DISCRETE HILBERT TRANSFORM FILTERING

by

M. Shaker SABRI

A thesis submitted to the School of Graduate Studies in partial fulfilment of the requirements for the degree of Ph.D. in Electrical Engineering

UNIVERSITY OF OTTAWA

OTTAWA, CANADA, 1976

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Prof. Willem Steenaart for his expert and excellent guidance throughout the course of this research.

The author also would like to thank Mme L. LeBlanc for typing this thesis.

This research was partially supported by the National Research Council of Canada.

ABSTRACT

A discrete filtering technique based on the discrete Hilbert transform (DHT) is presented. The filtering method and its implementation are shown to compare favorably with the discrete Fourier transform (DFT) or the Fast Fourier transform(FFT) filtering. The comparison is presented in terms of speed, error accumulation and complexity of hardware implementation.

The DHT and the derived discrete filter transforms are each presented in matrix form and may be implemented with the same hardware in either series or parallel form. In this manner a large variety of filtering characteristics may be implemented by changes of the stored coefficients only.

While the filtering process, in matrix form, is a circular convolution process, direct convolution filters are obtainable also by using one row of the matrix only and are realizable in the form of a non-recursive finite impulse response (FIR) digital filters.

An interesting application of the filtering matrix method for bandlimited signal extrapolation is presented. The total extrapolation process is achieved by a single matrix operation. All known band-limited signal extrapolation techniques require large computations, can be implemented only on a general purpose computer and cannot be applied in real time. The proposed technique and its implementation has the advantages that it requires much less computation , gives more accurate results,and can be applied in real time.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

The Hilbert Transform [1,2] has played an important role in signal processing and communication systems. It has a variety of applications such as generation of analytic signals [3], representation of bandpass processes [2], generation of single side band modulated signals [4], and filtering [5]. The discrete Hilbert transform (DHT) will be equally useful in digital signal processing and digital communication systems.

Part of this thesis handles the filtering aspects of the DHT i.e. realization of the DHT itself and realization of different filtering processes based on the DHT.

Digital filter realization of the DHT has been presented [6,7] in the form of nonrecursive FIR filters.

In chapter two a simple and accurate realization of the DHT is presented in the form of the Hilbert matrix i.e. transformation is done in matrix form. The derivation of the Hilbert matrix is presented and its properties are discussed.

To realize filters using the Hilbert transform [5] involve modulators, wide band 90^{o} phase shifters (Hilbert transformers) and demodulators in two or more branches which may limit its practical applicability for both analog and digital filters.

The Hilbert matrix with suitable coefficient modifications, realizes discrete frequency domain filters, in matrix form, thus eliminating the need for modulators or demodulators. A circular convolution process is established which compares well with the DFT (Discrete Fourier transform) or FFT (Fast Fourier transform) techniques.

It is shown that different types of discrete filters with near ideal discrete frequency responsesare realizable in single matrix form. Also this approach is generalized to a general discrete filtering process in matrix form.

The properties of different filter matrices are discussed and two basic implementations are proposed. A comparison between the matrix approach and the DFT or FFT filtering approach is made in terms of complexity of hardware and speed of operation.

In actual implementation of the DHT filtering technique, quantization and arithmetic errors are introduced due to finite word length representation of the input signal, matrix coefficients, and the product of arithmetic operations. In chapter three these sources of error are analysed for both fixed point and floating point arithmetic. Expressions for the output mean square error (MSE) to mean square signal (MSS) ratio are obtained and compared to those of the FFT. It is shown that the DHT filtering technique introduces significantly less error than the FFT.

Chapter four deals with an interesting application of the filtering matrix method for bandlimited signal extrapolation. Several extrapolation techniques have been reported in the literature [8,9,10,11,12,13]. Among these techniques is the prolate spheroidal expansion technique [11] and the iterative extrapolation technique [8,9] where a priori assumption of the signal band limits is made. A common feature of all these techniques is that they require extensive computations and can be implemented only on a general purpose computer or in other words cannot be applied in real time.

In the iterative extrapolation technique as presented by Papoulis [8] and Gerchberg [9], the DFT or FFT and their inverses are used iteratively to extrapolate the unknown part of the signal. Papoulis [8] has shown that this technique is basically the same as the prolate spheroidal expansion method but gives considerable computational

and storage savings as compared with the prolate spheroidal expansion method since it involves only DFT's and their inverses and can make use of the available computational savings using the FFT to evaluate the DFT.

Nevertheless, with the DFT or FFT used iteratively, a considerable amount of computation has to be performed. For example, if n is the number of iterations required to extrapolate the unknown part of the signal, the DFT or FFT has to be performed 2n times i.e. if the FFT is used, 2n N log N complex arithmetic operations are required. Often, the convergence of this algorithm is relatively slow and a large number of iterations is required. This would form a serious obstacle in applying such a technique in real time.

In chapter four an extrapolation technique is presented, where the total extrapolation process is achieved by a single matrix operation. First, the iterative extrapolation technique [8,9] is reviewed then a new model for the iterative extrapolation technique is presented which leads to the derivation of the Extrapolation Matrix equivalent to n iterations. Subsequently, the extrapolation matrix equivalent to an infinite number of iterations is derived. The properties of the extrapolation matrix are discussed and several examples are presented. A comparison is made between the direct extrapolation and the iterative extrapolation using the FFT, in terms of number of arithmetic operations (multiplication and addition) required in both cases, and the accuracy of the results obtained. Finally, a proposed fast extrapolator realization, based on the extrapolation matrix approach is presented which will allow adjustment to any given bandwidth, and can operate in real time.

While the discrete filters presented in chapter two implement a circular convolution process, direct convolutional filters are also realizable using one row of the Hilbert or any of the filter matrices of

chapter two. In chapter five it shall be shown that the DHT filtering can be used to generate filter designs using the middle row of the Hilbert or any of the filter matrices. These filters are realizable in the form of FIR nonrecursive linear phase digital filters. Several design examples are given and the properties of these filters are discussed. Also the analysis of the resulting approximation error and comparison between this approach and other approaches [14, 15, 16] are presented.

In chapter six the conclusion of this work is presented and future research points and extension of this work are discussed.

REFERENCES

[1] M. Schwartz, W.R. Bennet, and S. Stein, "Communication Systems and Techniques, McGraw Hill, 1966.

[2] D. Sakrison, "Communication Theory", Wiley, 1968.

[3] M.R. Schroeder, J.L. Flanagan, and E.A. Landry ", Bandwidth Compression of Speech by Analytic Signal Rooting", Proceedings of the IEEE 1967, 55, pp 396-401.

[4] J.R. Ragazzini and G.F. Franklin, "Sampled Data Control Systems", McGraw Hill 1958.

[5] G.D. Cain, "Hilbert Transform Description of Linear Filtering", Electronics Letters, July 1972, Vol. 8, No. 15, pp 380-382.

[6] Hermann, O., "Transversal Filter zur Hilbert-Transformation" Arch. Elek, Ubertrag., 1969, 23, pp. 581-587.

[7] McClellan, J., et al., "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters", IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, No. 6, December 1973, pp. 506-525.

[8] A Papoulis, "A New Algorithm in Spectral Analysis and Band-Limited Extrapolation", IEEE Trans. on Circuits and Systems, Vol CAS-22, No. 9, September 1975 pp 735-742.

[9] R.W. Gerchberg, "Super Resolution Through Error Energy Reduction", Optica Acta, Vol 21, No 9, pp 709-720, 1974.

[10] T.P. Burg "Maximum Entropy Spectral Analysis", presented at the 37th Int. SEG Meeting, Oct. 1967.

[11] D. Slepian, H.O. Pollak, and H.J. Landolt, "Prolate Spheroidal Wave Functions", Bell Syst. Tech. J., Vol 40, pp 43-84, Jan. 1961.

[12] T.J. Ulrych, "Maximum Entropy Power Spectrum of Truncated Sinusoids", J. Geophys. Res., Vol 77, pp 1396-1400, 1972.

[13] J.A. Edward and M.M. Fitelson, "Notes on Maximum-Entropy Processing", IEEE Trans. Inform. Theory, Vol IT-19, pp 232-234, March 1973.

[14] A. Oppenheim and R. Schafer, "Digital Signal Processing", Prentice Hall, 1975, pp 337-365.

[16] T. Parks and J. McClellan, "Chekyshev Approximation for Nonrecursive Digital Filters with Linear Phase", IEEE Trans. Circuit Theory, Vol CT-19, March 1972 pp 189-194.

[16] J. McClellan, T. Parks, and L. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters", IEEE Trans. on Audio and Electroacoustics , AU-21, No. 6, December 1973 pp 506-526.

CHAPTER TWO

DISCRETE HILBERT TRANSFORM FILTERING

2.1 INTRODUCTION

The Hilbert transform [1,2] has played an important role in signal processing and communication systems. The discrete Hilbert transform [3,4,5] will be equally useful in digital signal processing and digital communication systems.

Digital filter realization of discrete Hilbert transform (DHT) have been presented [7,8] in the form nonrecursive (FIR) digital filters.

In this chapter, a simple and accurate realization of the DHT is presented in the form of the Hilbert matrix, i.e. the transformation is done in matrix form.

To realize filters using the Hilbert transform [6] involves modulators, wide band 90^{o} phase shifters, and demodulators in two or more branches, which may limit its practical applicability for both analog and digital filters.

The Hilbert matrix, with suitable coefficient modifications, realizes discrete frequency domain filters directly, thus eliminating the need for modulators or demodulators. A circular convolution process is established which compares well with the DFT or FFT techniques as one matrix multiplication process is required only on each set of N input samples.

It is shown that different types of discrete filters with near ideal discrete frequency response are realizable in single matrix form. Also this approach is generalized to a general discrete filtering process in matrix form.

The analysis leading to the Hilbert matrix and different discrete

filter matrices is presented. The properties of these matrices are discussed.

Due to the symmetry properties of the Hilbert and filter matrices two basic realizations are proposed for actual implementation of these filters. Finally a comparison between the proposed matrix approach and the FFT or DFT techniques is made in terms of speed and complexity of hardware. In chapter 3 we shall present the comparison in terms of error accumulation after analyzing the different sources of error .

2.2 THE HILBERT MATRIX

The Hilbert transform of a signal f(t) is defined as [1] :

$$g(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} f(\tau) \frac{d\tau}{\tau - t} = \frac{1}{\pi t} * f(t) \qquad (2.1)$$

The Hilbert transform relation can be interpreted as a convolution between f(t) and $\frac{1}{\pi t}$ to produce the signal g(t). Relation (2.1) is expressed in the frequency domain as :

$$G(\omega) = H(\omega) \cdot F(\omega) \qquad (2.2)$$

where $G(\omega)$, $F(\omega)$ and $H(\omega)$ are the Fourier transform of g(t), f(t), and $\frac{1}{\pi t}$ respectively ;

$$H(\omega) = -j \, \mathrm{sgn}(\omega) \qquad (2.3)$$

where,

$$\mathrm{sgn}\,\omega = \begin{cases} 1 & \omega > 0 \\ 0 & \omega = 0 \\ -1 & \omega < 0 \end{cases} \qquad (2.4)$$

An expression for the DHT is obtained [4] by sampling the continuous time function $\frac{1}{\pi t}$, and finding its Fourier transform which is a periodic function of frequency. One period of this function is sampled and the inverse DFT of the sample values is obtained.

Let f_n be the input sample sequence of length N, n = 0, 1, 2, .., N-1.

Replacing $H(j\omega)$ by its discrete representation in the frequency domain H_k, $k = 0, 1, 2, \ldots N-1$, reduces (2.2) to :

$$G_k = H_k \cdot F_k \tag{2.5}$$

where, for N even :

$$H_k = \begin{cases} -j, & k = 0, 1, 2, \ldots, (\frac{N}{2} - 1) \\ 0, & k = 0, \frac{N}{2} \\ +j, & k = (\frac{N}{2} + 1), (\frac{N}{2} + 2), \ldots, (N-1) \end{cases} \tag{2.6}$$

and for N odd :

$$H_k = \begin{cases} -j, & k = 0, 1, 2, \ldots \frac{N-1}{2} \\ 0, & k = 0 \\ +j, & k = \frac{N+1}{2}, \frac{N+3}{2}, \ldots, N-1 \end{cases} \tag{2.7}$$

and,

$$F_k = \text{DFT}\{f_n\} = \frac{1}{N} \sum_{n=o}^{N-1} f_n \exp(-j \frac{2\pi nk}{N}) \tag{2.8}$$

$$G_k = \text{DFT}\{g_i\} = \frac{1}{N} \sum_{i=o}^{N-1} g_i \exp(-j \frac{2\pi ik}{N}) \tag{2.9}$$

Taking the inverse discrete Fourier transform of both sides of (2.5) gives :

$$g_i = \sum_{k=o}^{N-1} H_k \sum_{n=o}^{N-1} f_n \exp[j \frac{2\pi}{N}(i-n)] \tag{2.10}$$

Substituting the values of H_k given by (2.6) or (2.7) into (2.10) and interchanging the order of summations, then performing the summation over k, we obtain the following expression for the DHT [4,5] :

$$g_i = \frac{1}{N} \sum_{n=o}^{N-1} f_n \, a_{i,n} \tag{2.11}$$

where for N even :

$$a_{i,n} = \begin{cases} 2 \text{ cotan } [\frac{\pi}{N}(i-n)] & (i-n) \text{ odd} \\ 0 & i=n, (i-n) \text{ even} \end{cases} \tag{2.12}$$

and, for N odd

$$a_{i,n} = \begin{cases} \text{cotan } [\frac{\pi}{N}(i-n)] - \dfrac{(-1)^{i-n}}{\sin [\frac{\pi}{N}(i-n)]} & i \neq n \\ 0 & i = n \end{cases} \tag{2.13}$$

Converting (2.11) into matrix form :

$$[g] = [H]\,[f] \tag{2.14}$$

where [f] and [g] are the input and output sequences in vector form, both of length N. [H] is the Hilbert matrix which, for N even, is of the form:

$$[H] = \frac{1}{N} \begin{bmatrix} 0 & -a_1 & 0 & -a_3 & \cdots & -a_{N-1} \\ a_1 & 0 & -a_1 & 0 & \cdots & 0 \\ 0 & a_1 & 0 & -a_1 & \cdots & -a_{N-3} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ a_{N-1} & 0 & a_{N-3} & 0 & & 0 \end{bmatrix} \tag{2.15}$$

where, $a_m = 2 \text{ cotan } (\frac{\pi}{N} m)$. (2.16)

and, for N odd:

$$[H] = \frac{1}{N} \begin{bmatrix} 0 & -a_1 & a_2 & -a_3 & \cdots & a_{N-1} \\ a_1 & 0 & -a_1 & a_2 & \cdots & a_{N-2} \\ -a_2 & a_1 & 0 & -a_1 & & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ -a_{N-1} & a_{N-2} & -a_{N-3} & a_{N-4} & \cdots & 0 \end{bmatrix} \tag{2.17}$$

where,
$$a_m = \begin{cases} \text{cotan } (\frac{\pi m}{2N}) & \text{for } m \text{ odd} \\ \tan (\frac{\pi m}{2N}) & \text{for } m \text{ even} \end{cases} \qquad (2.18)$$

and

$$a_m = -a_{N-m} \quad , \quad \text{For both } N \text{ even and } N \text{ odd.}$$

2.2.1 Properties of the Hilbert Matrix :

For both cases of N even and N odd the Hilbert matrix is a square N x N matrix having the following properties:

(i) The matrix is circulant i.e. the $(i+1)^{st.}$ row (or column) of the matrix can be obtained from the i^{th} row by cycling one step to the right i.e. the last element in the i^{th} row is shifted to the first position and the rest of the elements are shifted one place to the right.

(ii) The matrix is skew symmetric.

(iii) For N even and $\frac{N}{2}$ also even the matrix, has only $\frac{N}{4}$ distinct nonzero coefficients (apart from the sign since $a_m = -a_{N-m}$) for N even and $\frac{N}{2}$ odd it has $\frac{N-2}{4}$ distinct nonzero coefficients (apart from the sign).

For N odd the matrix has $\frac{N-1}{2}$ distinct nonzero coefficients (apart from the sign).

(iv) For both cases of N even and N odd the first half of the first row completely defines the matrix. i.e. in order to generate the Hilbert matrix one needs to compute and store only $\frac{N}{4}$ coefficients for N even and $\frac{N}{2}$ even, $\frac{N-2}{4}$ coefficients for N even and $\frac{N}{2}$ odd, and $\frac{N-1}{2}$ coefficients for N odd.

From the previous discussions it can be seen that there is a definite advantage in choosing N even because of the zero elements occuring, which reduces the amount of storage and computation necessary by at least a factor of two as compared to N odd.

2.3 FILTER REALIZATION USING THE HILBERT MATRIX.

2.3.1 Lowpass Filter Matrix and Transform:

The analog lowpass filter configuration of fig. 2.1 consists of two Hilbert transformers, four multipliers and an adder . The digital equivalent will be realizable in total by a single matrix. This is a significant simplification over the realization of the system using two Hilbert transformers.

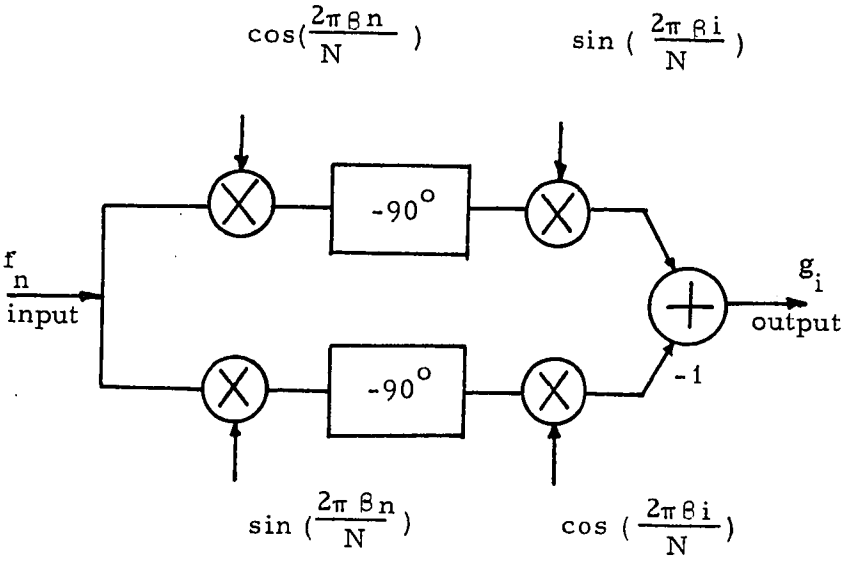
The input output relation for the digital equivalent of fig. 2.1 expressed in matrix form is :

$$\begin{aligned} [g] &= \{[B][H][A] - [A][H][B]\}[f] \\ &= [LP][f] \end{aligned} \tag{2.19}$$

where [f] and [g] are the input and output (filtered) sequences respectively, [H] is the Hilbert matrix. [A] and [B] are diagonal matrices defined as :

$$[A] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \cos\alpha & 0 & \cdots & 0 \\ \vdots & & \cos 2\alpha & & \\ & & & \cos 3\alpha & \\ & & & & \ddots \\ 0 & 0 & \cdots & \cdots & \cos(N-1)\alpha \end{bmatrix} \tag{2.20}$$

$$[B] = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & \sin\alpha & 0 & \cdots & 0 \\ \vdots & \vdots & \sin 2\alpha & & \\ & & & \sin 3\alpha & \\ & & & & \ddots \\ 0 & 0 & \cdots & \cdots & \sin(N-1)\alpha \end{bmatrix} \tag{2.21}$$



$$\frac{\beta}{N} = f_c T$$

Fig. 2.1 Typical lowpass filter configuration using Hilbert transform elements.

where $\alpha = \frac{2\pi\beta}{N}$, β is the filter cutoff frequency expressed as number of sample points in the frequency domain. The lowpass matrix [LP] defines the total process, and for N even is of the form :

$$[LP] = \frac{1}{N} \begin{bmatrix} 0 & b_1 & 0 & b_3 \cdots & b_{N-1} \\ b_1 & 0 & b_1 & 0 \cdots & 0 \\ 0 & b_1 & 0 & b_1 \cdots & b_{N-3} \\ \vdots & \vdots & \vdots & & \\ b_{N-1} & 0 & b_{N-3} & 0 \cdots & 0 \end{bmatrix} \qquad (2.22)$$

where

$$b_m = 2 \operatorname{cotan}\left(\frac{\pi m}{N}\right) \sin\left(\frac{2\pi\beta m}{N}\right), m \text{ is always odd.} \qquad (2.23)$$

For N odd:

$$[LP] = \frac{1}{N} \begin{bmatrix} 0 & b_1 & b_2 & b_3 \cdots & b_{N-1} \\ b_1 & 0 & b_1 & b_2 \cdots & b_{N-2} \\ b_2 & b_1 & 0 & b_1 \cdots & b_{N-3} \\ \vdots & \vdots & \vdots & \vdots & \\ b_{N-1} & b_{N-2} & b_{N-3} & b_{N-4} \cdots & 0 \end{bmatrix} \qquad (2.24)$$

where

$$b_m = \left[\operatorname{cotan}\frac{\pi m}{N} - \frac{(-1)^m}{\sin\frac{\pi m}{N}} \right] \sin\left(\frac{2\pi\beta m}{N}\right) \qquad (2.25)$$

and for both cases :

$$b_m = b_{N-m} \qquad (2.26)$$

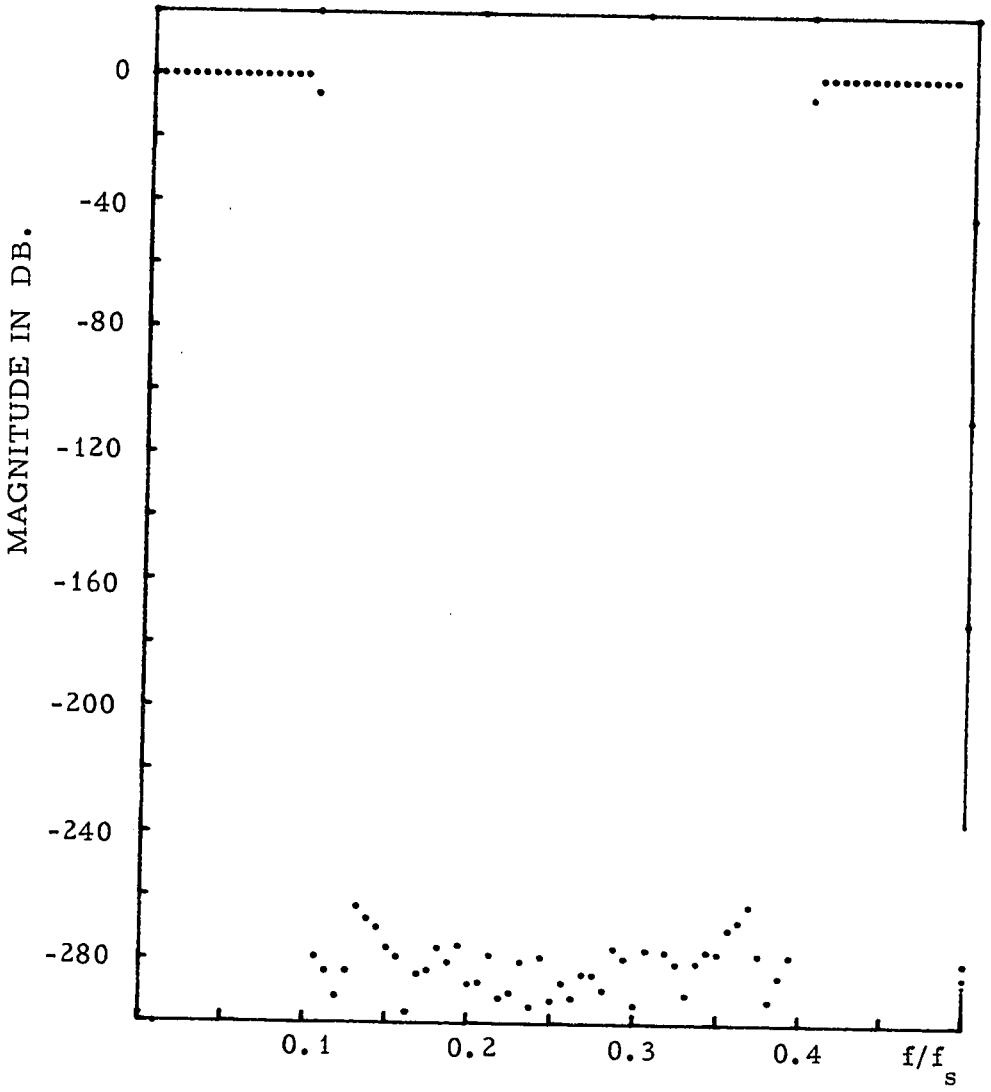


Fig. 2.2. Lowpass filter discrete frequency response for N even N = 160, β = 16

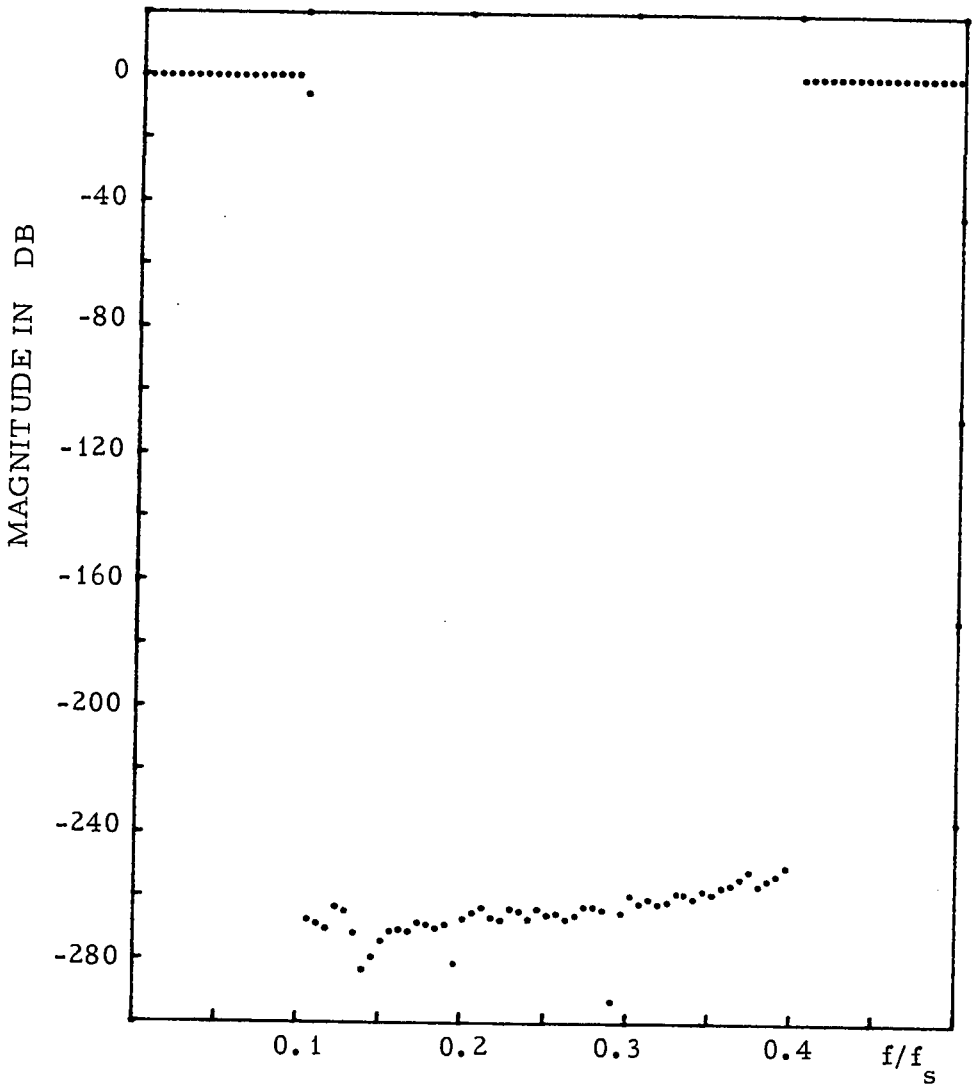


Fig. 2.3. Lowpass filter discrete frequency response for N odd . N = 179, β = 18.

The lowpass matrix has the same properties as the Hilbert matrix except that it is symmetric i.e. $b_m = b_{N-m}$.

The discrete frequency response for typical lowpass filters obtained is shown in fig. 2.2 and fig. 2.3 for N even and N odd respectively. We note that the discrete frequency response is near ideal i.e. attenuation in the stopband is greater than 200 db and ripples in the passband are less than 10^{-4} db, with zero phase response. Also the symmetry in filter characteristics is around $f_s/4$ instead of $f_s/2$ (f_s is the sampling frequency). This can be changed to $f_s/2$ by either doubling the sampling frequency or dropping the zero terms from the matrix for N even.

Alternatively a lowpass matrix can be derived directly from specification in the discrete frequency domain. In this case the matrix is given by :

$$[LP] = \{b_{i,n}\} \tag{2.27}$$

where

$$b_{i,n} = \begin{cases} \dfrac{2\beta+1}{N} & i = n \\ \dfrac{1}{N} \dfrac{\sin 2\pi [(i-n)(\beta+0.5)/N]}{\sin [\pi (i-n)/N]} & i \neq n \end{cases} \tag{2.28}$$

2.3.2 Bandpass Filter Matrix and Transform:

In general a bandpass filter can be realized by a parallel connection of two lowpass filters having the same phase response and different cutoff frequencies. The lowpass filters cutoff frequencies determine the upper and lower pass-band frequency edges. Fig. 2.4 shows the bandpass filter realization using the Hilbert transform. It consists of 4 Hilbert transform elements and 8 multipliers (modems) and three summers. All these operations can be replaced by a single matrix operation.

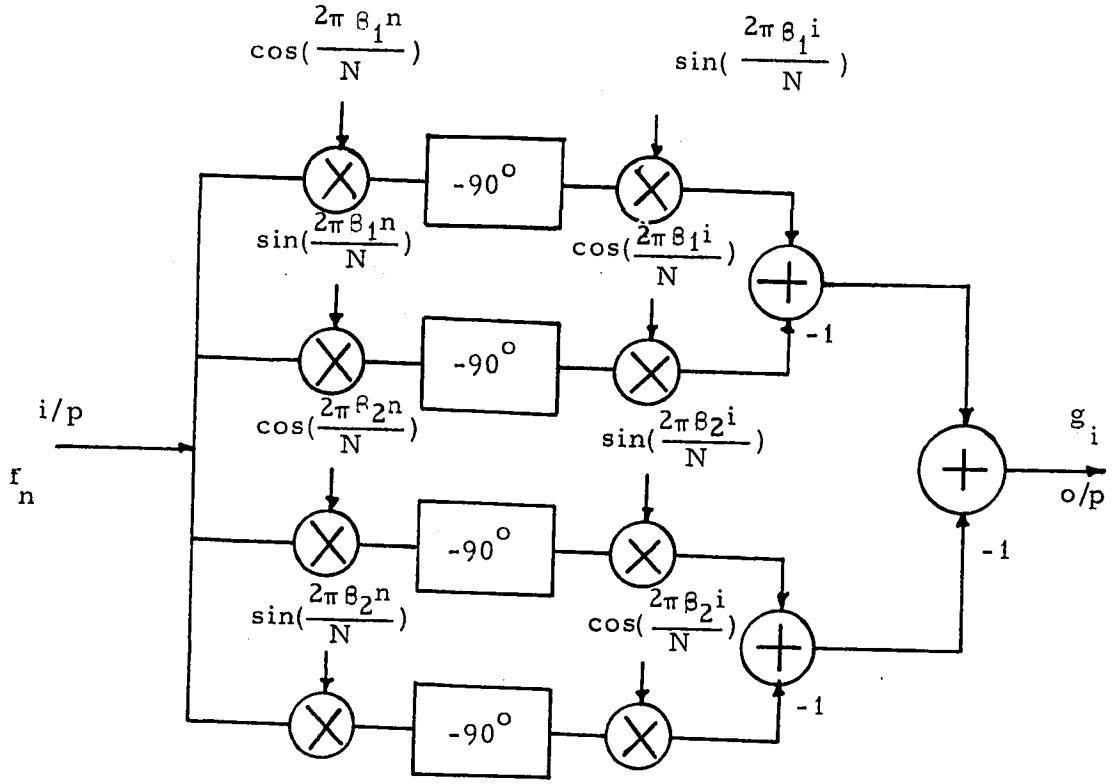


Fig. 2.4. Typical bandpass filter configuration using Hilbert transform elements.

The input output relation of fig. 2.4 expressed in single matrix form is :

$$[g] = \{[LP_2] - [LP_1]\}[f]$$

$$= [BP][f] \qquad (2.29)$$

where $[LP_1]$ and $[LP_2]$ are lowpass filter matrices having cutoff frequencies β_1 and β_2 respectively. $[BP]$ is the bandpass filter matrix, which for N even is of the form :

$$[BP] = \frac{1}{N} \begin{bmatrix} 0 & c_1 & 0 & c_3 & \cdots\cdots & c_{N-1} \\ c_1 & 0 & c_1 & 0 & \cdots\cdots & 0 \\ 0 & c_1 & 0 & c_1 & \cdots\cdots & c_{N-3} \\ c_3 & 0 & c_1 & 0 & \cdots\cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \\ c_{N-1} & 0 & c_{N-3} & 0 & \cdots\cdots & 0 \end{bmatrix} \qquad (2.30)$$

where,

$$c_m = 2 \cot\left(\frac{\pi m}{N}\right)\left[\sin\left(\frac{2\pi\beta_2 m}{N}\right) - \sin\left(\frac{2\pi\beta_1 m}{N}\right)\right] \qquad (2.31)$$

and for N odd :

$$[BP] = \frac{1}{N} \begin{bmatrix} 0 & c_1 & c_2 & c_3 & \cdots\cdots & c_{N-1} \\ c_1 & 0 & c_1 & c_2 & \cdots\cdots & c_{N-2} \\ c_2 & c_1 & 0 & c_1 & \cdots\cdots & c_{N-3} \\ c_3 & c_2 & c_1 & 0 & \cdots\cdots & c_{N-4} \\ \vdots & \vdots & \vdots & \vdots & & \\ c_{N-1} & c_{N-2} & c_{N-3} & c_{N-4} & \cdots\cdots & 0 \end{bmatrix} \qquad (2.32)$$

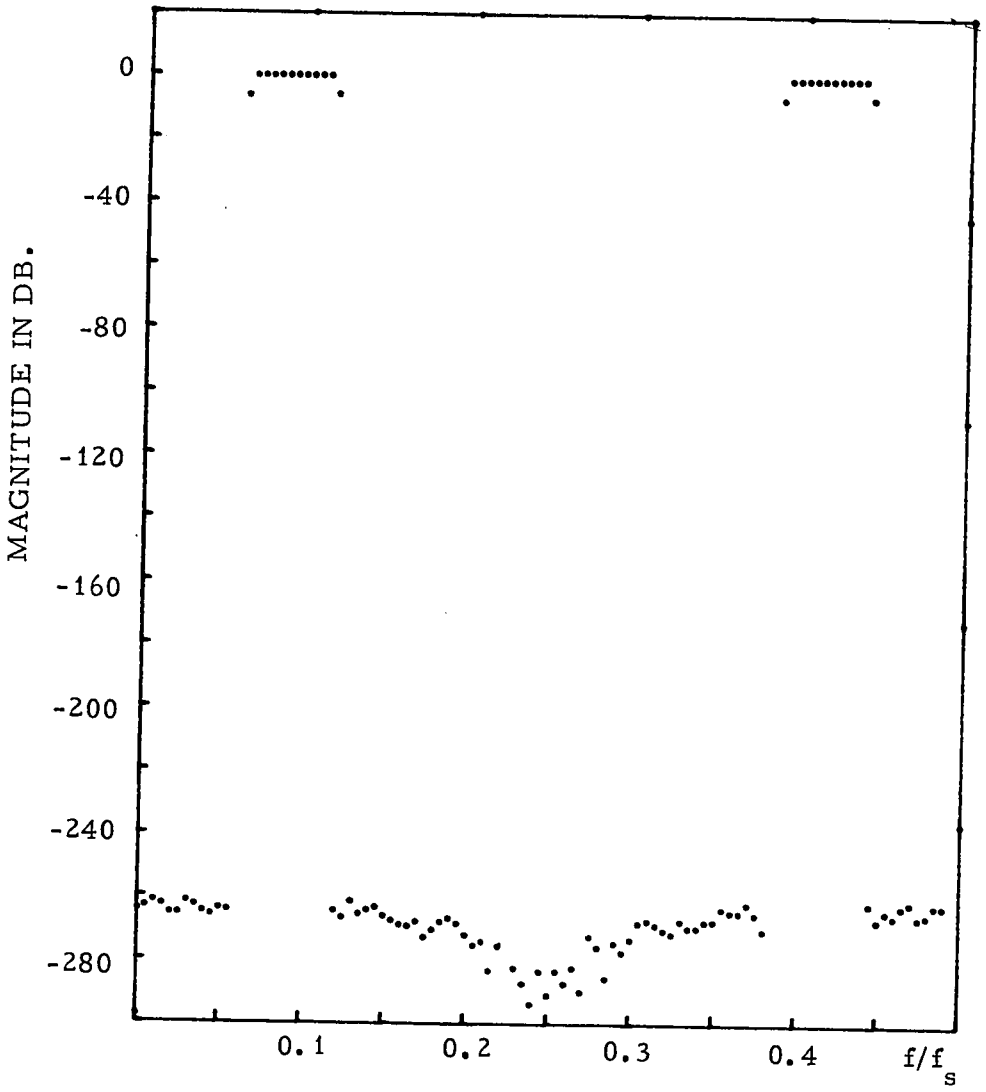


Fig. 2.5. Discrete frequency response of a bandpass filter for N even. N = 200, β_2 = 12, and β_2 = 23.

