INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Université d'Ottawa · University of Ottawa

Implementation and Optimization of a Modulated Filter Bank based on Allpass Filters

By

Min Chen

A thesis submitted to the
School of Graduate Studies and Research
in Partial fulfillment of the requirement for the degree of

Master of Science
in System Science

April 2001

Abstract

A filter bank based on an allpass IIR filter with brick-wall response was designed by A. J. Van Leest in [17]; however, the delay in the filter bank is too long to be used in real time applications. In order to reduce the delay, the orders of coefficients, transition bandwidth and filter bank structures must be optimized. The order of coefficients can be reduced by increasing the stopband attenuation. In order to further reduce the delay, the sharpness of the filter bank has to be reduced. This thesis also discussed the number of band and filter bank structure against to filter bank delay. The filter bank can be used in non-real time application such as CD compression with high order coefficient. The minimum transition bandwidth can be reached at 0.0325π/number of band.

This thesis expands upon DCT modulations of IIR based modulated filter banks and investigate the Hartley transformation in filter bank modulation as a new modulation technique. These modulation techniques generate the real output signal with real input signals. The quantization errors from quantizing the coefficient are studied. It is concluded that at least 16 bits are required in order for a filter bank to give a good performance as designed without quantization.

Acknowledgments

The author wishes to thank A. J. van Leest from the Netherlands for sending his thesis to me. Special thanks to Dr. Tet Yeap for supervising this thesis and for providing financial support during this period. The author also wishes to thank Dr. Eric Verreault from Newbridge for his useful discussion on this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Thesis Motivation

In the last ten years, we have witnessed explosive growth in the field of communications; especially in digital communications. Currently, the most widely transmitted signal through wired, wireless cellular and satellite media is the digital signal. These signals tend to be more efficient in processing, storing and transmitting data. However, they also require high data bandwidth and storage if they are to remain uncompressed.

Speech audio compression is a technology that converts speech into an efficient digitally encoded format that requires smaller storage and bandwidth. The encoded speech signal can later be decoded to produce a close audio approximation of the original speaker. Therefore, data compression is particularly useful because it enables devices to transmit the same amount of data using fewer bits.

Various speech compression techniques offer different levels of complexity, compressed speech quality and compression efficiency [1]. One technique is subband coding. In a subband coder, the signal is passed through the analysis banks of a filter bank and converted into subband signals. Each subband is then lowpass translated and the ensuing sampling rates are reduced to Nyguist rate for each band. The subbands are then coded using one of the time domain techniques. The number of bits assigned to each band can vary according to the bands perceptual importance. At the receiver, the sampling rates are increased and the bands are modulated back to their original positions by a synthesis bank. The ISO/MPEG is an example of using subband coding [1] to compress speech.

Initially, engineer in Newbridge tried to use subband coding in telephone switch, however, the quality of voice was not good due to much band overlap in the filter bank. Therefore, a near "brick wall like" filter bank need to be designed. In the past decade, a number of different digital filters and special filter structures have been developed. These designs include both finite impulse response (FIR) filters [2] [3] [4] and infinite impulse response (IIR) filters [5]-[17]. Usually, with the same order of coefficients, the IIR filter will produce a much sharper filter. During literature research, it was found that A.J. Van Leest [17] designed a modulated filter bank based on an IIR

allpass filter gave near "brick wall like" filter. Both the analysis and the synthesis filters in his design are stable and causal, but the filter bank does not have the desired perfect reconstruction property. This filter bank produces very small aliasing, phase and amplitude distortion errors. The 2M analysis filters in Leest's[17] design were derived from a prototype filter by exponential modulation. However, the Leest's paper gives only one example of a filter bank design and without any detailed discussion or implementation. During implementation of the filter bank based on Leest's design, it has been noted that the delay of this filter bank, which is over 1000 filter delays, is too long. The filter delay from this filter has to be reduced for real time applications. Another problem with Leest design is that the filter bank uses DFT modulation, which produces a complex output signal after analysis bank. This means that during transmission, the data doubles because both the real and imaginary part of the numbers have to be transmitted.

1.2 Thesis Objective

As discussed above, the filter bank designed by A. J. Van Leest produced a "brick wall like" filter bank. However, this filter bank has over 1000 sample delays. This kind of delay is too long to use telephone switch. Therefore, the purpose of this thesis is to optimize the filter bank based on A. J. Van Leest's design by reducing the delay. Minimizing this delay will allow us to use the filter bank in real time applications.

In order to perform optimization, a modulated filter bank based on A. J. Van Leest's model has been designed. A filter bank has been implemented in Matlab code, and simulation has been performed with the sound signal recorded from a telephone communication using this filter bank. Several conditions have been studied which include the order of coefficients for allpass filter, transition bandwidths, filter bank structures and band numbers. A tree structured filter bank has less sample delays than one of the 2M-modulated filter banks. However, in order to implement a tree structured filter bank, it is necessary to have a real signal after the signal passes through the analysis bank. Since the DFT modulation will produce a complex output signal with a real input signal, another objective of this thesis is to investigate the use of other modulation technique such as DCT and Hartley transformation to produce real output signals.

1.3 Thesis Organization

In Chapter 2, this thesis will discuss the background of the filter and filter bank. In Chapter 3, it will design and implement the 2M channel modulated filter bank based on allpass filters. This chapter will reinvestigate A. J. Van Leest's design and expand his work into the area of implementation and simulation. In Chapter 4, it will focus on the optimization the filter bank. Four possible influence parameters have been discussed. In Chapter 5, it will discuss other design considerations, which include the modulation technique and the quantization error from quantizing the filter bank coefficients. The last Chapter, Chapter 6, is conclusion and future work

1.4 Main Contribution

i. This thesis expanded on Van Leest's design into the implementation and simulation with several audio signals. This work explored the possibility of using an IIR based modulated filter bank with 'brick like' property in the telecommunication field.

ii. This thesis optimized the four possible parameters in 2M channel modulated filter bank in order to reduce sample delay and use in the telecommunication field.

iii. This thesis investigated other modulation techniques used in filter bank, that included Hartley and DCT transform. No literature has been found for any Hartley transform used in filter banks yet.

Chapter 2: Backgrounds

2.1 Digital Signals and Digital Filters

A continuous analog signal can be converted into a sequence of digital numbers, which is a digital signal, by using an A/D converter. According to Nyquist Sampling Theorem [18, pp. 39], the sampling rate has to be at least two times the analog signal frequency in order to ensure proper reconstruction

An ideal filter is able to select the desired band of frequency and reject others. For the passband, the frequency response H(ω) is 1, and for the stopband the response H(ω) is 0. Any realizable non-ideal filter has a transition band in between where its frequency H(ω) changes from passband to stopband. A filter is widely used in systems or networks that selectively changes the wave shape, amplitude-frequency and/or phase-frequency characteristics of a signal in a desired manner. Common filtering objectives are to improve the quality of a signal by extracting information from a signal that makes an available communication channel more efficient by separating two or more signals, which are previously combined signals.

A digital filter is a mathematical algorithm implemented in hardware and/or software that operates on a digital input signal in order to produce a digital output signal and thereby achieving a filtering objective.

The frequency response of digital filter can be expressed as

$$H(e^{j\omega}) = \left|H(e^{j\omega})\right| e^{j\phi(\omega)} \qquad (2.1)$$

The real-valued quantities $\left|H(e^{j\omega})\right|$ and $\phi(\omega)$ are called the magnitude responses and the phase response of filter, respectively. The frequency response can be obtained from the z-transform by setting $z = e^{j\omega}$. The plot of $20\log_{10}\left|H\left(e^{j\omega}\right)\right|$ as a function of ω is particularly useful in revealing the stopband details of the response. This response is called as magnitude response in this study.

2.1.1 IIR/FIR filter

Digital filters are broadly divided into two classes, namely infinite impulse response (IIR) and finite impulse response (FIR) filters. Both types of filters can be represented by the weighted current and previous samples of the signal. For the IIR

$$\textbf{IIR:} \quad y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{2.2}$$

and for the FIR

$$\text{FIR:} \quad y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{2.3}$$

For IIR filters, the impulse response is an infinite duration, whereas for FIR, it is a finite duration. In practice, it is not feasible to compute the output of the IIR filter using Equation 2.2 because the length of its impulse response is too long. The IIR filtering equation is expressed in a recursive form as Equation 2.4

$$\textbf{IIR:} \quad y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{N} a_k x(n-k) - \sum_{K=1}^{M} b_k y(n-k) \tag{2.4}$$

where the a_k and b_k are the filter coefficients. We note that the current output sample y(n) is a function of past outputs, as well as present and past input samples, and thus the filter is a feedback system. FIR filter is a function only of past and present values of input; therefore, this filter is a feed forward system.

