 3 and Jayant's quantizer are similar, but Scheme 3, like the other Schemes presented here, has a more constant snr over the same dynamic range.

We also examined the performance of a DPCM system employing a DRQ quantizer. In particular, the DRQ Scheme 2 was used in a First order DPCM encoder having an ideal integrator. In this case the sample presented to the input of the DRQ at the kth instant is the DPCM error sample $e_k = X_k - \hat{X}_{k-1}$, where $\hat{X}_{k-1}$ is the previous decoded value of the input sample X_{k-1}. The Y_k sample is formed according to Equation (6.36) with $\hat{X}_{k-1}$ being replaced by e'_{k-1}, i.e. the decoded value of e_{k-1}. Because the amplitude range of $\{e_k\}$ is smaller than that of the input $\{X_k\}$, the snr of the DPCM is increased by 5 dBs compared to the snr of DRQ Scheme 2, PCM system. This is shown in curves (h) and (c) of Figure 6.10.

6.3.5. Discussion.

The Dynamic Ratio Quantization technique presented in this section employs a different adaptation procedure than those used in the "slow adaptation" One Word Memory and Variance Estimator quantizers. The DRQ procedure enables the output sequence $\{Y_k\}$ of the adaptation system to closely follow the instantaneous variations of the components in the input speech sequence. Such an adaptation objective results:

i) In a non-symmetrical about unity sequence of ratio samples $\{X_k\} / \{Y_k\}$ (as described in section 6.3.) Therefore a Non-Linear Element is required to be inserted between the X_k/Y_k divider and the following fixed quantizer.

ii) The fact that the amplitude range of the samples to be quantized by the fixed quantizer can be reduced. Thus the step size δ of the fixed quantizer can be decreased.

How important are the above two points and especially the latter one in the performance of the DRQ? The answer has been given in the Estimation of snr, section 6.3.2, where it was shown that the snr of the DRQ except of being independent of the input power it is inversely proportional to df_i^2, i.e. to the step size of the fixed quantizer. The smaller δ (without the fixed quantizer being overloaded) the larger the snr. Therefore we developed an adaptive quantizer whose snr depends on the ability of its adaptation system to produce $\{Y_k\}$ such that the $\{F_k\}$ samples at the output of the Non-Linear Element are of minimum amplitude and well inside the C_1, C_2 range of values.

Three DRQ Schemes have been examined using as input a Gauss Markov process. In Scheme 1, Y_k is simply the weighted magnitude of the previous decoded sample $\hat{X}_{k-1}$ and yields an snr which is 3 dBs less than the peak snr, $\hat{snr}$, obtained with a uniform quantizer. The transversal filter described by Equation (6.37) has a length N = 8 in Scheme 2, which results in X_k/Y_k being confined to the C_1, C_2 range for 80% of the time. This, plus the fact that MEL is used instead of EL, increases the snr attained by Scheme 1. The snr obtained from the last Scheme equalled $\hat{snr}$. The next step in our DRQ investigations was to examine in detail the DRQ of Scheme 3 and evaluate its performance with speech as the input signal. This is described in the following section.

6.4 THE ENVELOPE - DRQ.

The DRQ Scheme 3 described in the previous section, showed a superior snr performance over the other two Schemes. Scheme 3 differs from the basic DRQ quantizer of Scheme 2 in that a positive sequence of samples $\{en_k\}$ representing the envelope of the input signal, is added to both the input and feedback sequences of samples. In this way the ratio of $\{X_k + en_k\}$ and $\{Y_k + en_k\}$ is closer to unity than the ratio of X_k and Y_k, and the long term C_k approaches closer the value of unity. Consequently the amplitude range of the fixed quantizer can be reduced and this increases the snr of the system.

We have found that by making this modification, Scheme 3 called the Envelope-DRQ provided good snr and subjective results, compared to the One Word Memory APCM, when encoding speech signals. This section describes in detail the Envelope-DRQ and presents Equations for its snr. The approach in calculating the snr is a deterministic one rather than statistical. It provides an insight to the behaviour of the Envelope-DRQ and indicates an improved performance. The computer simulation results following the snr analysis, confirms that an advantage of several dBs is obtained from the Envelope-DRQ over Jayant's APCM. In the last part of this section a simple method of hardware implementation is described.

6.4.1. Operation of the Envelope-DRQ.

The system representation of the Envelope-DRQ is shown in Figure 6.11. The feedback sequence $\{U_k\}$ is not employed to

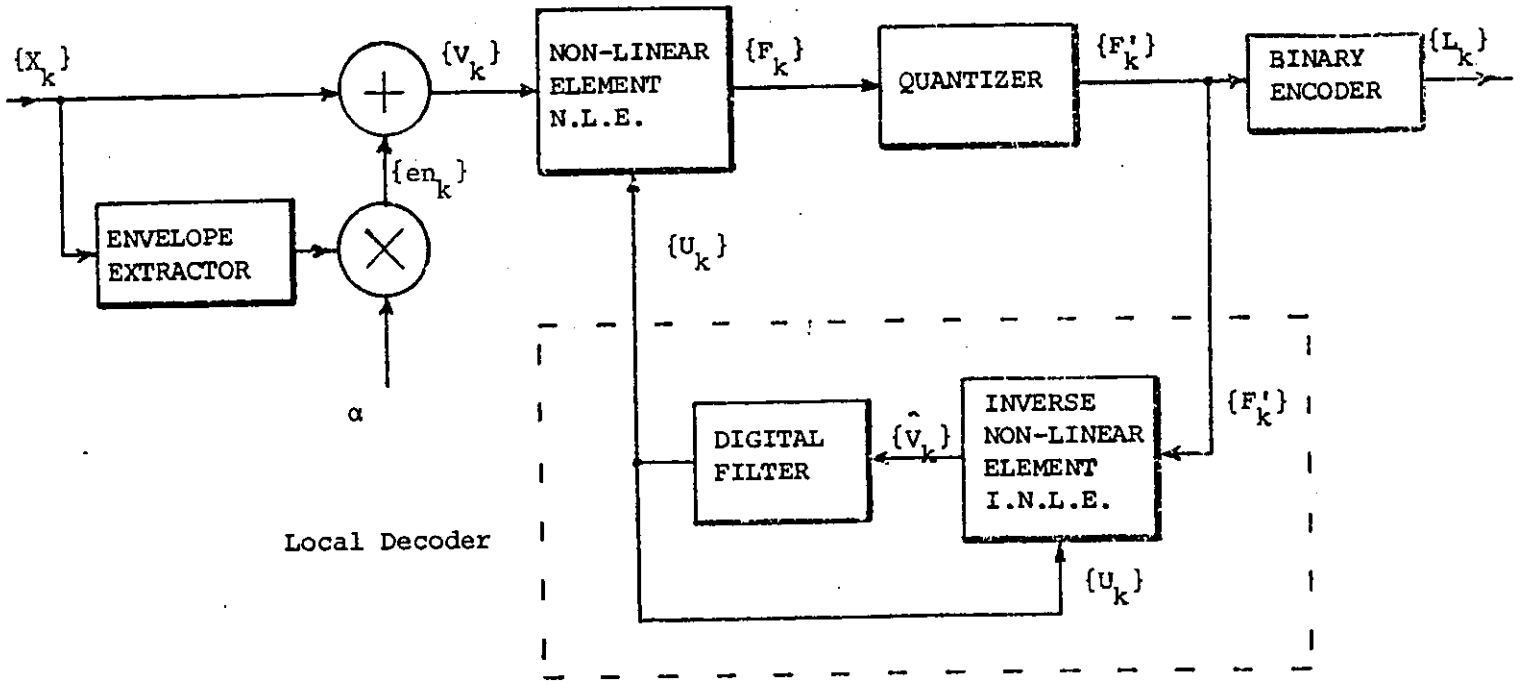


FIGURE 6.11(a) The Envelope-DRQ Encoder.

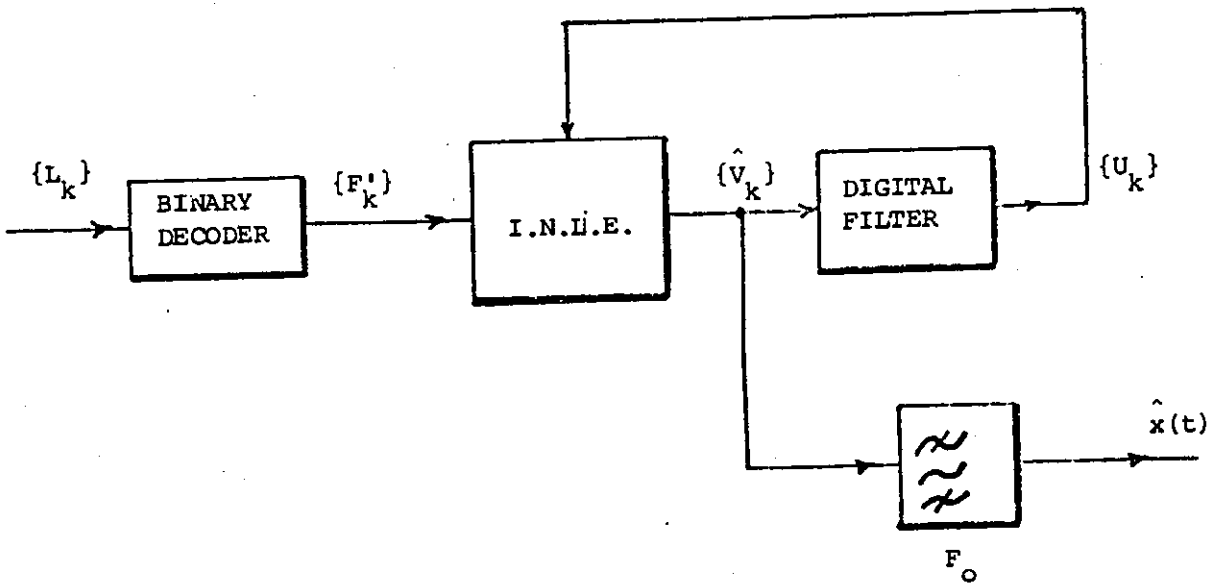


FIGURE 6.11(b) The Envelope-DRQ Decoder.

normalize the input sequence $\{X_k\}$, but a sequence $\{V_k\}$ which contains only positive components. This is achieved by adding the envelope of $\{X_k\}$, weighted by α, to itself. The reason for this addition is to maintain the normalization of $\{V_k\}$ by $\{U_k\}$ close to unity, and to ensure that the output sequence $\{F_k\}$ from the non-linear element (NLE) has only positive samples enabling the number of quantization levels in the fixed quantizer to be doubled for the same transmitted bits per code word.

The function of NLE is to match the asymmetrical sequence $\{V_k/U_k\}$ to the input-output characteristic of the uniform fixed quantizer. The sequence $\{F_k\}$ at the output of the non-linear element is quantized to $\{F'_k\}$ and transmitted after binary encoding. $\{F'_k\}$ is also locally decoded to produce the $\{\hat{V}_k\}$ sequence of samples.

The local decoder in the Envelope-DRQ is composed of an inverse non-linear element, INLE, and a digital filter. This filter acts as a predictor. It accepts the decoded sequence $\{\hat{V}_k\}$ and forms the sequence $\{U_k\}$ as a prediction of $\{V_k\}$.

Let us now consider how the NLE operates. Its detail block diagram is shown in Figure 6.12. At the rth instant, the NLE accepts an input sample V_r, consisting of the speech sample X_r and an envelope sample en_r, and a feedback normalizing sample U_r. The normalized sample V_r/U_r is applied to a non-linear function whose output f_r is

$$f_r = \frac{1}{\sqrt{\left(\frac{V_r}{U_r}\right)^2 + 1}} \quad , \quad \text{if } V_r > U_r \tag{6.42}$$

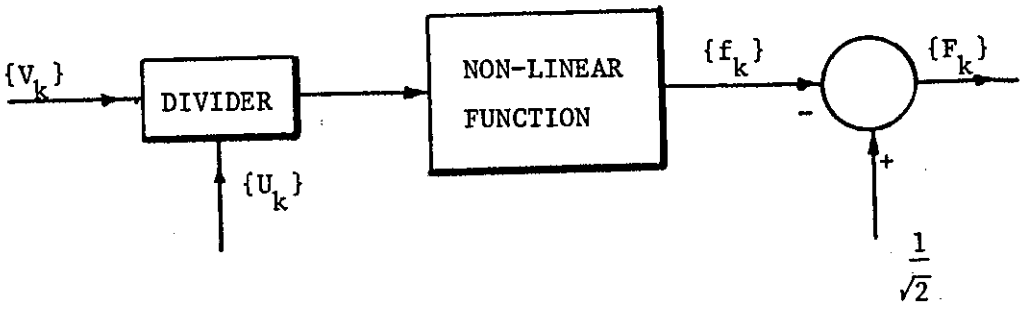


FIGURE 6.12 - The Non-Linear Element NEL.

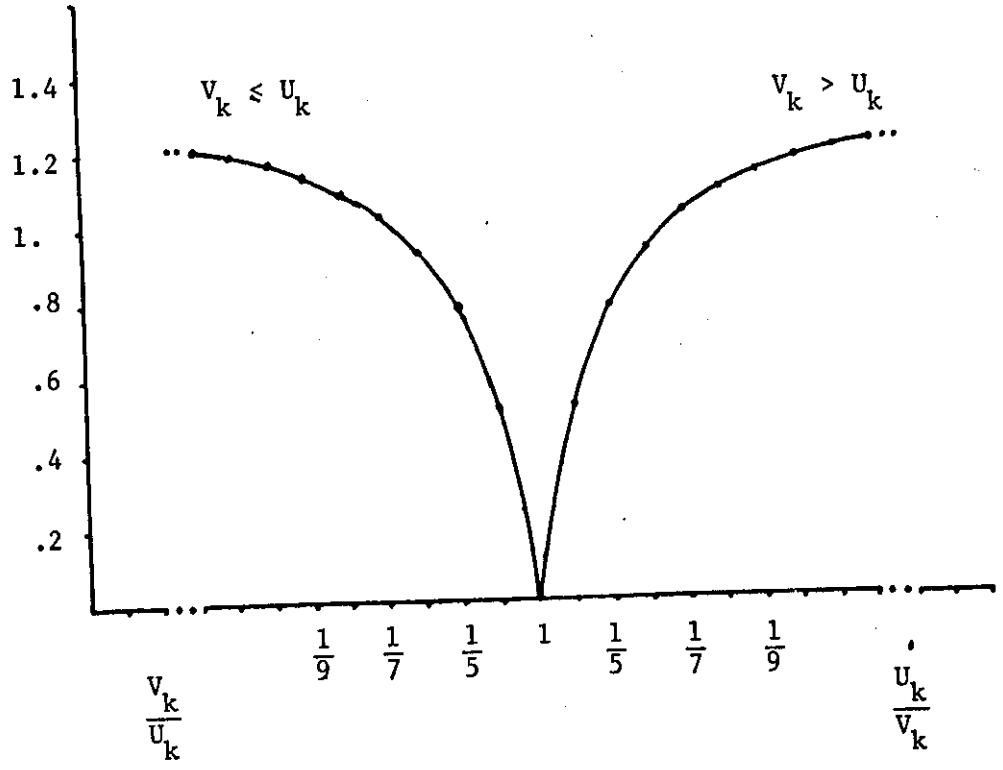


FIGURE 6.13 - Characteristic of the Non-Linear Element.

$$f_r = \frac{V_r/U_r}{\sqrt{\left(\frac{V_r}{U_r}\right)^2 + 1}} \quad , \quad \text{if} \quad V_r \leq U_r \tag{6.43}$$

The sample f_r is found and subtracted from $1/\sqrt{2}$ to give F_r at the output of the NLE

$$F_r = \frac{1}{\sqrt{2}} - f_r \tag{6.44}$$

This sample F_r has been produced so that its amplitude range is within the range of the fixed quantizer. The variation of F_r as a function of V_r/U_r is shown in Figure 6.13. Because V_r is restrained to be always positive by the presence of the envelope extractor (shown in Figure 6.11) and U_r is the prediction of V_r, only positive ratios need be considered. Observe the symmetry of F_r about $V_r/U_r = 1$. The range of the quantizer is between 0 and $1/\sqrt{2}$.