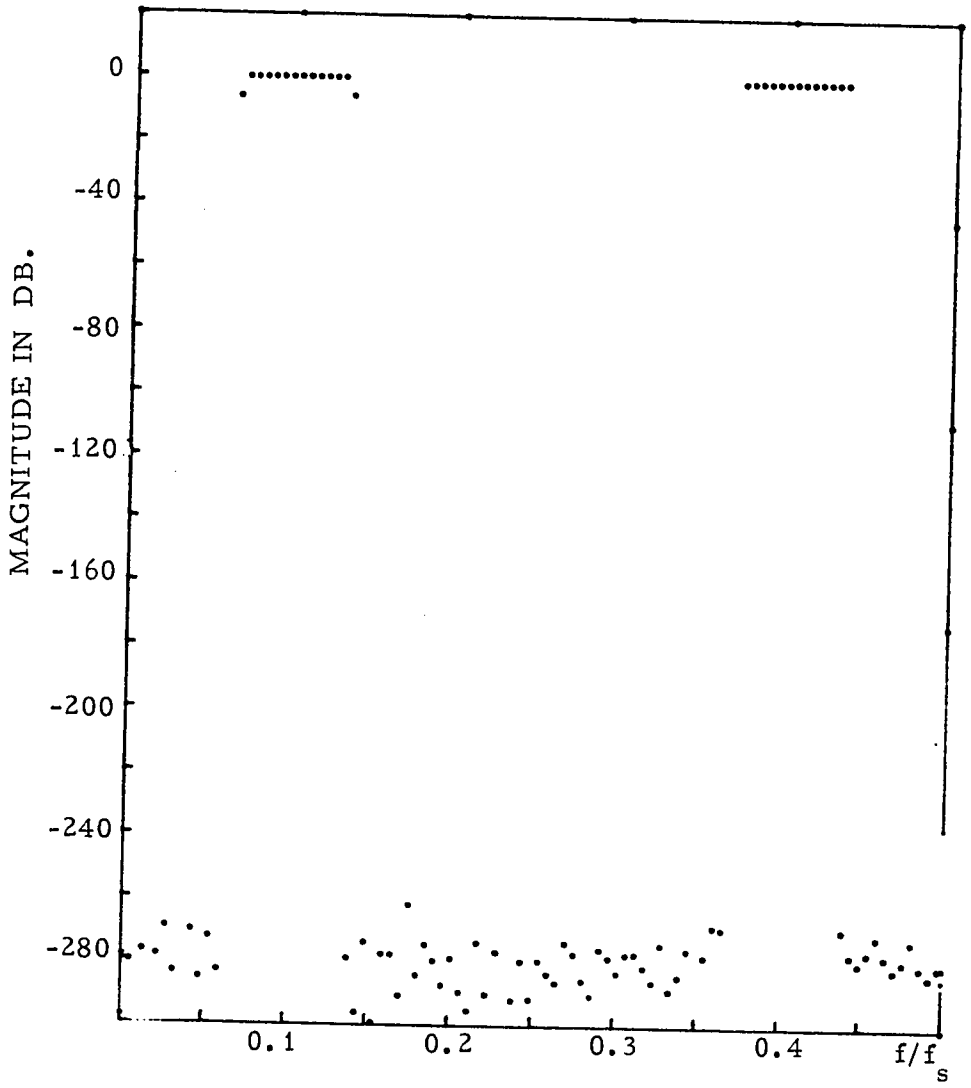


Fig. 2.6. Discrete frequency response of a bandpass filter for N odd. N = 189, $\beta_1 = 12$, and $\beta_2 = 25$.

where

$$c_m = \left[\cot\left(\frac{\pi m}{N}\right) - \frac{(-1)^m}{\sin\left(\frac{\pi m}{N}\right)}\right]\left[\sin\left(\frac{2\pi\beta_2 m}{N}\right) - \sin\left(\frac{2\pi\beta_1 m}{N}\right)\right] \quad (2.33)$$

The bandpass filter matrix has the same properties as the lowpass matrix. Fig 2.5 and fig 2.6 show the discrete frequency response obtained for both N even and N odd respectively.

2.3.3 Highpass Filter Matrix and Transform:

The highpass filter characteristic is obtained by virtually connecting, in parallel, an all pass filter with a lowpass filter both having the same phase response. The output of the highpass filter is formed by subracting the outputs of the lowpass and the allpass filters as shown in fig. 2.7. Since the lowpass filtering operation, in matrix form, has zero phase, the allpass filtering operation should also have a zero phase i.e it is simply a direct path from the input to the output. This is equivalent to multiplying the input, in vector form, by the identity matrix. Thus the operation of the discrete highpass filter is described, in total, by single matrix operation as :

$$[g] = \{[I] - [LP]\}[f] = [HP][f] \quad (2.34)$$

where $[I]$ is the identity matrix of order N, $[LP]$ is the lowpass matrix, and $[HP]$ is the resultant highpass matrix which, for N even, is of the form :

$$[HP] = \frac{1}{N}\begin{bmatrix} N & -b_1 & 0 & -b_3 & \cdots\cdots & -b_{N-1} \\ -b_1 & N & -b_1 & 0 & \cdots\cdots & 0 \\ 0 & -b_1 & N & -b_1 & \cdots\cdots & -b_{N-3} \\ \vdots & \vdots & \vdots & \vdots & & \\ -b_{N-1} & 0 & -b_{N-3} & 0 & & N \end{bmatrix} \quad (2.35)$$

where elements b_m are defined by (2.23) .

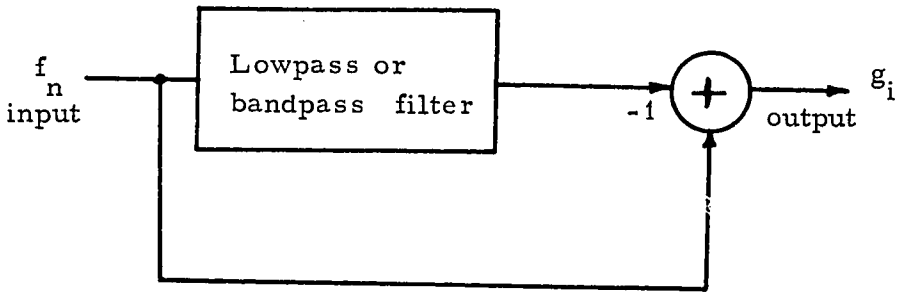


Fig. 2.7. Illustration of the highpass or bandstop filtering operation using a lowpass and bandpass filters.

and for N odd :

$$[HP] = \frac{1}{N}\begin{bmatrix} N & -b_1 & -b_2 & -b_3 \cdots\cdots & -b_{N-1} \\ -b_1 & N & -b_1 & -b_2 \cdots\cdots & -b_{N-2} \\ -b_2 & -b_1 & N & -b_1 \cdots\cdots & -b_{N-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -b_{N-1} & -b_{N-2} & -b_{N-3} & -b_{N-4} & N \end{bmatrix} \qquad (2.36)$$

and the elements b_m are defined by (2.25) .
For both cases of N even and N odd the highpass matrix [HP] has the same properties as the lowpass matrix except that the diagonal elements are unity instead of zero .

Fig. 2.8 shows the discrete frequency response obtained for a typical highpass filter.

2.3.4: Bandstop Filter Matrix and Transform

Bandstop filtering operation can be obtained in a similar way to the one presented in the previous section but replacing the lowpass matrix by the bandpass matrix, i.e. with reference to fig. 2.7, the bandstop filtering operation,expressed in a single matrix, is :

$$[g] = \{ [I] - [BP] \} [f] = [BS] [f] \qquad (2.37)$$

where [BP] is the bandpass matrix and [BS] is the bandstop matrix which has the same properties as the highpass matrix. Fig. 2.9 and fig. 2.10 show the discrete frequency response obtained for a typical bandstop filter and a notch filter respectively.

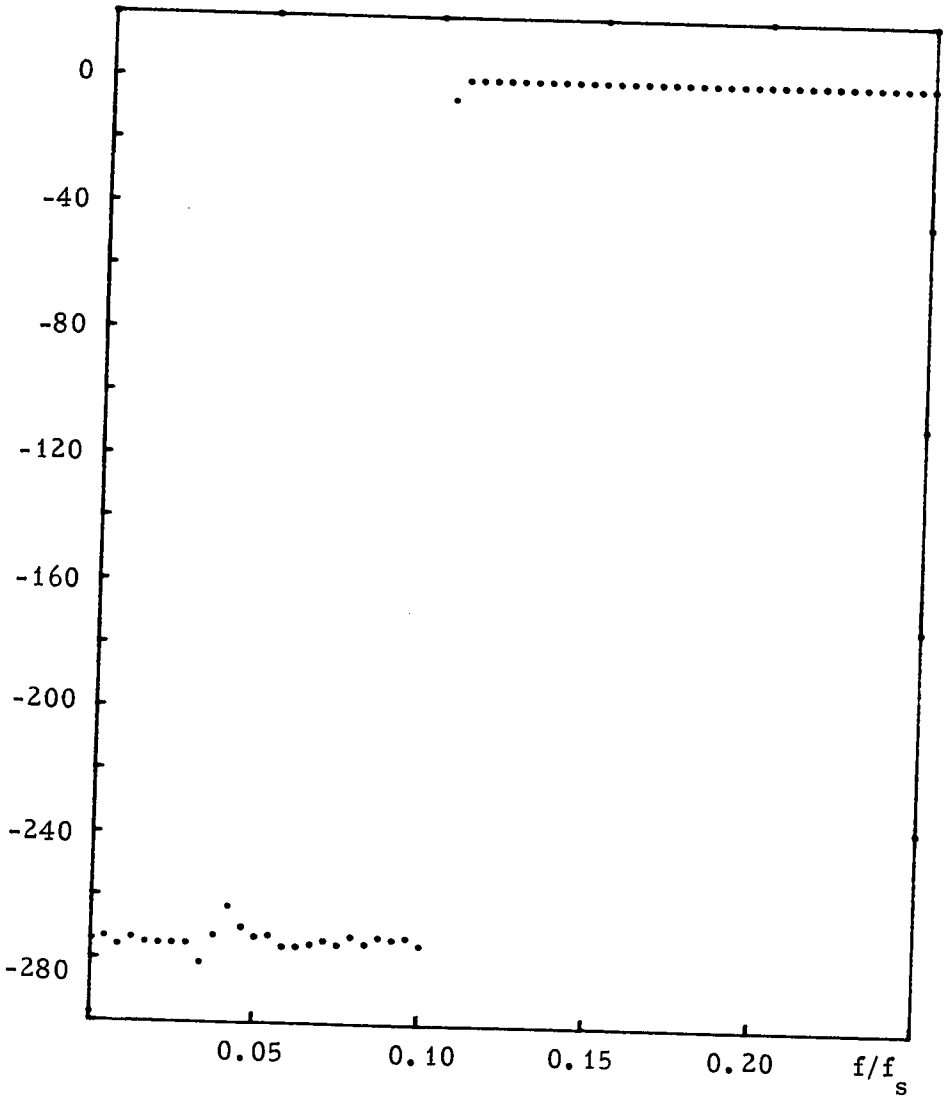


Fig. 2.8. Highpass filter discrete frequency response for N = 240, β = 25.

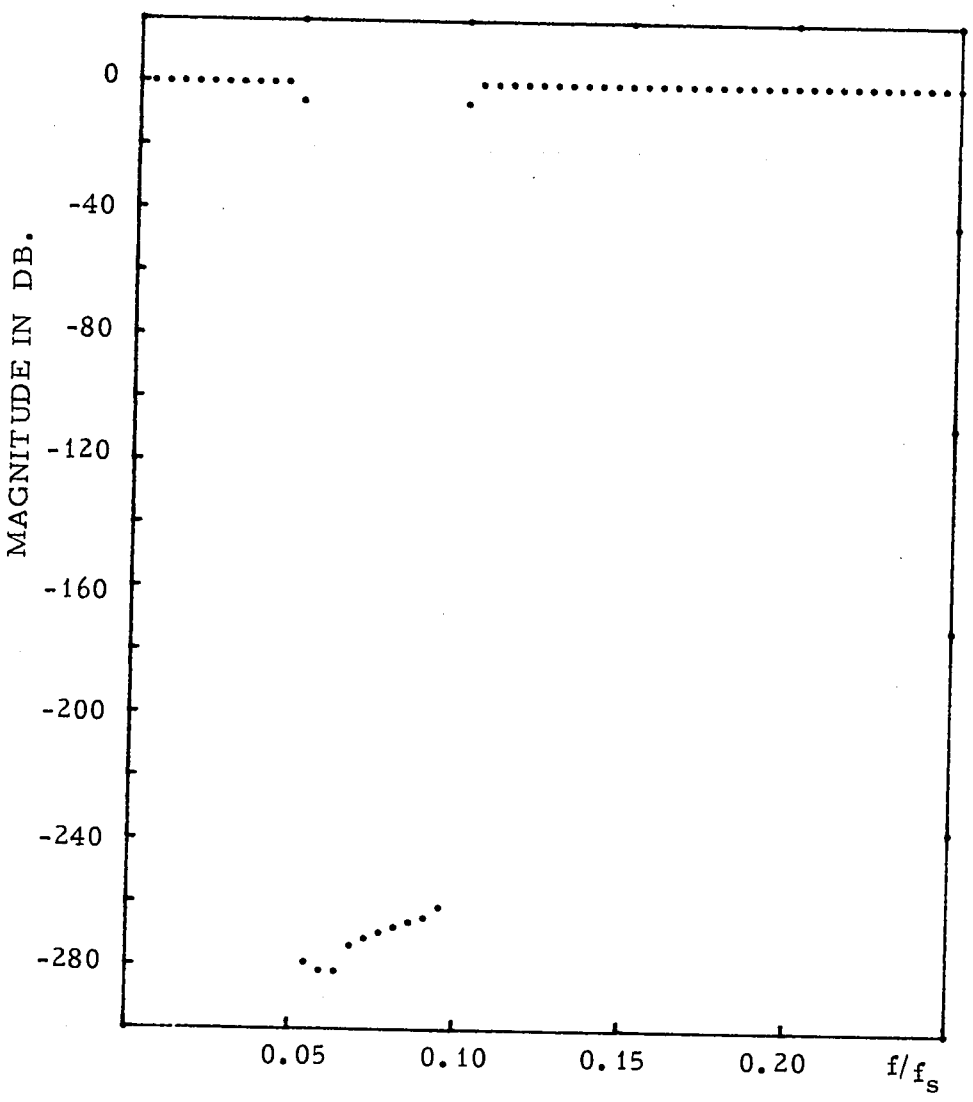


Fig. 2.9 Bandstop filter discrete frequency response, $N = 220$ $\beta_1 = 11$, and $\beta_2 = 22$.

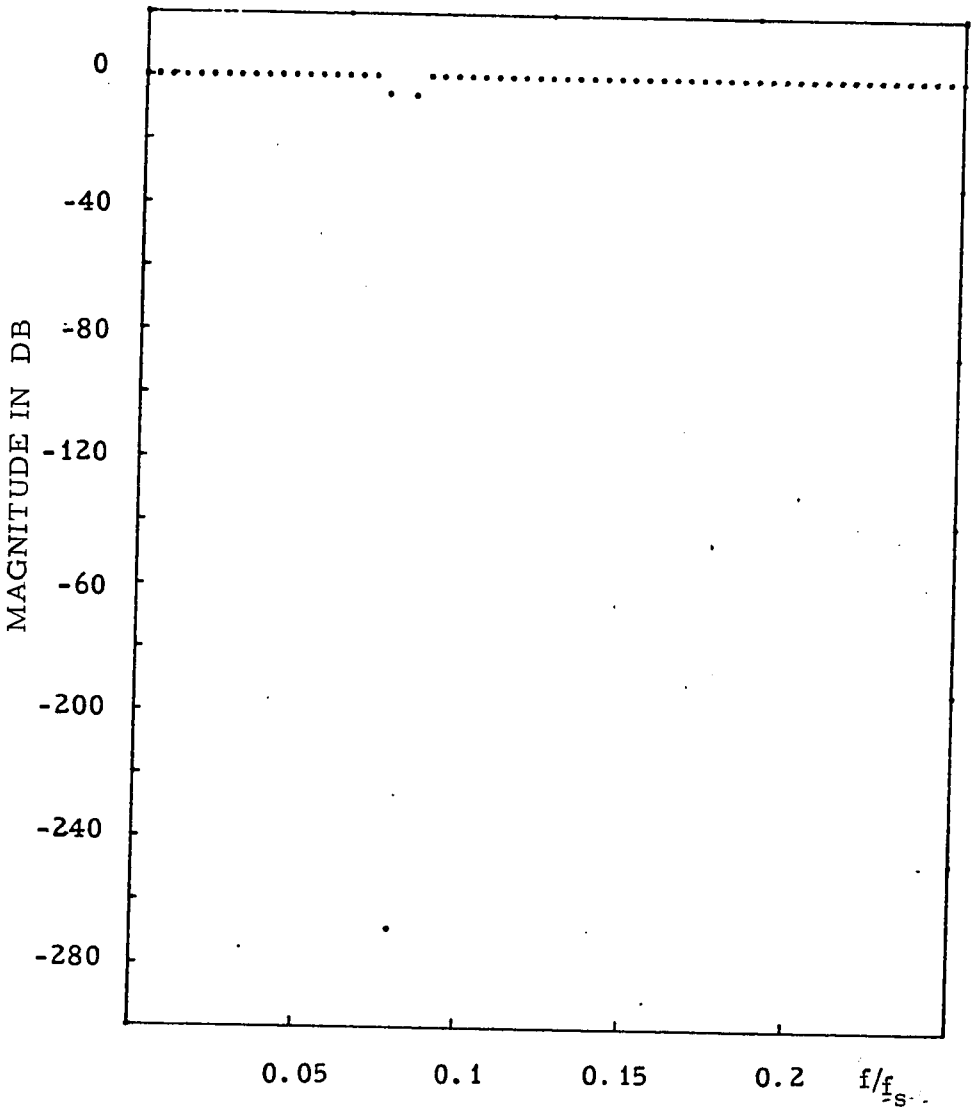


Fig. 2.10. Notch filter frequency response N = 240, $\beta_1 = 20$, $\beta_2 = 18$.

2.4. THE GENERAL FILTER MATRIX AND TRANSFORM

In general the DHT filtering approach, in matrix form, presented in the previous sections can be extended for realizing general discrete filters with arbitrary prespecified frequency response. The discrete filtering operation can be generally described as :

$$[g] = [F][f] \tag{2.38}$$

where $[f]$ and $[g]$ are the input and output (filtered) sequences, in vector form, respectively, $[F]$ is the general filter matrix and is expressed in the form :

$$[F] = \sum_{j=1}^{P} d_j [E]_j \tag{2.39}$$

where d_j is an arbitrary scalar weighting coefficient, $[E]_j$ can be any of the filter matrices presented before , or a combination of more than one of these, and P is the number of filter matrices necessary to realize the prespecified discrete frequency response. The general filter matrix $[F]$ has the same properties as the previous matrices. In the following we shall illustrate the general filter matrix approach and the choice of d_j and $[E]_j$ through several examples.

2.4.1 Multiple Passband Filter Matrix:

Considering the problem of realizing a multiple passband discrete filter. Such filters can, in general, be realized by virtually connecting P bandpass filters (P is the number of fundamental passbands) in parallel .

This operation can be expressed in total by a single matrix operation as:

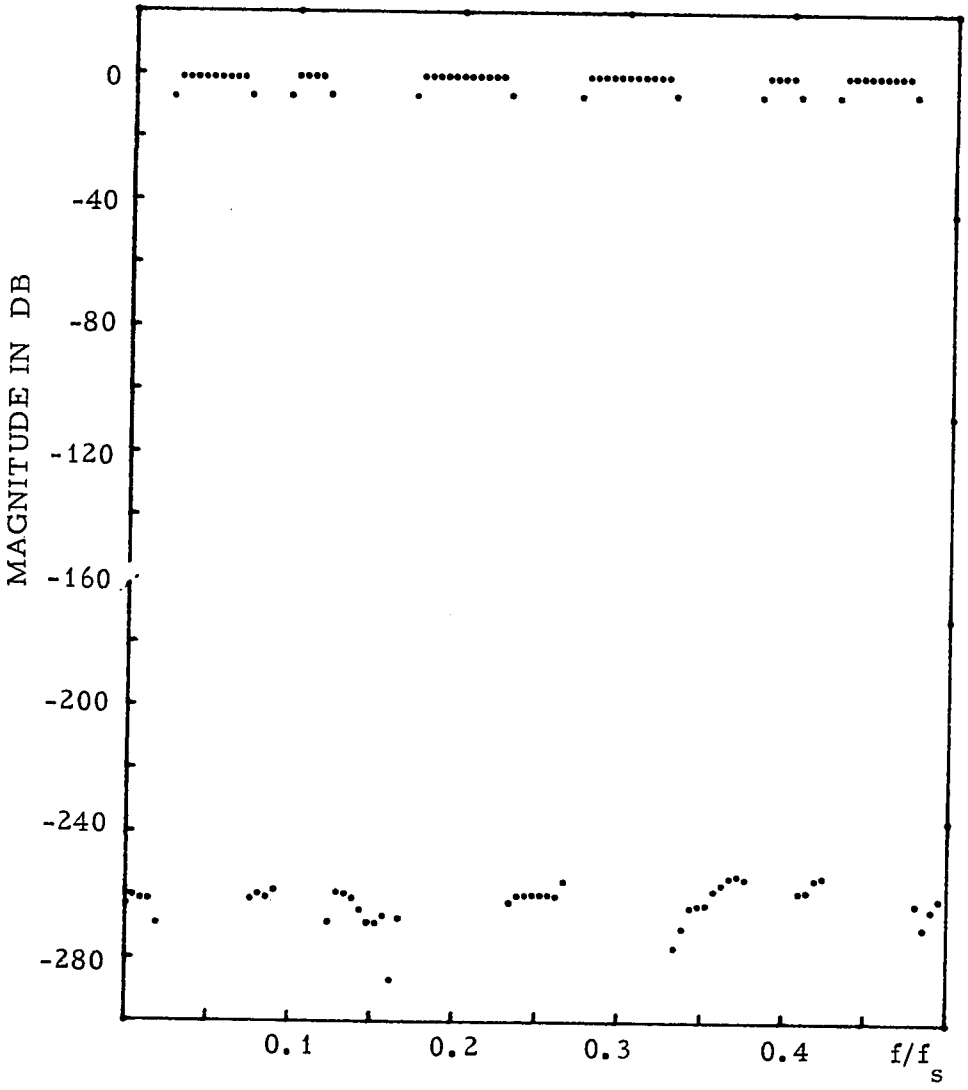


Fig. 2.11. Discrete frequency response of a multiple passband filter, N = 210. Upper band edges are : 15, 25, 48. Lower band edges are : 5, 20, 36.

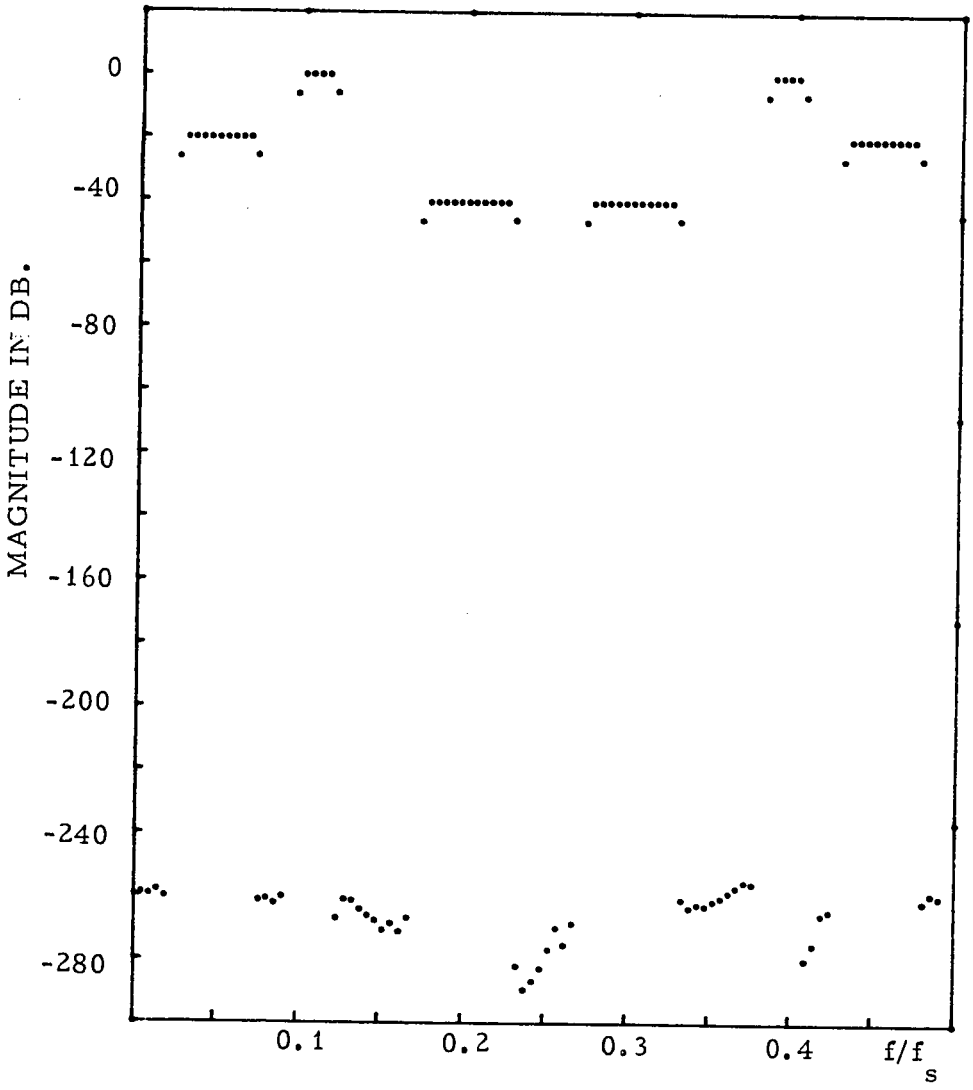


Fig. 2.12. Same as fig. 2.11 except that the passbands are at different levels.
first passband is - 20 db, $d_1 = 0.10$
second passband is 0 db, $d_2 = 1.00$
third passband is -40 db, $d_3 = 0.01$.

$$[g] = \{ \sum_{j=1}^{P} d_j [BP]_j \} [f] = [F] [f] \qquad (2.40)$$

where $[BP]_j$ is the j^{th} bandpass matrix which has upper and lower cutoff frequencies β_j and β'_j; d_j is the arbitrary scaling factor which determine the relative magnitude response of the j^{th} passband. For equal magnitude response for all the passbands d_j is taken equal to unity for all j, $j = 1,2,\ldots P$. If different magnitude responses are required d_j is determined accordingly.

Fig. 2.11 shows the discrete frequency response for a multiple passband filter for N = 210 and P = 3. The passbands have the same relative magnitude i.e. $d_j = 1$ for all j. Fig. 2.12 shows the same as fig. 2.11 but the passbands have different relative magnitude.

2.4.2 Differentiator Matrix :

A linear magnitude filter can be realized by virtually connecting P highpass filters of the type described in sec 2.3.3 in parallel. This is illustrated in fig. 2.13 for N = 24, where 3 highpass filter characteristics are used to construct the discrete linear magnitude characteristic. In general a linear magnitude filter is expressed in single matrix form as :

$$[g] = [L] [f] \qquad (2.41)$$

where $[L]$ is the linear magnitude filter matrix and is given by :

$$[L] = \sum_{j=1}^{P} d_j [HP]_j \qquad (2.42)$$

where

$$d_j = \begin{cases} C & j \text{ odd} \\ 0 & j \text{ even} \end{cases} \qquad (2.43$$

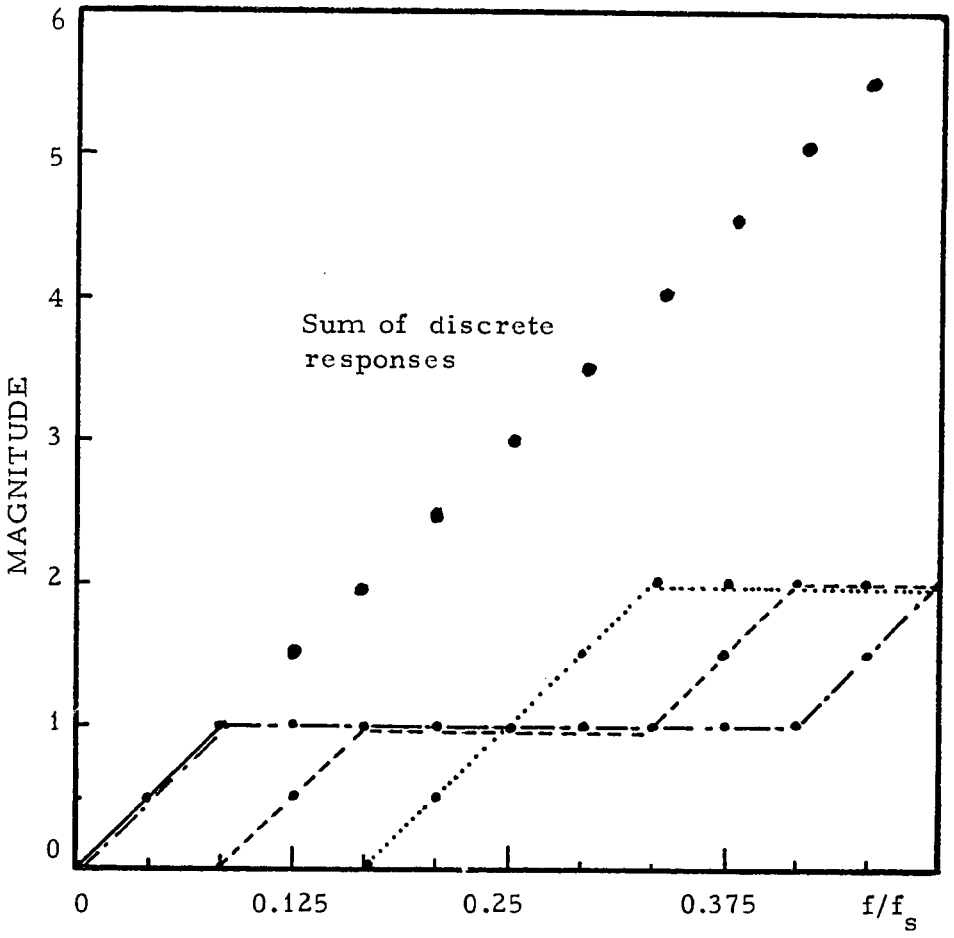


Fig. 2.13. Linear magnitude response for N = 24.

—·—·— highpass filter with $\beta = 1$

– – – – – – " " " $\beta = 3$

..... " " " $\beta = 5$

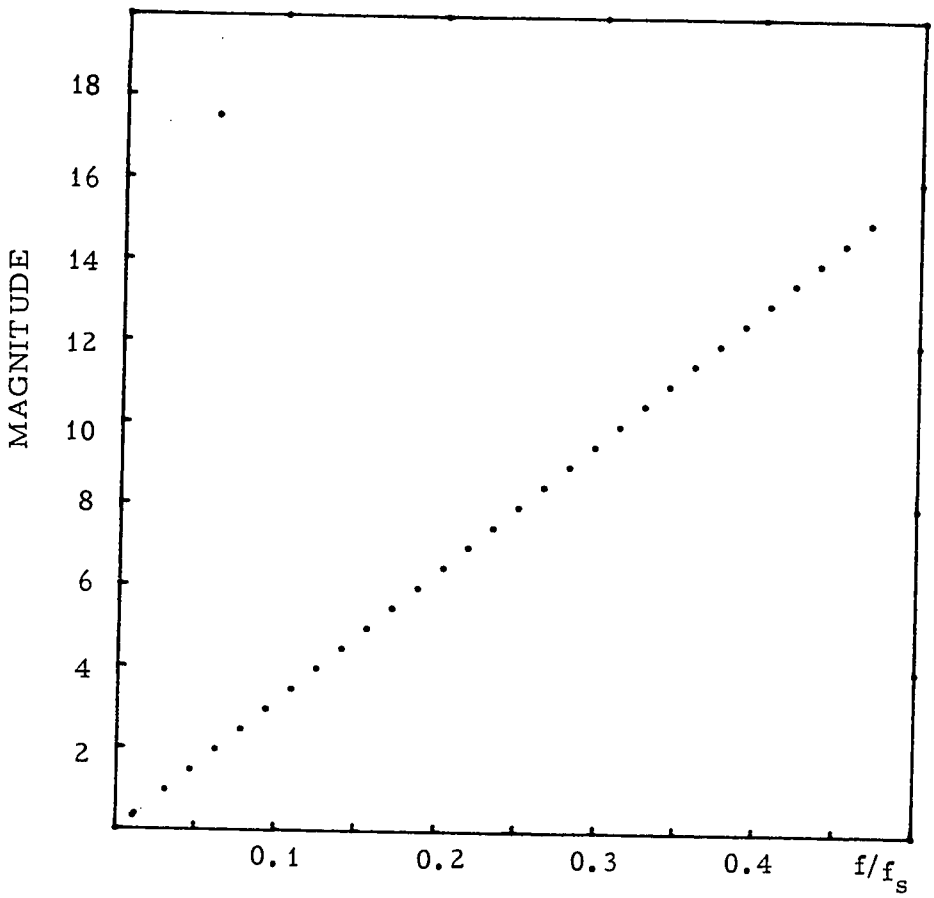


Fig. 2.14. Differentiator discrete frequency response for N = 64.
C = 1.0

$[HP]_j$ is a highpass matrix which has a cutoff frequency $\beta = j$.

For N even, and $\frac{N}{2}$ also even, a wide band linear magnitude response is obtained. In this case the matrix [L] is :

$$[L] = C \sum_{j=1,3,5}^{N/4-1} [HP]_j \; ; \quad j = \beta \tag{2.44}$$

by changing the value of C the slope of the linear magnitude response can be changed to any desired value.

The frequency response of an ideal differentiator is purely imaginary while the frequency response of the linear magnitude filter is purely real. Thus the discrete differentiator operation can be visualized as a cascade of a linear magnitude response filter with a Hilbert transformer. In matrix form :

$$[g] = [H] [L] [f] = [D] [f] \tag{2.45}$$

where [D] is the differentiator matrix, and [H] is the Hilbert matrix. Fig. 2.14 shows the discrete frequency response of a typical differentiator.

2.5. IMPLEMENTATION

Due to the symmetry properties of the Hilbert matrix and derived filter matrices, two basic simple methods of realization are proposed.

2.5.1 Serial Realization:

Fig. 2.15 shows the serial realization of the matrix filtering approach. The coefficients of the first half of the first row of the filter matrix are stored in the coefficient storage memory and are fed serially to the multiplier. The product terms are accumulated (summed) and after the $N^{th.}$ multiplication the first output sample g(o) is formed. Then the coefficients of the first row are cycled one step to the right and the process is repeated to form g(1) and so on. Alternatively the coefficients can be kept stationary and the input sequence is cycled in the reverse direction. We note that the number of coefficients to be stored is in the order of $\frac{N}{4}$ for N even and $\frac{N-1}{2}$ for N odd. The main advantage of such realization is its simplicity as it requires only one multiplier and one accumulator.

2.5.2 Parallel Realization :

Fig. 2.16.a. shows a parallel realization of the DHT filtering approach. The input sequence is fed in parallel. For N even the even and odd numbered output sequences i.e. g(o), g(2) ..., and g(1), g(3),... are obtained separately. Two identical set of multipliers and adders are used, which can also be combined in one set. The dotted line apply only for the cases of filter matrices whose diagonal elements are non-zero (e.g. highpass, bandstop and differentiator). The cycling diagram of the input sequence is shown in Fig. 2.16.b.

While these realizations i.e. the serial and parallel hold both extremes, i.e. purely serial and purely parallel, other realizations can be obtained by employing different degrees of parallel and serial processing.

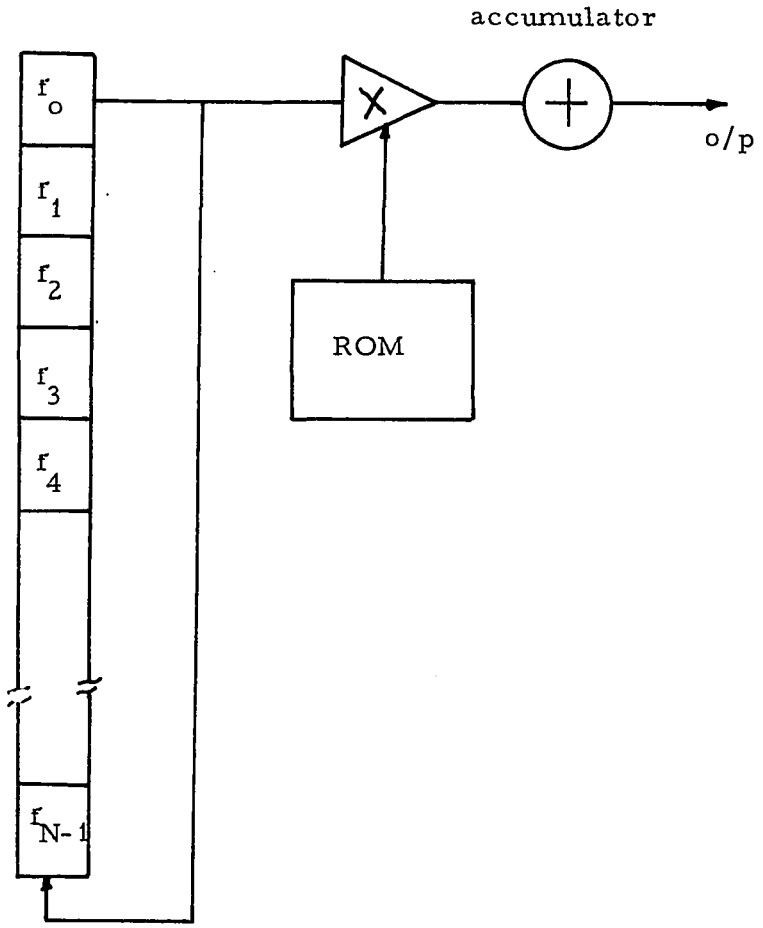


Fig. 2.15. Serial realization.

