National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Tree Structure Filter Bank For Wideband Signal Processing

By
Akpa A. Marcellin

A Thesis Submitted to
the School of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of
Master in Applied Science

Ottawa-Carleton Institute for Electrical Engineering

Department of Electrical Engineering
Faculty of Engineering
University of Ottawa

April, 1995

ISBN 0-612-04936-1

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

ACKNOWLEDGMENTS

I would like to gratefully thank my supervisor, DR W. Steenaart for his academic guidance, practical advice and financial support throughout my studies. I would also like to express my sincere gratitude to him for his moral and spiritual support.

Many thanks to the Ambassade de Cote d'Ivoire and the University of Ottawa for their important financial support.

Financial support for this research obtained from the Natural Science and Engineering Research Council of Canada (NSERC) under the University of Ottawa grant N° 8572 is gratefully acknowledged.

Finally, I would like to thank my wife Aissa Sangare for her support and encouragement during my studies.

Table of Contents

List of Figures

Abstract

A N-parallel branches maximally decimated filter bank is generally implemented using the polyphase components implementation. In this case, a N-th band lowpass filter is designed and its polyphase components are derived to constitute the branch 'subfilters'. This approach uses a NxN FFT matrix that will be the source of the complex (numbers) operations. Obviously, when the number of branches is equal to 2, the computations remain real.

In a tree structure filter bank, the computations remain real with or without polyphase implementation. When the polyphase implementation is used, the branch signals at each stage are computed using a set of 2x2 FFT matrices leading to real computations.

In this thesis, a new implementation approach based on the tree structured is proposed. The derivation of the structure is based on the equivalent parallel structure implementation of the tree structured filter bank. It uses the polyphase components of a given half-band lowpass filter (real coefficients) followed by a NxN Hadamard matrix. The computations, as in the original tree structured filter bank, remain real. A simplified version of the structure is a 'tree structure' followed by an NxN Hadamard matrix.
A comparison between this new structure and the N parallel branch maximally decimated filter bank is made based on reconstruction error, computation complexity and processing delay.

I INTRODUCTION

1.1 Introduction

Digital filter banks have been studied in great detail [1]-[3] due to their extensive use in a number of applications such as speech analysis, radar and sonar processing [4] and in all applications where the incoming signal is separated into frequency bands prior to processing [1], so that the branch signals may be processed at a reduced sampling rate.

A typical filter bank comprises 2 major components: the analysis filter bank and the synthesis filter bank. An N-channel analysis filter bank splits the incoming signal x(n) into N subband signals $y_i(n)$, $0 \leq i \leq N-1$. An N-channel synthesis filter bank recombines the N signals components $y_i(n)$, forming y(n) which is to approximate as close as possible the input signal x(n). In all cases, the synthesis filters are derived from the analysis filters. An N-channel analysis filter bank can be seen as "a single input - N-output system" while an N-channel synthesis filter bank will be an "N input-single output system". When both the analysis and the synthesis banks are combined to form a filter bank, the overall system becomes a "single input single output system". Shown in figure 1.1 are the basic elements of an N-channel filter bank.

The primary concern is to design the analysis/synthesis filter bank to obtain a perfect, or "very good" reconstruction of the original input signal. Note that a signal can never be perfectly reconstructed because there will always be an error due to quantization or an error caused by the introduction of noise or a processing error between the analysis and synthesis stages. The notion of perfect reconstruction is application dependent: a reconstructed signal with a mean square error less than 10^{-6} can be taken as a 'perfectly' reconstructed signal, while in another application, the same signal error may not be acceptable.

Perfect reconstruction requires that the squared magnitudes of the analysis filters add to unity over the total bandwidth; this sets the basis for the filter design.

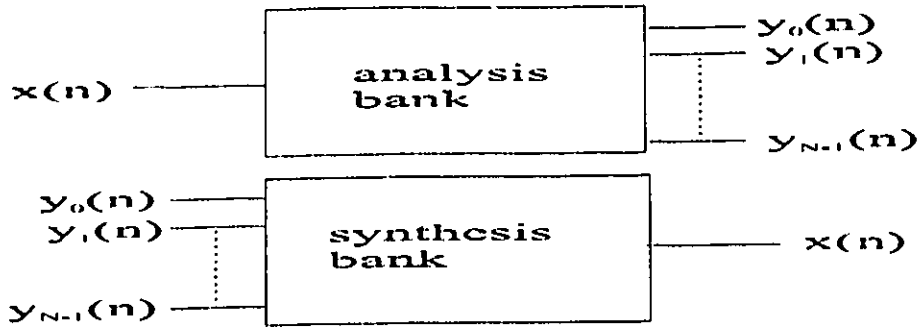


Fig. 1.1 N-channel filter bank elements: (a) analysis bank (b) synthesis bank

Filter banks differ in the way the center frequencies of the filters are spaced. Thus we have:

- uniform filter banks: the center frequencies of the analysis filters (and synthesis filters) are uniformly spaced (figure 1.2a) and do not depend on whether it is a cosine or exponentially modulated filter bank.
- non-uniform band filter banks: the class of filter banks in which the center frequencies of the filters are for example logarithmically spaced (logarithm base 2); they are sometimes referred to as logarithmic filter banks (figure 1.2b). There is also the class of filter banks with arbitrary spacing of the center frequencies, an example of such a filter bank will be a filter bank with rational decimation/interpolation factors.

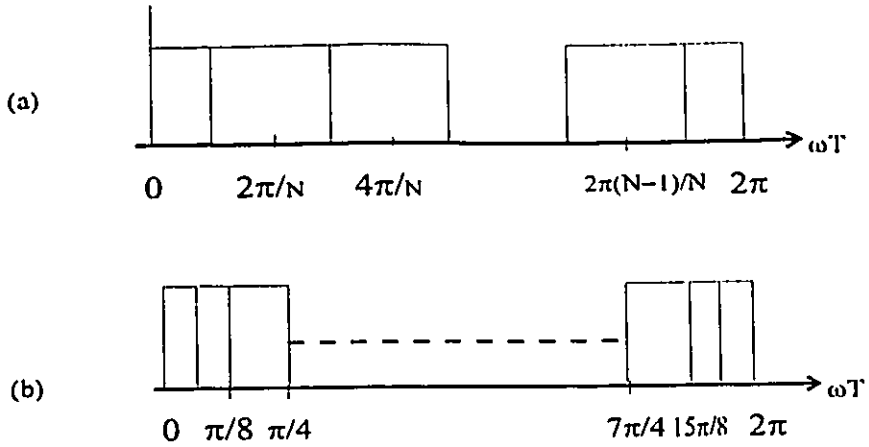


Fig. 1.2 Center frequencies spacing: (a) linear (b) logarithmic

The manner in which the frequency responses of the adjacent filters overlap (figure 1.3) in a filter bank depends mainly on the application. If perfect reconstruction is required, the magnitude characteristics of the adjacent filters are to overlap at -3 dB. This is generally required for filter banks that are used in spectrum analyzers [4].

The filters in a filter bank can be of the infinite impulse response (IIR) or the finite impulse response (FIR) type. Generally, finite impulse response filters are used because they are stable and simple to realize.

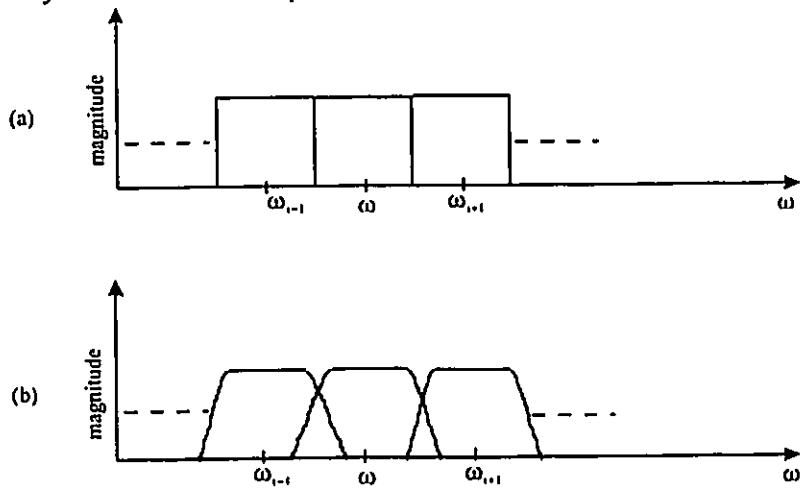


Fig. 1.3 Frequency coverage :(a) non-overlapping (ideal filters)
(b) slightly overlapping (practical filters)

1.2 Uniform Filter Banks

In an uniform band filter banks, the number of output branches in the analysis bank is equal to the overall decimation factor. This family of filter banks comprises the N parallel branches maximally decimated filter banks and the tree structured filter bank.

1.2.1 Uniform Parallel Structure Filter Bank

In the uniform parallel structure filter banks, the number of channels at the analysis filter bank output and thus the number of branches at the synthesis filter bank input, is equal to the decimation factor. In this case, the filter bank is said to be maximally decimated.

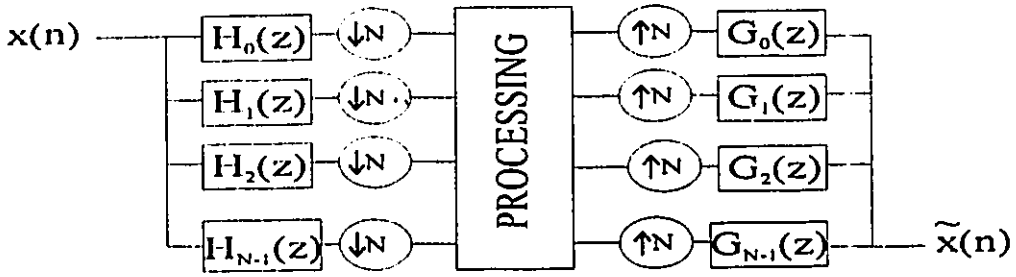


Fig. 1.4 N-parallel channel maximally decimated filter bank

1.2.2 Uniform Tree Structured Filter Bank

The tree uniform structured filter bank is a maximally decimated filter bank because the overall decimation factor is equal to the number of channels at the analysis filter bank output while the decimation factor per branch and per stage is 2. This structure gives rise to a successive lowpass-highpass splitting of the input. A 3-stage or 8 channel tree structured filter bank is illustrated in figure 1.5.

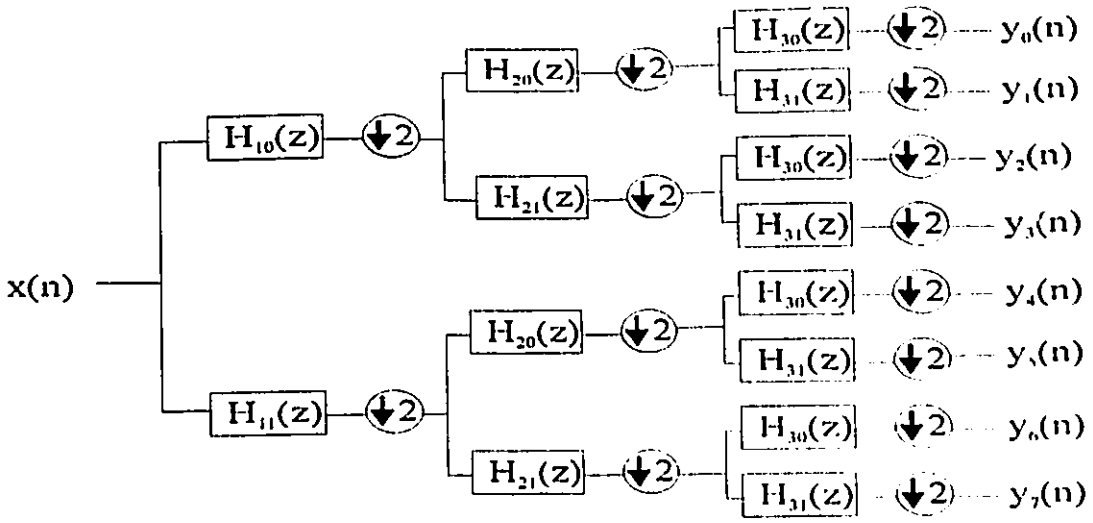


Fig. 1.5 3-level tree structure filter bank: analysis bank

In the above figure, if all the pairs of lowpass/highpass filters are identical in the analysis and synthesis filter banks, respectively, then the representation can be simplified by using the block representation as shown below.

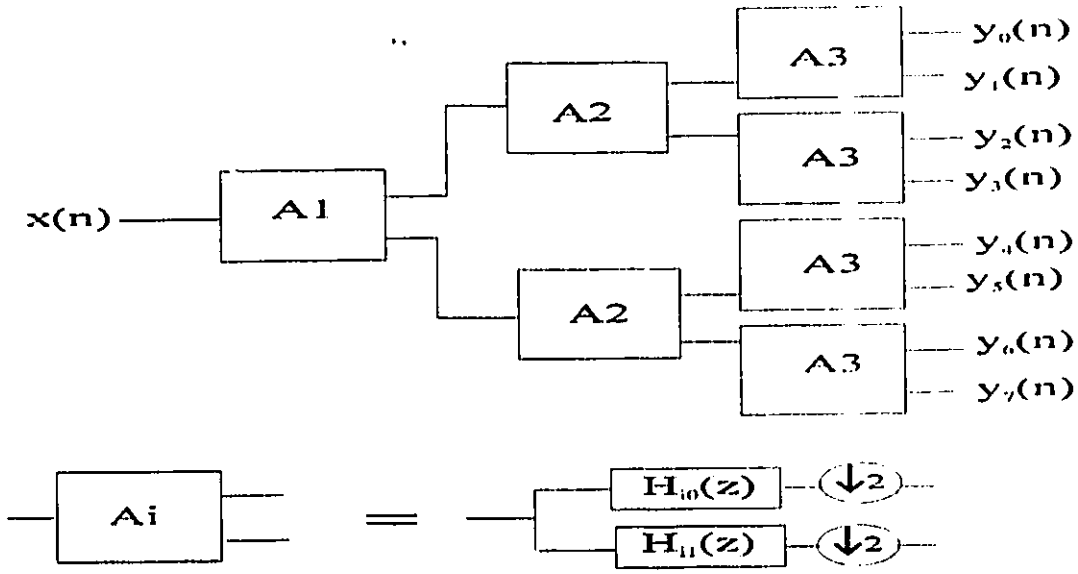


Fig. 1.6 Block representation of figure 1.5

1.3 Non-Uniform Filter Banks

In non-uniform filter banks, the decimation factors in the branches are different. They can be integer or fractional. Examples of non-uniform filter banks are shown in figure 1.7. The structure in figure 1.7 is mainly used for multiresolution analysis: the output of the lowpass filter at a given stage is further decomposed into a finer and coarser resolution. This structure (figure 1.7b) is similar to a tree structured filter bank, except that at a given stage, not all channels continue to the next processing filter set. Multiresolution analysis using the structures shown in figure 1.7 is found in [18-21].

Note that both structures shown in figure 1.7 are 'equivalent', i.e. if the decimation factors in figure 1.7b are moved to the output of the analysis bank, the resulting frequency band divisions of both analysis filter banks will be the identical.

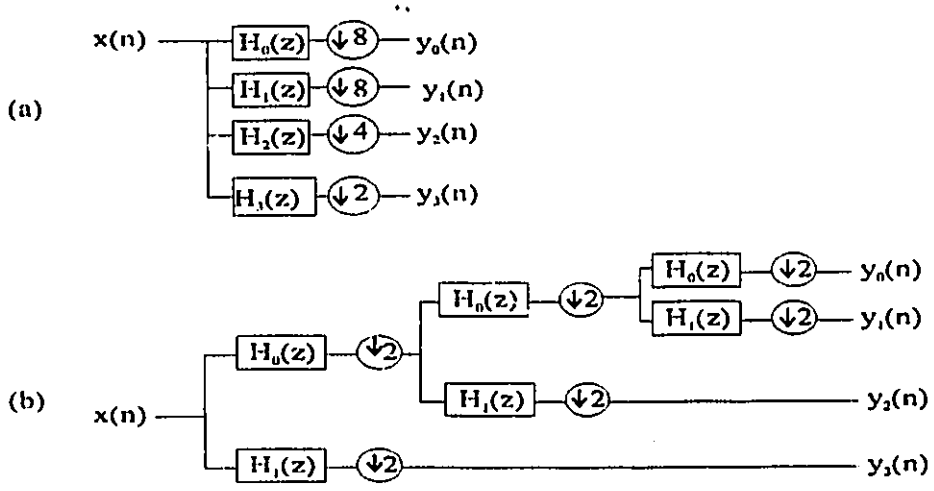


Fig. 1.7 Examples of non uniform filter banks: (a) parallel structure (b) wavelet-type structure

In the non-uniform parallel structure (example in figure 1.8), it is also possible to use non-integer decimation factors in the branches [22,29]. It is clear that in this case the

implementation of the system is going to be as straightforward as in the general case. One important condition that the decimation factors should satisfy is that the sum of their inverses should equal unity (in order for the filter frequency responses to cover the whole frequency axis). This condition is inherently satisfied for uniform filter banks which are maximally decimated.

$$\sum_i \frac{p_i}{q_i} = 1 \tag{1.1}$$

Note that (1.1) refers to the decimation factors and not the interpolation factors.

