# TRANSFORM CODING TECHNIQUES FOR LOW-BIT-RATE SPEECH COMMUNICATION

by

Mohammad Reza Ashouri

A thesis submitted for the degree of
Doctor of Philosophy in the Faculty
of Engineering, University of London.

Department of Electrical Engineering,
Imperial College of Science and Technology,
University of London.

January 1979

## ABSTRACT

The main concern of this thesis is the development of a new digital speech coding system that enables transmission of intelligible speech to take place, at bit-rates of less than 1 bit per sample where the complexity is not far greater than that of a simple waveform coder. The thesis begins with a brief review of various methods of digital speech coding techniques. The review is mainly concerned with those techniques which represent different classes of speech coding and emphasis is placed on those coding principles that are related to the development of new coding techniques.

To achieve this goal of an efficient speech coding system, the research effort has been divided into two parts, (i) on the development of an efficient coding algorithm and (ii) on the design of an efficient implementation technique. In the search for a new coding algorithm, investigations have been carried out in the areas of transform coding and linear prediction coding. A new linear predicition algorithm, referred to as Non-Uniform Linear Prediction (NULP), has been developed for efficient coding of speech in the transform domain. In this method a few base-band coefficients of the transform spectrum are transmitted with the rest of the spectrum modelled by using a few prediction coefficients. The computation of prediction coefficients is based on spectral correlation, non-uniform energy distribution and the least-squared-error criterion. For efficient coding of unvoiced sounds a different coding scheme has been introduced where no pitch extraction is involved.

The NULP technique has been applied in the Walsh-Hadamard and the Discrete Cosine domains where intelligible speech with different qualities at bit rates of 4, 6.5, and 8 k bits/sec have been achieved. Based on the recent development of objective measurement of speech quality, a simple method has been described for quality evaluation of the decoded speech from the NULP system. Comparisons have shown that the proposed algorithm has superior performance at low bit rates to that of the best non-pitch tracking systems while its complexity is significantly lower.

The second part of the thesis is devoted to a new technique for the implementation of the digital speech processors. This technique is based on a new system of binary number representation referred to as Intermediate Binary Numbers (IBN). The main advantages of the IBN system over the conventional binary system are the simpler analogue to digital conversion and less complex arithmetic units used in the hardware implementation. The analogue-to-IBN conversion and vice-versa based on delta-sigma and delta modulator is described and methods of IBN addition, subtraction and multiplication are developed. The IBN technique is also employed for the implementation of the Walsh-Hadamard transform, Fast Walsh-Hadamard transform, and Discrete Cosine transform with simplified relations derived for computing the finite register length error in such systems. It is also shown that other methods of digital signal processing based on delta-modulation techniques are special cases of the IBN digital signal processing approach.

ACKNOWLEDGEMENTS

I wish to express deep gratitude to Dr. A.G. Constantinides whom I am privileged to have had as supervisor. His enthusiasm for the subject and technical guidance and criticism have greatly influenced the course of this research.

Additional thanks are due to the staff and research students of the Communication Section for their various forms of assistance too numerous to mention.

I also wish to thank the following: Mr. R. Sulley who organised the PDP-15 computer facilities; the staff of Imperial College Computer Centre for their help and advice; and Miss E. Straker for her skillful typing of the thesis.

Special thanks are due to Mr. S. Samsam for arranging financial support from the Electronic Research Centre (Arya Mehr University), for the final year of this research study.

Finally, I bear the deepest gratitude to my parents for their moral and financial support during the years of this study. To them, I dedicate this thesis.

CONTENTS


| | | Page |
|---|---|---|
| TITLE PAGE | | 1 |
| ABSTRACT | | 2 |
| ACKNOWLEDGEMENTS | | 4 |
| CONTENTS | | 5 |
| LIST OF FIGURES AND TABLES | | 9 |
| CHAPTER 1 | Introduction | 12 |
| 1.1. | A Brief Survey of the Important Digital speech Coding Techniques | 13 |
| 1.2. | Time Domain Waveform Coding of Speech | 14 |
| 1.2.1. | Pulse Code Modulation (PCM) | 14 |
| 1.2.2 | Differential Pulse Code Modulation with a Fixed Predictor (DPCM-F) | 15 |
| 1.2.3. | Adaptive Pulse Code Modulation with a Fixed Predictor (A PCM-F) | 16 |
| 1.2.4. | Adaptive Differential Pulse Code Modulation with a Variable Predictor (ADPCM-V) | 16 |
| 1.2.5. | Delta Modulation (DM) | 17 |
| 1.2.6. | Time-Encoded Speech | 18 |
| 1.3. | Frequency Domain Coding of Speech | 19 |
| 1.3.1. | Sub-Band Coding (SBC) | 20 |
| 1.3.2. | Formant Vocoder | 21 |
| 1.3.3. | Homomorphic Vocoder | 21 |
| 1.4. | Transform Domain Coding of Speech | 23 |
| 1.4.1. | Walsh-Hadamard Transform | 23 |
| 1.4.2. | Discrete Cosine Transform | 25 |
| 1.4.3. | Karhunen-Loeve Transform | 27 |
| 1.4.4. | Transform Coding with Fixed Bit Allocation (TCFBA) | 28 |
| 1.4.5. | Adaptive Transform Coding (ATC) | 30 |
| 1.4.6. | Transform Coding of Dominant Coefficients (TCDC) | 31 |

| | | Page |
|---|---|---|
| 1.5. | Linear Prediction Coding of Speech (All-Pole Modelling) | 32 |
| 1.5.1. | Covariance and Autocorrelation Approach | 34 |
| 1.5.2. | Partial Correlation Approach (PARCOR) | 35 |
| 1.5.3. | Linear Prediction Vocoder | 37 |
| 1.6. | Non-Uniform Linear Prediction and Intermediate Binary Numbers: New Techniques for Designs of an Efficient Speech Coding System. | 39 |
| 1.7 | Organisation of the Thesis | 41 |
| CHAPTER 2 | Non-Uniform Linear Prediction Coding of Speech | 55 |
| 2.1. | The Inefficiency of the All-Pole LP Coding in Transform Domain | 56 |
| 2.2. | Non-Uniform Linear Prediction of Speech | 58 |
| 2.2.1. | Determination of the Weighting and Prediction Vector | 60 |
| 2.2.2 | Determination of Prediction Coefficients | 61 |
| 2.3. | Choice of Efficient Transforms | 61 |
| 2.4. | Non-Uniform Linear Prediction of the WH Spectrum | 63 |
| 2.5. | Voiced-Unvoiced Decision | 64 |
| 2.6. | Quantization of Parameters | 66 |
| 2.7. | Objective Evaluation of Speech Quality | 68 |
| 2.7.1 | Experimental Results | 71 |
| 2.8 | Non-Uniform Linear Prediction Coding of the Discrete Cosine Spectrum | 72 |
| 2.9 | Quantization and Computer Simulation Results | 74 |
| 2.9.1 | Comparison of Results | 75 |
| 2.10 | Discussion | 75 |
| CHAPTER 3 | Intermediate Binary Numbers: A New Approach to the Implementation of Signal Processors | 89 |
| 3.1. | Digital Signal Processing Using DM or DSM, A Brief Review | 91 |

| | | Page |
|---|---|---|
| 3.1.1. | Delta and Delta-Sigma Modulation | 91 |
| 3.1.2. | DM Non-Recursive Filter with Analogue Coefficients | 92 |
| 3.1.3. | Correlator Based on Delta-Sigma Modulator | 93 |
| 3.1.4. | Non-Recursive Digital Filters Implementation Using DM | 94 |
| 3.2. | Intermediate Binary Numbers | 94 |
| 3.3. | Analogue to IBN Conversion and Vice Versa | 96 |
| 3.4. | IBN Adder and Subtractor | 97 |
| 3.5. | Serial IBN Multipliers | 99 |
| 3.5.1. | IBN Multiplier with Binary Output | 101 |
| 3.5.2. | IBN Multiplier with IBN Output (Cascade IBN Multiplier) | 101 |
| 3.5.3. | Error Analysis of IBN Multipliers | 103 |
| 3.6. | Discussion | 104 |
| CHAPTER 4 | Orthogonal Transform Implementation Using Intermediate Binary Numbers | 114 |
| 4.1. | Implementation of Orthogonal Transforms, A Brief Survey | 116 |
| 4.1.1. | Implementation of the Discrete Fourier Transform | 116 |
| 4.1.2. | The Pipeline form of the FFT Processor | 118 |
| 4.1.3. | Implementation of the Walsh-Hadamard Transform | 119 |
| 4.1.4. | Implementation of Discrete Cosine Transform | 120 |
| 4.2. | Implementation of Walsh-Hadamard Transform Using Intermediate Binary Numbers | 120 |
| 4.2.1. | WHT Implementation Using the $N^2$ Computation Algorithm | 121 |
| 4.2.2. | Data-Reiteration WH Transformer | 123 |
| 4.2.3. | Walsh Function Generator | 124 |
| 4.2.4. | Pipeline Fast Walsh-Hadamard Transform | 124 |
| 4.3. | Error Analysis of Walsh-Hadamard Transformers | 126 |
| 4.3.1. | Error Analysis of the WH Transformer Using $N^2$ Computation Algorithm | 126 |
| 4.3.2. | Error Analysis of Pipeline WH Transformer | 131 |

| | | Page |
|---|---|---|
| 4.4. | WH Inverse Transform Implementation | 132 |
| 4.5. | Implementation of Discrete Cosine Transform | 134 |
| 4.6. | Error Analysis of the DCT Processor | 134 |
| 4.7. | Comparison of the Implementation Techniques | 137 |
| 4.8. | Discussion | 138 |
| CHAPTER 5 | Conclusion and Future Research | 161 |
| 5.1. | Summary of Results | 161 |
| 5.1.1. | Speech Coding Techniques | 161 |
| 5.1.2. | Implementation Techniques | 162 |
| 5.2. | Critical Discussions | 163 |
| 5.2.1. | Quantization of Parameters | 163 |
| 5.2.2. | Effects of Random Channel Error | 164 |
| 5.2.3. | Hardware Implementation of the Modelling and Quantization Part | 165 |
| 5.3. | Suggestion for Further Research | 165 |
| 5.3.1. | Voiced-Unvoiced Decision Based on Orthogonal Transformation | 165 |
| 5.3.2. | Non-Uniform Linear Prediction Vocoder | 167 |
| APPENDIX | | 169 |
| REFERENCES | | 171 |

## LIST OF FIGURES AND TABLES


Page

CHAPTER 1

Fig. 1.1. 44
Fig. 1.1. 44
Fig. 1.3. 45
Fig. 1.4. 46
Fig. 1.5. 46
Fig. 1.6. 47
Fig. 1.7. 48
Fig. 1.8. 49
Fig. 1.9. 50
Fig. 1.10. 51
Fig. 1.11. 51
Fig. 1.10. 51
Fig. 1.11. 51
Fig. 1.12. 52
Fig. 1.13. 52
Fig. 1.14. 53
Table 1.1 54

CHAPTER 2

Fig. 2.1. 78
Fig. 2.2. 78
Fig. 2.3. 79,80,81
Fig. 2.4. 82
Fig. 2.5. 83
Fig. 2.6. 84
Fig. 2.7. 85
Fig. 2.8. 86
Fig. 2.9. 86
Fig. 2.10. 87
Fig. 2.11 88

Page

CHAPTER 3


Fig. 3.1. 106
Fig. 3.2. 106
Fig. 3.3. 107
Fig. 3.4. 107
Fig. 3.5. 108
Fig. 3.6. 108
Fig. 3.7. 109
Fig. 3.8. 109
Fig. 3.9. 110
Fig. 3.10. 110
Fig. 3.11. 111
Fig. 3.12. 111
Fig. 3.13. 112
Fig. 3.14. 112
Fig. 3.15. 113


CHAPTER 4


Fig. 4.1. 140
Fig. 4.2. 141
Fig. 4.3. 141
Fig. 4.4. 142
Fig. 4.5. 142
Fig. 4.6. 143
Fig. 4.7. 143
Fig. 4.8. 144
Fig. 4.9. 145
Fig. 4.10. 146
Fig. 4.11. 147
Fig. 4.12. 147
Fig. 4.13. 148
Fig. 4.14. 148

CHAPTER 4 (continued)

Page


Fig. 4.15. 149
Fig. 4.16. 150
Fig. 4.17. 151
Fig. 4.18. 152
Fig. 4.19. 153
Fig. 4.20. 154
Fig. 4.21. 155
Fig. 4.22. 156
Fig. 4.23. 157
Fig. 4.24. 158
Fig. 4.25. 159
Table 4.1 160


CHAPTER 5


Fig. 5.1. 168

# CHAPTER I

## Introduction

In the past two decades there has been a growing interest regarding the study of digital speech communication systems. The major advantages of digital speech transmission are as follows:

(a) The transmission of digital speech is relatively insensitive to noise, crosstalk, and distortion, since digital signals, unlike analogue signals, can be easily regenerated.

(b) Switching of digital information can be simply realized using simple logic switches. This leads naturally to a fully electronic exchange where many problems of present day telephone exchanges, such as analogue crosstalk and mechanical contacts, are eliminated.

(c) Digital transmission allows time-division multiplexing to be employed. This can be accomp lished by digital circuitry, thus the necessity of complex filters which are required in frequency-division multiplexing are avoided.

(d) Digital speech can be easily scrambled for a guaranteed secret or private communication which has mainly military applications.

Normally, the digital transmission of speech requires wider bandwidth compared to the analogue case. PCM is the most common form of digitizing speech signals. The concept of PCM was first proposed by Reeves in 1938. Typically, transmission of digital speech resulting from PCM requires transmission rates of 56000 to 64000 bits per second. On the other hand, tests have shown that the human auditory system can only handle a maximum of 50 bits per second of information. Such tests show that obviously most of the transmitted bits are redundant and only little of what is transmitted is essential to understanding. The main object of speech processing is the removal

of redundancies, transmitting the remaining intelligence, and reconstructing the speech from a knowledge of what redundancies were removed at the transmitter.

In the first part of this introductory chapter a brief survey of the better known methods of digital coding of speech signals is presented. This is followed in section 1.6 by a brief introduction to new techniques of low-bit-rate speech coding and their implementation which is presented in later chapters of the thesis. A description of the thesis organisation is given in section 1.7.

## 1.1. A Brief Survey of the Important Digital Speech Coding Techniques

Many different speech coding techniques based on utilising certain properties of speech signals have been discussed by different authors. The number of existing techniques is so large that a single reference cannot cover all of them. Thus this chapter is limited to the review of only those techniques which are important from either a bit-rate or a low complexity point of view. The coding principles which represent different classes of coding techniques have priority. The emphasis has been placed on those coding principles which are related to the new coding systems described in this thesis. The chosen methods are divided into four groups where, in each group, the same attributes of speech are employed for coding.

The first group describes coding techniques that attempt to preserve the speech waveform in the time domain. The most important techniques belonging to this category are reviewed in section 1.2. In section 1.3 the most important coding techniques based on the frequency domain properties of speech are briefly reviewed. The frequency domain analysis and synthesis of speech is based on the fact that some regions of speech spectrum are perceptually more important than others. Transform domain coding techniques are considered in section 1.4. Transform domain coding is a more general form of frequency domain coding. Some orthogonal transforms are

more efficient from the hardware implementation and redundancy removal point of view than the Fourier transform and hence their use implies a move toward more efficient coding methods. The results of section 1.4 are useful in the later chapters of the thesis. Section 1.5 is devoted to the linear prediction method. Auto correlation, covariance, and partial correlation schemes are discussed in this section. Since the new coding developed in this thesis is mainly based on the principles of linear prediction approach, the concepts included in this section will be very useful indeed.

## 1.2 Time Domain Waveform Coding of Speech

The encoding of speech in this class of speech coders is based on the straight-forward reconstruction of the speech waveform in the time domain. This class is a sub-class of waveform coders which are based on the preservation of the speech waveform in any domain. The straight-forward coding of the time waveform is not a very efficient method since it does not utilize the fundamental features of speech signals. However, these types of coders offer low complexity systems for a wide range of digital speech coding applications such as those capable of reproducing sufficiently good quality for commercial telephony. The most important coding schemes belonging to this category which are discussed in the following sections are: (i) Pulse Code Modulation (PCM); (ii) Differential Pulse Code Modulation (DPCM); (iii) Adaptive Differential Pulse Code Modulation with a Fixed Predictor (ADPCM-F); (iv) Adaptive Pulse Code Modulation with a variable Predictor (ADPCM-V); (v) Delta Modulation (DM); and (vi) time-encoded speech.

### 1.2.1 Pulse Code Modulation (PCM)

In the PCM system an analogue signal $x(t)$ with band-width of W Hertz is sampled at the rate of $T^{-1}$ samples per second ($T \leq 1/2W$) where each sample is represented by a B-bits word. Thus the PCM bit rate is

$B/_T$ $bits/_{sec}$. When the samples are quantized with a fixed step size $\Delta$, the step size must be chosen so that $\Delta.2^B$ spans the maximum Peak-to-Peak range of the signal. It can be shown [1,2] that about 11 bits are required in order that the signal-to-noise ratio (SNR) be 60 dB. (This is often referred to as "toll quality" [3]). It is easily shown that the addition of one bit increases the SNR by 6dB. Thus an adequate PCM representation for most purposes requires about 88000 $bits/_{sec}$ (11 bit x 8 KHZ). For efficient transmission or storage, however, this bit-rate must be reduced significantly.

One way of reducing the overall bit rate is to use a nonlinear quantization technique. A nonlinear quantizer which has approximately a logarithmic characteristic has been found empirically nearly optimum[4]. Using a logarithmic quantizer 7 bits/sample are sufficient to obtain toll quality. An alternative approach is the use of a time varying step size [5,6,2], i.e., an adaptive quantizer. In these types of quantizers, when the signal level is low, a small step size is used, and when the amplitude is large, an appropriately large step size is employed.

### 1.2.2. Differential Pulse Code Modulation with a Fixed Predictor (DPCM-F)

Further reduction in the bit rate of PCM can be obtained if the difference between the input and a predicted signal is quantized. This scheme is based on the correlation between successive samples of time waveform of speech. If C1 is the correlation between adjacent samples and if it is greater than 0.5, then the variance of the difference between two adjacent samples, i.e. $d(n) = x(n) - x(n-1)$, is smaller than the variance of the speech signal itself[7]. As a result, it is advantageous to quantize $d(n)$ instead of $x(n)$, and use an integrator to reconstruct $x(n)$ from the quantized values of $d(n)$.

A more general DPCM-F is possible if one uses as quantizer

input a difference of the form

$$d(n) = x(n) - \sum_{i=1}^{p} a_i x(n-i) = x(n) - \hat{x}(n) \qquad (1.1)$$

where $x(n)$ is a linearly predicted version of $x(n)$. The block diagram of the DPCM-F is shown in Fig. (1.1).

It is important to note that in integrating the quantized values of $d(n)$, we do not accumulate quantization error. This is ensured by the feedback around the quantizer which results in the quantity $d(n) + E(n-1)$ to be quantized rather than $d(n)$, where $E(n-1)$ is the quantization error of the previous sample.

### 1.2.3. Adaptive Pulse Code Modulation with a Fixed Predictor (ADPCM-F)

The use of the adaptive quantizer in the DPCM-F results in ADPCM-F coder. Fig. (1.2) shows the block diagram of an ADPCM-F coder with a first order fixed predictor. One efficient method of step size adaptation for these coders was proposed by Cummiskey et al[5], based on the one word memory approach.

In a comparative study by Noll[9], the SNR of the DPCM-F and ADPCM-F over the log-PCM was shown to be 7 and 12 dB respectively.

### 1.2.4 Adaptive Differential Pulse Code Modulation with a Variable Predictor (ADPCM-V)

The ADPCM-V coder is a more sophisticated version of the ADPCM-F coder in which the prediction coefficients are adapted to the changing spectral properties of speech signals[8]. The block diagram of ADPCM-V is shown in Fig. (1.3).

To obtain the prediction coefficients, the input data is first buffered and delayed. From the buffered block, the short time

auto-correlation function is calculated and then the optimum prediction coefficients are determined according to the relationship

$$\underline{A}_{opt} = \underline{R}^{-1} \cdot \underline{r} \qquad (1.2)$$

where $A_{opt}$ is the optimum predictor vector, R and r are the auto-correlation matrix and auto-correlation vector of the data in the buffer, respectively.

The basic principle of linear prediction is discussed in some depth in section 1.5.

The ADPCM-V (with a prediction of order 12) has a gain of about 16dB over log-PCM a figure which is greater than the gain of the previous systems.

The complexity of the ADPCM-V coder is primarily dominated by the implementation of the adaptive predictor and the computation of the prediction coefficients. As seen from Fig. (1.3) this involves an auto-correlation computation and the solution of a set of linear equations in every analysis interval. This must be done using high speed digital computation. Therefore, the complexity of the ADPCM-V coder is substantially greater than that of the ADPCM-V.

### 1.2.5. Delta Modulation (DM)

Principally, in the DM encoding [9], the difference between two adjacent samples is quantized by a 1-bit quantizer. If the sampling rate is much higher than the Nyquist rate, then the correlation between adjacent samples is very close to unity and it is possible to use a 1-bit quantizer to obtain a good approximation of the input samples. When the step size of the quantizer is

fixed, the 1-bit can only indicate the sign of the difference signal. The fixed step size DM is known as Linear DM (LDM).

The LDM introduces two types of error: slope overload distortion and granular noise[10]. Slope overload occurs when the step size $\Delta$ is too small to follow a steep segment of the input waveform. On the other hand, granularity refers to a situation where the staircase function oscillates about the slowly varying parts of the waveform. It should therefore be possible to change the step size $\Delta$ to provide a minimum error power.

One solution to this problem is to let the step size vary so that $\Delta$ becomes large during slope overload and small during granular distortion. The problem in ADM, therefore, is to specify suitable rules for step size variation. Various methods of step size adaptation has been discussed by Steele[9]. In a method proposed by Jayant[6] the step size $\Delta$ is multiplied by a constant $P(P > 1)$ during overload and by $Q(Q < 1)$ during granularity. The slope overload is characterised by the pattern of the output binary sequence. For example a run of +1's or -1's means slope overload, while an alternating pattern means granularity. An optimum choice of the parameters P and Q is[6] $P=1.5$ and $Q=1/P$.

### 1.2.6. Time-Encoded Speech

The time-encoded coding method[11] is a new method of speech coding in the time domain. The main difference between this method and those discussed previously is that this method codes the shape of the waveform rather than the instantaneous signal values. In this approach, the time waveform of speech is divided into short time segments. The possible waveform shapes for each time duration are statistically derived. If a specific waveform shape corresponding to a specific time duration is represented by a symbol, then each symbol can be considered as an element of a two dimen ional array, where one axis of this array shows the number of possible time durations and the other axis shows the number of possible

shapes for each time duration. It has been shown[11] that a total of 23 symbols suffice for the reproduction of intelligible speech with speaker identification. In such a system the bit-rate necessary for reproducing intelligible speech of both female and male speakers has been reported to be 5.4 K bits/sec while 4.5 Kbits/sec is still satisfactory for the reproduction of male speech.

The time segmentation in this method is based on breaking the speech waveform into time segments defined by successive zero crossings of the time axis. It is, of course, conceivable that other segmentation procedures may be employed successfully.

## 1.3 Frequency Domain Coding of Speech

Frequency domain processing of speech has been traditionally one of the most important speech processing techniques. The most important statistical attribute of speech in the frequency domain is the unequal contriubtion of different regions of frequency spectrum to speech perception. The low frequency regions statistically contribute more towards perception than the high frequency regions. This phenomenon is known as the articulation index[15]. The other characteristic feature of the frequency spectrum of speech is the importance of the positions of the sharp peaks in the power spectrum. These peaks are called formants and correspond to the poles in the transfer function of the vocal tract. Various coding schemes in the frequency domain are based on preserving, with varying degrees of accuracy, the perceptually important parts of speech spectrum.

In the following sections some typical examples of frequency domain coding techniques are discussed. These are (i) the sub-band coding (SBC), (ii) formant vocoder, and (iii) homomorphic vocoder. These coders were chosen because they are representative of a number of different classes of coding techniques ranging from relatively simple and consequently inexpensive to highly complex and therefore costly schemes.

### 1.3.1 Sub-Band Coding (SBC)

The basis for coding speech signals in sub-bands[12-14] is based upon the experimental evidence that the contributions to the relative perceptual importance of various frequency bands are not equal, and hence the quality of the coded signal can be significantly improved by the distribution of quantizing noise across the speech signal spectrum. In a method discussed by Crochiere[12], the speech band is partitioned into 4 sub-bands by bandpass filtering. Each sub-band is low-pass translated, sampled at its Nyquist rate, and digitally encoded. The selection of the appropriate sub-bands is guided by the perceptual data contained in the so-called articulation index (AI)[15]. The articulation index denotes the contribution, on the average, of parts of the spectrum to the overall perception of the spoken sounds. Table 1.1 shows the articulation bands in which the frequency range 200-3200 Hz has been partitioned into 16 bands that contribute equally to the AI. The number of bits allocated to each sub-band is based on its perceptual importance, thus more bits are allocated to the sub-bands of lower frequencies.

On reconstruction, sub-band signals are decoded and bandpass translated back to their original bands. The sum of the outputs of the bandpass filters gives an approximation of the original speech signal.

The transmission bit-rate of the sub-band coder can be lowered successfully by limiting the width of the sub-bands and by tolerating some spectral gaps that do not contribute significantly to the AI. The sub-band coder can produce intelligible speech at bit-rate of approximately 7.2 K bits/sec. It is still possible to lower the bit-rate to 4.8 and still maintain the same speech quality by varying the sub-bands in accordance with formant frequencies[14].

The sub-band coding technique offers an attractive technique for the economical coding of speech in low bit-rates.

### 1.3.2 Formant Vocoder

In general, vocoders transmit two types of information about the speech signals. The first is the information about the shape of the frequency spectrum. The second type is the information about the excitation source, i.e. pitch period and voicing information. In the formant vocoder[16-22], instead of attempting to preserve every detail of the frequency spectrum, the position of formants plus, in some cases, the intensities of the formants are transmitted. The siutation can be further simplified since it is possible, to a large extent, to predict the intensity information from a knowledge of the formant frequencies[18]. As regards the speech excitation, a decision is made on whether the speech signal is voiced or unvoiced and if it is voiced, the pitch period is measured. The quantized duration of pitch together with one bit for voiced-unvoiced decision is the information transmitted about the excitation waveform. For most practical purposes, a minimum number of three formants is required for satisfactory representation of speech spectrum.

The most direct method of identifying the formant frequencies is the use of a large array of channel filters where the formants are the peak frequencies at which the filter output is the highest[16,17]. In modern formant vocoders the tendency is to use a digital computer for extraction of the formant data. Formant information can be obtained by a bank of digital filters or by the method of discrete Fourier transform, followed by a peak picking procedure[19]. Homomorphic method is another approach which can also be applied. This approach will be discussed in the next section. In addition to the above, the method of linear inverse filtering can also be used for accurate estimation of the formant frequencies[20,21].

### 1.3.3 Homomorphic Vocoder

Homomorphic filtering belongs to a class of non-linear signal processing techniques which are based on a generalisation of the

principle of superposition that defines linear systems. Such techniques have been applied in seperating signals that have been combined by multiplication and correlation[23,24]. The application of these techniques to speech processing is based on the assumption that speech production can be viewed as the convolution of an excitation function with vocal tract impulse response, i.e.

$$s(t) = e(t) \otimes v(t) \qquad (1.3)$$

where $s(t)$ is the speech signal, $e(t)$ is the excitation signal, and $v(t)$ is the vocal tract impulse response. Eq. (1.3) in the frequency domain can be written as

$$S(\omega) = E(\omega).V(\omega) \qquad (1.4)$$

Taking logarithm of both sides, the expression

$$\ln S(\omega) = \ln E(\omega) + \ln V(\omega) \qquad (1.5)$$

is obtained, and on taking the inverse Fourier transform ($F^{-1}$), we have

$$F^{-1}\left[\ln S(\omega)\right] = F^{-1}\left[\ln E(\omega)\right] + F^{-1}\left[\ln V(\omega)\right] \qquad (1.6)$$

The $F^{-1}\left[\ln S(\omega)\right]$ is called the cepstrum of the input signal $s(t)$. The part due to the excitation source i.e. the cepstrum $F^{-1}\left[\ln E(\omega)\right]$ appears as a periodic pulse train from which the pitch period can be easily determined.

The cepstrum contains all the information in the spectrum of a signal and can be viewed as another representation of the signal. Cepstrum approach has been applied in a speech analysis synthesis

scheme called homomorphic vocoder[25,26]. In this system, the low rate samples of cepstrum values and an estimate of pitch period serve as a representation of speech signal from which an acoustic wave can be reconstructed.

The basic block diagram of the homomorphic speech processor is shown in Fig. (1.4).

A comprehensive review of the homomorphic and the formant vocoder is given by Flanagan[27] and Holmes[28].

## 1.4 Transform Domain Coding of Speech

Transform domain representation is a generalized version of representing a signal in the frequency domain. The representation of speech in the transform domain also has the advantage of uneven distribution of information along the transform spectrum. Thus an efficient transform coding method should be based on allocating more information (bits) to those parts of the spectrum with greater importance. The reason that other transforms are preferred to the Fourier transform is to some extent the inefficiency of the discrete Fourier transform for software and hardware implementations requiring complex arithmetic operations. In the following sections, a few important orthogonal transforms are reviewed and three major transform coding techniques, namely, the transform coding with fixed bit allocation, adaptive transform coding, and transform coding of dominant coefficients are discussed.

### 1.4.1 Walsh-Hadamard Transform

The incomplete set of Rademacher functions[29] was completed by Walsh in 1923 to form the complete orthogonal set of rectangular functions, known as Walsh functions[30]. The Rademacher functions developed in 1922 are a train of rectangular pulses with $2^{m-1}$ cycles

in the half open interval [0,1] , taking values +1 or -1 as shown in Fig. (1.5). The set of Walsh functions is generally classified into three groups. These groups differ from one another in that the order in which individual functions appear is different. The three types of ordering are:(1) sequency or Walsh ordering, (2) dyadic or Paley ordering, and (3) natural of Hadamard ordering. Sampling of Walsh functions results in an (NXN) matrix, thus three types of matrices, different in ordering, would be obtained. These matrices are generally called Hadamard Matrices, since they can be obtained by re-ordering the rows of the so called Hadamard matrices. Fig. (1.6-8)shows the first 8 Walsh functions of each group with their corresponding Hadamard matrices. The Hadamard-ordered Hadamard matrices have an interesting characteristic that each matrix of order 2N from this group can be obtained recursively by a Kronecker product defined as

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \qquad (1.7)$$

The Walsh-Hadamard transform[31] (WHT) can be defined using matrix notation as follows. Let X denote an N-vector with elements X(m), m=0,...,N-1, of finite valued real numbers, then the WHT of X is defined as

$$Y = \frac{1}{N} H_N \cdot X \qquad (1.8)$$

where $H_N$ is the Hadamard matrix of order N and Y represents the

WHT vecotor of $\hat{X}$. Since three different groups of Hadamard matrices exist, there are defined three types of WHT, namely, Walsh-ordered WHT$(WHT)_w$, Paley-ordered WHT$(WHT)_p$, and Hadamard-ordered WHT$(WHT)_h$.

The direct computation of (1.8) needs $N^2$ operations (of addition or subraction) to be done while it can be done by $N \log_2 N$ number of the same kind of operations when a fast WHT algorithm is used[32]. Numerous fast WHT algorithms have been proposed, based on using different methods of matrix factoring and partitioning[31-35]. One algorithm proposed by Shanks[33], is illustrated in Fig. (1.9). This algorithm is similar to the Cooley-Tukey algorithm[36], used for the fast computation of the discrete Fourier transform. The Shanks algorithm can only compute the $(WHT)_h$. The fast computation of $(WHT)_w$ by Cooley-Tukey type has been introduced by Manz[35] and is shown in [31]. This algorithm has two more steps than the Shanks algorithm. The first step is to bit-reverse the input and order it in ascending index order. The second step is to define an "operation reversal" in which the positions of the additions and subtractions are interchanged, as shown in [31].

## 1.4.2 Discrete Cosine Transform

The elements of the Discrete Cosine Transform (DCT) matrix of order N denoted by T(i,j) are defined as

$$T(0,j) = \frac{1}{N} \qquad j = 0,1,2,\ldots,N-1$$

$$T(i,j) = \frac{1}{N} \cos \frac{(2j+1)i\pi}{2N} \qquad \begin{matrix} i = 1,2,\ldots,N-1 \\ j = 0,1,2,\ldots,N-1 \end{matrix} \tag{1.9}$$

and the elements of the inverse DCT matrix or order N denoted

by Q(i,j) are similarly defined as:

$$Q(i,0) = 1 \qquad i = 0,1,2,\ldots,N-1 \qquad (1.10)$$

$$Q(j,i) = 2\cos\frac{(2j+1)i\pi}{2N} \qquad \begin{matrix} i = 0,1,2,\ldots,N-1 \\ j = 1,2,\ldots,N-1 \end{matrix} \qquad (1.11)$$

The DCT and IDCT matrices defined here are slightly different from the definition adopted by Ahmed et al[37]. The difference is that the defined transforms are weighted by a constant factor so that no overflow occurs when this transformation is implemented by digital hardware.

The DCT of the N-vector X denoted by N-vector Y is defined as follows:

$$Y = T.X \qquad (1.12)$$

while the inverse transform is given by

$$X = Q.Y \qquad (1.13)$$

One method for fast computation of the DCT is using a 2N-point FFT, where also a 2N-point FFT can be used to obtain the IDCT coefficients[37]. Further computation reduction can be achieved by using two N point FFT rather than one 2N-point[38]. The recent algorithm, proposed by Chen[39], enables using only real number computation and can significantly reduce the computation time. This algorithm is shown in Fig. (1.9).

The important property that makes the DCT highly efficient for speech coding is that the basis vectors of DCT closely approximate the eigenvectors of a class of matrices called Toeplitz matrices[40] and it is important to note that the long term correlation matrix of speech is also a Toeplitz matrix.

### 1.4.3. Karhunen-Loeve Transform

The Karhunen-Loeve Transform (KLT)[31] is a signal dependent transform and has the unique property that the transform coefficients are uncorrelated, i.e. the covariance matrix in the KLT domain is diagonal. Let $R_{xx}$ be the covariance of the ensemble input signal x and $\lambda_i$ be its eigenvalue and $\phi_i$ be its eigenvector. The set of vectors $\phi_i$ comprise a set of orthogonal functions called the KL functions. If the covariance matrix of the transformed signal is $R_{yy}$, then

$$R_{yy} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N) \qquad (1.14)$$

where $\lambda_i$ values are the variance of transformed signal and also they are the eigenvalues of $R_{xx}$. The following relation exists between $R_{xx}$, $R_{yy}$ and $\lambda_i$ [40]

$$\det R_{xx} = \det R_{yy} = \prod_{i=1}^{N} \lambda_i \qquad (1.15)$$

It should be noted that for any other transform, the relation between $R_{xx}$ and $R_{yy}$ and coefficient variances ($\sigma_i^2$) is,

$$\det R_{xx} = \det R_{yy} < \prod_{i=1}^{N} \sigma_i^2 \qquad (1.16)$$

comparison of (1.15) and (1.16) shows that the multiplication of

KL coefficient variances is minimum. This makes the KLT result in a minimum distortion when used for non-uniform bit allocation coding techniques. The main disadvantage of using KLT is the lack of a fast algorithm for the computation of the transform coefficients, in addition, the matrix is signal dependent and the computation of its elements is not simple.

### 1.4.4 Transform Coding with Fixed Bit Allocation (TCFBA)

The variance of the transform coefficients of speech signals in general differs significantly over the transform domain. This property can be utilized for bit-rate reduction by allocating more bits to the coefficients with larger variance and less bits to those coefficients of smaller variance. Let us assume that the distortion due to quantization of coefficients $Y_i$ of variance $\sigma_i^2$ must not to exceed $D_i$, then the minimum number of bits $R_i$ necessary for quantizing $Y_i$ is given by[41]

$$R_i = \delta + \frac{1}{2} \log_2 \frac{\sigma_i^2}{D_i} \qquad i = 1,2,\ldots,N \qquad (1.17)$$

where $\delta$ is a constant value which depends on the type of quantizer employed and which also depends on the probability density function of the signal to be quantized. If $\overline{D}$ is the average distortion and $\overline{R}$ is the average bit-rate then the optimum $R_i$ is given by[42]

$$R_i = \overline{R} + \frac{1}{2} \log_2 \frac{\sigma_i^2}{\left[\prod_{i=1}^{N} \sigma_i^2\right]^{1/N}} \qquad (1.18)$$

The above bit assignment calculation may result in a negative number of bits for those coefficients with very small variance. The negative number of bits should be set to zero and the bit assignment must be re-optimized. With an optimum bit assignment, the average

distortion will satisfy the following relation[43,44]

$$\bar{D} = 2^{2\delta} \cdot 2^{-2\bar{R}} \left[ \prod_{i=1}^{N} \sigma_i^2 \right]^{1/N} \qquad (1.19)$$

Eq. (1.19) shows that the minimum distortion depends on variance distribution of coefficients.

In the time domain, the variance of speech samples tends to be constant, thus the samples must be quantized with an equal number of bits. If the variance of speech samples in the time domain is given by $\sigma^2$, the average distortion of a pulse-code-modulation (PCM) scheme is given by[44]

$$\bar{D}_{PCM} = 2^{2\delta} \cdot 2^{-2\bar{R}} \cdot \sigma^2 \qquad (1.20)$$

Comparing (1.19) with (1.20) the gain of the transform coding over PCM is obtained as

$$G_{TC} = \frac{\bar{D}_{PCM}}{\bar{D}} = \frac{\sigma^2}{\left[ \prod_{i=1}^{N} \sigma_i^2 \right]^{1/N}} \qquad (1.21)$$

This gain has been obtained for several transformations by Zelinski and Noll[44] and the results are shown in Fig. (1.10). These results are in agreement with those given by Companella and Robinson[45] and Gethoffer[46,47]. The optimum gain belongs to KLT, since the product of variances in this case corresponds to the product of the eigenvalues which according to ( 1.15 ) is minimum.

According to Fig. (1.10) the performance of DCT is almost as

good as that of KLT, the difference is being smaller than 0.5 dB. The TCFBA technique can improve the SNR by a maximum value of 10 dB (reference log-PCM). Better results can be obtained if the step-size is changed with the short-time statistics of speech signals[48,44]. One method of achieving this, is to estimate the rms level $\sigma^2$ of the specific data block. The rms value of the data block then adjusts the step-size of quantizer. This method can yield an additional gain of approximately 4 dB over TCFBA system with a fixed step-size. The structure of a TCFBA with an adaptive quantizer is shown in Fig. (1.11).

### 1.4.5 Adaptive Transform Coding (ATC)

The inefficiency of the TCFBA system is due to the fact that the coefficients' amplitude may differ significantly from their expected values. The bit assignment in the ATC technique varies from block to block and this variation follows the change in the short-term variance distribution. In the ATC scheme proposed by Zelinski and Noll[44], the estimate of the variance distribution for each block is obtained by a straight line interpolation method. The transform spectrum is first divided into M equal regions. The mean squared value of the coefficients in each region is then determined. The quantized form of these values comprise the side information that has to be transmitted to the decoder in addition to the quantized transform coefficients. The estimate of the variance distribution is obtained by enveloping these samples by a straight-line interpolation. The bit assignment in both coder and decoder is based on this estimate of the variance distribution. Fig. (1.12) shows the block diagram of the adaptive transform coder.

Experimental results of Zelinsky and Noll show that the ATC system has 17-23 dB gain over the log-PCM. This gain is more than that of non-pitch tracking systems discussed earlier. The degree of the complexity of the ATC system is rather high and is approximately

on the same order as the ADPCM-V coder. Moreover the side information and bit assignment computation require high speed digital computation.