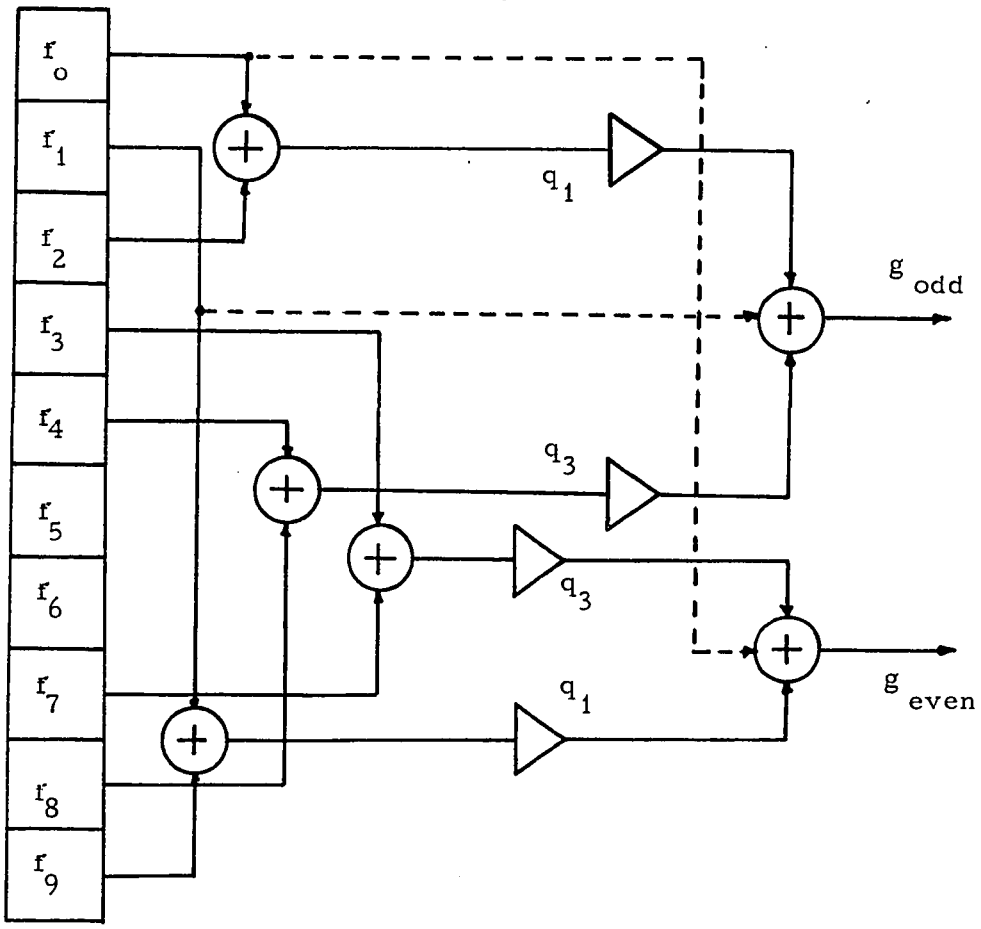


Fig.2.16.a Parallel realization for N = 10. q_1 and q_3 can take values depending on the type of filter, e.g. $q_1 = b_1$ and $q_3 = b_3$ for the lowpass filter. The dotted line applies only to highpass, bandstop and differentiator cases.

Fig.2.16.b. Cycling diagram.

f_o	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_0	f_1
f_4	f_5	f_6	f_7	f_8	f_9	f_0	f_1	f_2	f_3
f_6	f_7	f_8	f_9	f_0	f_1	f_2	f_3	f_4	f_5
f_8	f_9	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7

2.6 COMPARISON WITH THE FFT FILTERING APPROACH

In the FFT filtering approach, fig . 2.17, the input sequence is transformed to the frequency domain using the FFT. In the frequency domain the filtering operation is performed by properly weighting the spectrum of the signal. An inverse FFT completes the process. Thus the process requires 2 steps FFT and weighting in the frequency domain. The DHT filtering operation is done in one step.

The comparison between the FFT and the DHT filtering approaches is presented in terms of three major factors:

1 - Number of computations necessary to perform on the input sequence to obtain the output sequence.

2 - Complexity of the hardware necessary to realize such filters.

3 - The amount of error introduced by both techniques,

The first two factors are discussed in this section and the third one is discussed in chapter 3 after analyzing the different sources of error . It shall be shown that the DHT filtering approach introduces less errors than the FFT approach.

2.6.1 Number of Computations :

When performing any linear digital signal processing algorithm both multiplication and addition operations are required. Since the multiplication operation is usually much more complicated and time consuming than the addition, we shall compare the number of multiplications required for both filtering techniques.

For the FFT filtering technique, the number of complex multiplications required to transform a sequence of length N samples, where

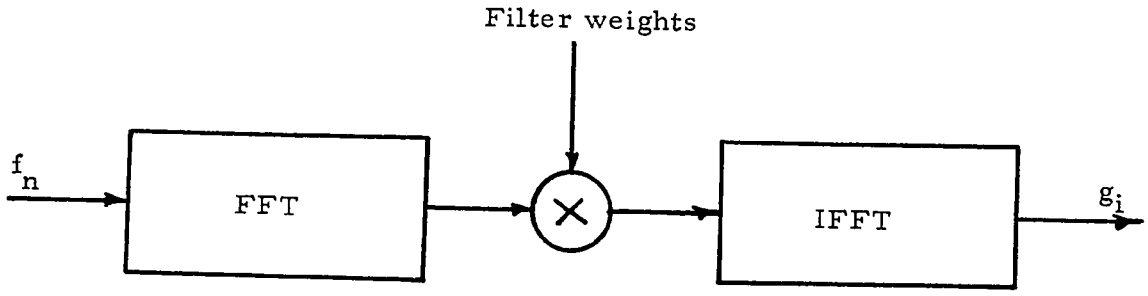


Fig. 2.17. FFT filtering technique.

N is an integer power of two; $N = 2^v$, $v = 1, 2, \ldots$, is $\frac{N}{2} \text{Log}_2 N$.

Since a complex multiplication is equivalent to 4 real multiplications and 2 real additions, thus the total number of real multiplications required for one step FFT (or inverse) is : $2N \text{Log}_2 N$. In the frequency domain N complex multiplications are required i.e. 4 N real multiplications. Hence the total number of real multiplications for the FFT filtering process ; K_{FFT} is :

$$K_{FFT} = 4 N \text{Log}_2 N + 4N = 4N \text{Log}_2 2N .$$

For the DHT filtering approach, N even, the number of real multiplications ; K_{DHT}, is :

$$K_{DHT} = \frac{N^2}{4}$$

Thus the number of real multiplications ratio $\bar{R}$ is :

$$\bar{R} = \frac{K_{FFT}}{K_{DHT}} = \frac{16(v+1)}{N} \quad ; \quad v = \text{Log}_2 N \qquad (2.46)$$

Fig. 2.18 shows a plot of the relation between $\bar{R}$ and N. For $\bar{R}$ greater than one the DHT filtering approach requires less number of multiplications than the FFT filtering approach and visa versa. It is clear that for up to N = 128 samples the DHT approach is better. It should be noted that the FFT is applied when N is a highly composite number e.g. $N = 2^v$, while the DHT approach applies to any value of N.

2.6.2 Complexity of Hardware :

Special hardware realization of the FFT algorithm has been widely investigated [9, 10]. There exist different FFT structures which are commercially available. In general the FFT has a much more complicated and complex structure as compared to the ones shown in fig. 2.15 and fig. 2.16. While the arithmetic units i.e. multipliers

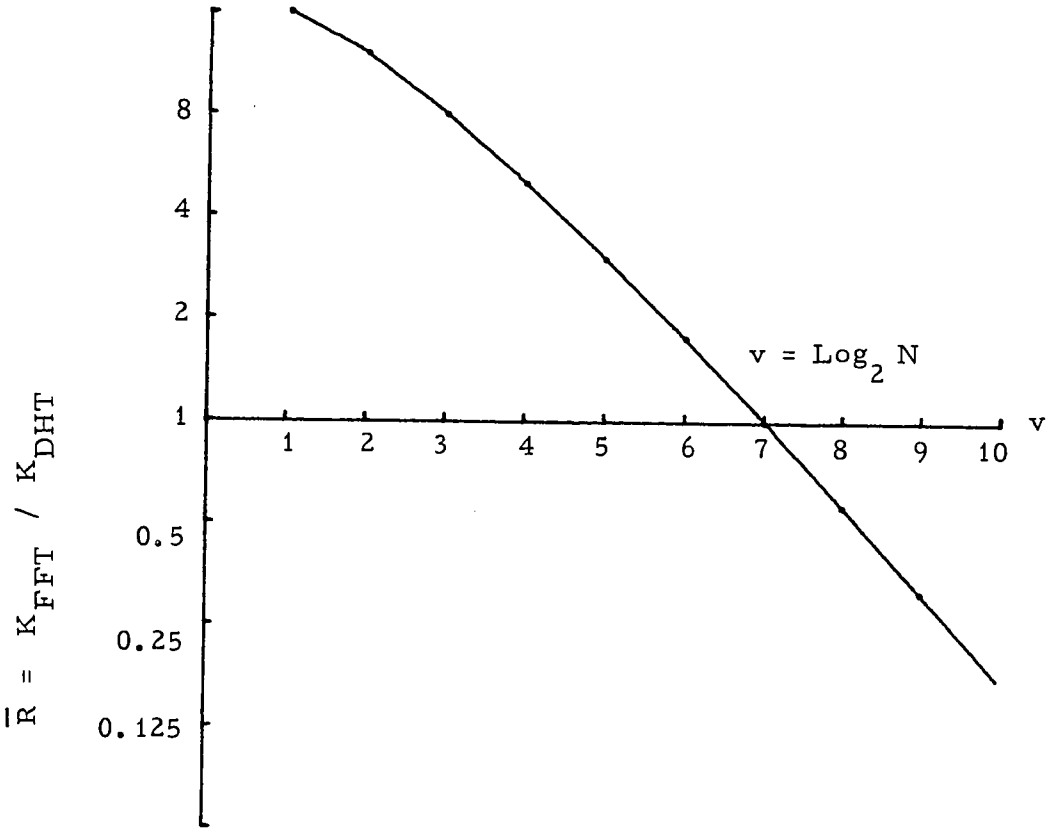


Fig. 2.18. Number of real multiplications ratio $\overline{R}$ for different values of N.

and adders have the same complexity as the FFT(except in case of the FFT complex arithmetic is required), the control unit for the FFT processor is much more complicated since at each stage of the FFT the sequence has to be reordered (permuted). In addition the memory i.e. storage requirements for the FFT is much higher than the DHT approach.

2.7. CONCLUSION

A discrete filtering technique based on the DHT has been presented. Different discrete filters with near ideal discrete frequency response were shown to be realizable. The modulation demodulation process in two or more branches is eliminated and any filtering operation is realized by single matrix operation. Two basic implementation of the DHT filtering technique were presented. It has been shown that the DHT filtering process has simpler implementation than the FFT filtering process and can achieve comparable speeds (up to N = 128) with the FFT, and thus should give viable alternative to the FFT, as the FFT has to be applied twice for the same filtering process.

In order to obtain the ideal filtering characteristics the input signals must have a discrete frequency representation (this also applies to FFT filtering process) i.e in general the input signal must be periodic having a discrete Fourier series representation. Such signals (tones) are widely used in telephone systems, e.g. touch-tone telephone system. Thus these discrete filters can replace the existing complex filters currently used for tone separation and are expected to give better performance and are simpler to implement.

REFERENCES

[1] Schwartz, M., Bennett, W.R., and Stein, S., "Communication Systems and Techniques", McGraw Hill, 1966.

[2] Sakrison, D., "Communication Theory", Wiley, 1968.

[3] Gold, B., Oppenheim, A.V. Rader, C.M., "Theory and Implementation of the Discrete Hilbert Transform", Proc. Symp. Comput. Process Commun. 1969, pp. 235-250.

[4] Cizek, V., "Discrete Hilbert Transform ", Trans. IEEE Audio and Electroacoustics, Vol. AU-18, No. 4, December 1970, pp. 340-343.

[5] Bonzanigo, F., "A Note on Discrete Hilbert Transform", IEEE Trans. on Audio and Electroacoustics, March 1972, pp. 99-100.

[6] Cain, G.D., "Staircase Digital Filters Using Hilbert Transform Elements", I.E.R.E. Proceedings No. 23, April 1972, pp. 101-112.

[7] Hermann, O., "Transversal Filter zur Hilbert-Transformation" Arch. Elek, Ubertrag., 1969, 23, pp. 581-587.

[8] McClellan, J., et al., "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters", IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, No. 6, December 1973, pp. 506-525.

[9] L. Rabiner and B. Gold, "Theory and Applications of Digital Signal Processing", Prentice Hall 1975, chapter 10.

[10] G.D. Bergland, "Fast Fourier Transform Hardware Implementation, an Overview", IEEE Trans. on Audio and Electroacoustics, AU-17, June 1969, pp. 104-108.

CHAPTER THREE

ERROR ANALYSIS FOR THE DHT FILTERING TECHNIQUE

3.1. INTRODUCTION

Digital signal processing algorithms are realized either as special hardware designed to perform a specific algorithm or as programs on a general or special purpose computer. In both cases signal values and system parameters are represented in binary format with finite word length. Due to this finite word length representation of the input signal and system parameters (coefficients), quantization errors are introduced. Even when signals and coefficients are exactly representable with finite word length, additional errors are introduced due to arithmetic operations, e.g. multiplication and addition. Due to these errors, the digital signal processor response will differ from the ideal response.

The amount of error at the output of any digital signal processor is dependent on many factors such as number of bits used to represent the signals and system coefficients, the type of arithmetic used e.g. floating point or fixed point arithmetic, and the method (or technique) by which the digital signal processing algorithm is implemented. For example the discrete filters presented in previous chapter may be implemented by the DFT or the FFT technique or by DHT matrix method. In both cases the end results are theoretically the same. In practice, though the two approaches are theoretically identical, the amount of errors introduced is different assuming the same

number of bits and the same type of arithmetic for both cases .

In this chapter we shall analyze the error introduced due to the finite word length representation of the signal and of the system coefficients as well as the arithmetic errors for the DHT filtering approach. The analysis shall handle both fixed point and floating point arithmetic. Also a comparison is made between the DHT matrix filtering and the FFT filtering in terms of the total error accumulation of each process.

It shall be shown that the DHT method introduces less error than the FFT and thus should give a viable alternative in many filtering applications.

3.2 ERROR ANALYSIS FOR FIXED POINT ARITHMETIC

In this section we shall analyze the errors introduced for the DHT filtering matrix approach, when fixed point arithmetic is used. Expressions for the ratio of the output mean square error (MSE) to mean square signal (MSS) are obtained for different filters.

Let $\hat{f}(n)$ be the ideal representation of the input signal (sequence), i.e. infinite word length representation, and f(n) be the finite word length representation of the signal with t bits . Thus a quantization error $\epsilon(n)$ is introduced and we have :

$$f(n) = \hat{f}(n) + \epsilon(n) \tag{3.1}$$

The signal quantization error sequence $\epsilon(n)$ is random, uniformly distributed between $\pm 2^{-t'}$, with zero mean and a variance

σ_1^2 [1], where t' is the number of bits representing the signal,

$$\sigma_1^2 = \frac{2^{-2t'}}{12} \tag{3.2}$$

Similarly let $\hat{h}_{i,n}$ be the infinite word length representation of the $(i,n)^{th}$ element of the Hilbert matrix or of any of the filter matrices presented in the previous chapter, and $h_{i,n}$ is the finite word length representation of length t bits. Thus we have :

$$h_{i,n} = \hat{h}_{i,n} + \delta_{i,n} \tag{3.3}$$

where $\delta_{i,n}$ is the error introduced due to matrix coefficient quantization. It has the same statistical properties as $\epsilon(n)$, i.e. it has zero mean and variance σ_2^2 [1], where :

$$\sigma_2^2 = \frac{2^{-2t}}{12} \tag{3.4}$$

When two numbers represented in fixed point arithmetic with t bits are multiplied and their product is rounded to t bits, an error is introduced [1] which is again is uniformly distributed with zero mean and variance σ_3^2, where :

$$\sigma_3^2 = \frac{2^{-2t}}{12} \tag{3.5}$$

On the other hand, when two numbers are added an overflow may take place. If this happens, then the sum must be shifted one position to the right and a bit is lost. If the bit is zero then there is no error. If it is a one then there is an error of $\pm 2^{-t}$ depending on whether the number is positive or negative. The variance of this error is [2] :

$$\sigma_4^2 = \frac{2^{-2t}}{2} = 6\,\sigma_3^2 \tag{3.6}$$

In order to find the output MSE to MSS ratio we consider the model shown in Fig. 3.1, where all errors are taken into account ; $\theta_{i,n}$ and $\xi_{i,n}$ are the roundoff error due to multiplication and overflow error due to addition respectively. In the following expressions shall be obtained for the upper and lower bounds on the MSE to MSS ratio. Also, it will be shown that by a slight modification of the actual hardware implementation a drastic reduction in output error is obtained for the DHT filtering approach.

3.2.1 Upper Bound on The Output MSE to MSS Ratio:

To obtain an expression for the upper bound on the MSE to MSS ratio we shall assume that at each addition an overflow takes place. The infinite wordlength output $\hat{g}(i)$, i.e. assuming no errors have been introduced, is expressed as :

$$\hat{g}(i) = \sum_{n=o}^{N-1} \hat{f}(n)\, \hat{h}_{i,n} \qquad (3.7)$$

where $\hat{f}(n)$ and $\hat{h}_{i,n}$ are the infinite wordlength representations of the input signal and of the matrix coefficients respectively.

From fig. 3.1 the actual output, g(i), is :

$$g(i) = \sum_{n=o}^{N-1} \{ f(n)\, h_{i,n} + (\theta_{i,n} + \xi_{i,n}) \} \qquad (3.8)$$

Substituting (3.1) and (3.3) in (3.7) gives :

$$g(i) = \sum_{n=o}^{N-1} \{ (\hat{f}(n) + \epsilon(n))\, (\hat{h}_{i,n} + \delta_{i,n}) + \theta_{i,n} + \xi_{i,n} \} \qquad (3.9)$$

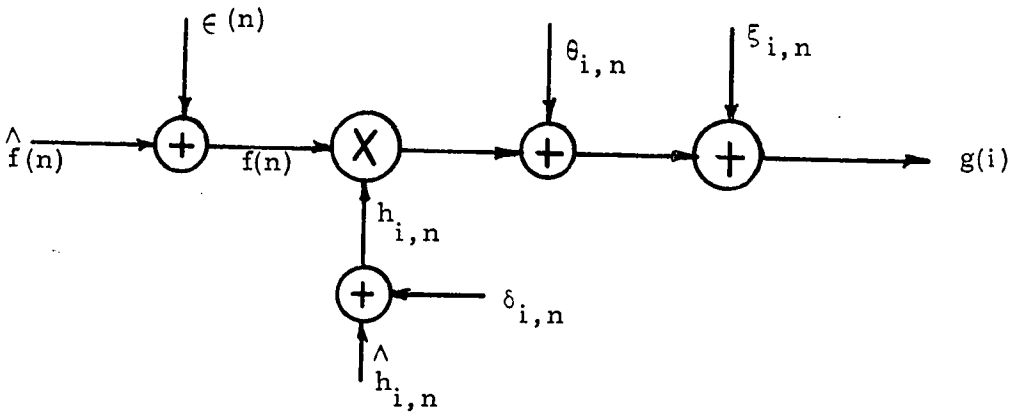


Fig. 3.1 Model for fixed point realization.

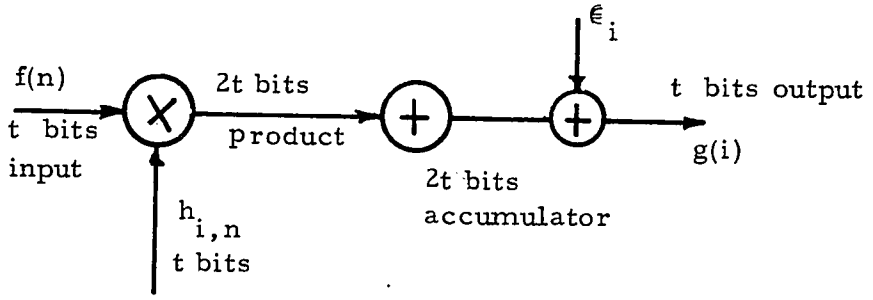


Fig. 3.2. Model for fixed point modified realization.

Rearranging (3.9) and neglecting second order error quantities gives for the actual output:

$$g(i) = \sum_{n=o}^{N-1} \{ \hat{f}(n)\, \hat{h}_{i,n} + \epsilon(n)\, \hat{h}_{i,n} + \hat{f}(n)\, \delta_{i,n} + \theta_{i,n} + \xi_{i,n} \} \qquad (3.10)$$

Thus the output error sequence e(i) is :

$$e(i) = g(i) - \hat{g}(i) = g(i) - \sum_{n=o}^{N-1} \hat{f}(n)\, \hat{h}_{i,n} \qquad (3.11)$$

$$e(i) = \sum_{n=o}^{N-1} \{ \epsilon(n)\, \hat{h}_{i,n} + \hat{f}(n)\, \delta_{i,n} + \theta_{i,n} + \xi_{i,n} \} \qquad (3.12)$$

Relation (3.12) gives an expression for the total output error due to different error sources; the first term is due to the quantization error of the input signal, the second term is due to the matrix coefficients quantization and the last two terms are due to multiplication round off and summation overflow respectively.

In order to find an explicit expression for the output MSE to MSS ratio or in other words the noise to signal ratio at the output, certain statistical assumptions have to be made. It shall be assumed that the error quantities $\epsilon(n)$, $\delta_{i,n}$, $\theta_{i,n}$, and $\xi_{i,n}$ are uncorrelated to each other and to the input signal and matrix coefficients. Also consider the input signal to consist of random uncorrelated samples, a white noise like signal, with zero mean and an auto-correlation function $R_{ff}(\ell)$, as:

$$R_{ff}(\ell) = E\{\hat{f}^2(n)\} = \begin{cases} \sigma_f^2 & \ell = 0 \\ 0 & \ell \neq 0 \end{cases} \qquad (3.13)$$

where $E\{\cdot\}$ denotes the expected value of $\{\cdot\}$, σ_f^2 is the input signal variance.

The expected value of the output square error is :

$$E\{e^2(i)\} = E\{(\sum_{n=o}^{N-1} \epsilon(n) h_{i,n} + f(n)\ \delta_{i,n} + \theta_{i,n} + \xi_{i,n})^2\} \qquad (3.14)$$

Taking into consideration the statistical assumptions made above, (3.14) reduces to :

$$E\{e^2(i)\} = \sum_{n=o}^{N-1} [E\{\epsilon^2(n)\} \hat{h}^2_{i,n} + E\{\hat{f}^2(n)\} E\{\delta^2_{i,n}\} + E\{\theta^2_{i,n}\} + E\{\xi^2_{i,n}\}] \qquad (3.15)$$

A general expression for the expected value of the output mean square, error is given in (3.15). With (3.2), (3.4), (3.5), and (3.6) this reduces, for N even, to :

$$E\{e^2(i)\} = \sigma_1^2(\sum_{n=o}^{N-1} \hat{h}^2_{i,n}) + \frac{N}{2}\ \sigma_f\ \sigma_2^2 + \frac{N}{2}\ \sigma_3^2 + \frac{N}{2}\ \sigma_4^2 \qquad (3.16)$$

and since :

$$\sigma_2^2 = \sigma_3^2 = \frac{\sigma_4^2}{6} \qquad (3.17)$$

Then:

$$E\{e^2(i)\} = \sigma_1^2 (\sum_{n=o}^{N-1} \hat{h}^2_{i,n}) + \frac{N}{2}\ \sigma_f^2\ \sigma_2^2 + \frac{7N}{2}\ \sigma_2^2 \qquad (3.18)$$

Also for N odd we have :

$$E\{e^2(i)\} = \sigma_1^2 (\sum_{n=o}^{N-1} \hat{h}^2_{i,n}) + (N-1)\ \sigma_f^2\ \sigma_2^2 + 7(N-1)\ \sigma_2^2 \qquad (3.19)$$

The basic difference between (3.18) and (3.19) i.e. N even and N odd, is, (as stated in chapter two) that for N even only half the number of multiplications and additions are required as for N odd, because of the zero elements in the matrix. Thus in the subsequent analysis we shall take only even N into consideration.

From (3.18) we can see that the output mean square error has three main components i.e. input signal quantization (first term), coefficient quantization (second term), and arithmetic errors (3^{rd} term). Only the first term is dependent on the type of filtering to be performed e.g. DHT or lowpass filtering ... etc. . The second and third terms are the same for any filtering operation in matrix form.

Our objective is to develop an expression for the output mean square error (MSE) to mean square signal (MSS) ratio.

By definition :

$$MSE = \frac{1}{N} \sum_{i=o}^{N-1} E\{e^2(i)\} \tag{3.20}$$

and

$$MSS = \frac{1}{N} \sum_{i=o}^{N-1} E\{\hat{g}^2(i)\} \tag{3.21}$$

The MSS is obtained using (3.7) and Parseval's theorem [3] in discrete form :

$$\sum_{n=o}^{N-1} |\hat{h}_{i,n}|^2 = \frac{1}{N} \sum_{k=o}^{N-1} |H_k|^2 \tag{3.22}$$

where H_k is the discrete Fourier transform of $\hat{h}_{i,n}$.

From (3.7) :

$$E\{\hat{g}^2(i)\} = E\{(\sum_{n=o}^{N-1} \hat{f}(n)\, \hat{h}_{i,n})^2\} \tag{3.23}$$

Since the input signal samples and the matrix coefficients are uncorrelated (3.23) reduces to :

$$E\{\hat{g}^2(i)\} = \sum_{n=o}^{N-1} E\{\hat{f}^2(n)\}\,\hat{h}^2_{i,n}$$

$$= \sigma_f^2 \sum_{n=o}^{N-1} \hat{h}^2_{i,n}$$

$$= \sigma_f^2 \sum_{k=o}^{N-1} |H_k|^2 \qquad (3.24)$$

Thus the upper value of the output MSS to MSS ratio, R_u , is :

$$R_u = \frac{MSE}{MSS} = \frac{\sum_{i=o}^{N-1} E\{e^2(i)\}}{\sum_{i=o}^{N-1} E\{\hat{g}^2(i)\}} \qquad (3.25)$$

$$R_u = \frac{\sigma_1^2}{\sigma_f^2} + \frac{N^2 \sigma_2^2}{2 \sum_{i=o}^{N-1} \sum_{n=o}^{N-1} \hat{h}^2} + \frac{7N^2}{2 \sum_{i=o}^{N-1} \sum_{n=o}^{N-1} \hat{h}^2_{i,n}} \cdot \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.26)$$

(i) <u>For the DHT :</u>

$$\hat{h}_{i,n} = a_{i,n}$$

$$\sum_{n=o}^{N-1} a^2_{i,n} = \frac{1}{N} \sum_{k=o}^{N-1} |H_k|^2 = \frac{N-2}{N}$$

$$\simeq 1 \quad \text{for } N \text{ large} \qquad (3.27)$$

where $a_{i,n}$ are the coefficients of the Hilbert matrix.
Thus for the DHT, R_u is :

$$R_u = \frac{\sigma_1^2}{\sigma_f^2} + \frac{N}{2}\sigma_2^2 + \frac{7N}{2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.28)$$

(ii) For discrete lowpass filters :

$$\hat{h}_{i,n} = b_{i,n}$$

$$\sum_{n=o}^{N-1} b_{i,n}^2 = \frac{1}{N} \sum_{k=o}^{N-1} |H_k|^2 = \frac{4\beta + 2}{N} \qquad (3.29)$$

where β is the filter cutoff frequency, $b_{i,n}$ are the coefficients of the lowpass matrix and R_u is obtained by substituting (3.29) in (3.26) .

(iii) For discrete bandpass filters :

$$\hat{h}_{i,n} = c_{i,n}$$

$$\sum_{n=o}^{N-1} c_{i,n}^2 = \frac{1}{N} \sum_{k=o}^{N-1} |H_k|^2$$

$$= \frac{4(\beta_2 - \beta_1) + 2}{N} \qquad (3.30)$$

where $c_{i,n}$ are the lowpass matrix coefficients, β_2 and β_1 are the upper and lower cutoff frequencies. R_u is obtained by substitution of (3.30) in (3.26) . Similar expressions can be obtained for the different filter matrices following the same procedure.

In fact the upper bound on the output MSE to MSS ratio presented here is much higher than one would expect the actual error to be, since we can determine in advance the number of overflows that might take place once the numerical value of N is known. This shall be further discussed later on.

3.2.2. Lower Bound on The Output MSE to MSS Ratio:

In order to find the lower bound on the output MSE to MSS ratio, we assume that no overflow takes place in the accumulator

(summer) and hence no summing errors occur, i.e. $\xi_{i,n} = 0$ for all i and n . Then the lower bound MSE to MSS ratio; R_l, is :

$$R_l = \frac{\sigma_1^2}{\sigma_f^2} + \frac{N^2 \sigma_2^2}{2 \sum_{i=o}^{N-1} \sum_{n=o}^{N-1} \hat{h}_{i,n}^2} + \frac{N^2}{2 \sum_{i=o}^{N-1} \sum_{n=o}^{N-1} \hat{h}_{i,n}} \cdot \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.31)$$

For the DHT (3.31) reduces to :

$$R_l = \frac{\sigma_1^2}{\sigma_f^2} + \frac{N}{2} \sigma_2^2 + \frac{N}{2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.32)$$

3.2.3: Actual Output MSE to MSS Ratio:

As we have mentioned earlier, the upper bound on the output MSE to MSS ratio is much higher than one would expect it to be. The number of overflows that might take place, and consequently the number of shifts necessary, can be estimated once the numerical value of N, i.e. number of samples, is known . This can be done using the upper bound on the output of a circular convolution [4] i.e.

$$g(i) = \sum_{n=o}^{N-1} f(n)\, h_{i,n} \qquad (3.33)$$

and since in fixed point arithmetic the magnitude of any number is less than unity, then :

$$|g(i)| < \sum_{n=o}^{N-1} |h_{i,n}| \qquad (3.34)$$

The output g(i) cannot exceed the value of the right hand side summation in (3.34), and hence this summation gives the maximum number

of overflows i.e. necessary shifts. Table 3.1 gives the maximum number of ncessary shifts for several values of N, for the DHT. We should note that the shifting (due to overflow) takes place at the output, so both output signal and errors are scaled down and thus, has no effect on the MSE to MSS ratio.

3.2.4. Improved MSE to MSS ratio and Comments :

Examining closely the different terms of (3.31) i.e. the effect of different sources of errors to see how the MSE to MSS can be reduced or the signal to noise ratio can be improved :

(i) Input signal quantization errors:

Actually very little can be done to such errors, since the only way to reduce its effect is simply to add more bits to the representation of the signal and that would reduce σ_1^2. Such a solution will add to the complexity of the overall system, which the filter is just one part of.

Another approach is to increase the input signal variance σ_f^2, but there is a limit to that since we are working with fixed point arithmetic and signal samples magnitude should be less than 1.0, also increasing the input signal variance would add to the possibility of overflow in the accumulator.

(ii) Coefficient quantization errors:

For these errors the only way to reduce their effect is increasing the number of bits representing the coefficient and consequently reducing σ_2^2. Again that would be a general system consideration which would also add to the complexity of the whole system as well as the filter itself.

(iii) Arithmetic errors:

As in (i) and (ii) the arithmetic errors can be reduced by increasing the accuracy of addition and multiplication. Since such solutions are quite

obvious to any designer we shall assume that the number of bits representing the signal and different filter parameters has been chosen and fixed optimally with respect to accuracy, complexity of hardware, cost, and other system considerations. Thus the problem now is reduced to finding the best technique to perform the digital signal processing algorithm which shall introduce the least amount of arithmetic errors.

In fig. 3.1 we have assumed that a conventional t bit multiplier was used and the product of multiplication is rounded to t bits directly after the multiplier. Usually the rounding mechanism is incorporated inside the multiplier itself. The rounded products of the multiplication are fed to the accumulator (t bits) where it is added.

The arithmetic errors at the output can be reduced drastically by slightly modifying some details in the actual hardware. First we shall use a multiplier where the rounding mechanism has been eliminated i.e. when two t bits numbers are multiplied the result is (2 t) bits. Also instead of the t bit accumulator, a (2t) bit accumulator is used. The output of the filter is now (2t) bits numbers and a simple rounding mechanism is employed at the output to round it to the original t bits if required.

Now let us see how these slight modifications shall reduce the errors at the output. First, since there is no rounding after the multiplier roundoff errors are not introduced and hence in (3.8) to (3.32) $\theta_{i,n} = 0$ for all i and n. Thus the roundoff error due to multiplication has actually been eliminated. In the accumulator (summer), even if an overflow takes place and the product has to be shifted and one bit is lost an error is introduced , this error is very small since we are using (2t) bits accumulator instead of t bit. The variance of the error in this case is $\sigma_5^2 = \frac{2^{-4t}}{2}$ instead of $\sigma_4^2 = \frac{2^{-2t}}{2}$ and consequently can be ignored. Thus we have eliminated the overflow errors.

At the output of the filter, each output sample is now represented by $2t$ bits. If the overall system requirements would allow such increase in the number of bits representing output signal samples, then there are no arithmetic errors introduced by the proposed structure. In most cases this is not allowed and the output samples have to be rounded to t bits. Now an error is introduced which has a variance of $\sigma_6^2 = \frac{2^{-2t}}{12}$. Fig. 3.2 shows the modified statistical model where the rounding mechanism is placed after the accumulator and a $2t$ bit accumulator is used.

In order to see the improvement in the signal to noise ratio, i.e. reduction in MSE to MSS ratio, let us examine only the terms due to arithmetic errors. The output MSE to MSS ratio R' (due to arithmetic errors only) is :

$$R' = \frac{MSE}{MSS} = \frac{\gamma N^2}{2 \sum_{i=o}^{N-1} \sum_{i=o}^{N-1} \hat{h}_{i,n}^2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.35)$$

where $\gamma = 1$ for the lower bound and $\gamma = 7$ for the upper bound, and for the DHT was :

$$R' = \frac{\gamma N}{2} \frac{\sigma_2^2}{\sigma_f^2} = \frac{\gamma N}{24} \frac{2^{-2t}}{\sigma_f^2} \qquad (3.36)$$

After the modification shown in fig. 3.2., it is :

$$R' = \frac{MSE}{MSS} = \frac{N}{\sum_{i=o}^{N-1} \sum_{n=o}^{N-1} \hat{h}_{i,n}^2} \frac{\sigma_6^2}{\sigma_f^2} \qquad (3.37)$$

and for the DHT it is :

$$R' = \frac{\sigma_6^2}{\sigma_f^2} = \frac{2^{-2t}}{12 \sigma_f^2} \qquad (3.38)$$

As we can see the dependence on N has been eliminated or in other words the proposed modification has led to an improvement in the signal to noise ratio of approximately N times. We note that (3.37) gives the actual MSE to MSS ratio and not a bound.

It should be noted that the proposed modification, i.e. no rounding after multiplication and 2 t bits addition, can not be used where subsequent multiplication is required since that would mean that the number of bits would increase indefinitely . Algorithms like the FFT cannot make use of such an ideasince the final output is formed by successive stages and at each stage there is multiplication and addition.

3.3 ERROR ANALYSIS FOR FLOATING POINT ARITHMETIC

In this part the arithmetic errors for floating point arithmetic are derived. When two numbers represented in floating point arithmetic are added or multiplied and the results are rounded an error is introduced i.e :

$$f\ell\,(x + y) = (x + y)(1 + \epsilon)$$
$$f\ell\,(x \cdot y) = xy\,(1 + \epsilon) \qquad (3.39)$$

where ϵ is a random error independent of x and y and is bounded by $-2^{-t} \le \epsilon \le 2^{-t}$. ϵ is uniformly distributed with variance $\sigma_1'^2 = \frac{2^{-2t}}{3}$ [3]. $f\ell(\cdot)$ means that $(\cdot)$ is carried out in floating point arithmetic.

In order to determine the output MSE to MSS a statistical model similar to the one presented by Liu and Kaneko [5] for direct form realization of digital filters is used. Only the case of N even is considered though the results can simply be extended for N odd.

The actual output g(i) is expressed as :

$$g(i) = f\ell \left(\sum_{n=o}^{N-1} f(n)\ h_{i,n} \right) \tag{3.40}$$

Let $\rho_{i,n}$ be the error resulting from multiplying f(n) by $h_{i,n}$ and $\gamma_{i,j}$ is the error due to the j^{th} addition when computing the i^{th} output sample in the accumulator.

The errors $|\rho_{i,n}| < 2^{-2t}$, $|\gamma_{i,j}| < 2^{-2t}$ and each has a variance $\sigma_1^{'2} = \frac{2^{-2t}}{3}$, where t is the number of mantissa bits.