The sample F_r at the output of the NLE is quantized to F'_r, binary encoded and transmitted. F'_r is also locally decoded (see Figure 6.11). To achieve this, the same U_r used in the formulation of F_r is applied to an inverse non-linear element , INLE, together with F'_r. The INLE shown in Figure 6.14, forms f'_r from F_r according to

$$f'_r = \frac{1}{\sqrt{2}} - F'_r \tag{6.45}$$

and using a non-linear function, G_r is formed according to:

$$G_r = \frac{\sqrt{1 - (f'_r)^2}}{f'_r} \quad , \quad \text{if} \quad V_r > U_r \tag{6.46}$$

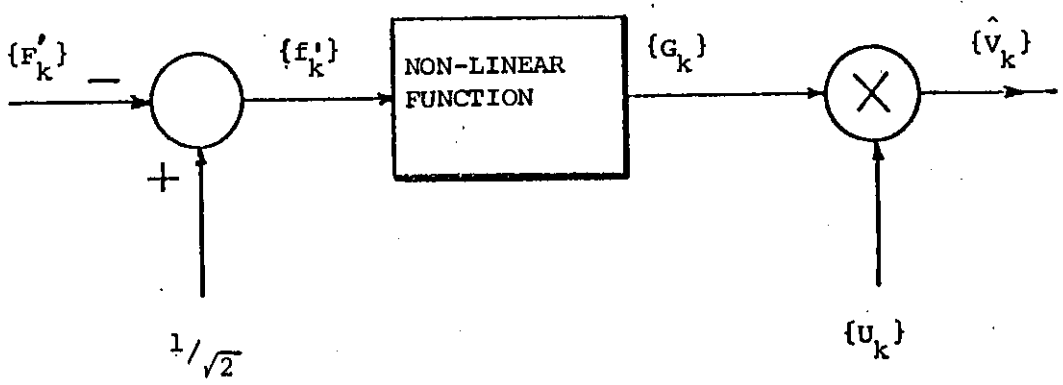


FIGURE 6.14 - The Inverse Non-Linear Element.

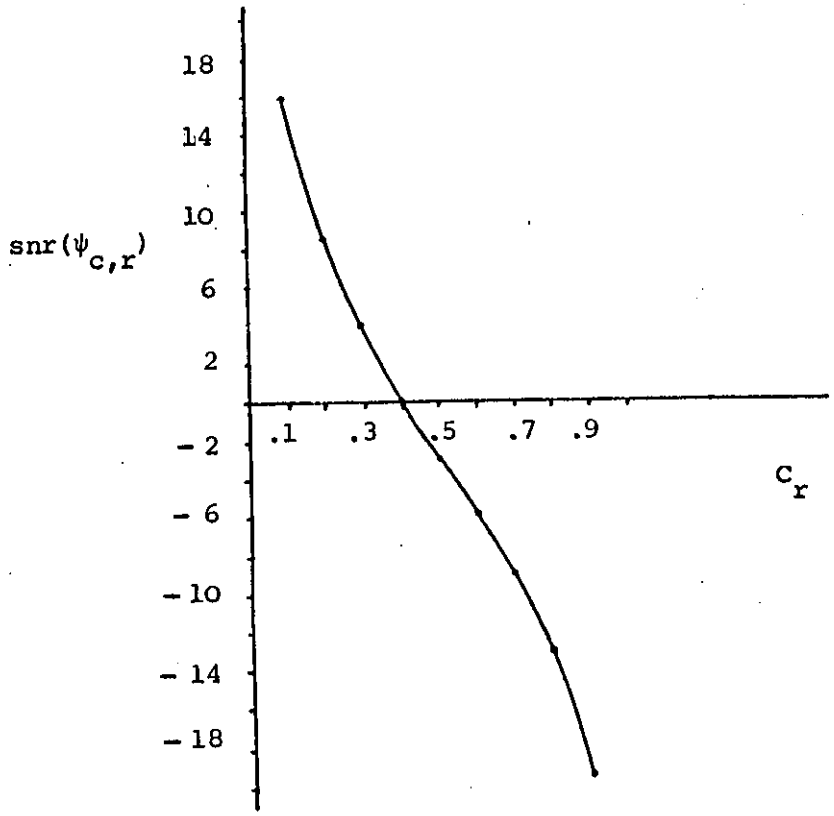


FIGURE 6.15 - snr ($\psi_{c,r}$) as a function of C_r.

$$G_r = \frac{f'_r}{\sqrt{1 - (f'_r)^2}} \quad , \quad \text{if} \quad V_r \leq U_r \tag{6.47}$$

Finally the recovered V_r, namely $\hat{V}_r$ is given by

$$\hat{V}_r = U_r \, G_r \tag{6.48}$$

At the receiver after decoding the transmitted binary sequence $\{L_k\}$ into $\{F'_k\}$ the latter is decoded into $\{\hat{V}_k\}$ by the same local decoder as used in the adaptive quantizer at the transmitter terminal. The arrangement is shown in Figure 6.11b. The sequence $\{\hat{V}_k\}$ contains the original speech sequence $\{X_k\}$, its envelope sequence $\{en_k\}$ plus a quantization noise sequence. By passing $\{\hat{V}_k\}$ through a bandpass filter F_o, the low frequency sequence $\{en_k\}$ and the high frequency out-of-band quantization noise is rejected. The original speech sequence $\{X_k\}$ together with in-band quantization noise emerges at the receiver output as X(t).

Thus by adding the envelope $\{en_k\}$ to $\{X_k\}$ the NLE element only has to accommodate positive signals yet $\{en_k\}$ is easily removed at the receiver by a simple band-pass filter.

6.4.2. Estimation of the snr for the Envelope-DRQ.

The operation of the Envelope-DRQ can be conveniently divided into two parts, (a) the extraction of the envelope information $\{en_k\}$ from the input speech sequence $\{X_k\}$ and its subsequent addition to $\{X_k\}$, and (b) the quantization of the resulting sequence $\{V_k\}$ by the Dynamic Ratio Quantizer (DRQ). We determine the signal-to-noise ratio snr_v for the DRQ, and then we consider the modification of

the input signal by its envelope to estimate the snr for the Envelope-DRQ. An error-free channel is assumed.

Estimation of snr_v.

The fixed quantizer in Figure 6.11 is the source of quantization noise in the system. We will estimate the snr_v in dBs as

$$snr_v = 10 \log_{10} \left[\frac{\sum_{i=1}^{N} V_i^2}{\sum_{i=1}^{N} dV_i^2} \right] \tag{6.49}$$

where the number of samples in $\{X_k\}$ is N, and at the rth sampling instant the locally decoded sample $\hat{V}_r$ is

$$\hat{V}_r = V_r + dV_r \tag{6.50}$$

i.e. $\hat{V}_r$ contains an error dV_r.

The fixed quantizer accepts a sample F_r, at the rth instant from the NLE and produces F'_r

$$F'_r = F_r + dF_r \tag{6.51}$$

where dF_r is the quantization error.

The value of snr is assumed to be sufficiently high for the inequal ities $V_r >> dF_r$ and $U_r >> dF_r$ to be valid. In order to determine snr_v we will find the change dV_r in V_r resulting from the change dF_r in F_r.

Case 1, $V_r > U_r$.

In estimating the quantization noise we are concerned with the magnitude of the noise component dV_r in V_r. Rearranging Equation (6.42)

$$V_r = U_r \frac{\sqrt{1 - (f_r)^2}}{f_r} \tag{6.52}$$

Substituting f_r from Equation (6.44) into the above Equation gives

$$V_r = \frac{U_r \sqrt{1 - \left(\frac{1}{\sqrt{2}} - F_r\right)^2}}{\left(\frac{1}{\sqrt{2}} - F_r\right)} \tag{6.53}$$

From which

$$\frac{dV_r}{dF_r} = \frac{U_r}{\left(\frac{1}{\sqrt{2}} - F_r\right)^2 \left\{1 - \left(\frac{1}{\sqrt{2}} - F_r\right)^2\right\}^{1/2}}$$

or

$$dV_r = \frac{U_r}{f_r^2 \left\{1 - f_r^2\right\}^{1/2}} \, dF_r \tag{6.54}$$

Eliminating f_r with the aid of Equation (6.42)

$$dV_r = \frac{\left(V_r^2 + U_r^2\right)^{3/2}}{V_r \, U_r} \, dF_r$$

$$= V_r \frac{\left(1 + C_r^2\right)^{3/2}}{C_r} \, dF_r \tag{6.55}$$

where $$C_r = \frac{U_r}{V_r} \tag{6.56}$$

Equation (6.55) provides a description of the amount of noise dV_r in the received sample $\hat{V}_r$ at the rth sampling instant due to the quantization error dF_r in the quantized sample F'_r. As expected with an adaptive quantizer, dV_r is proportional to the input sample V_r. This means that the DRQ quantizer has a constant snr for different input powers.

The ratio dV_r/V_r can also be expressed as a function of dF_r/F_r. From Equation (6.55),

$$\frac{dV_r}{V_r} = \frac{dF_r}{F_r} \frac{\left(1 + C_r^2\right)^{3/2}}{C_r} F_r$$

and from Equations (6.44) and (6.42), F_r can be found to give

$$\frac{dV_r}{V_r} = \frac{dF_r}{F_r} \left[\frac{\left(1 + C_r^2\right)^{3/2}}{\sqrt{2}\, C_r} - \left(1 + C_r^2\right) \right] \qquad (6.57)$$

This equation represents the ratio of the error dV_r in the received sample V_r to the value of V_r, as a function of the ratio of the quantization error dF_r to the value of the sample F_r applied to the fixed quantizer multiplied by a function which depends solely on C_r.

Case 2, $V_r \leq U_r$.

By proceeding in the manner above for $V_r > U_r$, and noting that $V_r = U_r$ is the axis of symmetry in Figure 6.13, we obtain

$$dV_r = - \frac{V_r \left(1 + C_r^2\right)^{3/2}}{C_r} dF_r \qquad (6.58)$$

where C_r is now defined as

$$C_r = \frac{V_r}{U_r} \tag{6.59}$$

The magnitudes of the noise components dV_r in Equations (6.55) and (6.58) are the same. Hence in terms of sample magnitudes, Equations (6.55) and (6.57) are valid for any V_r/U_r ratios.

The ratio of V_r to dV_r in dBs is

$$snr_{v,r} = 10 \log_{10} \left(\frac{V_r}{dV_r}\right)^2$$

$$= 20 \log_{10} \left[\frac{(F_r/dF_r)}{\frac{\left(1 + C_r^2\right)^{3/2}}{\sqrt{2}\, C_r} - \left(1 + C_r^2\right)} \right]$$

$$= 20 \log_{10} \left(\frac{F_r}{dF_r}\right) - 20 \log_{10} \left[\frac{\left(1 + C_r^2\right)^{3/2}}{\sqrt{2}\, C_r} - \left(1 + C_r^2\right) \right]$$

$$= 20 \log_{10} (\psi_{q,r}) - 20 \log_{10} (\psi_{c,r}) \tag{6.60}$$

The first term in this equation is the signal-to-noise ratio $snr\ (\psi_{q,r})$ of the uniform quantizer for the rth sample

$$snr_{v,r} = snr(\psi_{q,r}) - snr(\psi_{c,r}) \tag{6.61}$$

where $snr(\psi_{c,r})$ is the second term on the right hand side of Equation (6.60).

The variation of $\text{snr}(\psi_{c,r})$ as a function of C_r is shown in Figure 6.15. Observe that when $C_r = 0.4$, $\text{snr}_{v,r} = \text{snr}(\psi_{q,r})$, i.e. the $\text{snr}_{v,r}$ is the same as that achieved by fixed quantizer. By making $C_r > 0.4$ $\text{snr}(\psi_{c,r}) < 0$ and consequently $\text{snr}_{v,r} > \text{snr}(\psi_{q,r})$. The improvement in $\text{snr}_{v,r}$ is progressively enhanced as C_r approaches unity. Consequently we require C_r to be confined within the range

$$0.4 \leq C_r \leq 1.0 \tag{6.62}$$

but as close to unity as can be achieved by making U_r a good prediction of V_r.

Equation (6.60) represents the snr for the rth sample. As $\{V_k\}$ consists of N samples, the signal-to-noise ration snr_v of the Dynamic Ratio Quantizer is from Equations (6.49) and (6.57)

$$\text{snr}_v = 10 \log_{10} \left[\frac{\sum_{i=1}^{N} V_i^2}{\sum_{i=1}^{N} \left[\frac{dF_i}{F_i} \left\{ \frac{\left[1 + C_i^2\right]^{3/2}}{\sqrt{2}\, C_i} - \left(1 + C_i^2\right) \right\} V_i \right]^2} \right] \tag{6.63}$$

$$\text{snr}_v = 10 \log_{10} \left[\frac{\sum_{i=1}^{N} V_i^2}{\sum_{i=1}^{N} \left(\frac{\psi_{c,i}}{\psi_{q,i}} \right)^2 V_i^2} \right] \tag{6.64}$$

$\psi_{q,i}$ is determined by the fixed quantizer, but the snr_v can be enhanced by suitably selecting $\psi_{c,i}$, i.e. by making sure the inequalities (6.62) are satisfied.

The effect of the Envelope addition process on snr.