### 1.4.6 Transform Coding of Dominant Coefficients (TCDC)

Another approach of transform coding is to quantize and transmit the n most significant coefficients of each transform block. This method has been discussed by Anderson and Huang[49] and Wintz[43] for picture coding and by Gethoffer[47] and Shum et al[50] for speech coding. In this method, since the positions of dominant coefficients change from block to block as is evident from the short time speech signal statistics, additional bits are necessary to specify the sequencies of the n retained coefficients. The addressing bits substantially increase the bit rate, thus the number n has to be small in low bit rates coding. In the method of Shum et al, the dominant coefficients have been chosen from a 64 point block, which necessitates 6 addressing bits to be transmitted for each dominant coefficient. A partitioning scheme similar to that used by Peled and Liu[51] can be applied for reducing the addressing bit-rate. As a simple example, if we choose two dominant coefficients out of a block of 64 coefficients, the total addressing bits will be 12 (each coefficient can have one of the possible 64 positions). If the 64 point block is partitioned into two 32 point blocks and one dominant coefficient is chosen from each block, then the total number of addressing bits will be reduced to 10 bits. Substantial reduction in the addressing bit-rate can be achieved,in trade off with acceptance of some more distortion,if the transform block is partitioned into blocks each having a size of integer power of 2. However, the addressing bit-rate remains considerably high.

In a recent approach proposed by Ahmadi and Constantinides[52], the static variance distribution of transformed speech has been employed together with the dominant coefficients technique. In this method, the base-band coefficients,which statistically are the most

important part of the spectrum, are directly transmitted. The sub-bands are divided into two regions and each region is represented by one dominant coefficient. This approach has some advantages over the TCDC method in both bit-rate reduction and reducing the complexity of the system.

## 1.5 Linear Prediction Coding of Speech (All-Pole Modelling)

Linear Prediction (LP) is one of the most efficient techniques of speech coding. The importance of this method is due to its accuracy and computation speed.

The various methods of LP have in common the assumption that a speech sample s(nT), where n is an integer variable and T is the sampling interval, can be predicted approximately from a linearly weighted summation of a number of immediately preceeding samples. Let this approximate value of s(nT) be $\hat{s}$ (nT), given by

$$\hat{s}(n) = -\sum_{k=1}^{p} a_k s(n-k) \qquad (1.22)$$

where s(n) and $\hat{s}$ (n) represent s(nT) and $\hat{s}$ (nT), respectively. The terms $a_i, i=1,2,\ldots,p$ are called the prediction coefficients. These coefficients are found in such a way that the error between the actual value and the predicted value over the desired range of samples is minimum. Let the error between the actual value and the predicted value be given by e(n), where

$$e(n) = s(n) - \hat{s}(n) = s(n) + \sum_{k=1}^{P} a_k s(n-k) \qquad (1.23)$$

In general the parameters $a_k$ are obtained as a result of the minimization of the mean or total squared error with respect to each of these parameters.

The total squared error over a range of samples are denoted by E, where

$$E = \sum_{n} e^2(n) = \sum_{n} \left[ s(n) + \sum_{k=1}^{p} a_k s(n-k) \right]^2 \quad (1.24)$$

It is in the range over which the summation in (1.24) applies and the definition of the signal $s(n)$ in that range that the auto-correlation[20] and covariance[53] methods are different.

The total squared error E can be minimized by setting the partial derivatives of E with respect to all $a_k$'s, $1 \leqslant k \leqslant p$, to zero.

$$\frac{\partial E}{\partial a_k} = 0 \qquad 1 \leqslant k \leqslant p \qquad (1.25)$$

From (1.24) and (1.25) we obtain the set of equations

$$\sum_{k=1}^{p} a_k \sum_{n} s(n-k)s(n-i) = - \sum_{n} s(n)s(n-i) \quad (1.26)$$

Using either covariance or autocorrelation method Eq. (1.26) forms a set of p equations with p unknowns which can be solved to yield the predictor coefficients.

The minimum total squared error $E_p$ can be obtained by making use of (1.26) in (1.24). The result can be shown to be

$$E_p = \sum_{n} s^2(n) + \sum_{k=1}^{p} a_k \sum_{n} s(n)s(n-k) \qquad (1.27)$$

### 1.5.1 Covariance and Autocorrelation Approach

In the covariance method, the error in (1.24) is minimized over a finite interval, such as (0 to N-1 samples). Equations (1.26) and (1.27) then reduce to

$$\sum_{k=1}^{p} a_k \varphi(ki) = -\varphi(0i) \qquad (1.28)$$

$$E_p = \varphi(00) + \sum_{k=1}^{p} a_k \varphi(0k) \qquad (1.29)$$

where

$$\varphi(ik) = \sum_{n=0}^{N-1} s(n-i)s(n-k) \qquad (1.30)$$

The coefficients $\varphi(ki)$ in (1.28) form a covariance matrix, and consequently this method is referred to as the covariance method.

In the autocorrelation method, the error in (1.24) is minimized over the infinite duration $(-\infty, +\infty)$, thus the signal should be windowed before analysis. Equations (1.27) and (1.26) are then reduced to

$$\sum_{k=1}^{p} a_k R(i-k) = -R(i) \qquad (1.31)$$

$$E_p = R(0) + \sum_{k=1}^{p} a_k R(k) \qquad (1.32)$$

where

$$R(i) = \sum_{n=-\infty}^{\infty} s(n)\, s(n-i) \qquad (1.33)$$

is the autocorrelation function of the signal s(n). The matrix formed from R(i-k) is known as autocorrelation matrix, thus this method is called the autocorrelation method.

The main difference between the auto-correlation and covariance matrices is that in the auto-correlation matrix, the elements along lines parallel to the diagonal are equal while in the covariance matrix these elements are not normally so. The autocorrelation method has the advantage over the covariance method in the amount of storage and computation time necessary for solving the system of equations to obtain prediction coefficients. However, the covariance method is preferred when the analysis interval is relatively short, since in this case the covariance method leads to less mean-squared-error compared to the autocorrelation method. The covariance method reduces to the autocorrelation method as the length of the analysis interval tends to infinity.

### 1.5.2 Partial Correlation Approach (PARCOR)

Principally, the all-pole LP modelling object is to predict a sample from its past samples so that the error of the prediction is a white uncorrelated signal. Thus the prediction coefficients should be computed from the short term correlation characteristics in such a way that the error signal to be as uncorrelated as possible. In the autocorrelation or covariance method, a system of equations is simultaneously solved to yield the prediction coefficients.

In the PARCOR approach proposed by Itakura and Saito[54], the correlation is removed stage by stage. In the first stage the

correlation between adjacent samples is removed and in the second stage the correlation between every other sample is removed and so on. Thus after a number of stages, say 12, the residue signal related to speech will have a nearly white spectrum. The structure of a PARCOR analyser is shown in Fig. (1.13). This figure shows that after each stage we have two types of residues, namely, the forward prediction residue and the backward prediction residue. If the forward residue of the m-1th stage is shown by $x^{+}_{m-1}(n)$ and similarly, the backward residue by $x^{-}_{m-1}(n)$, then the forward and the backward prediction residues at the mth stage are[55]

$$x^{+}_{m}(n) = x^{+}_{m-1}(n) + K_m x^{-}_{m-1}(n) \qquad (1.34)$$

$$x^{-}_{m}(n) = K_m x^{+}_{m-1}(n-1) + x^{-}_{m-1}(n-1) \qquad (1.35)$$

where the initial conditions are given by

$$x^{+}_{0}(n) = x(n) \quad \& \quad x^{-}_{0}(n) = x(n-1) \qquad (1.36)$$

The coefficients Km, called by Itakura and Saito the partial correlation coefficients, are derived by calculating the correlation coefficients between the forward and the backward prediction residues

$$K_m = -\frac{\sum_{n=-\infty}^{\infty} x^{+}_{m-1}(n) \cdot x^{-}_{m-1}(n)}{\left\{\sum_{n=-\infty}^{\infty}\left[x^{+}_{m-1}(n)\right]^2 \sum_{n=-\infty}^{\infty}\left[x^{-}_{m-1}(n)\right]^2\right\}^{1/2}} \qquad (1.37)$$

A one-to-one relationship between the prediction coefficients and the PARCOR coefficients exists which enables one set of coefficients to be converted to another set. This relationship

is given below

$$a_{m+1,0} = 1 \qquad (1.38)$$

$$a_{m+1,i} = a_{mi} + K_{m+1}\, b_{mi} \qquad (1.39)$$

$$a_{m+1,m+1} = K_{m+1} \qquad (1.40)$$

where $a_{mi}$ is the ith prediction coefficient of a prediction with order m and

$$b_{mi} = a_{m,m+1-i} \qquad (1.41)$$

Although the PARCOR coefficients, which are also known as reflection coefficients, provide the same amount of information as the prediction coefficients, their properties under computation and quantization are different. It has been shown[56] that the PARCOR coefficients are the best set to be used for transmission.

## 1.5.3 Linear Prediction Vocoder

In the LP vocoder, originated by Atal and Hanaver[53], the speech signals are modelled by three sets of time varying parameters, (i) the prediction coefficients or PARCOR coefficients, (ii) the voicing decision and the pitch period, (iii) the speech energy. The block diagram of a LP vocoder is shown in Fig. (1.14). The vocoder is divided into two sections, analyser and synthesizer. The analyser extracts the pitch period, prediction coefficients, and the voicing information from the speech waveform. The pitch period in this

case can be derived by peak picking the prediction error signal[53].

The synthesizer reconstructs a synthetic speech signal by processing an excitation signal through a linear recursive all-pole filter. The coefficients of this filter are the prediction coefficients after quantization. Similarly, a lattice form recursive filter can be employed where its coefficients are the PARCOR coefficients. The main advantage of the lattice form is its stability which can be ensured simply by bounding its coefficients to be within the range $\pm 1$. Depending on the voicing decision, the input to the synthesizer filter is either a sequence of white samples or the pitch pulses. The spacing of the pitch pulses corresponds to pitch intervals of the original speech wave. The gain of the excitation signal is calculated by requiring that the synthetic speech energy be equal to the energy of the original speech wave.

Although the LP method is intrinsically suitable for speech analysis and synthesis, nevertheless, the system is far more complicated than waveform coding systems. Besides, the all-pole LP modelling is not always suitable for modelling of all types of speech signals. It is well known that for non-nasal voiced speech sounds the transfer function of the vocal tract has no zeros, but the transfer function of the vocal tract for unvoiced and nasal sounds usually includes zeros as well as poles. One method of taking account of zeros is by considering a large number of poles. But the extra poles cause the order of the filter to be high and consequently more parameters have to be transmitted which consequently diminishes the capability of the all-pole modelling for data compression applications. For further data compression, it is useful to be able to model the speech spectrum in terms of poles and zeros. Much effort has currently been made on the problem of pole-zero modelling[57-59]. The main difficulty is that the pole-zero problem is non-linear, even though it is possible to reduce the problem to a system of linear equations but the method still remains far more complicated and more expensive compared to the all-pole LP technique.

Linear prediction with all-pole modelling also requires pitch detection in the analyser and excitation sources in the synthesizer. The pitch detection by itself is an elaborate process and is also unreliable[60]. Even pitch-synchronous LP method results in a poor quality speech in the synthesizer for low pitch speakers[61]. The explanation is concerned with the fact that the LP parameters used in the synthesis allow pitch synchronous resetting of the synthesizer. Thus the higher the pitch, the more frequently the LP parameters are updated. Conversely, the lower the pitch, the less frequently the LP parameters are changed and the longer time the structure of the synthesizer is left unchanged. Since speech is really a non-stationary process it may be conjectured that the ear somehow senses that the LP parameters are being held too long, and interpret this excessive stationarity as a buzzy overtone[61]. Recently, methods for reducing the buzzy effect has been proposed[61,62] in which a non-impulsive source for exciting the LP synthesizer during voiced sounds, has been employed.

The above discussion shows that the main drawback to the all-pole LP modelling is the complexity which results from the computation of the auto-correlation or covariance matrix and the solution of a system of equations, and also pitch detection. Some methods for improving the performance of the all-pole LP modelling were discussed which in fact are a trade-off between quality of performance and complexity.

## 1.6 Non-Uniform Linear Prediction and Intermediate Binary Numbers: New Techniques for Design of an Efficient Speech Coding System[94,95]

Various digital coding techniques are generally judged by two factors, (i) efficiency in reducing the bit rate at a fixed specified speech quality, (ii) degree of complexity. Therefore research in this field can be generally divided into two categories. One is the search for an efficient method of speech modelling and quantization. The other category is concerned with seeking an

efficient implementation technique which can reduce significantly the necessary hardware compared to the conventional approaches.

Numerous methods of modelling and quantization have been proposed which vary in terms of bit-rate, efficiency and complexity, some of which were reviewed in this chapter. The choice of an appropriate method is generally based on a compromise between low bit-rate and low complexity. For a very low-bit-rate requirement, vocoding methods are the obvious choice; however, the majority of vocoding methods require the seperation of the excitation function and the vocal tract transfer function which involves a high system complexity. The waveform coding systems such as adaptive delta-modulator and adaptive differential PCM are extremely simple to implement, but, due to their primitive natures the quality of reproduced speech at very low-bit-rates, leaves much to be desired.

The main objective of research reported in this thesis in the area of speech modelling and quantization, was to develop a new waveform coding technique with a complexity comparable to that of a relatively simple waveform coder where its transmission bit-rate is in the range that can be handled by commercially available modems. The brief review given in the previous sections reveals the fact that most of the useful coding systems are those based upon the principle of linear prediction. On the other hand, the features of speech signals in a transform domain can be successfully employed for low-bit-rate coding of speech.

Based on this knowledge the search for a new speech modelling technique in this thesis has been directed towards the area of linear prediction and transform coding. A new coding technique based on linear prediction principle has been developed for efficient coding of speech in a transform domain. In this method, a few base-band coefficients of the transform spectrum are transmitted where the rest of the spectrum is modelled by a few prediction coefficients. The computation of prediction coefficients

is based on a new technique of linear prediction which can utilise the correlation and non-uniform variance distribution of transformed speech signals at the same time. The new coding method has been employed for the coding of the Walsh-Hadamard(WH) and Discrete Cosine(DC) spectrum. For a further quality improvement, a new technique of voiced-unvoiced detection based on the different variance distribution of the transformed voiced and unvoiced sounds has been developed and a different coding of the unvoiced sounds is described.

The proposed method may be digitally implemented by a conventional approach, i.e. using A/D converter and binary arithmetic units, nevertheless, thenew implementation approach, proposed in this thesis,is based on a new system of binary numbers referred to as Intermediate Binary Numbers (IBN). The main advantages of using IBN over that of the conventional binary system is the simpler analogue to digital conversion and simpler and more modular arithmetic units required in hardware realizations. The analogue-to-IBN and IBN-to-analogue conversion is described and IBN arithmetic operations are defined and also appropriate hardware designs of the arithmetic units are proposed. The IBN arithmetic units which are described in Chapter 3, have some fundamental differences with the conventional binary arithmetic units and these result in more efficient and modular hardware realisations. The detailed design of real time WHT, fast WHT, and DC transformer, which comprise the major part of the proposed system hardware,are examined in Chapter 4.

## 1.7 Organisation of the Thesis

The thesis consists of five chapters. The first chapter is an introduction to digital speech coding and a brief survey of important coding techniques. Chapters 2 - 4 contain a detailed presentation of the research reported in this thesis. In Chapter 2 the new coding technique is described. Chapter 3 discusses,

a new implementation technique for digital processing of low-bandwidth signals and in Chapter 4 this technique is employed for implementation of the WHT and DCT.

The new coding technique described in Chapter 2 is a combination of linear prediction and transform coding. Before describing the new techniuqe, the reasons why the conventional linear prediction method (i.e. all-pole modelling approach) is inefficient in a transform domain are discussed. Then the description of the new technique of linear prediction which can be matched with the characteristics of speech in a transform domain, follows. This technique is employed in the WH and DC domain. These transformations have been chosen since the former is the most efficient transform from the implementation point of view and the latter is an efficient signal independent transform for reducing the bit-rate. A new voiced-unvoiced decision is introduced, based on the different variance distributions of these sounds in transform domain and different coding procedures for unvoiced sounds are considered. The quality of decoded speech is evaluated based on recent methods of objective tests of quality measurement. These types of testing have many advantages over the subjective tests since they use computational rather than human assessment procedures. Amongst the advantages are: reduced testing time, less cost, independence of language and independence of human factors. Moreover, the results of these types of testing can lead us to development of more efficient speech coding criteria. Chapter 2 terminates with a comparison of the new technique with the best non-pitch tracking coding systems which demonstrate the capability of the new technique in achieving superior results in low bit-rates at considerably lower complexity.

In Chapter 3 a new system of binary numbers is defined. These numbers have proved to be more efficient than the conventional binary numbers for the implementation of digital processors of signals with low-bandwidth such as speech signals. The efficiency

is due to simpler conversion from and to the analogue form and simpler arithmetic units. The conversion employes linear delta encoding systems which are significantly simpler than A/D convertors. The new arithmetic units described in this chapter are not only significantly simpler than the conventional binary arithmetic units but also result in more modular hardware. The hardware realisations of arithmetic units are proposed and the pertinent and important finite register length error analysis is carried out.

Chapter 4 discusses the implementation of the WHT and the DCT employing the techniques of Chapter 3. Three different methods of WHT implementation are discussed, where two of them employ the straightforward $N^2$ computation criterion while the third one is based on the fast WHT algorithm. The finite register length error analysis for all the cases is also carried out. The hardware implementation of DCT is also described, and simplified relations for computation of finite register length error are derived.

In the final chapter (Chapter 5) an attempt is made to assess critically the results of this period of work and suggestions for further research are made.

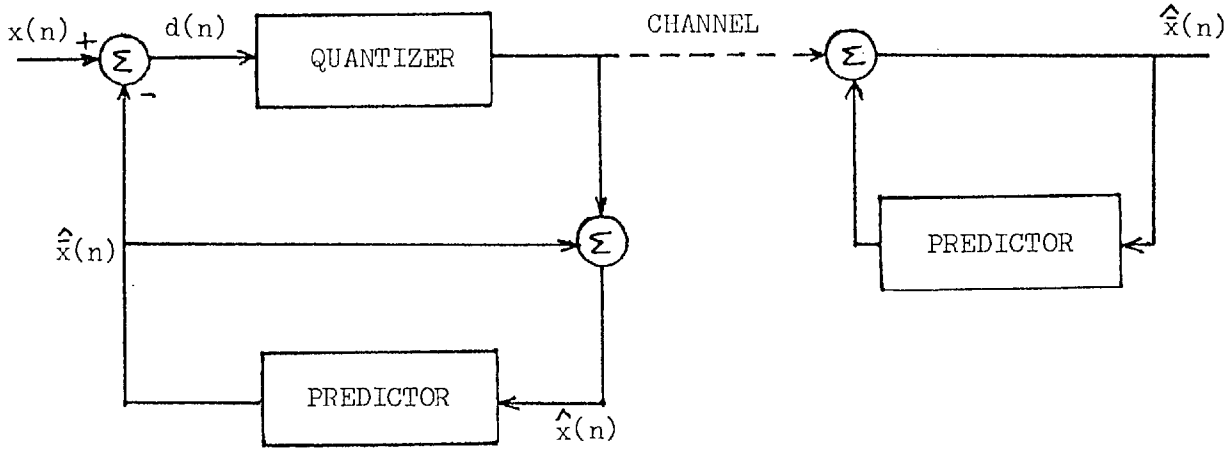



Fig.1.1. Block diagram of DPCM-F coder and decoder

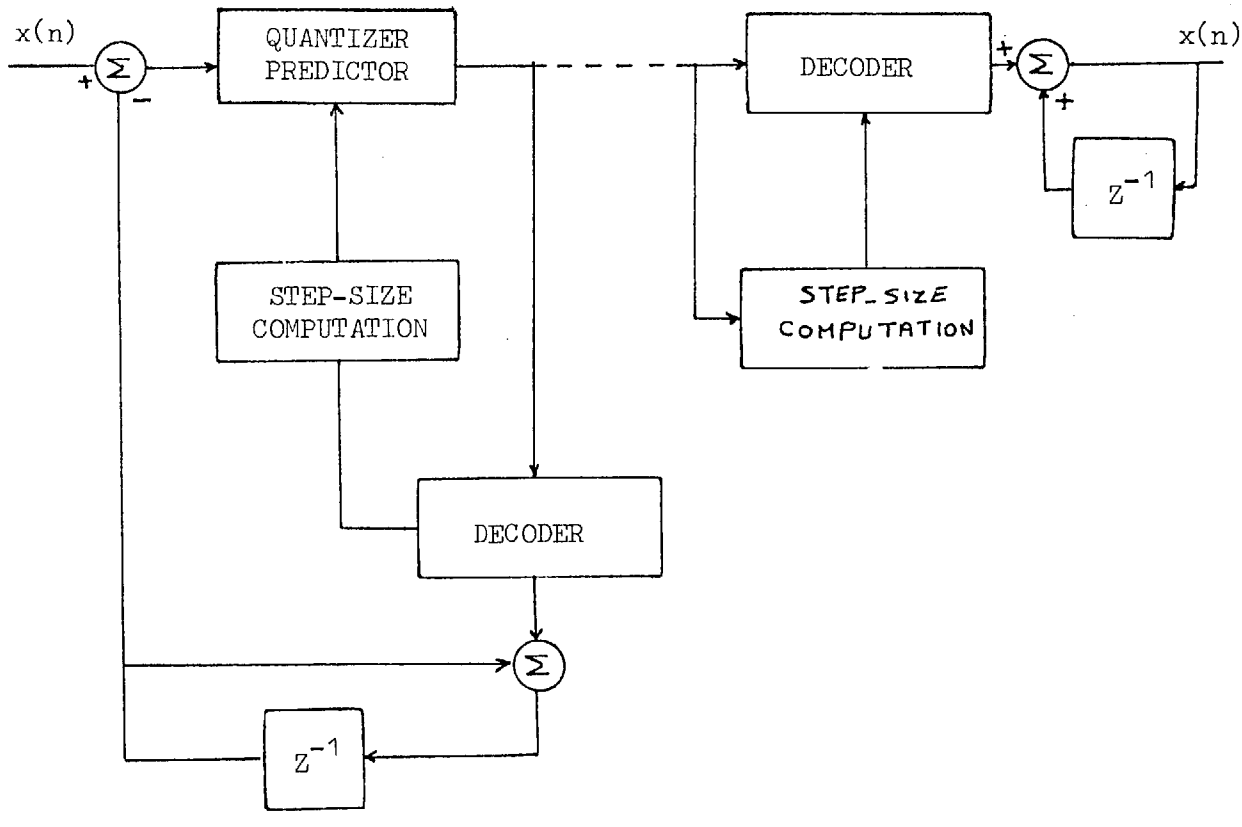



Fig. 1.2. Block diagram of ADPCM-F coder and decoder

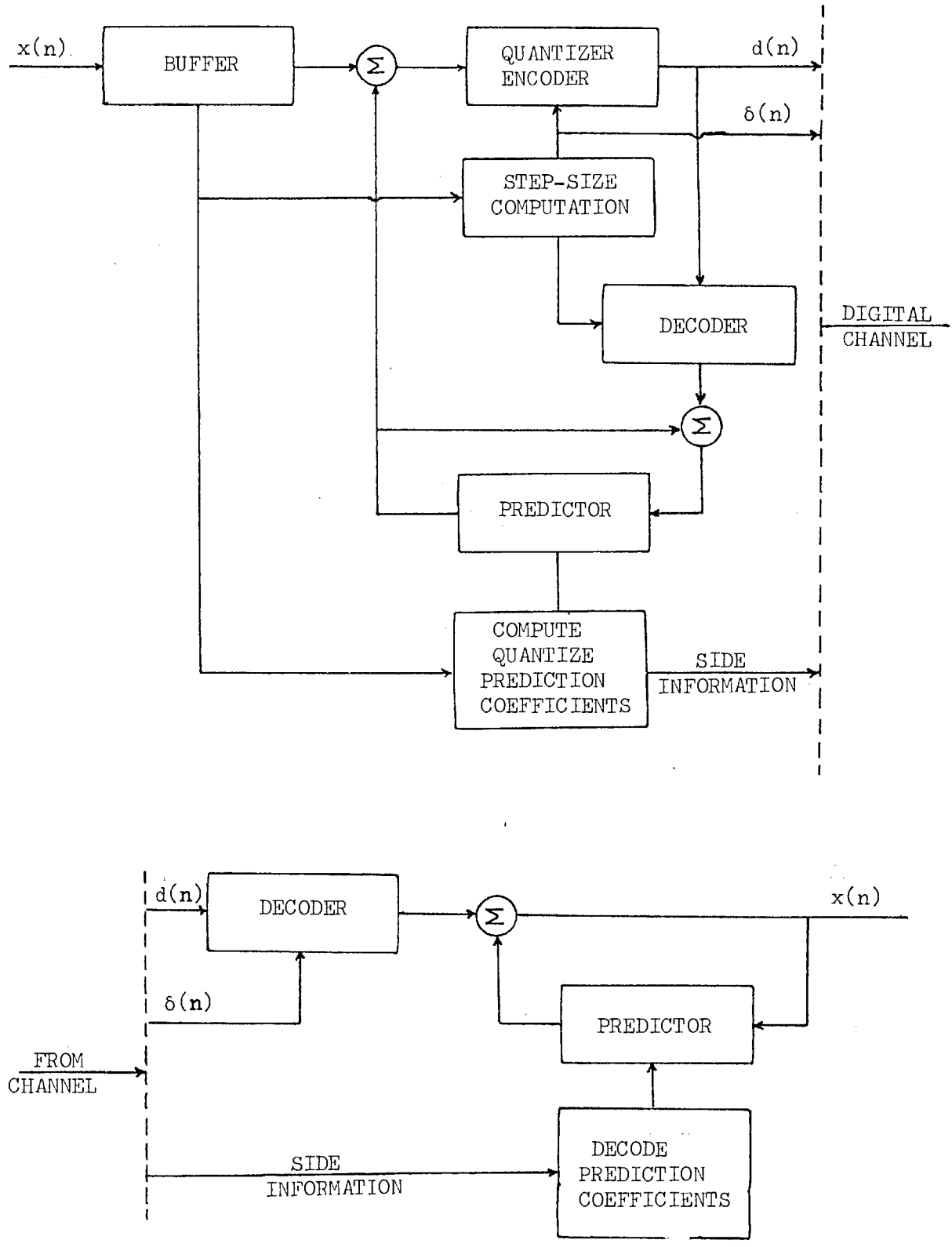



Fig.1.3. Block diagram of ADPCM-V

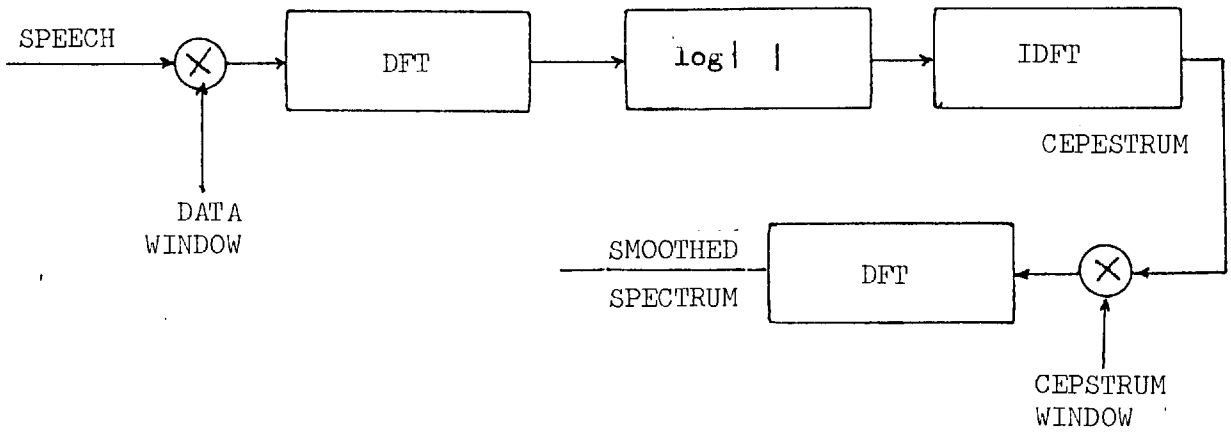



Fig.1.4. Block diagram of Homomorphic speech processor

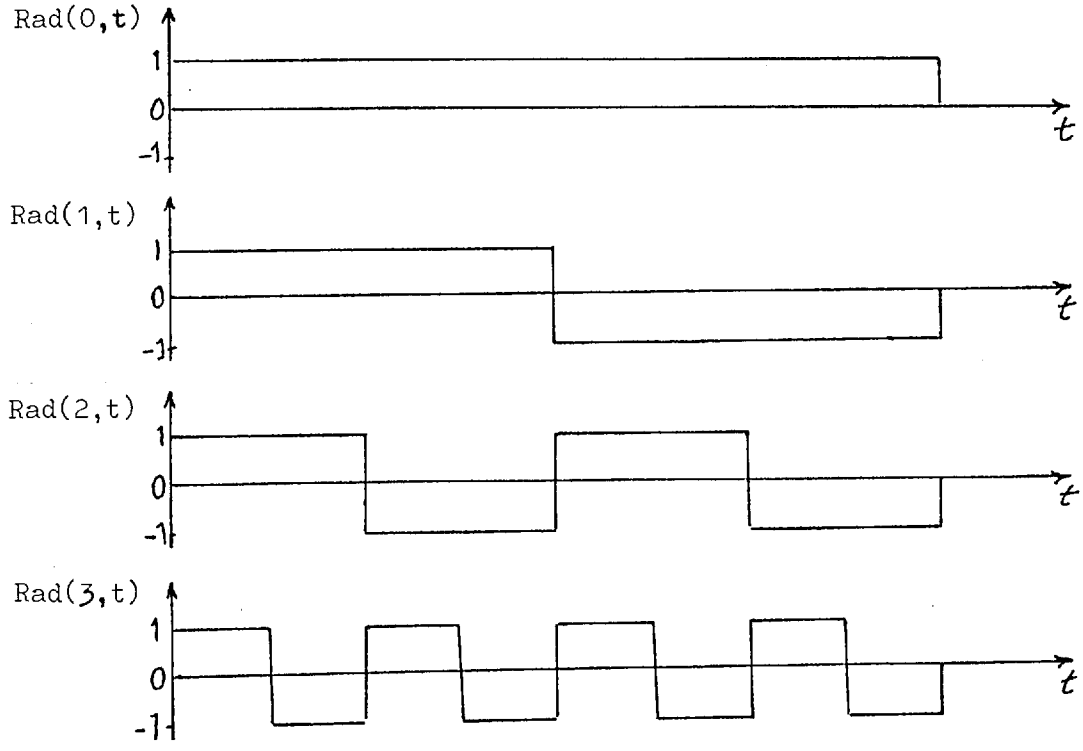
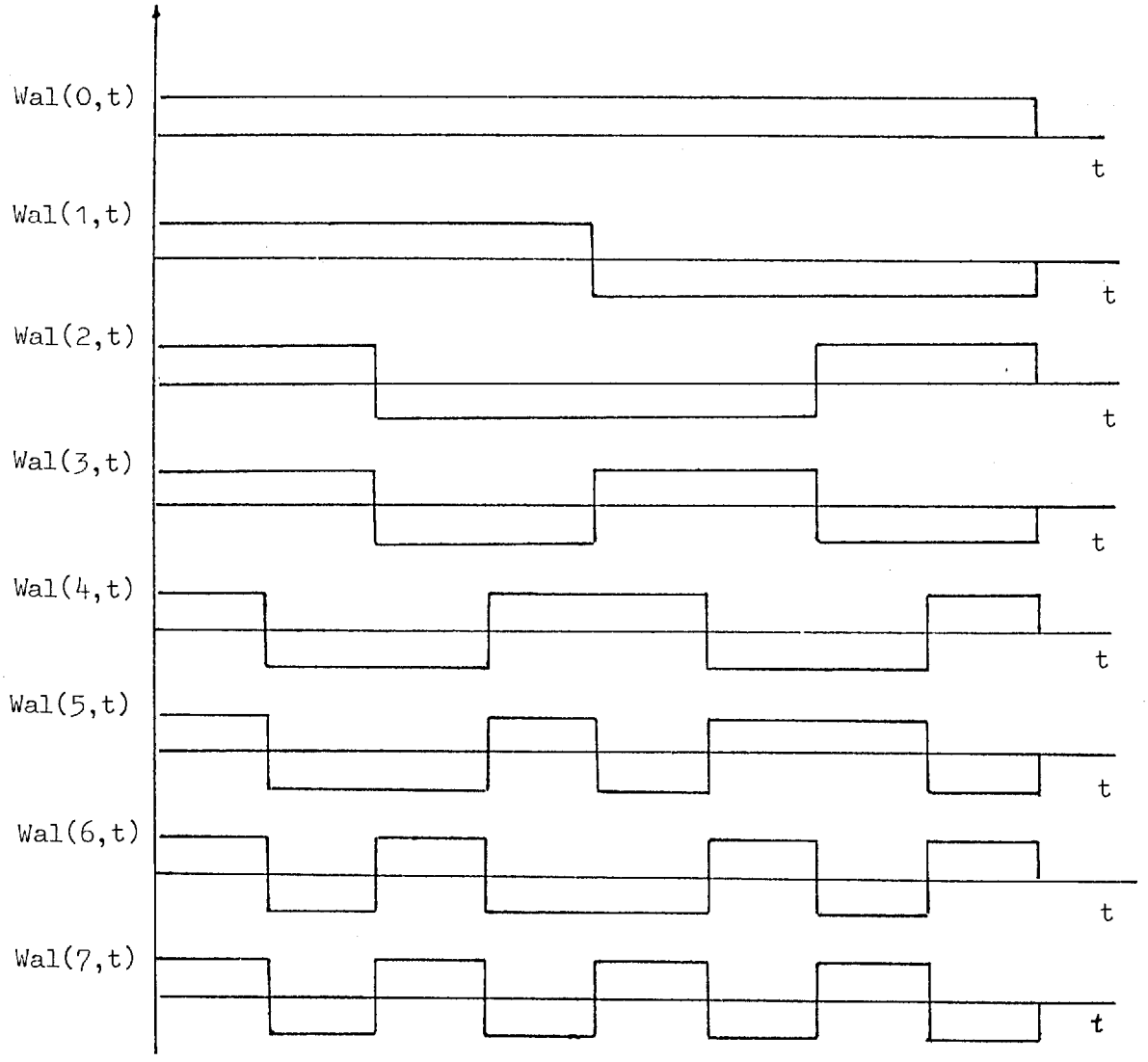



Fig.1.5. Rademacher functions

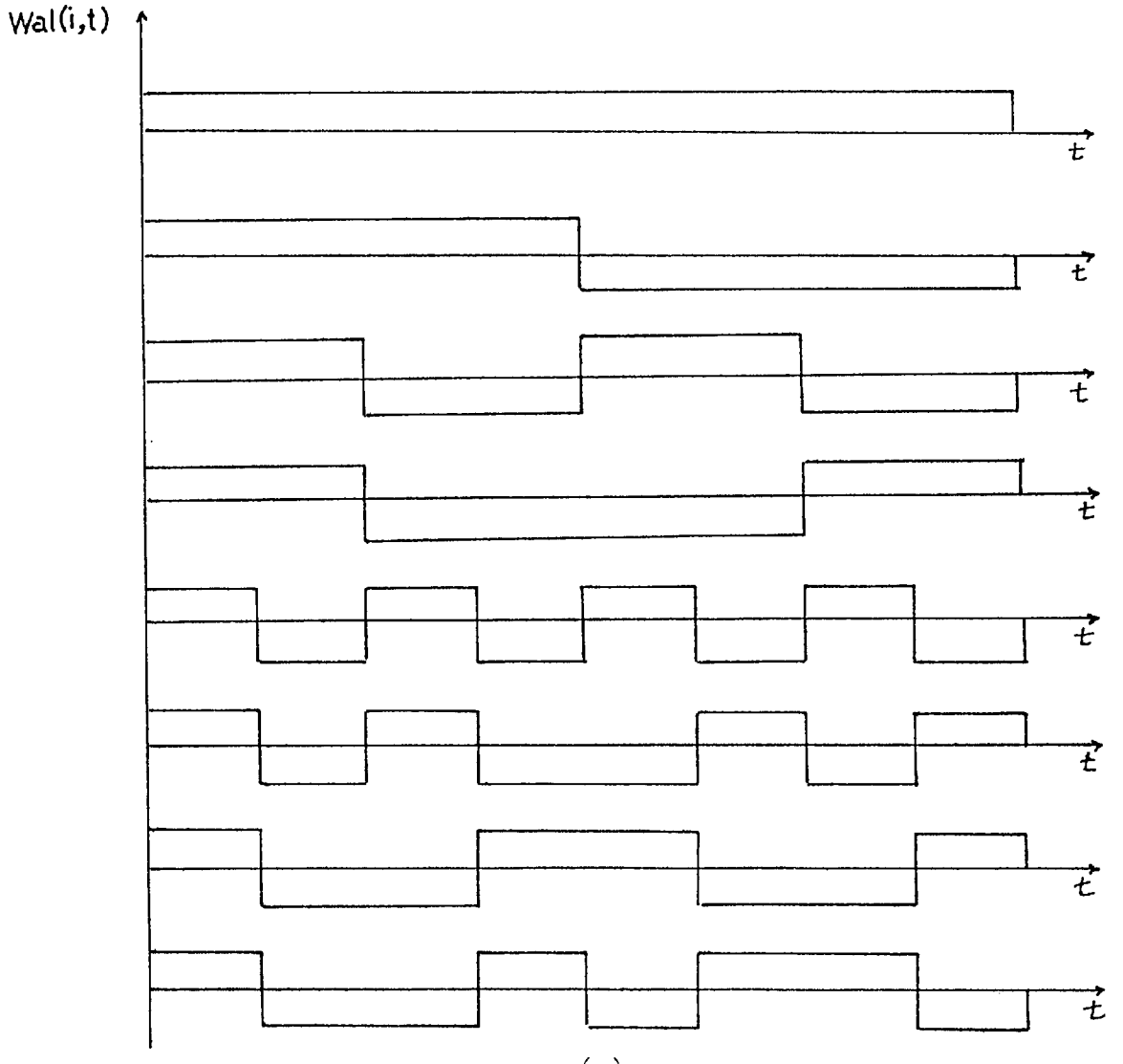



(a)

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

(b)