Fig. 3.3 shows the flow graph of error propagation and error quantities introduced. The actual output ; from fig. 3.3, is :

$$g(i) = \sum_{n=o}^{N-1} f(n)\ h_{i,n}\ \psi_{i,n} \tag{3.41}$$

where:

$$\psi_{i,n} = \begin{cases} (1+\rho_{i,0}) \prod_{j=3,5,7}^{N-1} (1+\gamma_{i,j}) & n = 1,\ i \text{ even} \\ (1+\rho_{i,n}) \prod_{j=n}^{N-1} (1+\gamma_{i,j}) & n = 3,5,7,\ldots,(N-1),\ i \text{ even} \\ (1+\rho_{i,0}) \prod_{j=2,4,6}^{N-2} (1+\gamma_{i,j}) & n = 0,\ i \text{ odd} \\ (1+\rho_{i,n}) \prod_{j=n}^{N-2} (1+\gamma_{i,j}) & n = 2,4,..,(N-2),\ i \text{ odd} \end{cases} \tag{3.42}$$

Thus the output error sequence e(i) is :

$$e(i) = \sum_{n=o}^{N-1} f(n)\ h_{i,n}\ (\psi_{i,n} - 1) \tag{3.43}$$

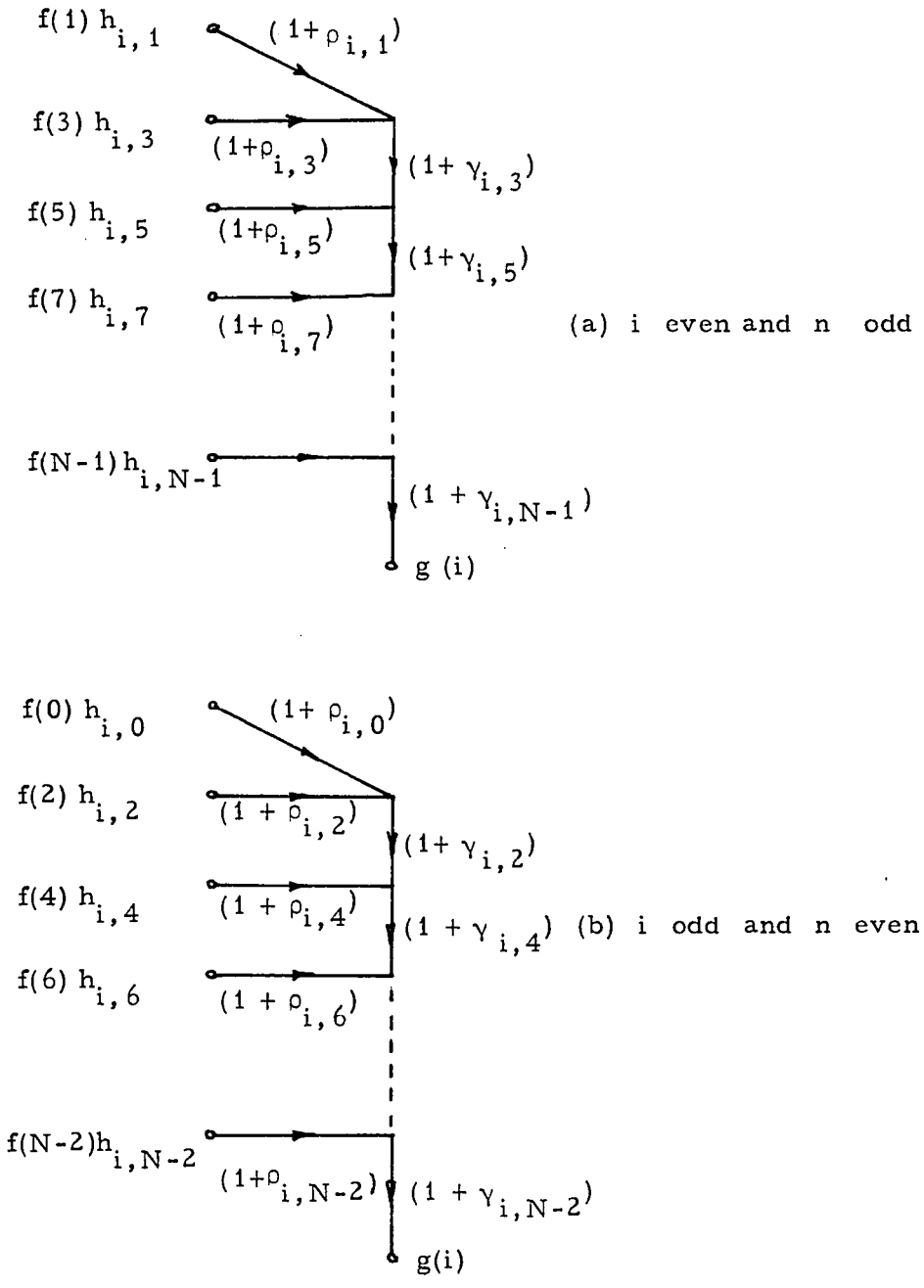


Fig. 3.3. Model for floating point arithmetic errors.

The expected value of the output square error is :

$$E\{e^2(i)\} = E\{(\sum_{n=o}^{N-1} f(n)\, h_{i,n}(\psi_{i,n}-1))^2\} \qquad (3.44)$$

Assuming a white random input signal with variance σ_f^2. Also the errors are uncorrelated with one another and with the input signal and the matrix coefficients. Thus (3.44) reduces to :

$$E\{e^2(i)\} = \sigma_f^2 \sum_{n=o}^{N-1} h_{i,n}^2\, E\{(\psi_{i,n}-1)^2\} \qquad (3.45)$$

The expected value of the output square signal is :

$$E\{g^2(i)\} = E\{(\sum_{n=o}^{N-1} f(n)\, h_{i,n})^2\}$$

$$= \sigma_f^2 \sum_{n=o}^{N-1} h_{i,n}^2 \qquad (3.46)$$

The output mean square error MSE to mean square signal MSS ratio R is :

$$R = \frac{MSE}{MSS} = \frac{\sum_{i=o}^{N-1} E\{e^2(i)\}}{\sum_{i=o}^{N-1} E\{g^2(i)\}}$$

$$= \frac{\sum_{i=o}^{N-1}\sum_{n=o}^{N-1} h_{i,n}^2\, E\{(\psi_{i,n}-1)^2\}}{\sum_{i=o}^{N-1}\sum_{n=o}^{N-1} h_{i,n}^2} \qquad (3.47)$$

As can be seen the output MSE to MSS ratio is no longer a function of the input signal variance. The summation in the denominator of (3.47) is evaluated using Parseval's theorem as in fixed point arithmetic . The numerator is dependent on the statistical properties

of $\psi_{i,n}$ which is obtained (appendix 3.1) from relation (3.42). For the DHT case, the output MSE to MSS, R, reduces to :

$$R \simeq \left(\frac{N}{8} + \frac{1}{2}\right) \sigma_1'^2 \qquad (3.48)$$

where $\sigma_1'^2 = \frac{2^{-2t}}{3}$, t is the number of bits in the mantissa. Similar expressions for the MSE to MSS ratio may be obtained for different filter matrices.

3.3.1 : Improved MSE to MSS ratio for floating point arithmetic:

The same idea presented in sec. 3.2.4 to reduce the output MSE to MSS for fixed point arithmetic can also be employed here for floating point arithmetic i.e. no rounding after the multiplier and a 2 t bits mantissa floating point accumulator is used instead of t bits followed by the rounding mechanism which rounds the mantissa to the original t bits.

In this case the errors $\rho_{i,n}$ and $\gamma_{i,n}$ are very small and can be neglected i.e the variance is $\frac{2^{-4t}}{3}$ instead of $\frac{2^{-2t}}{3}$.

The statistical model for this case is shown in Fig. 3.4. The output MSE to MSS is :

$$R = \frac{MSE}{MSS} = \sigma_1'^2 = \frac{2^{-2t}}{3} \qquad (3.49)$$

which is no longer dependent on N.

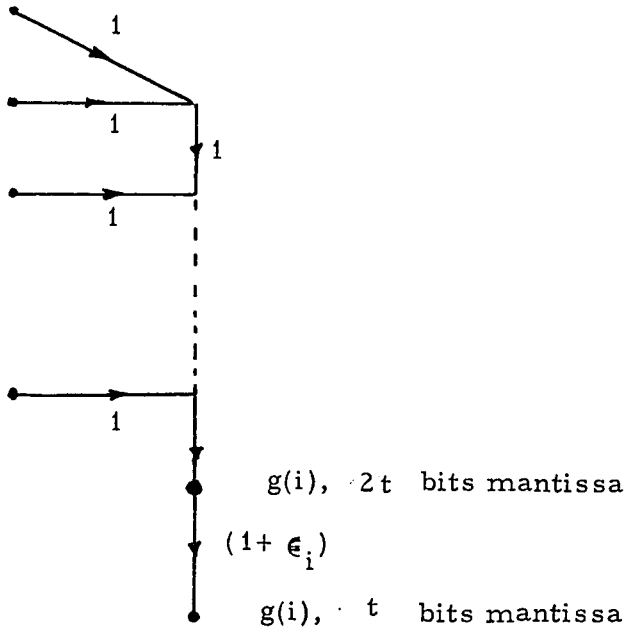


Fig. 3.4. Model of modified realization for floating point arithmetic errors.

3.4 COMPARISON WITH THE FFT FILTERING IN TERMS OF ARITHMETIC ERRORS

In chapter two a comparison between the DHT matrix filtering technique and the FFT filtering technique in terms of number of arithmetic operations and complexity of hardware has been presented. In this section we shall compare both techniques in terms of arithmetic errors introduced due to finite wordlength. It will be shown that the DHT matrix filtering technique introduces significantly less error than the FFT filtering technique for any value of N i.e. number of samples, for both fixed point and floating point arithmetic realization.

3.4.1 : Arithmetic errors in Fixed Point FFT [1,2,7] :

The output arithmetic errors resulting from fixed point FFT algorithms has been thoroughly investigated. For one step †FFT the output mean square error MSE to mean square signal MSS i.e. output noise to signal ratio is [7]:

$$R_{FFT} = \frac{MSE}{MSS} = \frac{5N}{3} \frac{2^{-2t}}{\sigma_f^2}$$

$$= 20N \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.50)$$

where $\sigma_2^2 = \frac{2^{-2t}}{12}$, σ_f^2 the input signal variance, t is the number of bits, and N is the number of samples. In deriving (3.50) the same statistical assumptions were made as in sec. 3.2. The expression (3.50) is the best reported result in the literature and is based on stage by stage scaling to prevent overflow. In case where the scaling

† by a one step FFT we mean the transform operation which is normally carried out in v stages for $N = 2^v$.

is done on the input sequence prior to the FFT process the output MSE to MSS would be proportional to N^2 rather than N [7]. We shall ignore this and restrict our comparison to the best possible case for the FFT i.e. relation (3.50).

We should note that for the FFT filtering technique, the FFT has to be performed twice (forward and inverse), so the actual MSE to MSS at the output of the FFT filter would be much higher than that shown in (3.50). It shall be shown that the output MSE to MSS resulting from the DHT filtering technique is much less than one step FFT.

3.4.2 Comparison for fixed point arithmetic:

In the error analysis of the DHT filtering matrix approach it has been shown that the output MSE to MSS ratio i.e output noise to signal ratio (arithmetic errors only) is :

(i) For the realization of fig. 3.1.

The upper bound is :

$$R_u = \frac{7N^2}{2\sum_{i=o}^{N-1}\sum_{n=o}^{N-1} h_{i,n}^2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.51)$$

which for the DHT case after substitution for $h_{i,n}$ reduces to :

$$R_u = \frac{7N}{2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.52)$$

and the lower bound is :

$$R_l = \frac{N^2}{2\sum_{i=o}^{N-1}\sum_{n=o}^{N-1} h_{i,n}^2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.53)$$

which for the DHT case reduces to :

$$R_1 = \frac{N}{2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.54)$$

(ii) For the modified realization, fig. 3.2, where rounding takes place after the accumulator instead of after the multiplier, the actual output MSE to MSS ratio is :

$$R = \frac{N}{\sum_{i=o}^{N-1} \sum_{n=o}^{N-1} h_{i,n}^2} \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.55)$$

which, for the DHT, reduces to :

$$R = \frac{\sigma_2^2}{\sigma_f^2} \qquad (3.56)$$

which is no longer dependent on the number of samples N. Fig. 3.5 shows a plot of the MSE to MSS as a function of N for the different cases discussed above and one step FFT. It is clear that the DHT filtering matrix technique introduces significantly less error than the FFT.

3.4.3 Arithmetic Errors in Floating Point FFT [1, 6, 7] :

For floating point arithmetic the output MSE to MSS for one step FFT is :

$$R = \frac{MSE}{MSS}\Big|_{\text{one step}} = 2 v \sigma_1'^2 \qquad (3.57)$$

where $v = \text{Log}_2 N$ i.e. $N = 2^v$ and $\sigma_1'^2 = \frac{2^{-2t}}{3}$.

For the FFT filter 2 steps of FFT are required. Assuming that no errors are introduced due to multiplication (weighting) in the frequency domain, the output MSE to MSS due to 2 steps FFT i.e. forward and inverse FFT is simply twice that of (3.57) [1]. Thus for the FFT filter :

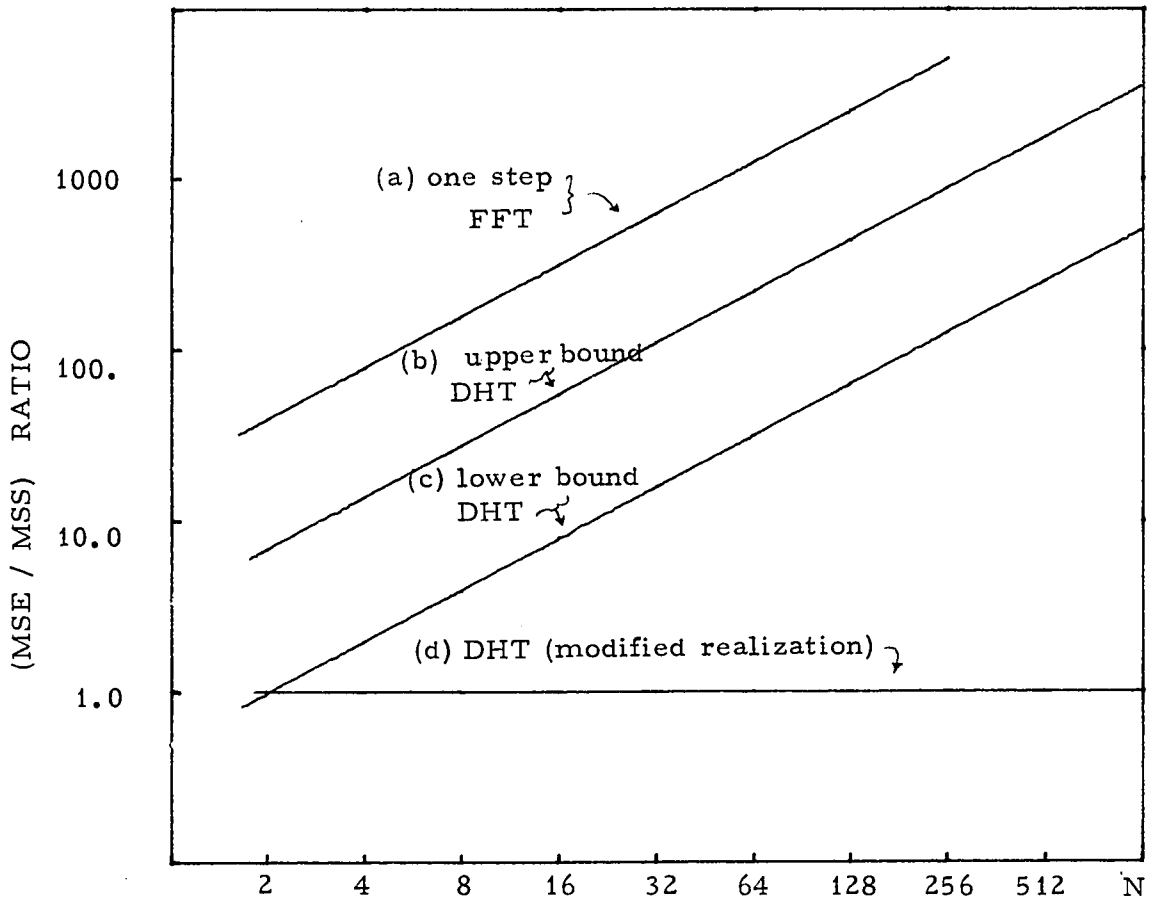


Fig. 3.5. MSE to MSS ratio for fixed point arithmetic :

(a) One step FFT
(b) Upper bound, DHT realization
(c) Lower bound, DHT realization
(d) DHT modified realization.

(MSE/MSS is normalized w.r.t. σ_x^2/σ_f^2)

$$R_{FFT} = \frac{MSE}{MSS} = 4 v \sigma_1'^2 \qquad (3.58)$$

3.4.4 Comparison for Floating Point Arithmetic :

It has been shown that the output MSE to MSS ratio for the DHT filtering matrix approach:

(i) For the realization in fig. 3.2 ; the output MSE to MSS ratio is :

$$R = \frac{\sum_{i=o}^{N-1} \sum_{n=o}^{N-1} h_{i,n}^2 \; E\{ \psi_{i,n} - 1)^2 \}}{\sum_{i=o}^{N-1} \sum_{n=o}^{N-1} h_{i,n}^2} \qquad (3.59)$$

which, for the DHT, reduces to :

$$R \cong \frac{N}{8} \sigma_1'^2 \qquad (3.60)$$

(ii) For the modified realization of fig. 3.3; the output MSE to MSS ratio is :

$$R \cong \sigma_1'^2 \qquad (3.61)$$

Comparing (3.58) with (3.60) and (3.61), we can see that :

1 - For the realization of fig. 3.3, the DHT filtering technique introduces less error that the FFT for values of N up to N = 256.

2 - For the modified realization of fig. 3.4, the DHT filtering approach introduces less error than the FFT for all values of N.

Fig. 3.6 shows a plot of the MSE to MSS ratio for the different cases reported in this section versus the number of samples N.

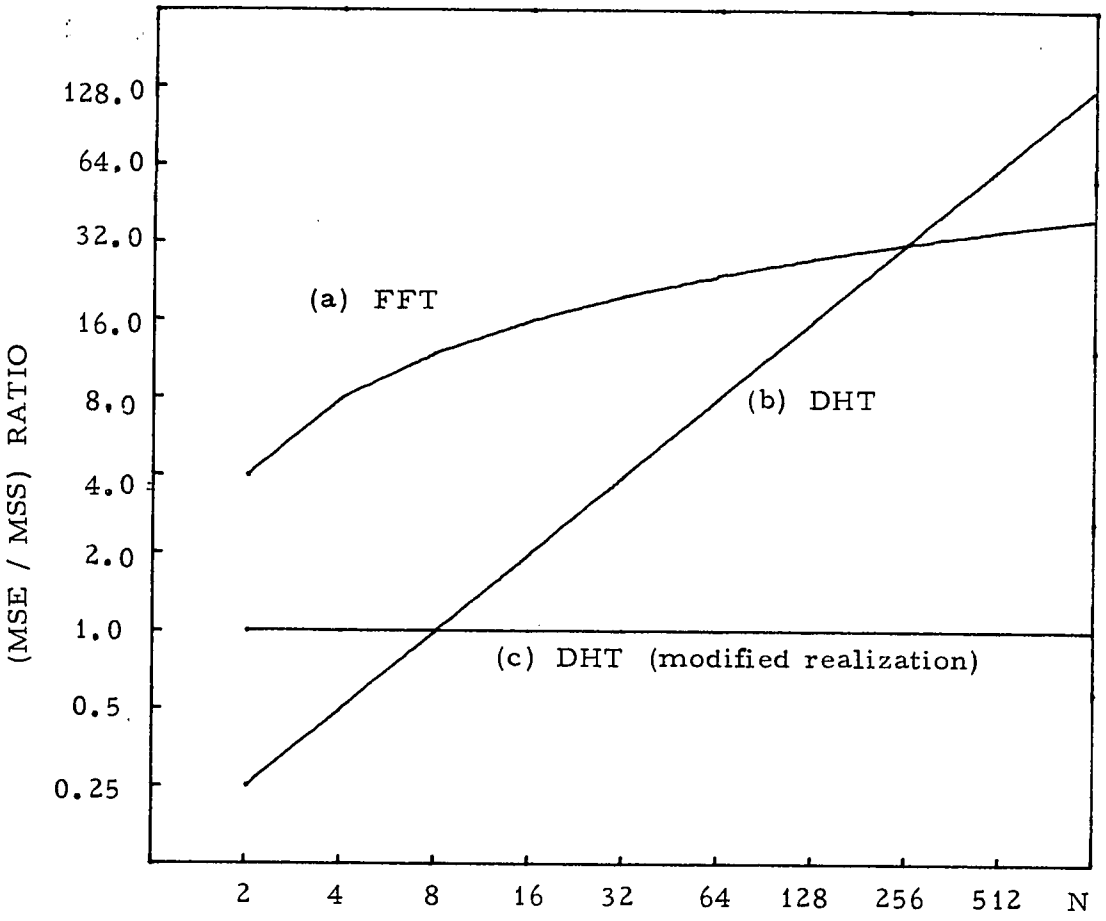


Fig. 3.6 MSE to MSS ratio for floating point arithmetic :

(a) two steps FFT.

(b) DHT, conventional realization.

(c) DHT, modified realization.

MSE to MSS ratio is normalized with respect to $\sigma_1'^2$,

3.5 CONCLUSION

The analysis of different sources of errors in the DHT filtering matrix method has been presented. It has been shown that for both fixed and floating point arithmetic the DHT filtering process introduces significantly less errors than one FFT process.

In addition, for the modified realization of the DHT filtering the output MSE to MSS signal ratio is no longer a function of the number of samples N.

Table 3.1

Bound on the output $|g_i|$ for the DHT

N	Max $\|g_i\|$	number of shifts in bits
8	1.41	1
16	1.84	1
256	3.65	2
512	4.05	3
1,024	4.49	3
25,000	6.52	3

APPENDIX 3.1

Statistics of $\psi_{i,n}$

From (3.42) $\psi_{i,n}$ is given by :

$$\psi_{i,n} = \begin{cases} (1+\rho_{i,o}) \prod_{j=3,5,7..}^{N-1} (1+\gamma_{i,j}) & n = 1, \quad i \text{ even} \\ (1+\rho_{i,n}) \prod_{j=n}^{N-1} (1+\gamma_{i,j}) & n = 3,5,7,\ldots(N-1), \; i \text{ even} \\ (1+\rho_{i,o}) \prod_{j=2,4,6}^{N-2} (1+\gamma_{i,j}) & n = 0, \quad i \text{ odd} \\ (1+\rho_{i,n}) \prod_{j=n}^{N-2} (1+\gamma_{i,j}) & n = 2,4,\ldots(N-2), \quad i \text{ odd} \end{cases} \tag{A.1}$$

The error quantities $\rho_{i,n}$ and $\gamma_{i,j}$ are uncorrelated to each other, to the input signal, and to matrix coefficients. They have a zero mean and a variance $\sigma_1'^2 = \frac{2^{-2t}}{3}$. Thus :

$$E\{\psi_{i,n}\} = 1.0 \tag{A.2}$$

$$E\{(\psi_{i,n}-1)^2\} = E\{\psi_{i,n}^2 - 2\psi_{i,n} + 1\}$$

$$= E\{\psi_{i,n}^2\} - 1 \tag{A.3}$$

From (A.1) :

$$E\{\psi_{i,n}^2\} = \begin{cases} (1 + \sigma_1'^2)^{N/2} & n = 1, \quad i \text{ even} \\ (1 + \sigma_1'^2)^{\frac{N-n+3}{2}} & n = 3,5,\ldots,(N-1), \quad i \text{ even} \\ (1 + \sigma_1'^2)^{N/2} & n = 0, \quad i \text{ odd} \\ (1 + \sigma_1'^2)^{\frac{N-n+2}{2}} & n = 2,4,\ldots,(N-2), \quad i \text{ odd} \end{cases} \tag{A.4}$$

Since $\sigma_1'^2 << 1$ (A.4) is reduced to :

$$E\{\psi_{i,n}^2\} \simeq \begin{cases} 1 + \frac{N}{2}\sigma_1'^2 & n = 0,1, \quad i \text{ even or odd} \\ 1 + \frac{N-n+3}{2}\sigma_1'^2 & n = 3,5,\ldots,(N-1), \text{ i even} \\ 1 + \frac{N-n+2}{2}\sigma_1'^2 & n = 2,4,\ldots,(N-2), \text{ i odd} \end{cases} \tag{A.5}$$

REFERENCES

[1] A. Oppenheim and R. Schafer, "Digital Signal Processing", Prentice-Hall, 1975. chapter 9, pp. 404-463.

[2] P. Welch, "A Fixed Point Fast Fourier Transform Error Analysis", IEEE Tran. on Audio and Electroacoustics, Vol. AU-17, June 1969, pp. 153-157.

[3] L. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing", Prentice-Hall 1975.

[4] A. Oppenheim and C. Weinstein, "A Bound on the Output of Circular Convolution with Application to Digital Filtering", IEEE Trans. on Audio and Electroacoustics, Vol AU-17, June 1969, pp. 120-124.

[5] B. Liu and T. Kaneko, "Error Analysis of Digital Filters Realized with Floating Point Arithmetic", Proc. IEEE, Vol 57, Oct. 1969, pp. 1735-1747.

[6] C. Weinstein, "Roundoff Noise in Floating Point Fast Fourier Transform Computation", IEEE Trans. Audio and Electroacoustics, Vol. AU-17, Sept. 1969, pp. 209-215.

[7] A. Oppenheim and C. Weinstein, "Effects of Finite Register Length in Digital Filtering and Fast Fourier Transform", Proc. IEEE, August 1972, pp. 957-976.

CHAPTER FOUR

A NEW APPROACH TO BANDLIMITED SIGNAL EXTRAPOLATION : THE EXTRAPOLATION MATRIX

4.1 INTRODUCTION

A central problem in signal processing is the determination of a bandlimited signal f(t) in terms of a known finite segment g(t) where :

$$g(t) = \begin{cases} f(t) & |t| < t_o \\ 0 & |t| > t_o \end{cases} \qquad (4.1)$$

i.e. it is required to extrapolate the unknown part of the signal f(t) in terms of g(t).

Such problem arises in several practical situations. In image processing systems, super-resolution [1,2] is obtained by extrapolating the spectrum of a finite object beyond the diffraction limits leading to picture enhancement , also in Fourier analysis and spectrum estimation where it is desired to find the Fourier transform, i.e. estimate the spectrum , $F(\omega)$ of the band-limited signal f(t) in terms of the known segment g(t). In this chapter we shall concentrate on the second application though the results are also applicable to the first problem.

Several extrapolation techniques has been presented [4,5,6,7, 8,9] . Among these techniques is the method of maximum entropy [6,8,9] where the freqency domain representation of the sampled version of the signal f(t) is assumed to have no finite zeroes (all poles) and the poles lie inside the unit circle in the z-plane. Also included is the prolate spheroidal function expansion technique [7] and the iterative extrapolation [4,5]

where a priori assumption of the signal band limits is made.

In the iterative extrapolation technique as presented by Papoulis [4] and Gerchberg [5], the DFT or FFT transforms and their inverses are used iteratively to extrapolate the unknown part of the signal and hence estimate its spectrum. Papoulis [4] has shown that this technique is basically the same as the prolate spheroidal expansion method but gives considerable computational and storage savings as compared with the prolate spheroidal expansion method since it involves only DFT's and their inverses and can make use of the available computational savings using the FFT to evaluate the DFT.

Nevertheless, with the DFT or FFT used iteratively, a considerable amount of computation has to be performed. For example if n is the number of iterations required to extrapolate the unknown part of the signal, the DFT or FFT has to be used $2n$ times i.e. if the FFT is used, $2n\ N \log N$ complex arithmetic operations are required. Often the convergence of this algorithm is relatively slow and a large number of iterations is required. This would form a serious obstacle in applying such technique in real time.

In this chapter we present an extrapolation technique based on the iterative extrapolation technique as presented in [4,5] and the filter matrix transform developed in chapter two, where the iterative procedure is eliminated and the extrapolation process is achieved by a single matrix operation equivalent to n iterations. Thus a significantly more efficient signal extrapolation is achieved, which makes real time or fast processing quite feasible.

First the iterative extrapolation technique [4,5], is reviewed

then a new model for this technique is presented which leads to the derivation of the Extrapolation Matrix equivalent to n iterations. Subsequently the extrapolation matrix equivalent to an infinite number of iterations is derived. The properties of the extrapolation matrix are discussed and several examples are presented. A comparison is made between the direct extrapolation and the iterative extrapolation using the FFT, in terms of number of arithmetic operations (multiplication and addition) required in both cases, and the accuracy of the results obtained. It will be shown that the direct method requires fewer computations than the iterative method. In addition we shall show that with the extrapolation matrix corresponding to an infinite number of iterations exact results are obtained.

Finally a proposed fast extrapolator realization, based on the extrapolation matrix approach, is presented which will allow adjustment to any given bandwith.

4.2 REVIEW OF THE ITERATIVE EXTRAPOLATION TECHNIQUE [4,5].

Given a known segment g(t) of a band-limited signal f(t) defined on $[-t_0, t_0]$, it is required to extrapolate the unknown part of f(t) and find its spectrum F(ω), where :

$$F(\omega) = 0 \qquad |\omega| > \omega_o \tag{4.2}$$

The iterative extrapolation presented by Papoulis [4] and Gerchberg [5] is summarized below :

1 - Given g(t), find its Fourier transform:

$$G(\omega) = G_o(\omega) = \int_{-t_o}^{t_o} g(t)\, e^{-j\omega t}\, dt \tag{4.3}$$

Let

$$g(t) = g_o(t) \tag{4.4}$$

2 - The first iteration proceeds as follows :

In the frequency domain the resulting spectrum $G_o(\omega)$ is band limited by setting the part of the spectrum corresponding $|\omega| > \omega_o$ to zero, i.e. :

$$F_1(\omega) = \begin{cases} G_o(\omega) & |\omega| < \omega_o \\ 0 & |\omega| > \omega_o \end{cases} \tag{4.5}$$

3 - Find the inverse Fourier transform of $F_1(\omega)$ i.e. $f_1(t)$:

$$f_1(t) = \frac{1}{2\pi} \int_{-\omega_o}^{\omega_o} F_1(\omega)\, e^{j\omega t}\, d\omega \tag{4.6}$$

4 - Replace the part of $f_1(t)$ corresponding to $|t| < t_o$ by the known part g(t) to form the signal $g_1(t)$ i.e. :

$$g_1(t) = \begin{cases} g(t) & |t| < t_o \\ f_1(t) & |t| > t_o \end{cases} \tag{4.7}$$

5 - The first iteration ends by computing the transform :

$$G_1(\omega) = \int_{-\infty}^{\infty} g_1(t)\, e^{-j\omega t}\, dt \tag{4.8}$$

and steps 2-5 are repeated for the 2^n, $3^{rd} \ldots n^{th}$ iterations.

In general the n^{th} iteration proceeds as follows :

$$\text{(i)} \quad F_n(\omega) = \begin{cases} G_{n-1}(\omega) & |\omega| < \omega_o \\ 0 & |\omega| > \omega_o \end{cases} \tag{4.9}$$

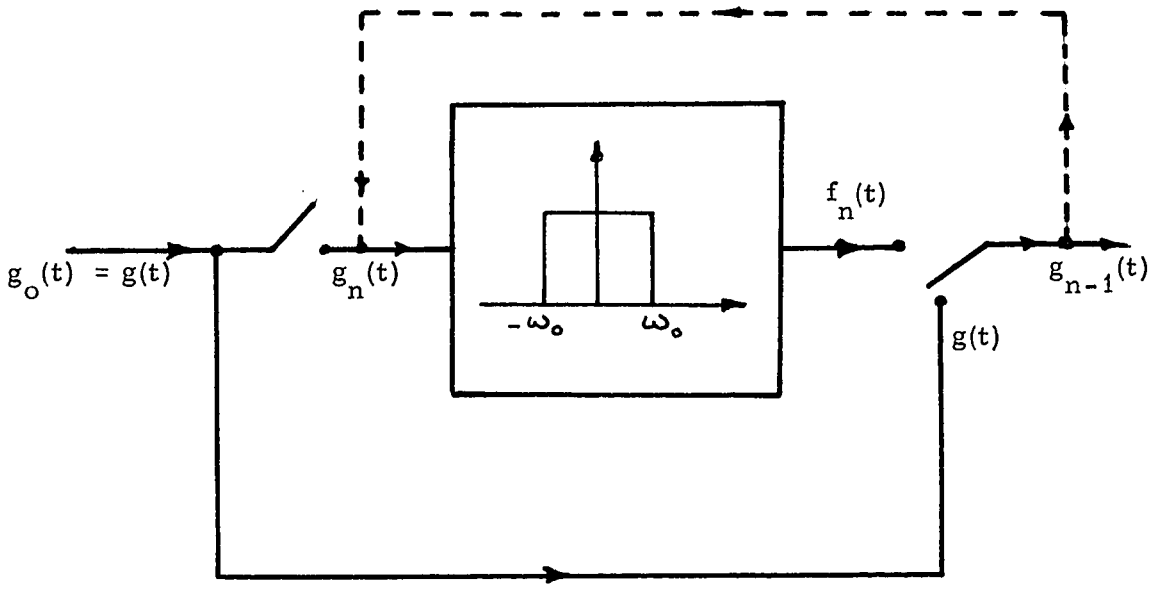


Fig. 4.1 Analog simulation of the iterative extrapolation procedure as proposed by Papoulis [4]. The switch to the left closes only at the beginning and remains open afterwords. The switch to the right switches to g(t) for $|t| < t_o$ and to $f_n(t)$ for $|t| > t_o$.

(ii) $$f_n(t) = \frac{1}{2\pi} \int_{-\omega_o}^{\omega_o} F_n(\omega)\, e^{j\omega t}\, d\omega \qquad (4.10)$$

(iii) $$g_n(t) = \begin{cases} g_o(t) & |t| < t_o \\ f_n(t) & |t| > t_o \end{cases} \qquad (4.11)$$

(iv) The n^{th} iteration ends by computing the transform

$$G_n(\omega) = \int_{-\infty}^{\infty} g_n(t)\, e^{-j\omega t}\, dt \qquad (4.12)$$

The signal $g_n(t)$ is the extrapolated signal after n iterations and its transform $G_n(\omega)$ is the estimated spectrum. Papoulis [4] has shown that the iterative procedure indeed converges and as the number of iterations n tends to ∞ the extrapolated signal $f_n(t)$ tends to $f(t)$. Indeed this is true if the signal bandlimits are estimated correctly.

The extrapolation process [4] is illustrated in fig. 4.1. which shows a model of the simulation of the iterative procedure. It consists of an ideal lowpass filter and a two way switch (right hand side switch).

4.3. AN ALTERNATIVE MODEL OF THE ITERATIVE EXTRAPOLATION TECHNIQUE

Examining carefully the simulation model of the iterative extrapolation, as shown in fig. 4.1, it is seen that at the start of the procedure the signal $g_o(t) = g(t)$ is passed through the ideal lowpass filter to form the signal $f_1(t)$. The signal $g_1(t)$ is then formed by replacing the part of $f_1(t)$ corresponding to $|t| < t_o$ by $g_o(t)$. The signal $f_2(t)$ is formed by passing $g_1(t)$ through the lowpass filter and so on.

Since the lowpass filtering is a linear operation we can divide the signal $g_1(t)$ into two added parts as :

$$g_1(t) = g_o(t) + f_1'(t) \qquad (4.13)$$

where

$$f_1'(t) = \begin{cases} 0 & |t| < t_o \\ f_1(t) & |t| > t_o \end{cases} \qquad (4.14)$$

Since the signal $f_2(t)$ is formed by passing $g_1(t)$ through the lowpass filter, then from (4.14) it can be seen that alternatively $f_2(t)$ can be formed by adding $f_1(t)$ to the signal resulting from passing $f_1'(t)$ through the lowpass filter. The signal $f_1'(t)$ is generated by passing $f_1(t)$ through a one way switch which is open for $|t| < t_o$ and is closed for $|t| > t_o$.

Fig. 4.2 shows an alternative model for the realization (or simulation) for the iterative extrapolation procedure. It consists of a first lowpass filter followed by cascaded sections, each composed of a one way switch and a lowpass filter. The extrapolated signal is formed by adding the outputs of the first lowpass filter and the successive stages as shown.

4.4. THE EXTRAPOLATION MATRIX

In the actual realization of the extrapolation algorithm [4,5], the Fourier transform is evaluated using the DFT or FFT and the signals involved have to be in digital form.