In order to determine the snr of the Envelope-DRQ the noise component dx_r in X_r resulting from the quantization noise dF_r in F_r will now be found. From Figure 6.11,

$$V_r = X_r + en_r \tag{6.65}$$

Substituting V_r from Equation (6.42) where $V_r > U_r$, into Equation (6.65),

$$X_r = \frac{U_r \sqrt{1 + (f_r)^2}}{f_r} - en_r$$

By substituting f_r from Equation (6.44)

$$X_r = \frac{U_r \sqrt{1 - \left(\frac{1}{\sqrt{2}} - F_r\right)^2}}{\left(\frac{1}{\sqrt{2}} - F_r\right)} - en_r$$

from which, as $den_r/dF_r = 0$,

$$\frac{dX_r}{dF_r} = \frac{U_r}{\left(\frac{1}{\sqrt{2}} - F_r\right)^2 \left\{1 - \left(\frac{1}{\sqrt{2}} - F_r\right)^2\right\}^{1/2}}$$

or

$$dX_r = \frac{U_r}{f_r^2 \left\{1 - f_r^2\right\}^{1/2}} \, dF_r \tag{6.66}$$

From Equations (6.54) and (6.66) we observe that $dV_r = dX_r$. This result is also valid if V_r from Equation (6.43), i.e. where

$V_r \lesssim U_r$, is substituted into Equation (6.65) and the above analysis repeated. Thus the addition of the envelope sample en_r at the input of the DRQ and its removal by the band-pass filter F_o at the receiver output is a catalytic process which maintains the quantization distortion dV_r unchanged ($dV_r = dX_r$), when the decoded sample $\hat{X}_r$ is obtained at the output of F_o.

By proceeding in the same manner as in Case 1, an equation similar to Equation (6.57) is obtained, namely

$$\frac{dX_r}{V_r} = \frac{dF_r}{F_r} \left[\frac{\left(1 + C_r^2\right)^{3/2}}{\sqrt{2}\, C_r} - (1 + C_r)^2 \right] \tag{6.67}$$

which gives

$$dX_r = \frac{dF_r}{F_r} \psi_{c,r} X_r + \frac{dF_r}{F_r} \psi_{c,r} en_r$$

or

$$\frac{dX_r}{X_r} = \frac{dF_r}{F_r} \psi_{c,r} \left(1 + \frac{en_r}{X_r}\right) \tag{6.68}$$

The ratio of X_r to dX_r in dBs is

$$snr_r = 10 \log_{10} \left(\frac{X_r}{dX_r}\right)^2$$

$$= 20 \log_{10} \psi_{q,r} - 20 \log_{10} \psi_{c,r} - 20 \log_{10} \left(1 + \frac{en_r}{X_r}\right) \tag{6.69}$$

$$= snr_{v,r} - snr(\psi_{e,r}) \tag{6.70}$$

where $\text{snr}(\psi_{e,r})$ is the third term on the right hand side in Equation (6.69). Equation (6.70) indicates that the snr_r of the Envelope-DRQ is equal to the DRQ signal-to-noise ratio $\text{snr}_{v,r}$ minus a quantity which depends upon the en_r/X_r ratio. When the value of en_r increases the $\text{snr}(\psi_{e,r})$ term increases, so does V_r and therefore U_r. Consequently C_r approaches unity which results in an improvement in $\text{snr}_{v,r}$. The increases in $\text{snr}_{v,r}$ and $\text{snr}(\psi_{e,r})$ oppose each other and the effect is to reduce the value of snr_r. However, the envelope sample en_r ensures that V_r is always positive and therefore the sample F_r applied to the fixed quantizer is always positive. As the polarity bit in the quantized code words is no longer required, we can use this bit to half the quantization step-size, thereby increasing $\text{snr}_{v,r}$. The result is an overall improvement in snr_r.

If $\{X_k\}$ consists of N samples, the snr for the Envelope-DRQ is from Equations (6.65) and (6.67).

$$\text{snr} = 10 \log_{10}\left[\frac{\sum_{i=1}^{N} X_i^2}{\sum_{i=1}^{N}\left\{\frac{\psi_{c,i}}{\psi_{q,i}}\,(X_i + \text{en}_i)\right\}^2}\right] \tag{6.71}$$

Observe that the snr is not inherently limited. It is as high as the accuracy of the prediction will allow. Consequently, the performance of the quantizer is dependent on the correlation of the input speech sequence $\{X_k\}$. For speech signals band-limited to 3.4 kHz and sampled at 8 kHz there is the extra correlation

produced by sampling above the Nyquist rate of 6.8 kHz. Further, as most of power in speech signals resides in its lower frequencies considerable oversampling occurs for most of the time. Indeed, speech signals can be conveniently represented by a first order Gauss Markov process having a correlation coefficient of 0.85.

6.4.3. Computer Simulation Results.

The Envelope-DRQ was simulated on the HP 2100A minicomputer based speech processing system, and its encoding performance was evaluated for speech band-limited to f_c Hz.

First, the snr performance was examined using the "An apple a day keeps the doctor away" sentence as the input signal. The snr of both the Envelope-DRQ and Jayant's quantizers (the latter used as the benchmark reference system) was calculated for sampling rates of 8 kHz and 5 kHz. The noise sequence used in the snr measurements was formed as follows. $\{\hat{V}_k\}$ was high pass filtered to remove the envelope information. The resulting sequence was then differenced with $\{X_k\}$ before low pass filtering with an 8th order Butterworth recursive digital filter having a cut-off frequency f_c Hz to give the noise signal.

The digital filter in the feedback loop of the Envelope-DRQ consisted of one sample delay followed by a multiplication by a constant coefficient P_1.

When the sampling rate was 8 kHz and f_c = 3.4 kHz, the variation of snr as a function of input power for different transmission bit rates is shown in Figure 6.16. Curves (a) and (b) were obtained from Jayant's adaptive quantizer operating with

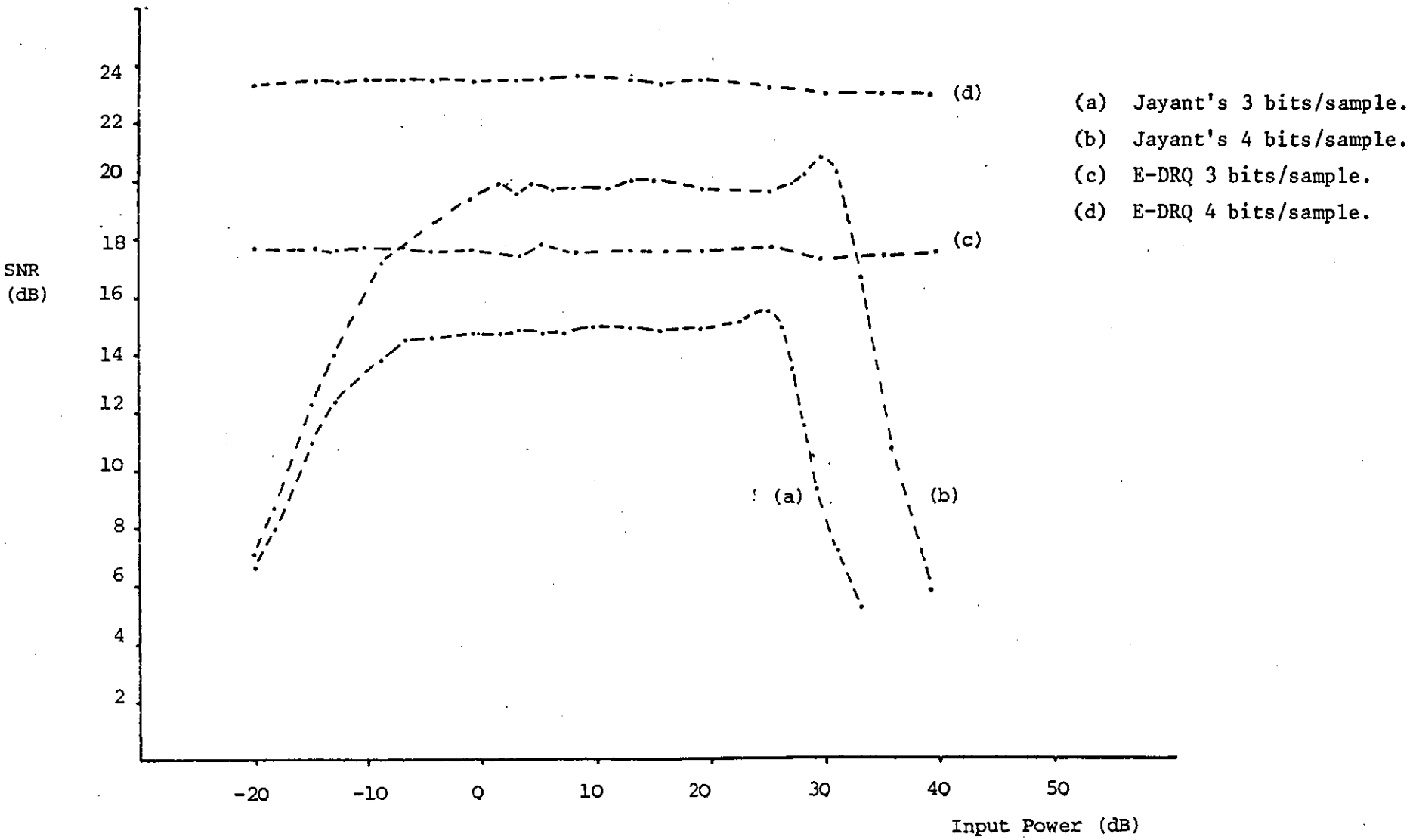


FIGURE 6.16 - snr as a Function of Input Speech Power, for f_c = 3.4 kHz, sampling rate = 8 kHz.

3 and 4 bits per sample respectively, i.e. at transmission bit rates of 24 Kbits/sec. and 32 Kbits/sec. The performance of the Envelope-DRQ using 3 and 4 bits per sample is shown by curves (c) and (d) respectively, when a = 2.7, P_1 = 1.1 and the maximum quantization level was 0.221.

The improvement in snr obtained by using the Envelope-DRQ is generally 2.5 dB for a transmission rate of 24 Kb/s, increasing to 3.5 dB when the transmission rate is 32 Kb/s. The improvement in snr is greater for the larger bit rate because the quantization noise is less and the components in $\{U_k\}$ tend to be a better approximation to those in $\{V_k\}$.

When the speech was band-limited to f_c = 2.2 kHz and sampled at 5 kHz to produce $\{X_k\}$, the performance of the Envelope-DRQ and Jayant's adaptive quantizer for 3 and 4 bits per sample, i.e. a transmission rate of 15 and 20 Kb/s is shown in Figure 6.17. In this case a = 2.7, P_1 = 1.2 and the maximum quantization level = 0.259. The improvement in the snr obtained by the Envelope-DRQ is reduced compared to the results shown in Figure 6.16. The reason for this decrease in the improvement in the snr is that when the sampling rate is reduced the correlation in $\{X_k\}$ decreases. As a consequence there are more occasions when V_k/U_k differs substantially from unity. This defect can be reduced by improving the prediction of $\{U_k\}$ from $\{\hat{V}_k\}$, i.e. by modifying the digital filter.

In order to find the maximum snr which the Envelope-DRQ could offer, the simple one delay fixed coefficient digital filter was substituted by an 8th order adaptive linear predictor. The

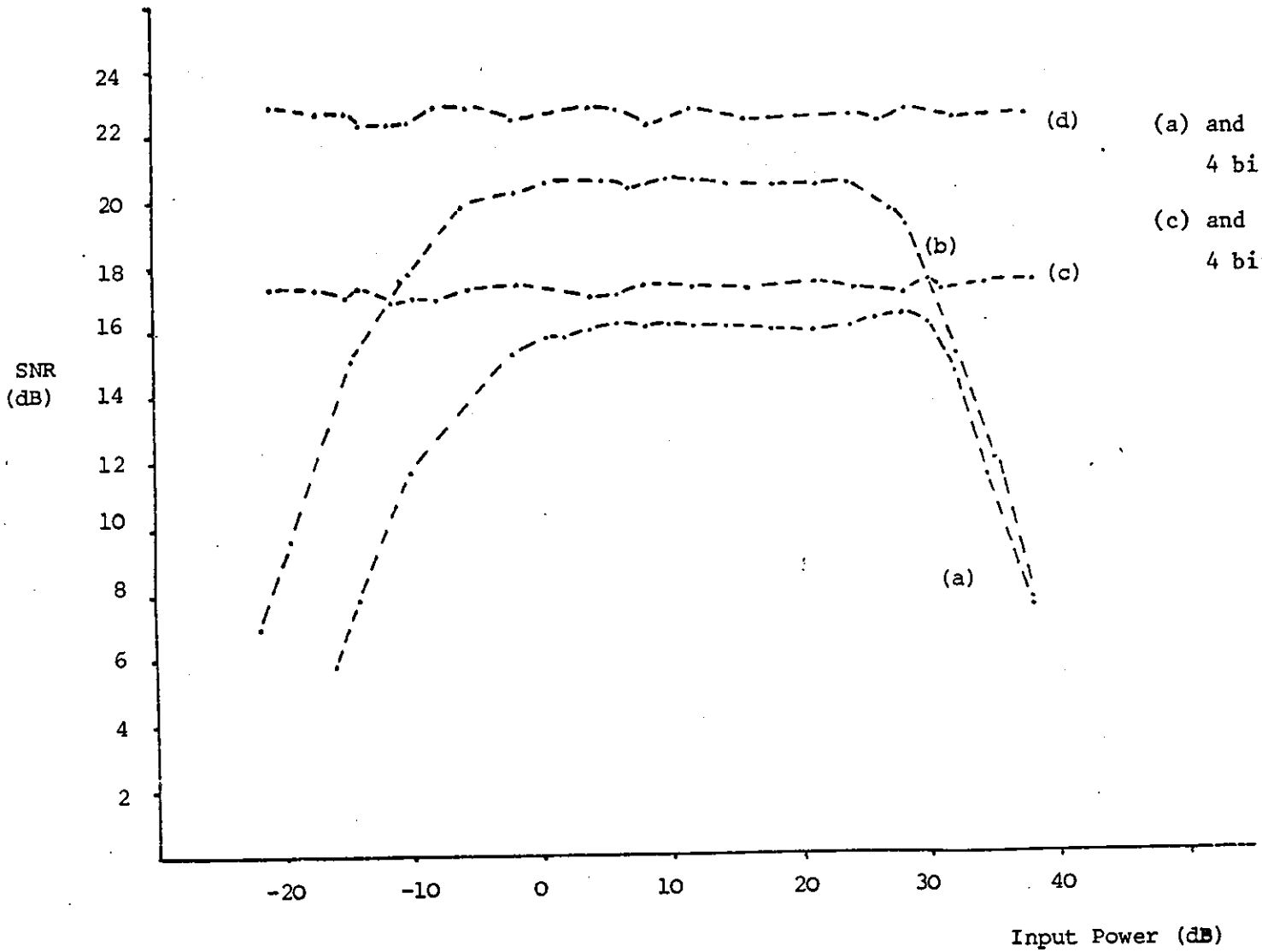


FIGURE 6.17 - snr as a Function of Input Speech Power, for f_c = 2.2 kHz. sampling rate = 5 kHz.

prediction coefficients were updated using the "Forward block adaptation" procedure where the optimum a_i coefficients are obtained after measuring the short term autocorrelation function of blocks of input speech samples $\{X_k\}$. The input samples to the adaptive predictor were the decoded speech samples $\{\hat{X}_k\}$, and $\{U_k\}$ was formed after adding the envelope samples $\{en_k\}$ to the samples at the output of the predictor. The process of encoding and transmitting the a_i coefficients was not considered and consequently it was assumed that the decoder at the receiver knew the values of the prediction coefficients. Using this optimum predictor and making $\{U_k\}$ approach $\{V_k\}$, the range of the fixed quantizer was reduced resulting to an advantage of approximately 4 dBs in addition to snr values shown in Figure 6.16. For example at transmission bit rates of 32 Kbits/sample a constant snr of 28 dB was measured.