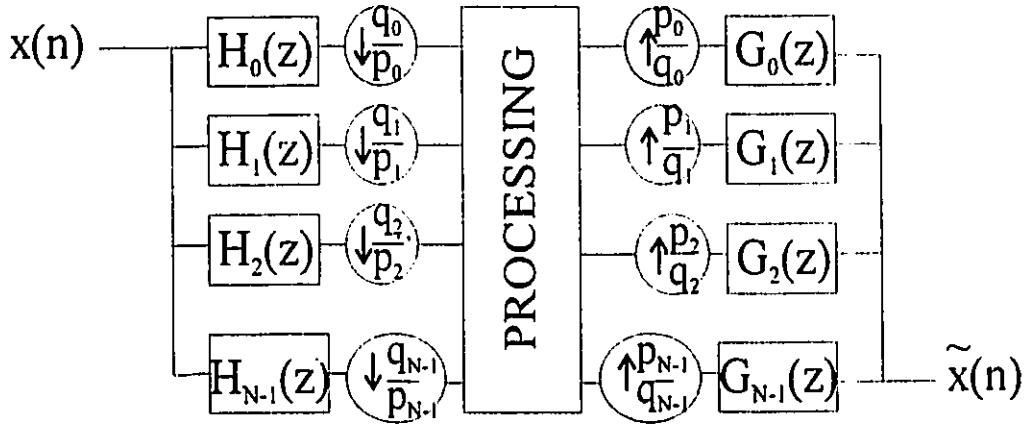


Fig. 1.8 Rational decimation factor analysis filter bank

1.4 Motivation and Organization of the Thesis

In this thesis, two filter bank structures are studied and compared: the tree structure filter bank and the N-parallel branch filter bank. Note that both structures are of the maximally decimated type and will be implemented using the polyphase components decomposition approach as outlined in section 2.2.3. An equivalent tree structure

implementation of the tree structure filter bank using the polyphase components and the Hadamard matrix is developed.

Based on the proposed structure, the performance (internal delay, computation complexity, ease of realization, speed of processing Vs signal bandwidth and reconstruction error) of the tree structure filter bank is compared with that of the parallel structure filter bank using the polyphase components and FFT matrix (figure 1.9). Wideband signals (pulse or pulse train) will be considered as input signal although any signal can be accommodated.

Fig. 1.9 Structure for comparison: increasing bandwidth input signal

The organization of the thesis is as follows. In chapter II, an overview of the uniform band filter bank is given. The polyphase components implementation or simply polyphase implementation of the filter banks including the DFT matrix is reviewed. The relation between the polyphase components and the structural subband components is outlined.

In chapter III, a new equivalent tree structure of the tree structured filter bank implemented with the polyphase components followed by the Hadamard matrix is presented. As only real filter coefficients are required in the lowpass and highpass filter branches of the tree structure, it can be shown that the Hadamard matrix, with real elements also, replaces the DFT matrix. A comparison based on filter length, input structure (serial or parallel input), computation complexity (number of computations per sample time), speed of operation and overall input/output delay between the new structure

and the N-parallel branch maximally decimated filter bank for a given decimation factor (a power of 2) is made.

In chapter IV, the overall reconstruction error performance of the parallel structure and the new developed structure for a given input signal is considered based on the simulation results. The advantages of the tree structured filter bank are shown, which are: the use of real coefficients only and ability to accommodate wider band signals (i.e. increased processing speed) than possible with the N-parallel branch filters.

In chapter V, the main conclusions of the thesis and suggestions for further research are discussed.

II UNIFORM BAND FILTER BANKS

2.1 Introduction

In this chapter, a review of maximally decimated filter banks is presented. An implementation using only the decomposition of the prototype lowpass filter together with the FFT algorithm is reviewed; this implementation is known as the polyphase implementation[13,14]. It will be shown that the polyphase implementation achieves some savings in the overall number of computations per sample time of the system. Also, the subband decomposition of a single filter (polynomial) and the relation to the polyphase components are outlined. Unless otherwise specified, all decompositions and computations will apply only to the finite impulse response filter (FIR) in maximally decimated filter banks.

2.2 Maximally Decimated Filter Banks

A maximally decimated filter bank is a filter bank in which the number of channels is equal to the overall decimation factor N; it is also referred to as critically decimated filter bank. In the following, we will assume that N is a power of 2 to match the requirements for the tree structured filter banks.

2.2.1 Parallel Structure Filter Banks

The filters $H_i(z)$ in figure 2.1 represent the filters of the analysis bank while the filters $G_i(z)$ represent the filters in the synthesis bank.

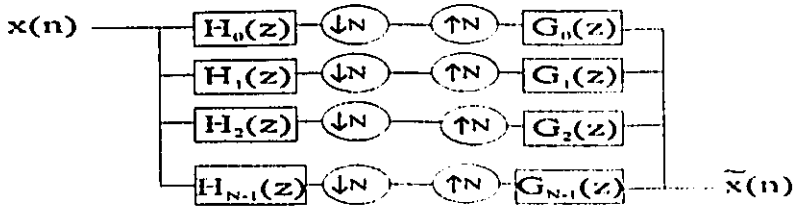


Fig. 2.1 N-channel maximally decimated filter bank

As shown in figure 2.1, the incoming signal x(n) is simultaneously filtered and decimated (downsampled) in each of the N branches before being processed. The processing can be of the form of coding which is done at the decimated sampling rate. At the synthesis bank, the received signals are decoded, interpolated (or upsampled), meaning that the signal is converted back to the original sampling frequency, and filtered prior to being added to approximate the original signal, $x(n)$. If the reconstructed signal $\tilde{x}(n)$ is equal to x(n), for example in the case where the processing is a direct connection between the analysis and synthesis filter banks and the filter bank is well designed, then the reconstruction is said to be perfect. Again, note that x(n) and $\tilde{x}(n)$ will never be completely equal. Generally, the reconstructed signal will be a delayed, scaled version of the original signal.

$$\tilde{x}(n) = cx(n - n_0) \tag{2.1}$$

where c and n_0 are the scaling factor and delay respectively, due to the filtering of the signal. Another source of reconstruction error is the digital processing performed on the subband signals in each channel branch.

For the uniform case, with other filters derived from the lowpass filter using complex modulators, we expect the analysis filters to divide the frequency axis as illustrated in figure 2.2 for ideal filters. In practice, the filters are designed so that the

overall frequency response in the analysis bank approximates a flat response. This flatness is the result of the in-band ripple of each filter and the complementary transition bands of the adjacent filters. The filters will then be designed to satisfy the following equation [3].

$$\sum_{i} \left| H_i(e^{j\omega}) \right|^2 = 1 \tag{2.2}$$

While there are N distinct filters in the filter bank of figure 2.1, only the lowpass filter is designed and the other filters are derived from it. The design of bandpass and highpass filters is done by frequency rotation (multiplication by complex exponential) of the lowpass filter design; this also is true for cosine modulated filter banks[28,29]. Another approach is to rotate the input signal in frequency and to use lowpass filters in all N branches of the filter bank [6].

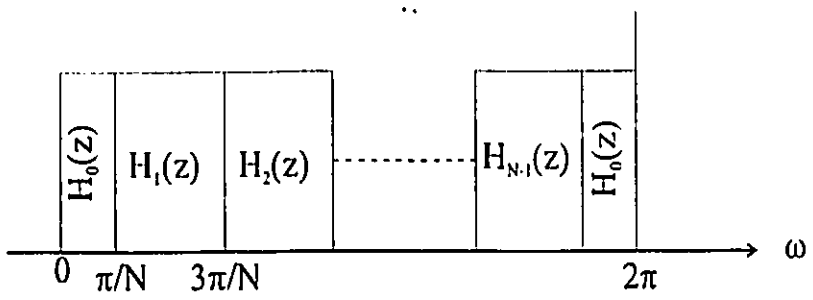


Fig. 2.2 Ideal frequency axis division of an N-parallel channel maximally decimated filter bank.

2.2.1.1 Filter Bank With Rotated Input Signal

The prototype filter is a lowpass filter with cutoff frequency at $\omega=\pi/N$ (or $f=f_S/N$) which remains the same in all N branches. In this implementation, the incoming signal is shifted in frequency in each branch, by multiplication with a complex exponential, (figure

2.3). The filters in the analysis and synthesis bank respectively are identical (lowpass filters) [5,6] with real coefficients. This filter bank has equal length filters and the output signals of the analysis filters are lowpass signals. It can be shown that this is equivalent to the N-branch filter bank with rotated filters by modulating the output of the analysis bank.

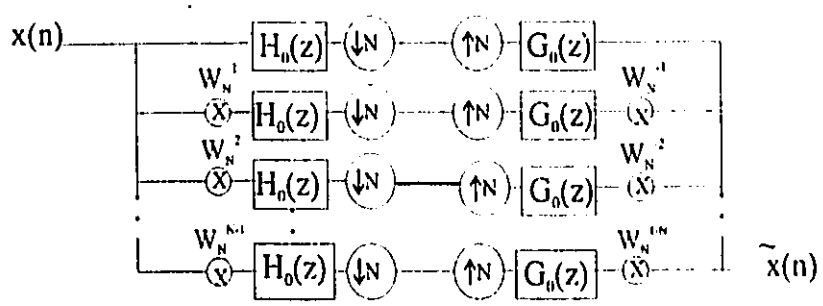


Fig. 2.3 Filter bank with complex modulators

In figure 2.3, the branch signals (prior to the decimators) are

$$x_k(n)=\left[x(n)W_N^{kn}\right]*h(n) \tag{2.3}$$

where (*) means convolution and $W_N = e^{-j2\pi/N}$

In the z-domain, (2.3) becomes

$$X_k(z)= X(zW_N^{-k})H(z) \tag{2.4}$$

Note in figure 2.3 that the input signal is 'complex-modulated' in each of the N branches prior to filtering. This operation is equivalent to rotating the original signal in the k-th branch by 2πk/N as shown in example 2.1.

Example 2.1

Let N=4 and the input signal be

$x(n)= 1 + \sin(0.2\pi n) + 4\sin(0.5\pi n) + 8\sin(0.8\pi n),\ 0 \le n \le 40$

The branch signals after modulation are then

$x_k(n)=x(n)\exp(j2\pi kn/4)$ or

$x_0(n)=x(n)$

$x_1(n)=x(n)\exp(jn\pi/2)=(\ j\ )^n x(n)$

$x_2(n)=x(n)\exp(jn\pi)=(-1)^n x(n)$

$x_3(n)=x(n)\exp(jn\pi 3/2)=(-j)^n x(n)$

The frequency responses are given in figure 2.4. Note the rotation of the signal around $f_s/2$ (corresponding to 0.5 on the frequency axis).

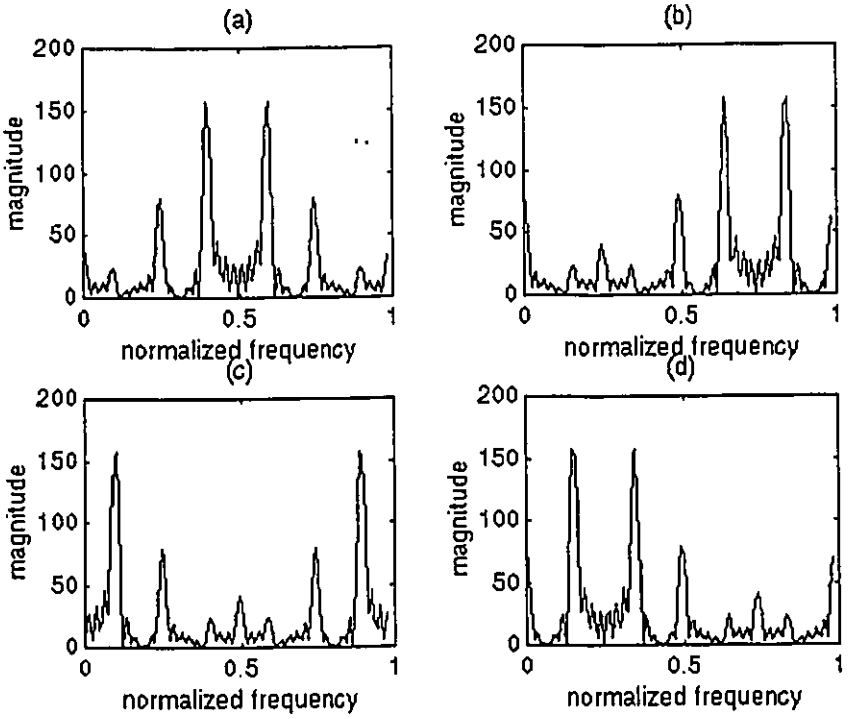


Fig. 2.4 Output of a 4-branch complex-modulator: magnitude of (a) $x_0(n)$, (b) $x_1(n)$, (c) $x_2(n)$, (d) $x_3(n)$.

If the length of the lowpass filter is L in figure 2.3, then the number of complex computation operations per sample will be 2LN plus N complex operations due the multiplication by the complex exponentials in the analysis bank. The signals at the filter output in the structure of figure 2.3 are lowpass signals.

2.2.1.2 Filter Bank With Rotated Lowpass Filter

A commonly used approach [1,2,7] is to derive the branch filters $h_i(n)$, by complex modulation of a lowpass filter, i.e. by multiplication of the lowpass filter coefficients with a complex exponential.

$$h_i(n) = h_0(n)W_N^{in} \qquad i = 0,..N-1 \tag{2.5}$$

where $h_0(n)$ is the prototype lowpass filter with cutoff frequency at π/N. This structure also leads to a filter bank with equal length filters: all the filters will be of the same length as the lowpass filter. However, the filters, except the lowpass and highpass filters, will have complex coefficients; processing at lower sampling rate will be allowed with the use of the polyphase realization.

Note that the difference between this approach and the previous one is that here, instead of modulating the incoming signal, it is rather the prototype lowpass filter that is modulated. As in the case of the modulated signal, the modulations by the complex exponential correspond to the rotation of the lowpass filter to yield the desired bandpass filters.

In the z-domain

$$H_i(z) = H_0(zW_N^{-i}) \tag{2.6}$$

An example of the filters frequency response for N=4 is shown in figure 2.5

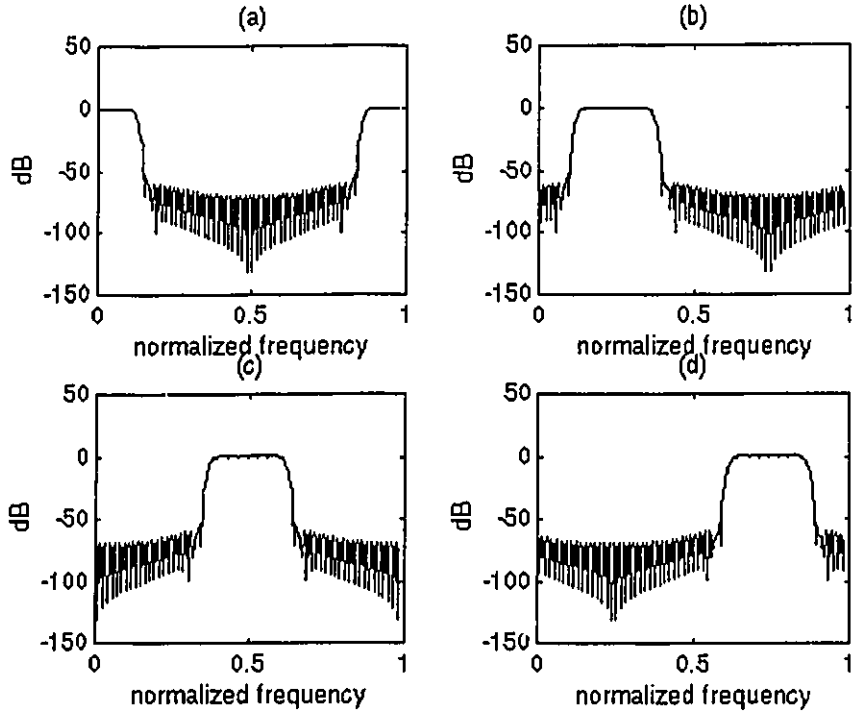


Fig. 2.5 Example of the frequency response of the analysis filters with N=4: (a) $H_0(z)$, (b) $H_1(z)$, (c) $H_2(z)$, (d) $H_3(z)$

Note also that the derivation of the branch filters using (2.5) generates filters that are not symmetric around π (or $f_s/2$) on the frequency axis (figure 2.5) except the lowpass and highpass filters. This is because the exponential modulators shift the lowpass filter one side only on the frequency axis.

The most important measure of performance of filter banks is how well the signal is reconstructed, i.e. how close the reconstructed signal is to the original one. Note however that there will be some aliasing in the branches of the filter bank. So an overall perfect reconstruction of the original signal will be obtained if the set of synthesis filters is

chosen to compensate for that aliasing. The analysis-synthesis filter banks overall transfer function of figure 2.1 will be

$$T_k(z) = H_k(z)G_k(z) \tag{2.7a}$$

The branch signal will be distortion free if the transfer function is a delay, i.e.

$$T_k(z) = cz^{-n_0} \tag{2.7b}$$

where c is a scaling factor introduced by the processing.

The function becomes a pure delay by choosing c to be unity. This then gives

$$H_k(e^{j\omega})G_k(e^{j\omega}) = e^{-j\omega n_0} \tag{2.7c}$$

One common solution is to choose the synthesis filters to relate to the analysis filters as follows [1]

$$g_k(n) = h_k(L_k - 1 - n), \qquad k = 0,1,..N-1 \tag{2.7d}$$

for the k-th branch, and where L_k is the length of $h_k(n)$ (analysis filters).

In the z-domain

$$G_k(z) = z^{1-L_k} H_k(z^{-1}) \tag{2.8}$$

For equal length analysis filters, as in the case where the analysis filters are all derived from the same prototype (L is the length of the prototype lowpass filter $h_0(n)$), (2.8) becomes

$$G_k(z)= z^{1-L} H_k(z^{-1}) \tag{2.9}$$

Substituting $h_k(n)$ from (2.5) and $H_k(z)$ from (2.6) in (2.8) and (2.9) respectively we have

$$g_k(n)= h_0(L-1-n)W_N^{k(L-1-n)} \tag{2.10a}$$

$$G_k(z)= z^{1-L} H_0(z^{-1}W_N^k) \tag{2.10b}$$

Applying (2.10b) to the structure in figure 2.3, we obtain

$$G_0(z) = z^{1-L} H_0(z^{-1}) \tag{2.11}$$

2.2.2 Special Case With N=2

The two band perfect reconstruction quadrature mirror filter (QMF) filter bank is a special case of the N-parallel channel maximally decimated filter bank in which the two analysis filters (lowpass and highpass filters) can be designed separately as in [7] or derived by applying (2.5) and (2.7). If the two filters are designed separately, the analysis filters need to satisfy the following conditions in order to achieve perfect reconstruction:
(a) an overall linear phase characteristic
(b) the sum of the lengths of the filters (lowpass and highpass) must be a multiple of 4 and
(c) both filters must be of either odd order and opposite symmetry or even order and symmetric (in the time domain).
The synthesis filters are derived using (2.7d).