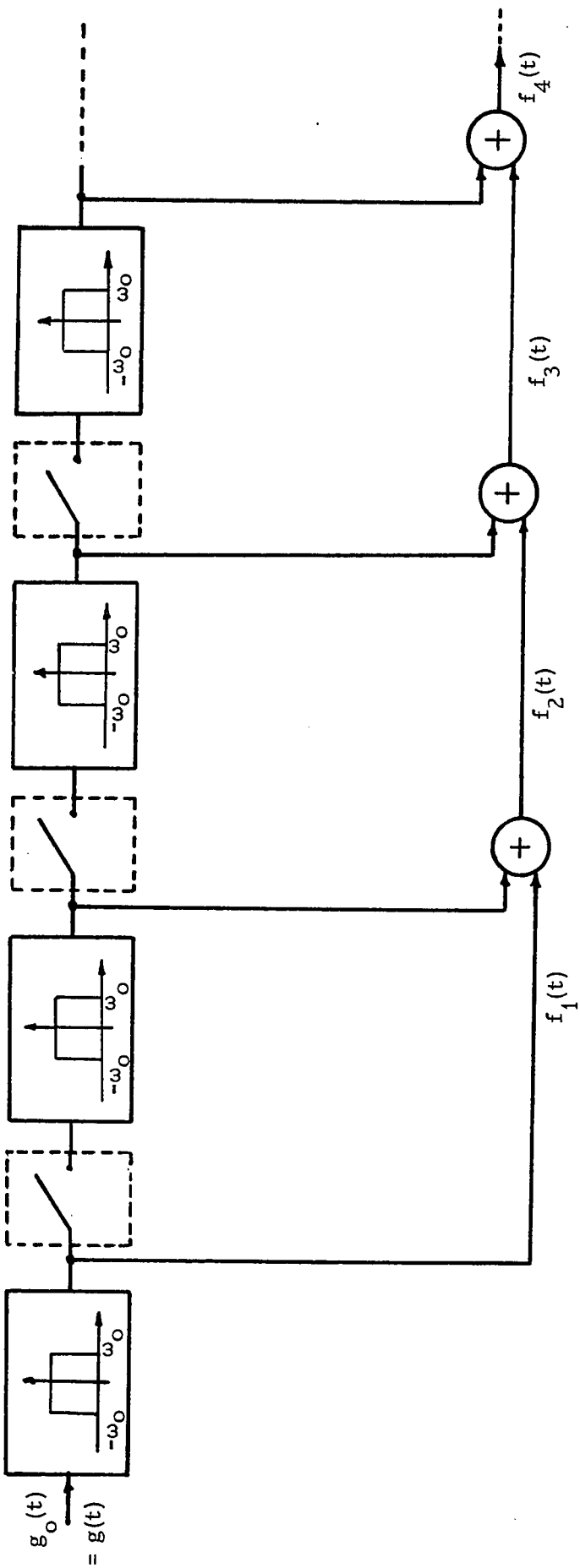


Fig. 4.2. Alternative model for the iterative extrapolation technique using cascade of one way switches and lowpass filters.

Before we proceed with the derivation of the extrapolation matrix the problem shall be reformulated in digital form.

Let $f(iT) = f(i)$ be the sampled version of the continuous time signal $f(t)$ i.e., $t = iT$, and $i = 0, \pm 1, .., \pm \frac{N-1}{2}$. The known part of the signal in sampled form is $g_o(kT) = g_o(k)$, $k = 0, \pm 1, \pm 2, ..., M$. The problem now is to extrapolate $(N - 2M - 1)$ unknown samples. T is the sampling interval.

Define:

$[f]$ = the signal to be extrapolated in sampled vector form of length N samples.

$$= [f(\tfrac{-N+1}{2}), \ldots, f(-1), f(0), f(1), \ldots f(\tfrac{N-1}{2})]' \qquad (4.15)$$

where the prime denotes transpose.

$[g_o]$ = vector of length N samples whose middle $(2M+1)$ elements are the known part of the signal samples and the rest of the elements are zeroes.

$$= [0, 0, \ldots, 0, g(-M), \ldots g(-1), g(0), g(1), \ldots, g(M), 0, \ldots 0, 0]' \qquad (4.16)$$

$[\bar{g}_o]$ = vector of length $(2M+1)$ containing the known signal samples.

$$= [g(-M), \ldots, g(-1), g(0), g(1), \ldots, g(M)]' \qquad (4.17)$$

$[f_j]$ = the extrapolated signal at the j^{th} iteration in vector form of length N samples.

$$= [f_j(\tfrac{-N+1}{2}), \ldots f_j(-1), f_j(0), f_j(1), \ldots f_j(\tfrac{N-1}{2})]' \qquad (4.18)$$

$[\hat{f}_j]$ = same as $[f_j]$ except the middle $(2M+1)$ elements in the vector are replaced with zeroes.

The lowpass filtering operation is performed by the lowpass matrix transformation as shown in chapter two, relation (2.27). Thus, with reference to fig. 4.3, we have :

$$[f_1] = [LP]\ [g_o] \tag{4.19}$$

and,

$$[f_2] = [f_1] + [LP]\ [X]\ [f_1] \tag{4.20}$$

where $[LP]$ is the lowpass matrix, $[X]$ is the switching matrix representing the operation of the one way switch, which is a diagonal matrix of the form :

$$[X] = \text{diag.}\ [\ 1,\ 1,\ 1,\ \ldots,\ 1,\ 0,\ 0,\ \ldots\ldots,\ 0,\ 0,\ 1,\ \ldots\ldots,1,1,1\ ] \tag{4.21}$$

As we can see the switching matrix is an identity matrix of order N, with (2M+1) centre diagonal elements replaced by zeroes.

From (4.19) and (4.20) we have :

$$[f_2] = \{[I] + [LP]\ [X]\ \}\ [LP]\ [g_o] \tag{4.22}$$

where $[I]$ is the identity matrix. Similarly :

$$[f_3] = \{[I] + [LP][X] + [LP]\ [X]\ [LP][X]\}\ [LP][g_o] \tag{4.23}$$

In general, for n iterations, the extrapolated signal, in vector form, $[f_n]$ is given by :

$$[f_n] = \{[I] + \sum_{m=1}^{n-1} ([LP]\ [X])^m\}\ [LP]\ [g_o] \tag{4.24}$$

Let,

$$[S] = [LP]\ [X] \tag{4.25}$$

Thus (4.24) gives :

$$[f_n] = \{\sum_{m=o}^{n-1} [S]^m\}\ [LP]\ [g_o] \tag{4.26}$$

$[S]$ is basically a lowpass matrix whose middle $(2M+1)$ columns are zeroes. $[S]^0$ is the identity matrix $[I]$. Relation (4.26) can be rewritten as :

$$[f_n] = [E_n]\,[g_o] \tag{4.27}$$

where $[E_n]$ is the Extrapolation Matrix given by :

$$[E_n] = [E'_n]\,[LP] = \{\sum_{m=o}^{n-1} [S]^m\}\,[LP] \tag{4.28}$$

The extrapolation process is reduced to a single matrix operation i.e. the extrapolated signal is obtained by multiplying the known part of the signal, in vector form, by the extrapolation matrix. Thus the iterative nature of the process is eliminated.

It shall be shown later that only a submatrix of the extrapolation matrix needs to be generated and used in the actual computation of the unknown signal samples.

Fig. 4.3 and fig. 4.4 illustrate the different steps of deriving the extrapolation matrix. In fig. 4.3 the operation of the one way switch and the lowpass matrix is achieved by the switching matrix and the lowpass matrix. In fig. 4.4.a both the switching and the lowpass filtering operations are combined into one matrix operation; $[S]$ matrix. In fig. 4.4.b the extrapolation process is further reduced to two matrix operations i.e. the lowpass matrix and $[E'_n]$. Finally in fig. 4.4.c the whole operation is reduced to a single matrix operation; $[E_n]$ the extrapolation matrix.

Another model of the realization (or simulation) of the iterative process is shown in fig. 4.5. This model is easily derived by rewriting (4.27) and (4.28) in the form:

$$[f_n] = \{\sum_{m=o}^{n-1} [S]^m\}\,[LP]\,[g_o]$$

$$= \{[I] + [S]\,[I + [S]\,\{[I] + [S]([I]+[S]([I]+[S]\ldots]\}.$$

$$[LP]\,[g_o] \tag{4.29}$$

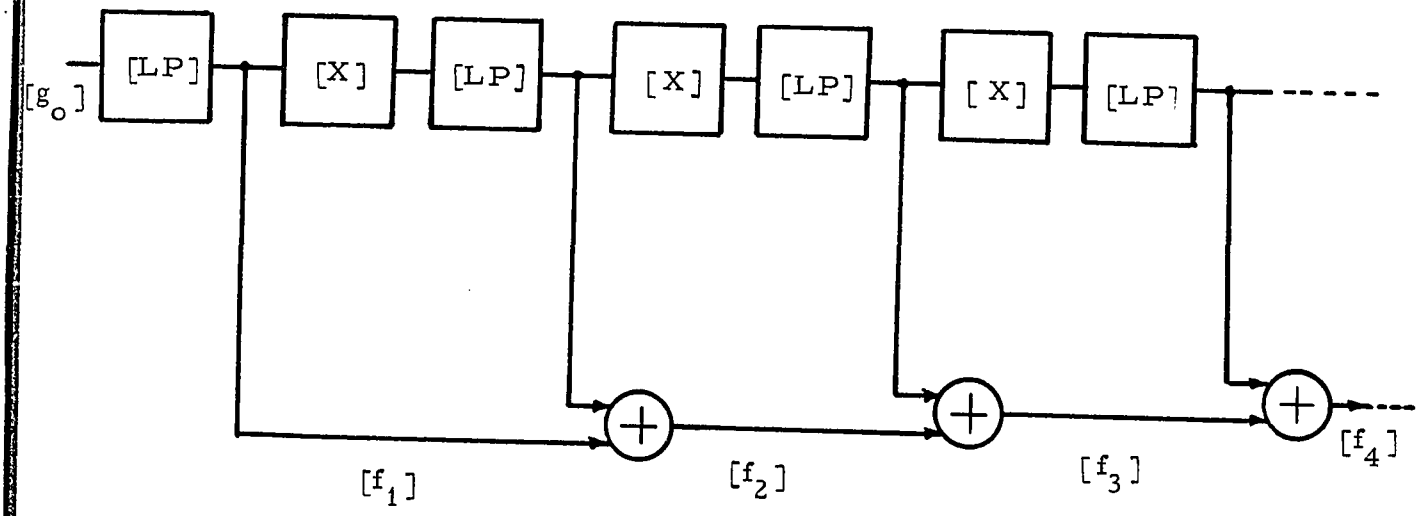


Fig. 4.3. Same as fig. 4.2 but the switching and lowpass filtering operations are replaced by matrix operations.

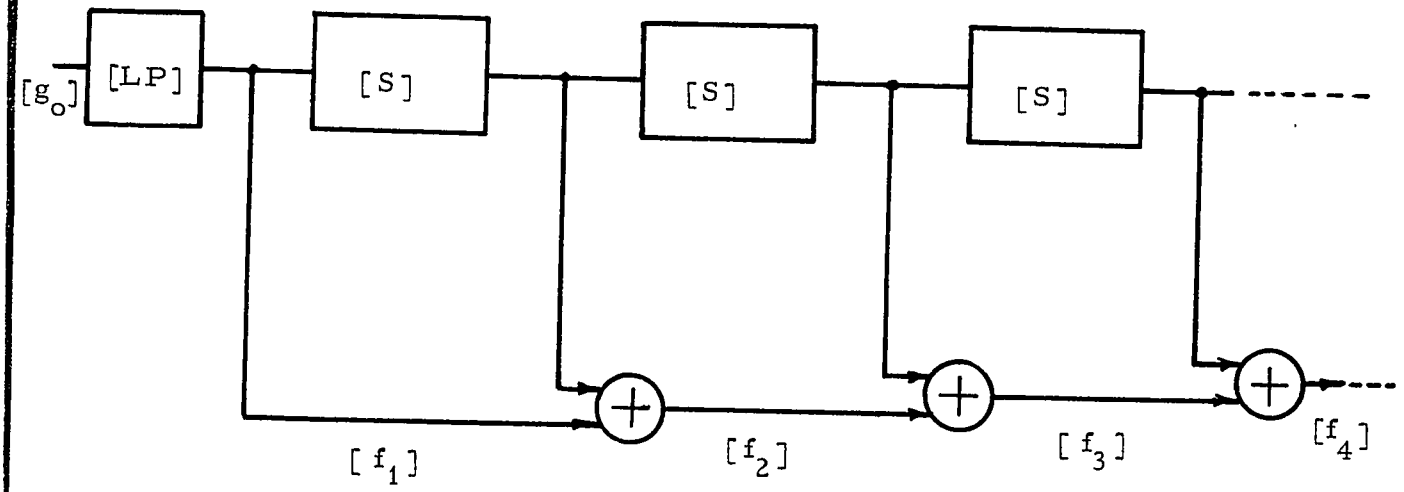


Fig. 4.4.a The lowpass and switching matrix operations are combined into single matrix operation.

Fig. 4.4.b Reduction of the iterative extrapolation process to two matrix operations

$$[E'_n] = \sum_{j=o}^{n-1} [S]^j$$

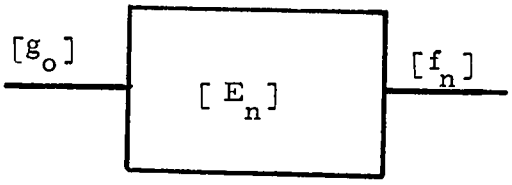


Fig. 4.4.c Reduction of the iterative extrapolation process to single matrix operation.

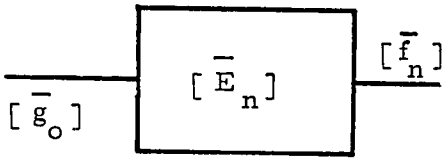


Fig. 4.4.d Final extrapolation process using the submatrix $[\bar{E}_n]$ to extrapolate the unknown part of the signal.

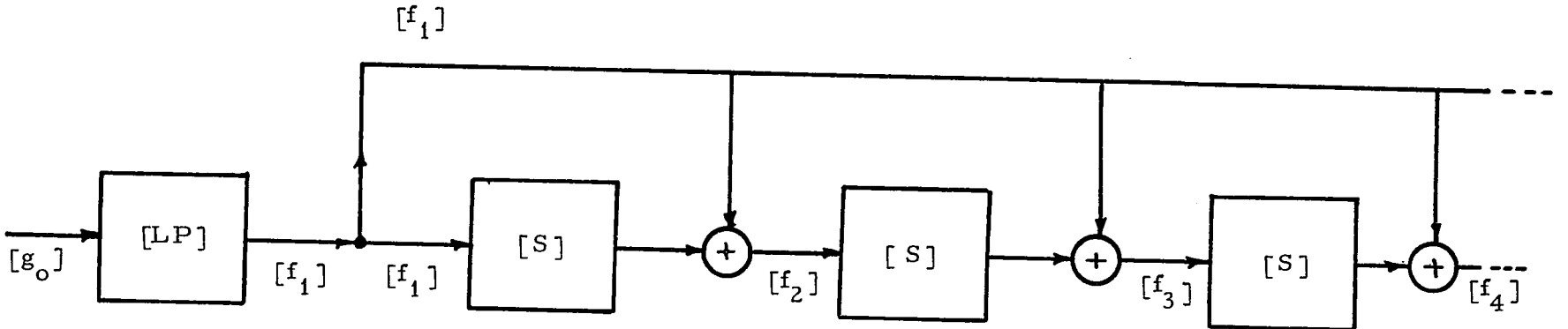


Fig. 4.5 Another model of the simulation of the iterative extrapolation procedure equivalent to the one shown in fig. 4.4.a.

4.5. THE EXTRAPOLATION MATRIX CORRESPONDING TO AN INFINITE NUMBER OF ITERATIONS.

In the previous section it has been shown that the matrix $[E'_n]$ is given by :

$$[E'_n] = \sum_{m=o}^{n-1} [S]^m \tag{4.30}$$

The matrix $[E'_n]$ is a power series sum and may be evaluated using matrix algebra [10]. Thus (4.30) gives:

$$[E'_n] = \{[I] - [S]^n\} \{[I] - [S]\}^{-1} \tag{4.31}$$

and the extrapolation matrix $[E_n]$ is :

$$[E_n] = \{[I] - [S]^n\} \{[I] - [S]\}^{-1} [LP] \tag{4.32}$$

As the number of iterations n tends to infinity the extrapolated signal $[f_\infty]$ becomes equal to [f], and an exact signal extrapolation is obtained.

The extrapolation matrix corresponding to an infinite number of iterations is obtained by putting $n = \infty$ in (4.30), giving :

$$[E'_\infty] = \sum_{m=o}^{\infty} [S]^m = \{[I] - [S]\}^{-1} \tag{4.33}$$

and the extrapolation matrix in this case is :

$$[E_\infty] = \{[I] - [S]\}^{-1} [LP] \tag{4.34}$$

A necessary condition for relation (4.33) to be valid i.e. the infinite series summation converges is [10]:

$$|\lambda_i| < 1 \quad , \text{ for all } i \tag{4.35}$$

where λ_i is the i^{th} characteristic value of the matrix [S].
Thus before using the extrapolation matrix $[E_\infty]$ one should check first if (4.35) is satisfied. In practice finding the characteristic values λ_i of an N by N matrix might not be an easy task. An alternative approach is to find the norm of the matrix [S], defined as [11] :

$$\|S\| = \sqrt{\sum_{i,j}^{N-1} s_{i,j}^2} \qquad \text{Euclidean norm} \tag{4.36}$$

or

$$\|S\| = \max_{i} \sum_{j=o}^{N-1} |s_{i,j}| \tag{4.37}$$

or

$$\|S\| = \max_{j} \sum_{i=o}^{N-1} |s_{i,j}| \tag{4.38}$$

Where $s_{i,j}$ is the $(i,j)^{th}$ element in the matrix [S], $\|S\|$ is the norm of the matrix [S].

For all the above norms we have [11] :

$$\|S\| \geq |\lambda_i| \qquad \text{for all } i \tag{4.39}$$

Thus a sufficient condition for (4.33) to converge is :

$$\|S\| < 1.0 \tag{4.40}$$

In general [S] is a function of N, M, and β, where β is the filter cutoff frequency. Thus with the proper choice of these parameters, relations (4.35) or (4.40) will be satisfied.

4.6. PROPERTIES OF THE EXTRAPOLATION MATRIX.

In order to illustrate the different properties of the extrapolation matrix, we shall start by investigating the properties and forms of the different matrices presented in the previous section. These properties are best illustrated through a simple example for N = 7, β = 1, M = 1, and n = 4.

The lowpass matrix properties have been discussed in chapter two. The matrix [S] is a square N by N matrix whose middle (2M+1) columns are zeroes i.e. for the example under consideration, [S] is :

$$[S] = \begin{bmatrix} 0.42 & 0.32 & 0 & 0 & 0 & 0.07 & .32 \\ 0.32 & 0.42 & 0 & 0 & 0 & -.11 & 0.07 \\ 0.07 & 0.32 & 0 & 0 & 0 & -.11 & -0.11 \\ -0.11 & 0.07 & 0 & 0 & 0 & 0.07 & -0.11 \\ -0.11 & -0.11 & 0 & 0 & 0 & 0.32 & 0.07 \\ 0.07 & -0.11 & 0 & 0 & 0 & 0.42 & 0.32 \\ 0.32 & 0.07 & 0 & 0 & 0 & 0.32 & 0.42 \end{bmatrix} \qquad (4.41)$$

for $n = 4$;

$$[E'_4] = \{[I] + [S] + [S]^2 + [S]^3\} \qquad (4.42)$$

From (4.41) and (4.42),the matrix $[E'_4]$ is :

$$[E'_4] = \begin{bmatrix} 2.20 & 0.87 & 0 & 0 & 0 & 0.37 & 0.98 \\ 0.87 & 1.97 & 0 & 0 & 0 & -0.15 & 0.37 \\ 0.24 & 0.59 & 1 & 0 & 0 & -0.30 & -0.15 \\ -0.21 & 0.03 & 0 & 1 & 0 & 0.03 & -0.21 \\ -0.15 & -0.30 & 0 & 0 & 1 & 0.59 & 0.24 \\ 0.37 & -0.15 & 0 & 0 & 0 & 1.97 & 0.87 \\ 0.98 & 0.37 & 0 & 0 & 0 & 0.87 & 2.20 \end{bmatrix} \qquad (4.43)$$

For the matrix $[E_n]$, the first $(\frac{N-1}{2})$ columns are exactly the same as the last $(\frac{N-1}{2})$ columns but in reverse order and the same applies for the rows.

Finally the extrapolation matrix $[E_4]$ is :

$$[E_4] = [E'_4]\ [LP] \qquad (4.44)$$

Thus ,

$$[E_4] = \begin{bmatrix} 1.56 & 1.11 & 0.30 & -0.26 & -0.15 & 0.54 & 1.31 \\ 1.11 & 1.17 & 0.67 & 0.00 & -0.34 & -0.10 & 0.54 \\ 0.30 & 0.67 & 0.69 & 0.33 & -0.12 & -0.34 & -1.55 \\ -0.26 & 0.00 & 0.33 & 0.48 & 0.33 & 0.00 & -0.26 \\ -0.15 & -0.34 & -0.12 & 0.33 & 0.69 & 0.67 & 0.30 \\ 0.54 & -0.10 & -0.34 & 0.00 & 0.67 & 1.17 & 1.11 \\ 1.31 & 0.54 & -0.15 & -0.26 & 0.30 & 1.11 & 1.56 \end{bmatrix} \qquad (4.45)$$

The extrapolation matrix is a square matrix of order N. It has the same properties as the $[E'_n]$ matrix. The first $(\frac{N-1}{2})$ columns are equal to the last $(\frac{N-1}{2})$ columns. The middle column is symmetrical around the centre element. This also applies for the rows of the matrix. Though these properties are illustrated in (4·45) for n = 4, it is quite general for any n.

Only a small portion of the matrix is needed to perform the extrapolation process. This will be shown below :

As shown before the extrapolation is implemented by the multiplication of the vector $[g_o]$, containing the known part of the signal, by $[E_n]$. Thus, for n = 4 as in the previous example we have:

$$[f_n] = [f_4] = [E_4]\ [g_o] \tag{4.46}$$

$$[f_4] = [E_4] \begin{bmatrix} 0 \\ 0 \\ g_o(-1) \\ g_o(0) \\ g_o(1) \\ 0 \\ 0 \end{bmatrix} \tag{4.47}$$

We note that because of the zero elements in $[g_o]$, only the middle (2M+1) columns are needed and as a first step towards simplification of the process, the N by N extrapolation matrix is reduced to N by (2M+1) matrix. For the example under consideration (n = 4, N = 7, and M = 1) the extrapolation process is reduced to :

$$[f_4] \quad \begin{bmatrix} f_4(-3) \\ f_4(-1) \\ f_4(-1) \\ f_4(0) \\ f_4(1) \\ f_4(2) \\ f_4(3) \end{bmatrix} = \begin{bmatrix} 0.30 & -0.26 & -0.15 \\ 0.67 & 0 & -0.34 \\ 0.69 & 0.33 & -0.12 \\ 0.33 & 0.48 & 0.33 \\ -0.12 & 0.33 & 0.69 \\ -0.34 & 0.00 & 0.67 \\ -0.15 & -0.26 & 0.30 \end{bmatrix} \begin{bmatrix} g_o(-1) \\ g_o(0) \\ g_o(t) \end{bmatrix} \qquad (4.48)$$

Again, because of the symmetry properties discussed earlier in this section, (4.48) can be further reduced to :

$$\begin{bmatrix} f_4(0) \\ f_4(1) \\ f_4(2) \\ f_4(3) \end{bmatrix} = \begin{bmatrix} 0.33 & 0.48 & 0.33 \\ -0.12 & 0.33 & 0.69 \\ -0.34 & 0.00 & 0.67 \\ -0.15 & -0.26 & 0.30 \end{bmatrix} \begin{bmatrix} g_o(-1) \\ g_o(0) \\ g_o(1) \end{bmatrix} \qquad (4.49)$$

It can be seen from (4.49) that the extrapolation process is reduced to multiplying the known part of the signal by a sub-extrapolation matrix (portion of the extrapolation matrix), i.e. a $(\frac{N+1}{2})$ by $(2M+1)$ matrix and for our example it is a 4 by 3 matrix.

Finally, the process can be further simplified by noting that part of the signal, is known, i.e $[\bar{g}_o]$, so there is no need to recompute it; for our example $f_4(0)$ and $f_4(1)$ need not be computed and they simply can be assumed $g_o(0)$ and $g_o(1)$. Hence the final extrapolation process is:

$$\begin{bmatrix} f_4(2) \\ f_4(3) \end{bmatrix} = \begin{bmatrix} -0.34 & 0.00 & 0.67 \\ -0.15 & -0.26 & 0.3 \end{bmatrix} \begin{bmatrix} g_o(-1) \\ g_o(0) \\ g_o(1) \end{bmatrix} \qquad (4.50)$$

Thus as we can see that the operation of the extrapolation matrix is reduced to multiplying a 2 by 3 matrix by the known part of the signal in vector form.

In general, to perform the extrapolation, it is only required to generate and store a submatrix of (2M+1) columns and $(\frac{N-1}{2} - M)$ rows as shown below :

$$[E_n]\,[g_o] = \left[\begin{array}{c} \\ \\ \\ \left(\frac{N-1}{2} - M\right) \\ \downarrow \end{array}\right. \overbrace{\qquad\qquad}^{2M+1} \left.\begin{array}{c} \\ \\ \\ [\bar{E}_n] \\ \\ \end{array}\right] \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ [\bar{g}_o] \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \qquad (4.51)$$

Therefore :

$[\bar{f}_n]$ = unknown part of the signal to be extrapolated

$$= [\bar{E}_n]\,[\bar{g}_o] \qquad (4.52)$$

Thus for signal extrapolation only $(\frac{N-1}{2} - M) \cdot (2M+1)$ real multiplications are required.

4.7. BANDPASS SIGNALS

In the analysis and discussions of the proposed extrapolation approach we have concentrated on bandlimited signals of the lowpass type, i.e. $F(\omega)$ is of the form :

$$F(\omega) = 0 \qquad |\omega| > \omega_o \qquad (4.53)$$

For bandpass signals extrapolation i.e. extrapolating a band-limited signal with a spectrum of the form :

$$F(\omega) = 0 \qquad |\omega| < \omega_1, \quad |\omega| > \omega_2 \qquad (4.54)$$

where ω_1 and ω_2 are the lower and upper signal frequencies respectively,

there are two approaches to handle this situation, first we can demodulate the signal to shift its frequency spectrum to be of the form (4.53), then perform the extrapolation as proposed and subsequently modulate the extrapolated signal to shift its spectrum to its original position.

The other approach is replacing the lowpass filters shown in fig. 4.1 to fig. 4.7 by bandpass filters, this would mean that the lowpass matrix is replaced by the bandpass matrix and the proposed extrapolation approach is applied directly. In this case the [S] matrix would be :

$$[S] = [BP]\ [X] \tag{4.55}$$

where [BP] is the bandpass matrix.

Later on we shall illustrate this case in several extrapolation examples.

4.8. COMPARISON BETWEEN THE PROPOSED EXTRAPOLATION MATRIX APPROACH AND THE ITERATIVE EXTRAPOLATION APPROACH.

Before we proceed with examples, it is necessary to compare the proposed approach with the iterative extrapolation approach [4,5] using the FFT. The comparison is made in terms of the number of arithmetic operations (e.g. multiplication or addition) necessary to perform the extrapolation process or in other words speed of both approaches. Also we shall compare the results obtained using both approaches.

For the iterative extrapolation approach using the FFT, the number of arithmetic operations (multiplication) necessary to perform n iterations is approximately :

$$K_{FFT} = 2\,n\,\frac{N}{2}\,\mathrm{Log}_2 N \quad \text{complex multiplication} \tag{4.56}$$

where N is the number of samples and n is the number of iterations.

Since one complex multiplication is equivalent to 4 real multiplications, thus (4.56) gives :

$$K_{FFT} = 4\, n\, N \log_2 N \text{ real multiplications} \qquad (4.57)$$

Note that in (4.56) and (4.57) we have ignored the multiplication in the frequency domain and the process of replacing the known part of the signal.

For the proposed extrapolation matrix approach it has been shown in section 4.5 that the number of real multiplications required for extrapolating the unknown part of the signal is :

$$K_p = \left(\frac{N-1}{2} - M\right)(2M+1) \quad \text{real multiplications} \qquad (4.58)$$

Comparing (4.57) and (4.58), it is clear that performing the direct extrapolation requires significantly fewer computations. For example if $N = 1024$, $M = 50$, thus in this case :

$$K_{FFT} = 4n \times 1024 \times 10 \simeq 40{,}000 \times n \qquad (4.59)$$

and

$$K_p = \left(\frac{1024-1}{2} - 50\right)(101) \simeq 45{,}000 \qquad (4.60)$$

It can be seen that, even for $n = 2$, i.e. two iterations, the direct extrapolation is computationally more efficient. In general as it has been shown by Papoulis [4] and Gerchberg [5], the convergence of the iterative extrapolation approach is relatively slow and a high number of iterations is required to obtain an approximate extrapolated signal. Papoulis [4] has shown one simple example of extrapolating a sinc function and showed that after 8 iterations a good approximation is obtained, he also pointed out that the reason of fast convergence is due

to the simplicity of the example. Gerchberg [5] showed examples of extrapolating 2 point target (2 sine wave functions) with a number of iterations in the order of 100 to 200 iterations to obtain an approximation to the desired signal.

It would be almost impossible using the iterative extrapolation approach to extrapolate any signal exactly since that would require infinite number of computations. We have shown in section 4.5.1 that this can be done i.e. exact extrapolation, using the extrapolation matrix $[E_\infty]$.

4.9. EVALUATION OF THE PERFORMANCE OF THE EXTRAPOLATION MATRIX APPROACH AND EXAMPLES.

In the previous section we have shown that the direct extrapolation is faster and much more computationally efficient than the iterative approach using the FFT. It also gives exact extrapolation results when the extrapolation matrix $[E_\infty]$ is used.

When using the extrapolation matrix two situations might arise :

(i) The matrix [S], satisfies relation (4.35) and hence the summation in (4.33) converges and consequently $[E_\infty]$ converges and relation (4.33) is valid. In this case exact extrapolation is obtained.

(ii) The matrix [S] does not satisfy relation (4.35) and consequently $[E_\infty]$ diverges. In this case one has to work with the extrapolation matrix equivalent to a finite number of iterations, i.e. $[E_n]$, and the results obtained are approximations to the original signal to be extrapolated.

Later on we shall show that situation (ii) can be avoided by varying the sampling frequency. The performance of the proposed approach is best illustrated through several examples.

4.9.1: Numerical Examples :

(1) Example 1 : Single sine wave signal:

In this example we consider the extrapolation a signal consisting of single sine wave given a small segment of the original signal.

$$f(t) = \sin(\omega_o t) \tag{4.61}$$

or in sampled form

$$f(iT) = f(i) = \sin(2\pi W_o i) \tag{4.62}$$

where W_o is the signal normalized frequency with respect to the sampling frequency. For this example $W_o = 0.0967$. Only $(2M+1) = 3$ samples were assumed to be known; $M = 1$ and N(total number of samples) is 31. The known part of the signal as well as the original signal are shown in fig. 4.6. The extrapolation matrix $[E_\infty]$ was generated using a bandpass filter matrix whose upper and lower cutoff frequencies, β_2 and β_1 are equal to 3 and 2 respectively. The extrapolated signal is shown in fig. 4.6. (exact agreement between the original and the extrapolated signals). This example has been carried out for several values of M i.e. M = 1,2,3 and in all cases exact results were obtained.

(2) Example 2 : Two sine wave signals:

This example is similar to the previous example, except we have considered a signal consiting of 2 sine waves added together i.e.,

$$f(t) = \sin(\omega_1 t) + \sin(\omega_2 t) \tag{4.63}$$

or in sampled form

$$f(i) = \sin(2\pi W_1 i) + \sin(2\pi W_2 i) \tag{4.64}$$

where W_1 and W_2 are the normalized signal frequencies with respect to sampling frequency. $W_1 = 0.0967$ and $W_2 = 0.129$. The number of samples assumed to be known is $(2M+1) = 5$, i.e. $M = 2$ and $N = 31$.

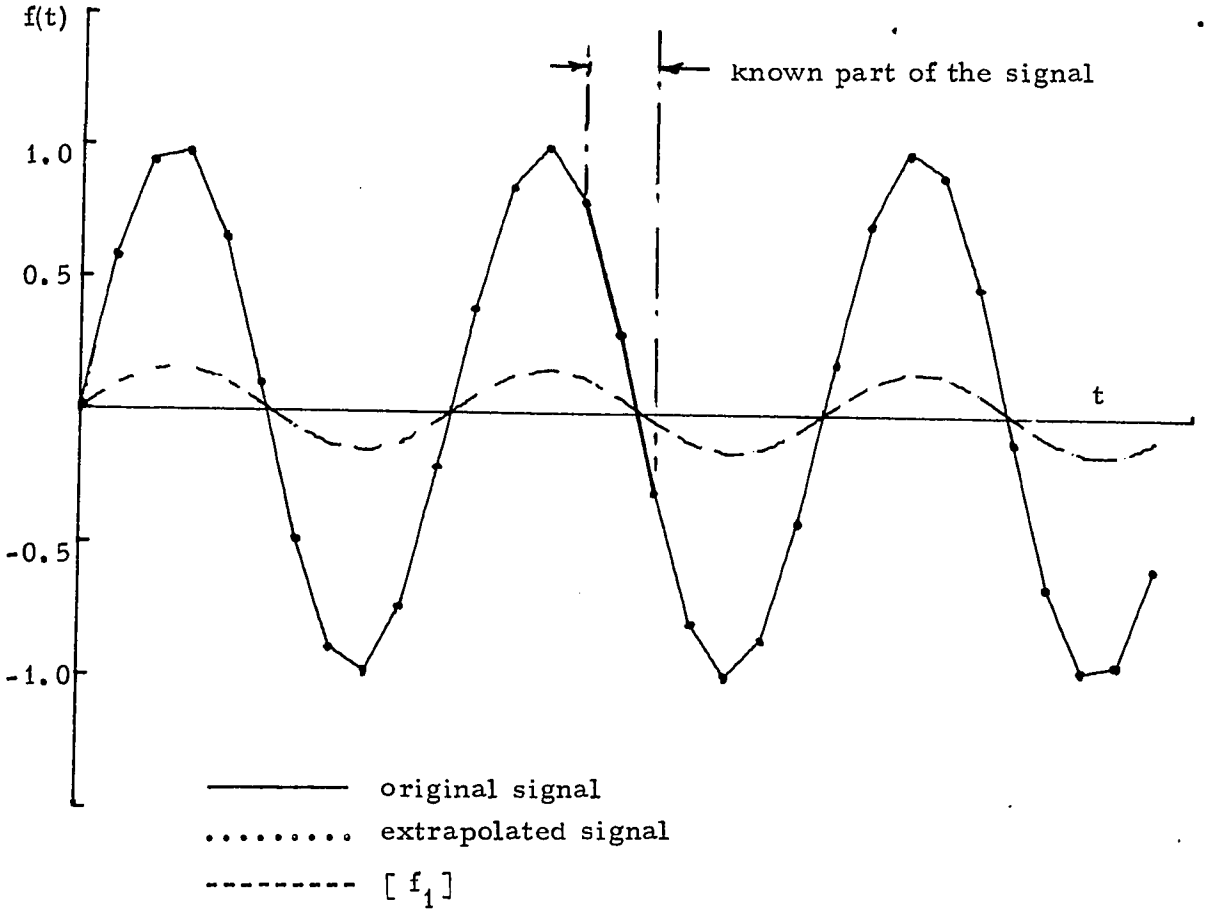


Fig. 4.6 Example 1 : Extrapolation of a sine wave signal using $[E_\infty]$. N = 31, M = 1, $\beta_1 = 2$, and $\beta_2 = 3$. The heavy solid line shows the known part of the signal.

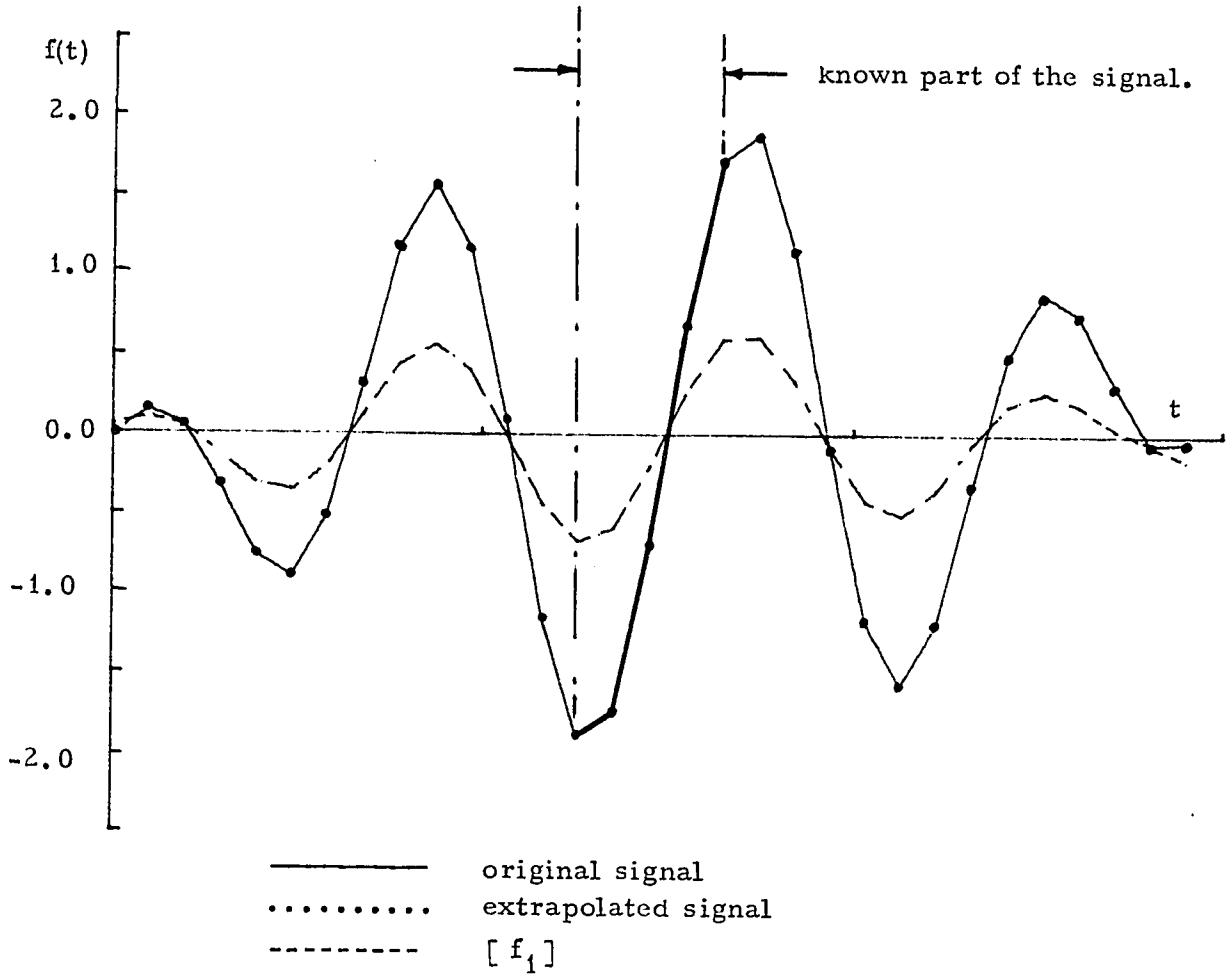


Fig. 4.7 Example 2 : Extrapolation of 2 sine waves having the same amplitude.
N = 31, M = 2, $\beta_1 = 2$, $\beta_2 = 4$.