Alternative representations for FIR and IIR filters are the z transform. The z-domain representations of FIR and IIR filters are shown in Equation 2.5 and 2.6, respectively.

$$\textbf{FIR:} \quad H(z) = \sum_{k=0}^{N-1} h(k)z^{-k} \tag{2.5}$$

$$\textbf{IIR:} \quad H(z) = \sum_{k=0}^{N} a_k z^{-k} / (1 + \sum_{k=1}^{M} b_k z^{-k}) \tag{2.6}$$

The following presents a

comparison between FIR and IIR filters. Their differences are summarized bellow [19][20].

1) Phase response: FIR filters can have an exact linear phase response. The implication of this is that no phase distortion is introduced into the signal by the filter. The phase responses of IIR filters are nonlinear, especially at the band edges.

2) Stability: FIR filters are realized non-recursively and thus are always stable. Because IIR filters are recursive systems, their stability cannot be guaranteed.
3) Error from quantization effect: The effects of using a limited number of bits to implement filters such as round off noise and coefficient quantization errors are much less severe in FIR than in IIR filters.
4) Number of filter coefficients: FIR filters require more coefficients for sharp cutoff than IIR filters. Thus for a given amplitude response specification, more process time and storage is required for FIR filter implementation.
5) Analogue filters can be readily transformed into equivalent IIR digital filters meeting similar specifications. This is not possible with FIR filters, as they have no analogue counterpart.

Therefore, IIR filters are used only when sharp cutoff and high throughput are desired. FIR filters are used when the number of filter coefficients is not too large and particularly when little or no phase distortion is desired.

2.1.2 Allpass IIR Filters

An allpass filter has the unique property to pass all frequencies equally [18]. The first order of a basic transfer function of an allpass filter is

$$H(z) = \frac{\alpha^* + z^{-1}}{1 + \alpha z^{-1}} \tag{2.7}$$

which represents the first order allpass filter. In a pole-zero plot, the poles and zeros occupy conjugate reciprocal locations. If α is a pole, then its reciprocal conjugate $1/\alpha^*$ is a zero. Thus when the allpass filter is stable, the pole lies inside the unit circle and the zeros lie outside the unite circle.

An Nth order allpass filter with real coefficients has the following form:

$$H(z) = \frac{\sum_{k=0}^{N} a_{N-k} z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \tag{2.8}$$

It is noted that the coefficients of the above equation in the denominator and numerator are reversed. The phase response of an Nth order allpass filter with real coefficient can be expressed as

$$\varphi(\theta) = -N\theta + 2\arctan\frac{\sum_{k=0}^{N} a_k \sin(k\theta)}{\sum_{k=0}^{N} a_k \cos(k\theta)} \qquad (2.9)$$

2.1.3 Halfband Filters

In some applications, a special type of filter known as a halfband filter is required. The passband of a halfband filter normally extends from zero to a quarter of the sampling frequency, which means that half the available bandwidth is passed and the other half is rejected. The halfband filter can be generated by either using FIR or IIR filters. While using an allpass IIR filter to generate the halfband filter, the transformation function is expressed as [17]

$$H(z) = \frac{1}{2}[A(z^2) + z^{-1}B(z^2)] = \frac{1}{2}B(z^2)[A(z^2)/B(z^2) + z^{-1}] \qquad (2.10)$$

where A(z) and B(z) are stable allpass functions and G(z) = A(z)/B(z) is an unstable allpass function. Assume that the phase responses of A(z) and B(z) are $\theta_N(\omega)$ and $\theta_M(\omega)$, respectively. The phase response $\theta_G(\omega)$ of G(z) is $(\theta_N(\omega) - \theta_M(\omega))$. The frequency response of $H_0(z)$ and $H_1(z)$ is expressed in Equation 2.11, whereas $H_0(z)$ and $H_1(z)$ are analysis filter for lowpass and highpass, respectively.

$$H_0(e^{j\omega}) = \exp\left\{j\frac{\theta_B(2\omega) + \theta_A(2\omega) - \omega}{2}\right\}\cos\frac{\theta_G(2\omega) + \omega}{2}$$

$$H_0(e^{j\omega}) = \exp\left\{j\frac{\theta_B(2\omega) + \theta_A(2\omega) - \omega}{2}\right\}\sin\frac{\theta_G(2\omega) + \omega}{2} \qquad (2.11)$$

For a Nth order stable allpass network, its phase is 0 when $\omega=0$, $-N\pi$ when $\omega=\pi$ and it is required to decrease monotonically with increasing frequency. This filter can be designed to be a halfband filter when the desired phase response $\theta_G(2\omega)$ is

$$\theta_G(2\omega) = \begin{cases} -\omega & \text{for } 0 \le \omega \le \pi/2 \\ -\omega \pm \pi & \text{for } \pi/2 \le \omega \le \pi \end{cases} \qquad (2.12)$$

The halfband filter has been used to build a correction filter by using a M-folder interpolation as Renders and Saramaki have demonstrated [7]. A simplified form is

$$H_I(z) = \frac{1}{2}\left[z^{-2L} + z^{-1}D_L(z^2)\right] \tag{2.13}$$

where L is the order of the allpass filter $D_L(z)$, which maybe different from L in $A_i(z)$.

2.1.4 M-Band Filters

The Mth band filter is a generalization of the halfband filter. It is used as a prototype to design M-band filter banks. The stopband of an Mth band low pass filter is located in π/M. If the lowpass Mth bands filter has been successively shifted in a $2\pi/M$ frequency, then an M-band filter banks can be obtained.

An Mth band filter can be designed by using FIR or IIR filters. In the early 1980's, several authors [24 - 27] who used IIR filters reported that the Mth band filters have the following form:

$$H(z) = \frac{1}{M}\sum_{i=0}^{M-1} z^{-i} A_i(z^M) \tag{2.14}$$

where $A_i(z)$ are stable real allpass filters of order L.

$$A_i(z) = \frac{\sum_{k=0}^{L} a_{L-k} z^{-k}}{\sum_{k=0}^{L} a_k z^{-k}} \tag{2.15}$$

These allpass filters $A_i(z)$ have approximate linear phase with a slope of -L+i/M. If H(z) is an Mth band filter, then $A_0(z) = z^{-L}$.

2.2 Decimation and Interpolation of Signals

2.2.1 Decimation

The decimation process is also known as downsampling. It is used to reduce the sampling rate of a signal. Decimation in the time domain involves breaking down a signal into shorter signals that are easier to handle [21]. The decimation is directly defined as

$$y(n) = x(M\,n) \tag{2.16}$$

which means that the output at time n is equal to the input at time M*n. As a consequence, only the input samples with sample numbers that are equal to a multiple of M are retained. Decimation results in aliasing unless x(n) is bandlimited in a certain way. In most cases, an anti-aliasing filter is

required. This filter ensures that the signal being decimated is bandlimited. Decimation corresponds to compression in the time domain whereas has a stretching effect in the frequency domain. The z transform is expressed by

$$Y_D(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(z^{\frac{1}{M}} W^k) \qquad (2.17)$$

where W=exp(-j2π/M).

2.2.2 Interpolation

When several narrow-band signals are combined to form a wide-band signal, their sampling rate needs to be increased first. The sampling rate also needs to be increased if a narrow-band signal is to be observed with a finer resolution in the time domain. The process to increase the sampling rate is known as interpolation, which is also known as upsampling. Typically, an upsampling device is followed by an anti-imaging filter.

The L fold interpolator is defined as

$$y_I(n) = \begin{cases} x(\frac{n}{L}) & \textit{if } n \textit{ is a multiple of } L \\ 0 & \textit{otherwise} \end{cases} \qquad (2.18)$$

The output y(n) is obtained by inserting L-1 zero valued samples between adjacent samples of x(n). The z transform of the interpolator output y(n) is given by

$$Y_I(z) = X(z^L) \qquad (2.19)$$

This means that Y(z) is an L fold compressed version of X in the frequency domain and an expanded version of X in the time domain.

2.3 Filter Banks

A digital filter bank is a collection of digital filters that consist of lowpass, bandpass and highpass filters, each with a common input for analysis banks and a common output for the synthesis banks. Filter banks decompose signals into a number of subbands based on their frequency in the analysis banks and recombine the subband signal to a single signal in the synthesis banks.

In the analysis bank, a digital signal x(n) is passed through a group of digital filters. The filtered signals (subband signals) are thus approximately bandlimited. They are then decimated by M, so that the number of samples per unit time is the same as that for x(n). The decimated subband

signals are then quantized and transmitted. At the receiver end, the signals are recombined to obtain an approximation of the original signal x(n). This recombination is accomplished through the signal, which then passes through the synthesis filters. Such recombination is subject to several errors. One of these is aliasing, which is created by the decimation of subband signals in the analysis bank. Others are amplitude and phase distortion. One of the major goals in developing filter banks is to eliminate or reduce these errors at finite cost.