If we choose to design only the lowpass filter and derive the high pass and the synthesis filters from it, we must replace the equality sign in (2.2) by an approximation to

satisfy the linear phase condition. The prototype lowpass filter is then constrained to be of even order and symmetric [8] so that the highpass filter obtained as

$$h_1(n)=(-1)^n h_0(n) \tag{2.12}$$

is also of even order and symmetric fulfilling conditions (a) and (b) mentioned above. Design examples can be found in [9,10, 11]. Figure 2.6 shows examples of possible analysis filters magnitude and phase

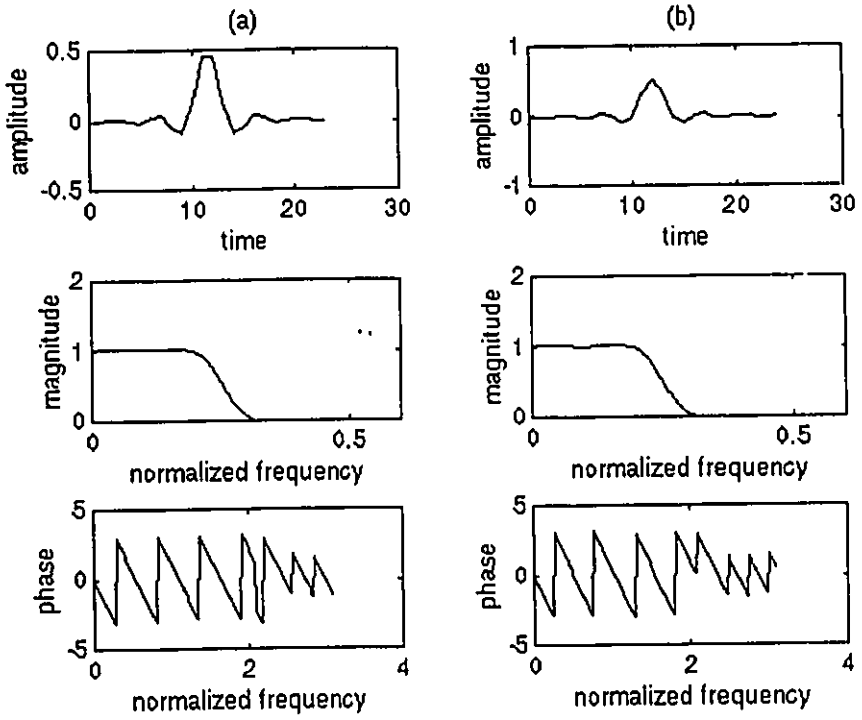


Fig. 2.6 Example overall filters phase:
(a) symmetric, even length prototype lowpass filter
(b) symmetric, odd length prototype lowpass filter

In [12], a different design approach is used: the author relaxes the linear phase constraint of the analysis filters to design a prototype lowpass filter with an arbitrary phase (most of the filter design algorithms generate linear phase filters). The analysis highpass filter and

the synthesis filters are derived such that the overall system becomes linear phase allowing the input signal to preserve its phase (no phase distortion). The established relations are:

$$h_1(n)=(-1)^{L-1-n}h_0(L-1-n) \quad (2.13a)$$

$$g_0(n)=(-1)^n h_1(n) \quad (2.13b)$$

$$g_1(n)=(-1)^{n-1}h_0(n) \quad (2.13c)$$

or in frequency domain

$$H_1(z)= z^{1-L}H_0(-z^{-1}) \quad (2.14a)$$

$$G_0(z)= H_1(-z) \quad (2.14b)$$

$$G_1(z)= -H_0(-z) \quad (2.14c)$$

where L is the length of $h_0(n)$.

Note in (2.12) that $h_1(0)$ and $h_0(0)$ will be the same. In (2.13a), $h_1(0)$ and $h_0(0)$ may be of the same or opposite sign depending on the length of $h_0(n)$.

For example, if $h_0(n)$=[h0 h1 h2 h3], then

$h_1(n)$=[h0 -h1 h2 -h3] with (2.12), and

$h_1(n)$=[-h0 h1 -h2 h3] with (2.13a)

2.2.3 Polyphase Components Decomposition

Maximally decimated filter banks often use the polyphase implementation approach. The decomposition consists of rewriting the lowpass filter function as sum of N-power terms.

In general, for any given function (polynomial) H(z), it is always possible to write [13,14]

$$H(z)=\sum_{k=0}^{N-1} z^{-k} E_k(z^N) \tag{2.15}$$

where the functions $E_k(z)$ are called the polyphase components of H(z), which are composed of N-powered terms of H(z).

For example, if

$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + h_4 z^{-4} + h_5 z^{-5} + h_6 z^{-6} + h_7 z^{-7} + h_8 z^{-8}$

then

for N = 4

$H(z) = \left[h_0 + h_4 z^{-4} + h_8 z^{-8}\right] + z^{-1}\left[h_1 + h_5 z^{-4}\right] + z^{-2}\left[h_2 + h_6 z^{-4}\right] + z^{-3}\left[h_3 + h_7 z^{-4}\right]$

$H(z) = E_0(z^4) + z^{-1}E_1(z^4) + z^{-2}E_2(z^4) + z^{-3}E_3(z^4)$

so

$E_0(z) = h_0 + h_4 z^{-1} + h_8 z^{-2}$

$E_1(z) = h_1 + h_5 z^{-1}$

$E_2(z) = h_2 + h_6 z^{-1}$

$E_3(z) = h_3 + h_7 z^{-1}$

In the time domain, the vectors of coefficients will be

$e_0(n)=[h_0\ h_4\ h_8]$

$e_1(n)=[h_1\ h_5]$

$e_2(n)=[h_2\ h_6]$

$e_3(n)=[h_3\ h_7]$

or

$$e_k(n)= h(nN + k) \quad , n=0,1,2... \tag{2.16}$$

Note that the components $e_k(n)$ are the time shifted (by k) and decimated (by N) versions of h(n).

The advantage of the polyphase decomposition is the saving in computation when it is used in filter bank implementation. To see this, assume that we have the scheme shown figure 2.7: filtering followed by a decimation by N,

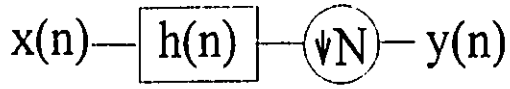


Fig. 2.7 Filtering followed by a decimation

The output Y(z) is given by

$$Y(z) = [X(z)H(z)]\downarrow_N \tag{2.17a}$$

where $[V(z)]\downarrow_N$ means V(z) followed by a decimation by N.

Replacing H(z) by its polyphase components, we have

$$Y(z) = [X(z)\sum_{k=0}^{N-1} z^{-k} E_k(z^N)]\downarrow_N \tag{2.17b}$$

and the corresponding implementation of (2.17b) of H(z) is shown figure 2.8. Note the delay elements on the input side between the channel inputs. An actual serial to parallel conversion of the input signal can be achieved if the decimation can be moved to the input side, which is possible for the structure with polyphase components.

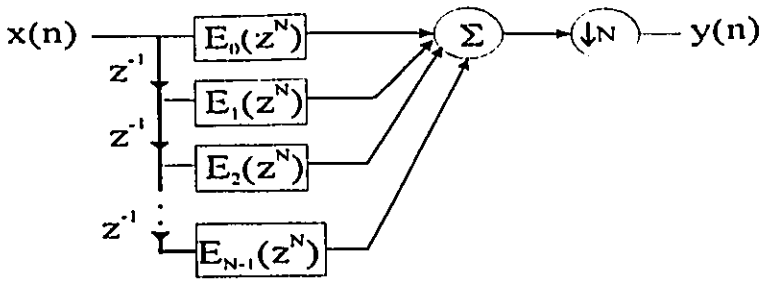


Fig. 2.8 Equivalent polyphase implementation of the filter in fig. 2.7

Let's rewrite (2.17b) with the substitution $z = e^{jw}$ (i.e. T=1)

$$Y(e^{jw})=\frac{1}{N}\sum_{l=0}^{N-1}X(e^{j(\frac{w-2\pi l}{N})})\sum_{k=0}^{N-1}e^{jk(\frac{w-2\pi l}{N})}E(e^{jN(\frac{w-2\pi l}{N})}) \qquad (2.17c)$$

or

$$Y(e^{jw})=\frac{1}{N}\sum_{l}\sum_{k}X(e^{j(\frac{w-2\pi l}{N})})e^{jk(\frac{w-2\pi l}{N})}E_k(e^{jw})$$

Note that $E_k(e^{jw})$ does not depend on l. It can thus be taken out of the first summation

$$Y(e^{jw})=\frac{1}{N}\sum_{k}E_k(e^{jw})\sum_{l}X(e^{j\frac{w}{N}}e^{\frac{-j2\pi l}{N}})e^{j\frac{kw}{N}}e^{\frac{-j2\pi lk}{N}} \qquad (2.17d)$$

(2.16d) can be written in a short form as

$$Y(z)=\sum_{k}E_k(z)[z^{-k}X(z)]\downarrow_N \qquad (2.17e)$$

This last equation is the derivation of the following noble identities [13] shown in figure 2.9

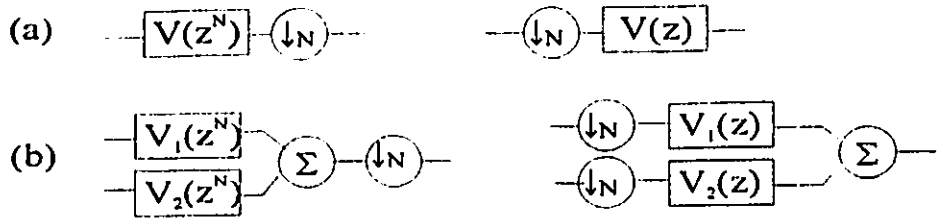


Fig. 2.9 Noble identities: (a) decimation, (b) addition

Figure 2.8 can then be redrawn as in figure 2.10.

The change from figure 2.7 to figure 2.10 is in essence a change from a realization with a direct serial input to one with parallel inputs derived from a serial input spread over the N channels. This is essentially a series to parallel input conversion. In figure 2.10, instead of computing the convolution and then decimating, we first decimate by N and then convolve, working at a lower sampling rate (f_S/N). This conversion (figure 2.10) allows a processing N times faster than the structure in figure 2.7.

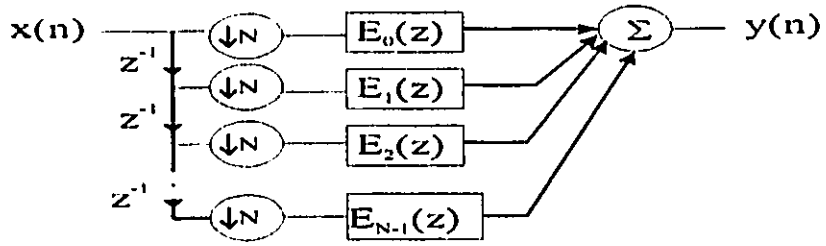


Fig. 2.10 Equivalent realization of figure 2.8

Now apply the polyphase components implementation to the N-parallel channel maximally decimated filter bank shown in figure 2.1. With the branch signals given in (2.5), we have

$$H_i(z)=\sum_{k=0}^{N-1}(zW_N^{-i})^{-k}E_k(z^NW_N^{-iN})$$

where $W_N^{-iN} = e^{j2\pi iN/N} = 1$

so

$$H_i(z)=\sum_{k=0}^{N-1} z^{-k} W_N^{ik} E_k(z^N) \tag{2.18}$$

for example, if N=4 then

$$\begin{bmatrix} H_0(z) \\ H_1(z) \\ H_2(z) \\ H_3(z) \end{bmatrix} = \begin{bmatrix} 1 & z^{-1} & z^{-2} & z^{-3} \end{bmatrix} \begin{bmatrix} E_0(z^4) & 0 & 0 & 0 \\ 0 & E_1(z^4) & 0 & 0 \\ 0 & 0 & E_2(z^4) & 0 \\ 0 & 0 & 0 & E_3(z^4) \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & j \\ 1 & 1 & j & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

and the outputs of the analysis bank are

$$Y_k(z)=\frac{1}{N}\sum_{j=0}^{N-1} E_j(z) W_N^{kj} \sum_{l=0}^{N-1} z^{-l/N} W_N^{jl} X(z^{1/N} W_N^l) \tag{2.19a}$$

or simply

$$Y_k(z) = [X(z)\sum_{j=0}^{N-1} z^{-j} W_N^{jk}]\downarrow_N E_j(z) \tag{2.19b}$$

Note that W^{jk} is the j-th row, k-th column element of the NxN discrete Fourier transform (DFT) matrix, **W**. The computations of $y_k(n)$ can thus be carried out via an N-point FFT algorithm: for N being a power of 2, faster computation is achieved.

The implementation of (2.19b) is shown in figure 2.12.

To achieve perfect reconstruction, we have [13]

$$f_i(n) = e_i(L_i - 1 - n) \tag{2.20}$$

where L_i is the length of $e_i(n)$.

Equation (2.20) means that overall perfect reconstruction will be achieved if the signal in each branch is perfectly reconstructed as mentioned earlier. And this applies to any N.

$$F_i(z) = z^{-L_i} E_i(z^{-1}) \tag{2.21}$$

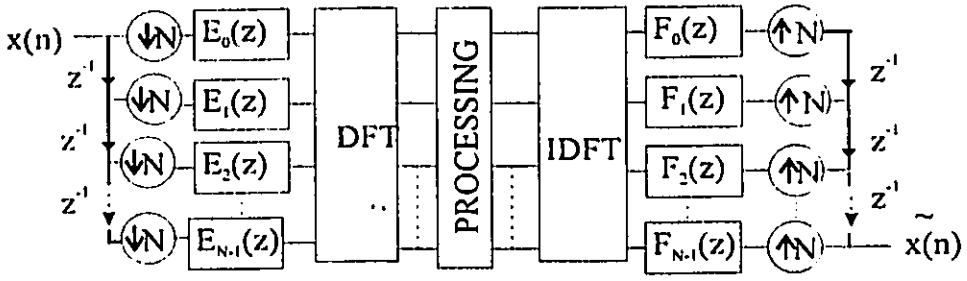


Fig. 2.11 N-channel maximally decimated filter bank: polyphase implementation

We can also apply the polyphase decomposition to the filter bank with complex modulators (figure 2.3). The difference with the previous implementation will be that here the derivation has to take into account the modulated branch signals from which we will obtain the FFT matrix. Extra modulators will then be added at the analysis bank output as shown in figure 2.12. The following illustration is actually to show how we can modify the structure in figure 2.3 in order to use the polyphase implementation.

Consider figures 2.12 a and b

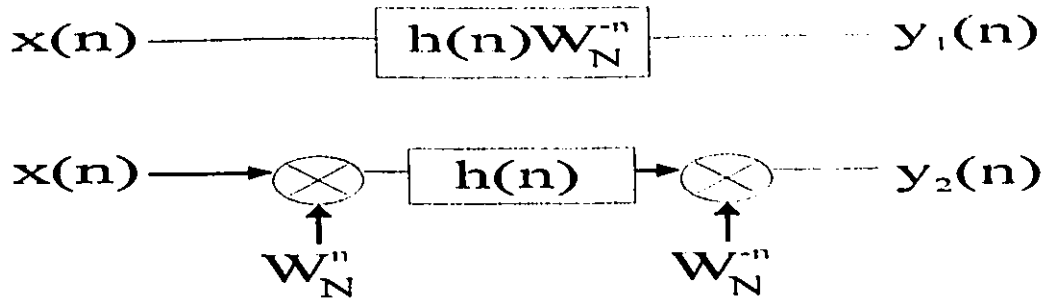


Fig. 2.12 Equivalence: (a) modulated filter, (b) 'doubly' modulated signal

The time domain equations of the outputs are

$$y_1(n) = x(n) * \left[h(n)W_4^{-n}\right] \tag{2.22a}$$

$$y_2(n) = \left[\left[x(n)W_4^{n}\right] * h(n)\right]W_4^{-n} \tag{2.22b}$$

where (*) means convolution.

Taking the z transform, we obtain

$$Z\left[h(n)W_4^{-n}\right] = Z\left[h(n)e^{2j\pi n/4}\right] = Z\left[h(n)(-jz)^{-n}\right] = H(-jz)$$

where Z[v(n)] is the z-transform of v(n)

$$Y_1(z) = X(z)H(-jz) \tag{2.22c}$$

and

$$\begin{aligned} Y_2(z) &= \left[X(jz)H(z)\right] * \delta(-jz) \\ &= X(z)H(-jz) \\ &= Y_1(z) \end{aligned} \tag{2.22d}$$

Figures (2.12a) and (2.12b) are thus equivalent.

Let's apply the same derivation to the system of figure 2.3,

$$X_k(z) = X(zW_N^{-k})\sum_{l=0}^{N-1} z^{-l}E_l(z^N) \tag{2.23}$$

If we now replace z by zW_N^{-k} in (2.23), this means that the signal at the output of the analysis filter bank is multiplied by a complex exponential, then we have

$$X_k(zW_N^{-k}) = X(z)\sum_{j=0}^{N-1} z^{-j}W_N^{jk}E_j(z^N W_N^{-kN}) \tag{2.24}$$

or

$$X_k(zW^{-k}) = X(z)\sum_{j=0}^{N-1} z^{-j}W^{jk}E_j(z^N) \tag{2.25}$$

Note that the branch signals given in the LHS of (2.25) are the frequency-shifted versions of $X_k(z)$. The RHS of (2.25) is similar to RHS of (2.18) and can thus be implemented the same way.