Finally we briefly mention another set of experiments where the fixed quantizer in the Envelope-DRQ was substituted by Jayant's adaptive quantizer whose $M_{(j)}$ coefficients had values very close to unity. It was observed that by optimizing the $M_{(j)}$ set of values, a further 1 dB advantage was obtained compared to Envelope-DRQ snr values shown in Figures 6.16 and 6.17.

The subjective performance of the Envelope-DRQ and its reference quantizer was evaluated using an RSRE(C), Christchurch voiced tape which provided standard sentences spoken by a male. These sentences were stored on a HP 7970E digital magnetic tape unit and processed by the adaptive quantizers. We found that the few dB improvements in the snr shown in Figure 6.16 corresponded with our informal listening experiences.

6.4.4. Implementation of the Envelope-DRQ.

The characteristic of the non-linear element NLE together with the quantizer decision thresholds, quantization levels and corresponding binary codes is shown in Figure 6.18 for an 8-level quantizer, 4-bit code words. The quantized output level F'_f for a given C_r at the rth sampling instant is

$$C(-1) < C_r \leq C(1) \quad , \quad F'_f = F'(1)$$

$$C(j) < C_r \leq C(j+1) \quad , \quad F'_f = F'(j+1)$$

$$j = 1,2,\ldots,7$$

and

$$C(m-1) < C_r \leq C(m) \quad , \quad F'_r = F'(|m|+1)$$

$$m = -1, -2,\ldots,-7$$

Although it might appear that the weakness of the Envelope-DRQ is the difficulty in implementing the NLE, this is not so. For a given number of quantization levels the C decision levels are fixed and are easily determined.

Utilizing the symmetrical nature of the characteristic shown in Figure 6.18, the NLE and the fixed quantizer can be produced as shown in Figure 6.19. Here two dividers are used. If $V_r > U_r$, $C_r > 1$, switch S1 is forced to position A, and V_r/U_r is compared in 8 comparators whose thresholds are marked on Figures 6.18 and 6.19. The selection matrix inspects the comparator outputs and observes which one has the high output level. If this belongs to the nth comparator, the quantized output is F'(n), and a binary code word L_r is generated whose magnitude bits represent the binary value of F'(n) and whose

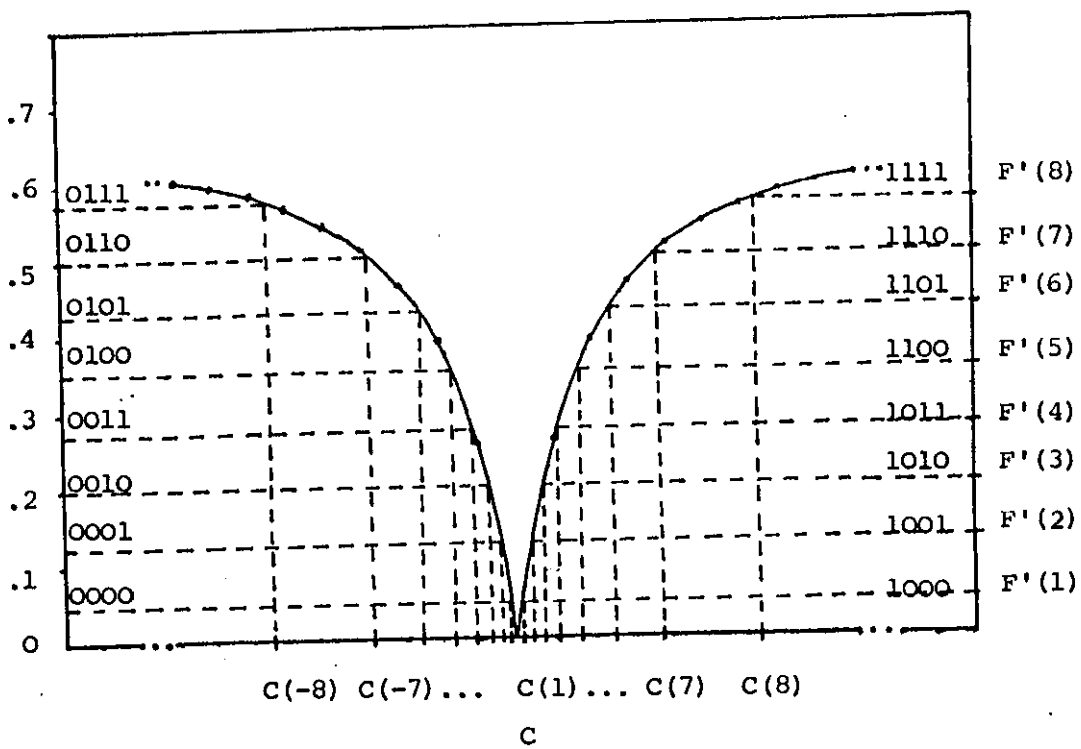


FIGURE 6.18 - Characteristic of NEL showing Decision Thresholds, Quantization Levels and Binary Codes. The Quantization Levels are normally concentrated near C of Unity, rather than as shown.

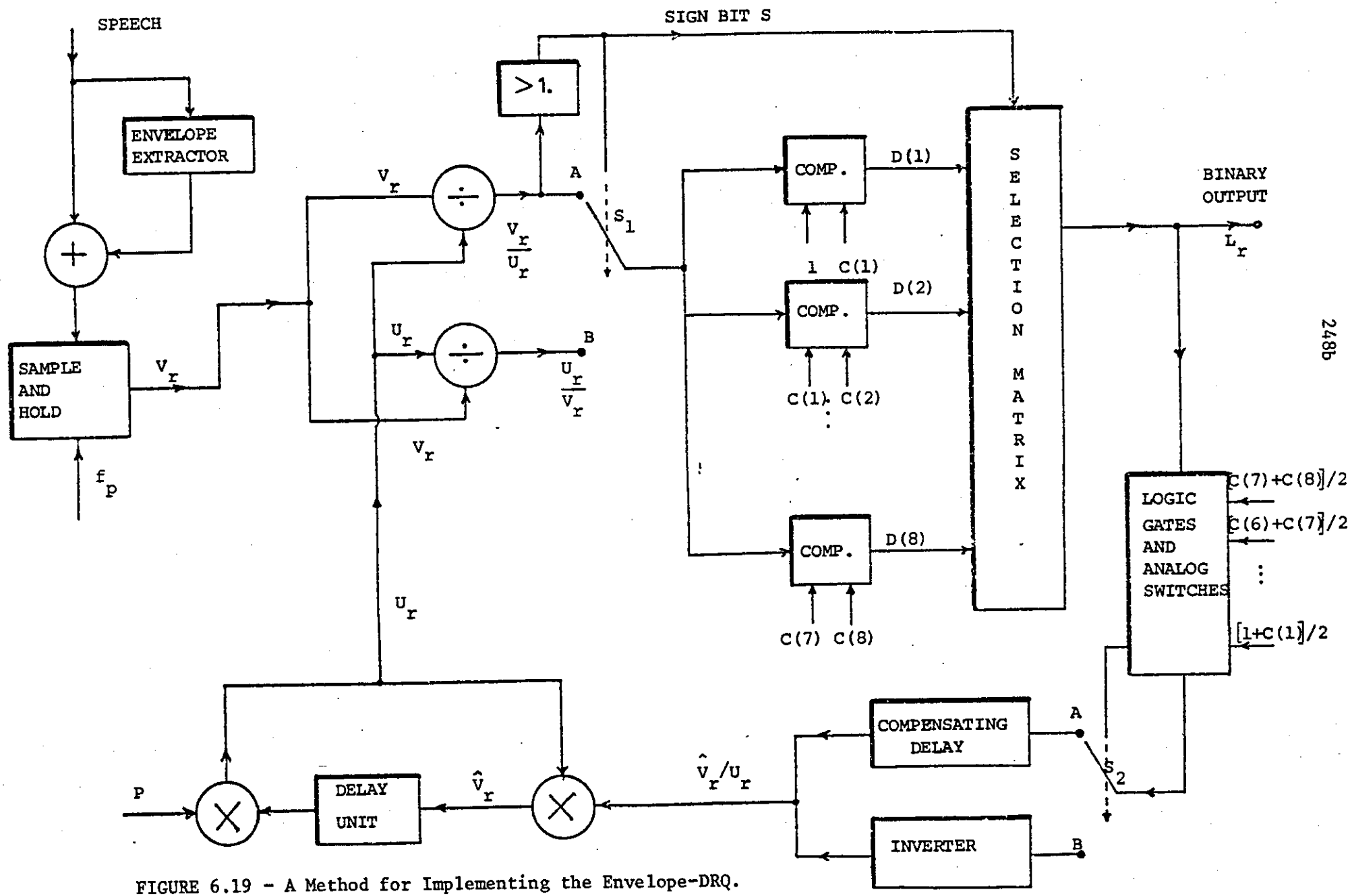


FIGURE 6.19 - A Method for Implementing the Envelope-DRQ.

polarity bit, the m.s.b., is one if $V_r/U_r > 1$ or zero if $V_r/U_r \leq 1$. The binary code words for each quantization level are displayed in Figure 6.18. The code word L_r is both transmitted and applied locally to the inverse non-linear element INLE residing in the feedback loop.

The first-stage of the INLE inspects the four bits in the code word. As $V_r > U_r$, the polarity bit is a one which sets switch S2 to position A. The magnitude bits of the code word are used to close analogue switches connected to voltages whose values are mid-way between the decision thresholds. For example, if L_r = 1011, the logic gates connect (C(3) + C(4))/2 to switch S2. This value $\hat{V}_r/U_r = \hat{C}_r$, passes through a delay equal to the delay in the inverter connected to position B, switch S2. $\hat{C}_r$ is then multiplied by U_r to yield the decoded sample $\hat{V}_r$. The value of the U_{r+1} is predicted as

$$U_{r+1} = P_1\hat{V}_r$$

where P_1 is a coefficient.

Thus the predictor used here is the simplest, although a more complex version can be employed.

If $V_r \leq U_r$, $C_r \leq 1.0$. This means that the signal at position B in switch S1 is greater than the signal at position A. The sign bit is consequently zero and is used to switch S1 to position B. Because of the symmetry of Figure 6.18, the same decision thresholds can be used in the comparators and L_r is generated as previously described. For example, if L_r = 0011, a signal $U_r/\hat{V}_r$ having a magnitude of (C(3) + C(4))/2 is produced. To obtain $\hat{V}_r/U_r$, the signal at output of the logic gates and analogue switches section

must be inverted. This is accomplished by the polarity bit of zero activating switch S2 to change to position B. Having produced $\hat{V}_r/U_r$, $\hat{V}_r$ and U_r are derived as for the previous case of $V_r > U_r$.

The decoder consists of the same INLE and the simple digital filter used in Figure 6.19. The sequence $\{\hat{V}_k\}$ at the output of the multiplier is filtered by a band-pass filter to give the recovered speech signal $\hat{X}(t)$.

The Envelope-DRQ can be easily time shared with other speech channels. The input to the dividers is a time division multiplex p.a.m. signal, produced by sampling each of the speech channels in succession and at a rate above the Nyquist. For each speech channel the sample at the output of the multiplier in Figure 6.19 must be stored until this speech channel is again connected to the input of the Envelope-DRQ. In this way the only modification to the adaptive quantizer in order to extend its handling to N channels is to increase the number of sample and hold circuits by N times.

6.5 NOTE ON PUBLICATION.

A paper entitled "Dynamic Ratio Quantizer" in co-authorship with Dr. R. Steele, has been published in the Proceedings of I.E.E. Vol. 125, No.1 January 1978. The paper is a version of the DRQ described in section 6.3.

CHAPTER VII

RECAPITULATION

7.1 INTRODUCTION.

In this thesis a number of novel digitization techniques for speech signals have been proposed and investigated. The motivation for the work was the design of an efficient speech digitization system having:

i) either a large bit rate compression characteristic with the recovered speech having acceptable quality, or an improved encoding performance for a particular bit rate,

ii) a modest implementation complexity and therefore low cost.

There are two alternative approaches in the design of a speech digitizer, namely vocoding and waveform encoding methods. Our investigations were focused on waveform encoding techniques operating at "medium" transmission bit rates, i.e. between 16 Kbits and 32 Kbits per second.

It was soon realized that Differential encoding techniques offered a promising approach for the design of a waveform encoder satisfying the above two objectives, and consequently became the subject of our investigations. Differential Pulse Code Modulation is the form from which other Differential systems, like Delta Modulation can be derived and therefore it absorbed most of our attention.

First we searched for methods that, by introducing as little added complexity as possible in a DPCM encoder, could improve its

performance at a given transmission bit rate. Thus we considered the possibility of combining Delayed encoding with DPCM. Since multipath search Delayed encoding can increase considerably the complexity of a DPCM encoder, we considered simplified Delayed encoding algorithms. Two such algorithms were developed and the performance of the resulted Delayed DPCM encoder was evaluated and compared to that of conventional DPCM. The new systems showed, unfortunately, an insignificant improvement over DPCM. Our investigations were then directed towards the elements of the basic DPCM structure and in particular the predictor and the quantizer because, by increasing the estimation accuracy of the predictor and/or the encoding accuracy of the quantizer the performance of the DPCM system can be improved.

A "prediction system" composed of two separate predictors, housed in the feedback path of a DPCM encoder was examined. One operated on a pitch synchroneous basis and exploited the correlation between successive pitch periods of voiced speech, while the other made use of the correlation between successive speech samples. As a result two Pitch Synchron ous Differential Encoders were developed which showed large improvements in encoding performance, and a modest increase in complexity when compared to DPCM.

Next the quantization process was considered and an adaptive quantization technique conceived and evaluated when encoding First Order Gauss Markov sequences, or when included in a DPCM system where it encoded the difference samples formed by the subtraction between the input samples and their predicted values. When encoding speech signals, a significant improvement in performance was observed

compared to a well known adaptive quantizer.