If a filter bank achieves perfect reconstruction, there are delays between input and output signal. The output signal can be expressed as a function of input signal as following:

$$\hat{x}(n) = cx(n-k) \quad \text{where } c \neq 0 \tag{2.20}$$

Depending on the application, the spectra of M subbands will overlap either heavily, lightly, or not at all. In addition, the subband width can be uniform or non-uniform, and the stopband attenuation may be large or small.

Filter banks can be implemented by using FIR or IIR digital filters. Design of the filterbank based on FIR filters has been extensively discussed in the literatures [2,3,4,]. The FIR filter has many advantages, one of which is its reliability of exact linear phase characteristics. However, if high stopband attenuation and narrow transition bands are required, high-order filters will be needed. These filter bank will store large data, require heavy computation and follow possibly unacceptable signal delay. A few references [6-17] have described the design of the filter bank based on IIR filters and IIR allpass filters. The design principle and prototype for filter banks based on allpass filters will be discussed in detail in Chapter 3. The structure of filter banks can be classified into several different types but only M-channel and tree structures have been used in this thesis.

2.3.1 Filter Bank Structures

2.3.1.1 Parallel Structured Filter Banks

M-channel filter banks are parallel structured filter banks with which all bands are arranged in parallel. A special case is M of 2, which will give 2-band filter bank (see Figure 2-1). An analysis filter bank with filters $H_0(z)$ and $H_1(z)$ decomposes the input signal X(z) into subband signals $X_0(z)$ and $X_1(z)$ with one high-frequency component and one low-frequency component.

This is followed by a synthesis filter bank with filters $G_0(z)$ and $G_1(z)$, which reconstructs the output signal $\hat{X}(z)$ from the subband signals.

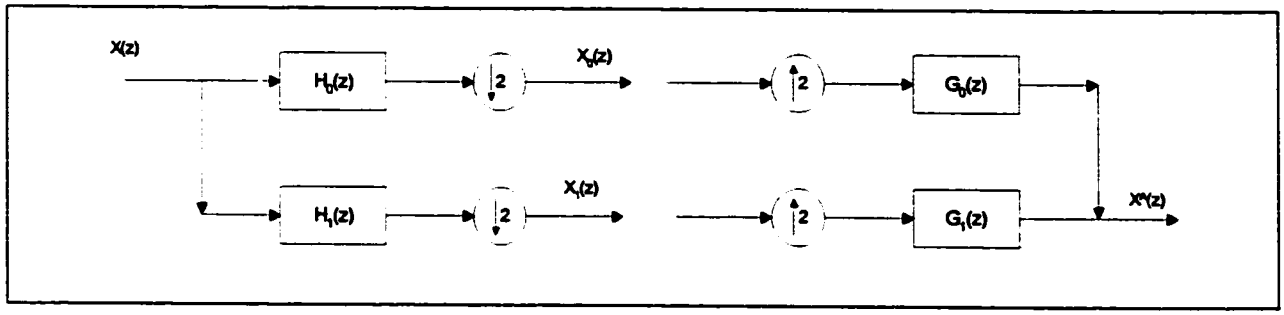

Figure 2-1: Two Channel Analysis bank and synthesis bank

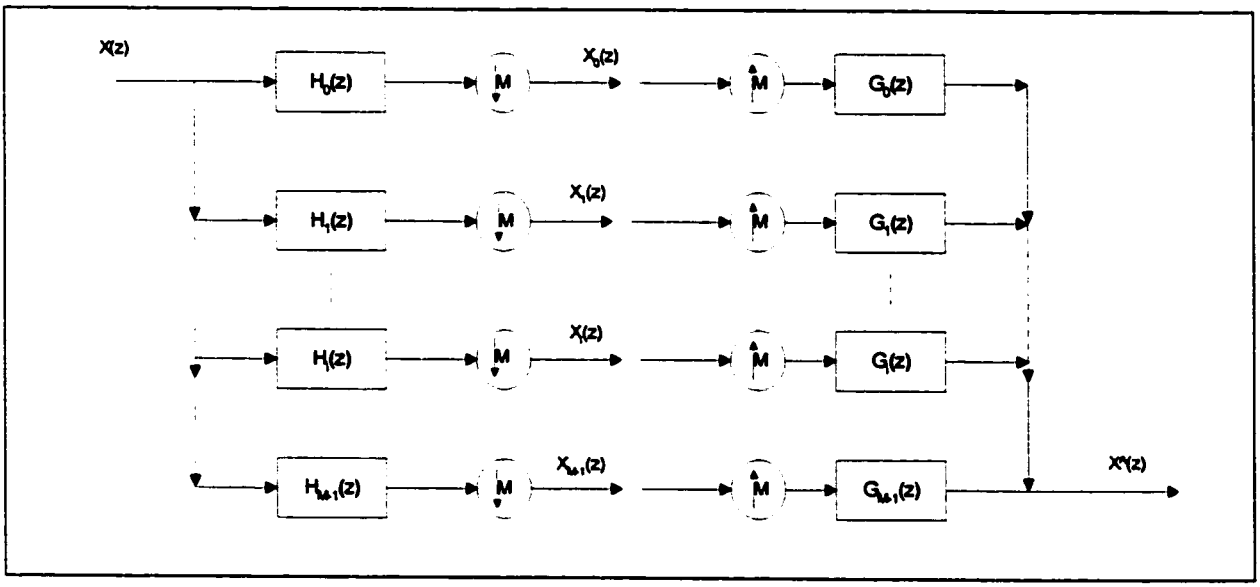

Figure 2-2: M-channel analysis filter bank and synthesis bank

The general structure of a M-channel filter bank is shown in Figure 2-2. This filter bank consists of M individual filters equally spaced in frequency, each having the same bandwidth. The spectrum of the input signal can lie in the frequency range of 0 to π. The filter bank decomposes this spectrum into a set of adjacent subspectra, each having a bandwidth of π/M

It is difficult to find M different transfer functions that provide perfect reconstruction. Instead of this, the M separate functions will be derived from a signal low-pass prototype. The

prototype has been modulated to produce an M-channel filter bank. The prototype can be implemented with either a FIR or an IIR filter.

2.3.1.2 *Tree-structured Filter Banks*

A tree structured filter bank is constructed by cascading smaller systems [18]. The first level consists of a M-band analysis filter bank that splits the input signal into M channel. Then the subband signal, which is further channeled by another analysis bank, splits even further. The synthesis bank has a similar tree in reverse order. The entire bank is a perfect reconstruction if and only if every bank in the tree structure is a perfect reconstruction. Figure 2-3 is an example of a uniform tree structured filter bank. It uses a two-channel system as a subsystem. The signal first splits into 2-channel, and then the subband signals further splits into another 2-channel.