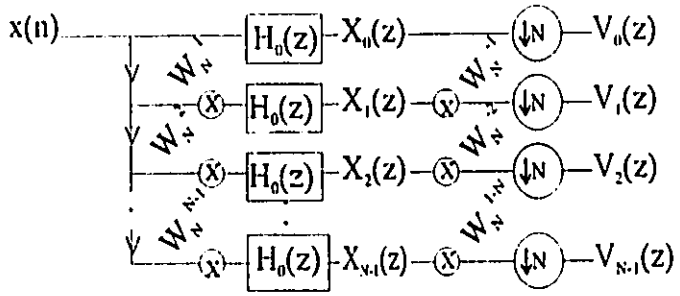


Fig. 2.13 Complex modulated analysis filter bank with extra modulators

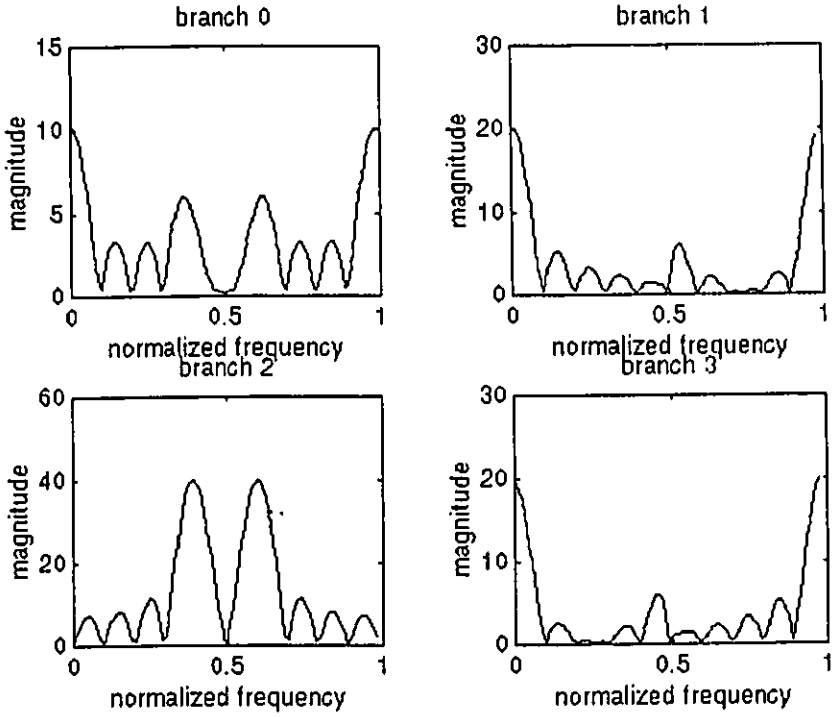
Example of the analysis filter bank output is shown in example 2.2

Example 2.2

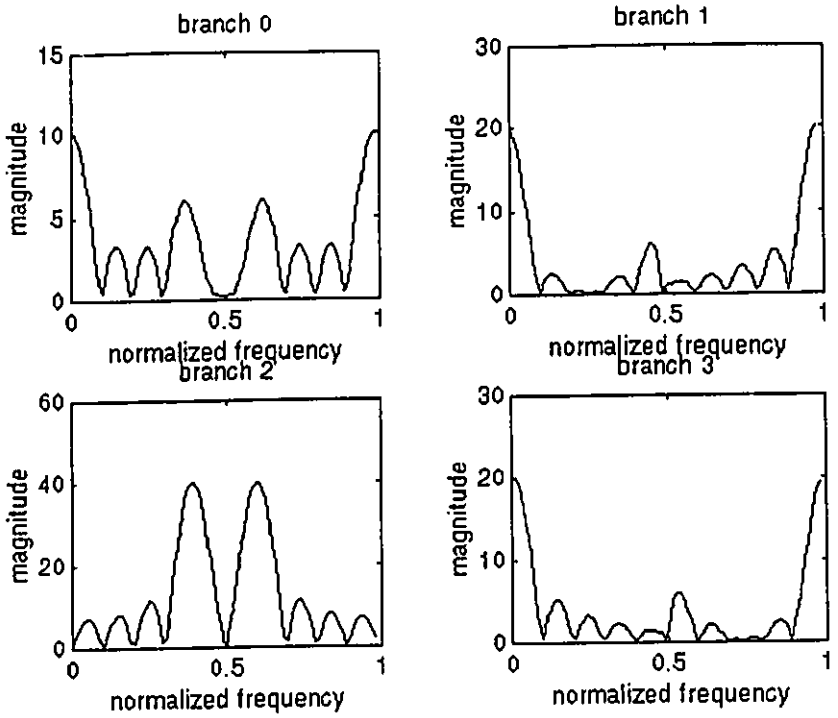
Let the input signal be as in example 2.1, i.e.

$x(n)= 1 + \sin(0.2\pi n) + 4\sin(0.5\pi n) + 8\sin(0.8\pi n),\ 0 \le n \le 40$

and also let N=4. The outputs of the analysis filter bank (figure 2.13) and the outputs of the equivalent polyphase implementation are shown in figure 2.14



(a)



(b)

Fig. 2.14 Branch signals: (a) with extra modulator
(b) with polyphase implementation

Consider the filter bank of figure 2.1. If we assume that all the filters are derived using (2.5), then to compute the N outputs, approximately 2LN (L is the length of the prototype lowpass filter) complex operations are required per output. This number is further reduced with the polyphase implementation of figure 2.11: 2L real operations + 1.5N $\log_2$N (due to FFT) complex operations .

The major problem with the N-parallel branch filter bank is that the design of the prototype lowpass filter becomes more and more difficult as the number of branches increases (smaller passband). This increase in the number of branches also leads to larger filter length and thus more computations. An alternative solution could be to use a multistage implementation [13,15,16].

Example 2.3

Assume that we want to decompose and reconstruct a pulse train of 6 pulses (no processing) using figure 2.11 with N=2 and N=4. Also assume that the RMS value of error should be less or equal to 10^{-3}. The idea here is to give a feeling of the difference in length of both filters.

Using the FIR1 function with the Hamming window in the MATLAB program, we enter the values 0.2506π (N=4) and 0.5238π (N=2) as cut off frequencies. The filter lengths that allow us to achieve an error less than 10^{-3} are 212 for N=4 and 34 for N=2 with $4.72\text{x}10^{-5}$ and $8.27\text{x}10^{-9}$ as RMS errors respectively.

2.2.4 Tree Structured Filter Banks

A tree structured filter bank is a cascade of blocks of the N-parallel branch filter bank with N=2 in each block. The overall decimation in each branch is determined by the total number of stages in the tree. A 2-level tree structured filter bank is shown in figure 2.15.

When we apply (2.6) to the QMF tree structured filter bank shown in figure 2.15, in which $H_0(z)$ and $H_1(z)$ are related as follows,

$$H_1(z) = H_0(-z) \tag{2.26}$$

we obtain the following equation and equivalent implementations:

$$\begin{bmatrix} H_o(z) \\ H_1(z) \end{bmatrix} = \begin{bmatrix} 1 & z^{-1} \end{bmatrix} \begin{bmatrix} E_0(z^2) & 0 \\ 0 & E_1(z^2) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

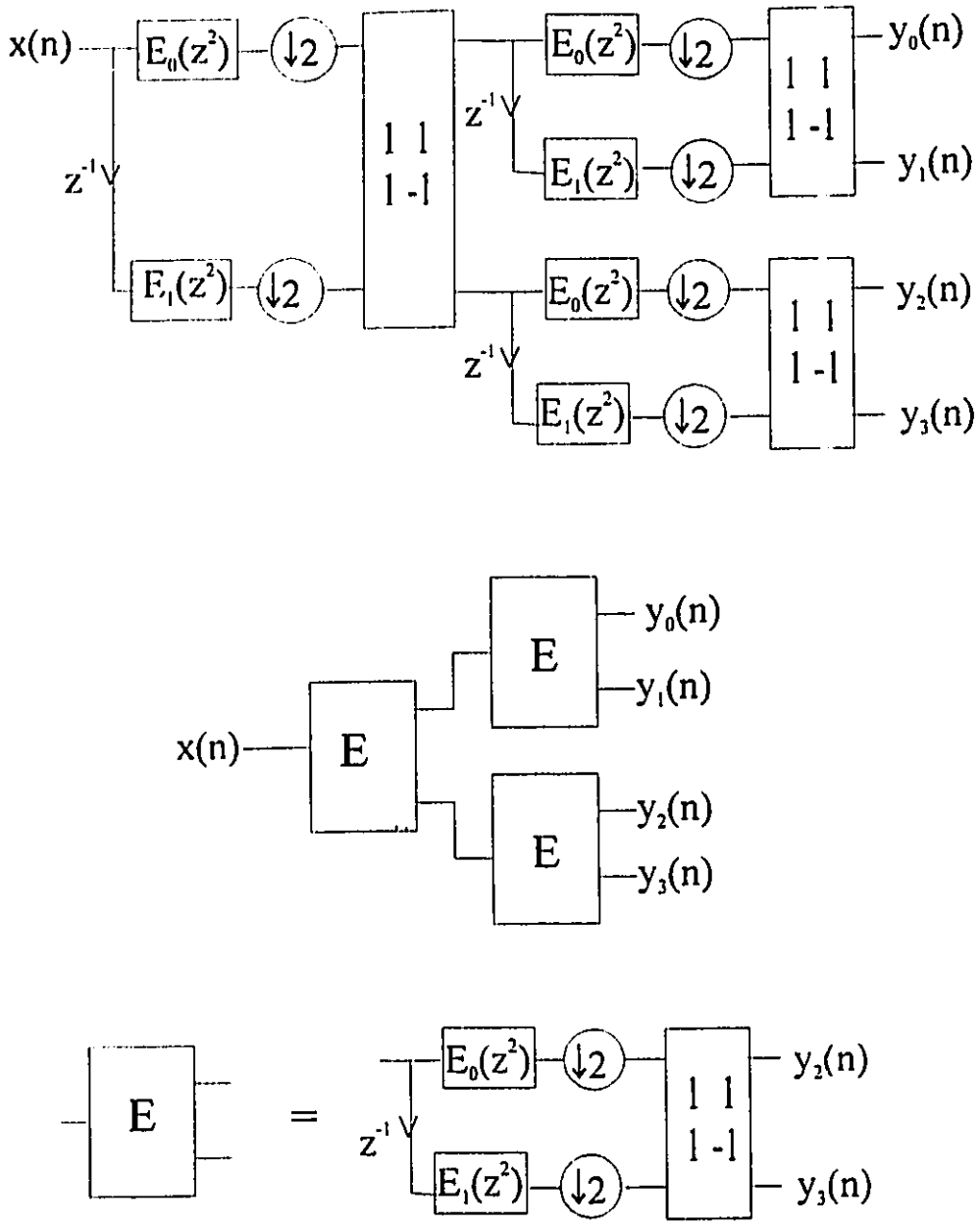


Fig 2.15 2-level tree structured filter bank and its block diagram implementation

2.3 Subband Decomposition

As in the case of the polyphase components decomposition, any FIR filter function V(z) can be written as [23]

$$(z)= \sum_{i=1}^{N} T_i(z) R_i(z^N) \tag{2.27}$$

where $r_i(n)$'s are the subfilters (if v(n) is of course a filter) and $T_i(z)$ is derived from the rows of the NxN Hadamard matrix; i.e. for N=4, the 4x4 Hadamard matrix is

$$J_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

and the functions $T_i(z)$ are

$$T_1(z)= 1+z^{-1}+z^{-2}+z^{-3}$$
$$T_2(z)= 1-z^{-1}+z^{-2}-z^{-3}$$
$$T_3(z)= 1+z^{-1}-z^{-2}-z^{-3}$$
$$T_4(z)= 1-z^{-1}-z^{-2}+z^{-3}$$

or

$$T(z) = [1 \;\; z^{-1} \; z^{-2} \; z^{-3}]\, J_4 \tag{2.28}$$

note in (2.28) that J_4 is used instead of the transpose. This is because the matrix J_4 and its transpose are related by 1/4.

The functions $r_i(n)$ are given by

$$\begin{bmatrix} r_1(n) \\ r_2(n) \\ \vdots \\ r_N \end{bmatrix} = \frac{1}{N} J_N \begin{bmatrix} v(nN) \\ v(nN+1) \\ \vdots \\ v(nN+N-1) \end{bmatrix} \tag{2.29}$$

Note that (2.29) gives also the relationship between the polyphase components $e_i(n)$ and the subfilters $r_i(n)$ of v(n). The two alternate implementations of a single filter are shown below (figure 2.16). The structure in figure 2.16b is appropriate for adaptive filtering [24]

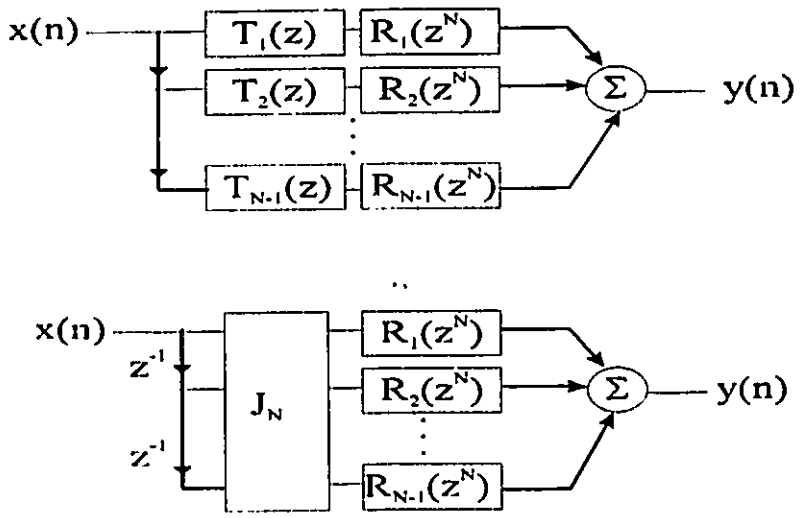


Fig. 2.16 Subband implementations of a filter

III TREE STRUCTURED FILTER BANKS AND THE EQUIVALENT PARALLEL STRUCTURE

3.1 Introduction

A tree structured filter bank contains sets of 2-parallel branch filter banks in which the highpass filter is the mirror image of the lowpass filter (figure 3.1). The highpass filter is derived to satisfy the power complementary condition (2.2), i.e.

$$\sum_i \left|H_i(e^{j\omega T})\right|^2 = 1 \tag{3.a}$$

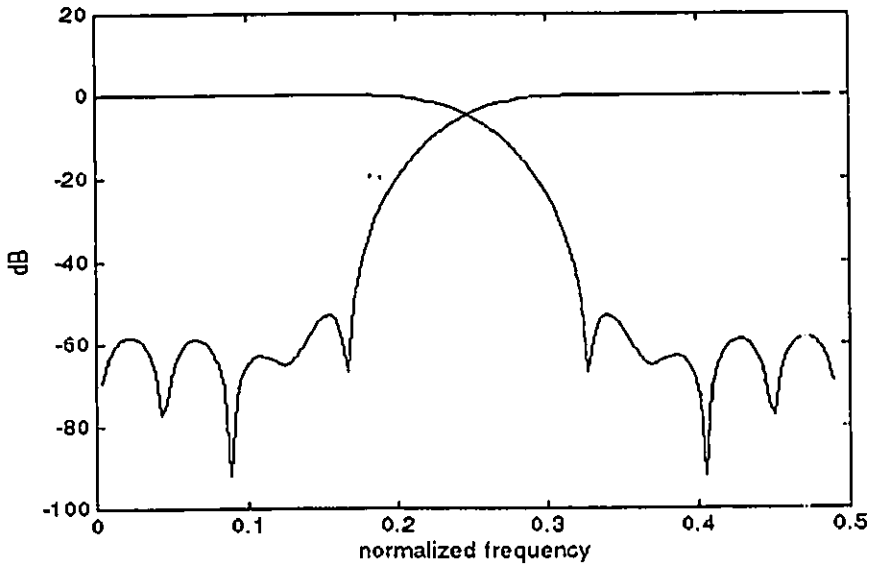


Fig. 3.1 A lowpass filter characteristic and its mirror image

The highpass filter $h_1(n)$ is obtained by inverting the sign of every odd index of the lowpass filter $h_0(n)$, i.e.

$$h_1(n)=(-1)^n h_0(n) \tag{3.1}$$

or in the frequency domain

$$H_1(z) = H_0(-z) \tag{3.2}$$

The design of the filters is always limited to the design of an half-band lowpass filter regardless of the number of stages. The main difference will reside in the ripples (passband and stopband ripples) imposed on $H_0(z)$: the ripples imposed on the lowpass filter designed for one stage tree structured filter bank may be different from the ripples on a lowpass filter designed for four stages tree structured filter bank. In the worst case, if δ_1 (linear) is the ripple on the one-stage lowpass filter, then the ripples on the four-stage lowpass filter will have be $\delta_1/4$ to the first degree approximation for the passband. The ripple in the stopband at the fourth stage however, will be decrease. It will be equal to the ripple of the first stage to the power 4, i.e. if is δ_s (linear) the ripples on the one-stage lowpass filter, then the ripples on the four-stage lowpass filter will be δ_s^4. For the lowpass-highpass filters pair, we must always have

$$\left|H_0(e^{j\omega})\right|^2 + \left|H_1(e^{j\omega})\right|^2 = 1 \tag{3.3}$$

Note that condition (3.3) will not be easily satisfied in most cases. There will be some tradeoffs:

-Achieving (3.3) means that the prototype lowpass filter must have a sharp transition region and small ripples in the passband and stopband regions. The direct consequence will be an increase in the filter length.

-Recall that perfect reconstruction means that the reconstructed signal is a delayed and scaled version of the original signal. In order to satisfy the perfect reconstruction condition (2.1), the analysis and synthesis filters should be designed such that an overall (input-output) linear phase is achieved. The possible design choices are: relax (3.3) as shown in

(3.4) to allow analysis filters with linear phase or maintain (3.3) as it is and design analysis filters with minimal phase distortion.

$$\left|H_0(e^{j\omega})\right|^2 + \left|H_1(e^{j\omega})\right|^2 \approx 1 \tag{3.4}$$

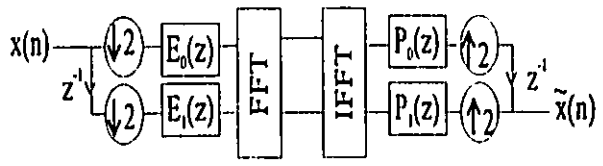
Equations (3.3) and (3.4) show that the frequency responses of the filters $h_0(n)$ and $h_1(n)$ overlap. Thus, one should expect that decimation by 2 will cause aliasing.