In the following sections of this concluding chapter the characteristics of the new speech digitization techniques are summarized. Suggestions for further research are also made in a number of topics which may be of interest to workers in the area of waveform speech encoding.

7.2 SIMPLE DELAYED ENCODING TECHNIQUES APPLIED TO DPCM.

In Chapter IV we considered the application of Delayed Encoding to DPCM codecs. Since multipath-search delayed algorithms when applied to encoders with multilevel quantizers are complex and impractical, we developed two simple "single" decision look-ahead delayed algorithms. Both of them, and especially the second one, resulted in a minimum increase in the complexity of the DPCM encoder.

A DPCM encoder has its peak snr when its fixed quantizer is overloaded during the sharp impulsive spikes of the error waveform presented to its input. The delayed algorithms can detect the overload condition and modify the prediction samples in the feedback loop of the DPCM encoder so as to reduce the amplitude of the error samples and therefore the overload noise.

In the first Delayed DPCM system of Scheme 1 the feedback samples are modified by adding to them samples proportional to the overload noise of the quantizer. Operating at transmission bit rates of 24 and 32 Kbits/sec., the Scheme 1 encoder showed a peak snr advantage of approximately 1 dB over that of a conventional First Order DPCM encoder. When the input speech signal caused

severe overload in the DPCM, the Delayed DPCM encoder increased its snr advantage to 2 dBs.

The second Delayed Encoder of Scheme 2 was developed in an attempt to further simplify the Delayed Encoder of Scheme 1. Its operation is based on the same concept and it uses a multiplicative coefficient to modify the prediction samples in the feedback loop of the DPCM encoder. The Scheme 2 system showed a peak snr gain of only 0.4 dBs compared to conventional First Order DPCM. However it has companding properties and maintained its peak snr for values of input speech power where the snr of the Scheme 1 encoder was decreasing due to overload noise. This constant snr region is not extended as in the case of an ADPCM encoder, which offers a much wider Dynamic range.

Looking back to the Delayed DPCM encoding section of our research program, we feel that "single" decision look-ahead Delayed algorithms for encoding speech signals can offer only a small improvement to a DPCM encoder compared to that obtained from a multipath-search Delayed algorithm. This limitation is mainly due to the following reasons:

i) The algorithms in Schemes 1 and 2 operate and improve the encoding accuracy of DPCM only when overload is detected.

ii) Although the algorithms decrease the overall quantization noise, i.e. granular plus overload noise, they increase the granular noise for a few samples before slope overload occurs.

We feel that future work on Delayed DPCM encoding should be focused on simplified, and thus practical, multipath-search algorithms.

7.3 PREDICTION TECHNIQUES APPLIED TO DPCM.

In Chapter V we first examined by means of computer simulations the performance of Block adaptation, Time-invariant and Stochastic approximation Linear predictors. The performance of DPCM encoders employing the above three predictors was also evaluated and several research alternatives were given which it was thought could result in an improved DPCM predictor. From the latter suggested approach we developed two Pitch Synchronous Differential systems, while of the remaining alternatives we expect that investigations leading to the use of a sequentially adaptive Lattice predictor in DPCM, can be potentially rewarding. This is because such a predictor when employing an efficient sequentially adaptive algorithm can follow more rapidly than an adaptive Linear predictor the instantaneous variations in the speech signal. This project is closely related to that of improving the Stochastic approximation adaptation algorithm as the same algorithm can be used to update the coefficients of a Lattice predictor.

The reason our investigations were focused on Pitch Synchronous processing of speech signal is that such a carefully designed encoder can produce an error signal with a minimal amplitude range, almost free of sharp impulsive excitation spikes, which can be quantized with minimum quantization distortion. This is accomplished using two different types of prediction in the DPCM encoder. One is a conventional Linear predictor which removes the redundancy between successive speech samples. The other removes the redundancy due to waveform similarities between adjacent pitch periods. It is the second predictor which actually eliminates the excitation pulses from the error signal.

First we developed the Pitch Synchronous First Order DPCM (PSFOD) system where a sequence of difference samples between adjacent pitch periods is initially formed. The goals achieved by this pitch based differential procedure are:

i) the variance of the resulted difference sequence is considerably reduced compared to that of the input voiced speech signal,

ii) the sequence is free of excitation spikes which upset the performance of conventional waveform encoders.

Then the difference samples are DPCM encoded. Consequently the variance of the error sequence of samples presented to the quantizer in the DPCM is very small and this results to comparatively small quantization distortion. During unvoiced speech, the input samples are directly DPCM encoded.

Three important points are worth mentioning:

a) The difference between adjacent pitch periods has to be formed through a feedback closed loop system otherwise there is an accumulation of quantization distortion.

b) The difference in duration between successive pitch periods has to be taken into consideration when the sequence of difference of samples is formed. This is because straightforward subtraction results into a sequence having large amplitude spikes.

c) The receiver to recover the speech signal requires the prior knowledge of voice/unvoiced transitions, and also the duration of the pitch periods in voiced speech.

All the above points have been considered during the design

of the PSFOD system so that:

i) it is a closed loop feedback system,

ii) the amplitude range of the sequence of difference samples between adjacent pitch periods is minimum, and

iii) the system includes a synchronizing procedure so that the voiced/unvoiced, and pitch information is available to the receiver.

Three PSFOD systems were simulated in the computer, the PSFOD-LI, PSFOD-AI, and PSFOD-AF. The PSFOD-LI codec using a DPCM encoder with a fixed quantizer and an Ideal integrator in the feedback loop, showed a snr advantage over conventional DPCM of approximately 6 dBs for 3 and 4 bits per sample quantization. The PSFOD-AI used ADPCM, having Jayant's adaptive quantizer and an ideal integrator, to encode the difference sequence of samples. Computer simulation results indicated an approximate 8 dB advantage compared to the case where the same input speech signal was encoded with the above ADPCM. Finally the PSFOD-AF system employed ADPCM having Jayant's quantizer and a fixed coefficient linear predictor. An additional 1 dB advantage was obtained over the snr of the PSFOD-AI system, when the fixed predictor used one coefficient. Experiments involving amplitude prediction in the outer pitch loop of the PSFOD system showed no snr improvement.

The other important point established during the PSFOD experiments was that although its performance depends upon the pitch duration measurements of the Pitch Sequence Extractor, there is no need to specify the pitch period with the accuracy required in Analysis-Synthesis systems. By pitch we mean the similarities in the voiced waveform measured between major waveform peaks. If the Pitch Sequence

Extractor selects another maximum peak than the one which corresponds to the pitch period, the performance of the PSFOD codec is marginally effected.

The second system developed, called Pitch Synchronous Differential Predictive Encoding System, (PSDPE), is an improvement of PSFOD. To increase further the prediction accuracy of the linear predictor operating on successive input speech samples, the process of forming the difference sequences had to be reversed. That is, we formed first the difference samples between the input samples and their estimates at the output of the linear predictor and then a sequence of error samples is formed by subtracting difference samples corresponding to adjacent pitch periods.

Again as with the PSFOD, the PSDPE system was designed to be a closed loop feedback system employing:

i) an algorithm to minimize the amplitude range of the error signal, and

ii) a synchronizing procedure to convey the voiced/unvoiced and pitch duration information to the decoder at the receiving end.

We found that the basic PSDPE system, i.e. PSDPE-AI using an adaptive quantizer to encode the error samples, showed an overall snr advantage of approximately 0.5 dB over the snr of the PSFOD-AI digitizer. When the PSDPE system used a First Order predictor to form the difference samples, its snr was further improved by 2 dBs, for 3 and 4 bits per sample quantization accuracy. This is because the First Order predictor operated on the more correlated input speech samples instead of the difference samples as in the PSFOD case, and hence its prediction accuracy was increased.

A topic for further investigations on Pitch Synchroneous Differentially encoding systems is the use of sequentially adaptive prediction in the place of fixed prediction. We feel that an adaptive predictor should improve the performance of the PSDPE system operating at bit rates below 24 Kbits/sec. This is because as shown in section 5.2.2. the performance of the Stochastic Approximation predictor used in DPCM is considerably higher than that of a Fixed predictor when the error samples are quantized with less than 3 bits per sample accuracy.

Another item worthy of further research is the use of Pitch Synchroneous systems at transmission bit rates of 4.8 and 9.6 Kbits/sec. In such a case the input speech signal will be band limited to 2.2 kHz and sampled at 4.4 kHz. Using a two quantization levels PSDPE or PSFOD system the transmission bit rate including the synchronizing information will be of 4.8 Kbits/sec. A 9.6 Kbits/sec. codec will employ a 2 bits per sample quantization accuracy. What has to be determined for this low-bit rate application, is the quantization strategy and prediction procedure which result to the best possible subjective performance. We expect that a 9.6 Kbits/sec. PSDPE codec will provide good speech quality at relatively low implementation cost. We also expect that a 4.8 Kbits/sec. PSDPE codec will transmit intelligible speech.

7.4 ADAPTIVE QUANTIZATION TECHNIQUES.

In Chapter VI we examined adaptive quantization strategies which could be applied to Differentially encoding systems. First we considered some well known adaptive quantizers and discussed their

weaknesses. As a consequence we presented a generalized model of an adaptive quantizer which takes the form of a closed loop feedback system including a divider, a fixed quantizer, and an adaptation system. The importance of the adaptation system in the overall performance of the quantizer was emphasized and new design objectives for the adaptation system were formulated. Thus the DRQ quantization technique together with a deterministic mathematical analysis of its snr behaviour, was developed.

The basic DRQ concept is that the output samples of the adaptation system are the predicted values of the incoming input samples to the quantizer. The better the input samples are predicted, the smaller the amplitude range of the fixed quantizer and therefore the higher the obtained snr. The above adaptation system required a Non-Linear Element before the fixed quantizer.

Three DRQ schemes were examined for different types of Non-Linear Elements and predictors, and their performance was compared to Jayant's quantizer when encoding a First Order Markov process. All Schemes produced a constant snr independent of the power of the input signal and limited only by the Dynamic range of the quantizer's, Non-Linear Element. Using simple First Order Fixed prediction, snr results were obtained competitive to those of the One Word Memory quantizer.

Since the last DRQ scheme showed the best performance compared to the other two, it was examined in detail when encoding speech signals and its snr behaviour was mathematically analysed. The system called the Envelope-DRQ showed a 2.5 and 3.5 dBs advantage over Jayant's quantizer at transmission bit rates of 24 and 32 Kbits/sec. respectively. At transmission bit rates of 15 and 10 Kbits per second, the snr

advantage was reduced to 2 and 1 dBs respectively. We found that the few dBs advantage in snr corresponded with our informal listening experiences.

It was also observed that when operating at the transmission bit rate of 15 Kbits/sec., the subjective performance of the Envelope-DRQ was competitive with an ADM encoder.

Further investigations should examine the compatability of the DRQ quantizer with Pitch Synchroneous Differential systems. Also the use of adaptive prediction in the adaptation system of the quantizer has to be carefully investigated. We feel that such a predictor will significantly improve the performance of the DRQ. Finally, as the effects of transmission errors on the quantizer's performance have been omitted, a topic of further research is to evaluate the DRQ performance in the presence of transmission errors and perhaps introduce a simple "leaky integration" effect on the adaptation system to combat these errors.

7.5 CLOSING REMARKS.

This thesis describes investigations to conceive and evaluate new encoding techniques for accommodating speech signals based on preserving the integrity of the waveform, rather than using the more complex frequency domain encoding strategies. More specifically, we focused our research in two main areas: differential encoding systems which exploit the quasi-periodicity of voiced speech, and instantaneously adaptive quantizers.

Although several questions related to our work have yet to be answered, we would like to believe that the developed systems would find applications in the speech digitization area, possibly in a modified form, and that our efforts laid the foundation for more fruitful research in the near future.

APPENDIX

Numerous programs have been developed during the course of the work described in this thesis. We concentrate however, only on the main aspects of some representative programs due to

i) space limitations,

ii) most of the programs are straightforward interpretation of the algorithms and calculations given in the thesis. The interested reader should be able to originate the appropriate program with the help of the flow charts of the programming procedures described in Chapters IV and V.

A Low-Pass Filter.

The pre and post-encoding band-limiting operation was performed using Recursive Butterworth low-pass digital filters.[120]

The gain characteristic of a Nth order Butterworth filter is given by

$$|H(e^{j2\pi fT})| = 1/\left[1 + (\tan\pi fT/\tan\pi f_c T)^{2N}\right]^{1/2} \qquad \text{(A1)}$$

where f_c is the cut-off frequency and T the sampling period. The higher the value of N the better is the approximation of the filter's gain characteristic to an ideal low-pass characteristic.

A Nth order filter has N poles which lie on a circle in the z plane. Their co-ordinates are given by:

$$\left.\begin{aligned} U_m &= (1 - \tan\pi f_c T)/d \\ V_m &= 2\tan\pi f_c T \sin(m\pi/N)d \end{aligned}\right\} \qquad \text{(A2)}$$

where $d = 1 - 2\tan\pi f_c T \cos(m\pi/N) + \tan^2\pi f_c T$

$m = 0,1,\ldots,2N-1$ and if N is even $m\pi/N$ should be replaced by $\pi(2m+1)/2N$.

The poles are in complex conjugate pairs and for each pair Z_M, a second order recursive filter can be formed whose transfer function is:

$$H_M(Z) = \frac{1 + 2Z^{-1} + Z^{-2}}{1 - (Z_M + Z_M^*)Z^{-1} + Z_M Z_M^* Z^{-2}} \qquad \text{(A3)}$$

The transfer function of the N order filter is equal to the product of all $H_M(Z)$, $M = 1,2,\ldots$.

As an example we consider the design of a low pass Butterworth filter whose specifications are:

Clock frequency (f_s) = 8 kHz.

Cut off frequency (f_c) = 3.4 kHz.

Gain at zero frequency = 0 dB.

The order of the filter can be found using Equation (A1), i.e.

$$-28 \text{ dB} = 10 \log_{10} X$$

$$X = 1/630$$

then

$$630 = 1 + \left[\frac{\tan \frac{\pi \times 3.6}{8}}{\tan \frac{\pi \times 3.4}{8}} \right]^{2N}$$

$$629 = (1.515797629)^{2N}$$

$$2N = \frac{\log_{10} 629}{\log_{10} (1.51579)}$$

$$= 15.49$$

$$N = 7.74 \simeq 8$$

Using Equations (A2) four pairs of poles, within the unit circle, are obtained and their co-ordinates are

$$Z_B = -0.8195 \pm j0.409$$

$$Z_C = -0.7115 \pm j0.300$$

$$Z_D = -0.6470 \pm j0.1824$$

$$Z_E = -0.6165 \pm j0.0610$$

Figure A1 shows the location of the poles in the Z plane. The transfer function of the second order recursive filters is then formed according to Equation (A3). For example, the first pair of poles Z_B gives:

$$Z_B + Z_B^* = -1.6390$$

$$Z_B \times Z_B^* = 0.83828$$

hence

$$H_1(Z) = \frac{1 + 2Z^{-1} + Z^{-2}}{1 + 1.6390\ Z^{-1} + 0.83828\ Z^{-2}}$$

and its implementation is shown in Figure A2. Proceeding in the same way for the remaining three pairs of poles, the 8th order digital filter is formed as shown in Figure A3.