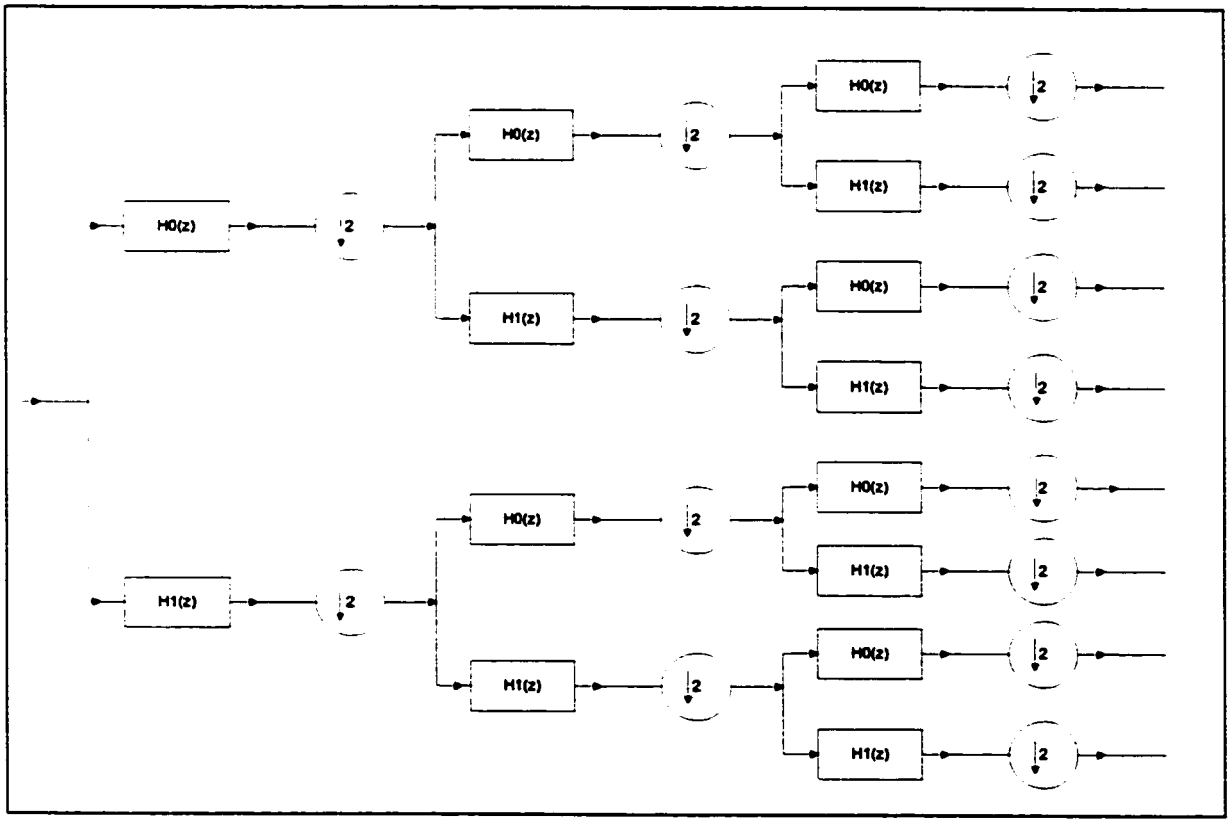

Figure 2-3. A uniform-band tree-structured filter bank (Analysis Bank).

Tree structures are useful in creating a filter bank with non-uniform decimation/interpolation factors. A non-uniform filter bank can be obtained by cascading systems with different decimation factors, such as the structure in Figure 2-4.

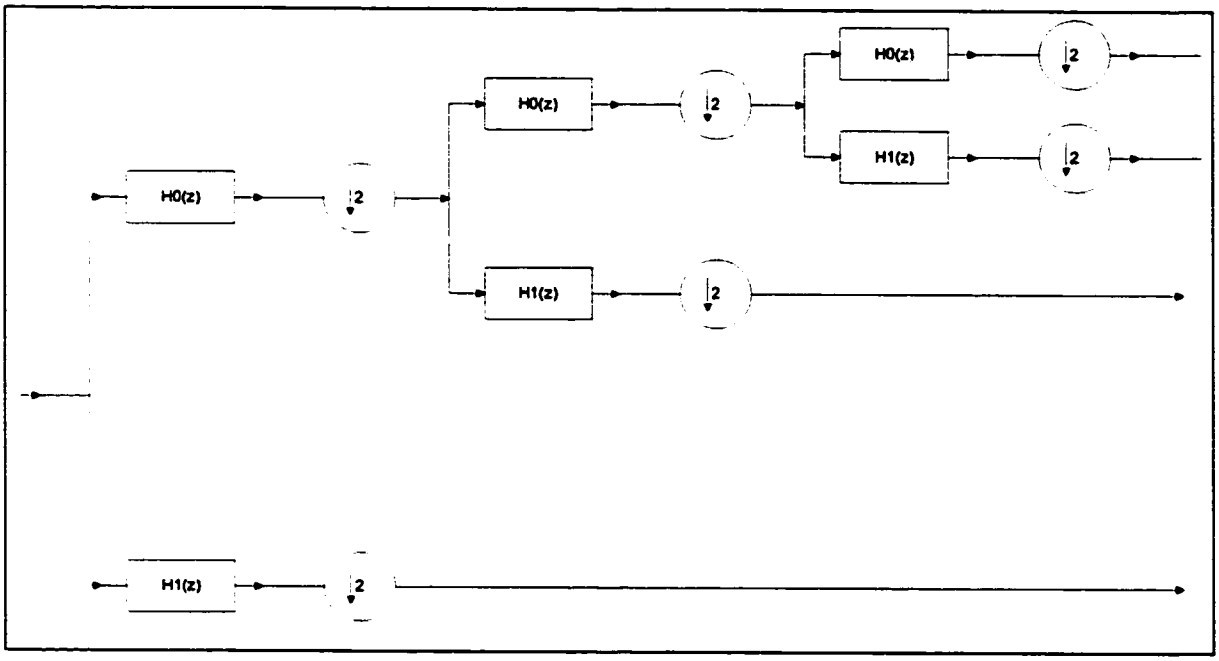

Figure 2-4. The Octave-band tree structure (Analysis bank)

2.3.2 Polyphase Implementation of Filter Bank

Polyphase representations are important advancements in the multirate signal processing. They produce computationally efficient implementations of decimation/interpolation filters, as well as filter banks. In the z-transform, if we are given any integer M, H(z) can always be decomposed as the following formula

$$H(z) = \sum_{n=-\infty}^{\infty} h(nM)z^{-nM} + z^{-1}\sum_{n=-\infty}^{\infty} h(nM+1)z^{-nM} \ldots + z^{-(M-1)}\sum_{n=-\infty}^{\infty} h(nM+M-1)z^{-nM} \qquad (2.21)$$

For example, for a FIR filter $H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$, if M is 2, H(z) can be separated into the even numbered coefficients of h(n) and the odd numbered ones.

The compact form of Equation 2.19 $H(z) = \sum_{l=0}^{M-1} z^{-l} A_l(z^M)$ **(2.22)**

where

$$A_l(z) = \sum_{n=-\infty}^{\infty} a_l(n) z^{-n}$$

with

$$a_l(n) = h(Mn + l), \quad 0 \le l \le M-1$$

This is called the Type 1 polyphase representation and $A_l(z)$ is the polyphase components of H(z). Type 2 polyphase is shown in Equation 2.23

$$H(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} B_l(z^M) \tag{2.23}$$

where $A_l(z)$ are permutations of $B_l(z)$ or $B_l(z) = A_{M-1-l}(z)$.

A polyphase structure is one of the most efficient methods to implement a filter bank. Consider a DFT modulated filter bank as an example, $H_k(z) = H_0(z W^k)$ where $W=e^{-j2\pi/M}$. Assume the prototype $H_0(z)$ has been expressed as Type 1 polyphase and the kth filter can now be expressed as

$$H_k(z) = H_0(zW^k) = \sum_{l=0}^{M-1} (z^{-1}W^{-k})^l A_l(z^M) \tag{2.24}$$

because $(zW^k)^M = z^M$. With $X_k(z)$ denoting the output of $H_k(z)$, we obtain

$$X_k(z) = \sum_{l=0}^{M-1} W^{-kl} (z^{-l} A_l(z^M) X(z)) \tag{2.25}$$

Therefore, the M filters can be implemented by using the structures shown in Figure 2-5. If $A_l(z)$ is rational (i.e., a ratio of polynomials in z or z^{-1}), using noble identities (see Figure 2-6), the filter bank can be expressed as shown in Figure 2-7. When Figure 2-5 and Figure 2-7 are compared, the differences are that one downsamples after the signals through a filter with the power of z^M in Figure 2-5 and another one downsamples before the signal through a filter with the power of z in Figure 2-7 . The filter with the power of z reduced the computation dramatically from the power of z^M.

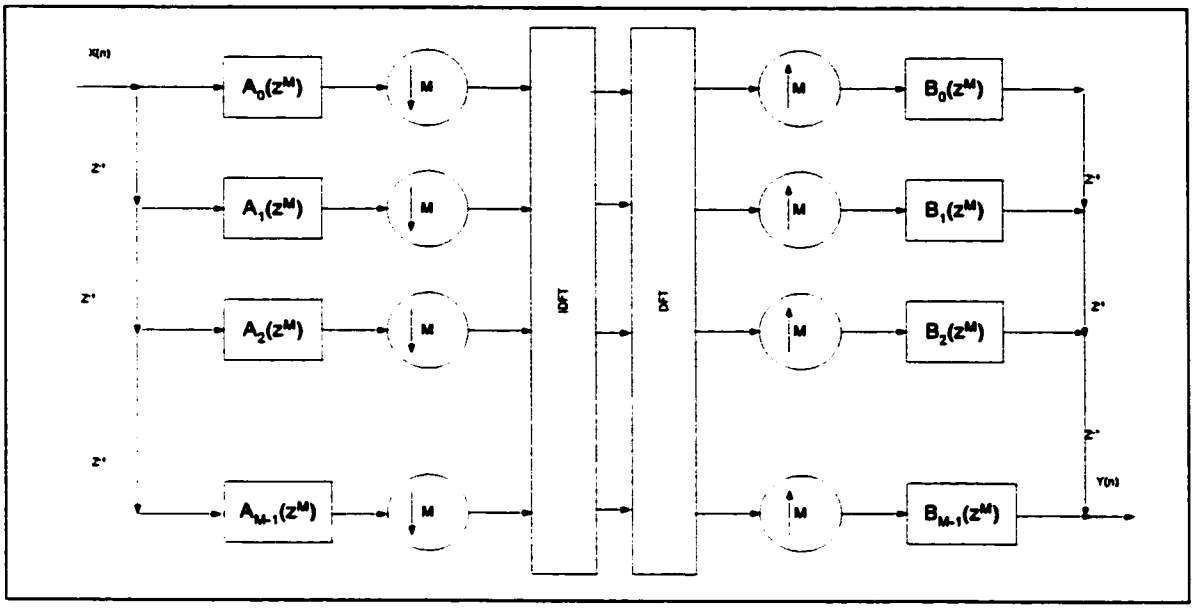

Figure 2-5. An M-channel filter bank.