Fig.1.6. Walsh-Ordered Walsh functions:(a)Continous (b)Discrete

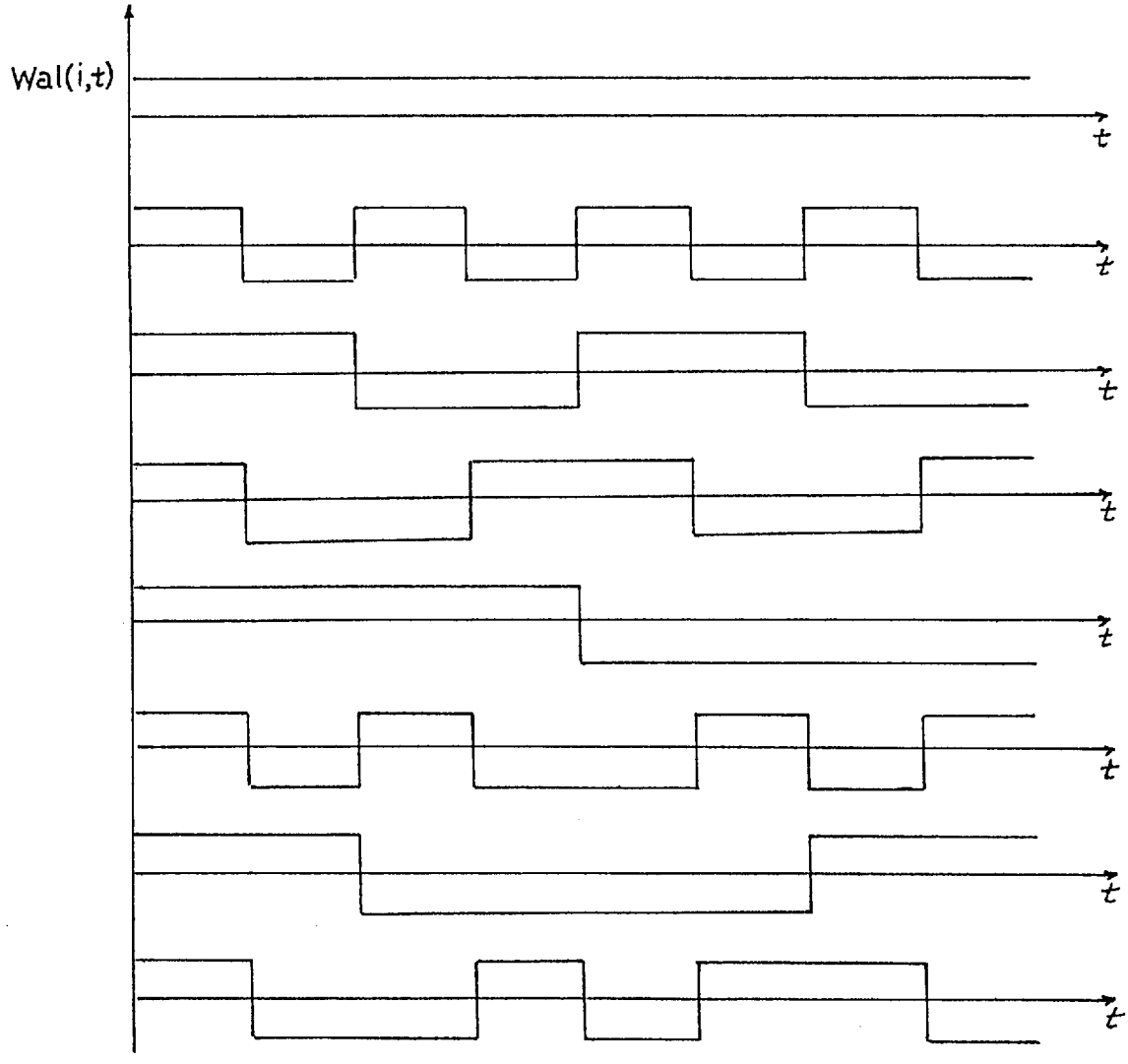



(a)

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

(b)

Fig.1.7 Paley-Ordered Walsh functions:(a)Continuous (b)Discrete

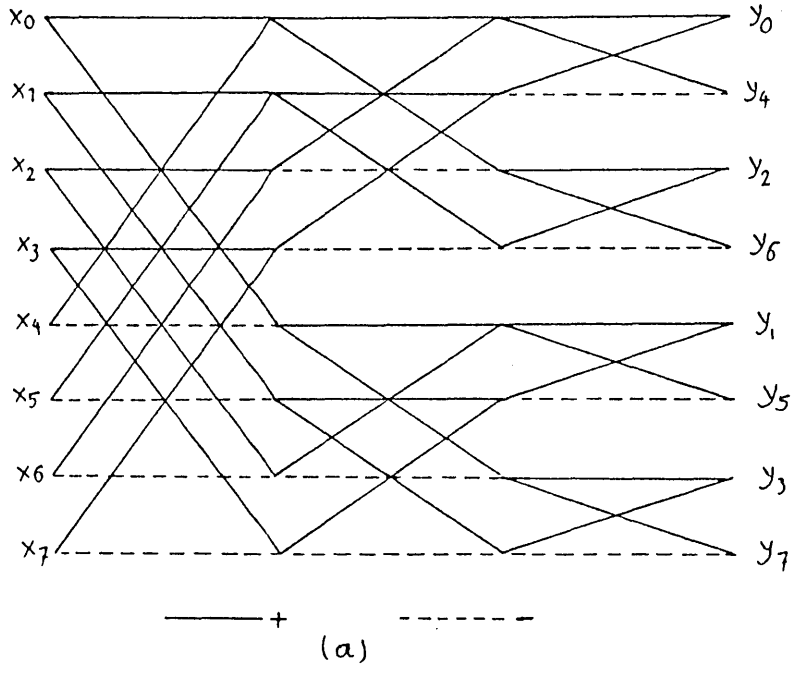



(a)

$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{bmatrix}$$

(b)

Fig. 1.8 Hadamard-Ordered Walsh functions: (a) Continuous (b) Discrete



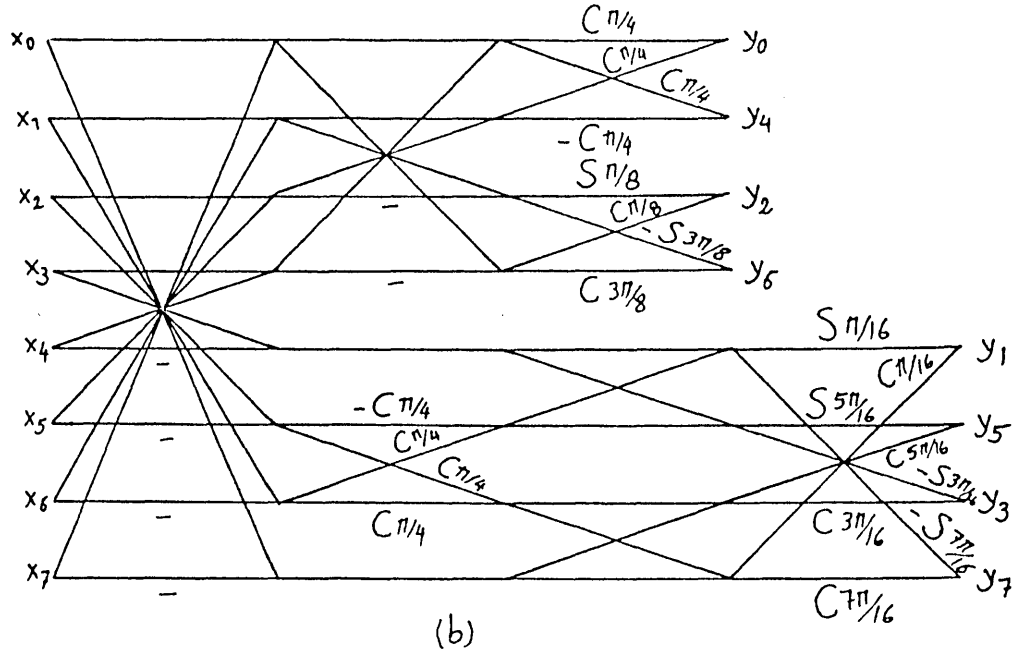




Fig.1.9. (a)Shanks' algorithm of the WI transform
(b)Fast Discrete Cosine Transform algorithm
for N=8 ; Ci=Cosi , Si=Sini

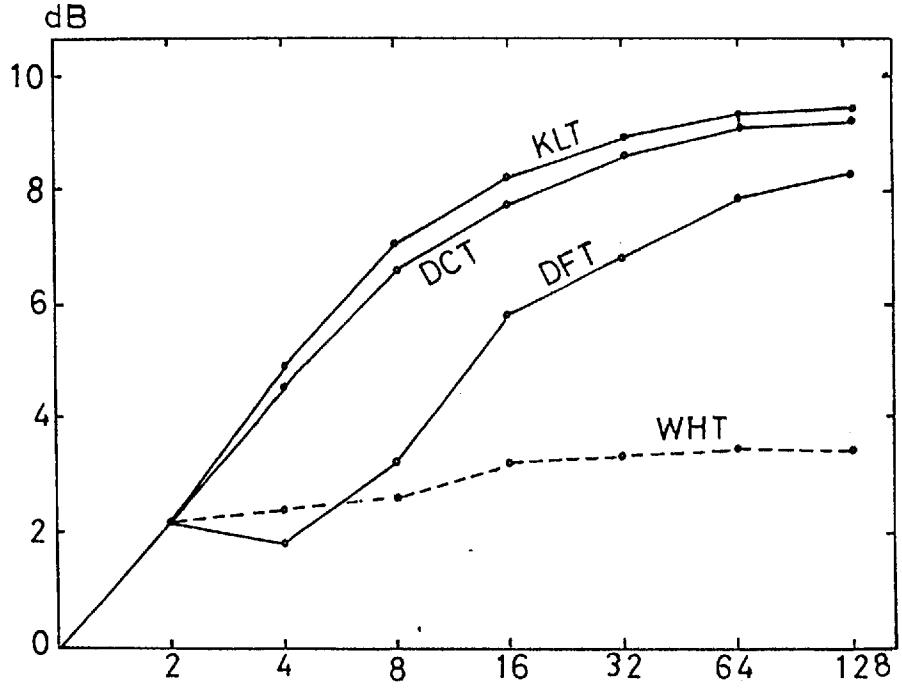



Fig.1.10. Gain in SNR over PCM for different block lenth N and various transforms

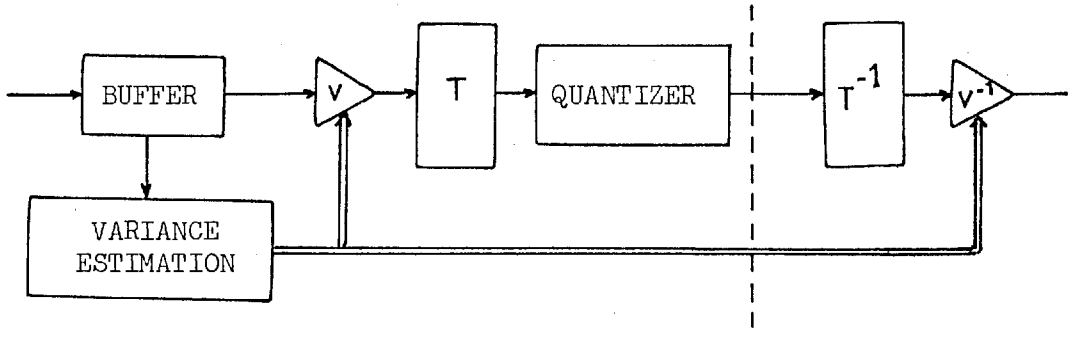



Fig.1.11. TCFBA structure with adaptive quantizer

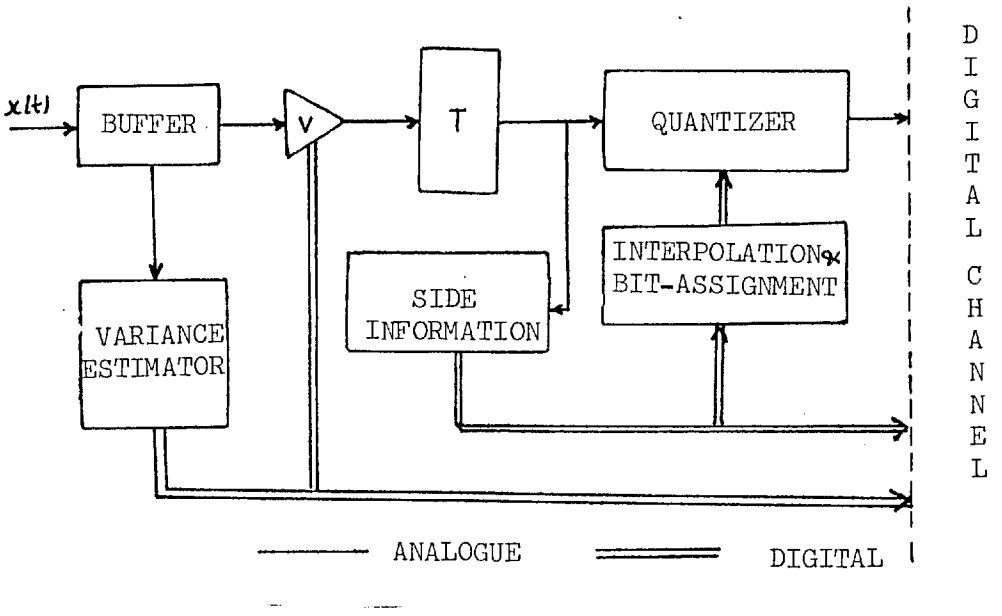



Fig.1.12. Structure of an adaptive transform coder

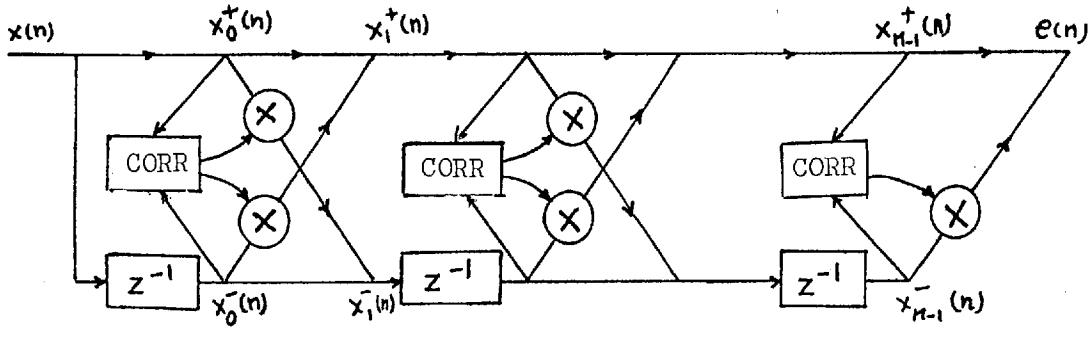



Fig.1.13. Structure of the PARCOR analyser

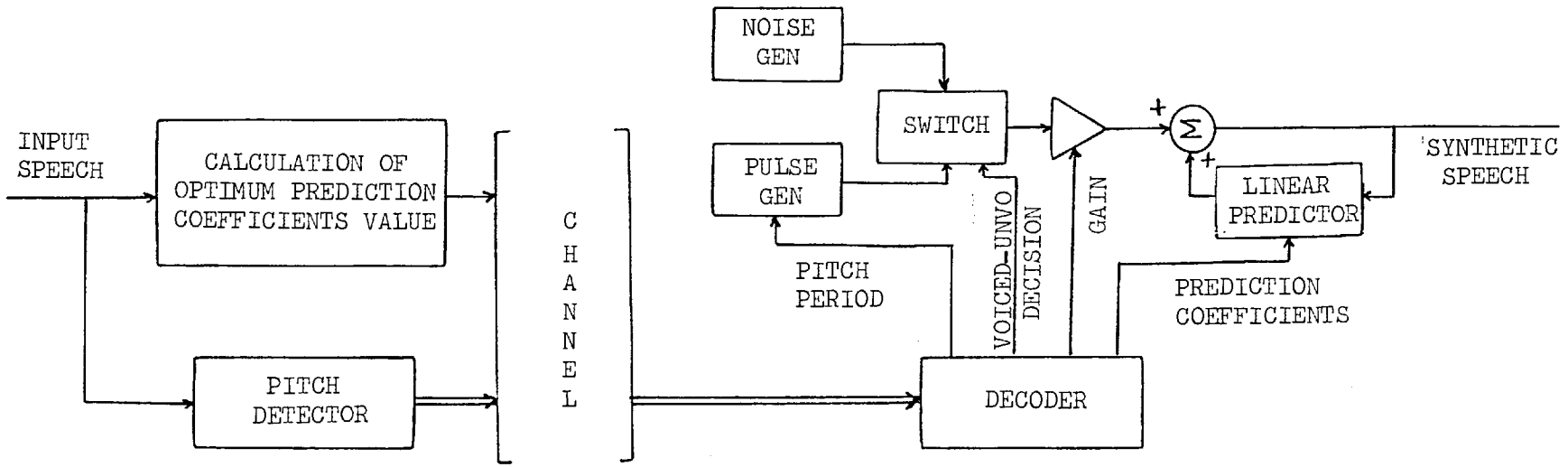



Fig.1.14. Linear prediction vocoder system

| BAND NO. | LIMITS(HZ) | MEAN | WIDTH |
|---|---|---|---|
| 1 | 200-330 | 270 | 130 |
| 2 | 330-430 | 380 | 100 |
| 3 | 430-560 | 400 | 130 |
| 4 | 560-700 | 630 | 140 |
| 5 | 700-840 | 770 | 140 |
| 6 | 840-1000 | 920 | 140 |
| 7 | 1000-1150 | 1070 | 150 |
| 8 | 1150-1310 | 1230 | 160 |
| 9 | 1310-1480 | 1400 | 170 |
| 10 | 1480-1660 | 1570 | 180 |
| 11 | 1660-1830 | 1740 | 170 |
| 12 | 1830-2020 | 1920 | 190 |
| 13 | 2020-2240 | 2130 | 220 |
| 14 | 2240-2500 | 2370 | 260 |
| 15 | 2500-2820 | 2660 | 320 |
| 16 | 2820-3200 | 3000 | 380 |

Table 1.1. Articulation bands

## CHAPTER 2

## Non-Uniform Linear Prediction Coding of Speech

The all-pole LP vocoder has been extensively employed for speech coding in the time domain[53-56]. However the application of this method is rather limited bacause of the high complexity and cost involved in its realisation. Numerous modified approaches have been proposed for reducing the complexity and improving the quality of the all-pole LP method. Methods such as the pole-zero modelling[56-59], using non-impulsive excitation for voiced sounds[62] and variable frame rate analysis techniques[63,64] which improve the quality, result generally in more complex systems which are not of interest in this thesis. Other non-pitch tracking methods based on the principle of LP technique[8,65] are in the category of the adaptive differential pulse code modulation schemes. These systems, in addition to the bit-rate necessary for transmission of the prediction coefficients, need extra bit-rate for transmission of the residue signals. By allocating at least one bit per sample to the residue signal, the total bit-rate will exceed the overall rate of one bit per sample. Besides, since the residue signal has a sudden change of energy at the beginning of the pitch periods, its quantization at the rate of 1 bit per sample will necessitate a complicated adaptation algorithm to be employed.

The new technique of linear prediction described in this chapter can provide a coder with a complexity far less than that of the all-pole LP modelling whilst still performing at a low bit-rate of around 0.5 bit per sample. This technique combines the principles of LP and transform coding, thus in addition to the correlation,which is the only property used by all-pole LP modelling, the characteristics of speech in transform domain can also be utilised. The main property of speech in the transform domain is the non-uniform distribution of energy at different sequencies. The new LP technique which exploits this property is consequently called the Non-Uniform Linear Prediction (NULP) method. In the NULP approach, the coefficients with larger variance

which are objectively and from the perceptual point of view more important, are more accurately predicted. A few base-band coefficients which generally contain most of the signal energy are directly transmitted thereby a low percentage of signal remains to be predicted by the prediction coefficients. This reduces significantly the necessary number of prediction coefficients, consequently, reducing the necessary computation. The NULP coding technique is employed in the WH and the DC domain. The reason for choosing these transforms is that the WHT can be most efficiently implemented while the DC transform is the best signal independent transform which can produce maximum non-uniformity in the energy distribution of the speech signals. Since the LP coding in general is inefficient for the coding of unvoiced parts of speech signals and this is because these parts are much less correlated than the voiced sounds, a new coding scheme for more efficient coding for unvoiced sounds is introduced. A new voiced-unvoiced decision based on the different energy distribution of these sounds in the transform domain is also described. The quality evaluation of the decoded speech reported in this chapter is based on the recent techniques of objective measurement of quality[66-71]. A simple objective method of quality measurement which employs the WH spectrum of speech is discussed and the qualities of the decoded speech at bit-rates of 4,6.5, and 8 kbit/s are compared with those of the best non-pitch tracking techniques[44,69].

## 2.1 The inefficiency of the All-Pole LP Coding in Transform Domain

Experimental results of using the all-pole modelling technique in the WH domain and in the DC domain showed the inefficiency of this method in the transform domain and reasons are given as follows: The values of prediction coefficients in this case tend to be nearly zero, thus the error of prediction given by Eq. (1.27) remains high. The reason for this can be searched in the difference between the correlation characteristics of speech in the time and in

the transform domain, since the LP analysis in general is a correlation approach. In the time domain, each sample is mostly correlated to its immediate preceeding samples, hence predicting each sample from these samples should result in an efficient prediction. This is the procedure employed by the all-pole LP modelling[66]. On the other hand, in the transform domain the correlation characteristics are functions of the type and the order of the transformation and the coefficients are not generally correlated to their immediate past coefficients since in the transform domain the past or future has no physical meaning. The analytic explanation of the above discussion can be given as follows:

Let us consider a first order all-pole LP modelling in which each sample is predicted from its previous sample, i.e.,

$$\hat{s}(n) = -a_1 s(n-1) \qquad (2.1)$$

where the value of $a_1$ is obtained by minimizing the squared-error between original and predicted samples. It can be shown simply that the value of $a_1$ can be obtained from the following relation

$$a_1 = -\frac{\left[s(n)\, s(n-1)\right]_{av}}{\left[s^2(n-1)\right]_{av}} \qquad (2.2)$$

and the total squared error is

$$E = \sum_n \left[s(n) + a_1 s(n-1)\right]^2 \qquad (2.3)$$

combination of Eq. (2.2) and Eq. (2.3) gives E in terms of sample values,

$$E = \sum_n s^2(n) - \frac{\left[\sum_n s(n)s(n-1)\right]^2}{\sum_n s^2(n-1)} \qquad (2.4)$$

Eq. (2.4) shows that the error E consists of two parts, the first part, i.e., the first term of the right side of Eq. (2.4) represents short term energy of signal which according to the Parseval theorem, has the same value in the time or in the transform domain[47]. The second term in the right hand side of Eq. (2.4) represents the amount of reduction in the error achieved by performing linear prediction. Thus the greater the value of this term, the less the error. The denominator of this term also represents the short term energy. The efficiency of the all-pole LP method in this case is then characterised by the value of the numerator which is related to the correlation between adjacent samples. The value of the root of the second term, in right side of Eq. (2.4), has been measured in three cases. In all cases the measurements were performed on a sentence spoken by a male speaker. In the first case, which is performed in the time domain, the normalized value obtained for this term was 0.86. In the second experiment, this term was measured in the Walsh-ordered WH domain and the measured value was found to be 0.2. Finally, this value was 0.01 in the Hadamard-ordered WH domain. The squared of the measured values represent the error given by the first order all-pole LP modelling in the three domains, respectively. These experiments show the inefficiency of the all-pole LP modelling in the WH domain compared to that in the time domain.

## 2.2 Non-Uniform Linear Prediction of Speech

The major differences between the time domain and the transform domain from applying a linear prediction analysis and synthesis point of view are in two aspects (i) The difference in the

correlation function. (ii) The difference in the variance distribution . Thus for an efficient linear prediction analysis and synthesis of speech in the transform domain these two factors should be considered. The NULP method, discussed in this chapter, can take into account both of these factors. In this method, before computing the prediction coefficients, two types of information which are related to the statistics of speech in the employed domain, should be obtained and stored in the analyser and synthesizer. The first type is referred to as prediction matrix. The prediction matrix specifies the coefficients which are the most appropriate for predicting any coefficient in the transform spectrum. The second type is the weighting matrix which weights the transform coefficients before and after performing the linear prediction processing.

Let us consider a first order NULP method for a simpler explanation. In this case the prediction matrix and the weighting matrix will be vectors. If we denote the elements of the prediction vecotr by $p(n)$ and similarly show the elements of the weighting vector by $w(n)$. Then the first order NULP similar to the all-pole LP case is defined as,

$$\hat{C}(n) = -a_1 . w(n) . C[p(n)] \qquad (2.5)$$

where $a_1$ is the prediction coefficient and $C[p(n)]$ is the most appropriate coefficient from which the coefficient $C(n)$ should be predicted. The total squared-error in this case can be obtained similarly by subtracting the original coefficients from their predicted values.

$$E = \sum_n [C(n) - \hat{C}(n)]^2 = \sum_n [C(n) + a_1 . w(n) . C[p(n)]]^2 \qquad (2.6)$$

In the following, the optimum values of $w(n)$, $a_1$, and $p(n)$ will be

determined to minimize the value of squared-error E.

## 2.2.1 Determination of the weighting and prediction vector

The optimum values for the weighting vector elements can be obtained by setting the derivation of Eq. (2.6) with respect to W(n) to zero. The speech signal in this case is assumed to be stationary since only one set of coefficients is aimed to be determined. The derivative of (2.6) is

$$\frac{\partial E}{\partial w(n)} = \sum_{n} 2\left\{w(n).C^2[p(n)].a_1^2 + C(n).C[p(n)].a_1\right\} = 0 \qquad (2.7)$$

By solving this equation, the optimum value of W(n) will be obtained by the following relation

$$w(n) = \frac{1}{a_1} \frac{\sum_{n} C(n) \cdot C[p(n)]}{\sum_{n} C^2[p(n)]} \qquad (2.8)$$

$a_1$ in this case is a constant since the speech signal is assumed to be stationary. The summation should be over a long term speech ensumble which consists of speech of different speakers of both sexes, so that the weighting vector can be used equally for different circumstances. Regarding Eq. (2.8), the optimum value of W(n) is the average projection of C(n) over C [p(n)] multiplied by a constant $\frac{1}{a_1}$. It is apparent that the larger the average projection, the more accurate C(n) will be predicted from C [p(n)]. Two extreme cases are when the average projection are unity and zero, where in the former case the prediction is 100 per cent accurate and in the latter case the two variables will be orthogonal, consequently the prediction in this case is inefficient. This leads us to choose p(n), i.e. the values of prediction matrix, in such a way that maximise the value of w(n).

## 2.2.2 Determination of Prediction Coefficients

The calculation of the prediction coefficient $a_1$ in the first order NULP can also be carried out by setting the derivative of (2.6) with respect to $a_1$ to zero. Differentation of (2.6) with respect to $a_1$ will result in

$$\frac{\partial E}{\partial a_1} = \sum_n \left\{ w^2(n).C^2[p(n)].a_1 + w(n).C(n).C[p(n)] \right\} = 0 \quad (2.9)$$

where $a_1$ can be obtained by solving the above equation.

$$a_1 = \frac{-\sum_n w(n).C(n).C[p(n)]}{\sum_n w^2(n).C^2[p(n)]} \quad (2.10)$$

The summation in this case is over the chosen interval. The optimum value of the total squared-error can be obtained by using (2.10) in (2.6) and the result is given below

$$E_p = \sum_n C^2(n) - \frac{\left\{ \sum_n w(n)\, C(n)\, C[p(n)] \right\}^2}{\sum_n w^2(n)\, C^2[p(n)]} \quad (2.11)$$

The first order NULP can easily be extended to higher orders. Since this chapter is only dealing with the first order NULP because of its simplicity in implementation, the extension remains to be described in another literature.

## 2.3 Choice of Efficient Transforms

In different speech coding methods using transform coding, the efficiency of a transform,in reducing the bit-rate or increasing

the quality at a specified bit-rate, mainly depends on the degree of the non-uniformity in the variance distribution obtained by that transform. Companella and Robinson[45] have compared the degree of efficiency of the KL, DF, and WH transforms in the block quantization method. The results of SNR measurements have shown that the KLT is the best and the DFT is better than WHT in improving the SNR. In other transform coding methods, such as adaptive block quantization technique[44], the degree of non-uniformity in the variance distribution also determine the efficiency of the employed transform in improving the quality of decoded speech.

In the NULP coding method, the non-uniform variance distribution is in two aspects in favour of this method. The two advantages are as follows: (i) More concentration of energy in the base-band coefficients. This characteristic, as will be seen later, can be advantageously used in conjunction with the NULP technique. (ii) The prediction coefficients obtained by the NULP method will convey more information when the transform signal energy is more non-uniformly distributed over the transform coefficients. This is due to the fact that in a very non-uniform case, a few coefficients are carrying almost the whole energy of the entire interval whilst the information carried by the rest is very small. Thus, in practice, fewer coefficients are needed to be predicted thereby resulting in more accuracy and less overall error in the signal representation.

Amongst the numerous existing transformations two have been chosen to be used in conjunction with the NULP method. The first one is the WHT which is a most efficient transform from an implementation point of view. Also, the WHT creates a reasonably good non-uniform speech spectrum[44]. The second chosen transform is the Discrete Cosine Transform (DCT). This transform is a most efficient signal independent orthogonal transform in creating a more non-uniform distribution for speech signals[44].

## 2.4 Non-Uniform Linear Prediction of the WH Spectrum

In this section, the NULP method which was described in the section (2.2) is closely investigated in the WH domain. The order of the WH transform chosen here is 64. This is because, using 64 point transform provides a frequency resolution of 125 HZ which is adequate for the analysis of long-term spectral distribution of speech. The coefficient indices used in this section are according to the Walsh-ordered WHT[31]. The coding algorithm employed in this section is a special case of the NULP algorithm discussed in section 2.2. In this method a few base-band coefficients are directly transmitted and the remaining coefficients are analysed and synthesized by the NULP approach. A special case of the NULP method is in the sense that the coefficients to be predicted are predicted from the base-band coefficients. This decision is based on the properties of the long term covariance matrix of speech signal in the WH domain which shows a substantial correlation between the base-band coefficients and the remaining coefficients. These long term covariance Properties also exist in the DC domain.

Construction of the weighting vector and the prediction vector in this case is simpler than the general case. To obtain these vectors, first a 64 x 64 covariance matrix is formed. This matrix can be constructed directly from WH spectrum of speech, by using Eq. (2.12), or by employing the time domain covariance matrix and Eq. (2.13).

$$R_{yy} = E\left[(Y-\bar{Y})(Y-\bar{Y})^{t}\right] \qquad (2.12)$$

$$R_{yy} = H \cdot R_{xx} \cdot H \qquad (2.13)$$

where $R_{yy}$ is the covariance matrix in the WH domain, $R_{xx}$ is the time domain covariance matrix, Y is the transform vector, and H

is the Hadamard matrix (note that the Hadamard matrix and its inverse are identical). The covariance matrix $R_{yy}$ used in this chapter was obtained from four sentences, two spoken by males and two by females, involving 10 seconds of speech. The 8 coefficients 3-10 were chosen as base-band coefficients which should be directly transmitted. This is because these coefficients have largest variances and in average carry about 65 per cent of the 64 point WH block. The next step is forming the weighting vector. For doing so, elements of the 8 colums (3-10) of the covariance matrix are divided by the variance of the base-band coefficients corresponding to each column. The maximum value of each row, that consists of eight elements, forms one element of the weighting matrix. The weighting vector obtained by this procedure is shown in Fig. (2.1).

The ith element of the prediction vector is the index of the base-band coefficients which correspond to the ith element of the weighting vector. The prediction vector in this case is shown in Fig. (2.2). This information i.e. the weighting vector and the prediction vector is stored in the analyser and synthesizer as information characterising the statistical properties of the WH spectrum of speech signals.

The prediction coefficients now can be computed using Eq. (2.10) where in the synthesizer the WH coefficients excluding the 8 base-band coefficient can be approximately recovered using Eq. (2.5).

## 2.5 Voiced-Unvoiced Decision

The linear prediction coding technique is generally inefficient for coding of unvoiced sounds[53]. The efficiency of the all-pole LPC for coding of unvoiced sounds is about 10 times less than that of the voiced sounds and Atal and Hanaver[53] have used this discrimination for seperating voiced from the unvoiced parts of speech signals.

The NULP is also found less efficient for the unvoiced coding. Two reasons conceivably contribut to this inefficiency: (i) the lack of correlation in unoiced sounds; (ii) different variance distribution of unvoiced sounds in the transform domain where more energy is concentrated towards the higher sequency coefficients This makes the base-band coefficients carry little information about the unvoiced sounds.

In this section a new criterion of voiced-unvoiced decision, based on the different energy distribution of these sounds in the WHT, is introduced. This criterion can be employed in a similar fashion in other transform domains. The different distribution of variance in the WH domain of unvoiced sounds,first was noticed by Gethoffer[46] for German speech. This difference also exists for speech in other languages since it is related only to the correlation characteristics of speech. The method for detecting the difference,in this section,is based on the ratio of the base-band energy to the whole energy of the interval

$$R = \frac{\text{Base-Band Energy}}{\text{Energy of the Interval}}$$

where R can have values between 0 and 1. The distribution of energy of typical speech over different sequencies is shown in Fig. (2.3) where each distribution corresponds to an R value which is denoted in the figure. This figure shows that small values of R (near zero) correspond to unvoiced parts of speech while larger R (near 1) represent voiced parts of speech. In subjective tests, the appropriate threshold value of R was found to be 0.3 so that for larger value than this the segment of speech signals under examination maybe considered voiced and when R is smaller than 0.3 the segment of speech signals under examination may be considered unvoiced.

After seperating the voiced from the unvoiced sounds,in the manner indicated above,the unvoiced sounds are coded by retaining

those coefficients which in average carry more energy than the other coefficients.

## 2.6 Quantization of Parameters

The basic principle of quantizing a continuous value is to choose a step-size and then evaluate how many step-sizes can be fitted in that value. If this step-size is equal for different values and for different samples, the quantization is referred to as linear quantization. For signals which have an exponential PDF (probability density function) such as speech signals, the logarithmic type of quantization which is in the category of non-uniform quantization, will reduce the amount of quantization noise and will result in a better signal-to-noise ration as well as a better dynamic range[72].

Another technique of quantization, which recently has been of great interest, is the adaptive quantization [5,8,73]. The basic principle of adaptive quantization is to increase the step-size during overload and to reduce it during granularity. Adaptive quantization can substantially enhance the SNR and improve the dynamic range, compared to the non-adaptive quantization. A comparative study of different quantization techniques has been presented by Noll[8].

Adaptive quantization has also been employed in this chapter for quantization of parameters. The base-band coefficients of the voiced parts and the retained coefficients of the unvoiced parts are quantized adaptively. The adaptation strategy chosen here is based on the adaptive quantization with a one-word memory technique, proposed by Jayant[73]. In this adaptation method, the step-size of each new sample is obtained by multiplying the step-size of the previous sample by a constant coefficient called multiplication factor,

$$\Delta(r) = \Delta(r-1) \times M$$

where $\Delta(r)$ is the quantizer step-size used for quantizing the rth sample and $\Delta(r-1)$ is the step-size of the (r-1)th sample. M is the multiplication factor which can have a maximum of $2^{(b-1)}$ values, where b is the total number of bits allocated to each sample. Thus each slot from a number of $2^{b-1}$ slots will correspond to a multiplication factor. For example, in a two-bit quantizer, in which two magnitude levels and the sign can be represented, we can have two multiplication factors where one of them is less than 1 (M1) and the other greater than 1 (M2). If the slot corresponding to the smaller magnitude is engaged at the (r-1)th sample, then the step size at the rth sample is M1 $<1$ times than that of the (r-1)th sample. Otherwise the multiplication factor M2 $>1$ will be used.

In this chapter two different systems with different numbers of parameters and different bit allocation shcemes have been considered. These systems employ the NULP coding for voiced sounds and unvoiced sounds are coded by the technique of section (2.5). In the first system the voiced sounds are modelled by their 8 base-band coefficients of the WH spectrum and 2 prediction coefficients to model the rest of the spectrum. The unvoiced sounds are represented by 16 coefficients of sequency indices 25 - 40. The base-band coefficients are coded by 3 bits per coefficient and 2 bits are allocated to each of 16 coefficients belonging to the unvoiced parts. Multiplication factors of the former case were chosen as M1 = 0.9, M2 = 1., M3 = 1., M4 = 1.75 and for the latter case they were chosen as M1 = 0.845 and M2 = 1.96. 4 bits were allocated to each prediction coefficient with a constant step-size quantization. This system, which for simplicity is referred to as NULPWHD1, has a bit-rate of 4 kbit/s.

In the second system which is referred to as NULPWHD2, the voiced spectrum is modelled by 8 base-band and 8 prediction coefficients. The unvoiced spectrum is represented by the 32 coefficients of sequency indices 17 - 48. Similar to the NULPWHD1, all the parameters of NULPWHD2, except the prediction coefficients, are quantized

adaptively. The only difference in quantization of parameters in these two systems is that in the latter system 4 bits are allocated to each base-band coefficient. The related multiplication factors in this case are, M1 = 0.9, M2 = 0.9, M3 = 0.9, M4 = 0.9, M5 = 1.2, M6 = 1.6, M7 = 2, M8 = 2.4. The choice of multiplication factors has been based on the statical properties of coded coefficients. The base-band coefficients of the voiced speech parts and the retained coefficients of the unvoiced part are almost uncorrelated signals. Adaptive quantization of this type of signal have been studied by Jayant[73] and Crochiere[12]. Results of these studies were employed for choosing multiplication factors given in this chapter.

The block diagram of the NULPWHD1 system is shown in Fig. (2.4).

## 2.7 Objective Evaluation of Speech Quality

In recent years, numerous methods for digital encoding and decoding of speech signals have been developed. These systems vary in terms of data-rate, complexity, cost, and quality. The problem of comparing these systems to determine the user acceptance is a difficult task. It should be noted, however, that the perception of speech is a highly complex process involving not only the different aspects of language, also the speaker's attitude and emotional state, and the characteristics of human auditory system. Various intelligibility tests,which are concerned with these factors, require extensive testing with human listeners, which is an expensive process in terms of both time and money. Besides, most of the coding systems produce usually highly intelligible speech. Thus such tests may not suffice to resolve small differences in acceptability. Direct user preference tests have been found more useful in terms of cost and application for wider range of systems[69], but these methods are still not highly cost effective. On the other hand, objective measurements would enable evaluation to be done by computer means as well as ensuring uniformity in speech quality evaluation. Also,

objective measurement can be incorporated into the design of better quality coding systems.

Due to the complexity of speech perception, the development of an algorithm which can predict the listener reaction to any distortion is not expected to be known in a near future. However, recently a great deal of research has been reported which enables to predict the subjective behaviour of many classes of speech coding systems[66-71]. One of these classes is the waveform coding for which the proposed objective measurements have been found to be highly correlated with their subjective results[71].

Amongst various objective measurement techniques the conventional signal to noise ratio measurement (SNR) is perhaps the most common form. The SNR is defined as

$$SNR = 10 \log \left[ \frac{\sum_n x^2(n)}{\sum_n [y(n) - x(n)]^2} \right] \qquad (2.14)$$

where x(n) and y(n) are the input and output signals of the coder and decoder, respectively. The summations in Eq. (2.14) are taken over the entire utterance. The SNR is not an adequate method of quality measurement even though in some systems[71,68] the SNR measurement results are highly correlated with the results obtained from an accurate quality measurement. One reason for the inadequency of the SNR test is that the soft parts of the speech in this test are almost ignored.

An improved SNR measure, proposed by Noll[66], averages the SNR of short segments and therefore assigns equal weight to loud and soft parts of utterance.

$$SNR_{seg} = \frac{1}{M} \sum_{m=0}^{M-1} 10 \log \frac{\sum_{n=0}^{N-1} x_m^2(n)}{\sum_{n=0} [y_m(n) - x_m(n)]^2} \qquad (2.15)$$

Eq. (2.15) shows that the utterance is divided into segments each containing N samples and the SNR in each segment is measured in dB. The average of these measures over M segments, in the utterance, is the $SNR_{seg}$.

In the $SNR_{seg}$ measurement the importance of errors at different frequencies are assumed to have equal effects while it is well known from the psycho-acoustic properties of hearing that the ear is more sensitive to noise at lower frequencies and higher frequency noise is less disturbing[15,12]. Based on this property, a frequency weighting criterion can be incorporated for making the $SNR_{seg}$ a more closed test to the subjective evaluations.

A simple form of frequency weighted $SNR_{seg}$ can be defined as follows:

$$SNRF = \frac{1}{M} \sum_{m=0}^{M-1} \frac{1}{B} \sum_{j=1}^{B} \left[ 10 \log \frac{(\sigma_{xj}^2)_m}{(\sigma_{nj}^2)_m} \right] \qquad (2.16)$$

where $\sigma_{xj}^2$ is the original signal power in a frequency band j, $\sigma_{nj}^2$ is the noise power in the same frequency band, B is the number of frequency bands and M is the number of segments in the utterance.

In this section two methods of objective measurement have been employed in order to assess the performance of NULPWHD systems, which appeared in Section 2.6. The first chosen test is the simple form of SNR measurement, since this test is very common-place.