The extrapolation matrix $[E_\infty]$ was generated using a bandpass filter matrix whose cutoff frequencies β_2 and β_1 are 4 and 2 respectively. Fig. 4.7 shows the original signal, known part of the original signal, and the extrapolated signal. This example is similar to the one presented by Gerchberg [5]. While Gerchberg obtained, after 240 iterations, an approximation to the original signal. We have obtained the exact signal using single matrix operation.

(3) Example 3: Two sine wave signals with different amplitude:

The signal f(t) is of the form :

$$f(t) = 0.5 \sin(\omega_1 t) + \sin(\omega_2 t) \qquad (4.65)$$

which is the same as the previous example except one sine wave signal has half the amplitude of the other. Same procedure was carried out, i.e. N = 31, M = 2, $\beta_1 = 2$, and $\beta_2 = 4$. Exact results were obtained. The results are shown in fig. 4.8.

(4) Example 4:

The signal to be extrapolated is of the form :

$$f(t) = \sum_{\ell=1}^{3} a_\ell \sin(\omega_\ell t) \qquad (4.66)$$

or in sampled form

$$f(i) = \sum_{\ell=1}^{3} a_\ell \sin(2\pi W_\ell i) \qquad (4.67)$$

where $a_1 = 1.0$, $a_2 = 0.25$, and $a_3 = 0.75$

and $W_1 = 0.121$, $W_2 = 0.151$, and $W_3 = 0.181$

The number of known samples (2M+1) = 9 i.e. M = 4, The extrapolation matrix $[E_\infty]$ was generated for N = 33, $\beta_1 = 3$ and $\beta_3 = 6$. Fig. 4.9 shows the known part of the signal, original signal, and the extrapolated signal. Exact agreement between the original and the extrapolated signals is obtained.

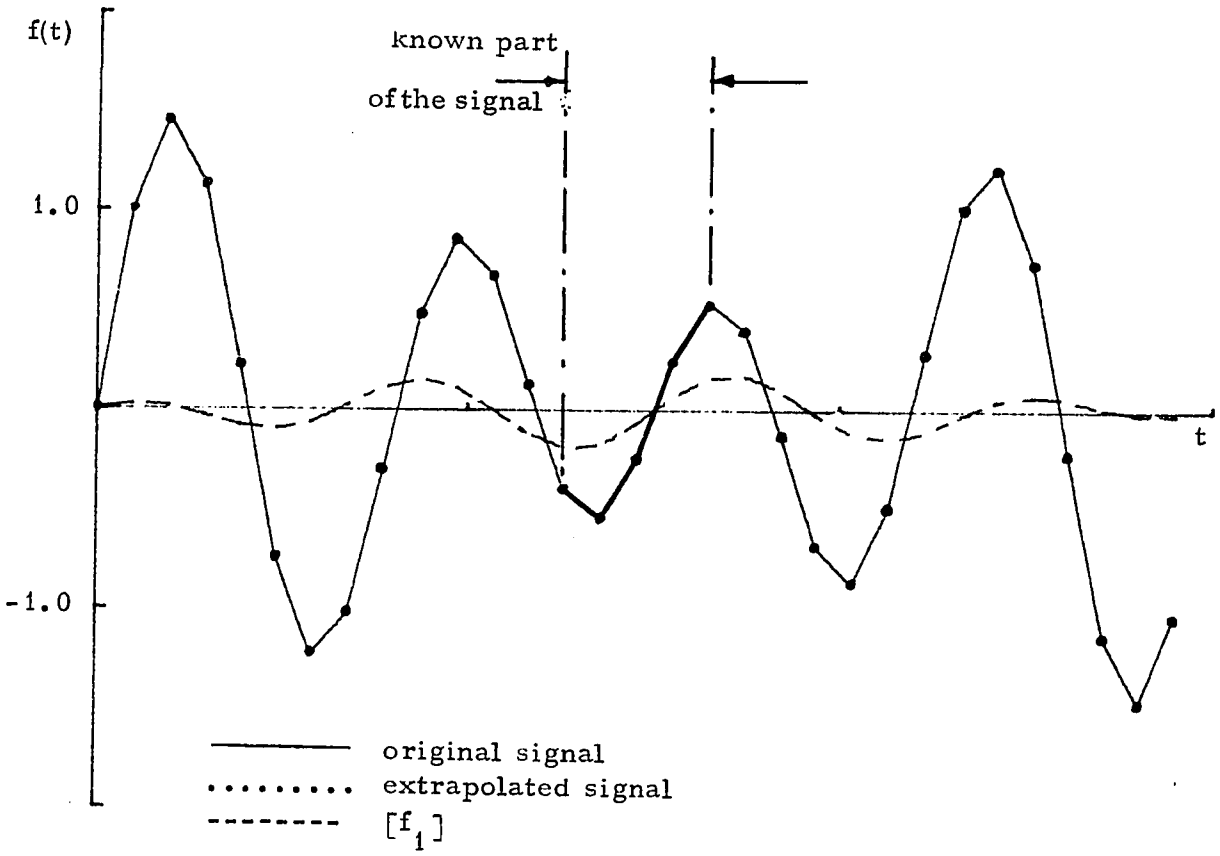


Fig. 4.8 Example 3: Extrapolation of 2 sine wave signals with different amplitudes. N = 31 M = 2, $\beta_1 = 2$, and $\beta_2 = 4$.

(5) Example 5 :

The signal f(t) is of the form :

$$f(t) = \frac{\sin \omega_o t}{\omega_o t} \tag{4.68}$$

or in sampled form :

$$f(i) = \frac{\sin 2\pi W_o i}{2\pi W_o i} \tag{4.69}$$

where $W_o = \frac{f_o}{f_s}$, f_o is the signal bandlimit (cutoff frequency) $= \frac{\omega_o}{2\pi}$, f_s is the sampling frequency. Fig. 4.10 shows the original signal for $W_o = 0.0427$, known part of the signal and the extrapolated signal for N = 199, $\beta = 8$ and M = 5. The extrapolation matrix $[E_{25}]$, i.e. n = 25, was used to extrapolate the signal. Fig. 4.11 shows the same as fig. 4.10 except the extrapolation matrix $[E_{100}]$, i.e. n = 100, was used instead of $[E_{25}]$. We note that the accuracy of extrapolation i.e. approximation to the original signal improved as we increased n.

4.10. REALIZATION OF A FAST DIGITAL EXTRAPOLATOR BASED ON THE EXTRAPOLATION MATRIX .

It is clear that the extrapolation matrix $[E_\infty]$ or $[E_n]$ is a function of the filter matrix cutoff frequency $W_c = \frac{\beta}{N}$ and the number of known samples. One might be led to the conclusion that for each class of band limited signals having the same frequency band limits, an extrapolation matrix has to be generated. While this conclusion is valid when using fixed sampling frequency, we can avoid this situation by using variable sampling frequency as we shall illustrate later.

By varying the sampling frequency we are effectively varying the normalized filter cutoff frequency and hence we can adjust that to

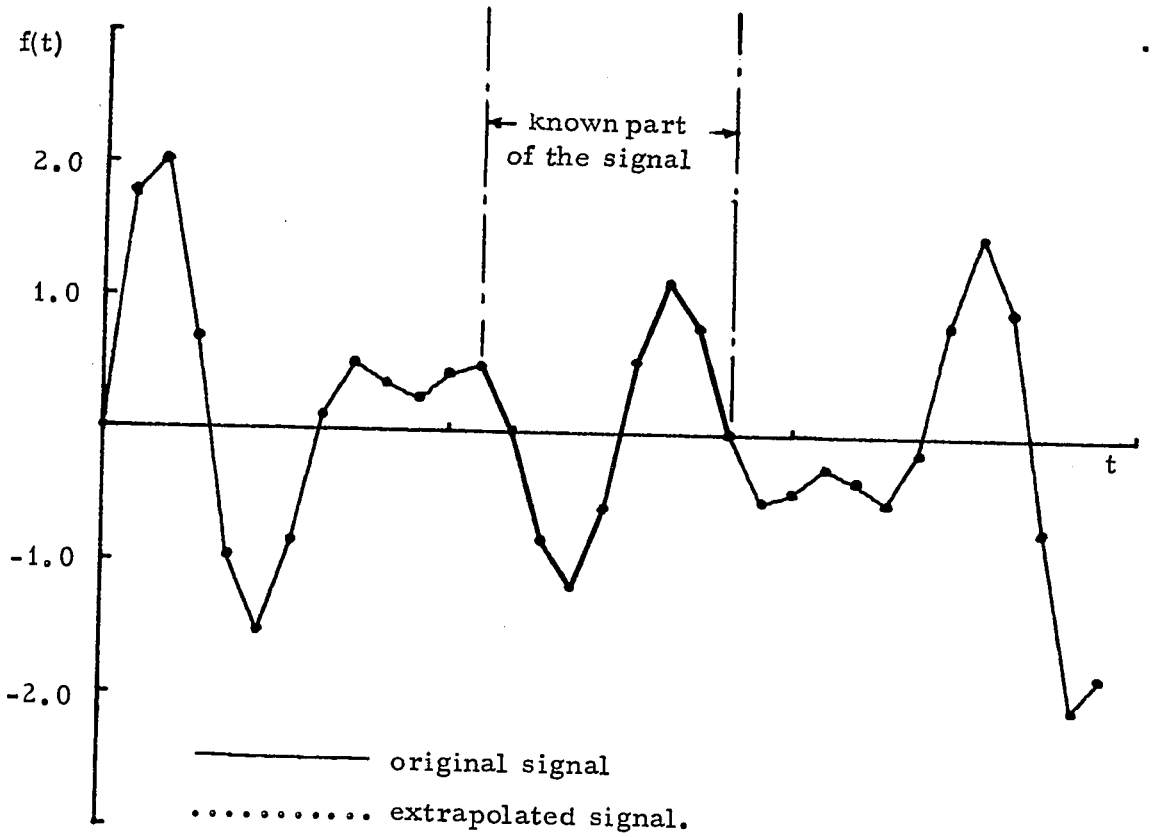


Fig. 4.9 Example 4 : Extrapolation of 3 sine wave signals having different amplitudes.
N = 33, M = 4, $\beta_1 = 3$, and $\beta_2 = 6$.

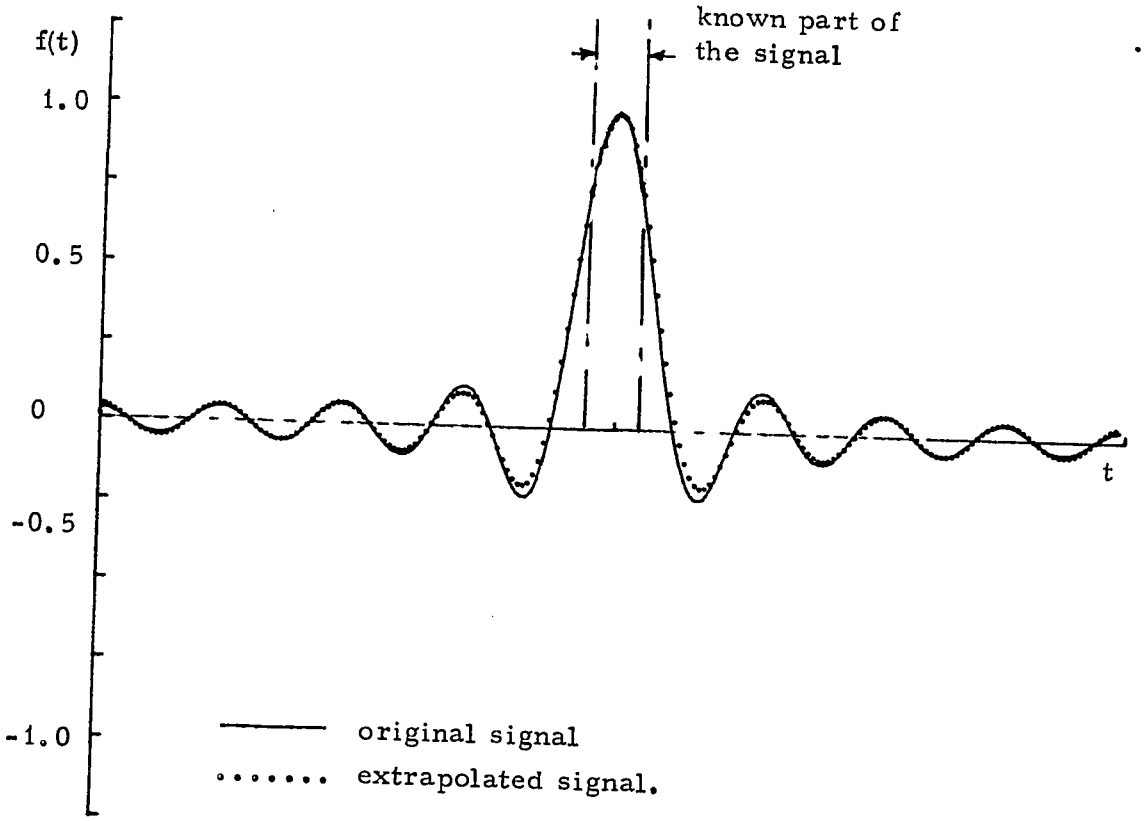


Fig. 4.10 Example 5 : Extrapolation of a sinc function using $[E_{25}]$. N = 199, M = 5, and $\beta = 8$.

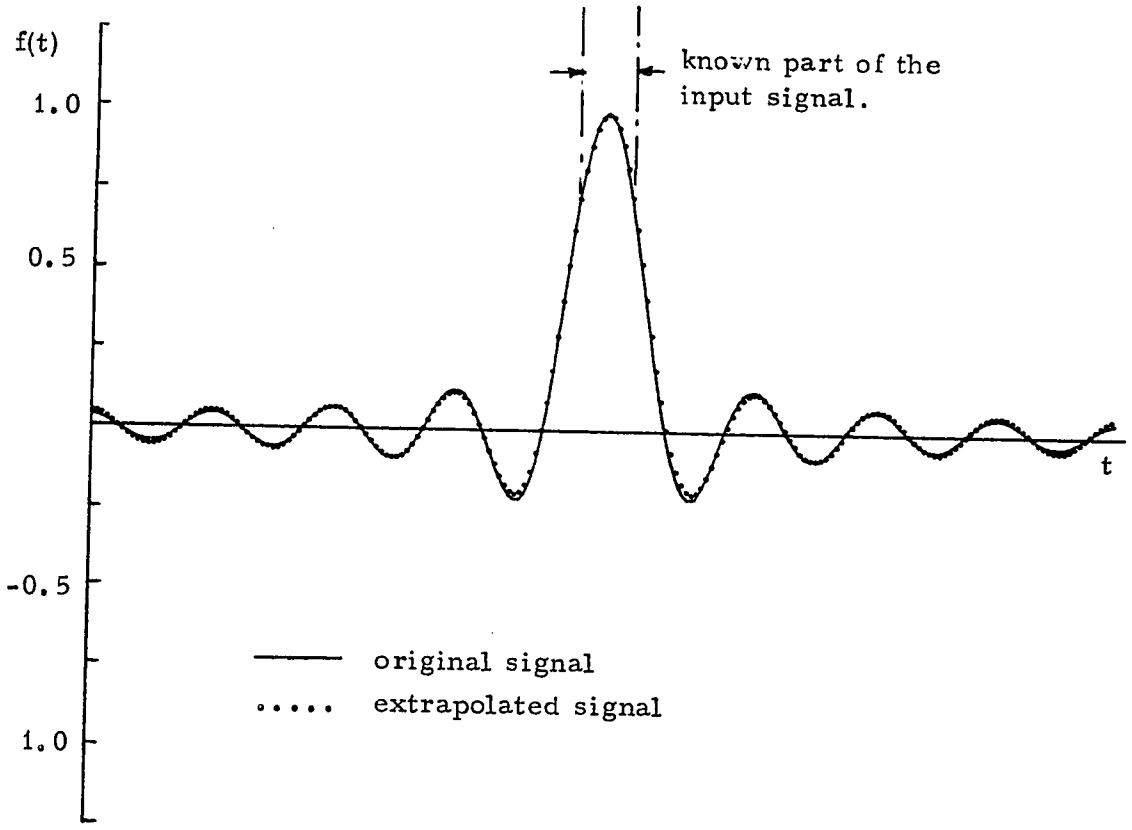


Fig. 4.11 Example 5 : Extrapolation of a sinc function using $[E_{100}]$. N = 199, M = 5, and β = 8.

coincide with the signal frequency band limit. Thus for a certain extrapolation matrix a wide class of band limited signals can be extrapolated using the same matrix. Also for a certain extrapolation matrix the number of known samples is fixed, this would form no problem because even if the number of known samples exceeded (2M+1), one would pick only (2M+1) samples and discard the rest. Of course, this would form limits on the signal duration bandwidth product using this specific extrapolation matrix.

The lower and upper limits on the signal duration bandwidth product would determine the range for which a certain extrapolation matrix can be used i.e. the minimum and maximum duration of the known part of the signal to be extrapolated also the minimum and maximum bandwidth.

Let us assume that we have already decided on certain extrapolation matrix i.e. determined N, M and $W_c = \frac{\beta}{N}$. The submatrix $[\bar{E}_\infty]$ (or $[\bar{E}_n]$) is generated and its coefficients are stored. Let the duration of the known portion of the signal be t_o and the signal is band limited to $|B|$ Hz.

The sampling frequency is adjusted so that :

$$f_s = \frac{B}{W_c} \tag{4.70}$$

Thus if the signal duration is t_o, then the number of known samples, corresponding to this duration is :

$$\frac{t_o}{T} = t_o f_s = \frac{t_o B}{W_c} \tag{4.71}$$

This should be greater or equal to (M+1), and of course less than $(\frac{N-1}{2})$.

Thus:

$$(\frac{N-1}{2}) > \frac{t_o B}{W_c} \geq (M+1) \tag{4.72}$$

or

$$W_c(\frac{N-1}{2}) > t_o B \geq W_c(M+1) \tag{4.73}$$

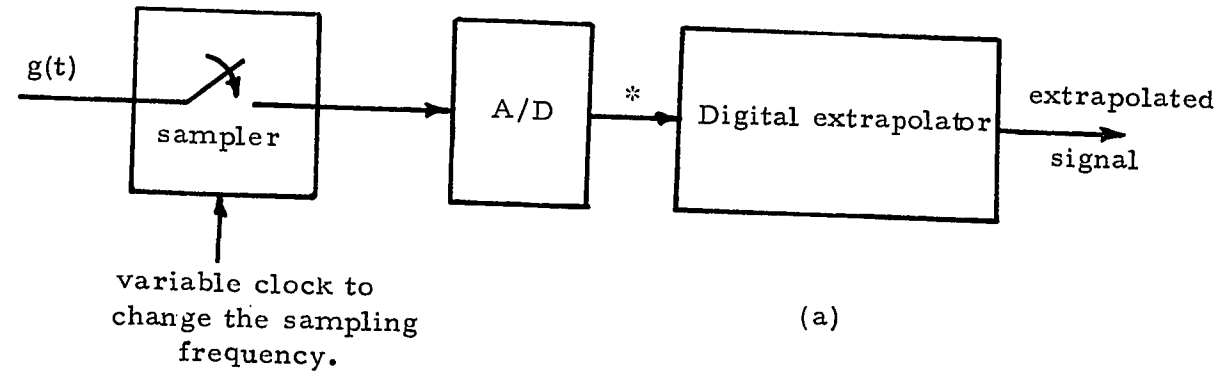
Relation (4.73)gives the lower and upper limits of signal duration bandwidth product. For example if $N = 501$ and $M = 9$ and $W_c = 0.1$. For signals of 1 m. sec duration, the min and max signal bandwidths B is

$$1 \text{ kHz} < B < 25 \text{ kHz} \tag{4.74}$$

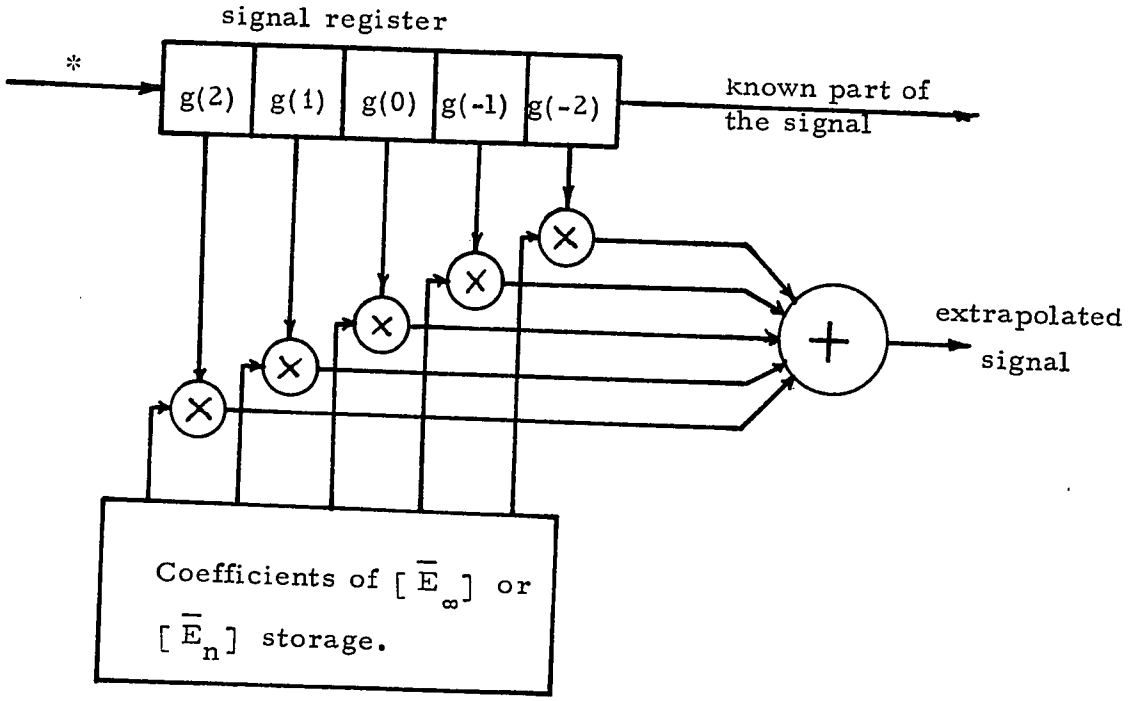
or alternatively for signals of 1 kHz bandwidth B the min and max duration t_o is

$$1 \text{ m sec} < t_o < 25 \text{ m sec.} \tag{4.75}$$

The above discussion leads us to propose a fixed realization of a wide range fast digital extrapolator. The proposed realization is shown in fig. 4.12. It consists of a variable frequency sampler, analog to digital converter and the digital extrapolator which performs multiplication and addition. The coefficients of the submatrix $[\bar{E}_\infty]$ (or $[\bar{E}_n]$) are stored in coefficient storage memory as shown. The sampling frequency is adjusted according to relation (4.70). The known part of the signal in digital form is fed to the signal register and then fed to the multipliers. The coefficients of the extrapolation matrix ($[\bar{E}_\infty]$ or $[\bar{E}_n]$) are fed to the multipliers and the results are added to form the extrapolated signal. The maximum number of multipliers is (2M+1), and can be reduced even to one multiplier depending on how fast one would require to process the signal. In the coefficient storage memory more than one extrapolation matrix set of coefficients can be stored to allow for a wider operation range.



(a)



(b)

Fig. 4.12 Proposed realization of a fast digital extrapolator. (a) Overall system

(b) Details of the digital extrapolator.

4.11. CONCLUSION

In this chapter a fast bandlimited signal extrapolation technique has been presented. The total extrapolation process is achieved by a single matrix operation. The proposed technique has three basic advantages, first, it requires less computation than the iterative extrapolation, secondly it can be applied in real time, and finaly it gives exact extrapolation results when the matrix $[E_\infty]$ is used. It would be almost impossible to obtain exact results using the iterative technique since that would simply mean infinite number of computations.

Actual hardware implementation of the direct method has been proposed which allows for adjustments to any signal bandwidth.

REFERENCES

[1] T.S. Huang, W.F. Schneiber, and O.J. Tretiak, "Image Processing", Proc. IEEE, Vol. 59, pp 1586-1609, Nov. 1971.

[2] P.A. Jansson, R.H. Hunt, and E.K. Plyler, "Resolution Enchancement of Spectra", J. Opt. Soc. Amer., Vol. 60, pp 596-599, May 1970.

[3] A. Papoulis, "Minimum Bias Windows for High Resoltuion Spectral Estimates", IEEE Trans. Information Theory, Vol. IT-19, pp 9-12, Jan. 1973.

[4] A. Papoulis, "A New Algorithm in Spectral Analysis and Band-Limited Extrapolation", IEEE Trans. on Circuits and Systems, Vol. CAS-22, No. 9, September 1975, pp 735-742.

[5] R.W. Gerchberg, "Super Resolution Through Error Energy Reduction", Optica Acta, Vol. 21, No. 9, pp 709-720, 1974.

[6] T.P. Burg, "Maximum Entropy Spectral Analysis", presented at the 37th Int. SEG Meeting, Oct. 1967.

[7] D. Slepian, H.O. Pollak, and H.J. Landolt, "Prolate Spheroidal Wave Functions", Bell Syst. Tech. J., Vol. 40, pp. 43-84, Jan. 1961.

[8] T.J. Ulrych, "Maximum Entropy Power Spectrum of Truncated Sinusoids", J. Geophys. Res., Vol 77, pp 1396-1400, 1972.

[9] J.A. Edwards and M.M. Fitelson, "Notes on Maximum-Entropy Processing", IEEE Trans. Inform. Theory, Vol IT-19, pp 232-234, March 1973.

[10] D.T. Finkbeiner, "Matrices and Linear Transformations", D.B. Traporevala Sons and Co. Ltd., Bombay, pp 237-249.

[11] L. Fox, "Numerical Linear Algebra", Clarence Press, Oxford, pp 48-51.

CHAPTER FIVE

DIRECT CONVOLUTIONAL DHT FILTERS

5.1. INTRODUCTION

For the discrete filters presented in chapter two, in order to obtain the ideal filtering characteristics, the input signal (sequence) has to be of finite number of samples or generally periodic.

In several filtering applications the input signal is of infinite duration and not readily separable to sequences of length N samples. In this case a direct convolutional filtering process is required.

In this chapter it will be shown that the DHT filtering approach can be used to generate filter designs using the middle row of the Hilbert or any of the filter matrices presented in chapter two. These filters are realizable in the form of FIR (non recursive) linear phase digital or sampled data filters. In this way almost any filtering characteristic can be realized using the DHT approach.

Several design examples are given and the properties of these filters are discussed . Also analysis of the resulting approximation error and comparison between this approach and other approaches [1,2,3] are presented.

We note that though the filters presented here are based on the Hilbert transform, the process of modulation or demodulation has been eliminated and all filters are realizable in total directly as FIR filters.

5.2 FIR LINEAR PHASE FILTER DESIGNS USING THE DHT

The Hilbert or any of the filter matrices presented in chapter two can be generally expressed in the form :

$$[F] = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,N-1} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,N-1} \\ p_{2,0} & p_{2,1} & p_{2,2} & \cdots & \\ \vdots & \vdots & \vdots & & \vdots \\ p_{r-1,0} & p_{r-1,1} & p_{r-1,2} & \cdots & p_{r-1,N-1} \\ p_{r,0} & p_{r,1} & p_{r,2} & \cdots & p_{r,N-1} \\ \vdots & \vdots & \vdots & & \\ p_{N-1,0} & p_{N-1,1} & p_{N-1,2} & \cdots & p_{N-1,N-1} \end{bmatrix} \qquad (5.1)$$

where $$r = \begin{cases} \frac{N+1}{2} & N \text{ odd} \\ \frac{N}{2} & N \text{ even} \end{cases} \qquad (5.2)$$

The properties of this matrix havebeen discussed in chapter two and the coefficients were given.

Each row of the matrix when used as the coefficients of an FIR digital or sampled data filter , gives an approximation to the desired frequency response. No matter which row is used, the resulting frequency response would be in exact agreement with the desired response at the discrete frequency points W_k given by :

$$W_k = \frac{\omega_k}{\omega_s} = \frac{k}{N}, \qquad k = 0, 1, 2, \ldots, (N-1), \qquad (5.3)$$

where W_k is the normalized discrete frequency with respect to the sampling frequency, and ω_s is the radian sampling frequency.

In between these discrete frequency points, the resulting frequency response exhibits some ripples . The amount of these ripples (deviation from desired response) depends on which row is used. The frequency response $H_i(W)$ corresponding to the i^{th} row of the filter matrix is :

$$H_i(W) = \sum_{n=o}^{N-1} p_{i,n}\, e^{-j2\pi W n} \qquad (5.4)$$

where W is the normalized frequency and $p_{i,n}$ is the n^{th} element of the i^{th} row. Fig. 5.1 shows the different responses obtained for a lowpass filter when using different rows of the lowpass matrix. It is clear that the middle row of the matrix gives the best approximation to the desired characteristic i.e. minimum ripples as compared to other rows. In addition when the elements of the middle row are used as the FIR filter coefficients a linear phase design results. This shall be illustrated later on .

In fact, for N even, there are two middle rows i.e. $(\frac{N}{2} - 1)^{th}$ and $(\frac{N}{2})^{th}$ which gives identical magnitude response but have different linear phase responses. For example ; for the Hilbert matrix the elements $p_{i,n}$ of (5.1) are given by :

$$p_{i,n} = \begin{cases} \frac{2}{N} \cot \frac{\pi}{N}(i-n) & (i-n) \text{ odd} \\ 0 & (i-n) \text{ even and } i=n \end{cases} \qquad (5.5)$$

Thus the two middle rows of the Hilbert matrix are given by

$$h_1(n) = \begin{cases} \frac{2}{N} \cot\left[\frac{\pi}{N}\left(\frac{N}{2} - n - 1\right)\right] & \left(\frac{N}{2} - n - 1\right) \text{ odd} \\ 0 & \left(\frac{N}{2} - n - 1\right) \text{ even} \end{cases} \qquad (5.6)$$

$$h_2(n) = \begin{cases} \frac{2}{N} \cot\left[\frac{\pi}{N}\left(\frac{N}{2} - n\right)\right] & \left(\frac{N}{2} - n\right) \text{ odd} \\ 0 & \left(\frac{N}{2} - n\right) \text{ even} \end{cases} \qquad (5.7)$$

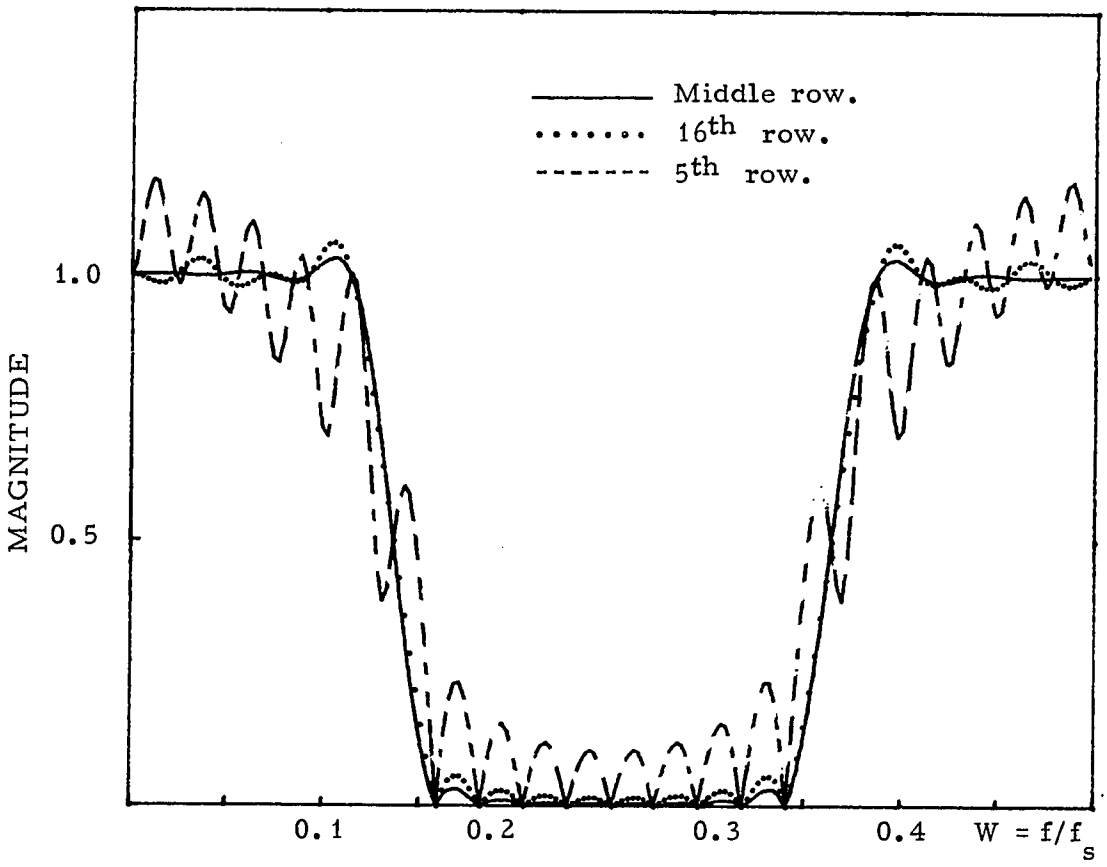


Fig. 5.1. Frequency responses of FIR lowpass filters when different rows of the lowpass matrix are used as filter coefficients . N = 44, M = 22 and β = 6. (M is the number of nonzero filter coefficient)

where $n = 0, 1, 2, \ldots\ldots (N-1)$, and $h_1(n)$ or $h_2(n)$ is the impulse response of the FIR Hilbert transform. As an illustrative example let us consider the case of the Hilbert matrix for N = 8, the middle two rows are :

$$[\ -a_3 \quad 0 \quad -a_1 \quad 0 \quad a_1 \quad 0 \quad a_3 \quad 0\ ] \qquad (5.8)$$

$$[\ 0 \quad -a_3 \quad 0 \quad -a_1 \quad 0 \quad a_1 \quad 0 \quad a_3\ ] \qquad (5.9)$$

where $a_1 = \cot \frac{\pi}{N}$, $a_3 = \cot \frac{3\pi}{N}$.

The last element in (5-8) which is zero has no effect on either the magnitude or the phase responses while the first element in (5.9) has no effect on the magnitude response but adds a linear phase term of $e^{-j\omega T}$ to the phase response and both can be dropped. The final result is that we have an FIR Hilbert transform with odd length and satisfies the conditions for linear phase [4] i.e. the impulse response is antisymmetrical.

5.2.1 Change of Symmetry of the Magnitude Response :

As we have mentioned in chapter two, for filter designs based on the DHT, the magnitude response has symmetry around $f_s/4$. For N even, this symmetry can be changed to $f_s/2$ (instead of $f_s/4$) as in conventional digital filters by simply dropping the zero elements form the corresponding filter matrix and consequently from the middle row of the matrix. While the above statement is true for any filter matrix (N even) whose diagonal elements are zeroes (e.g. lowpass and bandpass matrices), it is not sufficient for filters whose matrices have nonzero diagonal elements (e.g. highpass and bandstop filters). For such filters symmetry around $f_s/2$ is obtained with one of the following two methods . First we can drop the zero terms from the middle row of

the matrix and incorporate a half period delays i.e. $z^{-1/2}$ as shown in fig. 5.2. Or alternatively we can add a delay line in parallel with the lowpass filter or bandpass filter as shown in fig. 5.3. The delay line will act as an allpass linear phase network which when connected in parallel with the lowpass filter results in a highpass filter characteristic.

5.2.2. Properties of Filter Designs Based on the DHT:

In order to illustrate the properties of the FIR filter designs using the DHT i.e. the middle row of the Hilbert or any of the filter matrices, several design examples are first considered:

1 - FIR Hilbert Transformers :

The middle row of the Hilbert matrix was generated for several values of N even , the resulting designs are of odd length M. The frequency responses of different Hilbert transformers are shown in fig. 5.4 for N = 32 and M = 31, fig. 5.15 for N = 20 and M = 19, and fig. 5.19 for N = 80 and M = 79. We note that all the resulting designs are of M^{th} order and the number of nonzero filter coefficients are $\frac{N}{2}$. i.e. in actual realization only $\frac{N}{4}$ multipliers needed.

2 - FIR Lowpass Filters :

The middle row of the several lowpass matrices was generated for several values of N. Fig. 5.5 shows the resulting frequency responses of lowpass filters having the same cutoff frequency but different orders, M = 8, 16, and 32. The symmetry of the resulting characteristics has been changed to $f_s/2$ instead of $f_s/4$. In this way, only N/4 multipliers are required. Table 5.1 gives the filter coefficients of the lowpass filter designs shown in fig. 5.5. The frequency responses of other lowpass filter designs are shown in fig. 5.1 and fig. 5.17 .