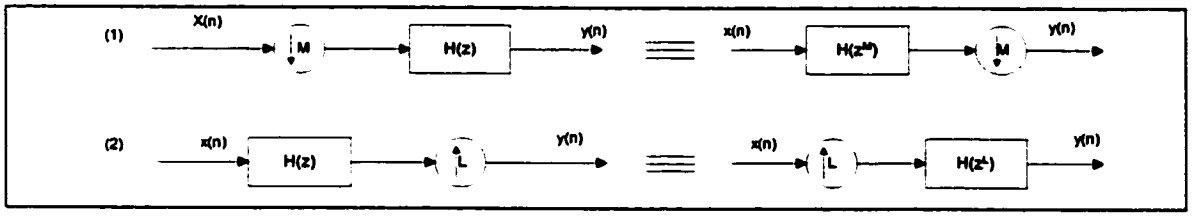

Figure 2-6. The noble identity

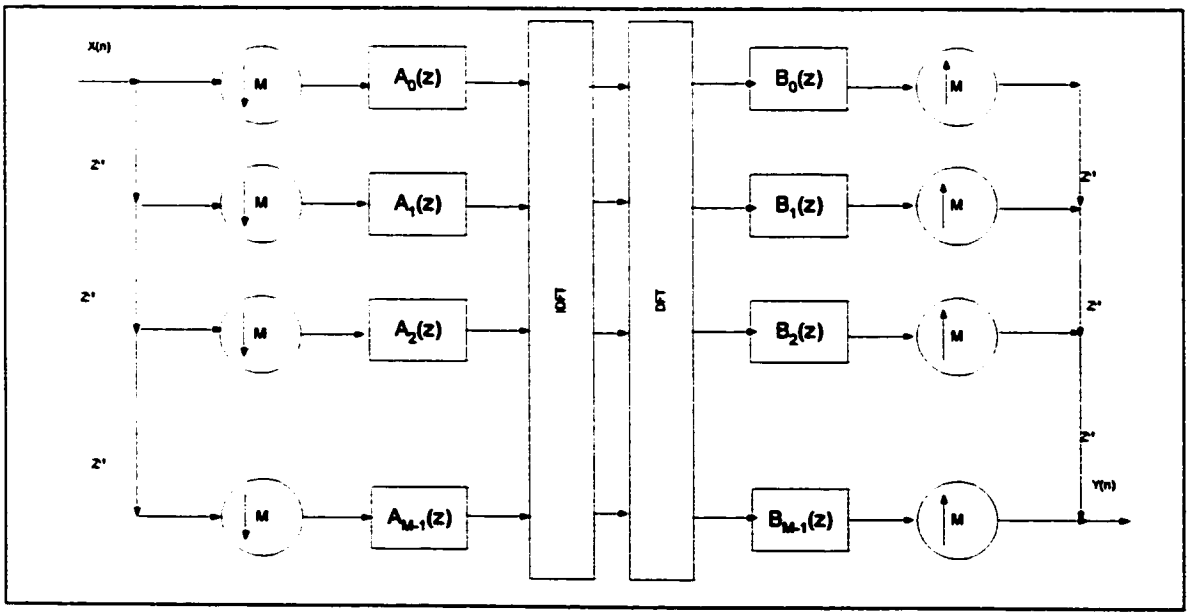

Figure 2-7. An M-channel filter bank with a polyphase structure.

FIR filters H(z) can always be restated in the form as Equation 2.26. For IIR filters, such decomposition is only possible if the transfer function is expressed in a form where the denominator contains only powers of z^M. In order to make any IIR filter implementable using polyphase structures, in 1976, Bellanger et al. [22] attempted to implement IIR filter prototype with a polyphase structure by transforming the original IIR transfer function into the desired form. In 1979, Martinez and Parks [23] described a Remez-type algorithm for a directly optimizing filter with powers of only z^M in the denominator in 1979. These techniques are somewhat limiting, however, because they use subfilters that have the same denominator. In early 1980, several authors [24 - 27] observed that more efficient polyphase filters could be obtained if the subfilters are distinct allpass filters, where H(z) has the form as Equation 2.14.

In 1987, Renfors and Saramaki [7, 8] used a Remez-type design algorithm to design both the nonlinear-phase and approximately linear-phase filters. This method is significantly faster than other exiting methods. Recently, Creusere and S.K. Mitra [13] used the same polyphase structure to design a modulated filter bank and achieve perfect reconstruction. They also used this filter bank in image coding. T. Tuncer and T.Q. Nguyen [6] used the same polyphase structure with IIR filers as previous.

2.3.3 Modulation Technique used in Filter Bank

As we mentioned previously, to design an M-channel filter bank, prototype filters of analysis and synthesis banks need to be designed first and then modulated to the appropriate frequency band. This method produces the M analysis filters and M synthesis filters. The greatest advantage of this method is the simplicity of design and the speed of implementation, when the whole analysis bank is based on one filter H_0. The most popular modulation technique is the Discrete Fourier Transform. A few others are also used in the literature. In this thesis we will investigate three different modulation techniques: DFT, DCT and Hartley transformations.

2.3.3.1 DFT (Discrete Fourier Transform) Modulated Filter Bank

The DFT modulation is one of the more popular modulation techniques to shift a prototype to M different frequency band. For a DFT modulated filter bank, the analysis filters are obtained from modulations of the low pass prototype,

$$H_k(z) = H_0(zW^k) \qquad \text{where } W=e^{-j2\pi/M} \tag{2.26}$$

The synthesis filter is obtained by using the same modulation. The disadvantage of DFT modulation is that the output signals become complex even if the input signal is real.

2.3.3.2 *Cosine Modulated Filter Bank (DCT)*

The analysis and synthesis filters of a Cosine Modulated filter bank are cosine-modulated version of a prototype filter. The advantages of this filter bank are the same as a DFT modulated filter bank: efficient implementation and good frequency responses. The major difference between a DFT modulated and a DCT modulated filter bank is that with DCT modulation, the real input will produce real output. In the literature, the most reported cosine modulated filter banks were FIR based; only two papers were based on IIR allpass filters [15, 16].

There are four types of cosine modulation, which are defined [18] as

$$\begin{cases} TypeI: C_{k,n}^{I} = \sqrt{\frac{2}{M}}\left[c_k c_n \cos(kn\pi / M\right] \\ TypeII: C_{k,n}^{II} = \sqrt{\frac{2}{M}}\left[c_n \cos\left((k+\frac{1}{2})n\pi / M\right)\right] \\ TypeIII: C_{k,n}^{III} = \sqrt{\frac{2}{M}}\left[c_k \cos\left((n+\frac{1}{2})k\pi / M\right)\right] \\ TypeIV: C_{k,n}^{IV} = \sqrt{\frac{2}{M}}\left[\cos\left((k+\frac{1}{2})(n+\frac{1}{2})\pi / M\right)\right] \end{cases} \tag{2.27}$$

where k, n = 0,...,M-1. $c_n = \begin{cases} 1 & if \quad n \neq 0 or M \\ 1/\sqrt{2}, & if \quad n = 0 or M \end{cases}$

The superscript on C indicates the type of modulation. In a cosine-modulated filter bank, only Type II and IV are used in the literature since they first shift the prototype a halfband before shifting it into a regular sequence. It produces M bands with the same bandwidth. The regular DFT modulation will generate M+1 bands, where bandwidth of first and last band is only half of

other bands. The Type I and III modulation will give exactly the same shift as DFT modulation. For example, an 8-channel filter bank will produce 9 bands. Let H(z) and F(z) be the prototype filters of the analysis and synthesis banks, respectively. The cosine-modulated (Type IV) analysis and synthesis filter $H_k(z)$ and $F_k(z)$ are defined as

$$\begin{cases} H_k(z) = a_k b_k U_k(z) + a_k^* b_k^* U_k^*(z) \\ F_k(z) = a_k b_k V_k(z) + a_k^* b_k^* V_k^*(z) \end{cases} \tag{2.28}$$