The second method is a modified form of the frequency weighted $SNR_{seg}$. This method which is referred to as Sequency Weighted $SNR_{seg}$ ($SWSNR_{seg}$) employs Walsh Spectrum Weighting scheme rather than the frequency weighting. Because of the great similarity between the Fourier and the Walsh spectrum of speech, the difference in the obtained

results is negligible, but the $SWSNR_{seg}$ needs significantly less computation compared to the frequency weighted $SNR_{seg}$ specially for the evaluation of the NULPWHD system in which the Walsh spectrum of the original and the recovered signal is available. In this method a 64 point WHT is employed. This spectrum corresponds to the 0-4000 HZ frequency band and of Fourier spectrum. The region which consists of coefficients of sequency indices 4 - 51, corresponding to the articulation band 200 - 3200 HZ, is divided into four regions. These regions contain the coefficients 4 - 12, 13 - 22, 23 - 34, and 35 - 51, respectively. The division has been carried out in such a way that the contribution of regions to the articulation index[15] is approximately the same. The overall $SWSNR_{seg}$ is then the average of $SNR_{seg}$ in each region and over the utterance, according to Eq. (2.16).

Four sentences, two spoken by males and two by females were analysed and synthesized. These sentences were digitized using the A/D convertor and a PDP-15 computer. The sampling rate was 8 k sample/s and the quantization of these samples at the first stage is a 12 bit linear quantization which is converted to 9 bit logarithmic quantization. The choice of 9 bits/sample is because the word-length of the PDP-15 computer which is 18 bits so that each word can be used to store two samples of speech. The digital speech obtained by this scheme can be stored temporarily on the Disc of the PDP-15 or permanently on a magnetic tape (deck tape). Before performing any processing by the software of the PDP-15, the stored speech is transferred from deck tape to disc. The speech is then transferred from the Disc to the central memory of the computer where each word of the Disc occupies two words of the central memory in which each word consists of one sample of speech. Further information is available in Chen[74].

### 2.7.1 Experimental Results

(a) Conventional SNR measurement: The conventional SNR of

the decoded speech from the NULPWHD1 system,which operates at 4 kbit/s, and from the NULPWHD2 system,which operates at 8 kb/s, was measured. The results are shown in Fig. (2.5).

(b) Sequency Weighted $SNR_{seg}$ measurement: The $SWSNR_{seg}$ of the NULPWHD1 and NULPWHD2 was measured and the values are shown in Fig. (2.6).

The results of the conventional SNR measurements, shown in Fig. (2.5), indicate that about 70 and 80 per cent of the original signal can be recovered after transmission at 4 kbit/s and 8 kbit/s respectively. The SNR results in this case are independent of the speaker's sex. The 4 kbits system, i.e. NULPWHD1, seems to be more efficient in terms of SNR, since the 8 kbit system, i.e. NULPWHD2 SNR improvement is only about 20 per cent at the cost of 100 per cent increase in the bit rate.

The results of $SWSNR_{seg}$ measurement (Fig. 2.6) show a better quality for decoded speech for male speakers. Listening tests also confirmed this discrimination and corroborated the validity of the $SWSNR_{seg}$ measurements. The $SWSNR_{seg}$ results also show a better performance for the NULPWHD systems than that given by SNR results. In the study of a number of waveform coders presented by Tribolet et al[69] the results of frequency weighted SNRseg shown to be about 10 dB less than that of SNR measurements while in the NULPWHD system being around 2 dB. The reason is that in the NULPWHD systems, priority is given to those coefficients which have larger variance and are perceptually more important.

## 2.8 Non-Uniform Linear Prediction Coding of the Discrete Cosine Spectrum

The criterion for employing the NULP method in the DC domain (NULPDCD) is similar to that described for coding of WH spectrum.

The implementation of DCT in general is more complicated than that of the WHT. However, a 16 point DCT has been adapted here to enable a simple implementation. The implementation of DCT using the IBN technique is described in Chapter 4. In the NULPDCD , described in this section, we choose three base-band coefficients of indices 0, 1, 2 and represent the 12 coefficients of indices 3 - 13 by one prediction coefficient. Coefficients of indices 14 and 15 are ignored, since after low-pass filtering of input signal, these coefficients are almost eliminated. The covariance matrix of speech in the DC domain,for a 16 point interval,is shown in Fig. (2.7). The covariance matrix shows that the first three coefficients have the largest variance and are almost uncorrelated to each other while the 4th coefficient is highly correlated to the 2nd coefficient. This justifies the choice of the first three base-band coefficient to be directly transmitted.

The weighting matrix (vector in this case) is obtained from the first 3 columns or rows of the covariance matrix (covariance matrix is a symmetry matrix), according to the procedure described in Section 2.2. The weighting vector,which is used for the determination of the prediction coefficients in transmitter and for weighting the predicted coefficients in the receiver,is shown in Fig. (2.8).

The prediction vector,which defines the base-band coefficients from which other coefficients are predicted,is shown in Fig. (2.9). This vector is obtained in the same way described,for the prediction vector in WH case,in section 2.4

The prediction coefficients,which in this case represent the 11 coefficients of indices 3 - 13,can be obtained using Eq. (2.10) which is repeated below with different limits of summations.

$$a_1 = - \frac{\sum_{i=3}^{13} w(i).C(i).C[P(i)]}{\sum_{i=3}^{13} w^2(i) . C^2[P(i)]} \qquad (2.17)$$

Eq. (2.17) can be simplified since $C[p(i)]$ for i=3-13 can have one of three values C(0), C(1), and C(2), thus the number of multiplications in the numerator and denominator of this equation can be reduced from 11 to 3 by factorisation.

The NULPDCD system reduces the 16 DC coefficients to 4 parameters, namely, 3 base-band coefficients and one prediction coefficient. In the decoder, 11 coefficients of squency indices 3 - 13 are recovered by multiplication of prediction coefficient by its corresponding base-band coefficients which is obtained from the prediction vector, as shown in Eq. (2.18) below.

$$\hat{C}(i) = -a_1 \cdot w(i) \cdot C[p(i)] \qquad i = 3-13 \qquad (2.18)$$

where w(i) is the ith element of the weighting vector and P(i) is the ith element of the prediction vector. The values of 14th and 15th sequency coefficients, in this system, are taken to be zero before inverse transformation. The block diagram of the NULPDCD system are shown in Fig. (2.10).

## 2.9 Quantization and Computer Simulation Results

The quantization of the NULPDCD parameters is identical to the quantization scheme employed in section 2.6 for quantization of the NULPWHD1 parameters. Three base-band coefficients of sequence 0, 1, 2 are quantized by 3 bits APCM and 4 bits linear PCM allocated to the prediction coefficient. No seperate coding was considered for unvoiced sounds because of the short duration analysis intervals involved. With this quantization scheme, the bit-rate of NULPDCD system is 6.5 kbits/s.

The NULPDCD system was simulated on the PDP-15 computer and the same 4 sentences which have been used in section 2.7 were processed. The SNR of processed speech for each sentence is shown in Fig. (2.11).

The SNR obtained from the NULPDCD system at 6.5 kbits/s shows that this sytem is a very successful and promising system. It should be noted that the perceptual performance of this system is better than that the SNR results indicate, since priority is given to lower sequency coefficients which are perceptually more important.

### 2.9.1 Comparison of Results

In this section we compare the NULPDCD system with the Adaptive Transform Coding (ATC) system[44] which is known as the best non-pitch tracking speech coding system. Considering low-bit rates, the ATC system results in 14 dB SNR at 9.6 kbits/s[69]. This system employs a 128 point DCT and a relatively complicated algorithm for interpolation of the variance distribution and bit allocation. In the NULPDCT system, a SNR of 10 dB can be achieved at 6.5 kbit/s which is comparable with the results of ATC system. On the other hand,the NULPDCD system employs only a 16 point transform and the computation in the coder is limited to calculating one prediction coefficient which results in a much simpler implementation. This elementary comparison shows that the performance of a simple form of the NULPDCD system is comparable with that of a sophisticated form of ATC. The performance of the NULPDCD apparently can be improved by using a higher order transform and increasing the number of prediction coefficients. This is due to the higher efficiency of DCT at higher orders[44].

## 2.10 Discussion

In this chapter, a new technique for speech analysis and synthesis has been developed. This method is based on an extension of the linear prediction techniques. It has been shown that the linear prediction coding can be advantageously employed in conjunction with the transform coding methods. The results of such a combination offer new solutions of major problems in the conventional linear

prediction approach. The main features of this method are as follows: (i) In the transform domain, most of the energy of speech signals is concentrated in a few coefficients, which for most of the well-known transforms, these coefficients are in the base-band. This characteristic allows us to adopt a new strategy in which these few coefficients are directly transmitted and only the remaining coefficients are processed by the linear prediction method. This results in a lower number of prediction coefficients at a specified prediction error, which consequently results in a simpler computation and implementation scheme. (ii) The elimination of pitch detection. The base-band of the Fourier spectrum of speech is well-known in carrying pitch information and this has been utilised in the voiced-excited vocoders. Other spectra of speech such as the WH spectrum and the DC spectrum which are very similar to the Fourier spectrum of speech, naturally have this property. Thus transmission of the base-band coefficients can also convey pitch information and eliminate the necessity of pitch detection which by itself is a very complicated process. (iii) Natural speech recovery. In the linear prediction vocoder and other vocoders, the filter representing the vocal tract, is generally excited by impulse trains during voiced sounds. Since the impulse is not a perfect representation of voiced excitation, the recovered speech loses its natural quality. Transmission of the base-band coefficients instead, can solve this problem and results in a natural speech recovery.

Another result included in this thesis is the adoptation of a new technique for voiced-unvoiced detection based on the energy distribution properties of speech in the transform domain. This technique is of very low cost when taken in conjunction with the NULP method, since the transform spectrum of speech will be available for NULP coding. This voiced-unvoiced detection scheme has been investigated using the WH spectrum*.

---

*Extensive experiments have confirmed also the possibility of using other spectra such as the DC spectrum.

Seperation of unvoiced sounds allows a better coding for these sounds than using the NULP technique, since the linear prediction coding in general is inefficient for coding of unvoiced sounds because, as already mentioned, of lack of correlation in these signals. The proposed strategy for coding of unvoiced sounds is quantization and transmission of those coefficients which in average have largest variances. The transmission of these coefficients which are positioned in the higher sequencies will improve the high sequency quality of decoded speech. Since it is well-known that the unvoiced sounds are a noisy type signal and thus perceptually are more tolerant to the quantization noise, less number of bits can be allocated to unvoiced coefficients, thus increasing the number of transmitted coefficients which result in improvement in quality.

Finally, the recent techniques of quality evaluation of speech signals have been studied in this chapter. These methods which are based on the objective measurement technique are very promising in substituting the costly and time consuming methods of subjective tests. A new objective test,based on utilising the WH spectrum of speech,has been proposed and test results using this method are included. The main advantages of these type of tests are the ease of implementation and independence of variable human factors.

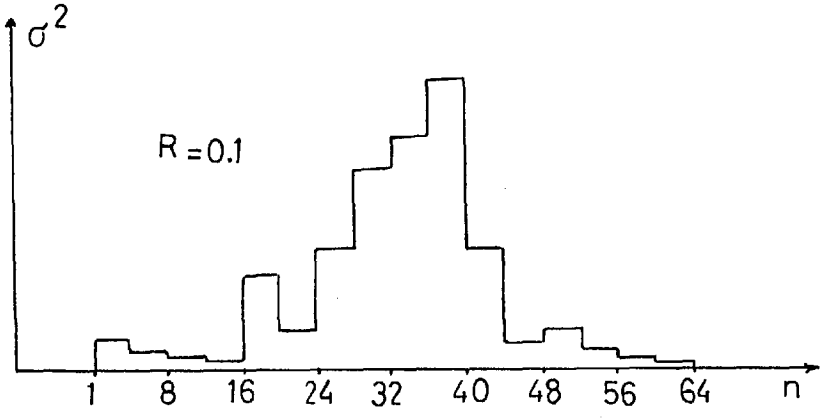
The comparison of results obtained by the new system with those obtained by the ATC has shown the better performance of the NULP system at very low bit-rates while its complexity is significantly lower than that of the ATC. This has been carried out in the basis of signal to noise ratio computation.

w(n)= [0.102,0.341,-0.23,1,1,1,1,1,1,1,1,-0.38,0.323,-0.27,
0.389,0.065,0.028,-0.09,-0.13,-0.06,-0.07,-0.28,-0.26,
-0.02,-0.35,0.455,0.298,-0.14,0.148,-0.18,0.174,0.081,
-0.05,-0.12,0.069,0.081,-0.09,-0.16,-0.12,-0.11,-0.13,
-0.12,-0.14,0.067,-0.07,0.035,-0.06,0.062,-0.01,-0.04,
0.053,-0.06,-0.06,-0.016,-0.15,-0.15,0.172,0.184,0.167,
-0.07,0.068,-0.06,0.071,0.016]

Fig,2.1. Weighting vector of the 64 point WH spectrum of speech

p(n)= [6,5,6,3,4,5,6,7,8,9,10,4,9,10,9,3,9,10,3,4,5,6,7,8,9,
10,3,4,9,10,7,6,5,8,4,6,5,6,7,8,9,10,3,5,7,10,5,4,5,
6,3,4,5,6,7,8,9,10,3,4,9,10,7]

Fig.2.2. Prediction vector of the 64 point WH spectrum of speech



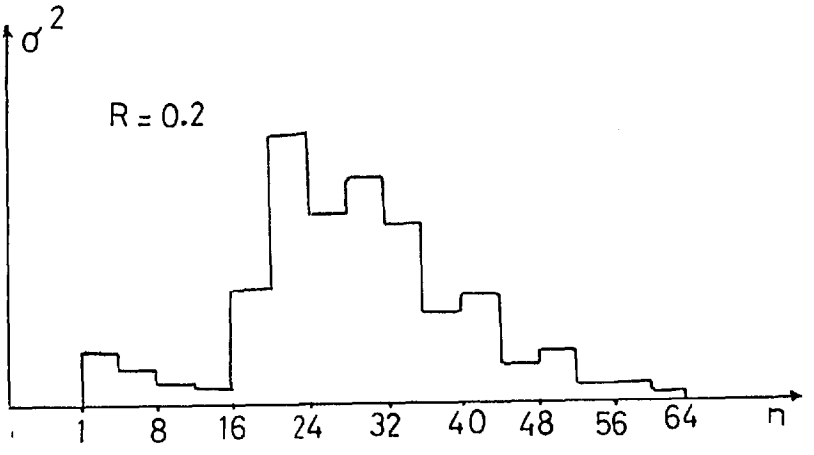




Fig. 2.3.a. Energy distribution of speech signals over different sequencies for R=0.1 and 0.2 in the WH domain.

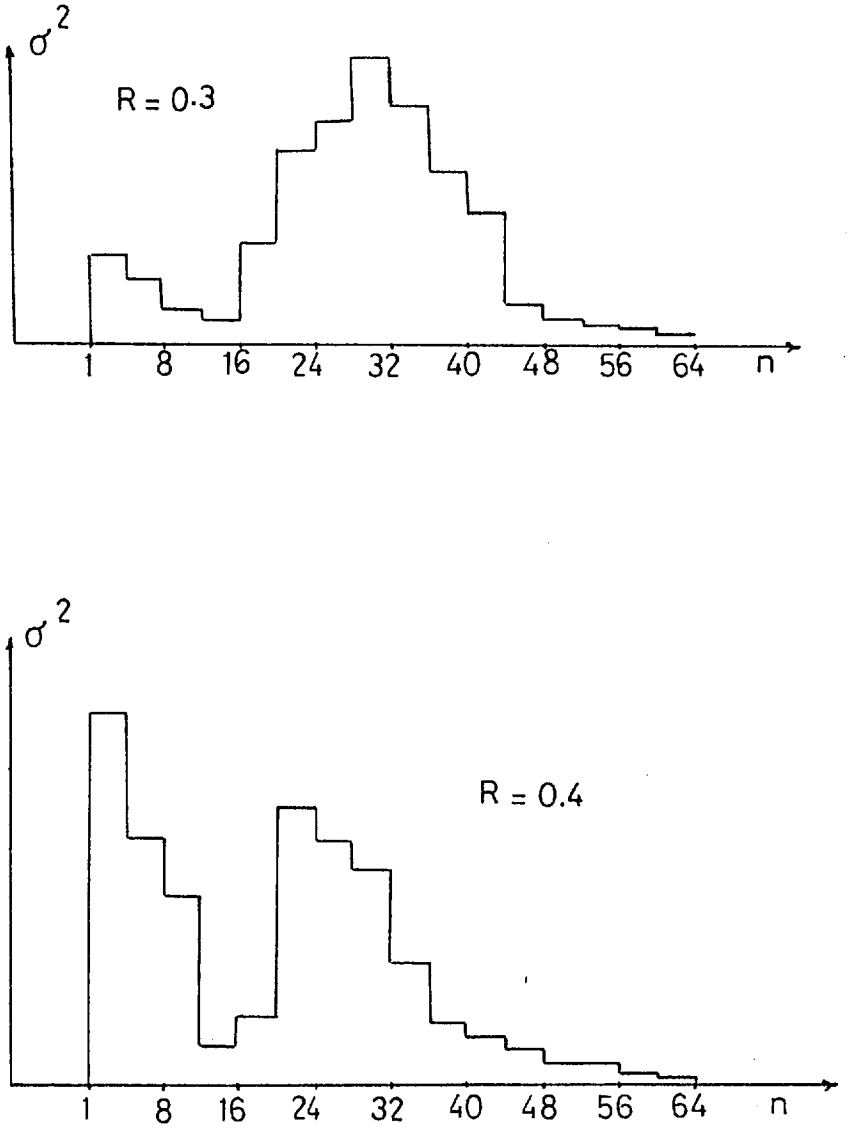



Fig. 2.3.b. Energy distribution of speech signals over different sequencies for R=0.3 and 0.4 in the WH domain.

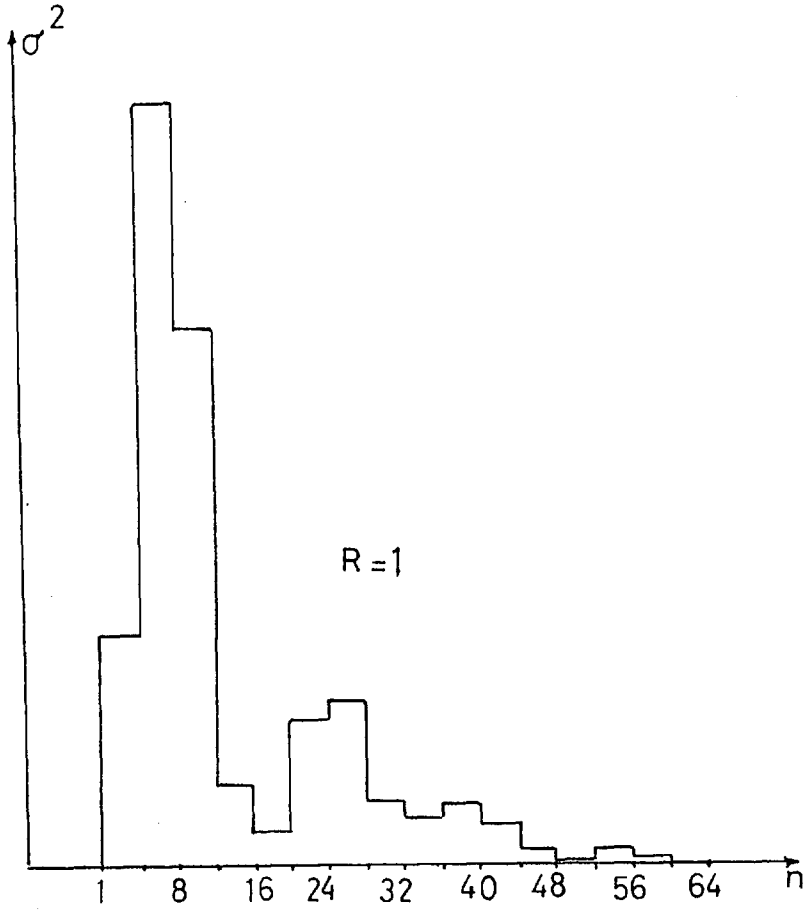



Fig. 2.3.c Energy distribution of speech signals over different sequencies for R=1. in the WH domain.

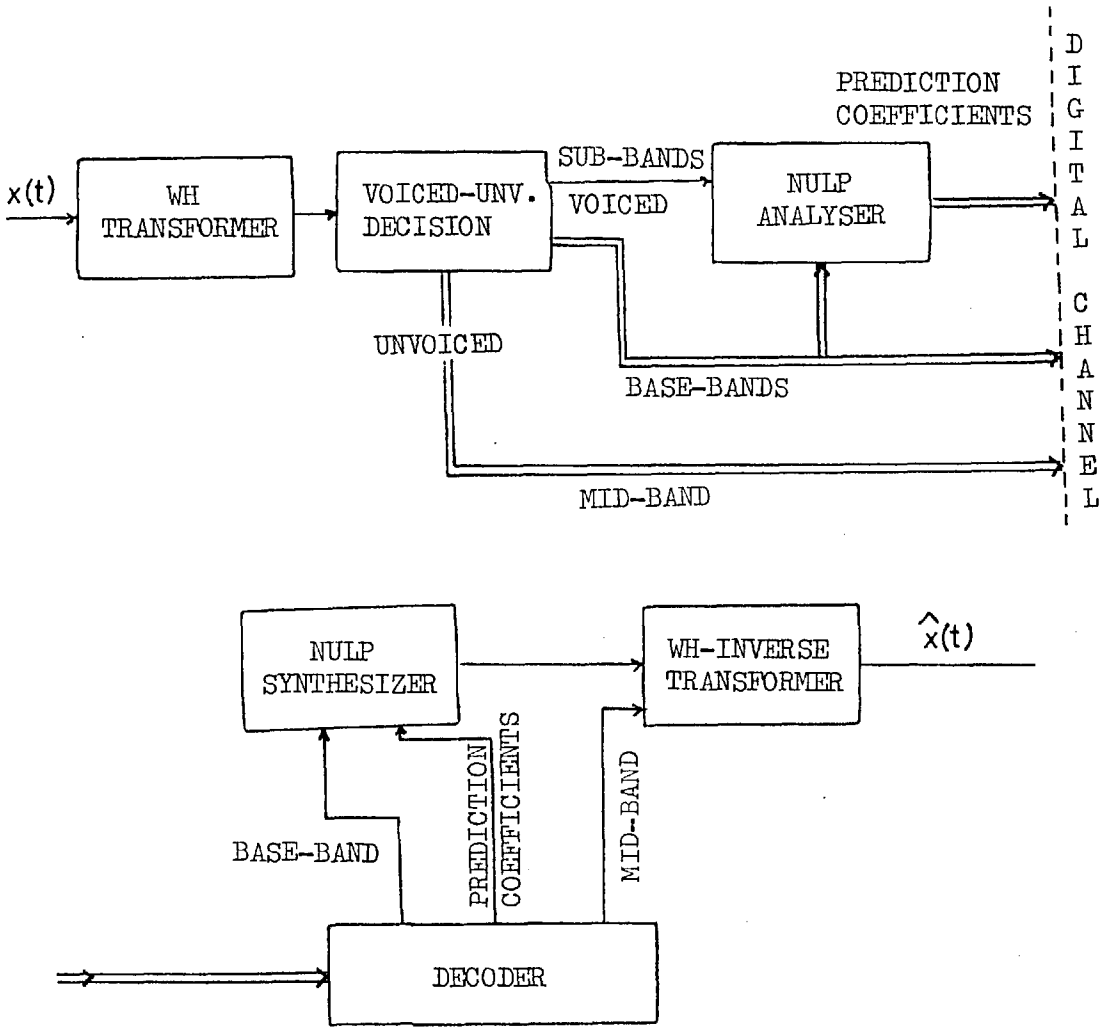



Fig. 2.4. Block diagram of the NULPWHD coder system

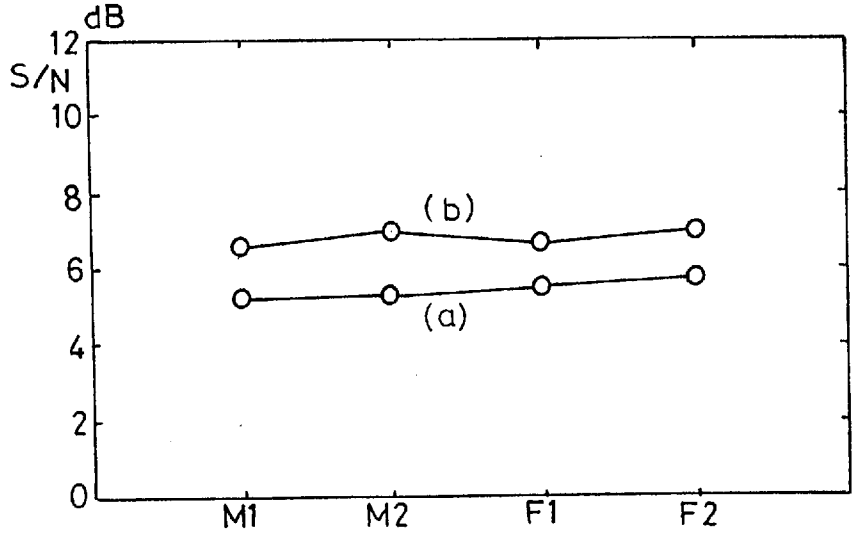



Fig. 2.5. (a) Signal to noise ratio of the NULPWHD1 system (4 kbit/s). (b) Signal to noise ratio of the NULPWHD2 system (8 kbit/s).

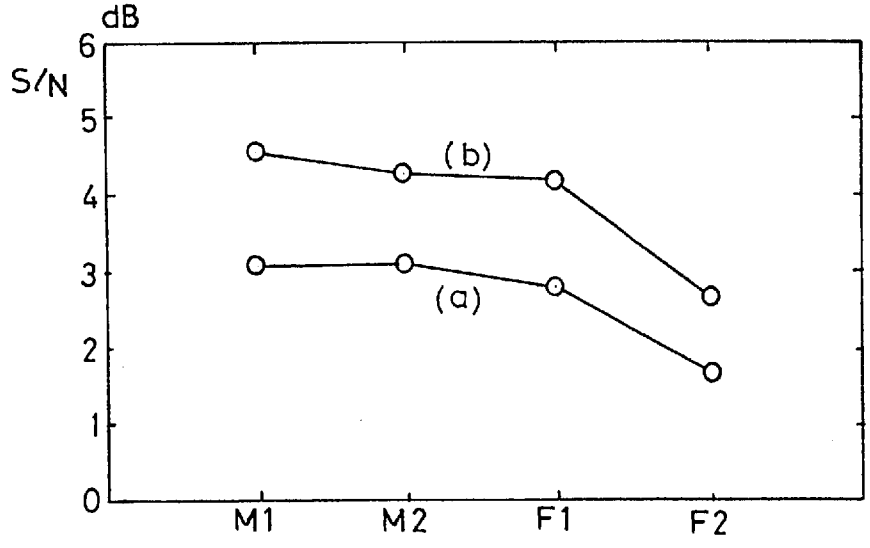



Fig. 2.6. (a) Sequency weighted segmented signal to noise ratio of the NULPWHD1 system. (b) Sequency weighted segmented signal to noise ratio of the NULPWHD2 system.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.000 | -0.029 | -0.273 | 0.033 | -0.076 | -0.002 | 0.017 | -0.025 | -0.032 | -0.007 | -0.001 | -0.010 | -0.017 | -0.000 | -0.006 | -0.000 |
| -0.029 | 1.603 | -0.080 | -0.572 | 0.001 | -0.084 | -0.021 | -0.106 | 0.015 | -0.057 | -0.017 | -0.030 | -0.001 | -0.015 | 0.000 | -0.000 |
| -0.273 | -0.080 | 1.310 | 0.077 | -0.017 | 0.055 | -0.014 | -0.045 | -0.012 | 0.005 | -0.036 | 0.004 | 0.000 | 0.000 | -0.001 | -0.000 |
| 0.033 | -0.572 | 0.077 | 0.470 | -0.005 | 0.074 | 0.017 | 0.048 | -0.010 | 0.023 | 0.001 | -0.001 | -0.001 | 0.005 | -0.001 | 0.000 |
| -0.076 | 0.001 | -0.017 | -0.005 | 0.051 | -0.000 | 0.036 | 0.003 | -0.008 | 0.002 | -0.004 | 0.003 | -0.004 | 0.000 | -0.001 | 0.000 |
| -0.002 | -0.084 | 0.055 | 0.074 | -0.000 | 0.065 | 0.004 | -0.025 | 0.004 | -0.007 | -0.005 | -0.010 | -0.001 | -0.003 | -0.000 | -0.000 |
| 0.017 | -0.021 | -0.014 | 0.017 | 0.036 | 0.004 | 0.144 | -0.004 | -0.038 | 0.001 | -0.014 | -0.002 | -0.008 | 0.000 | -0.003 | 0.000 |
| -0.025 | -0.106 | -0.045 | 0.048 | 0.003 | -0.025 | -0.004 | 0.203 | 0.006 | 0.012 | 0.002 | 0.000 | -0.000 | -0.000 | -0.000 | -0.000 |
| -0.032 | 0.015 | -0.012 | -0.010 | -0.008 | 0.004 | -0.038 | 0.006 | 0.097 | -0.003 | -0.007 | 0.001 | -0.003 | -0.001 | -0.000 | -0.000 |
| -0.007 | -0.057 | 0.005 | 0.023 | 0.002 | -0.007 | 0.001 | 0.012 | -0.003 | 0.062 | 0.000 | -0.008 | 0.001 | -0.001 | 0.000 | 0.000 |
| -0.001 | -0.017 | -0.036 | 0.001 | -0.004 | -0.005 | -0.014 | 0.002 | -0.007 | 0.000 | 0.061 | 0.003 | -0.005 | 0.001 | -0.001 | -0.000 |
| -0.010 | -0.030 | 0.004 | -0.001 | 0.003 | -0.010 | -0.002 | 0.000 | 0.001 | -0.008 | 0.003 | 0.034 | -0.001 | 0.003 | -0.000 | 0.000 |
| -0.017 | -0.001 | 0.000 | -0.001 | -0.004 | -0.001 | -0.008 | -0.000 | -0.003 | 0.001 | -0.005 | -0.001 | 0.009 | 0.000 | 0.002 | 0.000 |
| -0.000 | -0.015 | 0.000 | 0.005 | 0.000 | -0.003 | 0.000 | -0.000 | -0.001 | -0.001 | 0.001 | 0.003 | 0.000 | 0.002 | 0.000 | 0.000 |
| -0.006 | 0.000 | -0.001 | -0.001 | -0.001 | -0.000 | -0.003 | -0.000 | -0.000 | 0.000 | -0.001 | -0.000 | 0.002 | 0.000 | 0.001 | 0.000 |
| -0.000 | -0.004 | -0.000 | 0.001 | 0.000 | -0.001 | 0.001 | -0.000 | -0.000 | 0.000 | -0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 |



Fig. 2.7. Covariance matrix of speech in the 16-Point DC domain.

$$w(n)= [1,1,1,-.35,-.076,-.051,-0.01,-.062,-.032,0.034,-.001,\ 0.018,0.017,-0.00,0.,0.]$$

Fig.2.8. Weighting vector of the 16 point DC spectrum of speech

$$p(n)= [0,1,2,1,0,1,0,1,0,1,0,1,2,0,1,1]$$

Fig.2.9. Prediction vector of the 16 point DC spectrum of speeeh

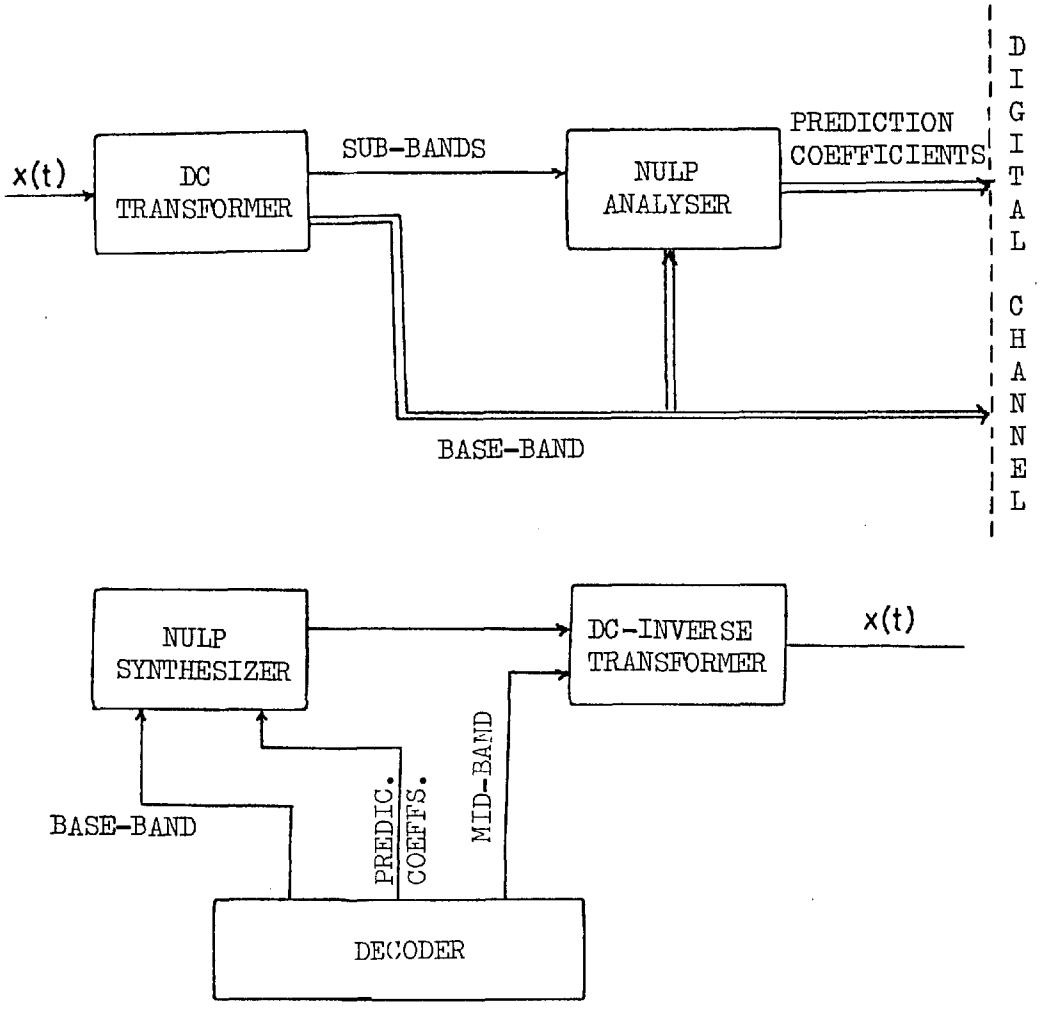



Fig. 2.10. Block diagram of the NULPDCD coder system.

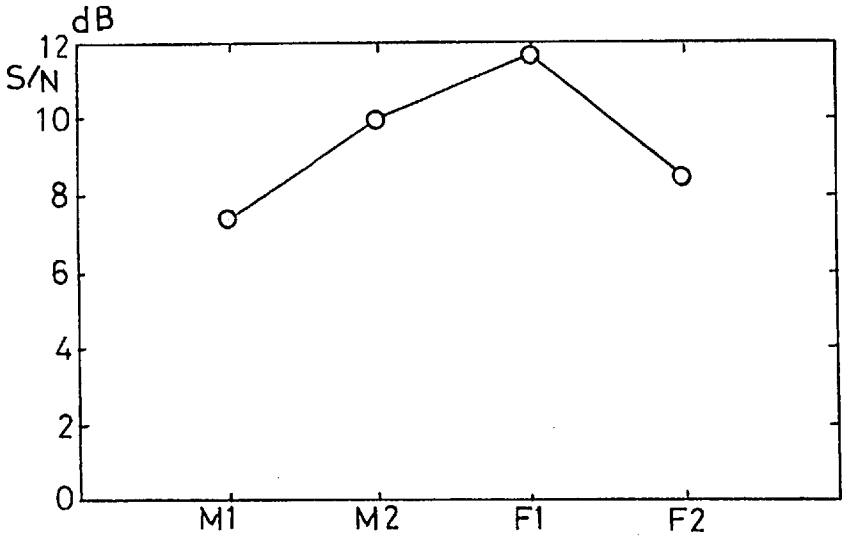



Fig. 2.11. The SNR of the NULPDCD system.

## CHAPTER 3

### Intermediate Binary Numbers: A New Approach to the Implementation of Signal Processors

In the conventional approach to the implementation of digital signal processors, the analogue signal is converted to digital by the PCM and the digitised signal is processed by binary arithmetic units[75]. An alternative for digital representation of a signal is the binary sequence obtained from the Delta Modulator (DM) or Delta-Sigma Modulator (DSM). This type of representation has two advantages (i) Simpler conversion from or to the analogue form is possible. The conversion which is accomp lished by the DM or DSM is much less complex and more flexible than the PCM convertor. (ii) The processing of the DM or DSM output can be achieved by simpler arithmetic units. The disadvantage is the high bit-rate for which the speed of processing must be increased, but for low-bandwidth signals, the speed still remains within the range that can be handled by ordinary logic components.

One class of processors using the DM or DSM is that which employs a hybrid form of processing, i.e. a combination of analogue and digital processing. The work of Lockhart[76,77] which is concerned with the implementation of digital filters, Lagoyannis[78] on the implementation of correlators and the work of Dempster and Steele[79] for implementation of a Wattmeter, are in this category. All of these systems are based on one principle of analogue (resistive) weighting the binary representation of signals obtained by the DM or DSM.

Goodman[80] and Peled and Liu[51] have also utilized the simple multiplication characteristic of the DM output in a different way and obtained a method for digital DM multiplication. This type of

multiplication is accompolished by effectively performing a number of additions and subtractions. These multipliers have been employed in the realisation of non-recursive digital filters[51,80].

The above techniques make the DM and DSM attractive for the implementation of low-bandwidth signal processors. However, there is a drawback in these methods which makes them have limited applications. This drawback is due to the fact that the multiplication, either in hybrid or in digital form can only be performed once on the binary numbers obtained from the DM or DMS. This is because after one DM multiplication the form of the product signal is changed and becomes either an analogue signal in the hybrid case or a binary positional (well known binary form) signal in the digital case. This prevents the signal from further processing involving the simple DM multiplication techniques. This is the major reason why only simple signal processing schemes such as non-recursive digital filtering have been considered by this technique so far.

In this chapter a new class of binary numbers referred to as Intermediate Binary Numbers (IBN) is presented. These numbers are a general form of binary number representation directly obtainable from either DM or DSM. The generalisation is with respect to the fact that in the direct use of the DM or DSM, the sampling rate is equal to the bit-rate while in the IBN system, the sampling rate and the bit-rate can be different. Thus for obtaining the IBN representation of a signal, a DM or DSM is preceeded by a sample-and-hold unit. A useful case is achievable when the sampling rate is equal to the Nyquist rate, when the resulting system is compatible with conventional binary systems. Also in this chapter the basic units needed for signal processing, namely the IBN adder, subtractor and multiplier, are developed. These arithmetic units are developed in such a way that their inputs and outputs are IBN-compatible i.e. both input and output remain in IBN form as opposed to earlier attempts outlined previously. In addition these arithmetic units can have much wider applications than in the hardware implementation of signal processors.

The effect of finite register length in these basic IBN arithmetic units is also considered in this chapter.

## 3.1 Digital Signal Processing Using DM or DSM, A Brief Review

The binary sequence obtained from a DM or a DSM has attributes and characteristics which are different from those of the conventional binary positional number. The DM or DSM binary numbers can be stored and delayed in the same way and by the same components as the conventional binary numbers, but due to their different nature, their processing and their conversion to analogue is somewhat different. In this section, after a brief review of the DM and DSM, some important techniques which employ these convertors are reviewed.