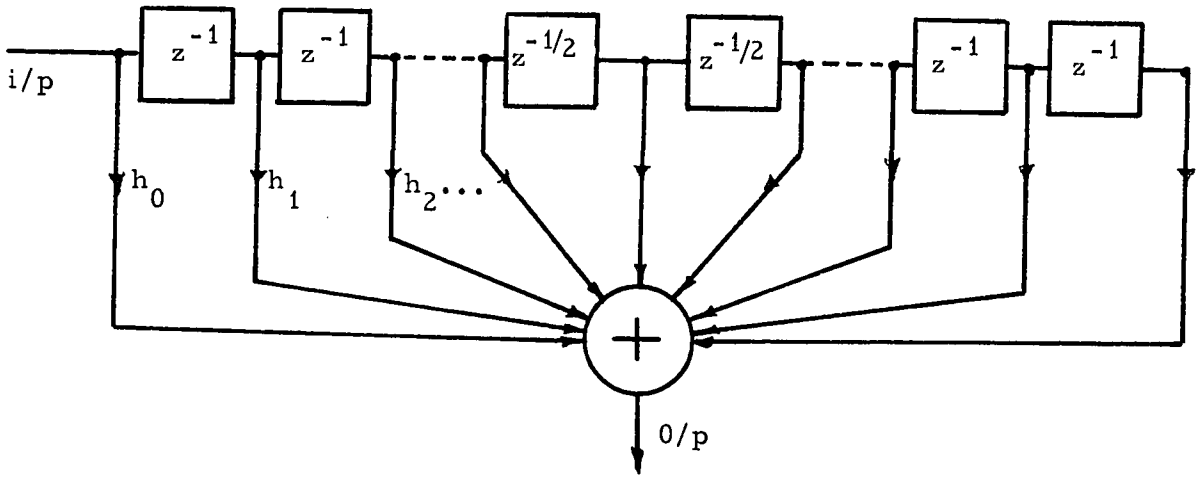


Fig. 5.2. Realization of highpass or bandstop filters with symmetry around $f_s/2$ using half sample delay elements.

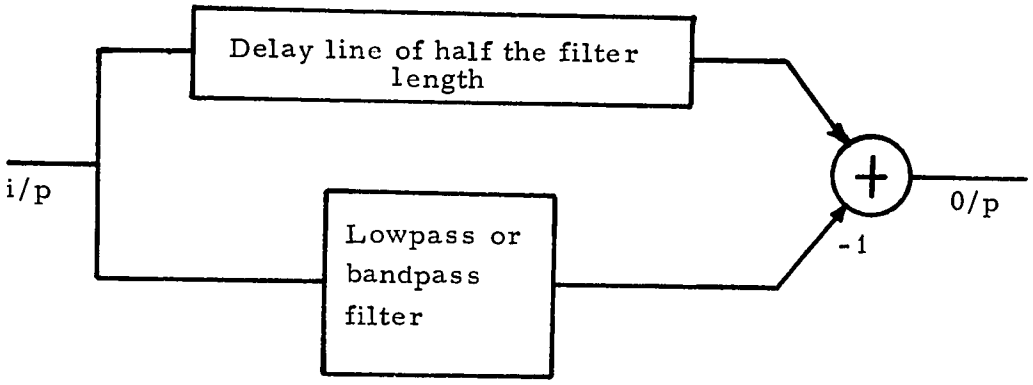


Fig. 5.3. Realization of highpass or bandstop filters using a parallel connection of a delay line and a lowpass or bandpass filter.

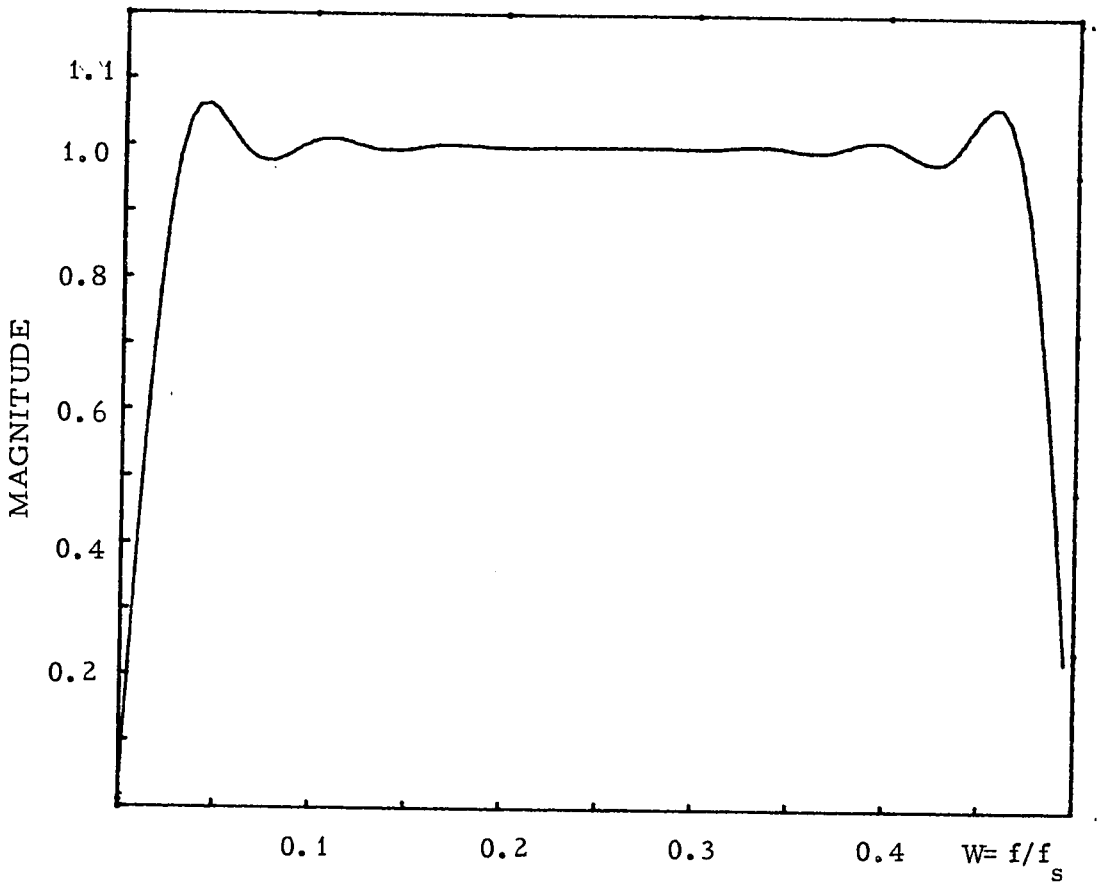


Fig. 5.4. Frequency response of a Hilbert transformer N = 32, M = 31.

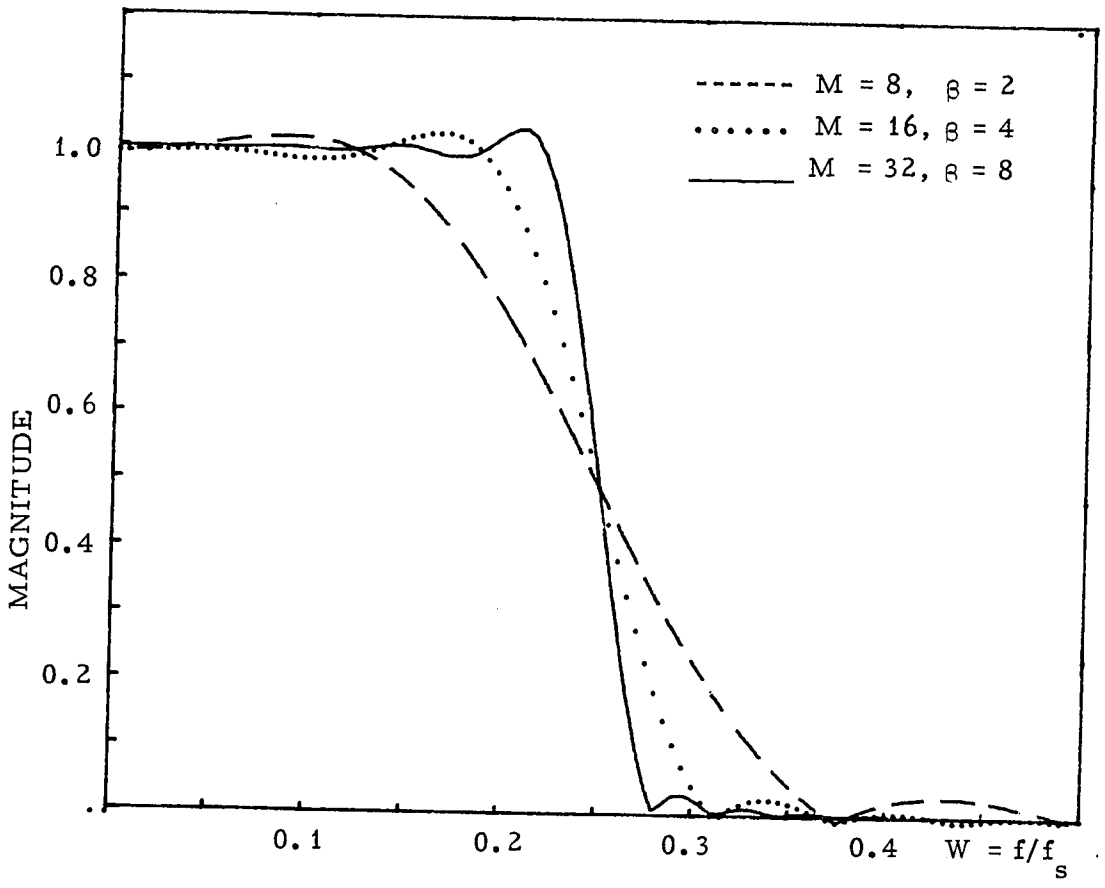


Fig. 5.5. Magnitude responses of FIR lowpass filters having different orders but the same cutoff frequency.

(M is the filter order)

3 - FIR Bandpass Filters :

Two examples are considered. Fig. 5.6 shows the frequency response of a typical bandpass filter for N = 80 , M = 40, β_1 = 8 and β_2=15. In fig. 5.7 the frequency response of a narrow bandpass filter is shown for N = 200, M = 100, β_1 = 23, and β_2 = 25. For both filters the symmetry has been changed to $f_s/2$. Both filters coefficients are given in table 5.2 and table 5.3.

4 - FIR Multiple Passband Filters:

The middle rows of two multiple passband filter matrices were generated as shown in section 2.4.1. The frequency response of a two passband filter is shown in fig. 5.8, where the two passbands have the same relative magnitude. Fig. 5.9 shows a two passband filter where the second passband is 12 db below the first one. For both designs the symmetry has been changed to $f_s/2$. The coefficients of both filter designs are given in table 5.4.

5 - FIR Highpass Filter :

Fig. 5.10 shows the resulting frequency response of an FIR highpass filter when the middle row of the highpass filter matrix is used as the filter coefficients. The symmetry has been changed to $f_s/2$ as was explained in section 5.2.1.

6 - FIR Bandstop Filters :

Two examples are given, in fig. 5.11 a typical bandstop filter frequency response is shown . Fig. 5.12 shows the frequency response of a narrow bandstop filter (notch) filter. The ripples in the passbands are shown in fig. 5.13.

7 - FIR Linear Magnitude Filter:

The linear magnitude filter matrix was generated as shown in section 2.4.2. The middle row of the matrix was used as the coefficients of the FIR filter. Fig. 5.14 shows the frequency response of a linear magnitude filter response for N = 64, M = 33. The filter coefficients are given in table 5.5. The slope of the linear magnitude response can be changed by simply multiplying the filter coefficients by an arbitrary constant.

From the above design examples, we note that FIR filter designs using the DHT have the following properties :

(i) The phase response is linear since the impulse response is symmetrical or antisymmetrical.

(ii) The width of the transition band(s) is inversely proportional to the filter order.

(iii) For frequency selective filters (e.g. lowpass, bandpass .....etc.) the magnitude response is antisymmetrical around the cutoff frequency point i.e. 6 db point. This can be seen from fig. 5.5. This property is quite essential in the design of channel bank filters (for example for speech analysis [8]), where it is required that the composite filter response (sum of individual filter responses) be flat over the whole frequency band to allow for reconstruction of the signal.

(b) The filter coefficients are completely defined in closed form expressions.

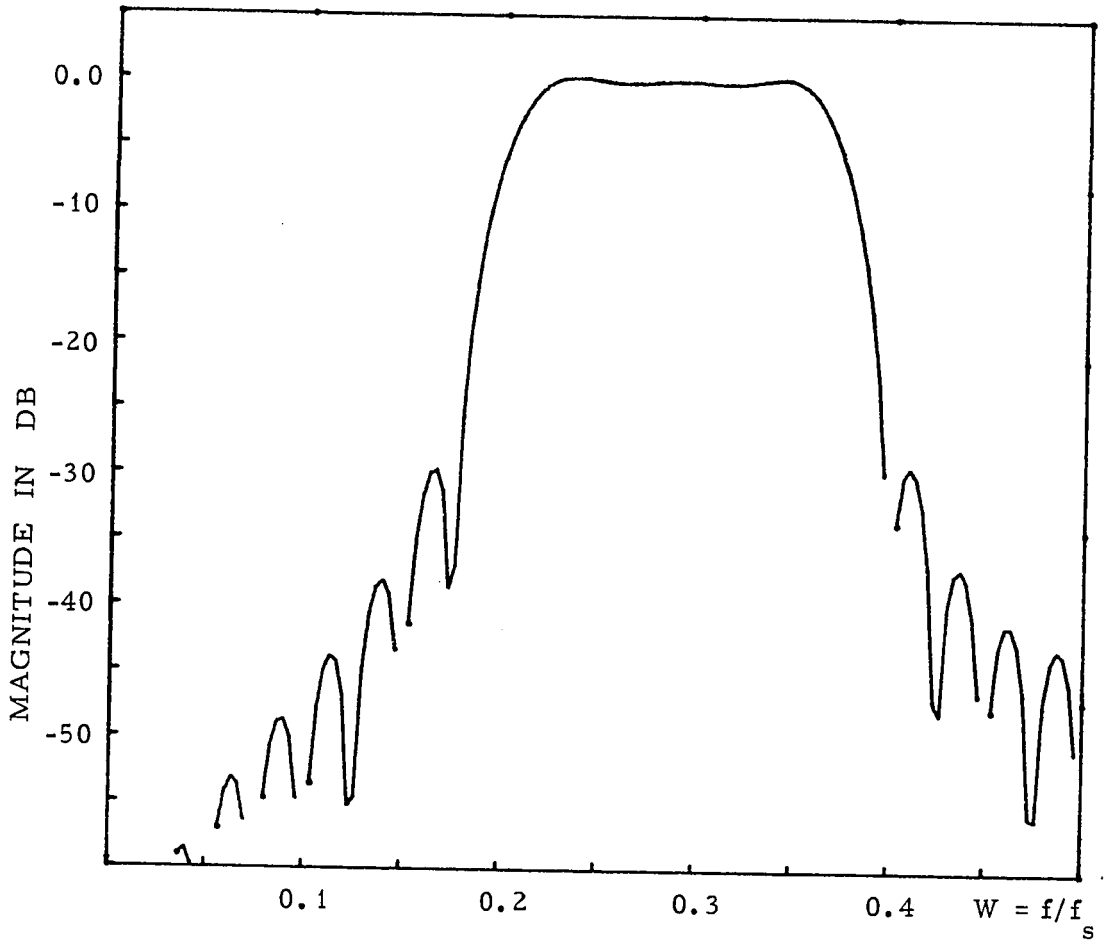


Fig. 5.6. Magnitude response of a bandpass filter.
M = 40, N = 80, $\beta_1 = 8$ and $\beta_2 = 15$
(M is the filter order)

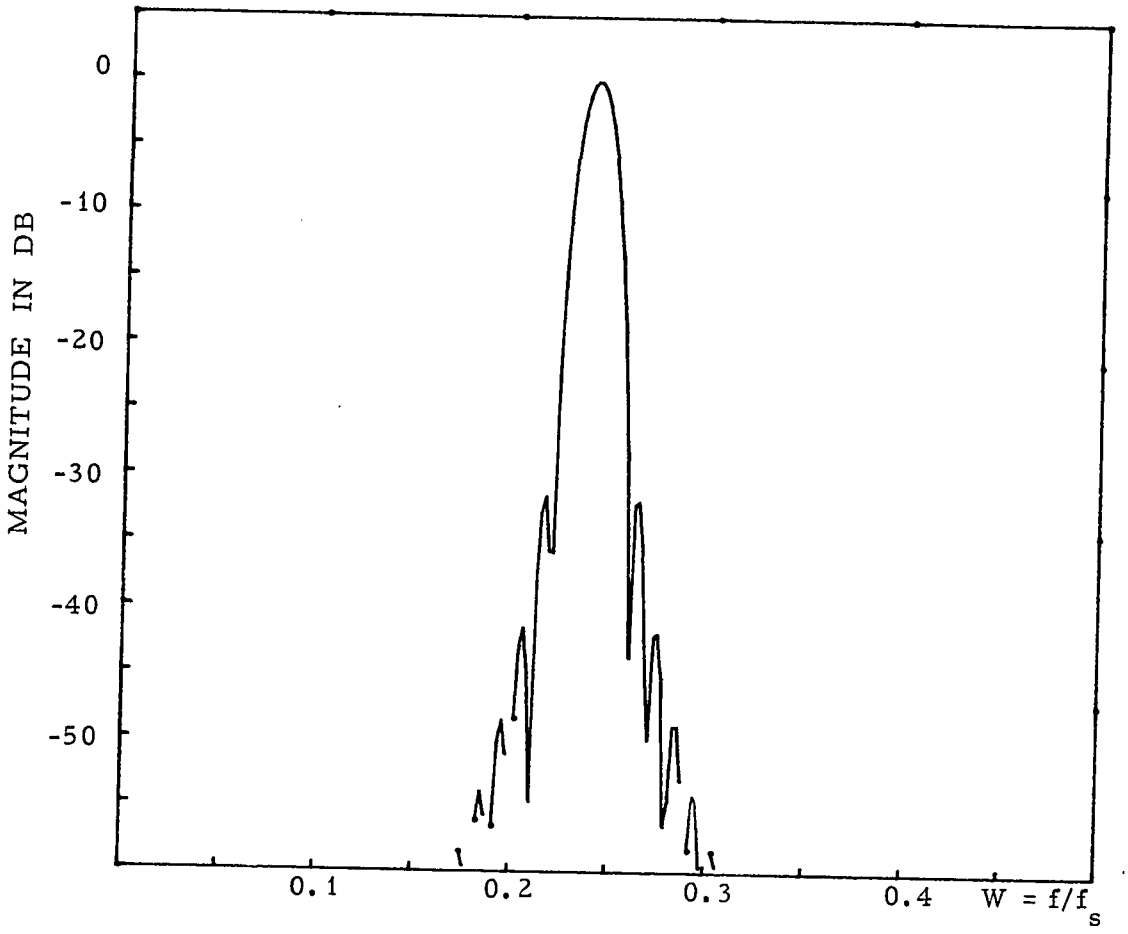


Fig. 5.7. Narrow bandpass filter, M = 100, N = 200
$\theta_1 = 23$, and $\theta_2 = 25$.

(M is the filter order)

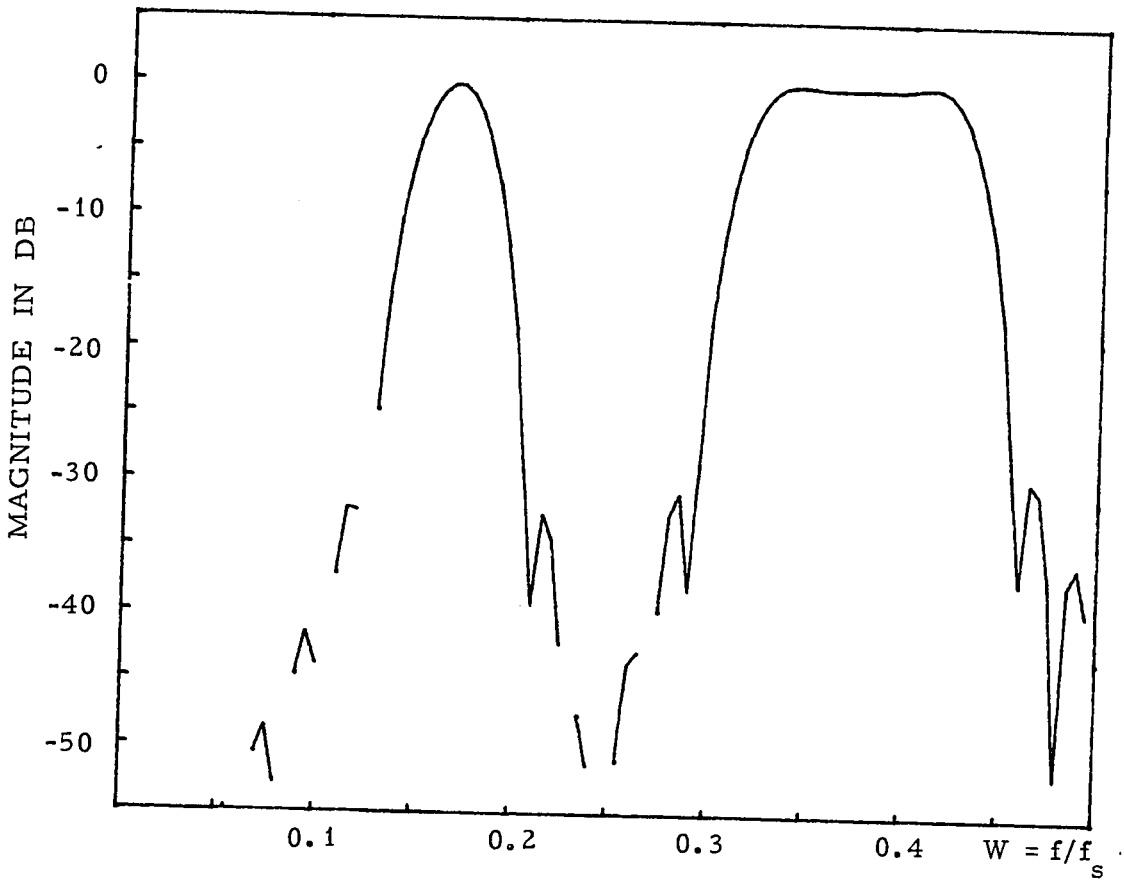


Fig. 5.8. Multiple passband filter: M = 48, N = 96.
for the first passband: $\beta_1 = 7$, $\beta_2 = 9$
for the second passband: $\beta_1 = 15$, $\beta_2 = 21$.
(M is the filter order)

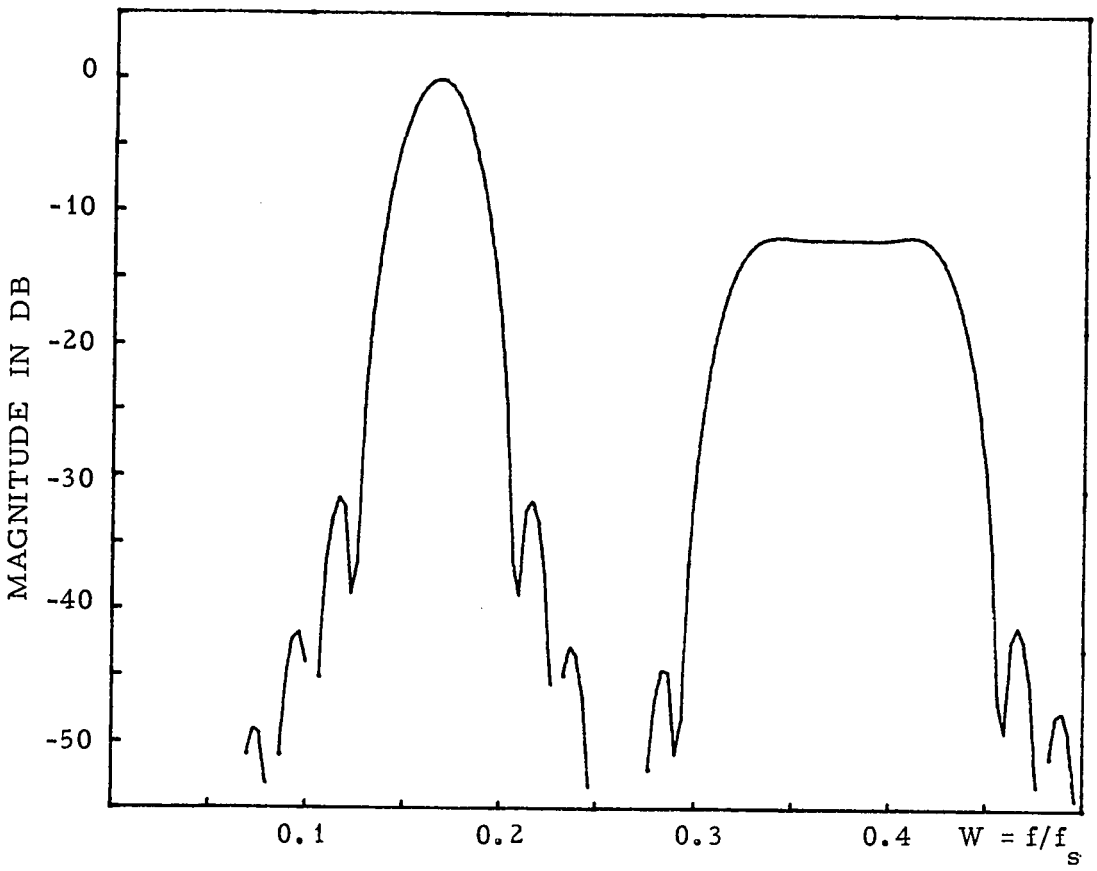


Fig. 5.9. Multiple passband filter, same as fig. 5.8 except the second passband is suppressed 12 db down.

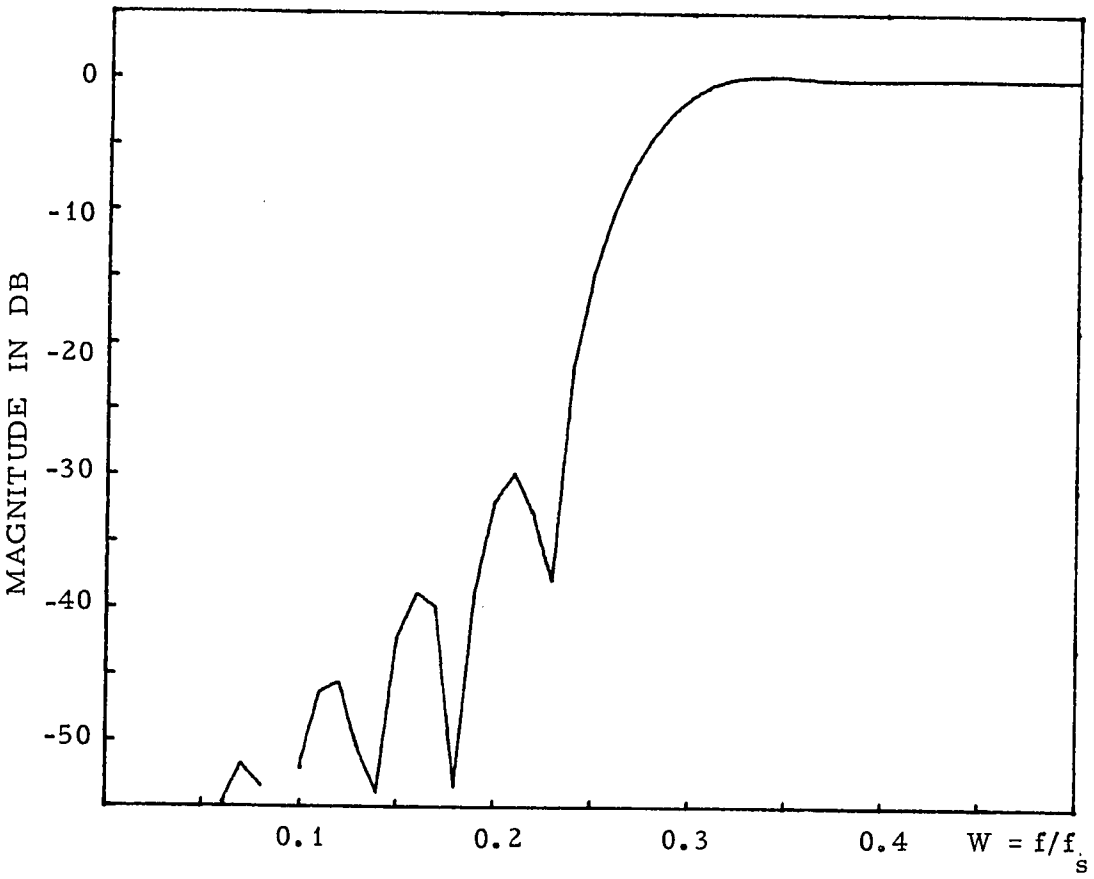


Fig. 5.10. Highpass filter M = 23, N = 44 and β = 6.

(M is the number of nonzero coefficients)

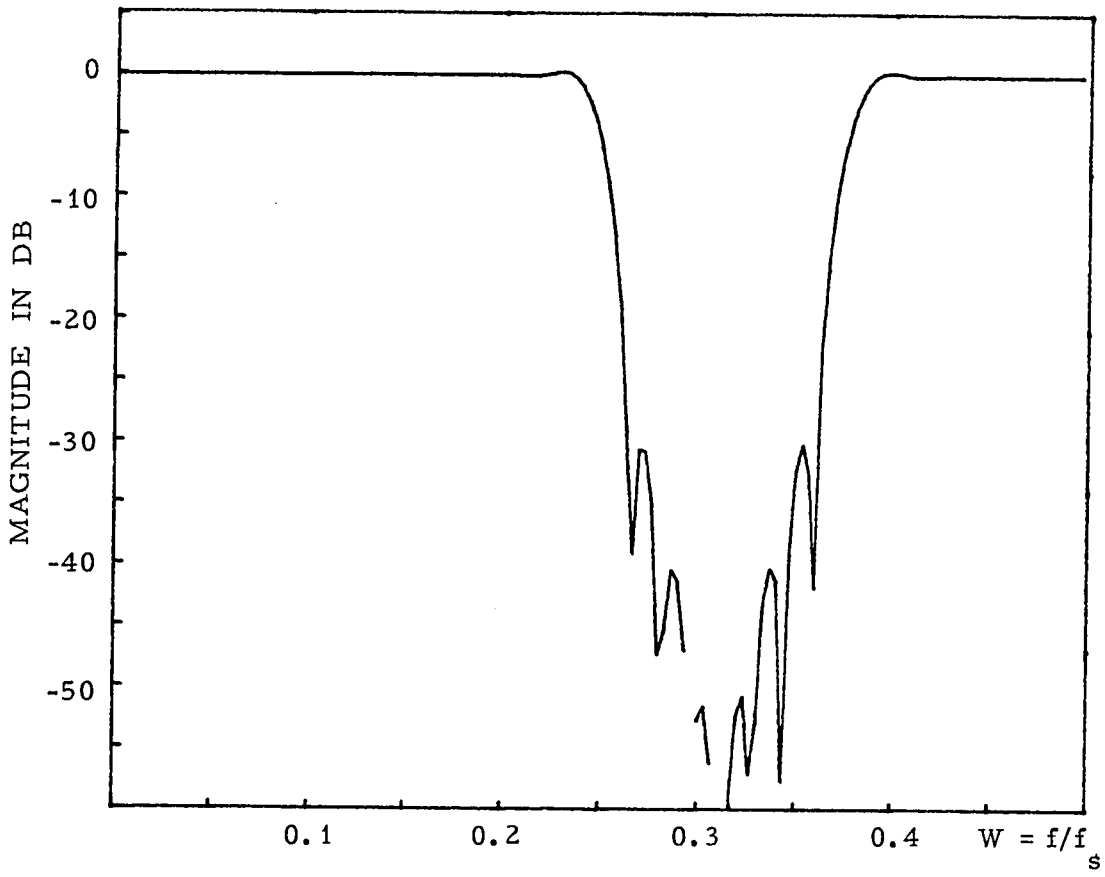


Fig. 5.11. Bandstop filter M = 65, N = 128

$\beta_1 = 16, \quad \beta_2 = 24$

(M is the number of nonzero filter coefficients)

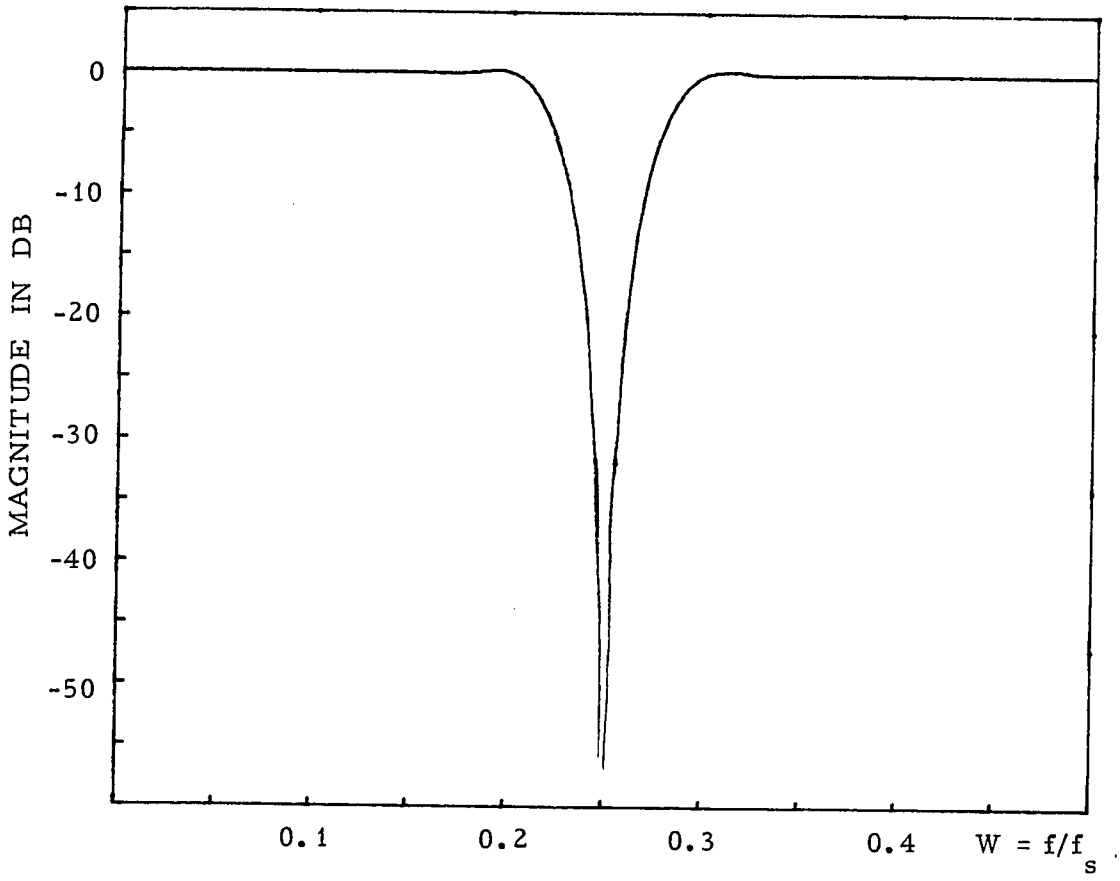


Fig. 5.12. Narrow bandstop filter M = 41, N = 80
$\beta_1 = 9$, and $\beta_2 = 11$.
(M is the number of nonzero filter coefficients)

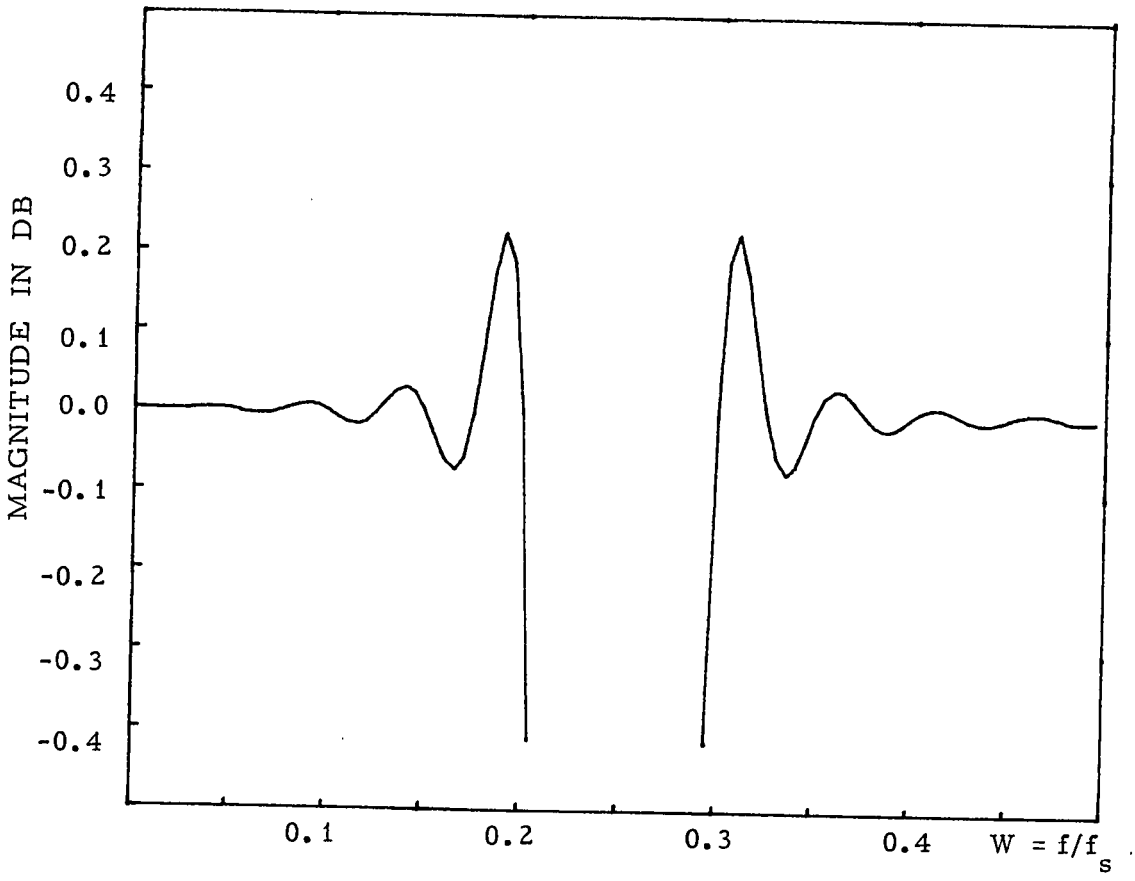


Fig. 5.13. Ripples in the passband of the narrow bandstop filter shown in fig. 5.12.