Example 3.1

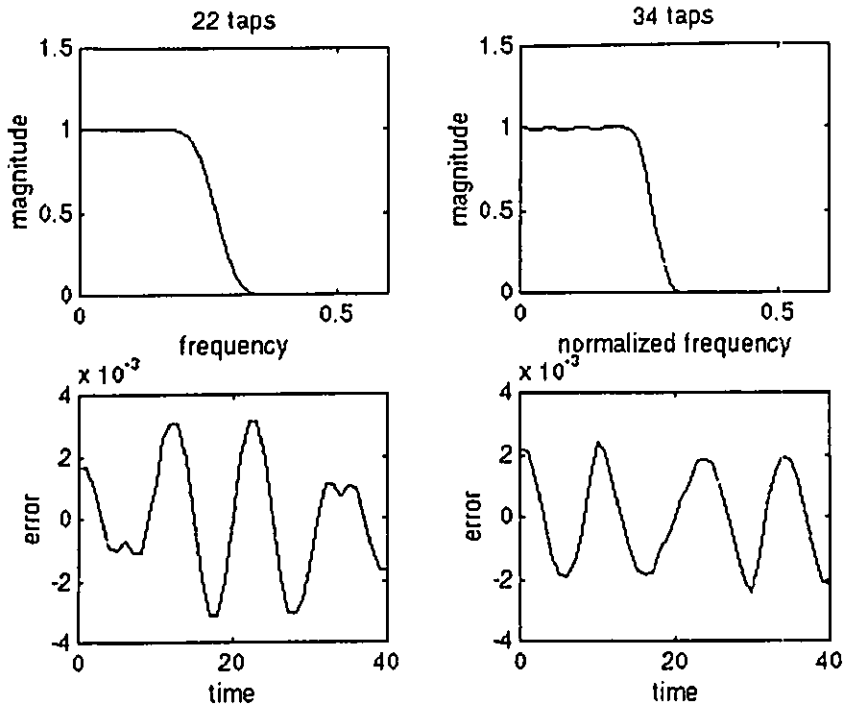
A sinusoid is to be reconstructed using a two level tree structured filter bank (or 4 parallel branches maximally decimated filter bank) with polyphase implementation.

x(n)=sin(0.2πn), n=0,1, ..., 40.

Using the Hamming windowing method, two prototype lowpass filters are designed such (3.4) is satisfied. The filters and the overall ripples are shown below. The filter lengths are 34 taps and 22 taps. The reconstruction error is illustrated in figure 3.2. Note that the filter with 22 taps has a transition region larger than the filter with 34 taps; but as can be seen, the reconstruction errors in both cases are within the same range. The reason for the similar reconstruction errors ($\tilde{x}(n) - x(n)$) is that the energy of the signal in both cases is outside the transition region.



(a)



(b)

Fig. 3.2 Example of reconstruction (a) filter bank, (b) set of filters, (c) reconstruction errors

3.2 Tree Structure Filter Bank and its Equivalent Polyphase Implementation

3.2.1 Parallel Conversion

In the general N-channel maximally decimated filter banks (MDFB), we have the choice of designing each analysis filter separately and obtain a structure with real computations for real-value filters coefficients (this design procedure becomes difficult for filter banks with large number of branches and is thus not generally used) or design only one prototype filter (lowpass filter) and use (2.5) to derive the remaining filters. In the second method (the commonly used one), the computations always require complex operations inherent to the derivation (use of the FFT or DFT matrix). When the tree

structure approach is used, the computations remain real (for real-value filter coefficients) because the elements of the FFT matrix are real. The prototype lowpass filter, which is a half-band lowpass filter, is also easy to design.

Real computations can also be achieved in the case of a maximally decimated filter bank of N parallel channels if the prototype lowpass filter H(z) is designed such that it can be written as in (3.5), i.e. if the overall lowpass filter is derived from a half-band lowpass: this is equivalent to deriving the equivalent lowpass filter (analysis) in a tree structure

$$H(z) = \prod_{i=0}^{s-1} H_0(z^{2^i}) \tag{3.5}$$

where $H_0(z)$ is a half-band lowpass filter and s is an integer related to N by

$$s = \log_2 N \tag{3.6}$$

For s to be an integer, it is required that N be a power of 2. This condition, as will be shown, will allow an implementation using the Hadamard matrix .

Equation (3.5) implies that the designed lowpass filter H(z) can be written as a product of a half-band lowpass filter and its interpolated (upsampled) versions. If the equivalent lowpass filter H(z) is designed such that (3.5) is satisfied, then instead of obtaining the remaining analysis filters using (2.5) or (2.6), we can use the derivation given later in this section. Note that filters satisfying (3.5) for N>2 will not be designed directly. These filters will rather be derived from a given half-band lowpass filter. An example of a filter given by (3.5) is the equivalent analysis lowpass filter of an s-stage tree structure filter bank [6].

In the case of a tree structure filter bank, the derivation of (3.5) is straightforward. In fact, all branch filters will be obtained from the combination of half-band lowpass and

highpass filters: the corresponding implementation becomes thus the equivalent parallel structure realization of the s-stage tree structured filter bank [27]. Shown in figure 3.3 are a 2-stage tree structured filter bank and the equivalent 4-branch parallel structure implementation .

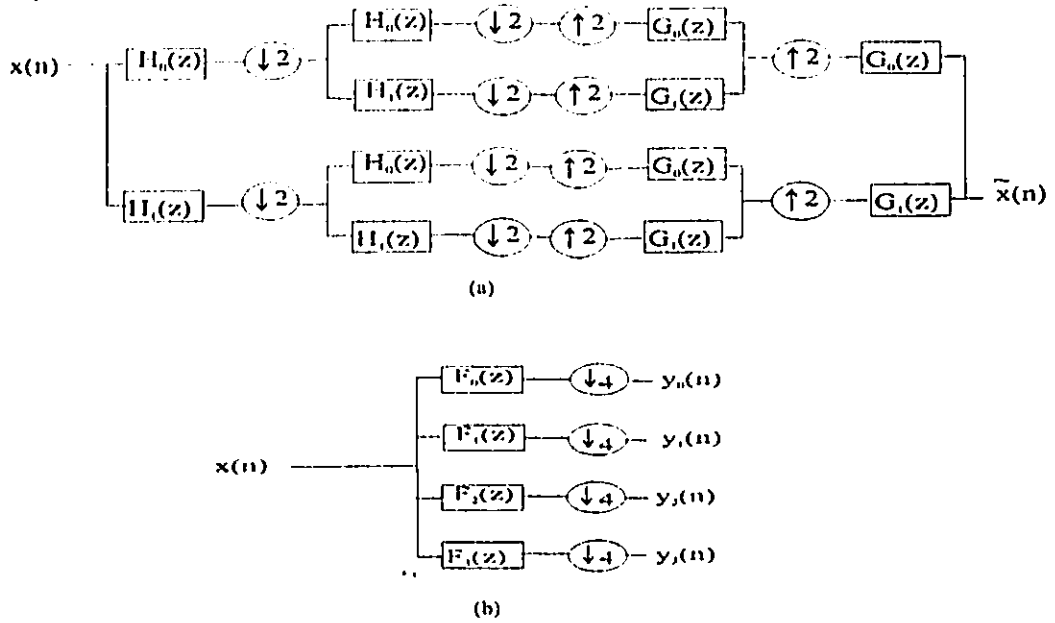


Fig. 3.3 (a) 2-level tree structured filter bank (b) equivalent 4-branch parallel filter bank (analysis bank)

The filters $F_i(z)$ and $H_i(z)$ in the two analysis banks are related as follows, using the z variable of the first stage as the overall z variable

$$\begin{aligned} F_0(z) &= H_0(z)H_0(z^2) \\ F_1(z) &= H_0(z)H_1(z^2) \\ F_2(z) &= H_1(z)H_0(z^2) \\ F_3(z) &= H_1(z)H_1(z^2) \end{aligned} \qquad (3.a)$$

The frequency responses of the above equivalent filters ($F_i(z)$) will be as illustrated in figure 3.4 . Note here that the obtained filters frequency responses are symmetric around f/2 (or π) compared to the frequency response of the filters obtained by the complex modulation (2.5).

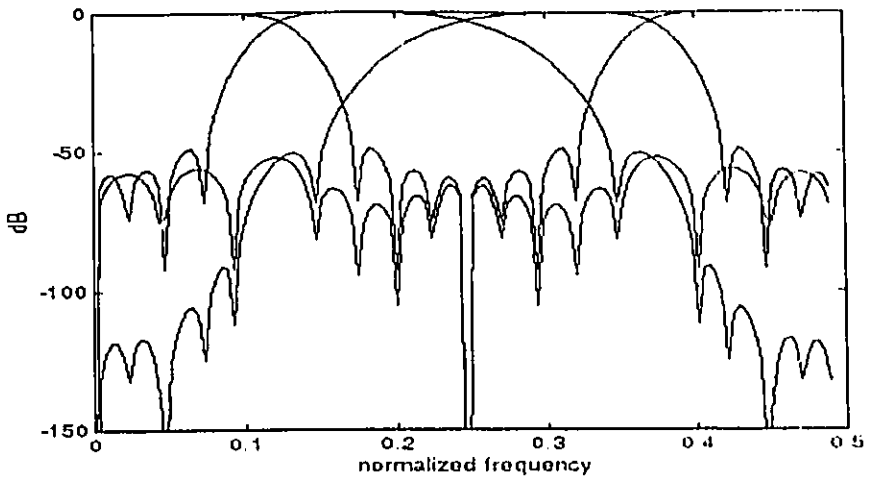


Fig. 3.4 Frequency characteristics axis of filters in figure 3.3 b

With this equivalent implementation approach (figure 3.3b), one notes that if the prototype half-band lowpass filter in the tree structure is of low order, then the equivalent filters in the parallel structure filter bank (figure 3.3b) will be of relatively high order depending on the value of the decimation factor.

Let's rewrite (3.a) using instead the polyphase components of $H_0(z)$,

$$F_0(z) = E_0(z^2)E_0(z^4) + z^{-1}E_1(z^2)E_0(z^4) + z^{-2}E_0(z^2)E_1(z^4) + z^{-3}E_1(z^2)E_1(z^4)$$
$$F_1(z) = E_0(z^2)E_0(z^4) + z^{-1}E_1(z^2)E_0(z^4) - z^{-2}E_0(z^2)E_1(z^4) - z^{-3}E_1(z^2)E_1(z^4)$$
$$F_2(z) = E_0(z^2)E_0(z^4) - z^{-1}E_1(z^2)E_0(z^4) + z^{-2}E_0(z^2)E_1(z^4) - z^{-3}E_1(z^2)E_1(z^4)$$
$$F_3(z) = E_0(z^2)E_0(z^4) - z^{-1}E_1(z^2)E_0(z^4) - z^{-2}E_0(z^2)E_1(z^4) + z^{-3}E_1(z^2)E_1(z^4)$$

so

$$F_i(z) = E_0(z^2)E_0(z^4) \pm z^{-1}E_1(z^2)E_0(z^4) \pm z^{-2}E_0(z^2)E_1(z^4) \pm z^{-3}E_1(z^2)E_1(z^4) \qquad (3.7)$$

or in matrix form,

$$F = Z E M_4 \tag{3.8}$$

where

$$Z = [1 \;\; z^{-1} \;\; z^{-2} \;\; z^{-3}]^T$$

$$E = [E_0(z^2)E_0(z^4) \;\; E_1(z^2)E_0(z^4) \;\; E_0(z^2)E_1(z^4) \;\; E_1(z^2)E_1(z^4)]$$

$$M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

The corresponding implementation is illustrated in figure 3.5

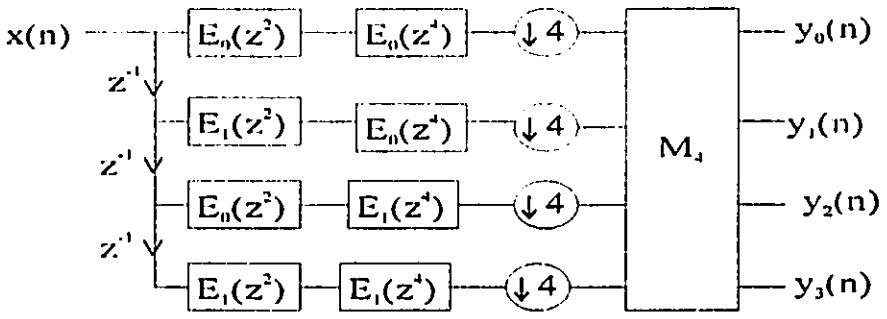


Fig. 3.5 Polyphase implementation of figure 3.3b

There are few points to note :

- the matrix M_4 with elements ± differs from the 4-point FFT matrix (it has no complex elements).

- the decimation factor can be distributed by applying the decimation noble identity of figure 2.9 to obtain the structure of figure 3.6

- the polyphase components are derived from the half-band prototype lowpass filter instead of the equivalent N-th band lowpass filter.

Note also that by rearranging the output (y_i) in the bit reversed order, i.e. $y_0(n)$, $y_2(n)$, $y_1(n)$, $y_3(n)$, it can be shown that the matrix M_4 becomes the 4x4 Hadamard matrix, H_4: indeed by interchanging the second and third columns in M_4, we obtain the Hadamard matrix.

From figure 3.6, one notices that there is a 50% overlap in the branch signals (after the first decimation factor), i.e. if a snapshot of the inputs is taken at times t_{i-1} and t_i , we see that the inputs at t_i contain half of the elements of the inputs at t_{i-1}; this is due to the fact that the number of branches (4) at the input is greater than (actually twice) the decimation factor (2) [first stage] and also the input value x(k) for $k < 0$ is 0. For example with N=4, the output of the first decimators will look like the following::

Sampling time	Branch 0	Branch 1	Branch 2	Branch 3
t_0	x(0)	0	0	0
t_1	x(2)	x(1)	x(0)	0
t_2	x(4)	x(3)	x(2)	x(1)
t_3	x(6)	x(5)	x(4)	x(3)

An example of input signal and the corresponding decimator outputs is illustrated in figure 3.7.

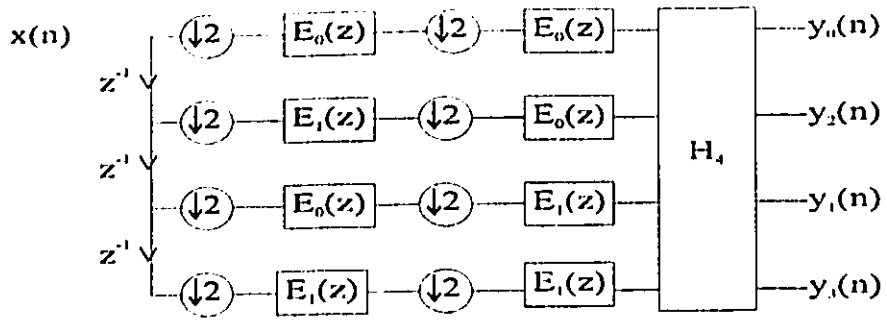


Fig. 3.6 Figure 3.5 with distributed decimation and Hadamard matrix

Note in the example above that the signals in branches 3 and 4 are the delayed versions of the signals in branches 1 and 2 respectively (see also example in figure 3.7).

Also note that the 'subfilters' in these branches are the same: $E_0(z)$ is in branches 1 and 3, and $E_1(z)$ is in branches 2 and 4. The first stage can thus be reduced to 2 branches at the input to yield the implementation shown in figure 3.8.

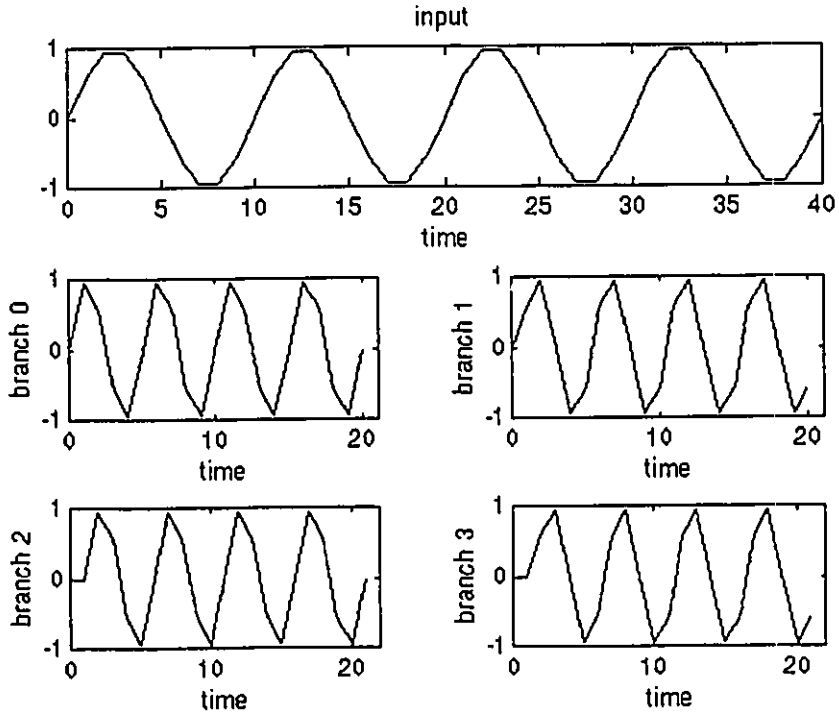


Fig. 3.7 Example of input signal and corresponding branch signals with N=4

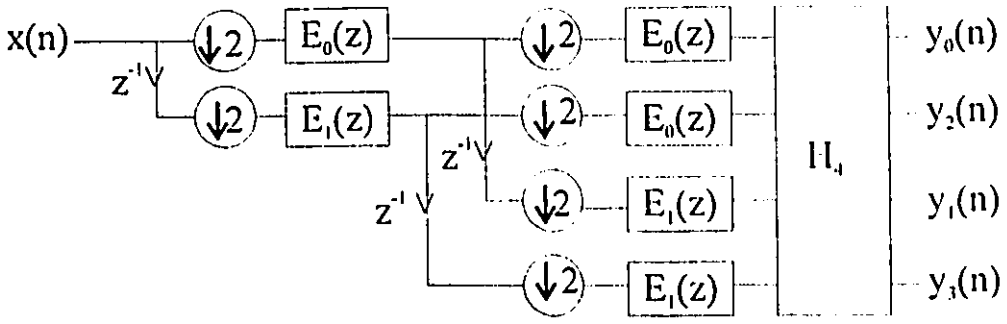


Fig. 3.8 Alternate implementation of figure 3.6

3.2.2 Generalization

The idea developed with N=4 can be generalized for any N (power of 2) as follows:

- Design a prototype half-band lowpass filter suitable for an s-level tree structured filter bank (s is related to N as in equation 3.6).