### 3.1.1. Delta and Delta-Sigma Modulation

Linear DM and DSM are the simplest form of analogue to digital conversion. A block diagram of a DM is shown in Fig. (3.1). Disregarding the low-pass filters which are needed at the input and the output of the coder and decoder in any type of binary convertor, the coder is constructed from a subtractor, a sign detector and an integrator. The decoder, which in general is a part of coder, is composed on an integrator.

The operation of the DM may be seen from the following simple approach: Let us assume that at time t=0, that is when the supply is turned on, the output of the DM is 1. At this moment, the output is integrated by the integrator and if the initial condition of the integrator is 0, then at the time $\Delta T$, the output of the integrator will be $\Delta V$. The output of the integrator is then subtracted from the input signal $x(t)$ and the sign of the difference signal constitutes the output of the DM. As shown in Fig. (3.2), the values of the sequence $L(n)$ at the output of DM will be in such a form that its integration follows approximately the input signal $x(t)$. The sign detector can be simply a D-type flip-flop and a simple

RC circuit can suffice as the integrator.

The DSM has a very similar structure to the DM and it is only slightly rearranged. Fig. (3.3) shows the block diagram of a DSM coder and decoder. The only difference between the coder of DM and the coder of DSM as can be seen in Figs. (3.1 and 3.3) is that in the DSM, the integrator is replaced from the feed-back loop to the forward loop. The decoder of the DSM is even simpler and consists of a low-pass filter only.

### 3.1.2. DM Non-Recursive Filter with Analogue Coefficients (76,77)

A non-recursive filter of order P is characterised by the following relation:

$$y(n) = \sum_{k=1}^{P} a_k x(n-k) \qquad (3.1)$$

where x(i) is the discrete input signal, $a_k$ are the filter coefficients and y(i) is the output of the filter. In the conventional method of realisation the input signal x(t) is converted to digital form by an A/D convertor and is then delayed by a P-tap delay line. The signal at the output of each delay unit is then multiplied by the filter coefficient using binary multiplier. A different approach for realisation of the non-recursive filter, which employes DM as the analogue to digital convertor, has been proposed by Lockhart(76,77). The major difference between this scheme and the conventional approach is that, in this scheme, resistors instead of digital multipliers are used for weighting the delayed signal. The block diagram of this form of realisation is shown in Fig. (3.4). This scheme results in a significantly simpler realisation, besides, the analogue-to-digital convertor, which in this case is DM, also adds to this simplicity. However, by this scheme only direct forms of non-recursive filters which are sensitive to filter coefficients can be realised. Also there will be problems caused by the tolerance of resistors and the choice of their correct values, since they are manufactured

in standard values.

A similar scheme has also been proposed by Lockhart[77] for the realization of DM recursive filters. This method is also applicable only for direct forms of realization.

### 3.1.3. Correlator Based on Delta-Sigma Modulator

In non-recursive filtering, as mentioned above, the sampled signal is multiplied by constants which are the filter coefficients. In this section a more general case is discussed in which two variable signals are multiplied. A multiplication scheme based on the DSM has been described by Lagoyannis[78] in the implementation of correlator. A similar scheme has also been described by Dempster and Steele[79] in the design of DSM Wattmeter.

A correlator can be expressed by Eq. (3.2) given below,

$$z(k) = \sum_{n=0}^{N-1} y(n)\, x(n+k) \qquad \begin{matrix} x(i) = 0 \\ \text{for } i > N \end{matrix} \qquad (3.2)$$

where x(n) and y(n) are two signals having correlation function given by z(n).

The block diagram of a correlator, based on the DSM, has been shown in Fig. (3.5). In this scheme, the signal (t) is converted to digital by a DSM and delayed by a P-tap delay line. The delayed signals then control switches which are either connected to (n) or $\bar{}$(n). The output of switches are low-pass filtered to give z(k). In the DSM Wattmeter[78], a more sophisticated DSM multiplier has been discussed in which two switches are incorporated. This multiplier reduces the effect of DSM idle noise.

### 3.1.4 Non-Recursive Digital Filters Implementation using DM

A completely digital technique for the realization of non-recursive digital filters using DM, has been proposed by Goodman[80] and Peled and Liu[51]. This method is based on a different multiplication scheme which is done only by addition and subtraction. The block diagram of this realization has been shown in Fig. (3.6). In this scheme a controlled negator, instead of multiplier, is employed which is controlled by the bit stream of the DM output. The input of each negator is the filter coefficient $a_k$ and its output is either $a_k$ or $\bar{a}_k$ depending whether the DM output is +1 or -1, respectively. The detailed circuit of the two's complement negator (complementor) has been given in[81]. These complementors can be combined with the adder of the next stage and the result will be either an adder or a subtractor. Comparison of Fig. (3.5) with Fig. (3.6) shows that the multiplication principle employed in this scheme is identical to that discussed in section 3.1.3.

## 3.2 Intermediate Binary Numbers

In this chapter a new system of binary numbers, referred to as Intermediate Binary Numbers (IBN), is defined and units to carry out arithmetic operations for these numbers are developed.

The IBN representation of a discrete value is non-positional and it obeys the following rule. With B bits, in this representation we can represent a discrete value which can have, between +1 and -1, B+1 different values. If S is a value between $\pm 1$ then the IBN representation of S is given by

$$S = \frac{1}{B} \sum_{i=0}^{B-1} w_i \qquad (3.3)$$

where $w_i$ is the IBN digits having weights of $\pm 1$. Eq. (3.3) shows

that S can have B+1 values ranging from -1 to +1.

The IBN representation has the following properties:

(i) The representation is defined for both positive and negative numbers. It should be noted that the conventional binary representation is only defined for positive numbers and for representation of negative numbers, other systems such as one's complement or two's complement must be employed.

(ii) The negative of an IBN can be simply obtained by inversion of its digits, as shown below

$$-S = \frac{1}{B} \sum_{i=0}^{B-1} \bar{w}_i \qquad (3.4)$$

where $\bar{w}_i$ is -1 when $w_i$ is 1 and vice versa. This symmetry property of IBN is very important, and as will be seen later, it leads to a simple IBN subtaction scheme. The conventional binary representation, which employs either two's complement or one's complement representation for negative numbers, does not have this property and the negation is obtained after a relatively complicated process[81].

(iii) The analogue-to-IBN and IBN-to-analogue conversion is simpler than PCM. The analogue-to-IBN conversion can be simply accomp lished by linear DSM (linear DM in some cases) preceeded by a sample and hold, as will be shown later. The IBN-to-analogue conversion can be done simply by low-pass filtering.

(iv) An IBN representation of a number is not unique, i.e. several combinations of 1 and 0 can produce the same representative of one number. This can be easily seen from Eq. (3.3), since it is the summation that matters in this case rather than the position of the $w_i$ bits.

(v) The IBN representation has the advantage that its error analysis is very similar to that of conventional binary numbers. In most cases, by using a simple transformation as $2^b \longrightarrow B$, the error in

the IBN case can be derived from that in the binary case, where b is the word-length of the binary positional representation and B is word-length in IBN representation.

(vi) IBN can be converted to the binary form by simply using an up-down counter. The reason that this type of representation is referred to as intermediate binary numbers is that these numbers can be easily converted to the analogue or to the binary form.

## 3.3 Analogue to IBN Conversion and Vice Versa

Analogue-to-IBN conversion can be implemented using a DSM preceeded by a sample-and-hold circuit. The block diagram of an analogue to IBN convertor is shown in Fig. (3.7). In this scheme, the analogue signal x(t) is first passed through a sample-and-hold circuit and the output of the sample-and-hold is then digitised by a DSM. In the following we show that w(n), i.e. the output of the DSM in this case satisfies Eq. (3.3). Referring to section 3.1.1 and to the DSM block diagram of Fig. (3.3), the feedback in the DSM circuit attempts to reduce the integral of the signal d(t) which is the difference between the input signal x(t) and the output sequence w(n).

$$\int x(t)dt - w(n)dt = \int d(t)dt = e(t) \quad (3.5)$$

Since w(n) is discrete valued, its integral can be written in the summation form,

$$\int x(t)dt - \sum_n w_n = \int d(t)dt = e(t) \quad (3.6)$$

In the ideal case, i.e. a free error DSM, e(t) can be assumed to be zero. In Fig. (3.7), the input of the DSM is the output of the sample-and-hold which is constant during each sampling interval. In this case Eq. (3.6) can be written as

$$\int_{mT}^{(m+1)T} x(m)dt = \sum_{n} w(n) \qquad (3.7)$$

since x(m) is constant during sampling intervals, the value of the left hand side of Eq. (3.7) will be a constant multiplied by x(m)

$$C.x(m) = \sum_{n} w(n) \qquad (3.8)$$

If the constant C is taken to be B, i.e., the word length of IBN, then Eq. (3.8) and Eq. (3.3) will be identical.

If the DM rather than the DSM is used in Fig. (3.7) then the digitised signal will represent the difference between adjacent samples in the IBN form. In fact the characters of the DSM and DM is similar to that of PCM and DPCM, respectively.

The IBN to analogue convertor is extremely simple and consists of only a low-pass filter as shown in Fig. (3.7).

## 3.4 IBN Adder and Subractor

This section discusses a criterion for adding or subtracting two IBN to produce the sum or difference of two numbers in IBN form. The adding algorithm of two IBN is as follows: When two digits are equal, the sum digit is equal to them and when they are different (one of the 1 and the other 0), the sum digit is equal to the carry bit and the carry bit is then complemented.

The sum obtained by this procedure is one half of the actual sum of two numbers. The difference algorithm has also this property of producing one half of the actual difference. This property is the main difference between IBN arithmetic units and the binary positional arithmetic units when realised in hardware. The main importance of this property in addition and subtraction is that the sum and the difference are always ensured against overflow. An example is given in Fig. (3.8) to illustrate the above. In this example two numbers $\frac{1}{4}$ and $\frac{3}{4}$ quantized to 8 bits each,in the IBN system,are added as shown in Fig. (3.8). Starting with the right hand side digits, and assuming 0 initial condition for the carry bit, the two first digits of the two numbers are 1, and hence the sum digit will be 1. In the second set of digits, one digit is $\underline{1}$ whilst the other is 0, so that the second sum digit is the carry bit which is zero which is complemented to become the carry bit 1 for the next operation, and so on. The result as shown in Fig. (3.8) is 2/4 which is half of the actual sum, i.e., 4/4

Since the negative of each IBN is obtained by inversion of its digits, thus the IBN subtractor is an IBN adder plus a logic invertor which inverts the digits of the subtrahend.

An efficient implementation of IBN adder and subtractor is shown in Fig. (3.9) and Fig. (3.10), respectively.

The adder circuit is composed of an exclusive OR gate which compares the digits of two numbers, the J-K flip-flop is for saving the carry bit, and a two input one output multiplexer which acts as a controlled logic switch. Truth tables of the ex-OR and the multiplexer are shown in Fig. (3.11) and Fig. (3.12), respectively.

The serial IBN adder and subtractor have several advantages over the well-known and most popular two's complement adder and subtractor:

(i) In the IBN adder and subtractor no overflow occurs and the results always fit within the word. This eliminates the hardware necessary for overflow checking.
(ii) The IBN adder and subtractor need no carry clearing at the beginning of each operation whereas this is necessary in the two's complement adder and subtractor.

(iii) Further gap exists between the IBN and two's complement subtractors, since the IBN adder and subtractor are almost identical, the difference is only in one invertor, while the difference between two's complement adder and subtractor is substantially greater, as given in[81].

Since the IBN sum and difference algorithms above involve an inherent division by 2, an extra noise will be added to the result of each operation. Assuming a B bits word length and that the value of each sample ranges between $\pm 1$, then the step size which can be obtained from Eq. (3.3) is $2/_B$ . Thus the numbers represented in the IBN notation range over $T/_B$ Where T is an even number. For an operation which will need an odd T value, T will be changed to closest even number, i.e., T+1 or T-1 depending on the initial condition of the carry bit. The error which is involved in this operation is given by the following equation:

$$\sigma_T^2 = \frac{1}{2} \cdot \left(\frac{1}{B}\right)^2 = \frac{1}{2B^2} \qquad (3.9)$$

where $\frac{1}{2}$ is the probability of this error to happen and $1/_B$ is the amount of error at each time that this error happens.

In two's complement addition and subtraction, no error is involved in the operation by itself but since the signal must be attenuated before operation, the operation will involve some error. It has been shown in the implementation of FFT[24] and FWT[94] (fast Walsh transform) that instead of attenuation at the input, if the result of each stage of algorithm is divided to 2, then significant improvement in the output signal to noise ratio will be achieved. Such operations, therefore, are eminently suitable for IBN implementation.

## 3.5 Serial IBN Multipliers

Multipliers are critically important in the implementation of digital signal processors. Different designs of binary multipliers exist[75]. These designs involve generally a trade-off between speed and complexity. The most adequate multiplier configurations are those which multiply the digitized signal of serial form with a binary number in parallel form which can be coefficients of digital filters or samples of orthogonal functions. In this chapter the development of IBN-Binary multiplier (IBN-BN) is discussed. Two types of IBN-BN multipliers are in fact described: (a) The IBN-BN multiplier with a binary output; (b) The IBN-BN multiplier with an IBN output form. The former type which employs only binary addition and subtraction can be used for the implementation of orthogonal transforms in their direct form only. The second type in which the product is in IBN form, can be employed for either the direct form or indeed for the fast form of implementation of the transform. The latter type is based on the property of the IBN adder and subtractor that produces one half of the sum and difference without the need to carry out a shift. The errors created by these multipliers are discussed in the next sections

### 3.5.1 IBN Multiplier with Binary Output

This type of multiplier which multiplies an IBN in the serial form by a binary number in the parallel form, employs only additions and subtractions (no shifting). A number S represented in the IBN system, when multiplied by a number t, will result in the following relation:

$$S.t = t \cdot \frac{1}{B} \sum_{n=0}^{B-1} w(n) \qquad (3.10)$$

or

$$S.t = \frac{1}{B} \sum_{n=0}^{B-1} t.w(n) \qquad (3.11)$$

since w(n) is either $\pm$ 1, the right hand side of Eq. (3-11) can be determined by B operations of additions or subtractions (ignoring the scaling factor $^1/B$.

A circuit for performing Eq. (3.11) is shown in Fig. (3.13). In this design, t is in the positional binary form and is stored in a circulating shift register. The accumulator contains the product after B operations where B is the word length of the IBN representation.

The above described multiplier is similar to that discussed by Goodman[80] and Peled and Liu[51], but the approach presented here is somewhat different.

### 3.5.2. IBN multiplier with IBN output (Cascade IBN Multiplier)

The novel digital multiplier, presented in this section, enables the implementation of a large groups of digital signal processors based on IBN techniques. This multiplier, multiplies a serial IBN with a parallel binary number where the product is in IBN form and can be processed again by IBN processors. Let the binary number be

expressed by t and assumed to be between -1 and +1. If t is represented by b bits, then t satisfies the following equation:

$$t = \sum_{i=1}^{b} v_i \, 2^{-i} \qquad (3.12)$$

where $v_i$ can be either +1 or -1 (this is zero in binary logic). This representation is slightly different from the conventional binary representation as it can represent both positive and negative numbers.

When t is multiplied by the number S which is in the IBN form, the result can be obtained by using Eq. (3.3) and Eq. (3.12) in the form:

$$S.t = \left[\sum_{n=0}^{B-1} w(n)\right] \cdot \left[\sum_{i=1}^{b} v_i \, 2^{-i}\right] \qquad (3.13)$$

Eq. (3.13) can be rearranged to give

$$S.t = \sum_{n=0}^{B-1} w(n) v_i \, 2^{-i} + w(n) v_{i-1} \, 2^{-(i-1)} + w(n) v_{i-2} \, 2^{-(i-2)} + \cdots\cdots \qquad (3.14)$$

or

$$S.t = \sum_{n=0}^{B-1} \left\{\left[w(n) v_i \, \bar{2}^1 + w(n) v_{i-1}\right] \bar{2}^1 + w(n) v_{i-2}\right\} \bar{2}^1 + \cdots \qquad (3.15)$$

Eq. (3.15) can be implemented simply by using exclusive-OR gates and IBN adders. The product $w(n).v(i)$ can be carried out by an exclusive-OR since they are bivalued. In the first stage, the output of the ex-OR which multiplies w(n). v(i), is one input of the first IBN adder and the other input of IBN adder is set equal to Zero.

Thus the output of the adder will be $w(n).\ v(i).\ 2^{-1}$. In the second stage, one input of the adder is the output of the ex-OR which produces $w(n).v(i-1)$ and the other input is the output of the IBN adder of the previous stage. Thus in this type of multiplier we need b stages in which each stage consists of an IBN adder and one ex-OR gate. The structure of this multiplier is shown in Fig. (3.14).

For comparison, a cascade binary multiplier is also shown in Fig. (3.15). This multiplier also employs a division by 2 scheme at the output of each stage. Comparison of Fig. (3.14) and Fig. (3.15) shows the two important advantages of the cascade IBN multiplier over the cascade binary multiplier: (i) The hardware of the former multiplier is significantly simpler since the one bit delay at the output of each stage is eliminated and also no control hardware is necessary for division by 2. (ii) Unlike the cascade binary multiplier, there is no delay between the input and the output in the cascade IBN multiplier and consequently there is greater modularity in these multipliers.

### 3.5.3. Error Analysis of IBN Multipliers

In the first type multiplier, discussed in Section 3.5.1, if the final word-length of the number represented in the IBN system is B bits, then the capacity of the accumulator of Fig. (3.13) must be at least $b+\log_2 B$ bits in order to avoid overflow. By rounding the accumulator to b bits, the amount of the error due to this rounding will be:

$$E_1 = \frac{2^{-2b}}{3} \qquad (3.16)$$

In the special case when $B=2^b$, this error becomes:

$$E_1 = \frac{1}{3B^2} \qquad (3.17)$$

In the second type of multiplier, discussed in section 3.5.2, the error arises from the division by 2 at each stage. The error produced by each stage will be halved before entering the next stage. If the error of each stage is $E_s$, then the total error at the output of the multiplier will be given by:

$$E_2 = \sum_{i=0}^{b-1} E_s \, 2^{-i} \qquad (3.18)$$

where b is the word-length of the parallel binary number. $E_s$ in this case is the error of the IBN adder which is given by Eq. (3.9), thus the error of multiplication can be approximated by:

$$E_2 = 2E_s = \frac{1}{B^2} \qquad (3.19)$$

## 3.6. Discussion

In this chapter a new technique for the implementation of a large group of digital signal processors has been proposed. This technique is mainly aimed at the implementation of orthogonal transforms and digital speech signal processors. Nevertheless, this technique can be employed for other types of processing for low-bandwidth signals. This technique is based on a new digital representation which is referred to as Intermediate Binary Numbers (IBN). In this method the analogue signal is first converted to digital by a linear delta-sigma modulator or in some cases by linear delta-modulator which are convertors significantly simpler than the well known A/D convertor. The digital signal is then processed by IBN arithmetic units and finally it is converted to its analogue form simply by low-pass filtering. The main advantage of the new approach presented in this chapter as compared with similar methods employing delta modulators is that the output of the developed IBN arithmetic units is also in IBN form and compatible within an IBN realisation i.e. it can be

further processed by IBN arithmetic units. This enables the implementation of complicated algorithms by the IBN technique. Another advantage of the IBN implementation technique is the modularity of IBN arithmetic units and the simplicity of their control units. The IBN adder and subtractor are significantly simpler than those corresponding to two's complement realization since they do not need any carry clearing at the beginning of each operation. Moreover the IBN adder can be converted to the IBN subtractor simply by inverting the subtrahend. Besides, the IBN adder and subtractor are always ensured against overflow and hence most importantly no signal scaling is necessary in this case.

The IBN multiplier, reported in this chapter, reveals very interesting properties. This multiplier offers a very simple way of multiplication in terms of its structure and required hardware. More importantly, there is no delay between the input and the output of these multipliers, i.e., when the first bit of the input enters the multiplier, the first bit of product leaves the multiplier. This, in addition to providing more modularity, increases a great deal the relative speed of the implemented digital signal processor.

Error analysis of the proposed arithmetic units is also discussed in this chapter and this can form a basis for the error analysis for the overall IBN signal processors.

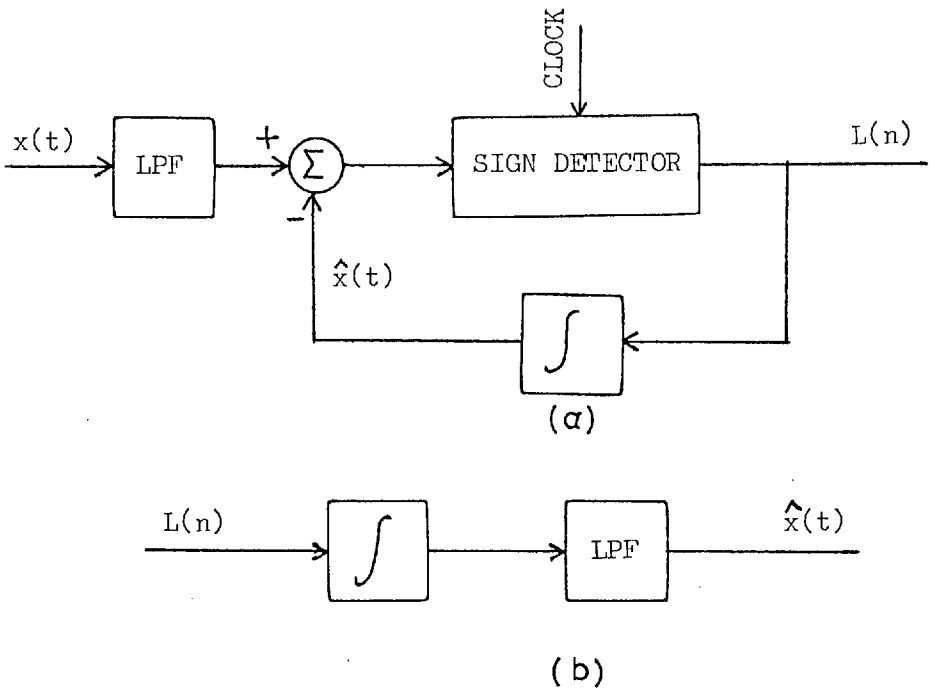



Fig. 3.1. Linear delta modulation system:(a)encoder,(b)decoder

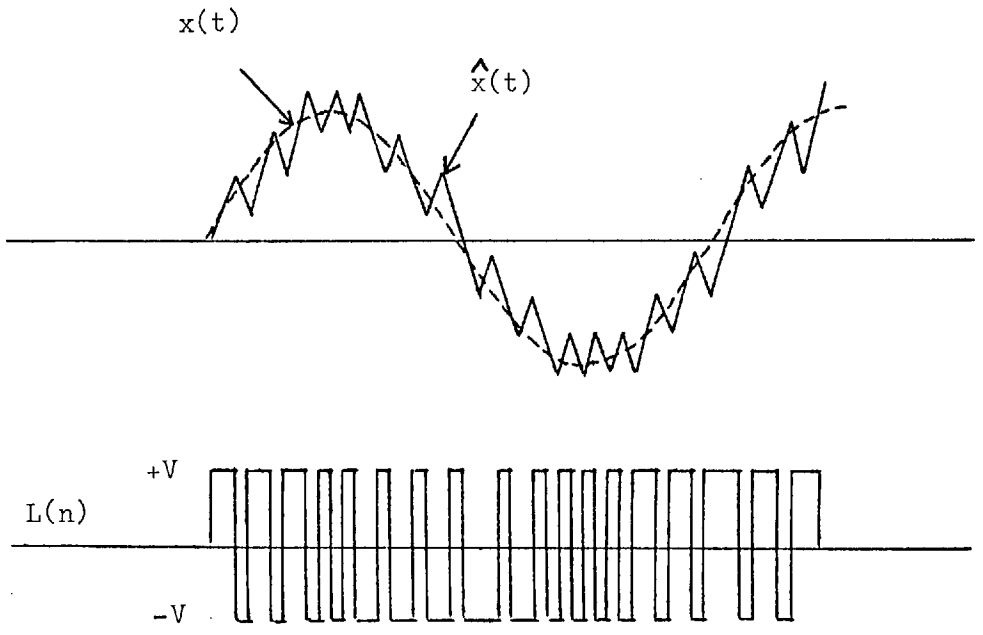



Fig. 3.2. Linear delta modulator waveform

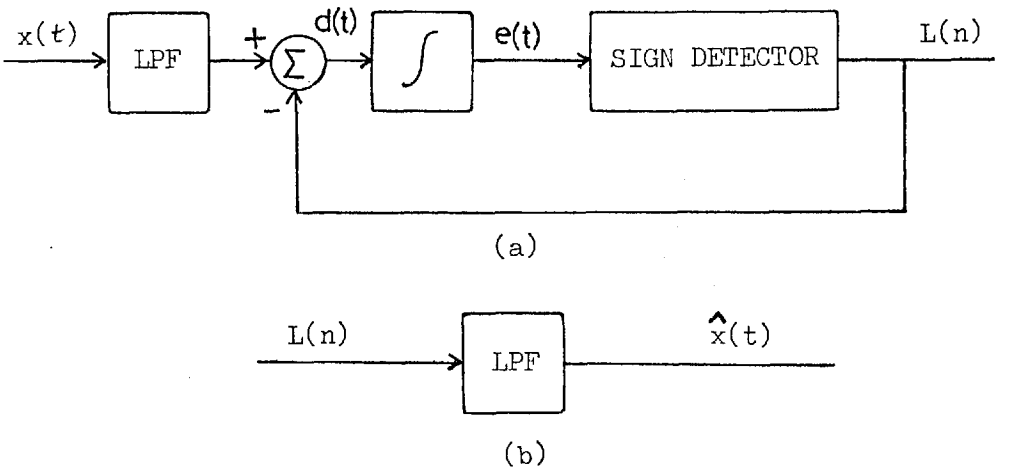



Fig. 3.3. Delta-sigma modulation system:(a) coder (b)decoder

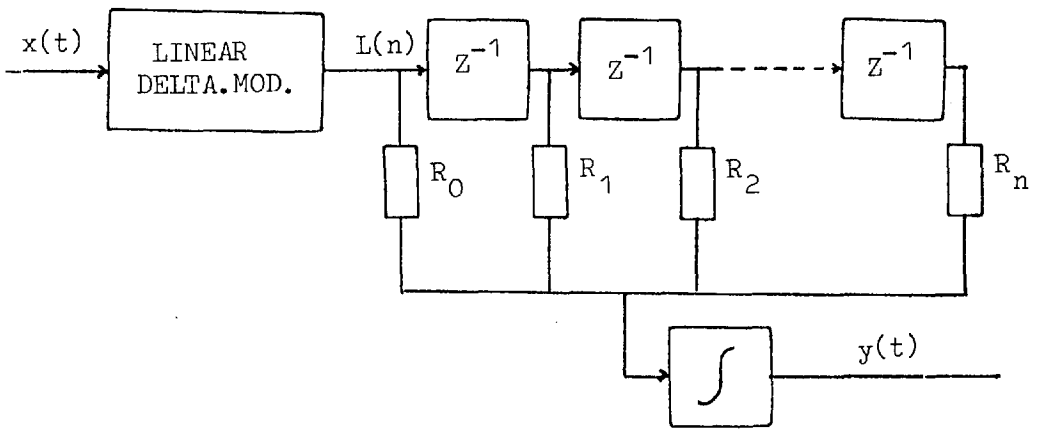



Fig.3.4. Delta modulator non-recursive filter using weighted resistors

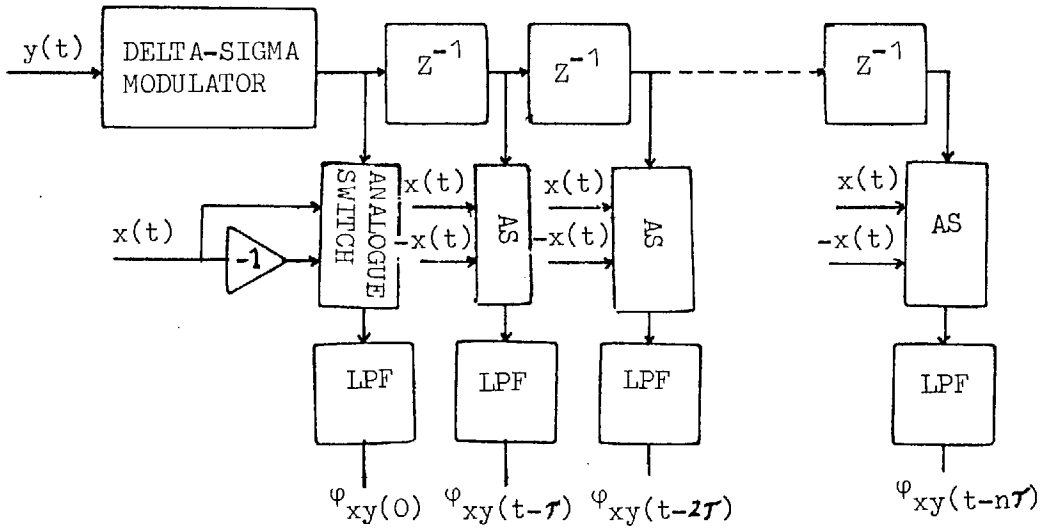



Fig.3.5. Correlator based on delta-sigma modulator

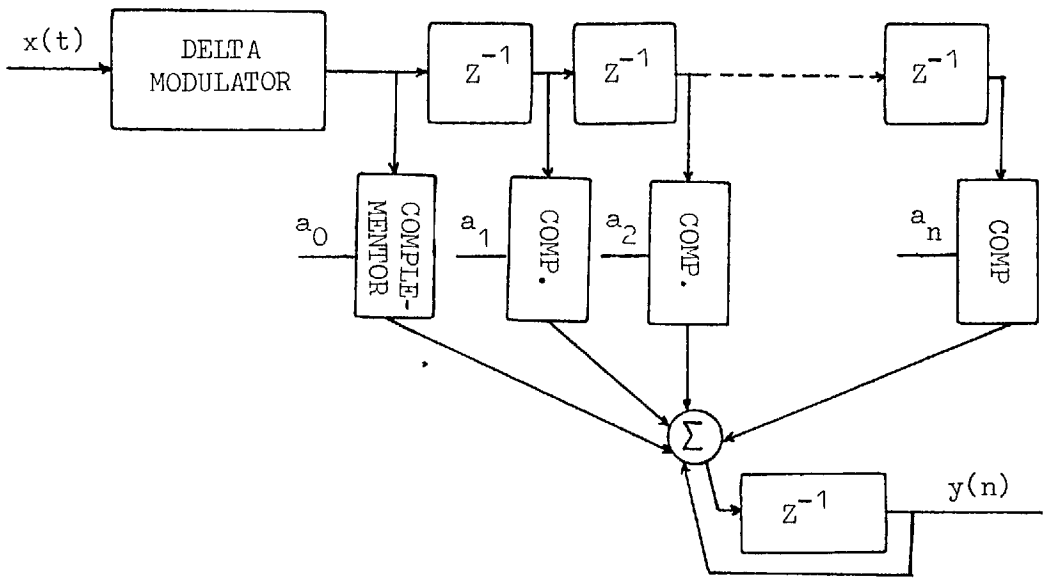



Fig.3.6. Digital non-recursive filter realisation using delta modulation

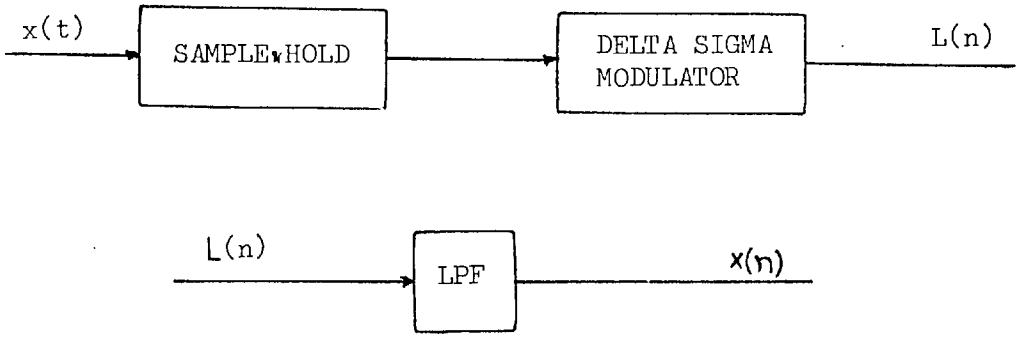



Fig.3.7. Analogue to IBN convertor and vice versa

| | | |
|---|---|---|
| 1 1 0 1 0 1 0 1 | $\frac{1}{4}$ | AUGEND |
| 1 1 1 0 1 1 1 1 | $\frac{3}{4}$ | ADDEND |
| 1 1 1 0 1 1 0 1 | $\frac{2}{4}$ | SUM |
| 1 0 0 1 0 1 1 0 | | CARRY |

Fig.3.8. An IBN addition example

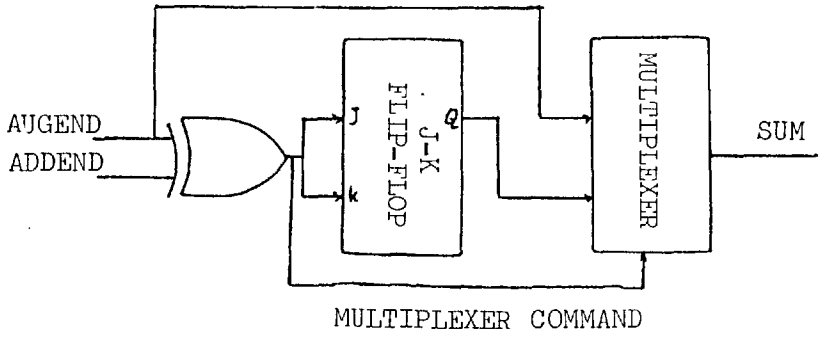



Fig. 3.9. Serial IBN adder

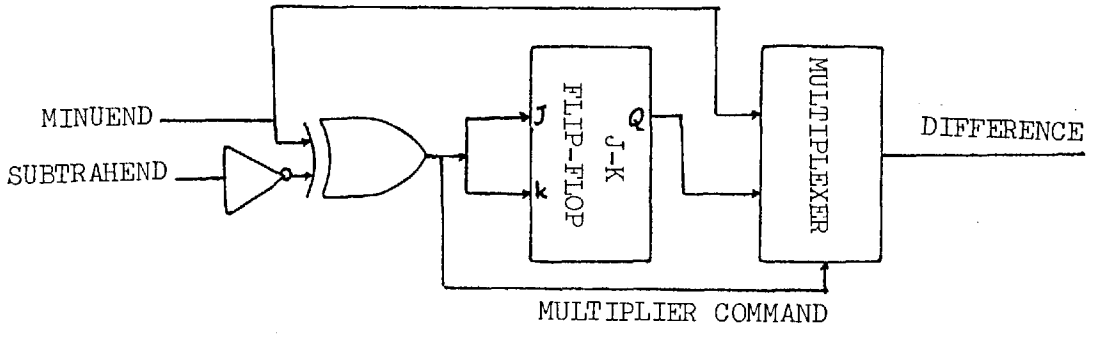



Fig.3.10. Serial IBN subtractor

| Input 1 | Input 2 | Output |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Fig.3.11. Exclusive-OR gate truth table

| Command | Input1 | Input2 | Output |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Fig.3.12. 2-input multiplexer truth table

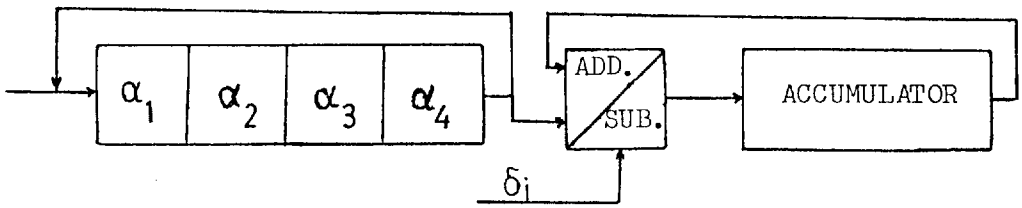



Fig.3.13. IBN multiplier with binary output
($\alpha_i$)digits of the binary number
($\delta_i$)digits of the IBN

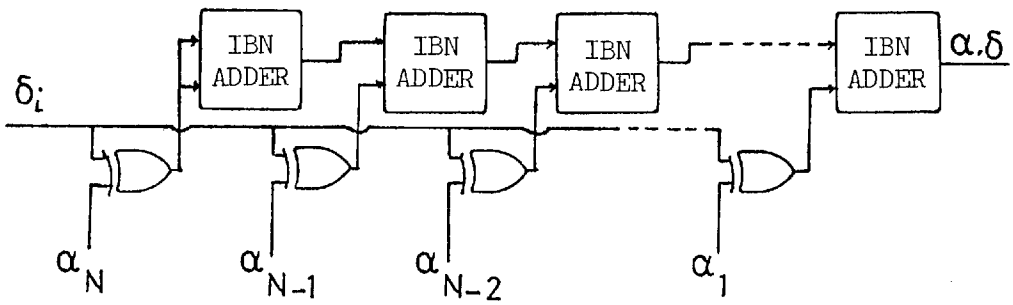



Fig.3.14. IBN multiplier with IBN output
(cascade IBN multiplier)
($\alpha_i$)digits of the binary number
($\delta_i$)digits of IBN

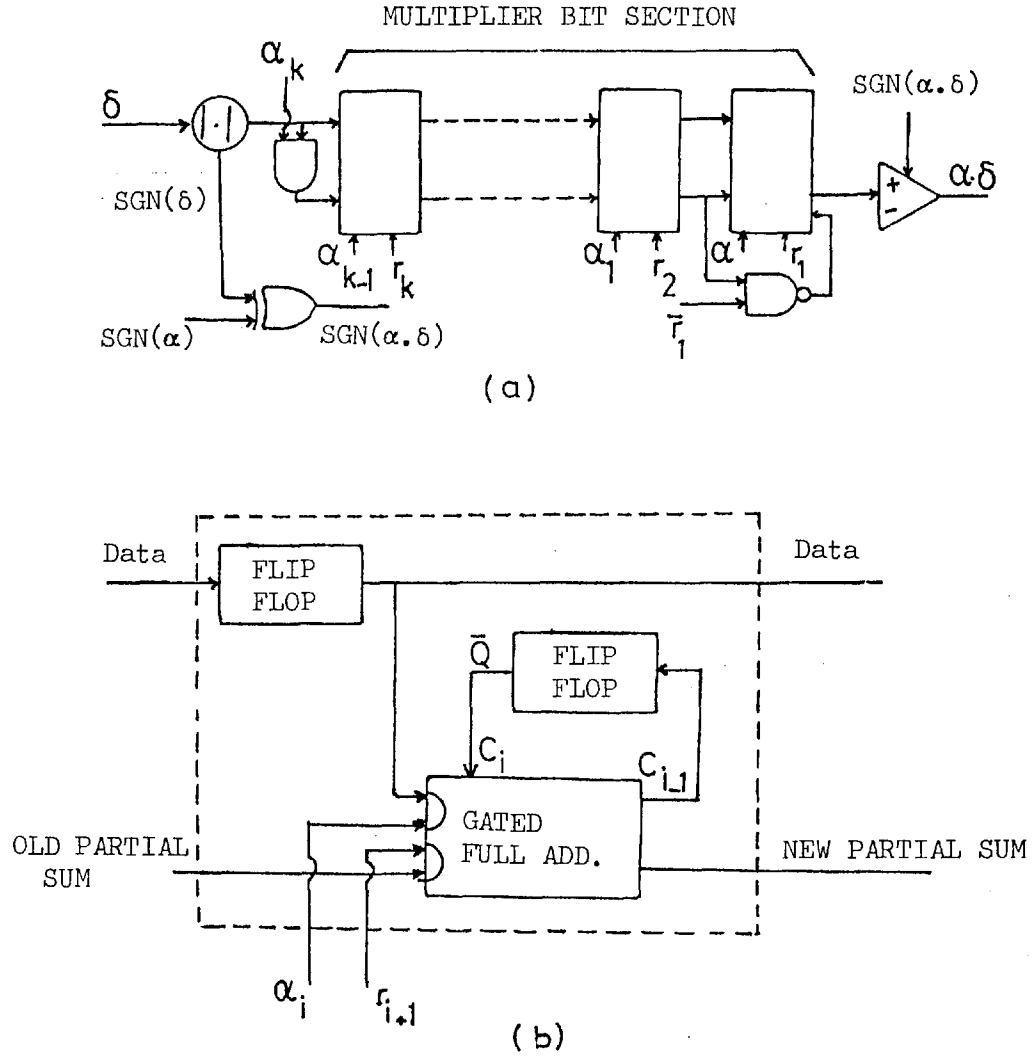



Fig.3.15. (a) Cascade binary multiplier
(b) A multiplier bit section

# CHAPTER 4

## Orthogonal Transform Implementation Using Intermediate Binary Numbers

Recent years have witnessed an increasing interest in orthogonal transforms for digital signal processing. In addition to the different methods proposed for signal processing by orthogonal transforms, many authors have also discussed various approaches for effecient implementations of these transforms. Amongst the different types of orthogonal transforms that have been known so far, the Discrete Fourier Transform (DFT), Walsh-Hadamard Transform (WHT) and Discrete Cosine Transform (DCT) have found the greater number of applications[31]. Much of the effor has been concentrated to wards the efficient implementation of these transforms. Fourier transforms have been used for many decades for the analysis of signals and systems. In the last two decades, because of extensive application of digital computers and the development of reliable and inexpensive logic components, the discrete form of the Fourier transform, i.e. the DFT, has substituted the continuous one. In the various methods of DFT implementation in which different fast algorithms have been employed, the aim has been mainly to develop a high speed Fast Fourier Transform (FFT) Processor which can be used as a part of a general purpose computer[82-90]. This computer would then enable an increased speed for FFT processing.From the data compression point of view, the WHT shows efficiencies similar to the FFT, but as far as the implementation is concerned, the WHT is well known to be much more efficient. The DCT is described by Ahmed et al[37] as the optimum signal independent transform for creating the most non-uniformity in the variance distribution of speech signals[44]. This transform has also been very efficient for image data compression[31].