Where

$$\begin{cases} U_k(z) = H(zW^{k+\frac{1}{2}}), \\ V_k(z) = F(zW^{k+\frac{1}{2}}), \end{cases} \qquad \begin{cases} a_k = e^{j(-1)^k \frac{\pi}{4}} \\ b_k = W^{-\frac{1}{2}(k+\frac{1}{2})} \end{cases}$$

and $w = e^{-j\frac{\pi}{M}}$

Therefore, for FIR based prototype, the analysis and synthesis filters will be

$$\begin{cases} h_k(n) = 2h(n)\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}-\frac{N}{2}\right)+(-1)^k\frac{\pi}{4}\right) \\ f_k(n) = 2f(n)\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}-\frac{N}{2}\right)+(-1)^k\frac{\pi}{4}\right) \end{cases} \tag{2.29}$$

However, for a prototype with an IIR based allpass filter, the equation cannot be simply deviated due to the fact that IIR filter is recursive. For a simple case like 2-band filter, it can be deviated. Nguyen published a paper using a 2-band filter bank only, however, the results were not as good as those produced using the FIR filters [16].

2.3.3.3 *Hartley Transformation*

The Hartley transform belongs to the family of frequency transforms that map temporal or spatial functions into frequency functions. Hartley accomplishes this in a manner similar to the well known Fourier Transform. The significant difference between the Fourier Transform and Hartley's alternative is that the Hartley Transform uses only real values. The kernel function of Hartley transform [18] is

$$\cos\left(\frac{2\pi kn}{N}\right) - \sin\left(\frac{2\pi kn}{N}\right) \tag{2.30}$$

whereas the kernel function of Fourier Transform is $e^{-2\pi ikn/N} = \cos(\frac{2\pi kn}{N}) - i\sin(\frac{2\pi kn}{N})$

Therefore, the Hartley transform produces real output for real input. The discrete version of the Hartley transform can be written explicitly as

$$H(a) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} a_n \left[\cos(\frac{2\pi kn}{N}) - \sin(\frac{2\pi kn}{N}) \right] = \Re F(a) - \Im F(a) \tag{2.31}$$

where H(a) is the transfor of an element from the original signal, a_n, and F(a) is function of Fourier Transform. The inverse transform is accomplished via the same function as the forward transform, which reduces the complexity of the computing. The Fast Hartley Transform (FHT) is rapid when compared with the Fourier alternative. Therefore, some people use FHT to calculate FFT by

$$F = T^{-1}HT$$

where F is the Fourier transform, H is Hartley transform and $T = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}$.

The power spectrum of the Hartley transform can be derived as

$$p(f) = \frac{[H(f)]^2 + [H(-f)]^2}{2} \tag{2.32}$$

Hartley transforms have been used in imaging processing [18]; however, no one to date published any literature for a Hartley modulated filter bank.

2.3.4 Application of Filter Banks - Subband Coding

Subband coding is a powerful and general method that can be used to encode audio signals efficiently. MPEG Audio is a popular example of subband coding. The subband coding depends on a phenomenon of the human hearing system called masking. Normal human ears are sensitive to a wide range of frequencies. However, when signal energy at one frequency band has a lot of power, the ear cannot hear audio signal of lower energy at nearby frequency bands. The louder audio signal of one frequency band masks the softer audio signal at other frequency bands. The louder audio signal is called the masker to mask soft audio signal. The basic purpose of subband coding is to save the signal bandwidth by throwing away information about frequencies that are masked. The result would be the same as the original signal, because human ears cannot hear the difference.

The structure of subband coding is shown in Figure 2-8. First, a time-frequency mapping (a filter bank) decomposes the input signal into subbands. The psychoacoustic model considers these subbands as well as the original signal and determines masking thresholds by using psychoacoustic information. Using these masking thresholds, each of the subband samples is quantized and encoded to keep the quantization noise below the masking threshold. The final step is to assemble all of these quantized samples into frames, so that the decoder can decipher these signals. Decoding is therefore easier since there is no need for a psychoacoustic model. The frames are unpacked, subband samples are decoded, and a frequency-time mapping turns decoded samples back into a single output audio signal. The main advantage of subband coding is that the quantization noise produced in one band is confined to that band. This prevents the quantization noise from masking frequency components in other bands, meaning that separate quantizer step-sizes can be used for each band.

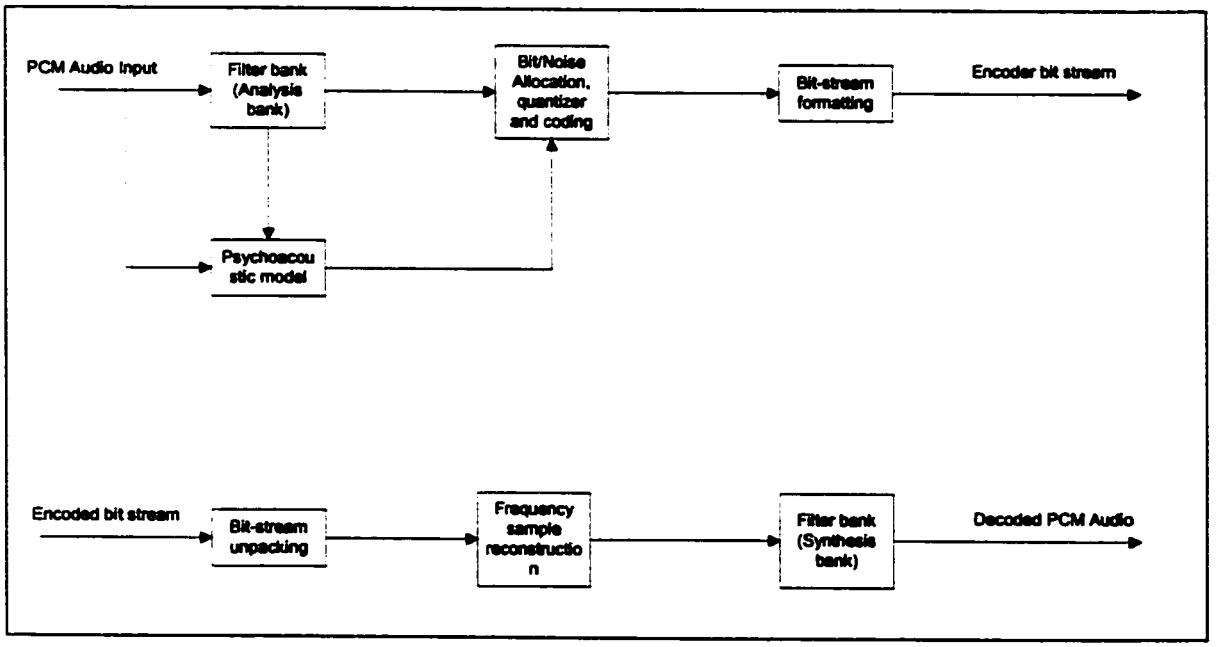

Figure 2-8. MPEG/Audio compression and decompression

In MPEG, a filter bank divides the audio signal into 32 constant width frequency bands. D. Y. Pan [1] pointed out that the design is a compromise with three notable concessions.

(1) The 32 constant width bands do not accurately reflect the ear's critical bands. The bandwidth is too wide for the lower frequencies so the number of quantizer bits cannot be specifically tuned for the noise sensitivity within each critical band.

(2) The filter bank itself would not recover perfectly the original input signal. However, the error introduced by the filter bank is small and inaudible.

(3) Adjacent filter bands have a significant frequency overlap. It will introduce more aliasing to filter bank.

A goal of this thesis is to try to design a filter bank with a "brick wall like" frequency response, both with small errors, which is not audible to the human ear, and with a bandwidth near the ear's critical bandwidth. The delay of this filter bank should small enough to use in telephone communication.

Chapter 3: Design and Implementation of the 2M-Channel Modulated Filter Bank Based on Allpass Filters

Starting from the 1980's, several authors have designed filter banks with the prototype based on IIR allpass filters and a DFT modulation. The advantage of the IIR filter banks is that they give sharp frequency response with lower order of filter coefficient. Rénfors and Saramaki [7, 8] studied the properties of an Nth-band digital filter in detail but did not design the filter bank; Creusers and Mitra[13] did not give any design detail nor any specification of the filter and filter bank; and Nguyen [9] proposed an almost similar prototype with Leest [17], but did not mention any synthesis bank. Only Leest et al. [17] designed a brick wall-like filter bank with both analysis and synthesis filter, but they did not include any filter bank implementation, simulation, or optimization.