- Obtain the 2 polyphase components $E_0(z)$ and $E_1(z)$ and arrange them in the bit reversed binary tree manner where 0 and 1 represent $E_0(z)$ and $E_1(z)$ respectively.

- Precede each subfilter $E_i(z)$ by a decimation by 2.

Noting as before that the signals in branches 2k and 2k+1 are the delayed versions of the signals in branches 0 and 1 respectively, one can further simplify the scheme as done in the case with N=4. Repeat the simplification at each stage.

Note that the resulting structure is also a tree structure with polyphase components. This is illustrated by example 3.2.

Example 3.2

Let N=8. We design a half-band LPF suitable for a 3-level tree structured filter bank i.e. if the desired overall ripple is δ_{tot} then the prototype ripples must be $\delta_{tot/3}$. From the

binary combination (000, 001, 010, 011, 100, 101, 110, 111), and thus the reversed binary tree combination (000, 100, 010, 110, 001, 011, 111), we obtain the structure shown in figure 3.9

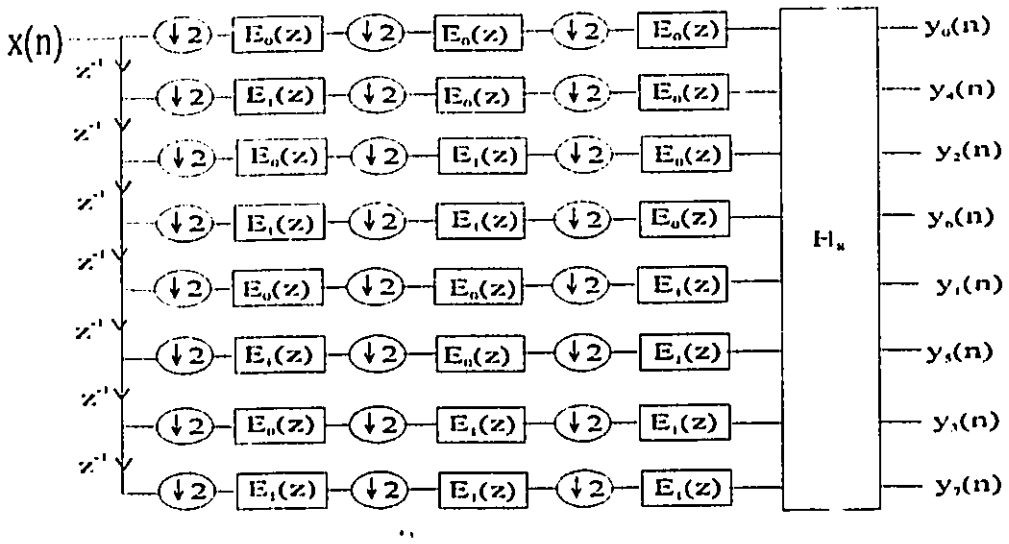


Fig. 3.9 Equivalent realization of 3-level tree structure: analysis bank

By removing the redundant branches in the first stage, figure 3.9 is re-drawn as shown in figure 3.10

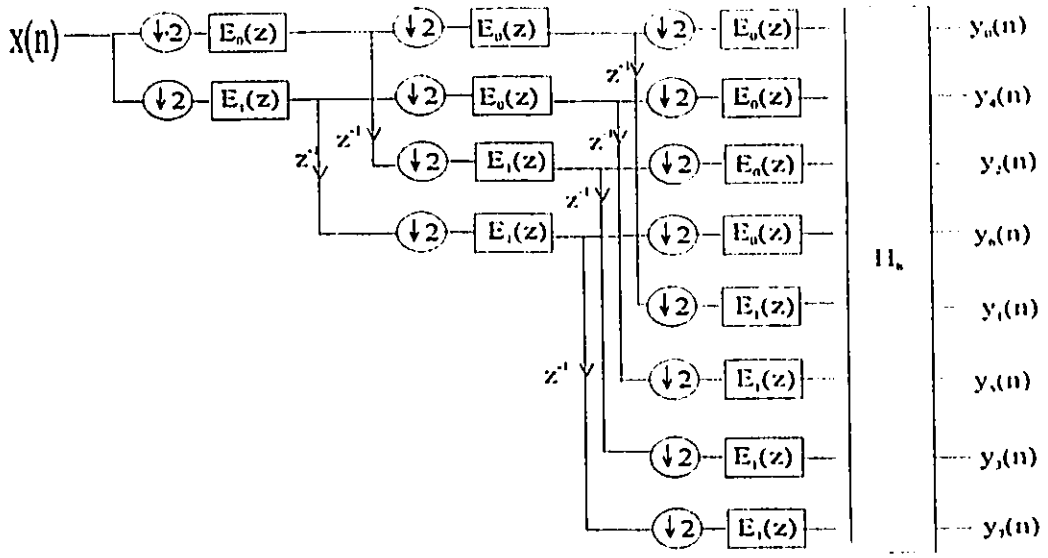


Fig. 3.10 Further simplification of figure 3.9

Perfect reconstruction is achieved if the synthesis filters (polyphase components) relate to the analysis filters (polyphase components) as mentioned in chapter 2.

$$p_k(n) = e_k(L_k - 1 - n), \quad k=0,1 \tag{3.9}$$

where L_k is the length of $e_k(n)$.

Equation 3.9 allows a stage-by-stage perfect reconstruction of the signal for each analysis/synthesis banks pair (for any N).

The inverse of the Hadamard matrix is the same matrix divided by the number of rows or columns. So,

$$H_N^{-1} = \frac{H_N}{N} \tag{3.10}$$

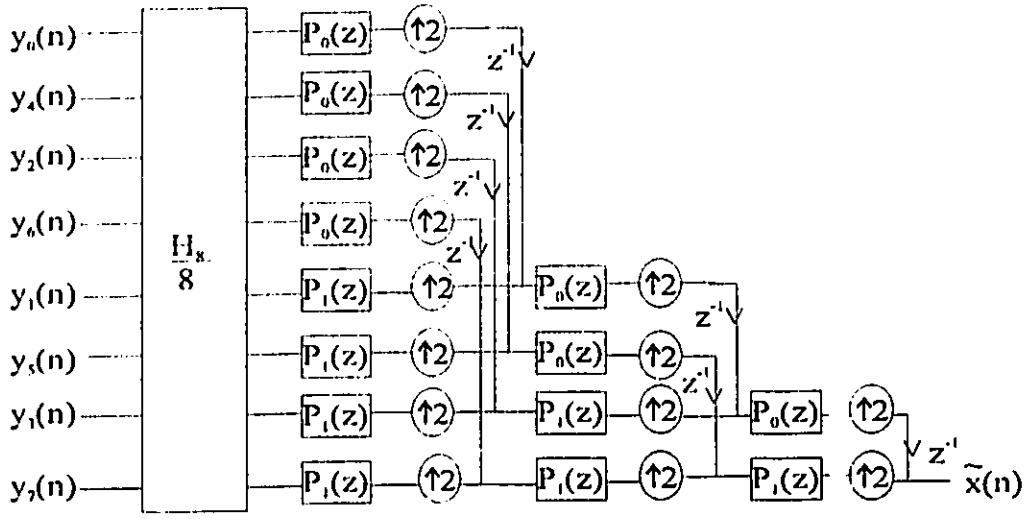


Fig. 3.11 Synthesis filter bank of figure 3.10

The differences between the proposed structure and the structure studied in [23] are as follow:

- a single filter (N-th band filter) is implemented using the Hadamard matrix.
- a possible implementation of a filter bank using that structure might require the use of a combination of FFT and Hadamard matrices. The FFT matrix being introduced by the derivation of the remaining (N-1) branch filters.

3.3 Parallel Structure Implementation From The Equivalent Lowpass Filter Only

Assume that the N-th lowpass filter is derived as given in (3.5) and the remaining filters as in (2.5). In this case, the N polyphase components are obtained not from the half-band lowpass filter but from the equivalent N-th band lowpass filter, i.e. if $f_0(n)$ is the equivalent derived lowpass filter, then the polyphase components become

$$e_k(n) = f_0(n + kN), \qquad k = 0,1,\ldots,N-1 \tag{3.11}$$

From (2.5) and (3.9), we can use the polyphase implementation with the FFT or DFT matrix. This approach has the following two disadvantages:

- the computations require complex operations (due to FFT)
- the overall reconstruction error is higher than in the previous case (example 3.3)

Example 3.3

Let N=4 and $h_0(n)$ be a half-band lowpass filter. Then the two sets of polyphase components will be

$$e_{2,k}(n) = h_0(n + 2k), \qquad k = 0,1 \tag{3.12}$$

$$e_{4,k}(n) = f_0(n + 4k), \qquad k = 0,1,2,3 \tag{3.13}$$

where $f_0(n)$ is derived as in (3.5).

Equations 3.12 and 3.13 will then be implemented as shown in figures 3.12a and 3.12b respectively.

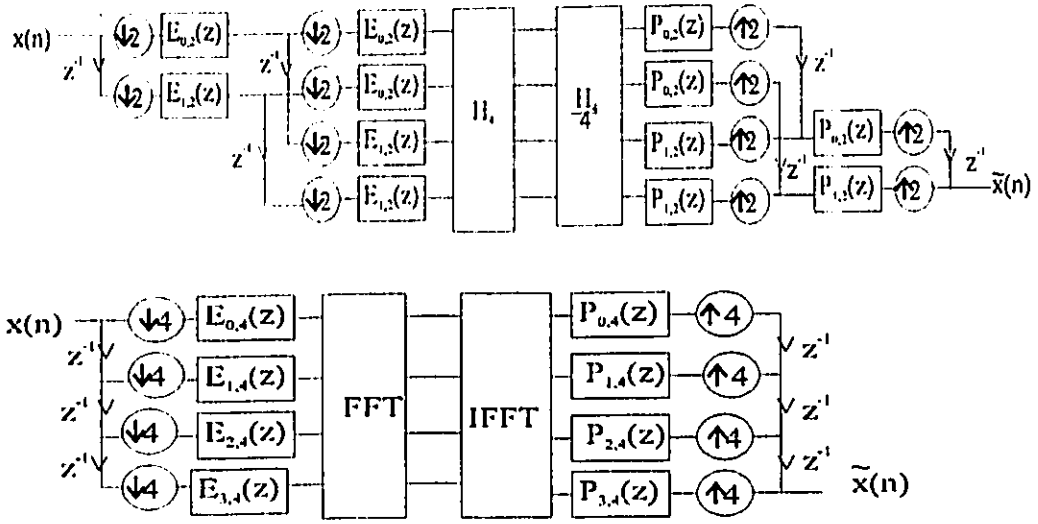


Fig. 3.12 2 Polyphase implementations: (a) with Hadamard, (b) with FFT

An example of input signal and the reconstructed signal errors using figures 3.12a and 3.12b are shown in figures 3.13b and 3.13c respectively. The error is computed as the difference between the original and reconstructed signals, i.e. $\tilde{x}(n) - x(n)$. The prototype half-band lowpass filter is 12-tap filter (12 tap filter (A) selected from [9]).

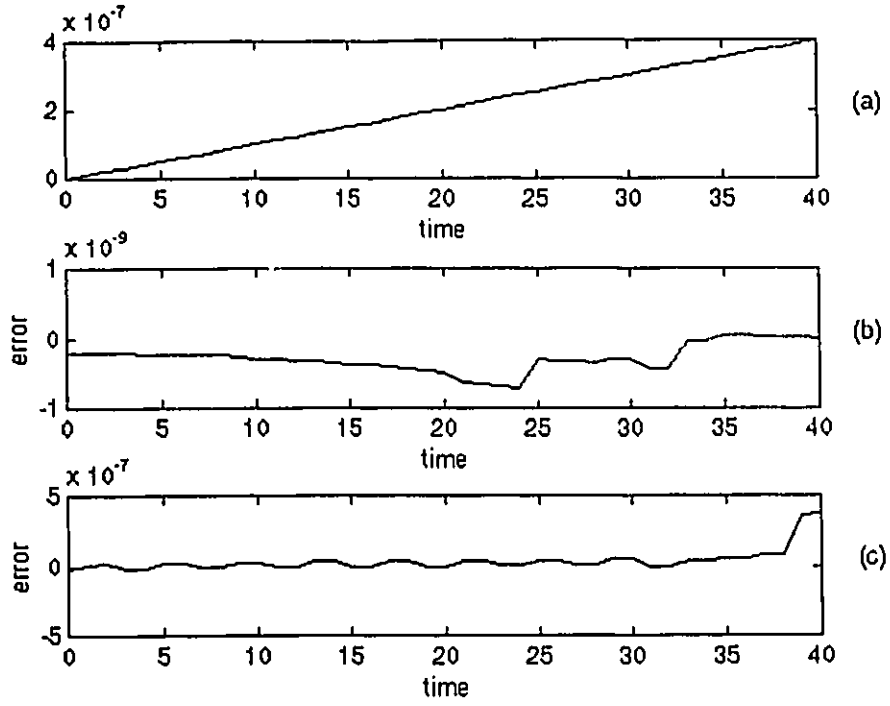


Fig. 3.13 Reconstruction errors: (a) input signal; (b) error with figure 2.12a (c) error with figure 2.12b

Note from figure 3.13 that the error in the case with the FFT (fig.3.13b) is relatively large. This is due to the fact that in the implementation with the FFT, the counter highpass filter which is supposed to compensate for the half-band lowpass filter imperfection is not taken into account. Also the convolution of the lowpass filter by itself (to obtain the fourth-band lowpass filter) will increase the ripples in the stopband. On the other hand, the implementation with the Hadamard matrix achieves better performance because it does employ the lowpass-highpass pair that allows perfect reconstruction. The sum of the magnitudes squared of filters in both systems (with FFT and Hadamard matrices) are shown in figure 3.14.

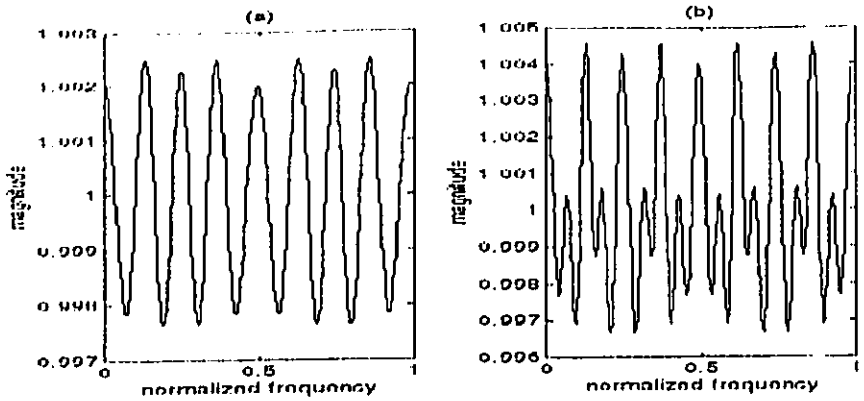


Fig. 3.14 Filters' ripples: (a) lowpass-highpass pair, (b) 4 parallel filters

3.4 Tree Structured Filter Bank Vs N Parallel Structure Filter Bank

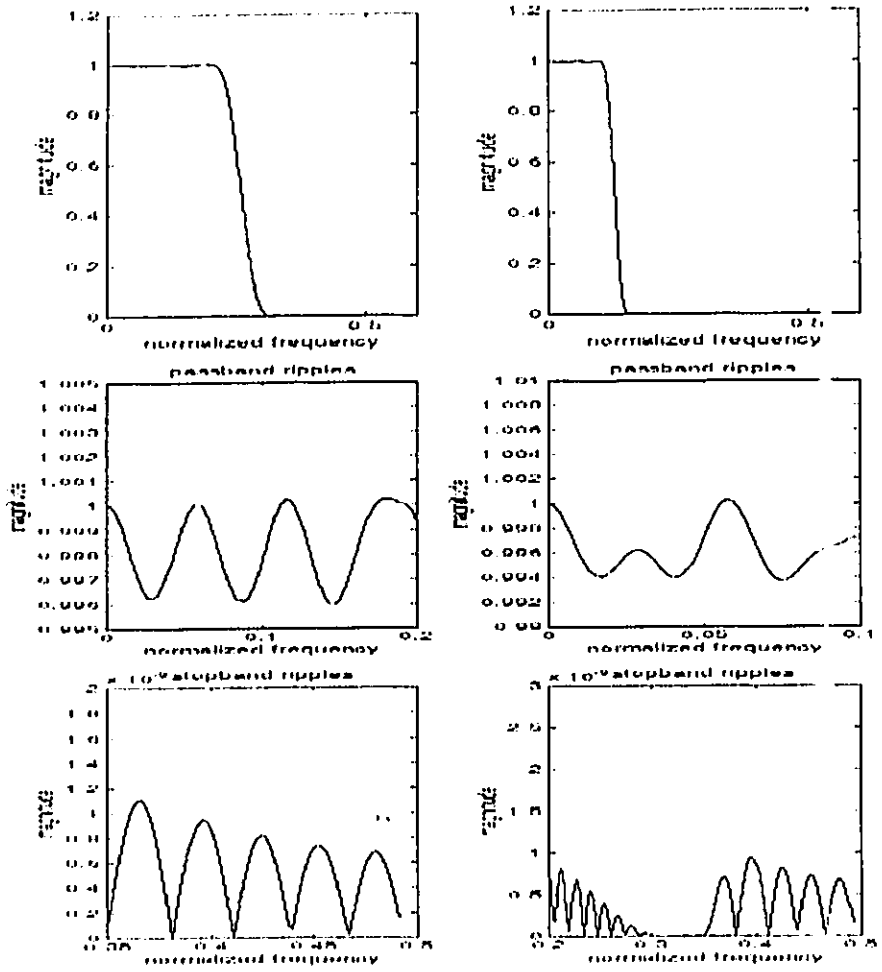
In this section, we consider the comparison between the equivalent N parallel structure filter bank with Hadamard matrix of a tree structured filter bank and an N parallel structure filter bank, i.e: Given N (power of 2) and the desired reconstruction error, one can choose to design a half-band lowpass filter and use the implementation with the Hadamard matrix, or one can design an N-band lowpass filter and use the implementation with the FFT matrix. For simplicity of the notation, we will refer to the structures as Hadamard and FFT respectively.