Several implementation approaches of the WHT which employ different fast WHT algorithms have been described of late[91-92]. It has been shown that the straightforward $N^2$ computation algorithm can be implemented as efficiently as the fast WHT algorithm, when the

processing rate is relatively low. In all of the above approaches either an implicit or an explicit assumption has been made that their realisation is to be done by conventional PCM and Two's Complement Arithmetic Units (TCAU). Recently the Charged Couple Device (CCD) and Acoustic Surface Wave (ASW) delay lines have been employed for the FFT and DCT implementation[93].

The new speech coding and decoding method, described in Chapter 2 of this thesis, is composed of two parts. The first part deals with the transformation of speech signals to WH or DC domain (in the coder) or inverse transformation(in the decoder). The second part is concerned with the modelling and inverse modelling of the transformed speech. The implementation of the latter part of the proposed system, is comparatively simple and involves the computation of a few prediction coefficients. This can be accomp lished by a few additions and multiplications and because of its simplicity the implementation of this part is not discussed further. The first part, i.e. the part related to the transformation or inverse transformation, involves more operations, since in this process, the data vector is multiplied by the Hadamard or DC matrices.

In this chapter the implementation of the WHT and the DCT is investigated and an implementation is suggested. The proposed implementation employs DSM and IBN Arithmetic Units ( IBNAU) as developed in Chapter 3. Three approaches of the WHT implementation are considered where two of them are based on the straightforward $N^2$ computation algorithm whilst the third implementation employs the fast WHT algorithm. Implementation of the DCT, based on the use of DSM and IBNAU is also described in this chapter.

Error analysis for all the proposed structures is discussed and the results are compared. The digital realisation circuits obtained by the different implementation techniques are simulated on a digital computer to confirm the theoretical results. Finally the different

implementation approaches are compared from a "Signal to Noise Ratio (SNR)" performance point of view.

## 4.1 Implementation of Orthogonal Transforms, A Brief Survey

Since this chapter is mainly concerned with the implementation of the WHT and DCT, those methods which are centrally related to these or have some bearing to the problem are reviewed in this section. In the first part of this section a number of important implementation methods of DFT are reviewed. The main reason for the study of DFT processors is that there exist great similarities between the Fourier and Walsh transform. Besides, a DFT transformer can be used directly for obtaining the DCT coefficients of a signal, since DCT coefficients are real parts of the DFT coefficients[31]. In the second part of this section, some existing methods of WHT implementation are discussed. Some of these methods are similar to those already employed for the DFT whilst others are based on some specific properties of the WHT. The final part of this section is devoted to the implementation of DCT.

### 4.1.1 Implementation of the Discrete Fourier Transform

The Fourier transform has been traditionally very important in the analysis of signals and systems. The traditional way of obtaining the frequency spectrum of a signal is achieved by using a bank of analogue band-pass filters. In recent years, because of the replacement of analogue systems with their digital counterparts and also the development of entirely independent digital systems, the DFT rather than the continuous Fourier transform has been employed. The implementation of the DFT can be based on the recently developed basic digital signal processing elements or indeed special purpose hardware based on the fast computational algorithms, reducing computational time and cost of the hardware realisation, can be employed.

The DFT and the Inverse DFT (IDFT) are defined by Eq. (4.1) and Eq. (4.2), below[37]

$$Y(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cdot w^{kn} \qquad (4.1)$$

$$x(n) = \sum_{k=0}^{N-1} Y(k) \cdot \bar{w}^{kn} \qquad (4.2)$$

where $w = e^{-j\frac{2\pi}{N}}$, $j = \sqrt{-1}$.

Various FFT algorithms have been reported in literature[75] which enable the computation of Eq. (4.1) and Eq. (4.2) to be done by $\frac{N}{2}\log_2 N$ Basic Operation (BO) (one complex multiplication and one addition and subtraction) instead of the direct approach requiring $\frac{1}{2}N^2$ BO. Amongst the various forms of FFT algorithms, the Cooley-Tukey algorithm is known as the most suitable form for the FFT implementation[75]. This algorithm for N=8 is shown in Fig. (4.1). As shown in this Figure, this algorithm consists of 12 BO. In the implementation method, known as the sequential method[83], one Basic Operation Unit (BOU) is used sequentially to perform the computation of this algorithm. In another scheme, known as the Cascade Processor[84,85], 3 BOU are employed, in which each BOU performs one iteration of the algorithm. The former processor is about 100 times whilst the latter one is 500 times faster than a general purpose computer performing an FFT[82]. In the parallel iterative processor[86,87], 4 BOU are employed, where these units compute the operations of each iteration in parallel whilst the iterations are performed sequentially. In the method in which the algorithm is realised in an array form, by using 12 basic operation units[88], the speed may be increased to 50,000 times more than that corresponding to a general purpose computer employed for the same purpose.

### 4.1.2 The Pipeline form of the FFT Processor

The pipeline FFT processor can process the samples of a signal at the same rate as they appear at its input. This kind of processor as described by Groginsky and Works[89], consists of $n=\log_2 N$ stages where each stage performs one iteration of the Cooley-Tukey algorithm. Each stage is composed of a BOU, a discrete-time delay line, and 4 switches, as shown in Fig. (4.2). The delay line of the first stage for an 8-point FFT operation consists of 4 delay units ($Z^{-1}$) and the second and third stages employ 2 and 1 delay units, respectively. The switches are controlled by a binary counter in such a way that the most significant bit controls the first stage and so on. Thus in a processor of order 8, a 3 bits binary counter is needed, as shown in Fig. (4.3). At the beginning of the process, the switches connect the input data to the input of the delay line of the first stage. Thus the first 4 samples are stored in the delay line. During the second 4 samples, the switches connect both the original input and the output of the delay line to the input of the BOU. The BOU has two outputs of which one produces the first 4 samples of the first iteration whereas the other output gives the second 4 samples of the iteration . The first output of the BOU is then connected to the input of stage 2 whilst the second output of the BOU is connected to the input of the delay line, i.e. the input to stage 1. The content of the delay line, i.e. the second 4 samples of the first iteration, enter the second stage during the first 4 samples of the next block. By this technique, the first stage of the processor performs the first iteration of the algorithm. The second and the third stage have similar performance and implement the 2nd and 3rd iteration of the algorithm, as shown in Fig. (4.3).

Finally it is worth noting that recently, Liu and Peled[90] have introduced a new technique of multiplication which employs Read Only Memory (ROM). By this technique a substantial increase in the speed of the Pipeline FFT Processor, can be achieved.

### 4.1.3 Implementation of the Walsh-Hadamard Transform

Walsh-Hadamard transform as discussed earlier in this thesis, has received increasing attention in the area of signal processing because of two reasons, (i) the similarities to the well known Fourier transform, (ii) the efficiency in computation or hardware implementation. The real time WH processors, in general, can be divided into two categories (a) Direct form WH processors in which the $N^2$ computation algorithm is employed. (b) Fast WH processors which take advantage of the fast WHT algorithms. Both of these categories are included below:

(a) $N^2$ Computation WHT processor: For obtaining one of the WH coefficients, N operations of addition or subtraction on the samples of the block to be transformed are needed. In a straightforward implementation method, described by Gethoffer[46], a shift register and a controlled adder/subtractor were employed, as shown in Fig. (4.4). At the beginning of a block, the shift register is cleared to zero. The samples of the input block are then added or subtracted from the content of the shift register. The choice of addition or subtraction depends only on whether the sample of the Walsh function corresponding to the input sample is +1 or -1. After N additions or subtractions, the shift register will contain the WH coefficient and is unloaded either in a parallel form or in a high speed serial form, before the first sample of the next block.

(b) Fast WHT Processor: In the $N^2$ computation form of implementation, N adder/subtractor units (N is the number of samples per block) and N shift registers with each having a length of N samples are needed. In a slightly different structure, it is possible to recirculate the input at a rate of N times more than the real time rate and use only one adder/subtractor unit, but in this case, the maximum speed of the process will be reduced by an order of N, since the adder/subtractor unit and the shift register must operate N times

faster than that of the former case.

A more common way of WH implementation is using a fast algorithm. Implementation of the fast WHT allows the number of adders and subtractors to be significantly reduced while keeping the same speed of processing as the $N^2$ computation scheme. One method of fast WHT implementation has been proposed by Walker[91] and Wintz[43] and is shown in Fig. (4.5). In this method, the number of adders and subtractors is reduced from N to $2N\log_2 N$ and $2(N-1)$ delay units are employed. The operation of this processor is similar to that of the Pipeline FFT processor discussed in Section 4.2.1. As will be shown later, the number of delay units in this system can be halved by doubling the number of switches.

### 4.1.4 Implementation of Discrete Cosine Transform

The fast computational algorithms of the DCT proposed by Ahmed et al[37] and Haralick[38] is based on the FFT algorithms. As shown in [37,38], the real values of a 2N point FFT processor output gives the DCT coefficients. Thus, a real time FFT processor can also be used as a real time DCT transformer. The recent fast DC algorithm described by Chen[39], which is only concerned with real numbers, may lead to a new technique of DCT implementation. Recently, CCD and ASW have been incorporated in the implementation of DCT[93]. This technique employs the Chirp-Z-transform DFT algorithm in which the convolution property of DFT is utilised.

## 4.2 Implementation of Walsh-Hadamard Transform using Intermediate Binary Numbers

In section 4.1.3, two implementation structures of the WHT were reviewed. The implementation of these structures by IBN technique, i.e., using DSM and IBNAU, is the purpose of this section. It is shown that the first structure which employs the straight-forward $N^2$ computation algorithm, can be implemented by means of

the IBN technique in a significantly more efficient way than using two's complement arithmetic. In this case, addition or subtraction can be performed simply by an ex-OR gate. The implementation of the fast WH algorithm using IBN techniques is discussed in Section 4.2.2. The fast structure used in this chapter, employs N-1 delay units ($Z^{-1}$), unlike the method of Section 4.1.4 in which 2(N-1) delay units are needed. This structure, which makes a more efficient use of delay units, is based on the approach adopted in the implementation of the Pipeline FFT, discussed in Section 4.1.2. Using IBN adders and subtractors in this structure leads to more modular WH transformer than using two's complement adders and subtractors. This is mainly because of the inherent modularity of the IBN adder and subtractor. In addition the division by 2 characteristics of the IBN adder and subtractor significantly improves the SNR performance of the processor. The WH transformer employing an IBN system realisation has two disadvantages compared with the realisation based on a binary positional realisation (such as two's complement). (i) Transformation of only low-bandwidth signals is feasible by this processor due to excessive logic speed required in the implementation. (ii) The word-length required in the processor for an equivalent kind of performance in comparison to a binary realisation is longer. The first disadvantage is not relevant to speech signals which are inherently of low-bandwidth. Moreover, recent advances in the LSI technology produced shift registers with a large number of bits at rather inexpensive prices and hence the second disadvantage can be eliminated.

### 4.2.1. WHT Implementation Using the $N^2$ Computation Algorithm

For the purpose of transforming a signal to the WH domain, the signal is first sampled and divided into intervals whereby the number of samples in each interval is an integer power of 2. The corresponding transform vector Y is then obtained from the following relation:

$$Y(k) = \sum_{i=0}^{N-1} x(i) \cdot Wal(i,k) \qquad (4.3)$$

Where Wal(i,k) represents the Walsh function samples that take values of either +1 or -1. For the computation of Y(k),N operations of addition or subtraction is needed as is evident from Eq. (4.3). This equation can be realised by a controlled negator and an accumulator. The negator negates x(i) whenever Wal(i,k) is -1. The output of the negator is then accumulated to yield Y(k). The block diagram of this processor is shown in Fig. (4.6).

The implementation of this block diagram by means of a TCAU is shown in Fig. (4.8). In a two's complement implementation the negator is referred to as a "two's complementer"[81] which is composed of a J-K flip-flop, an AND gate, an interverter gate and a logic switch. The accumulator, in this case, consists of a two's complement adder[81] and a shift register.

The realisation of the block diagram of Fig. (4.6) using the IBN system is shown in Fig. (4.7). The controlled negator in this case is simply an ex-OR gate which inverts the digits of the input signal whenever the other input, which is Wal(i,k), is -1. In this case an up-down counter performs the accumulation. Comparison of Fig. (4.7) and Fig. (4.8) shows the simplicity of implementation of the IBN system as compared with that of the two's complement system implementation, as far as the $N^2$ computation WHT algorithm is concerned. The simplicity of the analogue-to-IBN conversion should also be added to the above comments.

A complete parallel WH transformer, based on the IBN technique and the $N^2$ computation algorithm is shown in Fig. (4.9)

### 4.2.2 Data-Reiteration WH Transformer

The parallel structure shown in Fig. (4.9) can be modified in such a way that only one ex-OR gate and one counter is needed. This is achieved by storing the input data and reiterating these stored data at a rate of N times faster per analysis interval period. By this scheme, in each reiteration, one WH coefficient is obtained. The block diagram of this scheme is shown in Fig. (4.10).

For storing and reiterating the input samples, two shift registers, each capable of storing N samples, must be employed. When the samples inside one of the shift registers recirculates, at a rate of N times more than the rate of input data acquisition the other shift register stores the samples of the next interval at the real time rate. This Data-reiteration WH transformer offers a significant reduction in hardware complexity compared to the parallel structure. However, because of the limited speed of existing shift registers, this scheme can be employed only for relatively low order transformations.

### 4.2.3 Walsh Function Generators

For the implementation of the $N^2$ Computation WH algorithm it is necessary to have a Walsh function generator incorporated in the system as shown in Fig. (4.9), Fig. (4.10). Numerous methods of generating Walsh functions have been described[32]. When the Walsh functions generator is needed for the processing of wide-band signals such as image signals, the propagation delays of logic components which cause hazard and unorthogonality in the generated functions, become very important. For speech processing applications, the propogation delays are negligible compared to the width of Walsh functions pulses and hence in this case the main goal is reduction in complexity.

In this section two structures for the generation of Walsh

functions are given. (a) Parallel Walsh function generator, (b) Serial Walsh function generator. These generators are based on the multiplication property of Walsh function as discussed in Harmuth[32]. By this property, the multiplication of two Walsh functions will generate another Walsh function according to the following equation:

$$Wal(i,t).Wal(j,t) = Wal(i \oplus j,t) \qquad (4.4)$$

where the sign $\oplus$ indicates module 2 addition. From such a relationship, therefore, Walsh function can be generated from Rademacher functions as discussed in section 1.4.1. The first Rademacher function is Wal(o,t), the second rademacher function is Wal(3,t), whilst in general the nth Rademacher function is given by $Wal(2^n-1,t)$. The generation of other Walsh functions from Rademacher functions is indicated in Table 4.1.

(a) Parallel Walsh functions generator: This generator is implemented by using a counter to produce the Rademacher functions and ex-OR gates to multiply these functions according to Table 4.1. The logic gate arrangement of the Parallel Walsh function generator for N=8 is shown in Fig. (4.11).

(b) Serial Walsh functions generator: For the data-reiteration WH transformer shown in Fig. (4.10) a Walsh generator is needed which can create Walsh function in serial form. The structure of this generator is shown in Fig. (4.12) for N=8. It consists of 3 bits binary counters, 3 NAND gates and one 3 input ex-OR gate.

### 4.2.4 Pipeline Fast Walsh-Hadamard Transform

The fast WH processor, described in this section, is based on the Shanks algorithm as shown in Fig. (1.9). This processor,

referred to as the Pipeline FWHT, is similar in some ways to the Pipeline FFT described by Groginsky and Works[89]. Fig. (4.13) shows an arrangement for performing the first stage of the flow graph of the fast WHT, shown in Fig. (1.9). The operation is as follows:

The first four samples x(0) to x(3) are switched into the four stage delay elements $Z^{-4}$. During the first four samples, switches A and B are in position 1 and these four samples enter the delay line. During the second four samples from x(4) through to x(7), switches A and B are in position 2 and the output of the adder is connected to the input of the next stage whilst the output of the subtractor is sent to the delay line. Thus when the delay line is being filled by new input data belonging to the next block, the contents of the delay line are transferred to the next stage. The output of this stage is the first column of the algorithm of Fig. (1.9)

It is quite evident from Fig. (1.9) that the structural form of stage 1 is repeated twice in stage 2. Thus the next stage needs half the delay of the first stage and the rate of switching is doubled.

Fig. (4.14) shows a pipeline FWHT for N=8. The control mechanism of this system is simply a 3-bits binary counter. The binary number generated by this counter, identifies the sequency of the coefficient generated at the output of the system.

The hardware of the Pipeline, shown in Fig. (4.14), can be constructed by using either two's complement adders and subtractors or alternatively by employing the already described IBN adders and subtractors. Using the latter arithmetic units in the Pipeline realisation has two advantages: (i) Simplicity of the IBN adder and subtractor and also their modularity as discussed in Chapter 3 yields systems of highly reduced complexity.

(ii) The finite register length noise is lower than the corresponding binary positional realisation as the result of Section 4.3.2 indicates. The only disadvantages of using IBN arithmetic units are increased speed of operation and the necessity of having longer shift registers. Both of these disadvantages arise as a result of the longer IBN word-length. Since the length of the required shift registers are an integer power of 2 and they are inexpensively available, the Pipline transformer employing the IBN arithmetic units remains superior in terms of hardware efficiency.

## 4.3 Error Analysis of Walsh-Hadamard Transformers

The error performance of WH transformers is a very important attribute of specific implementations in addition to the economical design of these processors. The error at the output of a WH processor originates from two sources: (i) Quantization error of the input signal; (ii) Finite register length error. The errors are reduced by increasing the word-length of a specific implementation, but in general longer word-lengths require more complexity in the hardware realisation. Therefore the designer is required to compromise between a tolerable error level and a consequently reduced complexity.

In this section, the error performances of WH processors, described in section 4.2, are discussed. It is shown that the processors which employ the direct $N^2$ computation algorithm and IBNAU have an interesting error property which is desirable for speech coding by WHT.

### 4.3.1 Error Analysis of the WH Transform using $N^2$ Computation Algorithm

In the direct computation method of the WH coefficients, each coefficient is computed by N additions or subtractions where N is the order of the transform. If the two's complement representation is employed and the samples of the input signal are quantized by b bits, then each sample has a quantization noise with a variance[24],

$$\sigma_x^2 = \frac{2^{-2b}}{3} \qquad (4.5)$$

Eq. (4.5) is based on the assumption that the quantization noise is uncorrelated. When the input signal is transformed to the WH domain, then the variance of the noise associated with each coefficient is N times more than that of input samples, i.e.

$$\sigma_{nc}^2 = N \frac{2^{-2b}}{3} \qquad (4.6)$$

This relation is based on the fact that when N uncorrelated samples are added or subtracted, the variance of the result is N times more than the variance of each sample.

If the input signal is a random white signal with a flat probability density function and its amplitude is limited to be between $\pm 1$, its variance will be given by

$$\sigma_x^2 = \int_{-1}^{1} x^2(t) \cdot p(x) \cdot dx = \int_{-1}^{1} x^2(t) \cdot \frac{1}{2} \cdot dx = \frac{1}{3} \qquad (4.7)$$

The variance of the WH coefficients will be N times more than that because of N additions or subtractions, i.e.

$$\sigma_{xc}^2 = \frac{N}{3} \qquad (4.8)$$

The signal-to-noise ratic associated with each coefficient is then obtained from Eq. (4.6) and Eq. (4.8),

$$[SNR]_{TC} = \frac{\sigma_{xc}^2}{\sigma_{nc}^2} = 2^{2b} \qquad (4.9)$$

For the WH transformer based on the IBN system and the direct $N^2$ computation method, the following relation can be written directly from the circuit diagram of the DSM, shown in Fig. (3.3):

$$\int_{t_1}^{t_2} [x(n) - w(n)]\, dt = \int_{t_1}^{t_2} d(t)\, dt = e_2 - e_1 \qquad (4.10)$$

where x(n) is the samples of the signal to be transformed, w(n) is the binary output of the DSM, d(t) is the difference of the input signal x(n) and the output signal w(n), and $e_1$ and $e_2$ correspond to the output of the integrator at time $t_1$ and $t_2$ respectively. If the time interval between $t_1$ and $t_2$ is equal to one sampling interval T, then for constant input x(n) during a sampling interval, the integra l of x(n) between time $t_1$ to $t_2$ will be a constant multiplied by x(n). Hence Eq. (4.10) can be written in the following form:

$$x(n) - \frac{1}{B} \sum_{i=0}^{B-1} w(i) = \int_{(m-1)T}^{mT} d(t)\, dt = e_2 - e_1 \qquad (4.11)$$

where the left side of Eq. (4.11) shows the difference between input sample x(n) and its approximation by a representation of B bits in IBN. If the input signal is limited to $\pm 1$ and w(i) is either $\pm 1$, then the differnece signal d(t) can range between $\pm 2$ and the output of the integrator can change between $\pm \frac{2}{B}$ where $\frac{1}{B}$ is the constant of integration. From Eq. (4.11) the variance of the quantization noise associated with the IBN representation is then given by:

$$\sigma_n^2 = E|e_2 - e_1|^2 = 2\,E|e_1|^2 = 2\int_{-\frac{2}{B}}^{\frac{2}{B}} e_1^2\, p(e_1)\,de_1 \qquad (4.12)$$

Assuming that $e_1$ is uniformly distributed in the range of $\pm \frac{2}{B}$, $p(e_1)$ will be given by

$$p(e_1) = p(e_2) = \frac{1}{\frac{2}{B} - (-\frac{2}{B})} = \frac{B}{4} \qquad (4.13)$$

Eq. (4.12) can be simplified by using Eq. (4.13)

$$\sigma_n^2 = 2\int_{-\frac{2}{B}}^{\frac{2}{B}} e_1^2 \cdot \frac{B}{4} \cdot de_1 = \frac{8}{3B^2} \qquad (4.14)$$

The major difference between the conventional binary positional representation and the IBN representation from error analysis point of view, is that in the latter case the quantization noise of adjacent samples is correlated. In order to compute the SNR of the IBN-WH transformer, a second order transformer is first considered. The first coefficient of this transformer is obtained by one addition whilst the second coefficient requires one subtraction. The noise associated with the first coefficient is obtained by adding the quantization noise of two adjacent samples,

$$E_{C1} = E_1 + E_2 \qquad (4.15)$$

where $E_1 \triangleq [e_2 - e_1]$ and $E_2 \triangleq [e_3 - e_2]$ as given in Eq. (4.12). By substitution of these values in Eq. (4.15), we obtain

$$E_{C1} = E|e_3 - e_1| \qquad (4.16)$$

which has the same variance as $E_1$, given by (4.14)

$$\sigma_{C1}^2 = \frac{8}{3B^2} \qquad (4.17)$$

The noise for the second coefficient is obtained by subtracting $E_1$ from $E_2$,

$$E_{C2} = E_1 - E_2 = E|e_3 + 2e_2 - e_1| \qquad (4.18)$$

By considering that each $e_i$ has a variance half of that given by Eq. (4.14) and the contribution of $e_2$ in the noise variance, given by Eq. (4.19), is four times than that of $e_1$ and $e_2$ Eq. (4.18) becomes

$$E_{C2} = \frac{1}{2}\cdot\frac{8}{3B^2} + 4\cdot\frac{1}{2}\cdot\frac{8}{3B^2} + \frac{1}{2}\cdot\frac{8}{3B^2} = \frac{8}{B^2} \qquad (4.19)$$

Eq. (4.17) and Eq. (4.19) show an important error characteristic of the IBN-WH transformer namely that the distribution of noise over the coefficients is non-uniform. A general expression for the computation of the coefficient noise in an N order WH transform can be derived by extending the approach employed in the second order case. The nth coefficient noise is then given by

$$\sigma_{nc}^2(n) = \sigma_n^2(2n+1) = \frac{8(2n+1)}{3B^2} \qquad (4.20)$$

For a random signal with a flat pdf limited between $\pm 1$, its variance after transformation will be $N/3$ according to Eq. (4.7) and Eq. (4.8), the SNR for the nth coefficient is therefore

$$[SNR]_{C_n} = \frac{NB^2}{8(2n+1)} \qquad (4.21)$$

### 4.3.2 Error Analysis of Pipeline WH Transformer

In Section 4.2.4, the pipeline fast WHT was described. In this section the error analysis of this pipeline is discussed when the realisation is by means of either TCAU or IBNAU. The error analysis, presented in this section, is similar to that for the Pipeline fast FFT processor[24]. In the Pipeline processor realised by using two's complement adder and subtractor, the input signal must be scaled at the input in order that no overflow occurs at the output of two's complement adders and subtractors. By considering a random input with a flat pdf limited to $\pm 1$, the input must be divided by N before entering the processor. The variance of the input signal in this case is then given by

$$\sigma_x^2 = \frac{1}{3N^2} \qquad (4.22)$$

Thus for a word-length of b bits, the quantization noise of each input sample is obtained from Eq. (4.5) and hence the SNR at the input of the processor is given by

$$[SNR]_{TC} = \frac{2^{2b}}{N^2} \qquad (4.23)$$

Since the proper operation of two's complement adders or subtractors, involving no overflow, doesn't add extra noise to the processing signal, the SNR at the output will also be given by Eq. (4.23).

In the Pipeline implementation with IBN arithmetic units, no scaling of the input is necessary, since no overflow can happen in the IBN adder or subtractor (see chapter 3 ), the output noise in this case can be computed by considering the cumulative effects of noise generated at each stage. It should be noted that the noise variance at each stage is divided by 2 by the following stage. Hence, the output noise is obtained in the form

$$\sigma^2_{IBN} = \frac{1}{N}\cdot\frac{8}{3B^2} + \left[1-\left(\frac{1}{2}\right)^{\nu}\right]\cdot\frac{1}{B^2} \qquad (4.24)$$

$N=2^{\nu}$ $\quad \sigma_n^2 = \frac{1}{2B^2}$ adder noise

where the first term is the noise due to the input and the second term shows the noise generated in the Pipeline. For large values of N ($N>8$), Eq. (4.24) is approximated to

$$\sigma^2_{IBN} = \frac{1}{B^2} \qquad (4.25)$$

The output SNR can be evaluated by noting that the output signal variance is $\frac{1}{N}^{th}$ of the input signal variance of $1/3$ because of the division by 2 at each stage so that

$$[SNR]_{IBN} = \frac{B^2}{3N} \qquad (4.26)$$

Comparison of Eq. (4.23) and Eq. (4.26) shows that for $B=2^b$, the IBN processor results in a better signal to noise ratic , specially for large N.

## 4.4 WH Inverse Transform Implementation

For inverse transformation of a vector, this vector should be multiplied by the inverse of the matrix which has been used in the forward transform. However, the inverse of an orthogonal matrix is the same as its transpose[31]. Since the Hadamard matrices are symmetric their inverse and the original matrix are identical. Hence, the hardware realization of a WH transformer and the corresponding inverse transformer realization are identical. For the same word-length in the transformer and inverse transformer, the amount of noise

contributed by the inverse transformer is naturally equal to that due to the transformer. Thus in the inverse transformer employing the pipeline structure and TCAU implementation the input must be again divided to N to prevent overflow. This makes the SNR at the output drop by $\frac{1}{N}2$ when compared with the SNR at the input. If one assumes that the output of transformer reaches the input of the inverse transformer through an ideal channel, then the SNR of the inverse transform coefficients in the two's complement case is given by,

$$[SNR]_{TC} = \frac{2^{2b}}{N^4} \qquad (4.27)$$

where b is the word-length of the transformer and the inverse transformer, and N is the order of the transform.

When the inverse transformation is achieved by means of the Pipeline structure employing the IBN technique, the noise is created by the division by 2 performed at each stage. The signal power at the output drops by N times because of this division but the noise power is the same as in the transformer, since the effect of transmitter noise is eliminated by performing this division. The SNR of this inverse transformer is then given by

$$[SNR]_{IBN} = \frac{B^2}{3N^2} \qquad (4.28)$$

where B is the word-length of the transformer and also the inverse transformer.

In the IBN implementation of transformer and inverse transformer based on the direct $N^2$ computation algorithm, the word-length of the inverse transformer must be $v=\log_2 N$ bits more than that of the transformer until the transformer and the inverse transformer have

the same SNR performance. This is because the variance of the coefficients, obtained by the transformer, is N times more than the variance of the original input. The SNR of the inverse transform coefficients, in this case, is obtained by using Eq. (4.9) and Eq. (4.21).

## 4.5 Implementation of Discrete Cosine Transform

In this section the implementation of the DCT is described, using the direct $N^2$ computation algorithm. The chosen structures for this purpose are the parallel and the data-reiteration structure similar to those discussed in Section 4.2. The block diagram of the parallel structure is shown in Fig. (4.15). It is apparent from this figure that the samples of the DCT functions are stored in a ROM. The analogue-to-digital convertor is a DSM preceeded by a sample-and-hold. The IBN multiplier multiplies the output of the DSM by the output of the ROM. The products are then accumulated in an up-down counter, the contents of which constitute the DCT coefficients in binary form.

The block diagram of the data-reiteration DCT transformer is shown in Fig. (4.16).

The IBN-DCT transformer benefits from the simpler analogue to digital convertor and the simpler multipliers as compared with the DCT transformer that employs TCAU. Either of the two multipliers described in Chapter 3, may be employed in the IBN-DCT transformer implementation.

## 4.6 Error Analysis of the DCT Processor

The real time DC transformation, described earlier introduces

two types of noise: (a) Noise originated from the input quantization; (b) Noise created by the multiplication process. These two types of noise are assumed uncorrelated so that the variance of total noise is the sum of the variances of these two noise sources. In the two's complement system, the multiplication noise is equal to the quantization noise and their value is given by Eq. (4.5). This is because in quantization or multiplication, noise is caused by rounding a word of $(b+b_1)$ bits to b bits. Hence, the variance of the total noise of DCT coefficients, with regard to Fig. (4.16) is twice as much as given by Eq. (4.6), i.e.

$$\sigma_t^2 = 2N \frac{2^{-2b}}{3} \qquad (4.29)$$

This noise variance as given by Eq. (4.29) belongs to the structure employing TCAU where b is the word-length of the processor.

In the DCT transformer using IBNAU, the effect of the input quantization noise can be computed in the same way as approached in Section 4.3. Let us consider a second order DCT transform where the samples of the first DCT function are taken to be $a_1$ and $a_2$. By referring to Section 4.3.1, the quantization noise associated to the first and second samples are $e_2-e_1$ and $e_3-e_2$, respectively. These noises will be multiplied by $a_1$ and $a_2$ during the transform operation. Thus the contribution of the input quantization noise to the output noise variance is given by

$$\sigma_q^2 = E\left\{\left[a_1(e_2-e_1)+a_2(e_3-e_2)\right]^2\right\} \qquad (4.30)$$

Assuming $e_1$, $e_2$ and $e_3$ to be uncorrelated and to have equal variances, as was the case in section 4.3.1, we have

$$\sigma_q^2 = (a_1^2 + a_2^2 - a_1 a_2)\sigma_n^2 \qquad (4.31)$$

where $\sigma_n^2$ is the quantization noise of each IBN sample, as given in Section 4.3.1 i.e.

$$\sigma_n^2 = 2.E\left|e_1^2\right| = \frac{8}{3B^2} \qquad (4.32)$$

In general, for a transform of order N, the contribution of quantization noise at the output is given by

$$\sigma_{q(k)}^2 = \left[\sum_{n=0}^{N-1} a_{kn}^2 - \sum_{n=1}^{N-1} a_{kn}\, a_{k(n-1)}\right]\sigma_n^2 \qquad (4.33)$$

where $\sigma_q^2(k)$ is the effect of quantization noise at kth DCT coefficient and $a_{kn}$ is the element in the kth column and nth row of the DC matrix. By substituting the actual values of $a_{kn}$ in Eq. (4.33), this equation can be simplified to the following form:

$$\sigma_q^2(k) = \left[\left(\frac{N}{2} - 1\right)\mathrm{Sin}^2\frac{\pi k}{2N} + 1\right]\frac{8}{3B^2} \qquad (4.34)$$

This simplification is given in Appendix (1). The total amount of error variance is obtained by adding the error variance of multiplication to Eq. (4.34). When the cascade form of the IBN multiplier is employed (see chapter 3) the multiplication noise variance can be obtained by using Eq. (3.19) so that

$$\sigma_m^2 = (N-1)\cdot\frac{1}{B^2} \qquad (4.35)$$

where N-1 is the number of multiplications needed for the computation of each coefficient and $\frac{1}{B^2}$ is the noise of one multiplication.

Thus the total amount of noise variance is given by:

$$\sigma_t^2(k) = \left[\left(\frac{N}{2} - 1\right)\sin^2\frac{\pi k}{2N} + 1\right]\frac{8}{3B^2} + \frac{N-1}{B^2} \qquad (4.36)$$

where N is the order of the DCT transform and B is the word-length in the IBN representation.

## 4.7 Comparison of the Implementation Techniques

The techniques of WHT implementation, described in this chapter, are divided into two categories: (i) Techniques which employ the $N^2$ computation algorithm; (ii) Techniques based on the fast WHT algorithms. Each category also consists of two groups. The first group employs PCM and TCAU whilst the second group is based on the IBN approach, i.e. using DSM and IBNAU.

In this section, these four groups are compared in terms of signal to noise ratio performance at a specified word-length. As mentioned in Chapter 3, the word-length in the IBN representation is larger than that in the corresponding binary representation, so that, at a definite SNR value, a word-length of b bits in the binary representation corresponds to a word-length of $B=2^b$ bits in the IBN. The chosen word-lengths, in this section are b=4, 6, 8 bits, when TC is employed, and $B= 2^4, 2^6, 2^8$ bits, when this processor is implemented by IBNAU. The orders of transform chosen are N=16 and N=64 point, since they are the most common WH orders employed in speech processing[45,50]. The SNR values for each processor are plotted for comparison purposes, and the last part of this section is allocated to showing graphically the SNR performance of the DC transformer.

(a) SNR performance of the $N^2$ computation processor: The SNR of a 16 point WH transformer, when employing IBNAU and TCAU

is shown in Fig. (4.18) and Fig. (4.19), respectively. The SNR of the 64 point transformer is also shown in Fig. (4.20) and Fig. (4.21). As is evident from these figures, in the transformer which employs TCAU, the noise is uniformly distributed over the WH spectrum, whilst in the processor employing IBNAU, the coefficients of lower sequencies have better SNR values. Moreover, in the latter case, the SNR increases with increasing order of the transform. As mentioned earlier, this property is in favour of speech coding at low bit rates since the lower sequency coefficients of speech signals are perceptually more important[15] and these coefficients should be transmitted with a better accuracy.

(b) SNR performance of the Pipeline processor: The SNR values of a 16 point Pipeline processor, using IBNAU or TCAU, are shown in Fig. (4.22) and Fig. (4.23), respectively. The SNR values, corresponding to the 64 point transformer, are also shown in Fig. (4.24) and Fig. (4.25) . A comparison of these figures reveals a 7.2 dB advantage, in the SNR of the 16 point IBN Pipeline processor, over the TC-Pipeline processor. This advantage reaches to 10.4 dB in a 64 point transformer, as shown in Fig. (4.24) and Fig. (4.25). The distribution of SNR values in the IBN-Pipeline processor, unlike that of the $N^2$- computation processor, is uniform. This is because, the quantization noise of the input is almost eliminated, by the division by 2 at each stage, before it reaches the output.

## 4.8 Discussion

In this chapter, the implementation of two well-known orthogonal transforms, the WHT and the DCT, have been discussed. The proposed implementation techniques are based on the use of DSM as the analogue-to-digital convertor, and IBN arithmetic units developed in Chapter 3 . At the beginning of this chapter, some important techniques, for our purposes relevant to the implementation of WHT and DCT, were discussed. Some implementation schemes were proposed based on three structures. The first and the second

structured are based on the straightforward $N^2$ computation algorithm whilst the third one employs a fast WHT algorithm. It has been shown that the implementation of the first and the second structure, by the IBN technique, can significantly reduce the necessary hardware, as compared with a TCAU implementation. The correlation property of the IBN quantization noise, has been shown to result in better SNR values for lower sequency coefficients, a fact which is important in speech perception. Subsequently, the implementation of the Pipeline WHT processor based on the fast WHT algorithm, was described. The Pipelin e structure, employed in this chapter, needs a minimum number of delay units. Furthermore the realisation of this structure by either TCAU or IBNAU was described. It has been shown that the inherent property of division by 2 of the IBN adder and subtractor, leads to a significant improvement in the SNR of the WH coefficients. The implementation of the DCT by the IBN technique was also described. The structure of this realisation is based on the straightforward $N^2$ computation algorithm. The SNR performance of the proposed structure was investigated and a simple expression for the computation of the output SNR was derived. Finally, the SNR values of various structures were compared for several word-lengths.

Selection of one of the WH transformer schemes that appeared in this chapter, depends mainly on the type of coding employed in the transform domain. For the coding methods proposed in Chapter 2, the $N^2$ computation WH transformer seems more suitable. This is because this scheme produces the base-band coefficients more accurately which are more important in this coding system.