The listing of the subroutine for this particular filter is presented in List 1, where XOO is the input sample, YYY3 the output sample, and A is a real array.

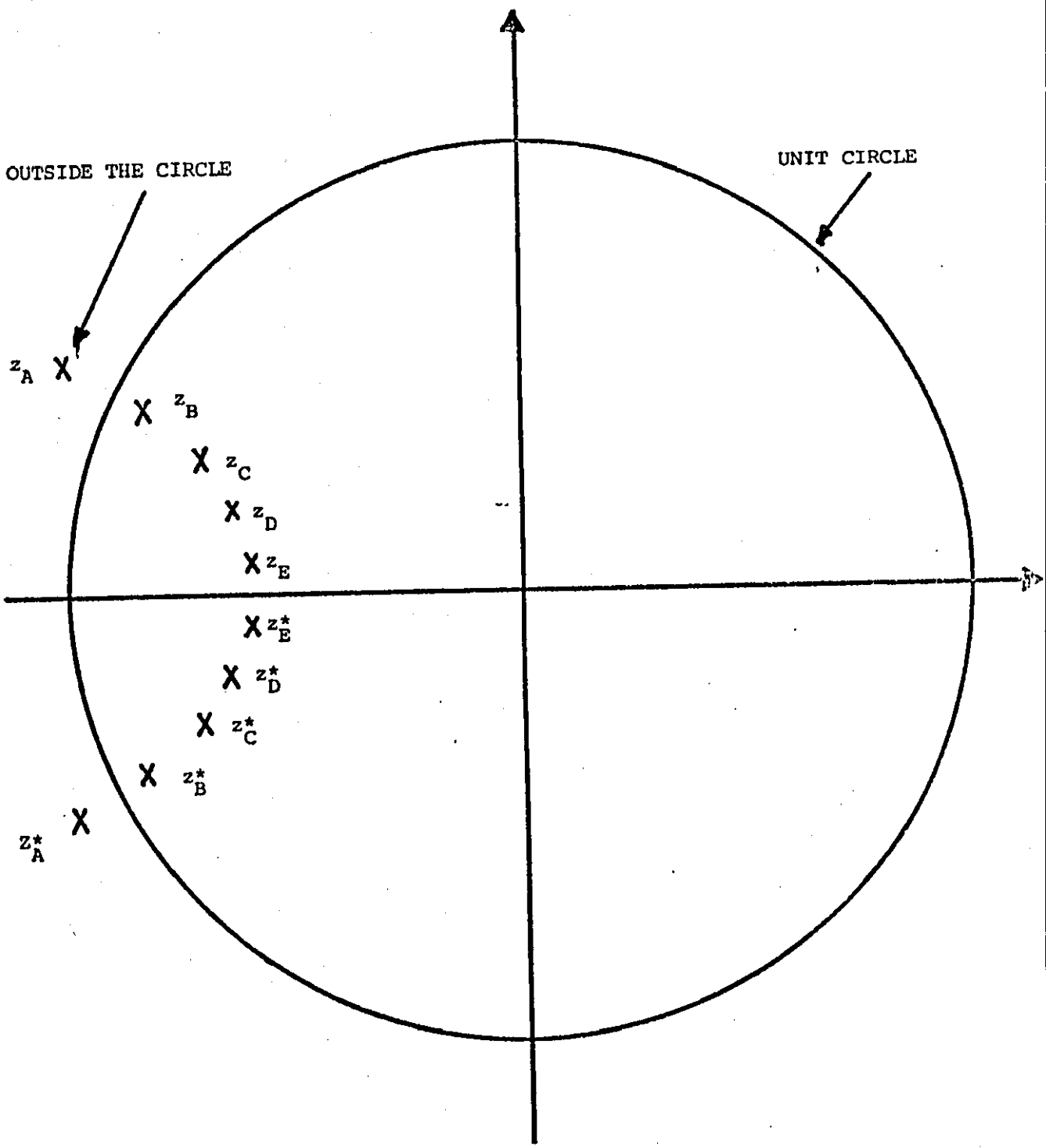


FIGURE A1.

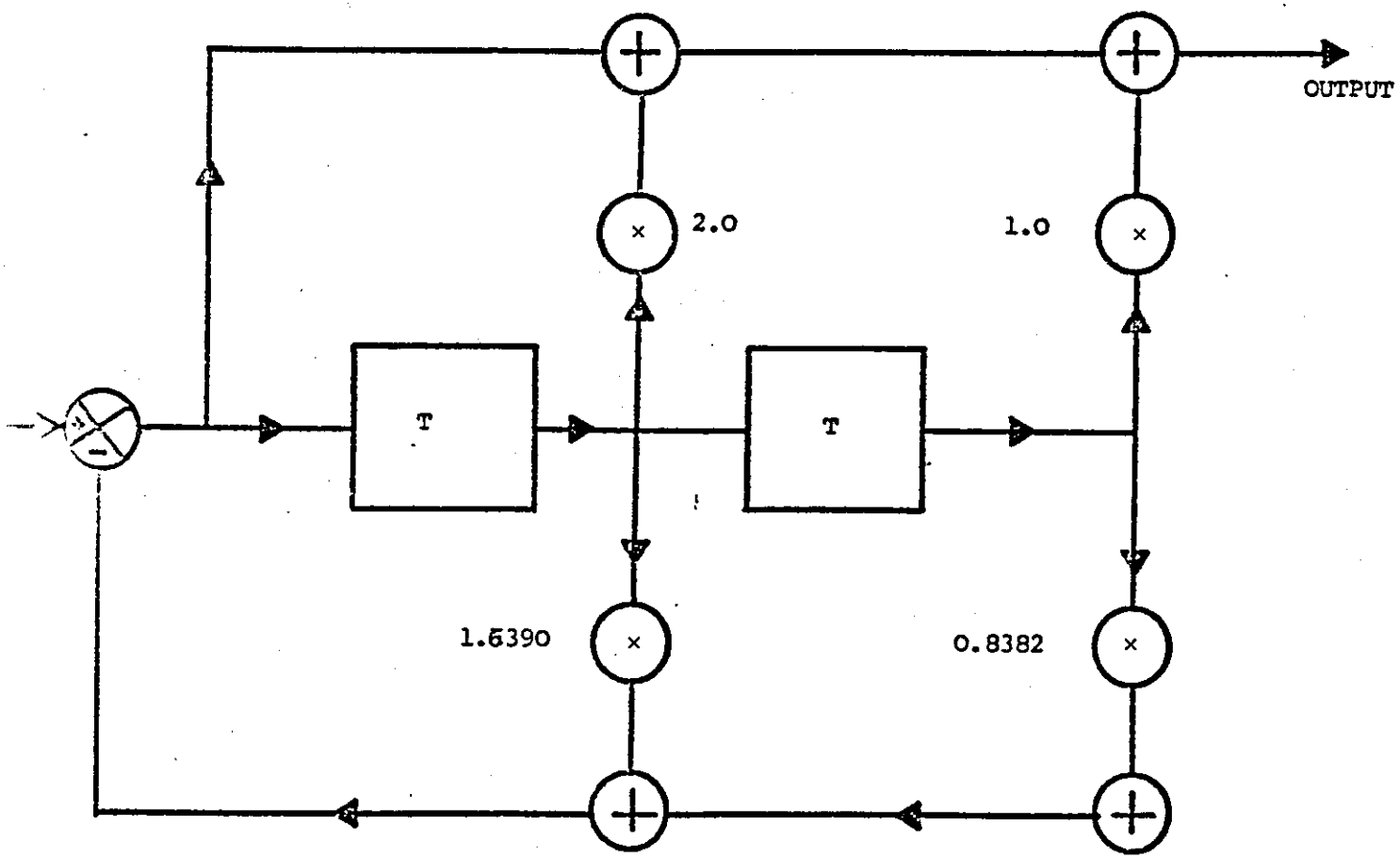


FIGURE A2 - The Second Order Recursive Filter.

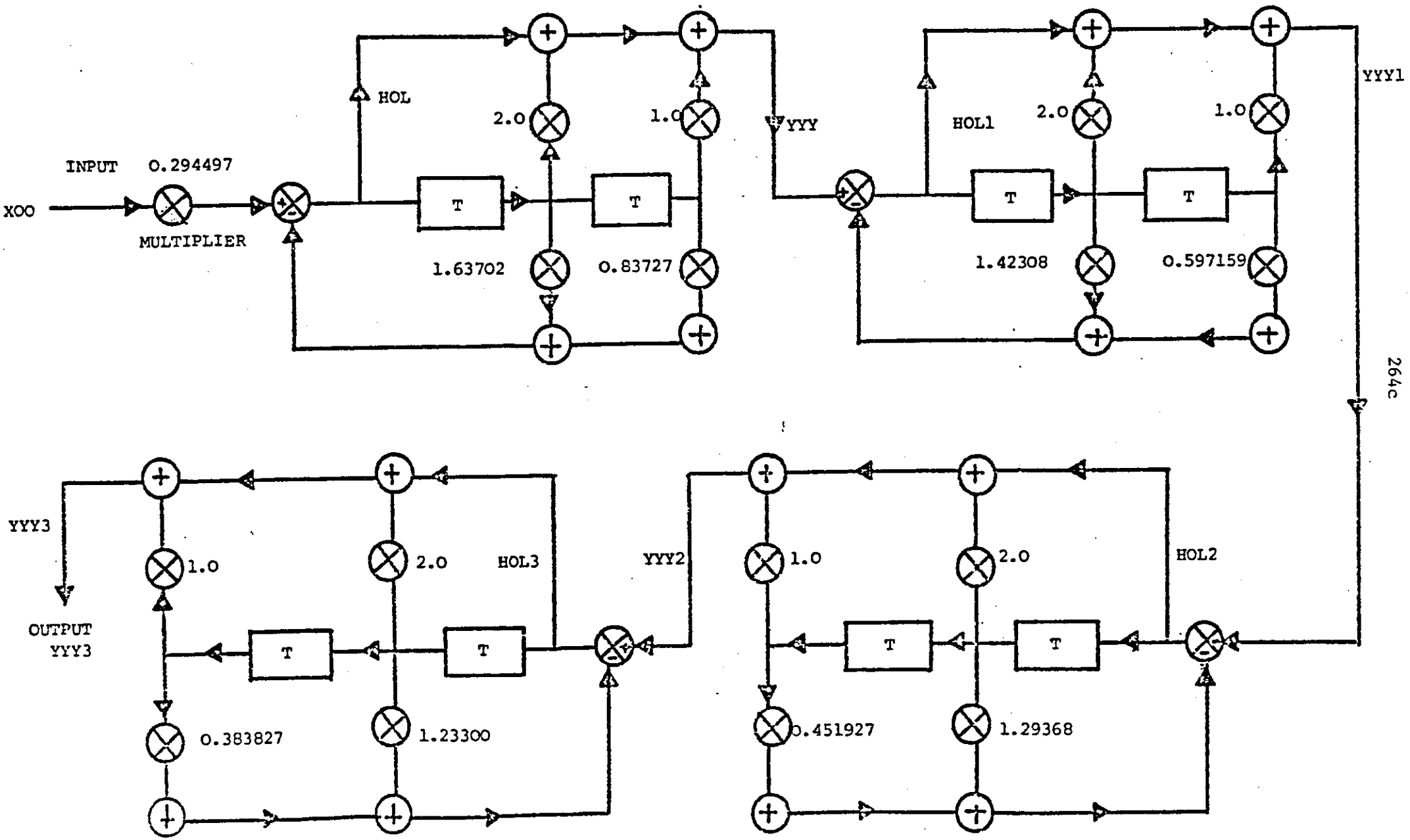


FIGURE A3 - The 8th Order Digital Filter.

```
SUBROUTINE FILT1(XOO,YYY3,A)
DIMENSION A(8)
XOO=XOO*0.294497
HOL=XOO-(1.63702*A(1)+0.837274*A(2))
YYY=HOL+2.0*A(1)+1.0*A(2)
A(2)=A(1)
A(1)=HOL
HOL1=YYY-(1.42308*A(3)+0.597149*A(4))
YYY1+HOL1+2.0*A(3)+1.0*A(4)
A(4)=A(3)
A(3)=HOL1
HOL2=YYY1-(1.29368*A(5)+0.451927*A(6))
YYY2=HOL2+2.0*A(5)+1.0*A(6)
A(6)=A(5)
A(5)=HOL2
HOL3=YYY2-(1.23300*A(7)+0.383827*A(8))
YYY3=HOL3+2.0*A(7)+1.0*A(8)
A(8)=A(7)
A(7)=HOL3
RETURN
END
```

B. Programs supporting the Input/Output operation of the HP 2100A speech processing system.

All the programs developed to operate with the HP 2100A minicomputer based speech processing system, employ the MAC1 and MAC2 subroutines. MAC1 is used to transfer speech data from a digital magnetic tape unit into the computer's memory. When the data is processed, it is stored back into the digital magnetic tape using the MAC2 subroutine.

Both subroutines have been written in Assembler programming language but are called from the main FORTRAN program. The listing of MAC1 and MAC2 is given in Lists 2 and 3 respectively.

In List 4 an absolute Assembler program is given which:

i) can transfer speech data from the Analogue-to-Digital Converter into the computer memory and thence into the digital magnetic tape,

ii) can transfer data from the digital magnetic tape into the computer memory and then outputs the data into the Digital-to-Analogue converter.