3.4.1 Input Structure

In the simplified implementation with the Hadamard matrix (figure 3.12a), the first stage at the analysis bank has only two branches at the input but all N branches are present at the input of the synthesis bank. This configuration makes the structure look like a 'pseudo parallel' structure filter bank while the structure with the FFT matrix (fig.3.12b) can be seen as a 'true parallel' structure filter bank because the number of branches throughout the system remains constant and equal to the decimation factor.

3.4.2 Filter Length

It is known that the length of a filter depends on parameters such as the desired ripple amplitude, the decimation factor and the cutoff rate. In the case of a filter bank using the equivalent parallel implementation with the Hadamard matrix, the cutoff frequency is fixed at $f_s/4$ (or $\pi/2$) and does not vary with the decimation factor N. So, only the desired overall ripples and the transition region affect the length of filter. On the other hand, in the filter bank with the FFT matrix, the N-th band filter length depends on N: the cutoff frequency is $f_s/2N$ (or π/N). It is obvious that in that case, the filter will be longer than the half-band lowpass filter. Unfortunately, this is not a fair comparison between the two cases because the implementation with the Hadamard matrix is a multistage implementation while the implementation with the FFT matrix (in which the filter length increases rapidly with N) is a single stage implementation. A comparison based on two different prototype filters (half-band and N-th band lowpass filters) results in the following filter lengths: seen from the analysis bank input with the decimation factor placed after the filter, the length of the filters in the parallel structure with FFT matrix and the tree structure with Hadamard matrix respectively will be approximately equal to L_N (where L_N is the length of the prototype N-th band lowpass filter) and $L_2\sum_{i=0}^{s-1}2^i$ (where L_2 is the length of the half-band filter and s represents the number of stages) respectively . Note that these values left in that form do not compare. So let instead derive all the filters from the same half-band lowpass filter, i.e. the N-th band lowpass filter is derived from the half-band lowpass filter. In this case, the length of the equivalent filters (the equivalent lowpass filters seen from the analysis bank) in both structures will be equal to $L_2\sum_{i=0}^{s-1}2^i$. So even thus the initial length (L_2) is of lower order, the overall filter length can be large for larger N and the ripples in the passband and stopband vary (figure 3.15).



(a) (b)

Fig. 3.15 Passband and stopband ripples with a 43-tap lowpass filter
(a) half-band lowpass filter
(b) equivalent fourth-band lowpass

3.4.3 Complexity

In the parallel structured filter bank with the FFT matrix (figure 2.11), we have one single stage working at f_s/N. Each polyphase component being of length L_N/N approximately, we have per branch $2L_N/N$ real operations plus the FFT penalty. Overall, there will be L_N real operations plus $1.5N\text{Log}_2N$ complex operations (due to the FFT) per output sample [6]

In an implementation as the one shown in figure 3.6, we have NL_2 real operations per stage. The overall computation requires approximately $2NL_2\sum_{i=1}^{s}2^i$ real operations (where s represents the number of stages) plus 2(N-1) additions due to the Hadamard matrix per output sample. The number of operations is reduced when the simplified structure is used (figure 3.8). Indeed, by reducing the number of branches at each stage, we achieve a saving in computation: $2L\sum_{i=1}^{s}2^{2i}+2N-2$ real operations.

3.4.4 Delay

We have assumed

$$h_{synth}(n) = h_{anal}(L-1-n) \tag{3.14}$$

or with the polyphase components

$$p_i(n) = e_i(L_i-1-n) \tag{3.15}$$

and the overall delay (D) with the N-parallel branches with FFT matrix (figure 2.11) is

$$D_{FFT} = L_N \tag{3.16}$$

where L_N is the length of the N-th band lowpass filter

The cascade nature and the distributed decimation by 2 of the system with the Hadamard matrix cause the delay to double from stage to the next (figure 3.8). The overall delay is given by

$$D_{HAD} = (N-1)L_2 - (N-2) \tag{3.17}$$

where L_2 is the length of the half-band lowpass filter.

For example, for N=4, if L_N = 180 and L_2 = 40 then the delays will be D_{FFT} = 180 sampling time units and D_{HAD} = 118 sampling time units

3.5 Summary

In the implementation of a uniform bandwidth filter bank, the tree structure or parallel structure is used, depending on the particular application. The tree structured filter bank is limited to values of N being a power of 2, while the N-parallel branch filter bank can have any number of branches. In either case, the polyphase implementation is preferred because of the saving in computation that is achieved. But overall, in the case of N a power of 2, the equivalent parallel structure with the Hadamard matrix will be preferred over the N parallel branch with the FFT matrix, because of the computation savings and reduced processing delay.

IV Bandwidth Effect: Tree Structure Vs Parallel Structure Filter Bank

4.1 Introduction

In the previous chapter, an equivalent implementation of the tree structured filter bank was introduced. This implementation uses the polyphase components of the prototype lowpass filter followed by a Hadamard matrix instead of an FFT matrix, leading to an overall computation with real coefficients: the filter coefficients and the elements of the Hadamard matrix. A comparison (delay, computation complexity, ..) with the parallel structure filter bank constructed of polyphase components was made in chapter 3. Note that the comparison was strongly influenced by the filter design: the design method, the filter order, the design specifications (cutoff frequencies, ripples). In an N branch maximally decimated filter bank where the prototype filter is an N-th band lowpass filter, the filter can be optimized to allow a good reconstruction. In this case, optimizing the filter coefficients can be very complex for N large.

In this chapter, given a decimation factor N (which will be a power of 2), two prototype lowpass filters are designed: one half-band lowpass filter to be used with the Hadamard matrix (polyphase tree structure) and one N-th band lowpass filter to be used with the FFT matrix (polyphase parallel structure). Two different approaches will be used. They are as follow

- Given the input signals, the prototype filters will be designed so that the reconstruction errors are within a given range. Note that the filter design method is different from the conventional design approach where the desired cutoff frequencies and ripples are provided to the algorithm and a filter length estimate is returned (along with the

coefficients). Here, the filter length and passband cutoff frequency are provided to the algorithm.

- Given the prototype filters, signals of different bandwidth will be processed. The reconstruction errors of both structures will be compared.

4.2 Definitions

Let's define f_b to be the fractional bandwidth of a given signal as the ratio of the signal bandwidth B over the bandwidth of the lowpass filter W i.e.

$$f_b = \frac{B}{W} \tag{4.1}$$

where

W = $f_s/4$ for the half-band lowpass filter

W = $f_s/2N$ for the N-th band lowpass filter

As mentioned in [26], a signal is narrowband when $f_b < 0.1$

Note that all the bandwidths mentioned here refer to the one-side bandwidths (figure 4.1). Note also that the bandwidth of the prototype filter in the tree structured filter bank will always be $\pi/2$ or $f_s/4$ (ideally) regardless of the value of N, while in the parallel structure, the bandwidth varies with N (π/N or $f_s/2N$). So for a given signal, the fractional bandwidth will be constant in case of the tree structure while it will vary with N in the case of the parallel structure.

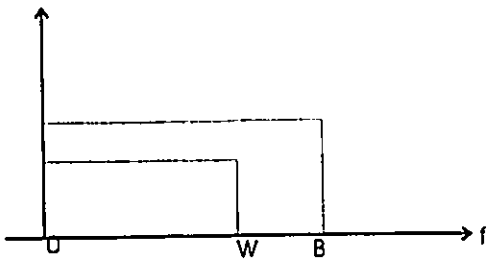


Fig. 4.1 Signal and lowpass filter bandwidths

4.3 Prototype Filter Versus Reconstruction Error

In the following sections, different signals will be processed[1] using an example of a 2-stage polyphase tree structured filter bank with the 4x4 Hadamard matrix and a 4-parallel branch maximally decimated filter bank with the 4x4 FFT matrix So the overall decimation factor will be 4 for both structures. Two cases will be considered:

- Design different prototype lowpass filters (non optimal)for each structure.
- Design one half-band lowpass filter (optimal and non optimal) for both structures.

The signals are pulse trains (with increasing amplitude) with the number of impulses varying from 1 to 6. The reason for this choice of signals is because it is easy to vary the signal bandwidth (by varying the number of pulses). The signals are sampled at 1 GHz.

4.3.1 Processing With a Filter Bank Using Hadamard Matrix

With the filter bank as shown in figure 4.2 (N=4), we design a prototype lowpass filter based on the Hamming[2] windowing method. In this design, note that instead of computing the filter length based on the desired filter ripples, the passband and stopband cutoff frequencies, the designer has to provide a filter length and a cutoff frequency to the

[1] See Appendix A for program listing
[2] This filter is designed using the FIR1 function in MATLAB with the Hamming windowing technique

algorithm. This design procedure has the advantage of letting the user fix the desired filter length. In our case, the designed filter is 22-tap lowpass filter with passband cutoff frequency of 0.5373π (the algorithm given in Appendix B is used to obtain the filter).

The frequency responses of the different input signals (S1, S2, S3, S4, S5, S6) are illustrated in figure 4.3. In Table 4.1, the filter length, the mean, standard deviations and RMS value of the reconstruction error are listed.

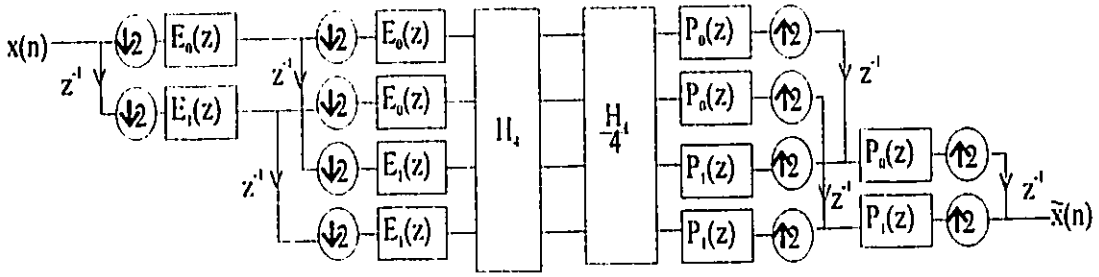


Fig. 4.2 Filter bank with Hadamard matrix

Table 4.1: Tree Structure with Hadamard Matrix(L=22, $w_c=0.5373\pi$)

	Error		
Input signal	mean	std deviation	RMS
S1	$<10^{-25}$	$<10^{-25}$	$<10^{-25}$
S2	-1.03×10^{-25}	1.46×10^{-25}	1.46×10^{-25}
S3	4.58×10^{-14}	1.21×10^{-13}	1.09×10^{-13}
S4	8.59×10^{-14}	1.21×10^{-13}	1.35×10^{-13}
S5	7.19×10^{-13}	2.37×10^{-12}	2.24×10^{-12}
S6	1.32×10^{-12}	2.98×10^{-12}	3.02×10^{-12}

Note from the table that the reconstruction errors (mean, standard deviation and root mean squared values) increase as the number of pulses increases.

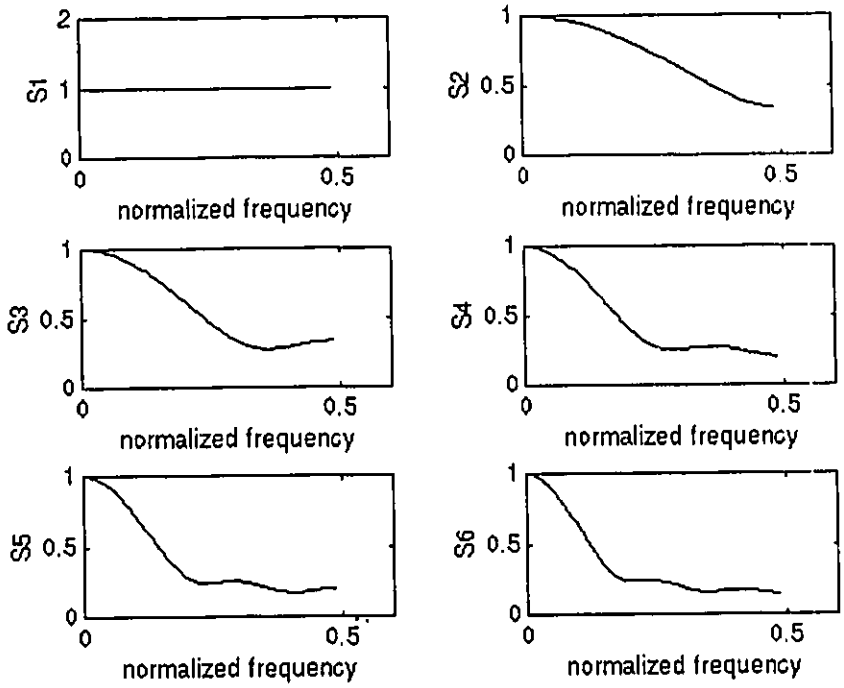


Fig. 4.3 Spectra of the input signals with normalized magnitude

4.3.2 Processing With a Filter Bank Using FFT Matrix

For comparison purpose, we chose N to be equal to 4 as in the previous case. The filter bank is illustrated in figure 4.4. A fourth-band lowpass filter is also designed using the Hamming windowing technique: the filter is obtained using the algorithm given in Appendix B. As mentioned earlier, this is not an optimal design.

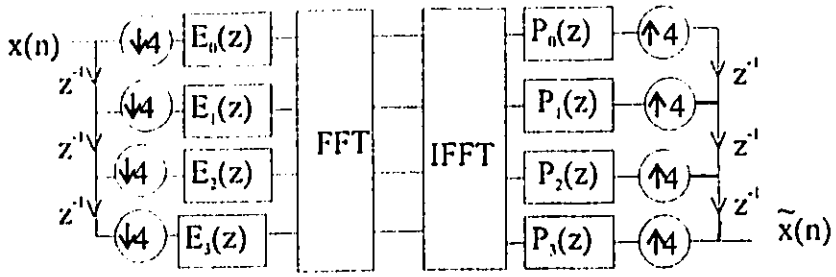


Fig. 4.4 Filter bank with FFT matrix

The different input signals are illustrated in figure 4.3. In Table 4.2, the mean, standard deviations and RMS value of the reconstruction error are listed.

Table 4.2a: Parallel Structure with FFT Matrix(L=212, w_c=0.2506π)

	Error		
Input signal	mean	std deviation	RMS
S1	$<10^{-25}$	$<10^{-25}$	$<10^{-25}$
S2	-1.03×10^{-25}	1.46×10^{-25}	1.46×10^{-25}
S3	2.44×10^{-11}	2.44×10^{-11}	3.15×10^{-11}
S4	1.83×10^{-11}	2.34×10^{-11}	2.73×10^{-11}
S5	-4.99×10^{-11}	5.47×10^{-11}	6.99×10^{-11}
S6	-5.80×10^{-11}	6.01×10^{-11}	7.99×10^{-11}

Note here also that the reconstruction errors (mean, standard deviation and root mean squared values) increase as the number of pulses increases.

Let now instead derive the fourth-band lowpass filter from the 22-tap half-band lowpass filter (from table 4.1), i.e. the half-band lowpass filter is cascaded with its interpolated by 2 version to obtain the fourth-band lowpass filter. The reconstruction errors in this case as illustrated in table 4.2b

Table 4.2b: Parallel Structure with FFT Matrix(L=22, w_c=0.5373π)

Input signal	Error		
	mean	std deviation	RMS
S1	$<10^{-25}$	$<10^{-25}$	$<10^{-25}$
S2	$<10^{-25}$	$<10^{-25}$	$<10^{-25}$
S3	-2.76×10^{-10}	2.76×10^{-10}	3.60×10^{-10}
S4	-2.07×10^{-10}	2.64×10^{-10}	3.09×10^{-10}
S5	3.7×10^{-10}	3.74×10^{-10}	4.99×10^{-10}
S6	3.81×10^{-10}	3.36×10^{-10}	4.88×10^{-10}

4.3.3 Reconstruction Error Using one Optimal Half-Band Filter

Using the same filter bank structures as illustrated in figures 4.2 and 4.4, the 64-tap(D) optimal half-band lowpass is chosen among the filters published in [9] to reconstruct the same signals (figure 4.3). The results are given in table 4.3

Table 4.3a: Tree Structure with Hadamard Matrix

Input signal	Error		
	mean	std deviation	RMS
S1	$< 10^{-25}$	$< 10^{-25}$	$< 10^{-25}$
S2	$<10^{-25}$	$<10^{-25}$	$<10^{-25}$
S3	2.01×10^{-15}	5.33×10^{-15}	4.79×10^{-15}
S4	3.77×10^{-15}	5.30×10^{-15}	5.94×10^{-15}
S5	9.74×10^{-15}	1.88×10^{-14}	1.94×10^{-14}
S6	1.54×10^{-14}	2.22×10^{-14}	2.55×10^{-14}

Table 4.3b: Parallel Structure with FFT Matrix

Input signal	Error		
	mean	std deviation	RMS
S1	10^{-9}	$< 10^{-25}$	10^{-9}
S2	$1.50\text{x}10^{-9}$	$7.07\text{x}10^{-10}$	$1.58\text{x}10^{-9}$
S3	$6.97\text{x}10^{-10}$	$2.34\text{x}10^{-9}$	$2.03\text{x}10^{-9}$
S4	$1.40\text{x}10^{-9}$	$2.68\text{x}10^{-9}$	$2.71\text{x}10^{-9}$
S5	$2.11\text{x}10^{-9}$	$2.85\text{x}10^{-9}$	$3.31\text{x}10^{-9}$
S6	$1.18\text{x}10^{-9}$	$2.77\text{x}10^{-9}$	$2.79\text{x}10^{-9}$

4.5 Summary

Note from the tables that:

- the reconstruction errors (mean, standard deviation, root mean squared values) in all cases increase as the number of impulses increases, i.e. from S1 to S6 where Si means that the input signal has i impulses.