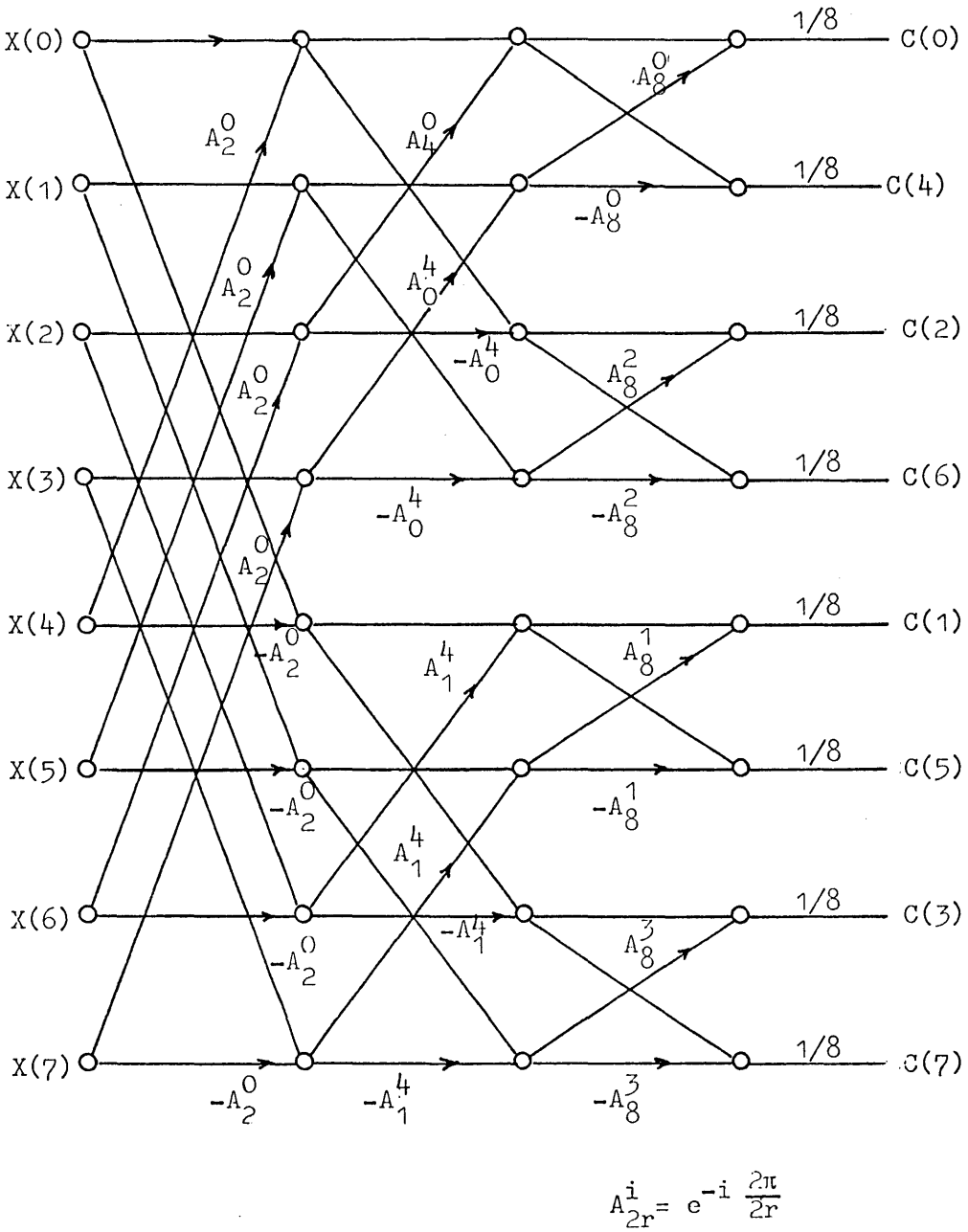



Fig.4.1. Cooley-Tukey FFT algorithm , N=8.

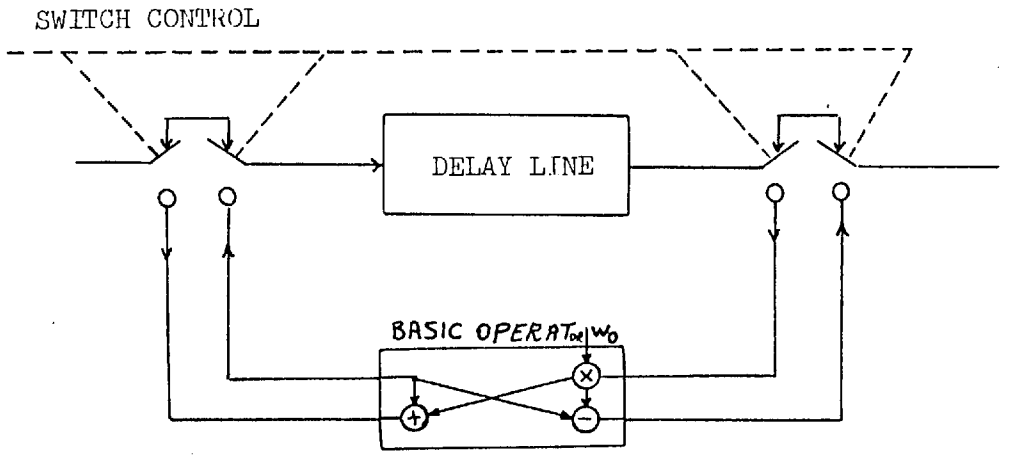



Fig.4.2. One stage of the pipeline FFT processor

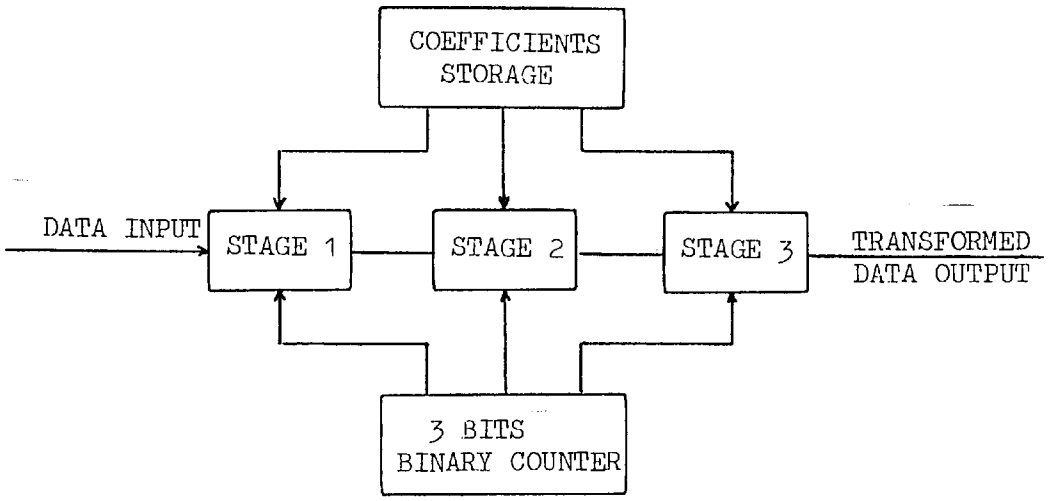



Fig.4.3. Pipeline FFT processor, N=8

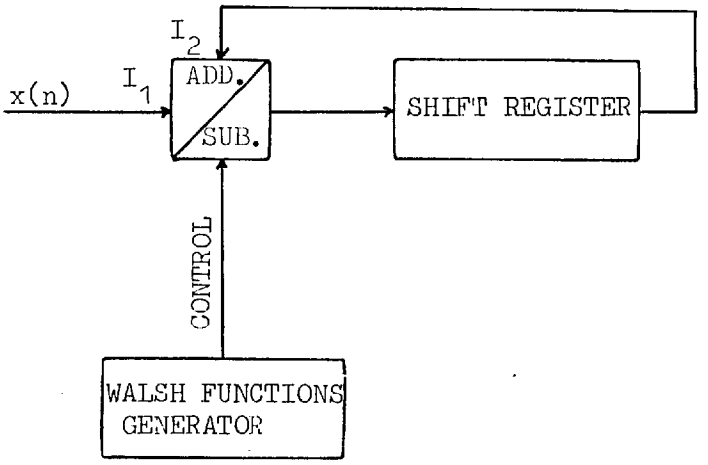



Fig.4.4. WH transformer based on the $N^2$ computation algorithm.

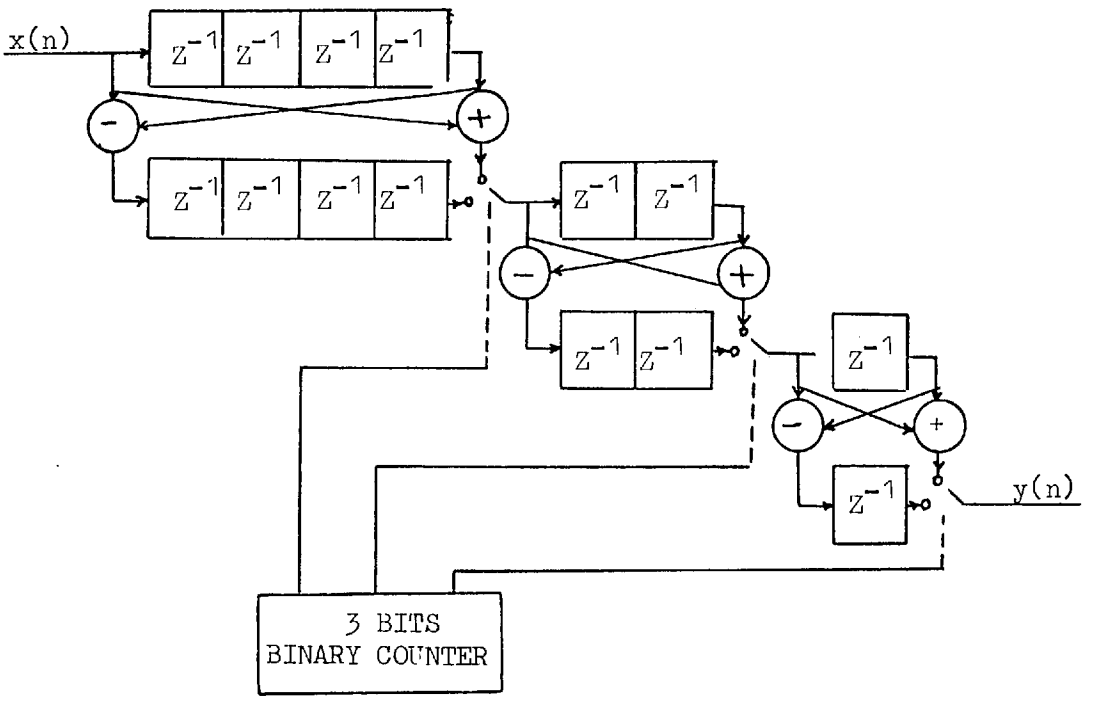



Fig.4.5. WH transformer based on the fast algorithm.

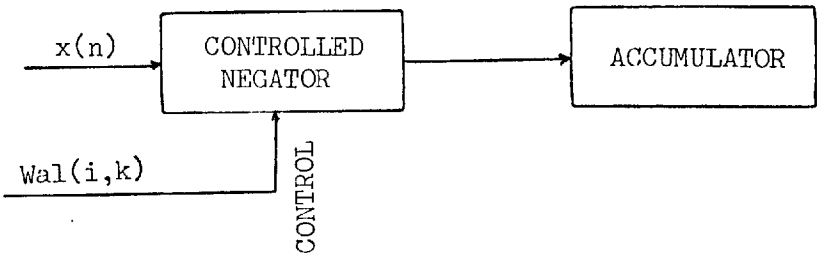



Fig.4.6. One stage of a WH transformer based on the $N^2$ computation algorithm.

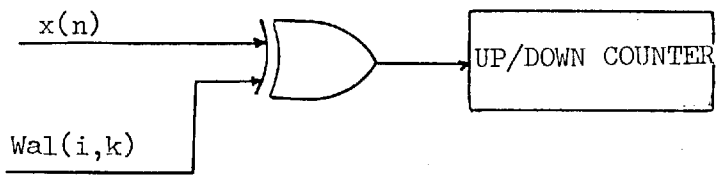



Fig.4.7. One stage of an IBN-WH transformer based on the $N^2$ computation algorithm.

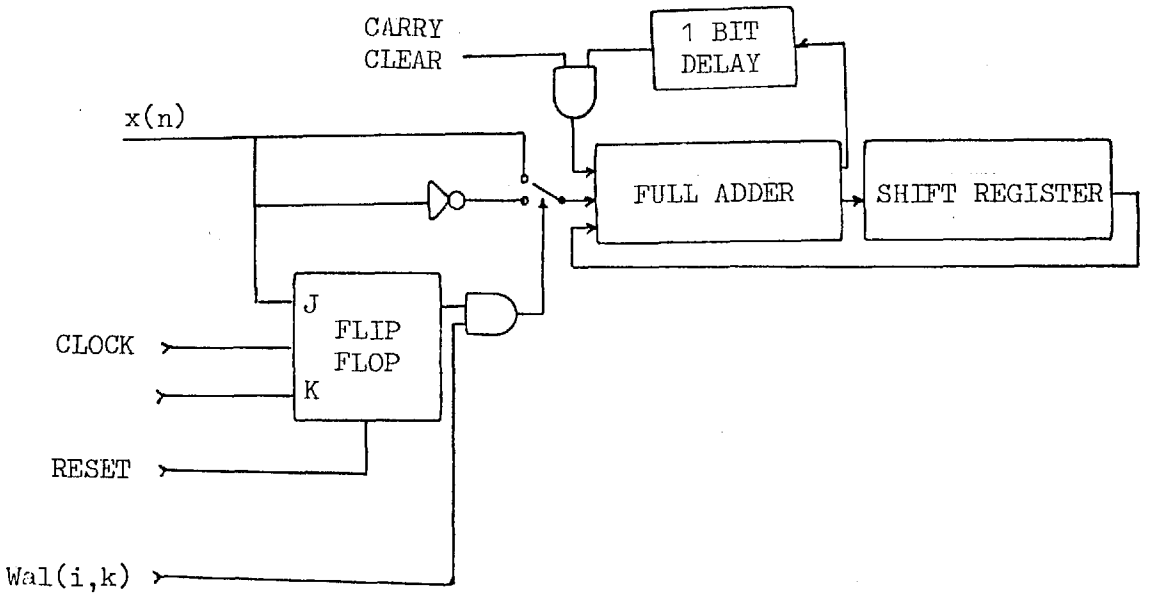



Fig.4.8. One stage of an parallel Two's Complement-WH transformer based on the $N^2$ computation algorithm.

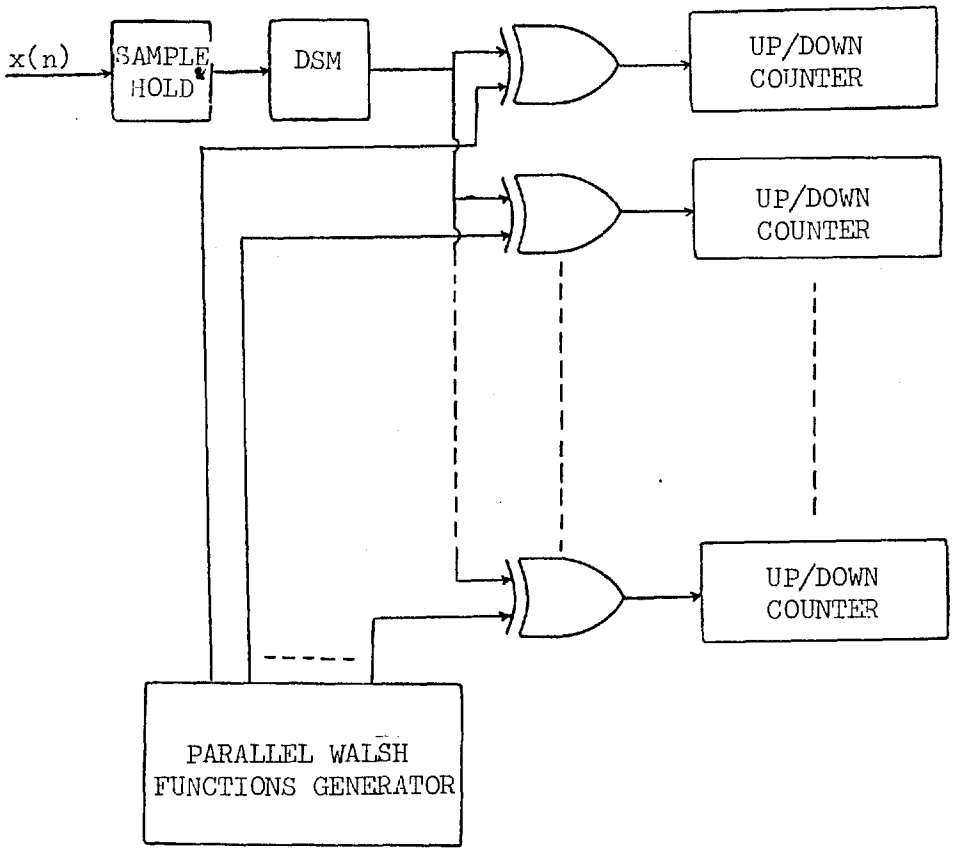



Fig.4.9. Parallel IBN-WH transformer based on the $N^2$ computation algorithm.

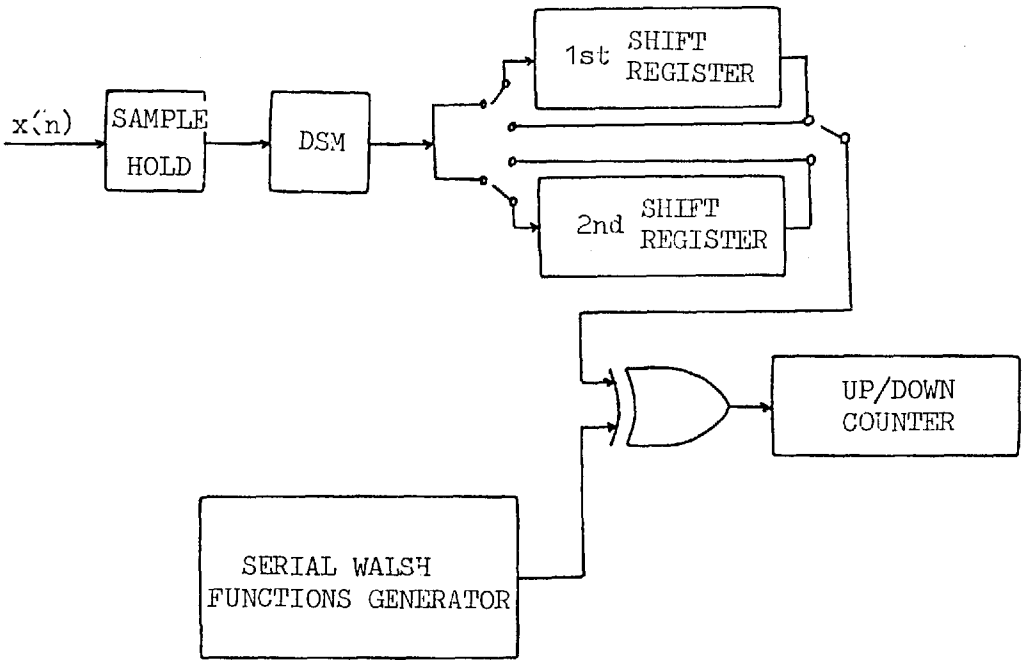



Fig.4.10. Data-reiteration IBN-WH transformer based on the $N^2$ computation algorithm.

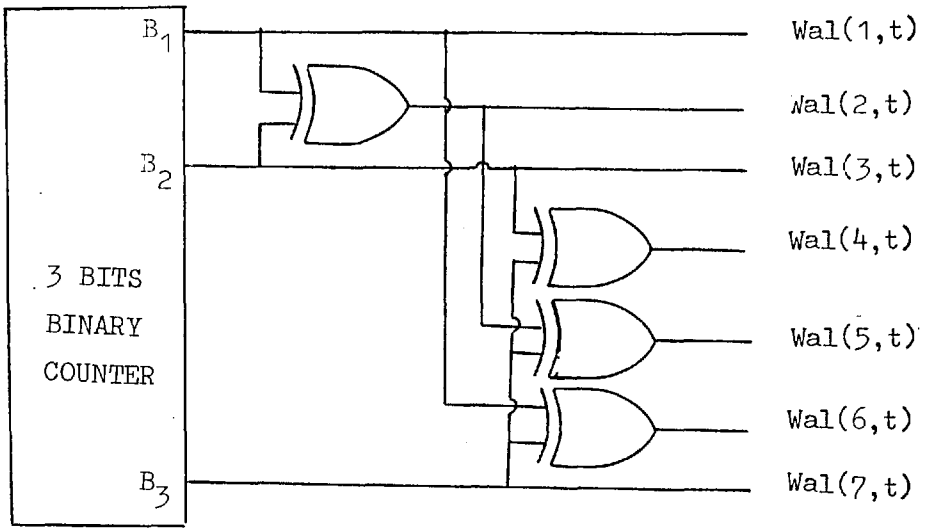



Fig.4.11. Parallel Walsh functions generator based on products of Rademacher functions.

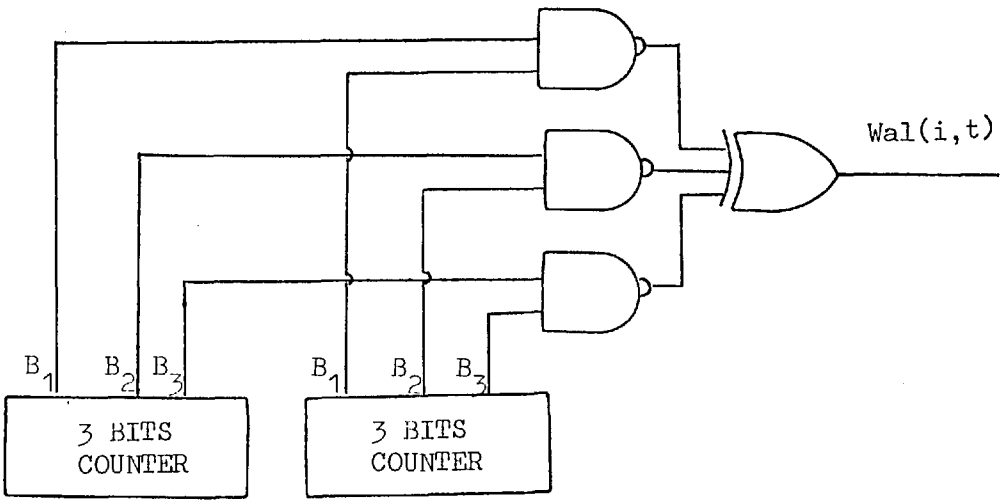



Fig.4.12. Serial Walsh functions generator based on products of Rademacher functions.

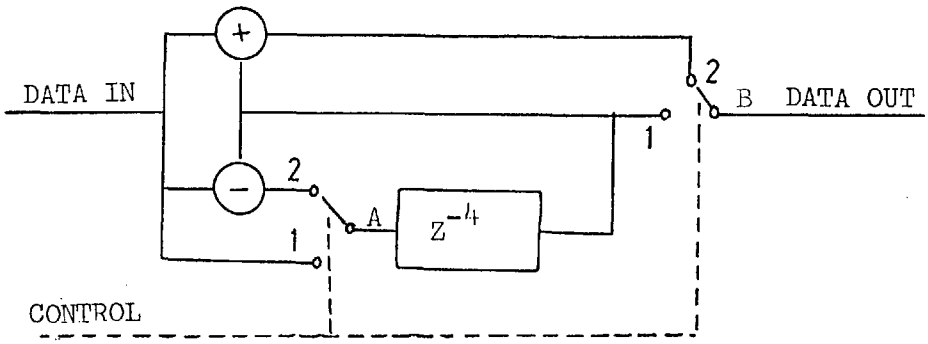



Fig.4.13. One stage of the pipeline WH transformer

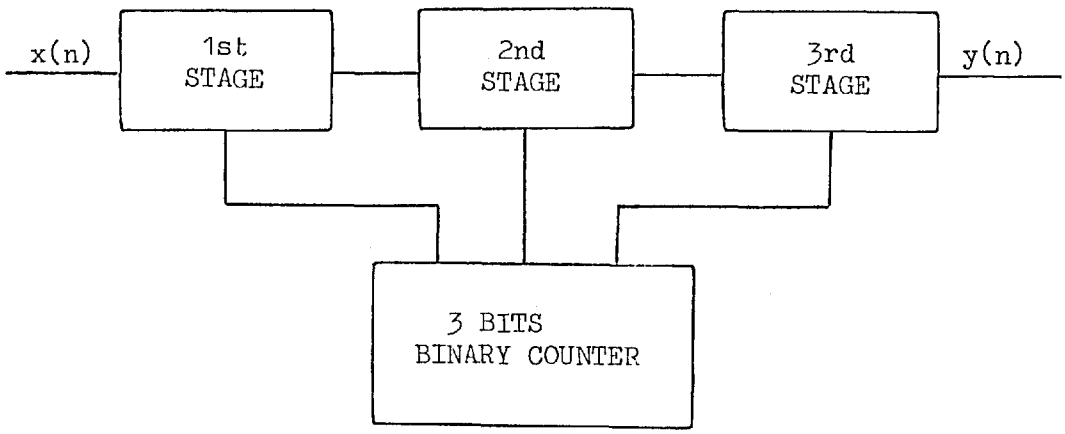



Fig.4.14. Pipeline WH transformer,N=8

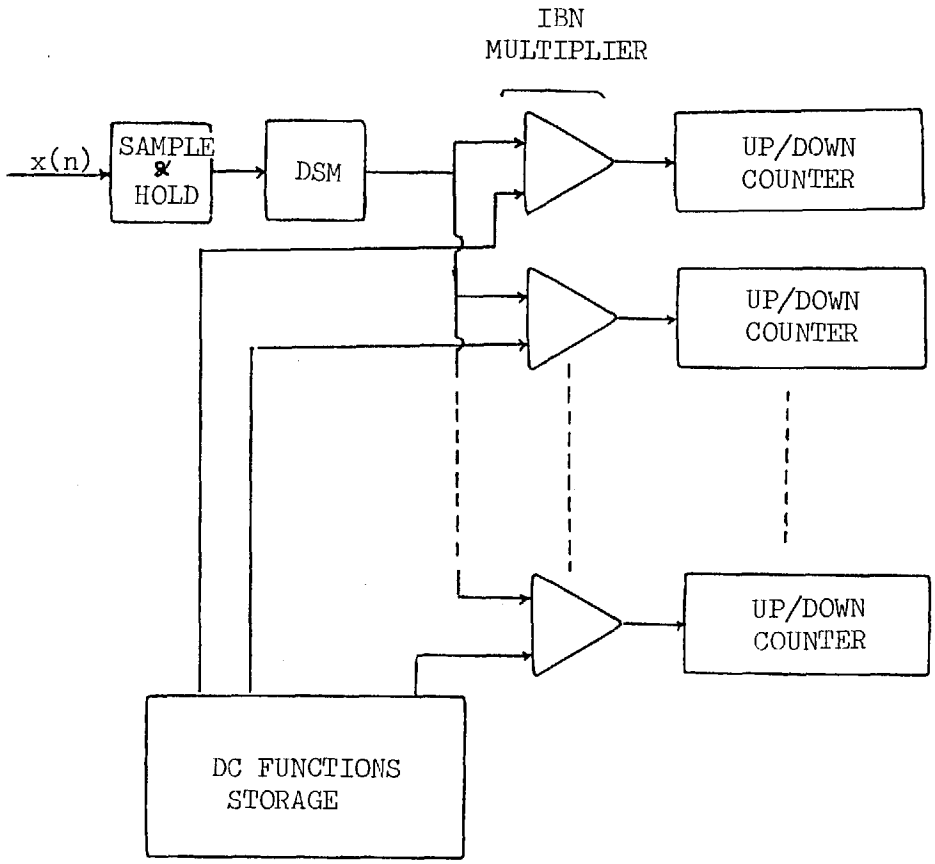



Fig.4.15. Parallel IBN-DC transformer

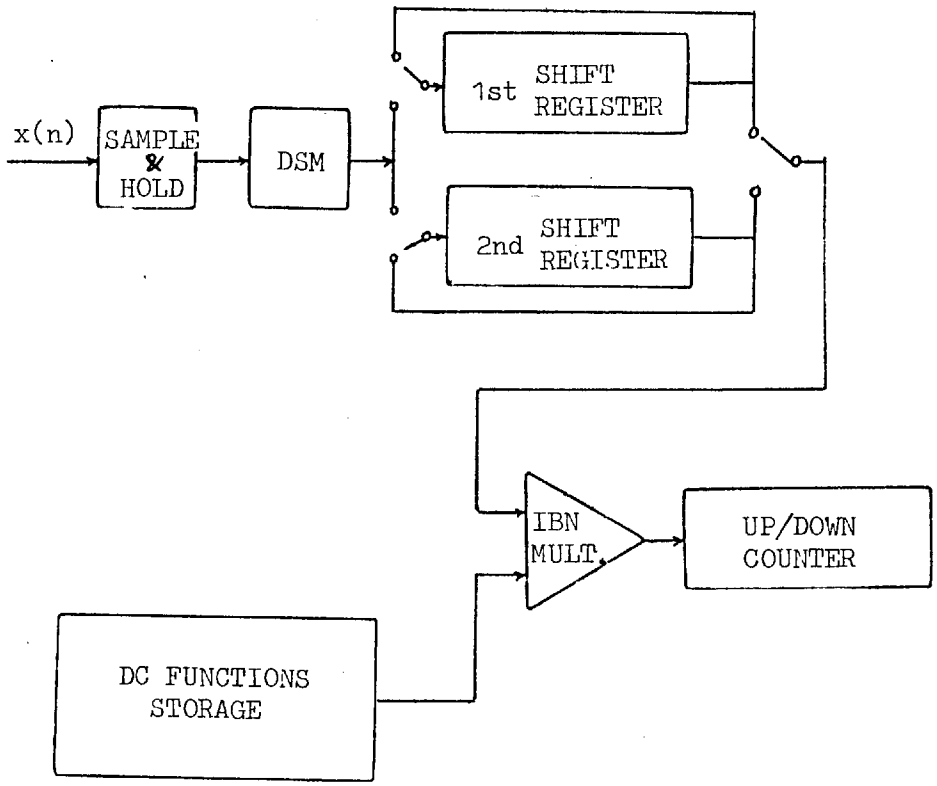



Fig.4.16. Data-reiteration IBN-DC transformer

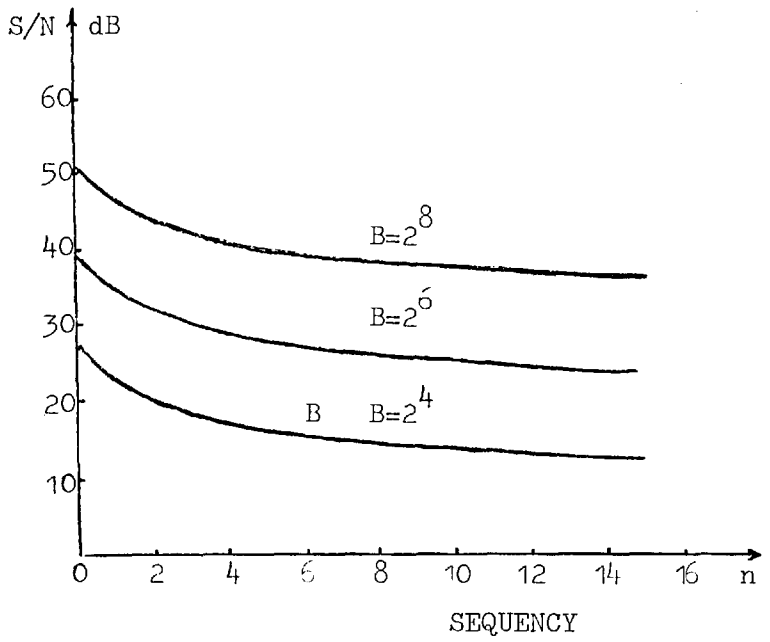



Fig.4.17. Signal to noise ratio of the 16-point IBN-WH Transformer based on the $N^2$ computation algorithm for word lenghes of $2^4$, $2^6$, $2^8$ bits.

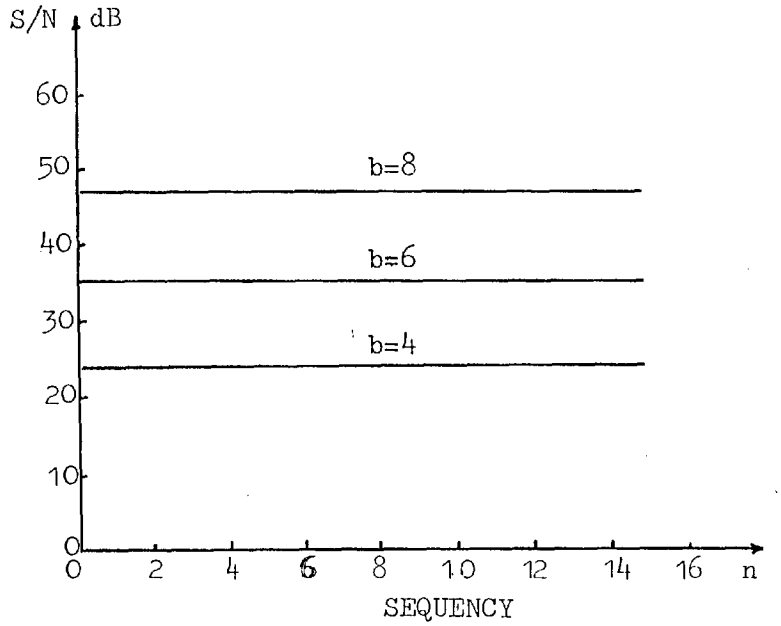



Fig.4.18. Signal to noise ratio of the 16-point Two's Complement-WHT transformer based on the $N^2$ computation algorithm for word lenghes of 4, 6, and 8 bits.

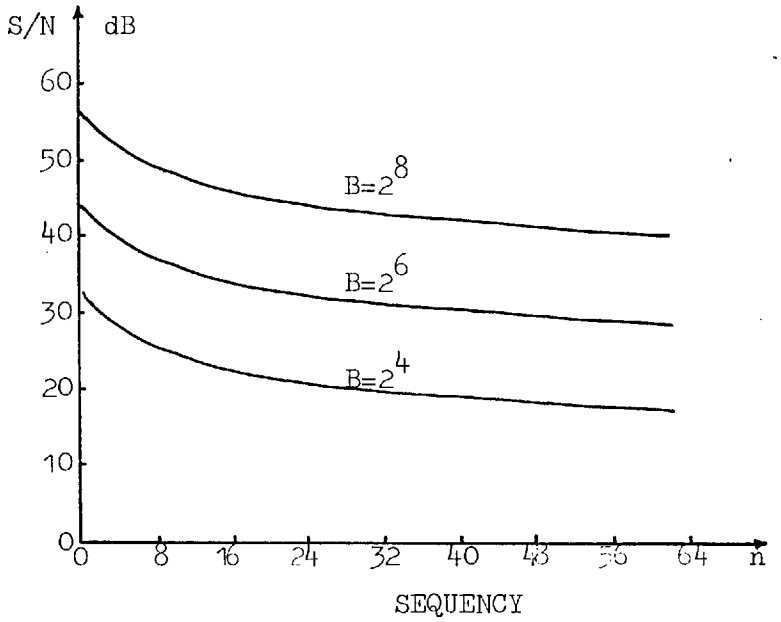



Fig.4.19. Signal to noise ratio of the 64-point IBN-WH transformer based on the $N^2$ computation algorithm for word lenghes of $2^4$, $2^6$, and $2^8$ bits.

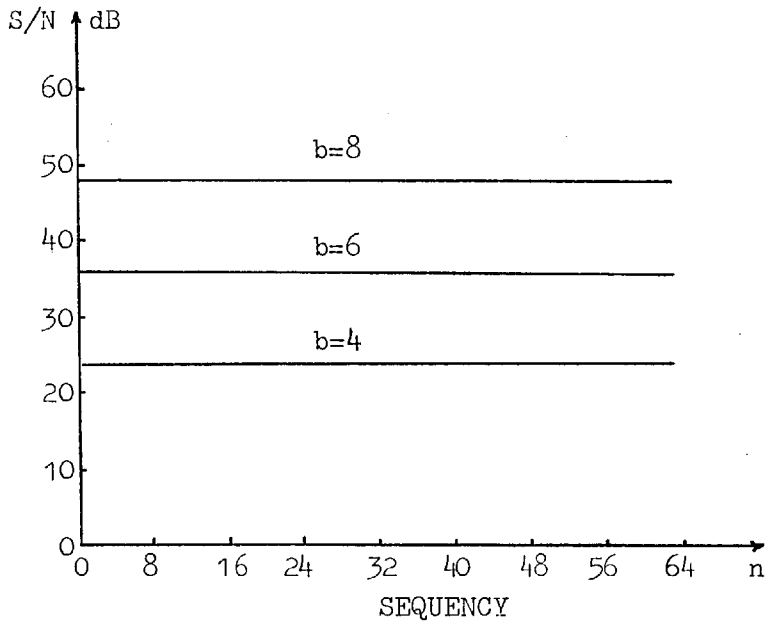



Fig.4.20. Signal to noise ratio of the 64 point Two's Complement-WH transformer based on the $N^2$ computation algorithm for word lenghes of 4, 6, 8 bits.

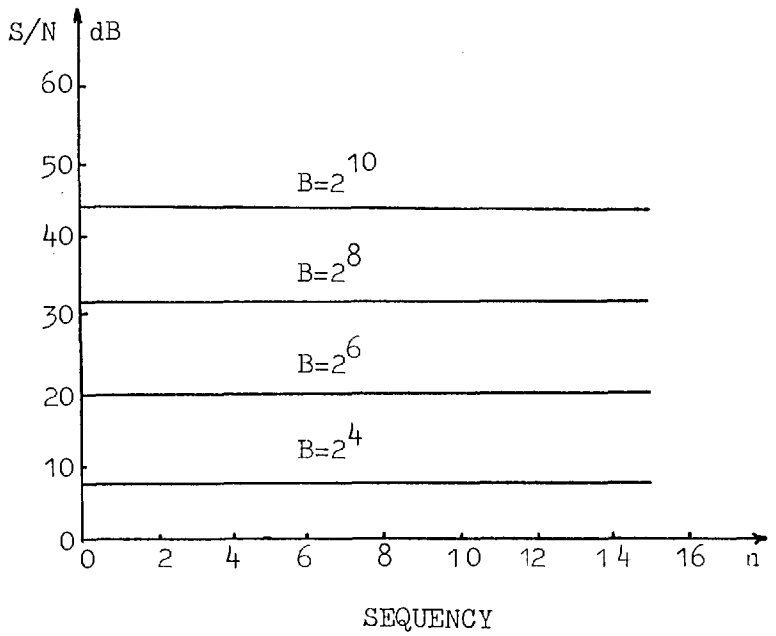



Fig.4.21. Signal to noise ratio of the 16-point IBN-WH pipeline transformer for word lengthes of $2^4$, $2^6$, $2^8$, $2^{10}$ bits.

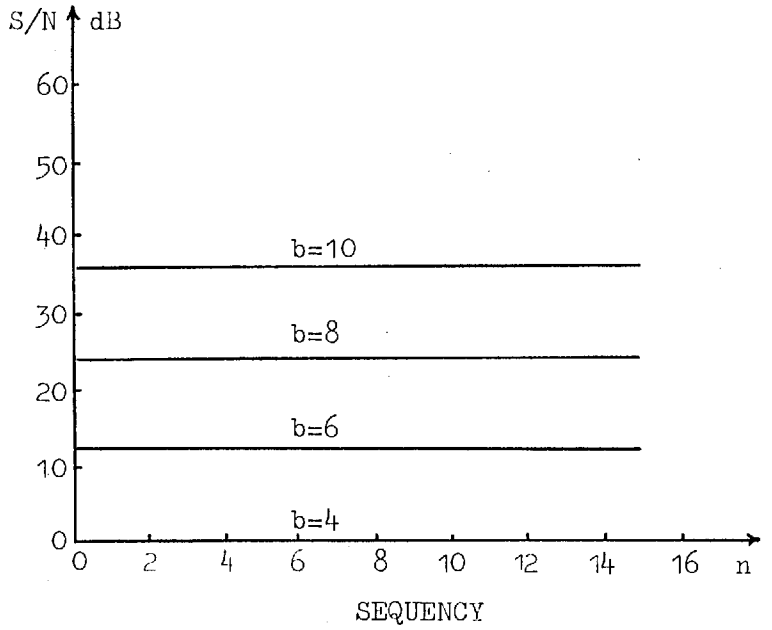



Fig.4.22. Signal to noise ratio of the 16 point Two's Complement-WH pipeline transformer for word lengthes of 4, 6, 8, and 10 bits.