A "two buffer" strategy is used in the above "Input", "Output" program.

```
        NAM MAC1
        ENT MAC1
        EXT .ENTP,.IOC.
AGMTS   BSS 2
MAC1    NOP
        JSP .ENTP
        DEF AGMTS
        LDA AGMTS
        STA P1
        LDA AGMTS+1,I
        STA P2
TRY1    JSP .IOC.
        OCT 010116          READ A BLOCK  UNIT 6
        JMP TRY1
P1      DEF 0
P2      DEC 0
P3      JSP .IOC.
        OCT 040016          GET STATUS IF BUSY LOOP P3
        SSA
        JMP P3
        JMP MAC1,I
        END
```

```
        NAM MAC2
        ENT MAC2
        EXT .ENTP,.IOC.
AGMTS   BSS 2
MAC2    NOP
        JSP .ENTP
        DEF AGMTS
        LDA AGMTS
        STA P1
        LDA AGMTS+1,I
        STA P2
TRY1    JSB .IOC.
        OCT 020117          WRITE A BLOCK UNIT 1
        JMP TRY1
P1      DEF 0
P2      DEC 0
P3      JSB .IOC.
        OCT 040017          GET STATUS IF BUSY LOOP P3
        SSA
        JMP P3
        JMP MAC2,I
        END
```

```
*     PROGRAM FOR DATA TRANSFER IN (A/D TO MEMORY TO MAG TAPE)
*     AND OUT (MAG TAPE TO MEMORY TO D/A)
*     TWO BUFFERS HAVE BEEN USED AFOR AND BFOR ..
      OTG 2F
      LDA UNIT     SELECT MAG UNIT 0
      OTA 21F
      SFS 21F      CHECK THAT ALL OPERATIONS INCLUDING MOTION ARE COMPLE
      JMP *-1
      NOP
      NOP
*     DISABLE THE INTERRUPT SYSTEM
      CLF 0C
      JSP INPUT
      HLT 61F
      JSP OUTPU
      HLT 62F
*
*          IN SUBROUTINE
*
INPUT NOP
      LDA BLOCK    DEFINE THE NO OF BLOCKS I IF TRANSFER
      CMA,INA
      STA COUNT
      STF 21F      INIT. COMP. FOR MAG
*          PREPARE DMA1 TO TRANSFER DATA TO BUFFER AFOR ,INPUT O
      LDA CW22
      OTA 6
      CLC 2
      LDA AFOR
      ICF MASK
      OTA 2
      STC 2
      LDA CW3
      CMA,INA
      OTA 2
      .....
      STC 22F,C    START INPUT DEVICE
      STC 6F,C     START DMA1
      .....
*          PREPARE DMA2 FOR INPUT OPERATION TO BFOR ,INTO MEMORY
F1    LDA CW22
      OTA 7
      CLC 3
      LDA BFOR
      ICF MASK
      OTA 3
      STC 3
      LDA CW3
      CMA,INA
      OTA 3
      .....
      SFS 6F       IF DMA1 NOT FINISHED, WAIT
      JMP *-1
      .....
      STC 22F,C    START DEVICE
      STC 7F,C     START DMA2
      .....
*          PREPARE DMA1 TO TRANSFER DATA FROM AFOR TO THE MAG TAPE
      LDA CW20
      OTA 6
      CLC 2
      LDA AFOR
      OTA 2
      STC 2
      LDA CW3
```

```
        CMA,INA
        OTA 2
        LDA WRITE
F1      OTA 21B
        LIB 21B
        RBR,RBR
        RBR,SLB
        JMP F1
        STC 20B,C
*       ..............................
        STC 21B,C    START TAPE
        STC 6B,C     START IMA1
*       ..............................
        SFS 6B       CHECK IF IMA1 IS FINISHED
        JMP *-1      IF NOT WAIT
*            PREPARE IMA1 FOR INPUT OPERATION TO ABUF ,IN MEMORY
        LDA CW02
        OTA 6
        CLC 2
        LDA ABUF
        IOR MASK
        OTA 2
        STC 2
        LDA CW3
        CMA,INA
        OTA 2
*       ..............................
        SFS 7B       CHECK IF INPUT OPERATION TO BBUF BY IMA2 IS FINISH
        JMP *-1      IF NOT WAIT
*       ..............................
        STC 22B,C    START INPUT DEVICE
        STC 6B,C       START IMA1
*       ..............................
*            PREPARE IMA2 TO TRANSFER DATA FROM BBUF TO MAC TAPE
        LDA CW20
        OTA 7
        CLC 3
        LDA BBUF
        OTA 3
        STC 3
        LDA CW3
        CMA,INA
        OTA 3
        LDA WRITE
F2      OTA 21B
        LIB 21B
        RBR,RBR
        RBR,SLB
        JMP F2
*       ..............................
        STC 20B,C
        STC 21B,C    START MAC TAPE
        STC 7B,C     START IMA2
*       ..............................
        SFS 7B       IS IMA2 FINISHED?
        JMP *-1      IF NOT WAIT
*       ..............................
        ISZ COUNT
        JMP F1
        LIA 21B
        JMP INPUT,I
*            END OF INPUT SUBROUTINE
*
*            OUT SUBROUTINE
*
OUTPUT NOP
        LDA CLOCK
```

```
        CMA,INA
        STA COUNT
        NOP
        NOP
*               PREPARE DMA1 TO TRANSFER DATA FROM MAC TAPE TO MEMORY, A
        LDA CW20
        OTA 6
        CLC 2
        LDA ABUF
        IOR MASK
        OTA 2
        STC 2
        LDA CW3
        CMA,INA
        OTA 2
        LDA READ
R3      OTA 21B
        LIB 21B
        RBR,RBR
        RBR,SLB
        JMP R3
*       ..............................
        STC 20B,C
        STC 21B,C    START MAC TAPE
        STC 6B,C     START DMA1
*       ..............................
*               PREPARE DMA2 TO TRANSFER DATA FROM ABUF TO THE OUTPUT DE
P2      LDA CW22
        OTA 7
        CLC 3
        LDA ABUF
        OTA 3
        STC 3
        LDA CW3
        CMA,INA
        OTA 3
*       ..............................
        SFS 6B       IS DMA1 FINISHED READ FROM MAC TAPE TO ABUF
        JMP *-1      IF NOT WAIT
*       ..............................
        STC 22B,C START OUTPUT DEVICE
        STC 7B,C     START DMA2
*       ..............................
*               PREPARE DMA1 TO TRANSFER DATA FROM MAC TAPE TO BBUF
        LDA CW20
        OTA 6
        CLC 2
        LDA BBUF
        IOR MASK
        OTA 2
        STC 2
        LDA CW3
        CMA,INA
        OTA 2
        LDA READ
R4      OTA 21B
        LIB 21B
        RBR,RBR
        RBR,SLB
        JMP R4
*       ..............................
        STC 20B,C
        STC 21B,C    START MAC TAPE
        STC 6B,C     START DMA1
*       ..............................
        SFS 6B       IS DMA1 FINISHED
        JMP *-1      IF NOT WAIT
```

```
*     ..............................
*           PREPARE IMA1 TO TRANSFER DATA FROM PBUF TO OUTPUT DEVICE
      LDA CW22
      OTA 6
      CLC 2
      LDA PBUF
      OTA 2
      STC 2
      LDA CW3
      CMA,INA
      OTA 2
*     ..............................
      SFS 7B      IS IMA2 FINISHED
      JMP *-1     IF NOT WAIT
*     ..............................
      STC 22B,C   START OUTPUT DEVICE
      STC 6B,C    START IMA1 FOR OUTPUT OPERATION
*     ..............................
*           PREPARE IMA2 TO TRANSFER DATA FROM MAG TAPE TO ABUF
      LDA CW20
      OTA 7
      CLC 3
      LIA ABUF
      IOR MASK
      OTA 3
      STC 3
      LIA CW3
      CMA,INA
      OTA 3
      LDA HEAD
15    OTA 21B
      LIB 21B
      FPB,FFB
      FPB,SLB
      JMP B5
*     ..............................
      STC 20B,C
      STC 21B,C   START MAG TAPE
      STC 7B,C     START IMA2
*     ..............................
      SFS 7B      IS IMA2 FINISHED
      JMP *-1
*     ..............................
      ISZ COUNT
      JMP B2
      LIA 21B
      JMP OUTPT,I
*           END OF CDL SUBROUTINE
*
UNIT  OCT 1400
CW20  OCT 120020
CW22  OCT 120022
CW3   DEC 1000
BLOCK DEC 201
MASK  OCT 100000
HEAD  OCT 23
WRITE OCT 31
ABUF  DEF STOP
PBUF  DEF STOP1
COUNT BSS 1
STOP  BSS 5000
STOP1 BSS 5000
      END
```

REFERENCES

1. CATTERMOLE, K.W. - "Principles of Pulse Code Modulation" (book), London Illiffe, 1973.

2. STEELE, R. - "Delta Modulation Systems" (book), Pentesh Press, 1975.

3. LATHI, B.P. - "Random Signals and Communication Theory" (book), Intertext Books, 1968.

4. PANTER, P.F. - "Modulation, Noise and Spectral Analysis" (book), McGraw-Hill, 1965.

5. REEVES, A.H. - Brit. Patent 535,860 (1939).

6. STEELE, R. and GOODMAN, D.J. - "Detection and Selective Smoothing of Transmission Errors in Linear PCM". B.S.T.J., Vol. 56, 1977.

7. FLANAGAN, J.L. - "Speech Analysis, Synthesis and Perception", (book), Springer-Verlag, 1965.

8. FANT, G. - "Acoustic Theory of Speech Production" (book), Mouton, 1970.

9. VON BEKESY, G. - "Experiments in Hearing", (book), McGraw-Hill, 1960.

10. DUDLEY, H. - The Vocoder, Bell Lab. Record 17, 122-126, 1939.

11. SCHROEDER, M.R. and Others - "New Applications of Voice-Excitation to Vocoders", Stockholm Speech Communications Seminar, 1962.

12. GOLD, B., TIERNEY, J. - "Digital Voice Excited Vocoder for Telephone Quality Inputs, using Bandpass Sampling of the Base-band Signal", J.A.S.A. Vol. 37, No.4, April 1965.

13. OPPENHEIM, A.V. - "Speech Analysis-Synthesis System Based on Homomorphic Filtering", J.A.S.A. Vol. 45, 459-462, 1969.

14. COKER, G.H. - "Computer Simulated Analyser for a Format Vocoder", J.A.S.A. Vol. 35, 1963.

15. MARKEL, J.D. - "Digital Inverse Filtering - A new Tool for Format Trajectory Estimation", I.E.E.E. TRANS. on Audio and Electroac., AU-20, 1972.

16. ATAL, B.S., HANAUER, S.L. - "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", J.A.S.A. Vol. 50, 635-655, 1971.

17. MARKEL, D.J., GRAY, A.H. - "Linear Prediction of Speech" (book), New York, Springer-Verlag, 1976.

18. MAKHOUL - "Linear Prediction: a Tutorial Review", Proc. I.E.E.E. Vol. 63, 1975.

19. MARKEL, J.D., GREY, J.R. - "On Autocorrelation Equations as Applied to Speech Analysis", I.E.E.E. TRANS. on Audio and Electroacoust. Vol. AU-21, 1973.

20. Final Report - Contract No. DCA 100-72-C-0036, "Development of a Configuration Concept of a Speech Digitizer based on Adaptive Estimation Techniques", Institute of Technology, Southern Methodist University, Dallas, Texas, U.S.A., 1973.

21. DALEY, D.J. - Loughborough University of Technology, Dept. of Electronic Eng. England. Private Communication.

22. ITAKURA, F., SAITO, S. - "On the Optimum Quantization of Feature Parameters in the PARCOR Speech Synthesizer", I.E.E.E. Proc. Conf. Speech Commun. Process 434-437, 1972.

23. SCAGLIOLA, C. - "Automatic Vocal Tract Parameter Estimation by an Iterative Algorithm", Proc. 8th Internat. Congress on Acoustics, London, Vol. 1, July 1974.

24. GOLD, B., RABINER, L. - "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain", J.A.S.A. Vol. 45, 1969.

25. NOLL, A.M. - "Cepstral Pitch Determination", J.A.S.A. Vol. 41, 1967.

26. ROSS, M., and Others - "Average Magnitude Difference Function Pitch Extractor", I.E.E.E. TRANS. Acoustics Speech and Signal Processing, Vol. ASSP-22, Oct. 1974.

27. RABINER, L. - "On the Use of Autocorrelation Analysis for Pitch Detection", I.E.E.E. TRANS. Comput. Vol. C-20, Jan. 1971.

28. MARKEL, J.D. - "The SIFT Algorithm for Fundamental Frequency Estimation", I.E.E.E. TRANS. Audio and Electroacoust. Vol. AU-20, Dec. 1972.

29. JORDAN, B.P., and KELLY, L.C. - "A Comparison of the Speech Quality of a Linear Predictive Coder and a Channel Vocoder", 8th Intern. Cong. on Accustics, London, July 1974.

30. GOLD, B. - "Digital Speech Networks", Proceedings of the I.E.E.E., Vol. 65, No.12, Dec. 1977.

31. PANTER, P.F., and DITE, W. - "Quantizing Distortion in Pulse Count Modulation with Non-uniform Spacing of Levels", Proc. I.R.E. 39, Jan. 1951.

32. SMITH, B. - "Instantaneous Companding of Quantized Signals", B.S.T.J. No. 27, 446-472, 1948.

33. ROE, G.M. - "Quantizing for Minimum Distortion", I.E.E.E. TRANS. on Inf. Theory., IT-10, 1964.

34. ALGAZI, V.R. - "Useful Approximation to Optimum Quantization", I.E.E.E. TRANS. on Comm. Techn., COM-14, 1966.

35. MAX, J. - "Quantizing for Minimum Distortion", I.R.E. TRANS. Infor. Theory, IT-6, May 1960.

36. PAEZ, M.D., GLISSON, T.H. - "Minimum Mean-Squared-Error Quantization in Speech PCM and DPCM Systems", I.E.E.E. TRANS. Comm. Vol. COM-20, Apr. 1972.

37. GOLDING, L.S., SCHULTHEISS, P.M. - "Study of an Adaptive Quantizer", Proc. of I.E.E.E. Vol. 55, Mar. 1967.

38. SCHLINK, W. - "A Redundancy Reducing PCM System for Speech Signals", Proc. Inter. Zurich Seminar Integrated Systems for Speech", Video and Data Communications, pp.F4/1-4, Mar. 1972.

39. STROH, R.W. - "Optimum and Adaptive DPCM", Ph.D. Dissertation, Polytechnic Inst. of Brooklyn, Brooklyn, New York, 1970.

40. NOLL, P. - "Adaptive Quantizing in Speech Coding Systems", Inter. Zurich Seminar on Digital Comm. (I.E.E.E.) pp.B3.1-B3.6, March 1974.

41. JAYANT, N.S. - "Adaptive Quantization with One-Word-Memory", B.S.T.J. Vol. 52, May 1974.

42. JAYANT, N.S. and Others - "Adaptive-DPCM", B.S.T.J. Vol. 52, No.9, Sept. 1973.

43. GOODMAN, D.J., GERSHO, A. - "Theory of an Adaptive Quantizer", I.E.E.E. TRANS. on Commun. Vol. COM-22, Aug. 1974.

44. JAYANT, N.S., RABINER, L.R. - "The Application of Dither to the Quantization of Speech Signals", B.S.T.J. July-Aug., 1293-1304, 1972.

45. WOOD, R.G., TURNER, L.F. - "Pseudorandomly Dithered Quantization of Speech Samples in PCM Transmission Systems", Proc. I.E.E. Vol. 119, May 1972.

46. CHEN, M., TURNER, L.F. - "Zero-Crossing Perservation in Low-Bit-Rate Pseudorandomly Dithered Quantization of Speech Signal", Proc. I.E.E. Vol. 122, Feb. 1975.