In Section 3.1, we will give the background of a 2M-channel modulated filter bank and provide examples based mostly on Leest's works. In section 3.2 and 3.3, we will give implementation and simulation results that continue Leest's work.

3.1 Design of 2M Channel Modulated Filter Bank

3.1.1 The Prototype of Modulated Filter Bank

In the early 1980's, several authors [24 - 27] observed that efficient polyphase filters can be obtained as Equation 2.14. Several authors used the same polyphase structure as previous for the IIR filters. The phase response of $z^{-i}A_i(z)$ should be chosen appropriately so that the H(z) gives good frequency response characteristics. If the phase response is chosen appropriately, H(z) can be a good low pass filter, except for one band of frequency response located in the stopband region [7], [9], [17]. In order to suppress this band, a correction filter has been designed by several authors who cascaded the correction filters with a prototype filter. Nguyen [9,16] reported that with the correction filter, the overall filter may not be an Mth band IIR filter. Further more, the additional complexity of an arbitrary correction filter increases the overall implementation cost. They proposed a solution to the problem by suppressing the stopband peaks by using an interpolation filter. They offered efficient implementation of M-band filters as a cascade of an Mth band filters and an interpolation filter. The allpass filters are designed by using the eigenfilter menthod.[9] The correction filter $H_I(z)$ can be obtained from a halfband filter by an M-fold interpolation independent from H(z), as Renders and

Saramaki [7] have shown. It is also possible to obtain a suitable correction filter by using the allpass filters of H(z) when M is even. The $H_I(z)$ is

$$H_I(z) = \frac{1}{2}\left[A_0\left(z^{2M}\right) + z^{-M} A_\sigma(z^{2M})\right] \tag{3.1}$$

$$\text{where } A_\sigma\left(z^{2M}\right) = \begin{cases} \text{interpolated } \frac{M}{2}\text{th allpass in Mth-band design for even } M \\ \text{Mth allpass in 2Mth-band design for odd } M \end{cases} \tag{3.2}$$

A simplified form is

$$H_I(z) = \frac{1}{2}\left[z^{-2L} + z^{-1} D_L(z^2)\right] \tag{3.3}$$

where L is the order of the allpass filter $D_L(z)$, which can differ from L in $A_i(z)$. $H_I(z)$ can be controlled separately if $A_\sigma(z^{2M})$ is designed independent of H(z), but this increases the number of allpass filters that should be designed in the Mth band filter.

Let G(z) = H(z) $H_I(z)$. If H(z) is an Mth band filter, $H_I(z)$ is an interpolating filter as defined above. G(z) is a 2Mth band filter. Leest and Ritzerfeld [17] used the correction filter as in Equation 3.5 and the design procedures are reported by X. Zhang and H. Iwakura [10, 11, 12]. This prototype has a passband edge frequency of $\pi/2M$ and a bandwidth of π/M.

3.1.2 The Analysis Bank

The prototype described in section 3.1.1 is a 2M-band filter. However, H(z) is expressed in only M polyphase components. Therefore, Leest [17] constructed 2M polyphases using mirrored filters as Equation 3.6 demonstrates.

$$\begin{cases} P_k(z) = \dfrac{H_I(z^M)A_k(z^M) + H_I(-z)A_k(-z^M)}{2} & 0 \le k \le M-1 \\ P_k(z) = z^M \dfrac{H_I(z^M)A_{k-M}(z^M) + H_I(-z)A_{k-M}(-z^M)}{2} & M \le k \le 2M-1 \end{cases} \tag{3.4}$$

In order to modulate the prototype with π/M, z has to be replaced with $z\, e^{-j\pi/M}$. The modulation can use a 2M x 2M DFT matrix to achieve this. The structure of the filter bank is shown in Figure 3-1.

The butterflies in Figure 3-1 can be written as a matrix

$$K = \frac{1}{2}\begin{pmatrix} I & I \\ I & -I \end{pmatrix} \tag{3.5}$$

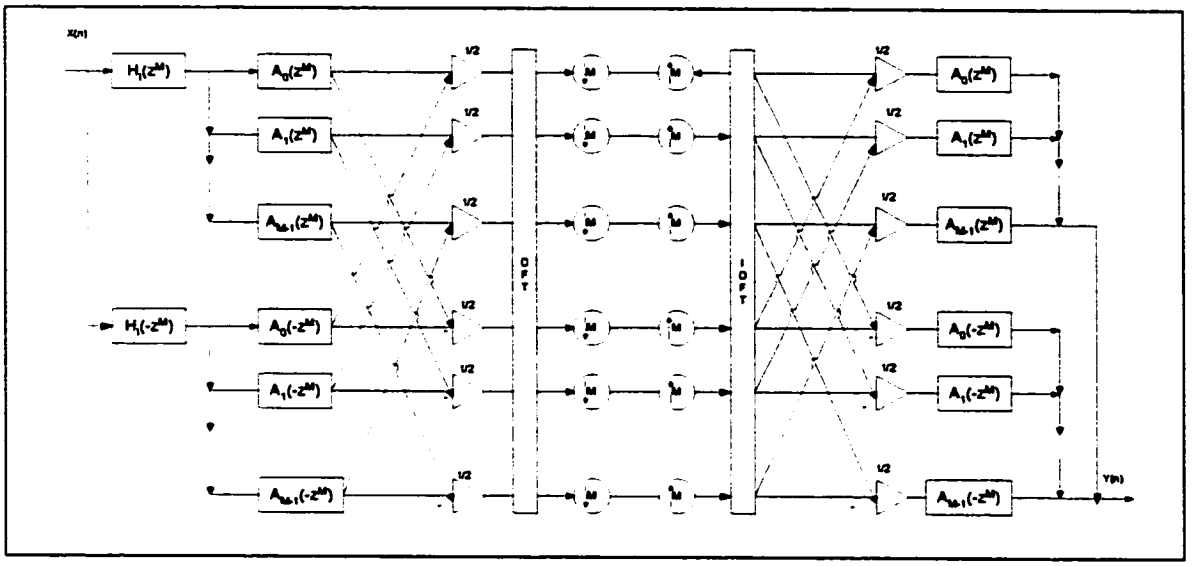

Figure 3-1. A filter bank with 2M-polyphase derivated from M-polyphase structure.

3.1.3 The Synthesis Bank

The analysis bank that Leest designed was described in Figure 3-1, Leest's [17] study shows that, with the analysis bank, the inverse of the polyphase matrix always has at least one entry where the poles lies outside the unit circle for both proposed halfband filters. Thus it is impossible to make the perfect reconstruction with a causal synthesis bank for a maximally decimated case. The prototype that is used has a small transition band. Therefore, it is expected that the inverse of the prototype will have approximately the same characteristics. As a result of this approximation, the filter bank introduces phase, amplitude distortion and aliasing. Leest et al. [17] proved these errors could be very small. They also proved that if downsampling and upsampling with a factor of M instead of a factor of 2M in analysis bank and synthesis bank, the transition band of $H_I(z^M)$ and $H_I(-z^M)$ will disappear. This method means that there is a need for an oversampling factor of 2.

An aliasing and amplitude distortion free filter bank can be obtained, but this requires a higher number of allpass filters in the filter bank quadratic with M. Leest et.al.[17] did some approximation to obtain the approximate aliasing and amplitude distortion free filter bank. However,

the main disadvantage of the filter bank is its non-linear phase response. If the M-fold interpolated halfband filter $H_I(z^M)$ is composed of a delay and an allpass filter as it is in the following,

$$H_I(z^M) + H_I(-z^M) = z^{-2ML} \tag{3.6}$$

then each channel has an approximate linear phase, and the filter bank will have a linear phase response as well. If the upper part of the filter bank were designed as in Equation 3.7 in the passband,

$$A_l(z^M)B_l(z^M) \approx z^{-N} \tag{3.7}$$

then the transition bands of $H_I(z^M)$ will be dependent on the orders of $A_l(z)$ and $B_l(z)$ for a certain integer N. In the lower part of the filter bank, the passband and the transition bands of $H_I(-z^M)$ is desired

$$A_l(-z^M)B_l(-z^M) \approx \pm z^{-N} \tag{3.8}$$

At zero frequency (z =1), the phase in the lower part of the filter bank $A_l(-1)$ $B_l(-1)$ can be 0 or $\pm\pi$. In the case that the phase is $\pm\pi$, output is obtained by subtracting Q from P rather than by adding Q to P. In the stopbands of $H_I(z^M)$ and $H_I(-z^M)$, the amplitude is approximately zero. The phase response in the stopbands can be taken as $-N\theta$. Then

$$\hat{X}(z) \approx z^{-2MN} z^{-N} X(z) \tag{3.9}$$

There are two choices for $B_l(z)$:

i. The phase response of $B_l(z)$ is that the slope of the phase response of $A_l(z)B_l(z)$ in the pass band and the transition band of $H_I(z)$ is approximately $-(N_A + N_B)$, where N_A and N_B are the orders of the allpass filters $A_l(z)$ and $B_l(z)$ respectively.

ii. The phase response of $B_l(z)$ is that the slope of the phase response of $B_l(z)$ in the pass band and the transition band of $H_I(z)$ is approximately $-(N_A + N_B) + 1$.