- the tree structured filter bank with polyphase components and Hadamard matrix allows real computations just like the 'original' tree structured filter bank. The multiplications in the computations result from the filtering operations only, the Hadamard matrix requiring additions only. The number of operations will be $2NL + 1.5N\log_2 N$ complex operations per output sample for the parallel structure filter bank and $4L\sum_{i=1}^{s} 2^{2i-1}$ + 2N-2 real operations for the tree structure with the Hadamard matrix.

- when the same prototype half-band lowpass filter is used in both the tree structured filter bank with Hadamard matrix and the parallel structure filter bank with FFT matrix, better reconstruction error is achieved with the tree structure. In this case, the reconstruction error decreases rapidly as the signal bandwidth increases, from S3 to S1 for example. For the same signals, note that the reconstruction error using the parallel structure filter bank (with FFT) is relatively higher and almost constant.

Another advantage of the tree structured filter bank is that the same filter can be used in other filter banks (tree structure) with different number of stages, i.e. when a good half-band prototype lowpass filter is designed (small ripples in passband and stopband, narrow transition region), it can be used in a tree structure filter bank other a two-stage two stages tree structure filter bank. This shows the flexibility of the structure.

Overall, the tree structured filter bank is preferred over the parallel structure filter bank for wideband signals processing.

V Conclusion and Suggestions for Further Research

The thesis can be divided in three parts: in chapter II, maximally decimated filter banks were reviewed. These filter banks comprise the N-parallel branch maximally decimated filter banks and the tree structured filter banks. In the case of the parallel structure, two types were reviewed:

1) filter bank with modulated (exponential modulator) input signal: the filters in the branches are lowpass filters (same filter in general) while the signal at the input of the analysis bank is multiplied by an exponential which corresponds to a frequency shift or a rotation of the signal in the frequency domain.

2) filter bank with modulated filter (exponential modulator): the filters in other branches are derived from the same lowpass filter by exponential modulation. In this case, emphasis was put on the polyphase implementation of this filter bank. The relation between the polyphase components and the general subband components was given.

In chapter III, the focus was the tree structured filter banks. By expanding the overall equivalent filter in each branch of the 2-level tree structured filter bank, a new implementation was derived. This implementation used the 2x2 Hadamard matrix with a rearrangement of the output signals at the analysis bank. Then the method was extended to arbitrary N power of 2. It has been noticed that the structure is simple and is implemented using a low order filter (the prototype lowpass filter is an half-band filter). The number of real operations, i.e. multiplications and additions (the computations throughout the filter bank remain real) per analysis output sample is $2L\sum_{i=1}^{s} 2^{s-i} + N\text{-}1$ (additions due the Hadamard matrix) where L is the length of the prototype lowpass filter and s is the number of stages in the tree. Overall, the required computations using the proposed structure remain all real (for real input signal and filter with real coefficients) and less than the computations in the case of MDFB (complex computation when the number

of branches is greater than 2): The computation complexities are $2NL + 1.5N\log_2 N$ complex operations per output sample for the parallel structure filter bank compared to $4L\sum_{i=1}^{s} 2^{2i-1}$ + 2N-2 real operations for the proposed structure per output sample

In chapter IV, a simulation was run to show the performance of the new structure compared to the parallel structure filter bank. The simulation showed that:

- the design of the prototype lowpass filter which is a half-band lowpass filter is simple in the case of tree structured filter bank or its equivalent structure.
- in general, the overall input/output delay of new structure is smaller than that of the N-parallel maximally decimated filter bank (MDFB)
- the number of operations for the parallel structure filter bank with N=4 and L=212 will be 1696 real operations plus 12 complex operations per output sample. With L=34 for the tree structure filter bank with the Hadamard matrix, these numbers will be 1366 real operations. Note that for this particular case, the hardware realization of the proposed structure will be ten times faster than the parallel structure. So, by reducing the number of operations per sampling frequency, signals with larger bandwidth can be allowed.
- better reconstruction error is achieved (proposed structure) for wideband signals.

Table 5.1: Summary table

	Tree Structured Filter Bank With Hadamard Matrix	Parallel Branch Filter Bank With FFT Matrix
Ease of prototype design	yes	decimation factor dependent
Processing delay	(N-1)L_2-(N-2)	L_N
Computation	real	complex
Computations/output sample	$4L\sum_{i=1}^{s} 2^{2i-1}$ + 2N-2	$2NL + 1.5N\log_2 N$
Wideband signal processing	yes	not fully

The research in the thesis can be extended by determining the real processing delay based on knowledge of the of the speed of the devices to be used for the implementation of the filter banks. Also, implementation of filter bank for multiresolution analysis using one analysis and one synthesis matrices can be investigated.

APPENDIX A

```
function z = addvec(x,y)
% written by Akpa M.
%
% z = addvec(x,y), where x and y vectors, adds the two vectors
% by first padding the shorter vector with zeros .
% example: z = addvec([1 2 3],[3 4]) = [1 2 3] + [3 4 0] = [4 6 3]
Lx = length(x);
Ly = length(y);
h = max(Lx,Ly);
y = [y,zeros(1,h-Ly)];
x = [x,zeros(1,h-Lx)];
z = x + y;


function v = combine4(w,x,y,z)
% written by Akpa M.
%
% combine4 is similar to combine2 except that it uses 4 vectors
% make sure you enter the arguments starting with the most delayed one
% example: combine4(x1,x2,x3,x4) will delay x1 by 3, x2 by 2, x3 by 1.
% see also COMBINE2 and COMBINE8

w = delay(w,3);
x = delay(x,2);
y = delay(y,1);
lx = length(x);
ly = length(y);
lz = length(z);
lw = length(w);
h = [lw lx ly lz];
m = max(h);
w = [w,zeros(1,m-lw)];
x = [x,zeros(1,m-lx)];
y = [y,zeros(1,m-ly)];
z = [z,zeros(1,m-lz)];
v = w + x + y + z;
```

```
function y = decim(x,S)
% written by Akpa M
%
% y = decim(x,S) decimates the vector x by S where S is an integer
% example: x = [1 2 3 4 5 6], y = decim(x,2) = [1 3 5]
% for decimation by a non-integer see DECIM2

L = length(x);
y = x(1:S:L);


function y = delay(x,S)
% written by Akpa M.
%
% y = delay(x,S) delays the vector x by S where S is an integer
% example: y = delay([1 2 3 4],2) = [0 0 1 2 3 4]

y = [zeros(1,S),x];


function [e0,e1]=equleng4(e0,e1)
% Written by Akpa M.
%
%Returns the set of 2 vectors with the same length
%by padding the shorter vector with zeros
%Example: if e0=[1 2 3 4] and e1=[5 6] then
%[e0,e1]=equleng2(e0,e1) returns
%e0=[1 2 3 4] and e1=[5 6 0 0]
%
%see also EQULENG4 and EQULENG8

l0=length(e0);l1=length(e1);
l=max([l0 l1]);
e0=zeropad(e0,l-l0);
e1=zeropad(e1,l-l1);
```

```
function [e0,e1,e2,e3]=equleng4(e0,e1,e2,e3)
% Written by Akpa M.
%
%Returns the set of 4 vectors with the same length
%
%see also EQULENG4 and EQULENG8

l0=length(e0);l1=length(e1);l2=length(e2);l3=length(e3),
l=max([l0 l1 l2 l3]);
e0=zeropad(e0,l-l0);
e1=zeropad(e1,l-l1);
e2=zeropad(e2,l-l2);
e3=zeropad(e3,l-l3);


function y=paral4(x,h)
%Written by Akpa M.
%
%Returns the output of the 4-parallel branch
%maximally decimated filterbank using the
%polyphase implementation
%syntax: y=paral4(x,h) where x is the input
%signal to be processed and h a forth-band
%lowpass filter.
%
%see also APARAL4

[e0,e1,e2,e3]=polphas4(h);
x0=decim(x,4);
x1=decim(delay(x,1),4);
x2=decim(delay(x,2),4);
x3=decim(delay(x,3),4);
v0=conv(x0,e0);
v1=conv(x1,e1);
v2=conv(x2,e2);
v3=conv(x3,e3);
[v0,v1,v2,v3]=equleng4(v0,v1,v2,v3);
L1=length(v0);
for i=1:L1,
   ft=fft([v0(i) v1(i) v2(i) v3(i)],4);
   y0(i)=ft(1);
   y1(i)=ft(2);
   y2(i)=ft(3);
   y3(i)=ft(4);
end
```

```
for i=1:L1,
   f1=ifft([y0(i) y1(i) y2(i) y3(i)],4);
   v0(i)=f1(1);
   v1(i)=f1(2);
   v2(i)=f1(3);
   v3(i)=f1(4);
end
f0=e0(length(e0):-1:1);
f1=e1(length(e1):-1:1);
f2=e2(length(e2):-1:1);
f3=e3(length(e3):-1:1);
x0=conv(f0,v0);
x1=conv(f1,v1);
x2=conv(f2,v2);
x3=conv(f3,v3);
x0=upsam(x0,4);
x1=upsam(x1,4);
x2=upsam(x2,4);
x3=upsam(x3,4);
y0=addvec(delay(x0,1),x1);
y1=addvec(delay(y0,1),x2);
y=addvec(delay(y1,1),x3);
y=real(y)*max(x)/max(real(y));..
l=length(h);
y=y(l:l+length(x)-1);


function [e0,e1] = polphas2(h)
% Written by Akpa M.
%
%Returns the 2 polyphase components of a given vector h;
%Example: [e0,e1]=polphas2([1 2 3 4 5 6 7]) returns
%e0=[1 3 5 7] and e1=[2 4 6];
%
%see also POLPHAS4 and POLPHAS8

l=length(h);
e0=h(1:2:l);
e1=h(2:2:l);
```

```
function [e0,e1,e2,e3]=polphas4(h)
% Written by M. Akpa
%
%Returns the 4 polyphase components of a given vector h;
%
%see also POLPHAS2 and POLPHAS8

l=length(h);
e0=h(1:4:l);
e1=h(2:4:l);
e2=h(3:4:l);
e3=h(4:4:l);
```

```
function y=tree2(x,h)
%Written by Akpa M.
%
%Returns the output of the two-stage tree
%structure filterbank specified by the input
%signal x and the half-band lowpass filter h.
%
%see also ATREE1, ATREE2, STREE1, STREE2

x0=decim(x,2); x1=decim(delay(x,1),2);
x2=decim(delay(x,2),2); x3=decim(delay(x,3),2);
[e0,e1]=polphas2(h);
v0=decim(conv(x0,e0),2); v1=decim(conv(x1,e1),2);
v2=decim(conv(x2,e0),2); v3=decim(conv(x3,e1),2);
w0=conv(v0,e0); w1=conv(v1,e0); w2=conv(v2,e1); w3=conv(v3,e1);
h4=hadamard(4);
[w0,w1,w2,w3]=equleng4(w0,w1,w2,w3);
lm=length(w0);
for i=1:lm,
   yy=[w0(i) w1(i) w2(i) w3(i)]*h4;
   y0(i)=yy(1); y1(i)=yy(2); y2(i)=yy(3); y3(i)=yy(4);
end
for i=1:lm,
   yy=[y0(i) y1(i) y2(i) y3(i)]*h4/4;
   w0(i)=yy(1); w1(i)=yy(2); w2(i)=yy(3); w3(i)=yy(4);
end
g0=e0(length(e0):-1:1); g1=e1(length(e1):-1:1);
v0=upsam(conv(g0,w0),2); v1=upsam(conv(g0,w1),2);
v2=upsam(conv(g1,w2),2); v3=upsam(conv(g1,w3),2);
```

```
x0=upsam(conv(g0,v0),2); x1=upsam(conv(g1,v1),2);
x2=upsam(conv(g0,v2),2); x3=upsam(conv(g1,v3),2);
x01=addvec(delay(x0,1),x1);
x12=addvec(delay(x01,1),x2);
y=addvec(delay(x12,1),x3);
y=y*max(x)/max(y);
l=length(h)*3 - 2;
y=y(l:l+length(x)-1);


function y = upsam(x,S)
% written by CSB
%
% y = upsam(x,S) interpolates the vector x by S where S is an integer
% example: x = [1 2 3], y = upsam(x,3) = [1 0 0 2 0 0 3]

L = length(x);
y(:) = [x;zeros(S-1,L)]; y = y.';
y = y(1:S*L-1);
```

APPENDIX B

TREE STRUCTURE

```
x=                      //enter the input signal
for N=19:99             // loop for the filter length
  for W=0.5:0.0001:0.56  // loop for the cutoff frequency
     h=fir1(N,W);       // design of the prototype filter
     y=tree2(x,h);      // reconstructed signal
     e1=x-y;            // error
     e=max(abs(e1));
     if e <= 10^-6      // test for maximum error
      N                 // estimate of filter length
      W                 // estimate of filter cutoff frequency
     end
  end
end
```

PARALLEL STRUCTURE

```
x=                      //enter the input signal
for N=21:4:299          // loop for the filter length
  for W=0.25:0.0001:0.26 // loop for the cutoff frequency
     h=fir1(N,W);       // design of the prototype filter
     y=paral4(x,h);     // reconstructed signal
     e1=x-y;            // error
     e=max(abs(e1));
     if e <= 10^-6      // test for maximum error
      N                 // estimate of filter length
      W                 // estimate of filter cutoff frequency
     end
  end
end
```

References

[1] P. P. Vaidyanathan, "Theory and Design of M-Channel Maximally Decimated Quadrature Mirror Filters with Arbitrary M, Having the Perfect-Reconstruction Property". IEEE Trans. on ASSP 35(4), April 1987

[2] M. Vitterli, "Filter banks allowing perfect reconstruction", Signal Processing, 10(3):219-244, April 1986

[3] R. E. Crochiere and L. R. Rabiner, "Multirate Digital Signal Processing", Englewood Cliffs, NJ; Prentice-Hall, 1983

[4] Zhang Genzao, "Non-Uniform-Band Digital Filter Banks: Design and Performance Analysis", Ph.D thesis, University of Ottawa, May 1992

[5] M. R. Portnoff " Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis", IEEE trans. on ASSP 28(1):55-69, February 1980.

[6] W. Steenaart and J. H. Lodge, "Tree-Structured Filter Banks Using Signal Modulation and Low-Pass Filters". Proceedings CCECE, pp MA4.10.1-4.10.5, Toronto, Sept 13-16, 1992.

[7] T. Q. Nguyen and P. P. Vaidyanathan, "Two-Channel Perfect-Reconstruction FIR QMF Structure Which Yield Linear-Phase Analysis and Synthesis Filters", IEEE trans. on ASSP 37(5):676-690, May 1989

[8] C. R. Galand and H. J. Nussbaumer, "New Quadrature Mirror Filter Structures", IEEE trans. on ASSP 32(3):522-531, June 1984

[9] J. D. Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks", ICASSP 1980: 291-294

[10] R. V. Cox, "The Design of Uniformly and Nonuniformly Spaced Pseudoquadrature Mirror Filters", IEEE trans. on ASSP 34(5):1090-1096, October 1986

[11] D. J. Goodman and M. J. Carey, "Nine Digital Filters for Decimation and Interpolation", IEEE trans. on ASSP 25(2):121-126, April 1977

[12] C. R. Galand and H. J. Nussbaumer, "Quadrature Mirror Filters with Perfect Reconstruction and Reduced Computational Complexity", ICASSP (2): 525-527, March 1985

[13] P. P. Vaidyanathan, "Multirate Systems and Filter Banks", Prentice-Hall, 1993

[14] H. S. Malvar, "Signal Processing with Lapped Transform", Artech House, 1992

[15] D. B. Chester, "A Generalized Rate Change Filter Architecture", ICASSP vol. 3: 181-184, April 1993

[16] M. Renfors and T. Saramäki, "Recursive N th-Band Digital Filter - Part II: Design of Multistage Decimators and Interpolators", IEEE trans. on Circuits and Systems, CAS 34(1):40-51, January 1987

[17] F. Mintzer, "Filters for Distotion-Free Two-Band Multirate Filter Banks", IEEE trans. on ASSP 33(3):626-630, June 1985

[18] C. S. Burrus and R. A. Gopinath, "Introduction to Wavelets and Wavelets Transforms", a set of notes handed out at the ICASSP, Minneapolis, Minnesota April 1993

[19] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis", IEEE trans. on Information Theory 36(5):961-1005, September 1990

[20] M. Vitterli and C. Herley, "Wavelets and Filter Banks: Theory and Design", IEEE trans. on SP 40(9):2207-2232, September 1992

[21] P. N. Heller and H. L. Resnikoff, "Regular M-Band Wavelets and Applications", ICASSP vol. 3: 229-232, April 1993

[22] J. Kovacevic and M. Vitterli, "Perfect Reconstruction Filter Banks with Rational Sampling Rate", IEEE trans. on SP 41(6):2047-2066, June 1993

[23] S. M. Mitra, M. Abhijit and T. Saramaki, "A Generalized Structural Subband Decomposition of FIR Filters and Its Application in Efficient FIR Filter Design and Implementation", IEEE trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 40(6):363-369, June 1993.

[24] M. R. Petraglia and S. K. Mitra, "Adaptive FIR Filter Structure Based on the Generalized Subband Decomposition of FIR Filters", IEEE trans. on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 40(6):354-358, June 1993.

[25] R. A. Gopinath and C. S.Burrus, "On Upsampling, Downsampling, and Rational Sampling Rate Filter Banks", IEEE trans. on Signal Processing, vol. 42(4):812-824, April 1994

[26] L. G. Weiss, "Wavelets and Wideband Correlation Processing", IEEE Signal Processing magazine, Vol. 11(1):13-32, Jan. 1994

[27] M. A. Akpa and W. Steenaart, "N Parallel Filter Bank Equivalent To Tree Structure", CSECE conference in Halifax, Vol. 2: 494-496, Sept 1994

[28] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine-Modulated FIR Filter Banks Satisfying Perfect Reconstruction", IEEE trans. on Signal Processing, Vol. 40(4):770-783, April 1992

[29] S. Chan and C. Kok, "Perfect Reconstruction Modulated Filter Banks Without Cosine Constraints", ICASSP, vol. 3: 189-192, April 1993

[29] R. A. Gopinath and C. S. Burrus, "On Upsampling, Downsampling, and Rational Sampling Rate Filter Banks", IEEE trans. on Signal Processing, Vol. 42(4):812-824, April 1994