47. ELIAS, P. - "Predictive Coding", I.R.E. TRANS. Information Theory, IT-1, 16-33, 1955.

48. HASKEW, J.R. - "A Comparison between Linear Prediction and Linear Interpolation", MS Thesis, Electrical Eng. Dept. Brooklyn Polytechnic Institute, New York, June 1969.

49. CUTLER, C.C. - "Differential Quantization of Communications", U.S. Patent 2-605-301, July 29, 1952.

50. OLIVER, B.N. - "Efficient Coding", B.S.T.J. Vol. 31, pp.724, July 1952.

51. HARRISON, C.W. - "Experiments with Linear Prediction in Television", B.S.T.J. Vol. 31, pp.764, July 1952.

52. KRETZMER, E.R. - "Statistics of Television Signals", B.S.T.J. Vol. 31, pp.75, July 1952.

53. McDONALD, R.A. - "Signal to Noise and Idle Channel Performance of DPCM Systems - Particular Application to Voice Signals", B.S.T.J. Vol. 45, No.7, Sept. 1966, pp.1123.

54. NITADORI, K. - "Statistical Analysis of DPCM", J. Inst. Electron. Commun. Eng. of Japan, pp.4-6, 1965.

55. CUMMISKEY, P. - "Adaptive DPCM for Speech Processing", Ph.D. Dissertation, Newark College of Engineering, Newark, N.J. 1973.

56. NOLL, P. - "Non-Adaptive and Adaptive DPCM of Speech Signals", Overdruk uit Polytech. Tijdschr. Ed. Elektrotech./ Eletron. (The Netherlands) No.19, 1972.

57. CHAN, D., DONALDSON, R.W. - "Subjective Evaluation of Pre-and Postfiltering in PAM, PCM, DPCM Voice Communication Systems", I.E.E.E. TRANS. on Commun. Technol. Vol. COM-19, pp.601, Oct. 1971.

58. YAN, J., DONALDSON, R.W. - "Subjective Effects of Channel Transmission Errors on PCM and DPCM Voice Communication Systems", I.E.E.E. TRANS. on Commun., Vol. COM-20, pp.281, June 1972.

59. O'NEAL, J.B. - "Signal to Quantizing Noise Ratios for Differential PCM", I.E.E.E. TRANS. on Comm. Techn., Vol. COM-19, No.4, p.568, Aug. 1971.

60. GISH, H. - "Optimum Quantization of Random Sequences", Defence Supply Agency Rp. AD 656-042, May 1967.

61. O'NEAL, J.B. STROH, R.W. - "Differential PCM for Speech and Data Signals", I.E.E.E. TRANS. on Commun. Vol. COM-20, No.5, p.900, Oct. 1972.

62. NOLL, P. - "A Comparative Study of Various Quantization Schemes for Speech Encoding", B.S.T.J., Vol. 54, No.9, p.1597, November 1975.

63. GIBSON, J.D., JONES, S.K., MELSA, J.L. - "Sequentially Adaptive Prediction and Coding of Speech Signals", I.E.E.E. TRANS. on Commun. Vol. COM-22, p.1789, Nov. 1974.

64. JONES, S.K. - Private Communication.

65. SAGE, A.P., MELSA J.L. - "Estimation Theory with Applications to Communications and Controls", New York, McGraw-Hill, 1971.

66. JAZWINSKI, A.H. - "Stochastic Processes and Filtering Theory", N.Y. Academic, 1970.

67. ATAL, B., SCHROEDER, M.R. - "Adaptive Predictive Coding of Speech Signals", B.S.T.J. Vol. 49, No.8, p.1973, Oct. 1970.

68. COHN, D.L., MELSA, J.L. - "The Residual Encoder - An Improved ADPCM System for Speech Digitization", I.E.E.E. TRANS. on Commun. Vol. COM-23, No.9, Sept. 1975.

69. QURESHI, S.U.. FORNEY, D.G. - "A 9.6/16 Kb/sec. Speech Digitizer", I.E.E.E. Int. Conf. on Commun. Vol. 11, June 16-18, pp.30, 1975.

70. ASH, R.B. - "Information Theory", John Wiley and Sons, New York, 1967.

71. O'NEAL, J.B. - "Entropy Coding in Speech and Television DPCM Systems", I.E.E.E. TRANS. on Information Theory, Vol. IT-17, Nov. 1971.

72. DE JAGER, F. - "Delta Modulation - a New Method of PCM Transmission using 1 Unit Code", Philips Research Report, 7, 442-446, Dec. 1952.

73. VAN DE WEG, H. - "Quantizing Noise of a Single Integration Delta Modulation System with an N-Digit Code", Philips Research Report No.8, 367-385, 1953.

74. O'NEAL, J.B. - "Delta Modulation Quantizing Noise Analytical and Computer Simulation Results for Gaussian and Television Signals", B.S.T.J. Vol. 45, No.1, p.117, Jan. 1966.

75. GOODMAN, D.J. - "Delta Modulation Granular Quantization Noise", BS.T.J. May-June, p.1197, 1969.

76. ZETTERBERG, L.H. - "A Comparison between Delta and Pulse Code Modulation", Ericsson Technies, Vol. 11, No.1, p.95, Jan. 1955.

77. PASSOT, M. - "Normal Spectra Application to the Study of Delta Modulation", Master's Thesis, Electronic and Electrical Eng. Dept., University of Technology, Loughborough, England, 1973.

78. PROTONOTARIOS, E.N. - "Slope Overload Noise in Differential Pulse Code Modulation Systems", BS.T.J. Vol. 46, No.9, p.2119, Nov. 1967.

79. GREENSTEIN, L.J. - "Slope Overload Noise in Linear DM with Gaussian Input", B.S.T.J. Vol. 52, No.3, p.387, March 1973.

80. CUTLER, C. - "Delayed Encoding: Stabilizer for Adaptive Coders", I.E.E.E. TRANS. on Commun. Vol. COM-19, p.898, Dec. 1971.

81. WINKLER, M.R. - "High Information Delta Modulation", I.E.E.E. Intern. Conv. Rec. Pt.8, p.260, 1963.

82. BELLO, P.A., LINCOLN, R.N., GISH, H. - "Statistical Delta Modulation", Proc. I.E.E.E. Vol. 55, No.3, p.308, March 1967.

83. ABATE, J.E. - "Linear and Adaptive Delta Modulation", Proc. I.E.E.E. Vol. 55, No.3, p.298, March 1967.

84. FLOOD, J.E., HAWKSFORD, M.J. - "Adaptive Delta-Sigma Modulation using Pulse Grouping Techniques", Joint Confer. on Digital Processing of Signals in Commun., University of Technology, Loughborough, April 1972.

85. BOSWORTH, R.H., CADOY, J.C. - "A Companded One Bit Coder for Television Transmission", B.S.T.J. Vol. 48, p.1459, July 1969.

86. JAYANT, N.S. - "Adaptive Delta Modulation with One Bit Memory", B.S.T.J. No.3, p.321, March 1970.

87. KYAW, A.T. - "Constant Factor Delta Modulation", Ph.D. Thesis, Electronic and Electrical Eng. Dept., Loughborough University of Technology, 1973.

88. SONG, C.L., GARODNICK, J., SCHILLING, D.L. - "Available Step Size Robust Delta Modulator", I.E.E.E. TRANS. on Commun. Vol. COM-19, No.6, p.1033, Dec. 1971.

89. GREEFKES, J.A., DE JAGER, F. - "Continuous Delta Modulation", Philips Res. Rept. No.23, p.233, 1968.

90. TOMOZAWA, A., KANEKO, H. - "Companded Delta Modulation for Telephone Transmission", I.E.E.E. TRANS. on Commun. Vol. COM-16, p.149, Feb. 1968.

91. BROLIN, S.J., BROWN, J.H. - "Companded DM for Telephony, I.E.E.E. TRANS. on Commun. Techn. Vol. COM-16, p.151, Feb. 1968.

92. GREEFKES, J.A., RIEMENS, K. - "Code Modulation with Digitally Controlled Companding for Speech Transmission", Philips Technical Review, Vol. 31, No. 11/12, p.335, 1970.

93a. WILKINSON, R.M. - "A Companded Delta-Sigma Modulation System", Report No. 68012, U.D.C. No. 621,395. 6+621. 376.5, S.R.D.E. Christchurch, Hants., July 1968.

93b. CARTMALE, A.A., STEELE, R. - "Calculating the Performance of Syllabically Companded Delta-Sigma Modulators", Proc. I.E.E., 1915,1921, 1970.

94. HUANT, J., SCHULTHEISS, P. - "Block Quantization of Correlated Gaussian Random Variables", I.E.E.E. TRANS. on Commun. Systems, Vol. CS-11, p.289, 1963.

95. ZELINSKI, R., NOLL, P. - "Adaptive Block Quantization of Speech Signals", Heinrich-Hertz-Institut., Berlin, Technical Report 181, 1975.

96. ZELINSKI, R., NOLL, P. - "Adaptive Transform Coding of Speech Signals", I.E.E.E. TRANS. on Acoustics, Speech and Signal Processing, Vol. ASSP-25, No.4, p.229, Aug. 1977.

97. PEARL, J. - "On Coding and Filtering Stationary Signals by Discrete Fourier Transforms", I.E.E.E. TRANS. on Inform. Theory, Vol. IT-19, p.229, 1973.

98. AHMED, N., NATARAZAN, T., RAO, K. - "Discrete Cosine Transform", I.E.E.E. TRANS. on Comput. Vol. C-23, 1974.

99. SHUM, F., ELLIOT, A., BROWN, W. - "Speech Processing with Walsh-Hadamard Transforms", I.E.E.E. TRANS. Audio Electroacoustics Vol. AU-21, p.174, 1973.

100. PRATT, W., CHEN, W., WELCH, L. - "Slant Transform Image Coding", I.E.E.E. TRANS. Commun. Vol. COM-22, p.1075, 1974.

101. CAMPANELLA, S.J., ROBINSON, G.R. - "A Comparison of Orthogonal Transformations for Digital Speech Processing", I.E.E.E. Commun. Techn. Vol. COM-19, p.1045, 1971.

102. FRANGOULIS, E., TURNER, L.F. - "Hadamard-Transformation Technique of Speech Coding: Some Further Results", Proc. I.E.E. Vol. 124, No.10, Oct. 1977.

103. MODENA, G., NEBBIA, L, SCAGLIOLA, C. - "Bit-Rate Reduction of Digital Speech Transmission by Linear Transformations", Signal Processing, W. Schussler, Ed., Erlangen, Germany, 1973.

104. SEVERWRIGHT, J.S. - "Interruption Techniques for Efficient Speech Transmission", Ph.D. Thesis, Dept. of Electronic and Electrical Eng., Loughborough University of Technology, Loughborough, England.

105. BASKARAN, P. - "Digital Coding of Speech", Loughborough University of Technology Report for Joint Speech Research Unit, London, Oct. 1972 - April 1973.

106. FREI, A.H., SCHINDLER, H.R., and Others - "Adaptive Predictive Speech Coding based on Pitch-Controlled Interruption - Reiteration Techniques", Proc. I.E.E.E. Int. Conf. Commun., Wash. June 1973, p.46-12.

107. WILKINSON, R.M. - "One Bit Adaptive PCM Report No. S.R.D.E. Christchurch, Hants. England, 19

108. SCHINDLER, H.R. - "Delta Modulation", I.E.E.E. Spectrum, Vol. 7, p.69, Oct. 1970.

109. CHEN, M. - "Low-Bit Rate Coding of Speech Signals", Ph.D. Thesis, Dept. of Electrical Eng. Imperial College of Science and Technology, University of London, 1976.

110. XYDEAS, C.S., STEELE, R. - "Pitch Synchroneous First Order DPCM System", Electronic Letters of I.E.E., Vol. 12, No.4, Feb. 1976.

111. GIBSON, J.D. - "Sequentially Adaptive Backward Prediction in ADPCM Speech Coders", I.E.E.E. TRANS. on Commun. Vol. COM-26, No.1, Jan. 1978.

112. PIRAMI, G., SCAGLIOLA, C. - "Performance Analysis of DPCM Speech Transmission Systems using Kalman Predictors", I.E.E.E. Intern. Cont. on Acoustics, Speech and Signals, Proc. April, 1976, Philadelphia.

113. EVCI, G. - "Private Communication", Dept. of Electronic Eng., Loughborough University of Technology.

114. JAYANT, N.S. - "Pitch-Adaptive DPCM Coding of Speech with Two-Bits Quantization and Fixed Spectrum Prediction", B.S.T.J. Vol. 56, No.3, March 1977.

115. GOLDBERG, A.J., and Others - "High Quality 16 Kb/s Voice Transmission", I.E.E.E. Intern. Conf. on Acoust., Speech and Signal Proces., April 1976, Philadelphia.

116. CODEX CORPORATION - "9.6/16 Kb/s Adaptive Gradient Speech Coder", Report on Contract No. DCA100-74-C-0053.

117. WIDROW, B. - "Adaptive Filters", Report prepared under Contract No. DA-01-021 AMC-90015(Y) System Lab. Stanford University, California.

118. CASTELLINO, P., and Others - "Bit-Rate Reduction by Automatic Adaptation of Quantizer Step Size in DPCM Systems", Proc. 1974, Int. Zurish Seminar Digital Commun. p.B6(1).

119. XYDEAS, C.S., STEELE, R. - "Dynamic Ratio Quantizer", Proc. of I.E.E. Vol. 125, No.1, Jan. 1978.

120. ACKROYD, M.H. - "Digital Filters", (book) Butterworths, London, 1973.

121. NEWTON, C.M. - "Delta Modulation with Slope Overload Prediction", Electronic Letters, Vol. 7, No.9, April 30, 1970.

122. ZETTERBERG, L.H., UDDENFELDT, J. - "Adaptive Delta Modulation with Delayed Decision", I.E.E.E. TRANS. on Commun. Vol. COM-22, No.9, Sept. 1974.

123. KOUBANITSAS, T.S. - "Application of the Viterbi Algorithm to Adaptive Delta Modulation with Delayed Decision", Proc. of I.E.E.E., July 1975, p.1076.

124. FORESTER, E. - "High Information Delayed DM", Final Year Project, Dept. of Electronic Eng. Loughborough University, Loughborough, 1977.

125. ANDERSON, J.B., BODIE, J.B. - "Tree Encoding of Speech", I.E.E.E. TRANS. on Inform. Theory, Vol. IT-21, p.379, July 1975.

126. LICKLIDER, J.C.R. - "The Intelligibility of Amplitude Dichotomized Time Quantized Speech Waves", J.A.S.A. Vol. 22, p.820, 1950.

127. MORRIS, L.R. - "The Role of Zero Crossings in Speech Recognition and Processing", Ph.D. Thesis, Imperial College, London, 1970.

128. XYDEAS, C.S., STEELE, R. - "Pitch Synchroneous Differential Predictive Encoding System", Electronic Letters of I.E.E., Vol. 12, No.5, July 1976.