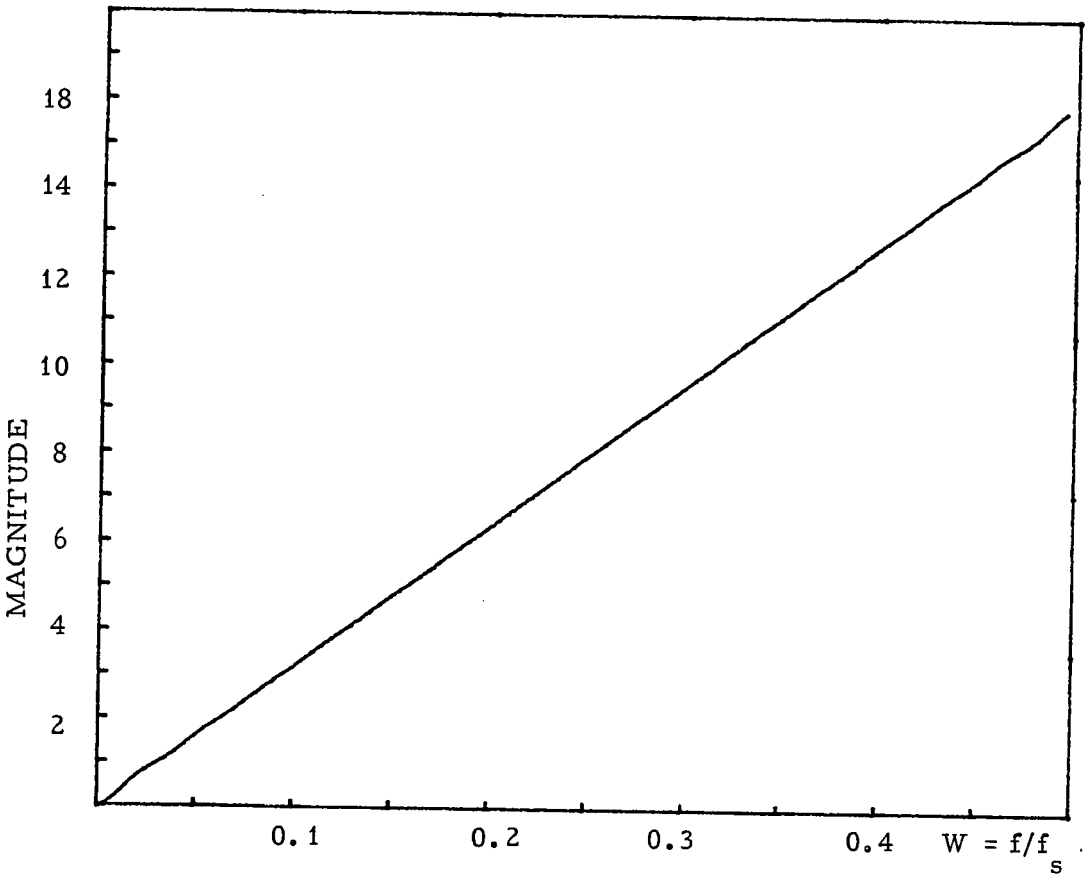


Fig. 5.14. Linear magnitude filter response
N = 64, M = 33.

(M is the number of nonzero filter coefficients)

5.3 ANALYSIS OF THE APPROXIMATION ERROR

Since all the filter designs discussed so far in this chapter are based on the DHT, it would be necessary to analyze the approximation errors of the resulting Hilbert transformers when using the middle row of the Hilbert matrix.

In order to simplify the analysis we shall first shift the impulse response h(n) to remove the phase linearity. The resulting impulse response is now a noncausal sequence with symmetry around n = 0. This has no effect on the magnitude response. For the DHT the resulting noncausal impulse response is :

$$h(n) = \begin{cases} \cot\left(\frac{\pi}{N} n\right) & n = +1, \pm 3, \pm 5, \ldots, \pm\left(\frac{N}{2}-1\right) . \\ 0 & n = 0, \pm 2, \pm 4, \ldots, \pm\left(\frac{N}{2}-2\right) . \end{cases} \tag{5.10}$$

The impulse response h(n) given by (5.10) is basically the $\left(\frac{N}{2}-1\right)^{th}$ row of the Hilbert matrix. The fact that it is noncausal should not form any problem since our main interest in this section is to analyze the approximation error and not to realize the filter .

Alternatively (5.10) can be rewritten as :

$$h(n) = \frac{2}{N} \cot\left(\frac{\pi n}{N}\right) \sin^2\left(\frac{n\pi}{2}\right), \; n = 0, \pm 1, \pm 2, \pm 3, \ldots, \pm\left(\frac{N}{2} - 1\right). \tag{5.11}$$

The frequency response of the Hilbert transformer whose impulse response is given by (5.11) is :

$$H(W) = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}-1} h(n)\, e^{-j2\pi Wn}$$

$$= \frac{2}{N} \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}-1} \cot\left(\frac{n\pi}{N}\right) \sin^2\left(\frac{n\pi}{2}\right) e^{-j2\pi Wn}$$

$$= \frac{-4j}{N} \sum_{n=1}^{\frac{N}{2}-1} \cot\left(\frac{n\pi}{N}\right) \sin^2\left(\frac{n\pi}{2}\right) \sin(2\pi Wn) \qquad (5.12)$$

where

$$W = \frac{\omega}{\omega_s} = \frac{\omega T}{2\pi} \quad , \text{ and } \quad T \text{ is the sampling interval.}$$

From (5.12) it is clear that H(W) is purely imaginary.

The ideal Hilbert transformer has a frequency response given as :

$$H_{id}(W) = \begin{cases} -j & 0 \le W \le 0.5 \\ +j & 0.5 \le W \le 1.0 \end{cases} \qquad (5.13)$$

The impulse response $h_{id}(n)$ of the ideal Hilbert transformer is simply obtained [5] by expressing (5.13) into its Fourier series representation, and the resulting $h_{id}(n)$ is given by :

$$h_{id}(n) = \frac{2}{\pi} \frac{\sin^2(\pi n/2)}{n} \qquad n = 0, \pm 1, \pm 2, \pm 3, \ldots \pm \infty \qquad (5.14)$$

The impulse response of the ideal Hilbert transformer as shown in (5.14) is of infinite duration.

Thus (5.13) can be rewritten as :

$$H_{id}(W) = \frac{2}{\pi} \sum_{n=-\infty}^{\infty} \frac{\sin^2(\pi n/2)}{n} e^{-j2\pi Wn}$$

$$= \frac{-j4}{\pi} \sum_{n=1}^{\infty} \frac{\sin^2(\pi n/2)}{n} \sin(2\pi Wn) \qquad (5.15)$$

In general the approximation error E(W) is :

$$E(W) = H_{id}(W) - H(W) \tag{5.16}$$

and for the frequency band $0 \leq W \leq 0.5$, E(W) is :

$$E(W) = -j \{ 1 - \frac{4}{N} \sum_{n=1}^{\frac{N}{2}-1} \cot(\frac{n\pi}{N}) \sin^2(\frac{n\pi}{2}) \sin(2\pi Wn) \} \tag{5.17}$$

Therefore,

$$| E(W) | = 1 - \frac{4}{N} \sum_{n=1}^{\frac{N}{2}-1} \cot(\frac{n\pi}{N}) \sin^2(\frac{n\pi}{2}) \sin(2\pi Wn) \tag{5.18}$$

Relation (5.18) gives the resulting error magnitude when the elements of the middle row of the Hilbert matrix are used as the coefficients of a FIR filter.* Also it can be seen,from (5.17), that the resulting approximation error affects only the magnitude response but the phase response is not affected i.e. ideal phase response.

5.4. COMPARISON WITH SOME FIR FILTER DESIGN TECHNIQUES

In this part we shall compare filter designs using the middle row of the filter matrix and two main FIR filter design techniques i.e. the truncated Fourier series approach [1] and the min-max approach [2,3]. In the truncated Fourier series approach the desired frequency characteristics is expressed as a Fourier series expansion. The resulting Fourier coefficients (impulse response) are truncated to a finite number of samples M, and these are used as the coefficients

* The approximation of the Hilbert transform and subsequent filters can be viewed as a windowing operation, where the window function $\phi(n)$ is given by :

$$\phi(n) = \frac{\pi n}{N} \cot(\frac{\pi n}{N}).$$

of the FIR filter. Fig. 5.15 shows a Hilbert transform designs using the middle row of the Hilbert matrix and truncated Fourier series expansion. Both designs are of the same order i.e. M = 79. We note that the truncated Fourier series design has larger ripples than the middle row of the Hilbert matrix. Both the Fourier series approach and the middle row of the matrix have the advantage that the filter coefficients are given in closed form expressions but the middle row gives less ripples. Table 5. gives the coefficients of both designs.

A well known FIR filter design technique was presented by Parks, MClellan, and Rabiner [2,3] where the Remez exchange optimization algorithm [7] has been used to find the optimum FIR filter coefficients in a min-max sense. Fig. 5.16 shows two FIR Hilbert transformer designs using the middle row of the Hilbert matrix and the McClellan et al. algorithm [4]. Both designs are of the same order, M = 19, and have the same transition bandwidth. Fig. 5.17 shows the approximation errors for both designs. We note that the middle row design has higher ripples near the transition band edges and lower ripples all over the rest of the passband as compared to the min-max (equal ripple) design. In other words it can be seen that the middle row gives a better approximation to the desired characteristics over approximately 75% of the passband and McClellan algorithm gives a better approximation over 25% of the band of interest.

Fig. 5.18 shows two FIR lowpass filter designs using the middle row of the lowpass matrix and McClellan et. al. algorithm. Both designs are of the same order, M = 40, and have the same transition bandwidth. Fig. 5.19 shows the approximation error in the passband for both designs. We note the following :

(i) In the passband:

The middle row gives a better approximation to the desired characteristic over approximately 80% of the passband and less accurate approximation near the transition band.

(ii) In the stop band:

The middle row gives a better attenuation over approximately 70% of the stop band and again less attenuation near the transition band.

In addition all FIR filter designs using the middle row of the matrix give filter coefficients which are completely defined in closed form expressions.

5.5. CONCLUSION

It has been shown that the DHT filtering approach can be used to generate filter designs using the middle row of the Hilbert or any of the filter matrices. These filters are realizable in the form of FIR linear phase digital filters. In this way almost any filtering characteristic can be realized using the DHT approach. Several design examples were presented and their properties were discussed. The comparison between the DHT filtering technique, the Fourier series technique,and the min-max technique shows that the DHT filtering gives better results than the Fourier series and comparable results with the min-max. It also has the advantage that the coefficients are completely defined in closed form expressions.

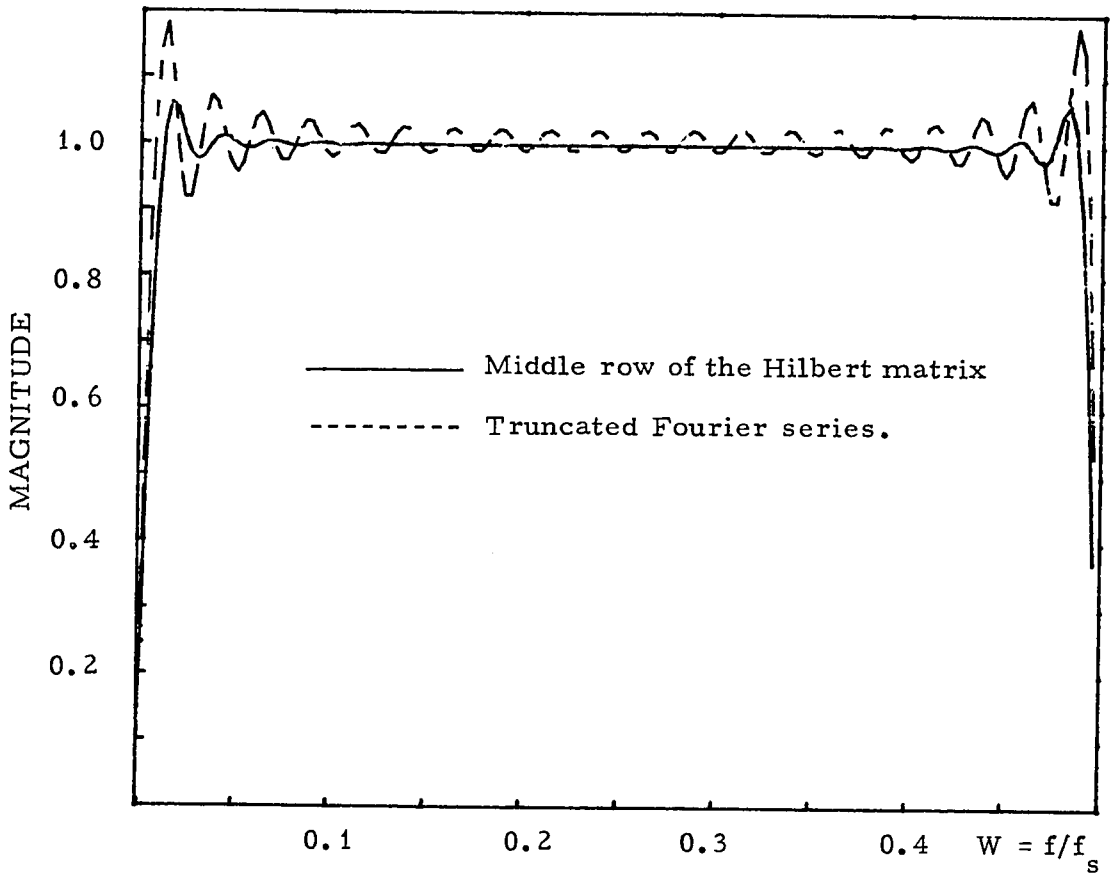


Fig. 5.15. Frequency responses of two FIR Hilbert transformers using the middle row of the Hilbert matrix N = 80, M = 79, and Truncated Fourier series M = 79. Both designs are of the same order.

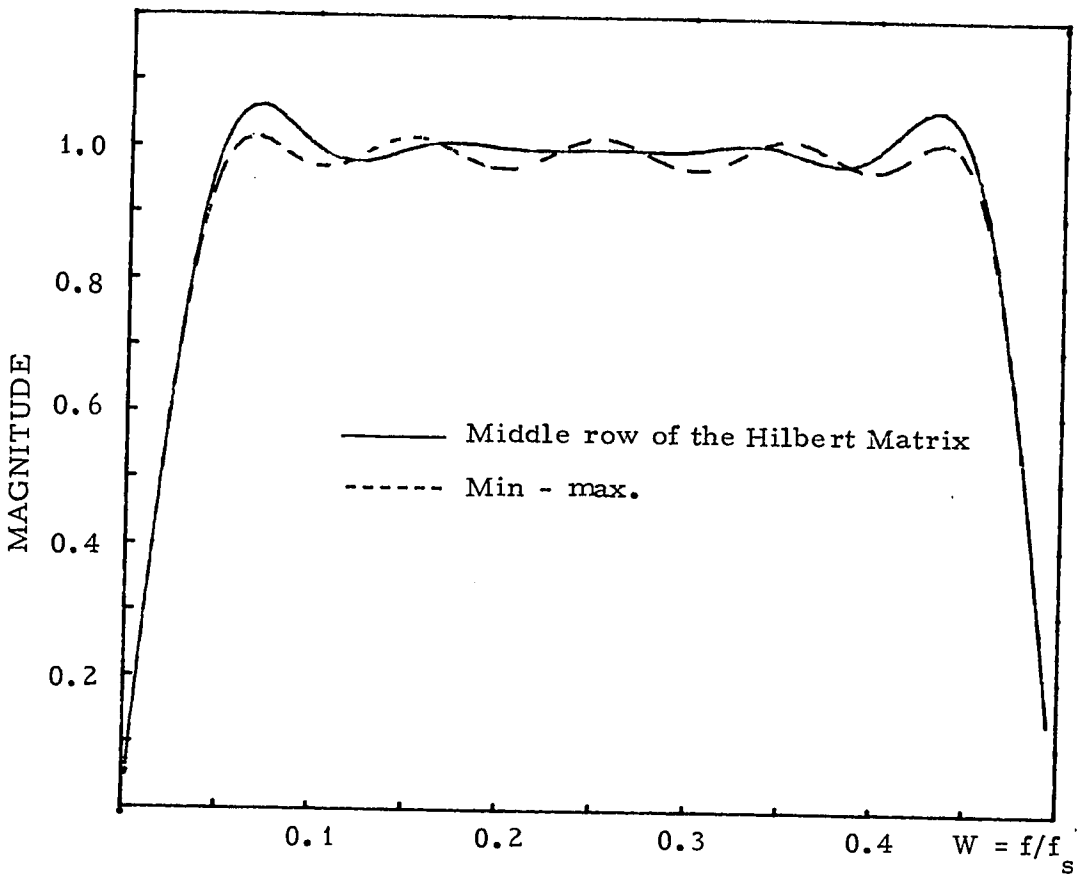


Fig. 5.16. Comparison between FIR Hilbert transformer designs having the same order and transition bandwidths using the middle row of the Hilbert matrix and the min - max technique [4], both filters are of order M = 19.

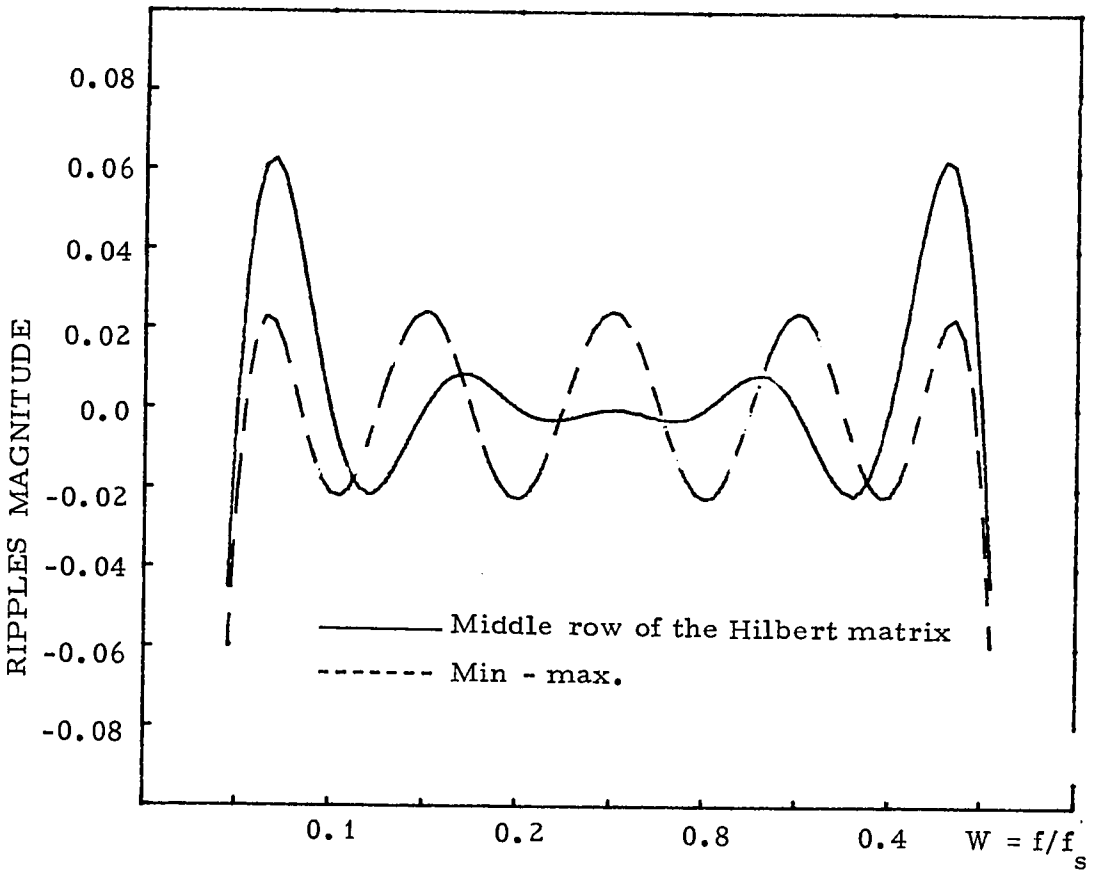


Fig. 5.17. Comparison of ripples magnitude for the two FIR Hilbert transformer designs shown in fig. 5.15.

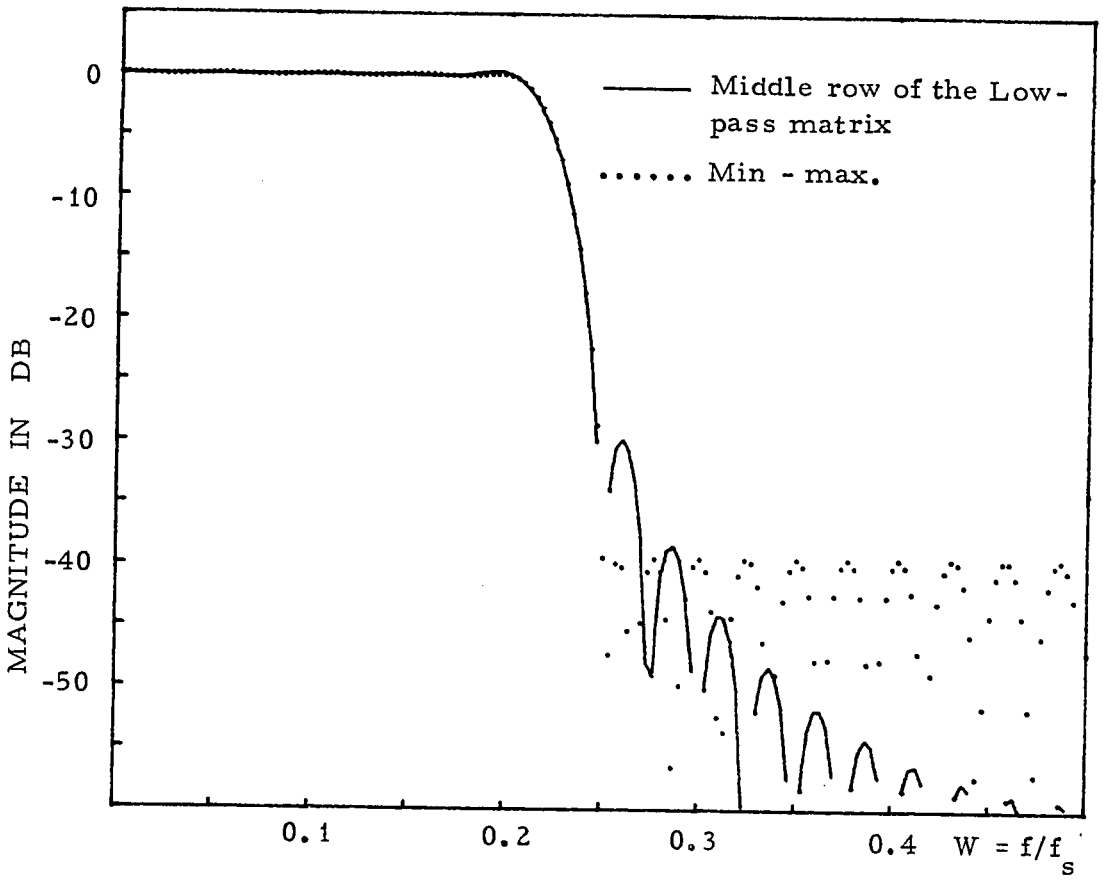


Fig. 5.18. Comparison between Two FIR lowpass filter designs using the middle row of the lowpass matrix and the min - max. algorithm [3]. Both designs are of the same order (M = 40) and same transition bandwidth.

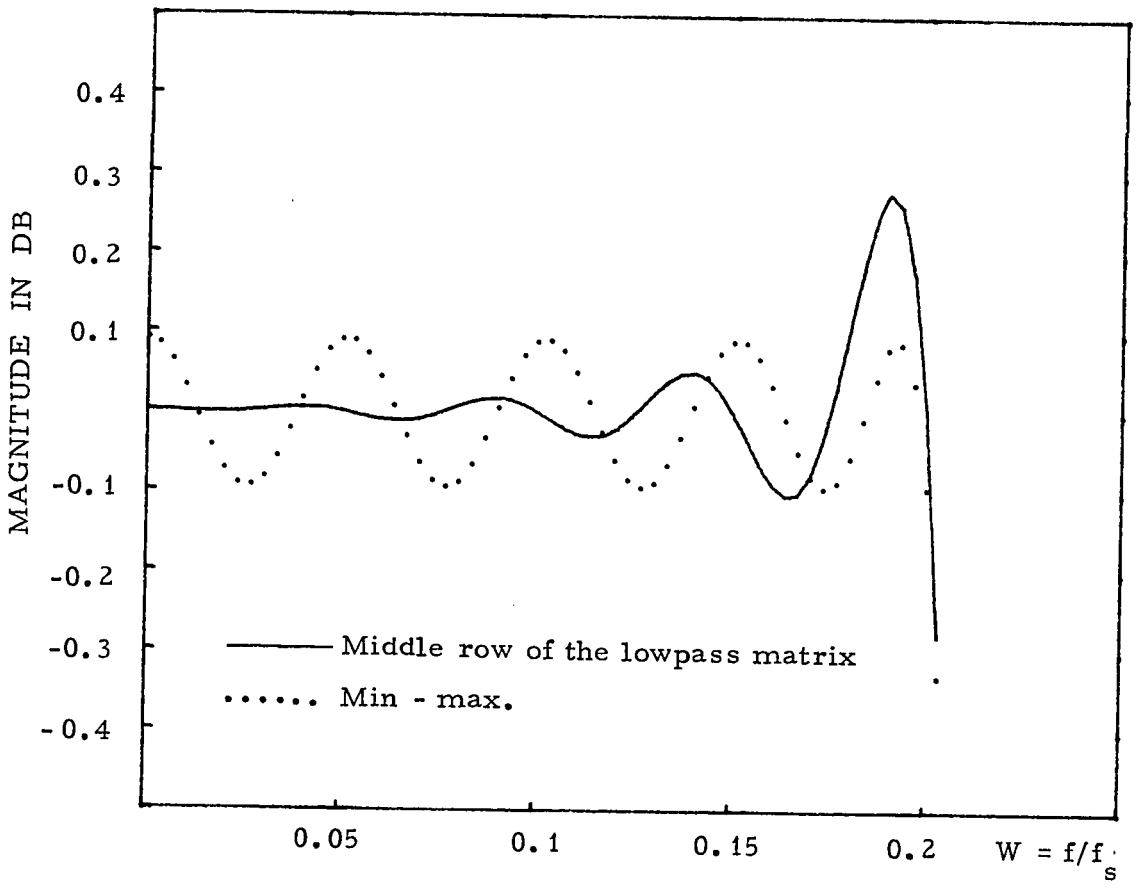


Fig. 5.19. Ripples in the passband of the two lowpass filter designs shown in fig. 5.17.

REFERENCES

[1] A. Oppenheim and R. Schafer, "Digital Signal Processing", Prentice Hall, 1975, pp. 337-365.

[2] T. Parks and J. McClellan, "Chebyshev Approximation for Nonrecursive Digital Filters with Linear Phase", IEEE Trans. Circuit Theory, Vol CT-19, March 1972 pp. 189-194.

[3] J. McCellan, T. Parks, and L. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters", IEEE Trans. on Audio and Electroacoustics, AU-21, No 6, December 1973 pp. 506-526.

[4] L. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing, Prentice Hall, 1975 pp. 75-183.

[5] B. Gold, A. Oppenheim, and C. Rader, "Theory and Implementation of the Discrete Hilbert Transform", Proc. Symp. Computer Processing in Communications, Vol. 19, Polytechnic press, 1970, New York.

[6] J.F. Kaiser and F.F. Kuo, "System Analysis by Digital Computer ", John Wiley, N.Y., Chapter seven.

[7] E. Remez, "General Computational Methods of Chebyshev Approximation", Atomic Energy translation 4491, Kiev, 1957.

[8] R.W. Shafer, L.R. Rabiner and O. Herrmann, "FIR Digital Filter Banks for Speech Analysis", BSTJ Vol 54, No. 3, March 1975 pp 531-544.

Table 5.1

n	M = 32	M = 16	M = 8
1	-0.00108556	-0.00435274	-0.01758153
2	-0.00327779	-0.01340615	-0.05905920
3	0.00553503	0.02362227	0.13228251
4	0.00790646	0.03626922	0.44435823
5	-0.01045114	-0.05385075	
6	-0.01324448	-0.08268148	
7	0.01638831	0.14568866	
8	0.02002763	0.44871097	
9	-0.02438037		
10	-0.02979447		
11	0.03686676		
12	0.04672036		
13	-0.06175722		
14	-0.08821651		
15	0.14896645		
16	0.44979653		

Table 5.2

n	Coefficients
1	-0.00148483
2	-0.00168178
3	0.00190301
4	0.00019161
5	0.01394206
6	-0.01118480
7	-0.01867325
8	0.01543291
9	0.00053561
10	0.02242730
11	-0.00554694
12	-0.05945862
13	0.03456708
14	0.02537257
15	0.01112250
16	0.02277558
17	-0.16620054
18	0.04809698
19	0.28171779
20	-0.21385426

Table 5.3

n	Coefficients
1	0.00000719
2	-0.00005657
3	-0.00019920
4	0.00025808
5	-0.00069589
6	-0.00050343
7	-0.00152939
8	0.00067361
9	-0.00269765
10.	-0.00064803
11.	-0.00416380
12	0.00031381
13.	0.00585786
14	-0.00042532
15	-0.00768081
16	-0.00164114
17	0.00951068
18	-0.00337452
19	-0.01121035
20	-0.00563103
21	0.01263649
22	-0.00837879
23	-0.01364917
24	-0.01154874
25	0.01412142
26	0.01503732
27	-0.01394821
28	-0.01871150
29	0.01305427
30	-0.02241576
31	-0.01140013
32	-0.02598070
33	0.00898615
34	-0.02923264
35	-0.00585410
36	0.03200377
37	0.00208629
38	0.03414213
39	-0.00219780
40	-0.03552078
41	-0.00684721
42	0.03604566
43	-0.01168706
44	-0.03566166
45	-0.01652773
46	0.03435636
47	-0.02117498
48	-0.03216147
49	-0.02544038
50	0.02915155

Table 5.4

n	Coefficients*	Coefficients**
1	0.00017909	0.00010271
2	-0.00210640	-0.00052660
3	0.00336945	-0.00059388
4	-0.00736773	-0.00463289
5	-0.00474396	-0.00118599
6	0.02100144	0.01196639
7	0.00260212	0.00986975
8	0.00166272	0.00041568
9	-0.02205905	-0.02060826
10	-0.00915997	-0.02065407
11	-0.02626779	-0.00656694
12	0.04837165	0.03738618
13	0.05514175	0.04261875
14	-0.03900113	-0.00975028
15	-0.01783828	-0.04022207
16	-0.05704659	-0.05329473
17	0.00581979	0.00145494
18	0.01267510	0.04807614
19	0.14825571	0.08447448
20	-0.05155408	-0.01288852
21	-0.13551848	-0.08521524
22	0.12361221	-0.02178728
23	-0.21714245	-0.05428561
24	0.16711489	0.09584736

* Coefficients of multiple passband filter of fig. 5.8

** Coefficients of multiple passband filter of fig. 5.9.

Table 5.5

n	Coefficients
1	0.00000000
2	-0.00783135
3	0.00000000
4	-0.00798440
5	0.00000000
6	-0.00830268
7	0.00000000
8	-0.00881269
9	0.00000000
10	-0.00956012
11	0.00000000
12	-0.01061916
13	0.00000000
14	-0.01210973
15	0.00000000
16	-0.01423019
17	0.00000000
18	-0.01732294
19	0.00000000
20	-0.02201586
21	0.00000000
22	-0.02955903
23	0.00000000
24	-0.04273717
25	0.00000000
26	-0.06883576
27	0.00000000
28	-0.13232695
29	0.00000000
30	-0.36286859
31	0.00000000
32	-3.24488329
33	8.0000000C

Table 5.6

n	Fourier series coefficients	DHT
1	-0.01632358	-0.00098225
2	0.00000000	0.00000000
3	-0.01720593	-0.00295894
4	0.00000000	0.00000000
5	-0.01818913	-0.00497280
6	0.00000000	0.00000000
7	-0.01929150	-0.00705072
8	0.00000000	0.00000000
9	-0.02053612	-0.00922298
10	0.00000000	0.00000000
11	-0.02195240	-0.01152515
12	0.00000000	0.00000000
13	-0.02357851	-0.01400067
14	0.00000000	0.00000000
15	-0.02546479	-0.01670446
16	0.00000000	0.00000000
17	-0.02767912	-0.01970841
18	0.00000000	0.00000000
19	-0.03031522	-0.02310976
20	0.00000000	0.00000000
21	-0.03350630	-0.02704484
22	0.00000000	0.00000000
23	-0.03744822	-0.03171234
24	0.00000000	0.00000000
25	-0.04244131	-0.03741514
26	0.00000000	0.00000000
27	-0.04897075	-0.04464071
28	0.00000000	0.00000000
29	-0.05787452	-0.05422919
30	0.00000000	0.00000000
31	-0.07073553	-0.06776546
32	0.00000000	0.00000000
33	-0.09094568	-0.08864331
34	0.00000000	0.00000000
35	-0.12732395	-0.12568348
36	0.00000000	0.00000000
37	-0.21220659	-0.21122393
38	0.00000000	0.00000000
39	-0.63661977	-0.63629248
40	-0.00000000	0.00000000

Table 5.7

n	min - max	DHT
1	0.00267320	0.00063792
2	0.00786304	0.00252291
3	-0.00357576	-0.00190301
4	-0.00713690	-0.00685591
5	0.00065589	0.00072362
6	0.01108370	0.01148962
7	0.00301158	0.00326839
8	-0.01435846	-0.01543291
9	-0.00951593	-0.01029761
10	0.01641209	0.01757280
11	0.01927744	0.02056506
12	-0.01574893	-0.01656965
13	-0.03305284	-0.03456708
14	0.01012448	0.01042116
15	0.05269528	0.05406202
16	0.00516014	0.00531681
17	-0.08528757	-0.08619409
18	-0.04777150	-0.04809698
19	0.17976689	0.18009800
20	0.41309827	0.41323891

CHAPTER SIX

CONCLUSION

A discrete filtering technique based on the DHT has been presented. The filtering method and its implementation have been shown to compare favorably with the DFT or FFT filtering technique.

The DHT and the derived discrete filter transforms are each presented in matrix form and may be implemented using the same hardware in either series or parallel form. In this manner a large variety of filtering characteristics may be implemented by changes of the stored coefficients only.

Analysis of the different sources of errors introduced has been presented and expressions for the output MSE to MSS ratio were obtained for both fixed and floating point arithmetic. It has been shown that the filtering matrix technique has the following advantages :

1 - requires simpler and less complex hardware realization than the FFT.

2 - introduces significantly less error than the FFT.

3 - can achieve comparable speed with the FFT for N up to 128 samples.

While the filtering process in matrix form is a circular convolution process, direct convolution filters were shown to be realizable in the form of FIR linear phase digital filters by using the middle row of the Hilbert, or any of the filter matrices, as the filter coefficients. The properties of the resulting designshave been discussed and the approximation error has been analyzed. The direct convolution DHT filters were compared with two well known FIR filter design techniques i.e. the

Fourier series and the min-max techniques. It has been shown that the DHT filtering method gives better results than the Fourier series technique and comparable results to the min-max technique but has the added advantage that the coefficients are fully defined in closed form expressions.

A new approach of bandlimited signal extrapolation has been also presented. This technique is based on the filtering matrix approach. It has been shown that the total extrapolation process in achieved by a single matrix operation. The extrapolation matrix approach has the following advantages over other bandlimited signal extrapolation technique:

1 - it requires much less computations.

2 - can give exact extrapolation results when the matrix $[E_\infty]$ is used.

3 - can operate in real time.

The derivation of the extrapolation matrix has been presented, and its properties were discussed. Several examples were given. A new fast digital extrapolator realization was proposed.

While in this work we handled one dimensional processing of signals, two dimensional signals processing should be considered in future plans.

Extension of the DHT filtering matrix approach to two dimensions is fairly straightforward and there should not be any difficulty in implementing same filters to achieve two dimensional processing.

For the extrapolation algorithm presented in chapter four, one has to be more careful when utilizing such technique in two dimensional applications. The main difficulty that might exist is that we shall no longer be dealing with N x N matrices but with four dimensional arrays . The problem of evaluating the sum of matrices in sections 4.4 and 4.5

(relatins (4.28),(4.31),and (4.34)) and insuring the convergence of the matrix power series have to be solved first. Once this is done, the technique can be applied on two dimensional signals.