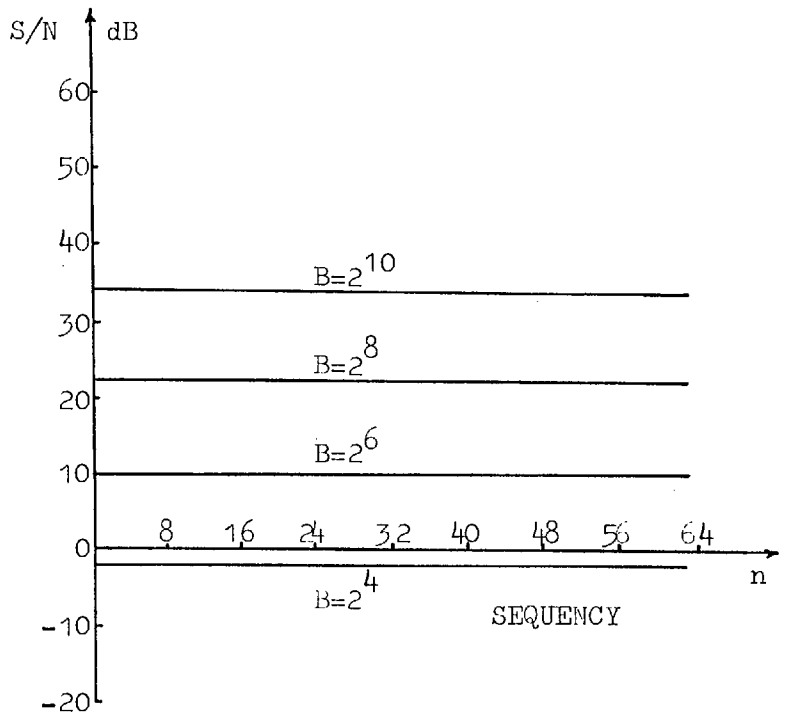



Fig.4.23. Signal to noise ratio of the 64-point IBN-WH pipeline transformer for word lengthes of $2^4$, $2^6$, $2^8$ and $2^{10}$ bits.

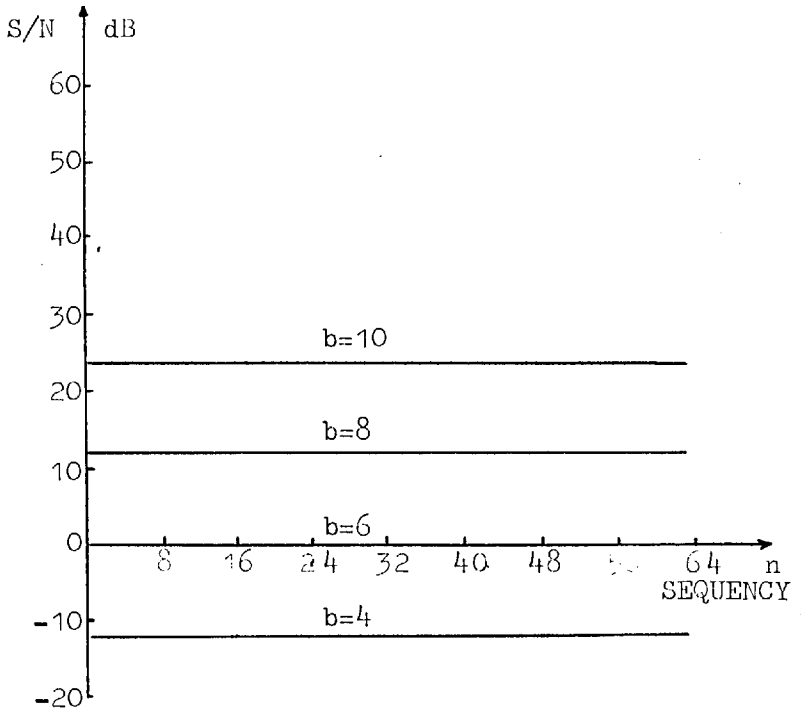



Fig.4.25. Signal to noise ratio of the 64-point Two's Complement-WH pipeline transformer for word lengthes of 4, 6, 8, and 10 bits.

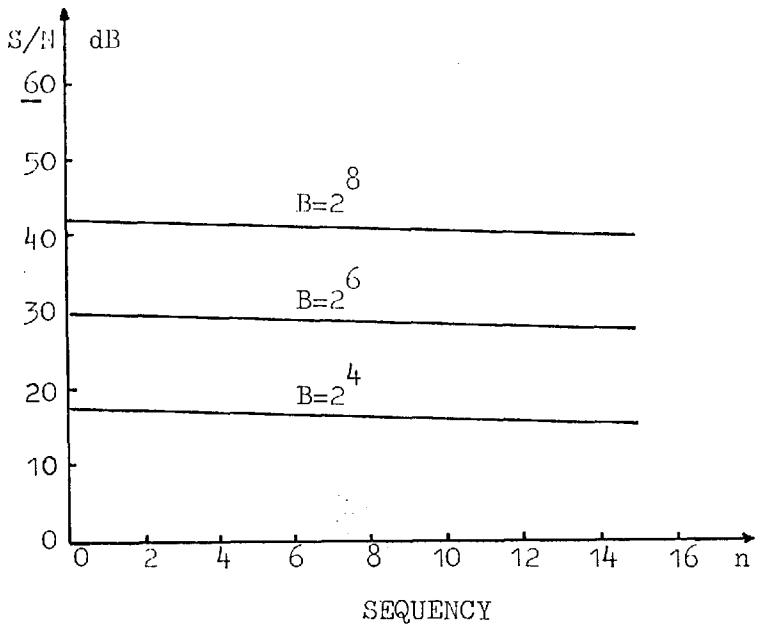



Fig.4.25. Signal to noise ratio of the 16-point IBN-DC transformer based on the $N^2$ computation algorithm for word lengthes of $2^4$, $2^6$, and $2^8$ bits.

$$
\begin{aligned}
&\mathrm{Wal}(0,t) = 1\\
&\mathrm{Wal}(1,t) = \mathrm{Rad}(0,t)\\
&\mathrm{Wal}(2,t) = \mathrm{Rad}(0,t) \times \mathrm{Rad}(1,t)\\
&\mathrm{Wal}(3,t) = \mathrm{Rad}(1,t)\\
&\mathrm{Wal}(4,t) = \mathrm{Rad}(1,t) \times \mathrm{Rad}(2,t)\\
&\mathrm{Wal}(5,t) = \mathrm{Wal}(2,t) \times \mathrm{Rad}(2,t)\\
&\mathrm{Wal}(6,t) = \mathrm{Rad}(0,t) \times \mathrm{Rad}(2,t)\\
&\mathrm{Wal}(7,t) = \mathrm{Rad}(2,t)
\end{aligned}
$$

Table 4.1. Production of the Walsh functions from the Rademacher functions.

# CHAPTER 5

## Conclusion and Future Research

In this thesis, a number of new techniques of speech coding have been presented, together with the results of an investigation of their implementation. In developing new speech coding techniques, our aim has not been to strive for an extremely low-bit-rate, low complexity, or high quality coder, but rather to develop a coder which can produce digital speech at bit-rates around the range that can be transmitted (over telephone links) by commercially available modems. The recovered speech is required to be effortlessly intelligible and to have reasonable quality. In addition the complexity of any proposed coder was required to be around that of a relatively simple waveform-coder. Further more, our aim has been to search for new implementation techniques which can significantly reduce the complexity of the realised coder as compared with the conventional methods of implementation. To achieve these aims two promising speech coding methods, namely Linear Prediction and transform coding, were combined. For the implementation of the proposed coders, a new technique based on the delta-encoding conversion, has been developed. In this concluding chapter a summary of the new techniques is presented in Section 5.1. In the second section of this cahpter, a critical assessment of the methods proposed in this thesis is presented. These are areas where further work is obviously necessary. Also in Section 5.4. some further proposals for future research are made.

### 5.1. Summary of Results

#### 5.1.1. Speech Coding Techniques

The search for new coding techniques has been made in the area of Linear Prediction and transform coding due to the following reasons*: (i) the reported success of applications of Linear Prediction in speech data compression and (ii) the ability of transform

---

* See Chapter 1

coding in obtaining reasonable compression rates with a relatively low complexity coder.

In the proposed coding methods the voiced signals are coded by a new Linear Prediction coding technique referred to as Non-Uniform Linear Prediction (NULP) coding. This method which is an extension of the well-known all-pole LP modelling, enables an efficient Linear Prediction analysis in a transform domain. The NULP method, in addition to the correlation characteristics of speech signals (which is the only feature employed by the all-pole LP method) utilises the non-uniform distribution of speech signal energy over different sequencies.

In the coding of unvoiced sounds, only the non-uniform variance distribution of transform coefficients is employed. This is because unvoiced sounds are poorly correlated and hence for them Linear Prediction is inefficient. These signals are identified before the coding operation by a method based on the different variance distribution of the voiced and unvoiced signals in the transform domain. The two transform domains employed, have been the WH and the DC domains. By performing extensive computer simulations intelligible speech with different qualities at bit-rates of 4, 6.5 and 8 kbits/s has been achieved. Judgement on the quality of the reproduced speech has been based on recent advances made towards the objective evaluation of quality. Elementary comparisons of the proposed systems with the adaptive transform coding (which is one of the best non-pitch tracking speech coding systems) have shown that the proposed systems are capable of achieving superior performance in both quality and complexity.

### 5.1.2. Implementation Techniques

A new method of implementation, based on delta-encoding, was described in Chapter 3. In this method a new system of binary numbers, referred to as the Intermediate Binary Number (IBN)

system, has been defined which enables the delta encoders to be employed systematically in the implementation of low-bandwidth signal processors. In the same chapter, IBN adders, subtractors, and multipliers were also developed and the error analysis of these arithmetic units was presented. The developed arithmetic units enable the design of a large group of digital signal processors where the simplicity of delta-encoding convertors is utilised effectively. In Chapter 4, the implementation of the WHT and the DCT, comprising the major hardware of the proposed speech coding systems, using IBN arithmetic units was discussed. These implementations are based on either the $N^2$-computation algorithm or a fast algorithm. The error analysis of each scheme was presented and their signal-to-noise ratio performance was compared for different word-lengths.

## 5.2. Critical Discussions

The research effort in this thesis was aimed at two tasks. Firstly, to search for new speech modelling and quantization methods capable of reproducing intelligible speech at low-bit-rates, with a complexity comparable with that of relatively simple waveform coders. The second objective has been the development of a new technique of hardware implementation which can significantly reduce hardware complexity of the realised coder as compared with the conventional approach. The research effort in both areas has been fruitful as indicated in the summary of new techniques, presented above. However many further questions have arisen during the course of this research and, moreover, it is considered that many areas have either been treated insufficiently, or have been omitted in the study. These are described briefly in the following sections.

### 5.2.1. Quantization of Parameters

In the proposed speech coding methods, three types of parameters are quantized and transmitted. These are the base-band coefficients,

the prediction coefficients and the mid-band coefficients. The base-band and the mid-band coefficients are quantized adaptively whilst non-adaptive quantization is employed for the prediction coefficients. The chosen adaptation strategy is based on the adaptive quantization with a one-word memory[73]. The base-band coefficients of the voiced signals and the mid-band coefficients of the unvoiced signals have been assumed in this case to be uncorrelated. Thus the related multiplication factors used in their adaptation, have been chosen amongst the multiplication factors of the uncorrelated signals, as given by Jayant[73]. These chosen multiplication factors are near to optimum. One can obtain optimum multiplication factors however, by employing the following relationship:

$$M_1^{P_1} . M_2^{P_2} \ldots M_{2^b-1}^{P_{(2^b-1)}} = 1 \qquad (5.1)$$

where $M_i$ is the multiplication factor corresponding to the ith slot and $P_i$ is the probability that the ith slot is occupied in an optimized non-adaptive quantizer.

The prediction coefficients in the present system are quantized non-adaptively. However, it is expected that the overall speech quality will improve if an adaptive quantization scheme were to be employed.

### 5.2.2. Effects of Random Channel Error

The effects of channel errors on the performance of the encoder have been omitted in this study. It seems that the proposed coder is relatively insensitive to the channel errors. This insensitivity can be related to the inverse transformation which is performed at the receiver end. Thus, error spikes, which are subjectively very annoying, are distributed over the output samples by the

inverse transformation. These error spikes have nearly a flat distribution after performing the inverse transformation, hence, subjectively they will be more tolerable.

### 5.2.3. Hardware Implementation of the Modelling and Quantization Part

In Chapter 3 a new approach of implementation of digital signal processors was proposed. This technique was adopted for the orthogonal transformations and their inverse transformations, in Chapter 4. The implementation of the NULP method was not investigated in this study. Although, the techniques reported in Chapter 3 could be employed in a straightforward manner for this implementation, significant reduction in the hardware complexity seems to be possible if the particular properties of the coder were incorporated. The main reason for disregarding this part is that these coders are designed for an optimum efficiency minimizing the prediction error, whilst it is possible, by considering a sub-optimum case, to reduce the complexity significantly. One such sub-optimum case (and very near to the optimum) case can be obtained when the coefficients with insignificant importance towards the quality of sound are deleted. The other possibility is to constitute a sub-optimum prediction matrix in which the number of base-band coefficients used for prediction of other coefficients is reduced. By using this approach the number of multiplications can be significantly reduced as compared with the optimum case, with a degradation in the quality of the recovered speech which is insignificant.

## 5.3. Suggestions for Further Research

### 5.3.1. Voiced-Unvoiced Decision Based on Orthogonal Transformations

The correlation characteristics of the voiced and unvoiced signals are a useful tool for the seperation of these sounds. The

all-pole linear prediction analysis which utilised the correlation characteristics of a signal, has been employed by others in the voiced-unvoiced decision[53]. It can be shown, however, that the different energy distribution of these sounds in the transform domain is also the result of such different correlation characteristics. This differerence in the energy distribution was utilised in the voiced-unvoiced seperation, in this presented work. This technique has a low cost when used in conjunction with the transform coding methods, since the spectrum of speech signals would be available. However, the NULP method could be employed for the voiced-unvoiced seperation and eliminate the necessity of variance distribution measurements. One method can be based on the measurement of the prediction error, similar to the method which employs time domain LP([53]). Thus for a prediction error greater than a certain threshold, the block is considered unvoiced. This is because the NULP coding is inefficient in representing the unvoiced signals. By using the prediction coefficients, obtained from the NULP analysis one may also be able to detect the voiced-unvoiced signals where large values of the prediction coefficients indicate voiced intervals, while small values represent unvoiced parts. This is because large values of prediction coefficients result in less prediction error, as shown in Chapter 2.

The coding strategy of unvoiced sounds, adopted in this study is accomp lished by adaptively quantizing and transmitting those coefficients which on average carry more energy, and disregarding the remaining coefficients. The bit allocation scheme for the chosen coefficients, in this case, has been 2 bits per coefficient. Since unvoiced signals are noise-type signals, their quantization noise is more tolerable. By allocating less than 2 bits per coefficient and still keeping the adaptibility of quantization, one can transmit more unvoiced coefficients without increasing the overall bit-rate. This can be accomp lished by employing the approach adopted in Crochiere[14], in which each coefficient is quantized by $1 + \frac{1}{K}$ bits. This quantization scheme represents each coefficient by two levels

where after each K samples, the step-size of this two level quantizer is adapted to the change of the short term statistics of speech signals.

### 5.3.2. Non-Uniform Linear Prediction Vocoder

Pitch synchronous NULP analysis and synthesis is expected to result in a more efficient speech data compression in trade off between a higher complexity resulting from the pitch detection. This efficiency is predicted to be from two sources: (i) The transform coefficients are expected to be more correlated when the analysis intervals are equal to the pitch period; (ii) The semi-periodic property of voiced sounds can be utilised in updating the transmitted parameters in every few intervals rather than in every interval. The block diagram of a NULP vocoder is shown in Fig. (5.1). In the analyser, after pitch extraction, the input signal is sampled at a variable rate so that each pitch interval consists of N samples. The sampled signal is then transformed and modelled by the NULP scheme in a similar way to that discussed in Chapter 2. The NULP parameters together with the pitch interval size and a binary signal for the voiced-unvoiced are quantized and transmitted. By using this vocoder, it is estimated that intelligible speech can be reproduced at less than 1 kbit/s, with a complexity significantly less than the conventional linear prediction vocoders.

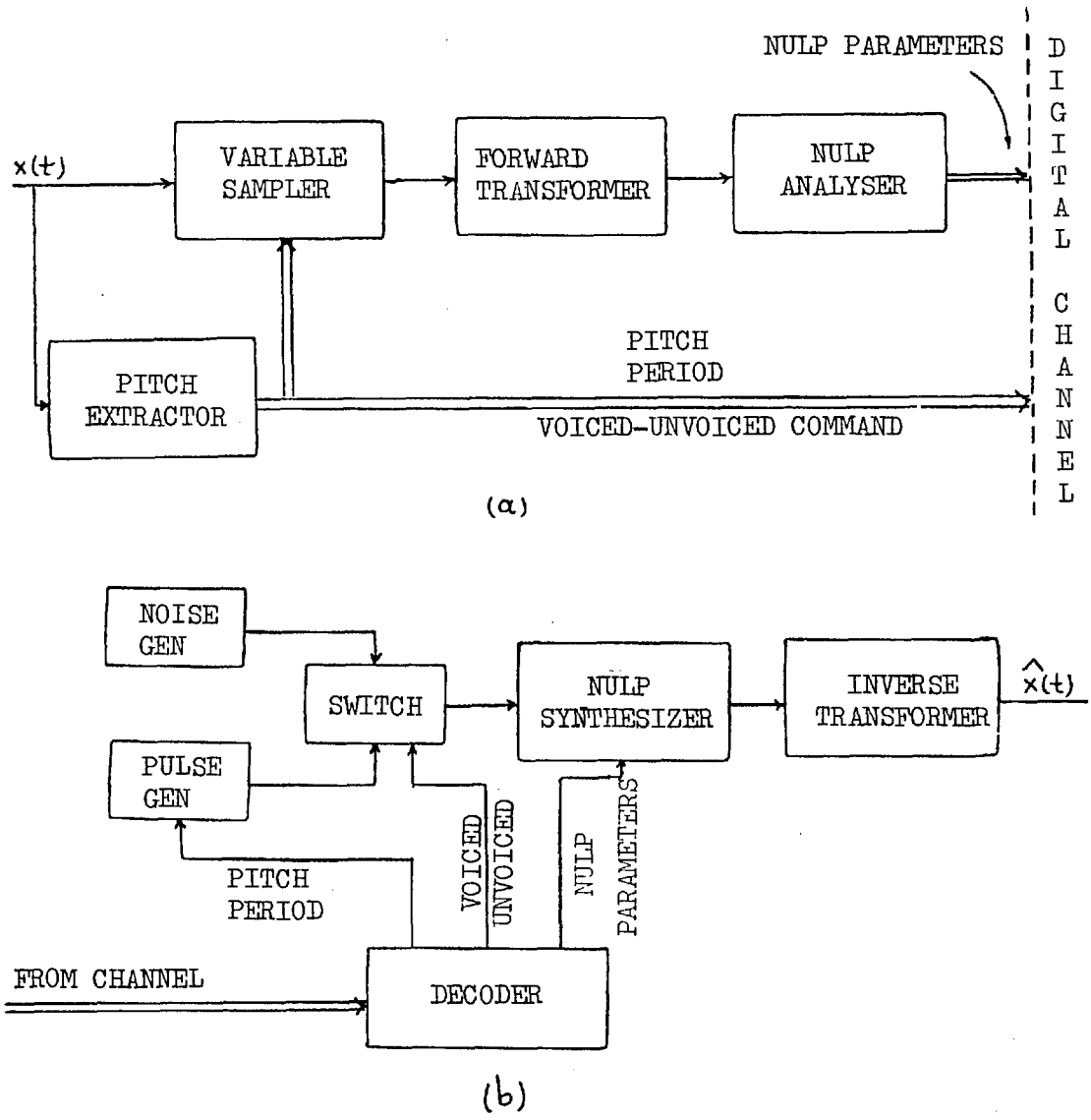



Fig. 5.1. Non-Uniform Linear Prediction vocoder system

## APPENDIX

### The Effects of Finite Register Length Error on Discrete Cosine Coefficients

The contribution of quantization noise is found by starting from Eq. (4.33) and re-arranging it to

$$\sigma_q^2(k) = \left[\sum_{n=1}^{N-1} (a_n - a_{n-1})^2 + a_0^2 + a_{N-1}^2\right] \sigma_n^2/2 \qquad (A.1)$$

where $\sigma_n^2$ is given by Eq. (4.14). By substituting the DC function samples in the above equation instead of $a_n$ we will obtain

$$\sigma_q^2(k) = \left\{\sum_{n=1}^{N-1} \left[\cos\frac{(2n+1)k\pi}{2N} - \cos\frac{(2n-1)k\pi}{2N}\right]^2 + \cos^2\frac{k\pi}{2N} + \cos^2\frac{(2n-1)k\pi}{2N}\right\}\sigma_n^2/2 \qquad (A.2)$$

By considering the following relationships

$$\sin^2\frac{\pi k}{2N} + \cos^2\frac{\pi k}{2N} = 1 \qquad (A.3)$$

$$\sum_{m=1}^{N-1} \sin^2\frac{m\pi k}{N} = \frac{N}{2}$$

Eq. (A.2) can then be compacted to yield Eq. (4.34) or Eq. (A.4), below

$$\sigma_q^2(k) = \left[\left(\frac{N}{2} - 1\right)\sin^2\frac{\pi k}{2N} + 1\right]\sigma_n^2 \qquad (A.4)$$

The total amount of error is the sum of the quantisation error, given by Eq. (A.4), and the error of multiplications. The total error variance for the DC transformer which employs the first type of IBN multiplier, discussed in Chapter 3, is given by

$$\sigma_{t1}^2(k) = \left[ \left(\frac{N}{2} - 1\right) \operatorname{Sin}^2 \frac{k\pi}{2N} + 1 \right] \frac{8}{3B^2} + \frac{(N-1)2^{-2b}}{3} \qquad (A.5)$$

where the second term of the right side of Eq. (A.5) indicates the multiplier noises obtained by using Eq. (3.16).

Similarly, the total error variance of the transformer using the second type IBN multiplier is given by Eq. (A.6), below, by using Eq. (3.19)

$$\sigma_{t2}^2(k) = \left[ \left(\frac{N}{2} - 1\right) \operatorname{Sin}^2 \frac{k\pi}{2N} + 1 \right] \frac{8}{3B^2} + \frac{N-1}{B^2} \qquad (A.6)$$

where B is the IBN word-length and k indicates that the obtained value of error related to the kth DC coefficient.

## REFERENCES


(1) H.A. Black, "Modulation Theory", Princeton, N.J., Van Nostard, 1953.

(2) N.S. Jayant, "Digital Coding of Speech Waveform", Proc. IEEE, vol. 62, pp. 611-632, May 1974.

(3) R.W. Schafer and L.R. Rabiner, "Digital Representation of Speech Signals", Proc. IEEE, vol. 63, No. 4, pp. 662-677, 1975.

(4) J. Max, " Quantizing for Minimum Distortion", IRE Trans. Inform. Theory, vol. IT-6, pp. 7-12, March 1960.

(5) P. Cummiskey, N.S. Jayant and J.L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech", Bell Syst. Tech. J., pp. 1105-1118, Sept. 1973.

(6) N.S. Jayant, "Adaptive Delta Modulation with a One-word Memory", Bell Syst. Tech. J., pp. 321-342, March 1970.

(7) R.A. McDonald, "Signal-to-Noise and Idle Channel Performance of DPCM Systems - Particular Application on Voice Signals", Bell Syst. Tech. J. pp. 1123-1151, 1966.

(8) P. Noll, " A Comparative Study of Various Quantization Schemes for Speech Encoding", Bell Syst. Tech. J., vol. 54, pp. 1597-1614, Nov. 1975.

(9) R. Steele, "Delta Modulation Systems", Pentech Press, London, 1975.

(10) J.B. O'Neal, "Delta Modulation Quantizing Noise Analytical and Computer Simulation Results for Gaussian and Television Signal", Bell Syst. Tech. J., vol. 48, No. 1, pp. 117-141, 1966.

(11) R.A. King and W. Gosling, "Time-Encoded Speech",Electron.Lett. No. 15, pp. 456-457, July 1978.

(12) R.E. Crochiere, S.A. Webber and J.L. Flanagan, "Digital Coding of Speech in Sub-Bands", Bell Syst. Tech. J., Vol. 55, pp. 1069-1085, Oct. 1976. See also Proc. 1976 IEEE Int. Conf. on Acoust. Speech, and Sig. Proc., pp. 233-236.

(13) R.E. Crochiere, "On the Design of Sub-Band Coders for Low-Bit-Rate Speech Communication", Bell Syst. Tech. J., vol. 56, pp. 747-771, May June 1977.

(14) R.A. Crochiere and M.R. Sambur, "A Variable-Band Coding Scheme for Speech Encoding at 4.8 KB/S", Bell Syst. Tech. J., vol. 56, pp. 771-779, May June 1977. See also Proc. 1977 IEEE Int. Conf. on Acoust. Speech and Sig. Proc., pp. 444-447.

(15) L.L. Beranek, "The Design of Communication System", Proc. IRE, 35, pp. 880-890, Sept. 1947.

(16) J.L. Flanagan, "Automatic Extraction of Formant Frequencies from Continuous Speech", J. Acoust. Soc. Am., vol. 28, pp. 110-118, 1956.

(17) J.L. Flanagan and A.S. House, "Development and Testing of a Formant-Coding Speech Compression System", J. Acoust. Soc. Am., vol. 28, pp. 1099-1106, 1956.

(18) G. Fant, "On the Predictability of Formant Levels and Spectrum Envelopes from Formant Frequencies", in For Roman Jackobson, S. Gravenhage, pp. 109-120, 1956.

(19) C.H. Coker, "Computer-Simulated Analyser for a Formant Vocoder", J. Acoust. Soc. Am., vol. 35, p. 1911, Oct. 1963.

(20) J.D. Markel, "Digital Inverse Filtering - A New Tool for Formant Trajectory Estimation", IEEE, Trans. Audio and Electroacoust., vol. AU-20, pp. 129-137, 1972.

(21) G.S. Kang, D.C. Coulter, "600-Bit-Per Second Voice Digitizer (Linear Prediction Formant Vocoder)", Report No. 8043, Naval Research Lab., Washington D.C., Nov. 1976.

(22) J.L. Flanagan, C.H. Coker, L.R. Rabiner, R.W. Schafer and N. Umeda, "Synthetic Voices for Computers", IEEE Spectrum, 7 (No. 10), pp. 22-45, 1970.

(23) A.V. Oppenheim, R.W. Schafer, and T.G. Stockham, "Non-Linear Filtering of Multiplied and Convolved Signals", Proc. IEEE, vol. 56, pp. 1264-1291, Aug. 1968.

(24) A.V. Oppenheim and R.W. Schafer, "Digital Signal Processing", Englewood Cliffs, N.J.: Prentice Hall, 1975.

(25) A.V. Oppenheim, "A Speech Analysis Synthesis System Based on Homomorphic Filtering", J. Acoust. Soc. Amer., vol. 45, pp 458-465, Feb. 1969.

(26) R.W. Schafer and L.R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech", J. Acoust. Soc. Amer., vol. 47, No. 2, pp. 634-648, Feb. 1970.

(27) J.L. Flanagan, "Speech Analysis, Synthesis and Perception, 2nd ed., Springer Verlag, New York, 1972.

(28) J.N. Holmes, "Speech Synthesis", Mills and Boon, London, 1972.

(29) H. Rademacher, "Einige Sätzevon Allgemeinen Orthogonalfunktionen", Math. Annalen 87, pp. 122-138, 1922.

(30) J.L. Walsh, "A Closed Set of Orthogonal Functions", Amer. J. of Mathematics 45, pp. 5 - 24, 1923.

(31) N. Ahmed and K.R. Rao, "Orthogonal Transforms for Digital Signal Processing", Springer-Verlag, 1975.

(32) H.F. Harmuth, "Transmission of Information by Orthogonal Functions," 2nd ed. Germany, Springer-Verlag, 1972.

(33) J.L. Shanks, "Computation of the Fast-Walsh Fourier Transforms," IEEE Trans. on Computers, vol. C-18, pp. 457-459, May 1969.

(34) Y.A. Geadah and M. Corinthios, "Natural, Dyadic and Sequency Order Algorithms and Processors for the Walsh-Hadamard Transform", IEEE Trans. on Computers, vol. C-26, No. 5, May 1977, pp. 435-442.

(35) J.W. Manz, "A Sequency-Ordered Fast Walsh Transform", IEEE Trans. Audio and Electro Acoustics AU-20, pp 204-205, 1972.

(36) J.W. Cooley and J.W. Tukey, "An Algorithm for Machine Computation of Complex Fourier Series", Mathematics of Computation, vol. 19, pp. 297-301, 1965.

(37) N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform", IEEE Trans. Computers, C-23, pp. 90-93, 1974.

(38) R.M. Haralick, "A Storage Efficient Way to Implement the Discrete Cosine Transform", IEEE Trans. Computers, C-23, p. 764, 1976.

(39) W. Chen, S.H. Smith and S.C. Fralick, "A Fast Computational algorithm for the Discrete Cosine Transform", T-Comu., p. 1004, Sept. 1977.

(40) R. Bellman, "Introduction to Matrix Analysis", New York, McGraw-Hill, 1960.

(41) T. Berger, "Rate Distortion Theory", Englewood Cliffs, N.J. Prentice-Hall, 1971.

(42) J. Huang and P. Schultheiss, "Block Quantization of Correlated Gaussian Randon Variables", IEEETrans. Commu. Syst., vol. CS-11, pp. 289-296, 1963.

(43) P.A. Wintz, "Transform Picture Coding", Proc. IEEE, vol. 60, pp. 809-820, 1972.

(44) R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals", IEEE Trans. Acous. Speech Signal Proc., vol. ASSP-25, pp. 299-309, August 1977.

(45) S.J. Campanella and S.G. Robinson, "A Comparison of Orthogonal Transformations for Digital Speech Processing", IEEE Trans. Commu. Technol., vol. COM-19, pp. 1045-1049, 1971.

(46) H. Gethöffer, "Speech Processing Using Walsh Functions", in Proc. 1972 Symp. on Theory and Applications of Walsh Functions, Naval Res. Lab., Washington D.C., pp. 163-168.

(47) H. Gethöffer, "Transform Coding of Speech Using Walsh Functions", in Proc. 1973, Symp. on Theory and Application of Walsh Functions, Naval Res. Lab., Washington D.C., pp. 194-201.

(48) G. Modena, L. Nebbia and C. Scagliola, "Bit Rate Reduction of Digital Speech Transmission by Linear Transformation", in Signal Processing, W, Schüssler, Ed., Erlangen, Germany, 1973.

(49) G.G. anderson and T.S. Huang, "Piece Wise Fourier Transform for Picture Bandwidth Compression", IEEETrans. Commu., Technol., Vol. COM-19, pp. 133-140, 1971.

(50) F. Shum, A. Elliot and W. Brown, "Speech Processing with Walsh-Hadamard Transforms", IEEE Trans. Audio Electroacoustic, vol. AU-21, pp. 174-179, 1973.

(51) A. Peled and B. Liu, "A New Approach to the Relaization of Non-Recursive Digital Filters", IEEE Trans. Audio Electro-acous., vol. AU-21, No. 6, December 1973.

(52) S. Ahmadi and A.G. Constantinides, "Formant Sequency Approach for Low-Bit-Rate Speech Transmission", Proc. 1978 Int. Conf., Signal Processing pp. 481-489, Florence, Italy.

(53) B.S. Atal and S.L. Hanaver, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", J. Acoust. Soc. Amer., vol. 50, pp. 637, August 1971.

(54) F. Itakura and S. Saito, "Analysis Synthesis Telephony Based on the Maximum Likelihood Method", in Ref. 6th Int. Cong. Acoust. Y. Kohasi, Ed., pp. C17-C20, August 1968.

(55) J.D. Markel and A.H. Gray, Jr., Linear Prediction of Speech, Springer-Verlag, 1976.

(56) J. Makhoul, "Linear Prediction: A Tutorial Review", Proc. IEEE, vol. 63, No. 4, pp. 561-580, April 1975.

(57) J.C. Chow, "On Estimating the Orders of an Autoregressive Moving Average Process with Uncertain Observations", IEEE Trans. Automat. Contr., vol. AC-17, pp. 707-709, 1972.

(58) A.G. Evans and R. Fischl, "Optimal Least Squares Time-Domain Synthesis of Recursive Digital Filters", IEEE Trans. Audio Electroacoust. vol. AU-21, pp. 61-65, 1973.

(59) M.A. Atashroo, "Auto-Correlation Prediction", Proc 1977 IEEE Int. Conf. Acoust. Speech Sig. Proc., pp. 732-735.

(60) L.R. Rabiner, et. al., "A Comparative Performance Study of Several Pitch Detection Algorithsm", IEEETrans. Acoust. Speech Sig. Proc., vol. ASSP-24, Oct. 1976.

(61) M.R. Sambur, et. al., "On Reducing the Buzz in LPC Synthesis", Proc. 1977 IEEE Int. Conf. Acoust. Speech Sig. Proc., pp. 401-404.

(62) J. Makhoul, et. al., "A Mixed Source for Speech Compression and Synthesis", Proc. 1978 IEEE Int. Conf. Acoust. Speech Sig. Proc., pp 163-166.

(63) S. Chandra and W.C. Lin, "Linear Prediction with a Variable Analysis Frame Size", IEEETrans. Acoust. Speech Sig. Proc., vol. ASSP-25, 1977.

(64) J.N. Holmes, "Variable-Frame-Rate Coding Scheme for Speech Analysis/Synthesis Systems", Electron. Lett, April 1974.

(65) J.G. Dunn, "An Experimental 9600 bits/$_s$ Voice Digitizer Employing Adaptive Prediction", IEEETrans. Commu. Tech. vol. COM-19, pp. 1021-1032, 1971.

(66) P.W. Noll, "Adaptive Quantization in Speech Coding Systems", Int. Zurich Seminar on Digital Communication (IEEE) pp. B. 31 to B. 36, October 1976.

(67) R.E. Crochiere, et. al., "A Study of Objective Measures for Speech Waveform Coders", Proc. 1978 Zurich Seminar on Digital Communications, March 1978.

(68) B.J. McDermott, "Perceptual and Objective Evaluation of Speech Processed by Adaptive Differential PCM", Proc. 1978, IEEE Int. Conf. Acoust. Speech Sig. Proc. pp. 581-584.

(69) J.M. Pribolet, et. al., "A Study of Complexity and Quality of Speech Waveform Coder", Proc. 1978 IEEE Int. Conf. Acoust. Speech Sig. Proc., pp. 586-590.

(70) R. Viswanathan, et. al., "Objective Speech Quality Evaluation of Narrowband LPC Vocoder", Proc. 1978, IEEE Conf. Acoust. Speech Sig. Proc., pp. 591-594.

(71) T.F. Barnwell and A.M. Bush, "Statistical Correlation Between Objective and Subjective Measures for Speech Quality", Proc. 1978 IEEE Acoust. Speech Sig. Proc., pp. 595-598.

(72) B. Smith, "Instantaneous Companding of Quantized Signals", Bell Syst. Tech. J., pp. 653-709, 1957.

(73) N.S. Jayant, "Adpative Quantization with a One-Word Memory", Bell Syst. Tech. J., vol. 52, pp. 1119-1144, 1973.

(74) M. Chen, "The Low Bit-Rate Coding of Speech Signals", Ph.D. dissertation, Dept. Elec. Eng., Imperial College, Un iversity of London, 1976.

(75) L.R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Englewood Cliff, N.J., Prentice-Hall, 1975.

(76) G.B. Lockhart, "Binary Transversal Filters with Quantized Coefficients" Electronic Letters, Vol. 7, No. 11, pp. 305-307, June 1971.

(77) G.B. Lockhart, "Digital Encoding and Filtering Using Delta Modulation", The Radio and Electronic Engineer, Vol. 42, No. 12, pp. 547-551, 1972.

(78) D. Lagoyannis, "Correlator Based on Delta-Sigma Modulation" Electronic Letters, Vol. 12, No. 10, pp. 253-254, May 1976.

(79) I.A. Dempster, and R. Steele, "Delta-Sigma Wattmeter", Electronic Letters, Vol. 7, No. 18, pp. 519-520, Sep. 1971.

(80) D.J. Goodman, "The Application of Delta Modulation to Analogue-to-PCM Encoding", Bell Syst. Tech. J., Vol. 48, No. 2, pp. 321-343, 1969.

(81) L.B. Jackson, J.F. Kaiser and H.S. McDonald, "An Approach to the Implementation of Digital Filters", IEEE Trans. on Audio Electroacous., Vol. AU-16, No. 3, pp. 413-421, 1968.

(82) G.D. Bergland, "Fast Fourier Transform Hardware Implementation - An Overview", IEEE Trans. Audio Electroa., vol. AU.17, pp. 104-107, 1969.

(83) R.R. Shively, "A Digital Processor to Generate Spectra in Real Time", IEEE Trans. Computers, Vol. C-17, pp. 485-491, May 1968.

(84) G.D. Bergland and H.W. Hale, "Digital Real-Time Spectral Analysis", IEEE Trans. Electronic Computers, Vol. EC-16, pp. 180-185, 1967.

(85) R.A. Smith, "A Fast Fourier Transform Processor", Bell Telephone Labs., Inc. Whippany, N.J., 1967.

(86) M.C. Pease, J. Goldberg, "Feasibility Study of a Special Purpose Digital Computer for on-line Fourier Analysis", Advanced Research Projects Agancy, Order 989, May 1967.

(86) M.C. Pease, J. Goldberg, "Feasibility Study of a Special Purpose Digital Computer for on-line Fourier Analysis", Advanced Research Projects Agancy, Order 989, May 1967.

(87) G.D. Bergland and D.E. Wilson, "An FFT algorithm for a Global, Highly Parallel Processor, IEEE Trans. Audio Electroa. Vol. AU.17, pp. 125-127, 1969.

(88) R.B. McCullough, "A Real Time Digital Spectrum Analyser" Stanford Electronic Labs, Stanford, Calif., Sci. Rept. 23, Nove. 1967.

(89) H.L. Groginsky and G.A. Works, "A Pipeline Fast Fourier Transform", IEEE Trans. Computers, Vol. C-19, pp 1015-1019, 1971.

(90) B. Liu and A. Peled, "A New Hardware Realisation of High Speed Fast Fourier Transforms", IEEE Trans. Acoust. Speech Sig. Proc., Vol. ASSP-23, pp. 543-547, 1975.

(91) R. Walker, "Hadamard Transformation - A Real-Time Transformer for Broadcast Standard PCM Television", Colliqum, Hatfield Polytechnique, U.K., 1973.

(92) S.C. Noble, "A Comparison of Hardware Implementations of the Hadamard Transform for Real-Time Image Coding", Proc. Soc. Photo-optical Instrumen. Engs. Vol. 66, San Diego, Calif., pp. 207-211 1975.

(93) A. Habibi et. al. "Real Time Image Redundancy Reduction Using Transform Coding Techniques", Int. Commu. Conf., June 1974.

(94) M.R. Ashouri and A.G. Constantinides, "A Pipeline Fast Walsh-Fourier Transform", in Proc. 1977 IEEE Int. Conf. on Acoustic, Speech and Signal Processing, pp.515-519.

(95) M.R. Ashouri and A.G. Constantinides, "Linear Prediction Techniques in the Walsh Spectral Domain for Speech Analysis and Synthesis", in Proc. 1978 IEEE Int. Conf. on Acoustic Speech and Signal Processing, pp. 476-479.