3.1.4 Design Procedure of the 2M-Channel Modulated Filter Bank

3.1.4.1 The Allpass Filter A(z)

The designing procedure for the allpass filter A(z) was described in Zhang and Iwakura's paper [12]. This design reduces the problem for the phase response of the digital allpass network to the eigenvalue problem using the Remez algorithm. This method also considers the phase error

between the phase response of the allpass network to be realized and the ideal phase response. The eigenvector corresponding to the maximum eigenvalue was calculated, and the filter coefficients were determined. Then the procedure was iterated to derive the optimal approximate solution by the iterative calculation. The phase error between the phase response $\theta(\omega)$ and the ideal phase response $\theta_d(\omega)$ is represented in Equation 3.10 and 3.11:

$$\exp[j\theta_e(\omega)]=\exp[j\theta(\omega)-\theta_d(\omega)]=\frac{\sum_{n=0}^{N}a_n\exp[j(n\omega-\frac{N\omega+\theta_d(\omega)}{2})]}{\sum_{n=0}^{N}a_n\exp[-j(n\omega-\frac{N\omega+\theta_d(\omega)}{2})]} \tag{3.10}$$

$$\theta_e(\omega)=2\tan^{-1}\frac{\sum_{n=0}^{N}a_n\sin(n\omega-\frac{N\omega+\theta_d(\omega)}{2})}{\sum_{n=0}^{N}a_n\cos(n\omega-\frac{N\omega+\theta_d(\omega)}{2})}=2\tan^{-1}\phi(\omega) \tag{3.11}$$

Thus the phase approximation problem for the allpass network is reduced to the minimization of phase error $\theta_e(\omega)$. Considering the equal-ripple phase approximation problem for the allpass network with two approximation bands, the phase error is very small and the phase error can be specified for different bands.

$$|\theta_e(\omega)|=\begin{cases}\theta_e & (\omega_{c1}\le\omega\le\omega_{c2})\\ K\theta_e & (\omega_{c3}\le\omega\le\omega_{c4})\end{cases} \tag{3.12}$$

K can be assigned as a weight function; therefore,

$$W(\omega_i)\phi(\omega_i)=W(\omega_i)\frac{\sum_{n=0}^{N}a_n\sin(n\omega-\frac{N\omega+\theta_d(\omega)}{2})}{\sum_{n=0}^{N}a_n\cos(n\omega-\frac{N\omega+\theta_d(\omega)}{2})} \tag{3.13}$$

If representing the Equation above in the form of a matrix

$$PA=\delta QA \tag{3.14}$$

where $A=[a_0,a_1,...a_N]^T$

$$P=\begin{bmatrix} W(\omega_0)\sin\Theta_0(\omega_0) & W(\omega_0)\sin\Theta_1(\omega_0) & \cdots & W(\omega_0)\sin\Theta_N(\omega_0) \\ W(\omega_1)\sin\Theta_0(\omega_1) & W(\omega_1)\sin\Theta_1(\omega_1) & \cdots & W(\omega_1)\sin\Theta_N(\omega_1) \\ \cdots & \cdots & \ddots & \cdots \\ W(\omega_N)\sin\Theta_0(\omega_N) & W(\omega_N)\sin\Theta_1(\omega_N) & \cdots & W(\omega_N)\sin\Theta_N(\omega_N) \end{bmatrix}, \quad (3.15)$$

$$Q=\begin{bmatrix} \cos\Theta_0(\omega_0) & \cos\Theta_1(\omega_0) & \cdots & \cos\Theta_N(\omega_0) \\ -\cos\Theta_0(\omega_1) & -\cos\Theta_1(\omega_1) & \cdots & -\cos\Theta_N(\omega_1) \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^N\cos\Theta_0(\omega_N) & (-1)^N\cos\Theta_1(\omega_N) & \cdots & (-1)^N\cos\Theta_N(\omega_N) \end{bmatrix} \quad (3.16)$$

and $$\Theta_j(\omega_i)=(j-\frac{N}{2})\omega_i-\frac{\theta_d(\omega_i)}{2} \quad (3.17)$$

When the phase error ratio K between the approximation bands is given, the weight function can be determined and all elements of matrixes P and Q can be known. The equation becomes equivalent to the eigenvalue problem of the matrix; $1/\delta$ is an eigenvalue of matrix, $P^{-1}Q$, and A is an eigenvector.

Design algorithm:

i. The order of the filter N, the ideal phase response $\theta_d(\omega)$ and the weight function $W(\omega)$, are specified.

ii. (N+1) initial values for the sample frequencies ω_I(I = 0, 1,..., N) are set at equal intervals in the approximation bands.

iii. For each sample frequency, matrices P and Q are calculated using Equation (3.15) and (3.16). Calculating the eigenvector for the maximum eigenvalue of $P^{-1}Q$, the filter coefficients a_n are determined.

iv. Using the obtained a_n, the peak points ϖ_I(I=0,1,..., L) of the phase error characteristic $\theta_e(\omega)$ are determined.

v. (L-N) superfluous frequencies ω_i are deleted, and the remaining (N+1) frequencies are stored as the corresponding sample points Θ_i.

vi. When $|\Theta_i-\omega_i| < \varepsilon$ (I = 0, 1, ...N) are satisfied, the procedure ends. If not, the procedure goes to the next step; ε is the specified tolerance.

vii. Letting $\omega_i = \Theta_i$ (I = 0, 1,..., N), the procedure goes back to step (3)

Zhang and Iwakura calculated the maximum eigenvalue and corresponding eigenvector in step 3. In step 4, they used Remez exchange algorithm to reduce the computation complexity.

3.1.4.2 *Rémez Exchange Algorithm*

The Rémez Exchange algorithm [4] was originally used to design equiripple nonrecursive digital filters. It is used to search for the maxima of the error function in order to reduce the amount of computation. The procedure is as follows:

i. Select r+1 external frequencies $\omega_0, \omega_1, \ldots, \omega_r$ such that m+1 of them, namely $\omega_0, \omega_1, \ldots, \omega_m$, are uniformly spaced in the frequency range 0 to ω_p and r-m of them, namely $\omega_{m+1}, \omega_{m+2}, \ldots, \omega_r$, are uniformly spaced in the frequency range ω_a too π. In addition, it is required that $\omega_m = \omega_p, \quad \omega_{m+1} = \omega_a$

Further, the frequency interval $\omega_I - \omega_{I-1}$ is required to be approximately the same in the two bands.

ii. Compute $E(\omega)$ over a dense set of frequencies $\omega = 0, i_p, 2i_p, \cdots, \omega_p, \omega_a, \omega_a + i_a, \omega_a + 2i_a, \ldots, \pi$ where i_p and i_a are the passband and stopband frequency increments, respectively.

iii. Let $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_{kp}, \ldots, \hat{\omega}_\mu, \hat{\omega}_{\mu+1}, \ldots, \hat{\omega}_{ka}, \ldots, \hat{\omega}_\xi$ be potential extremal frequencies for the next iteration, and assume that $\hat{\omega}_0$ to $\hat{\omega}_\mu$ and $\hat{\omega}_{\mu+1}$ to $\hat{\omega}_\xi$ are located in the passband and stopband, respectively. Determine $\hat{\omega}_0$ to $\hat{\omega}_\xi$ by the following procedure:

(a) If $[\{E(0)>0$ and $E(0) > E(i_p)\}$ or $\{E(0) < 0$ and $E(0) < E(i_p)\}]$ and $|E(0)| \geq |\delta|$, set $0 \rightarrow \hat{\omega}_0$

(b) If $\{E(\omega) > 0$ and $E(\omega - i_p) < E(\omega) > E(\omega + i_p)\}$ or $\{E(\omega) < 0$ and $E(\omega - i_p) > E(\omega) < E(\omega + i_p)\}$, set $\omega \rightarrow \hat{\omega}_{kp}$

(c) If $\{E(\omega_p) > 0$ and $E(\omega_p) > E(\omega_p - i_p)\}$ or $\{E(\omega_p) < 0$ and $E(\omega_p) < E(\omega_p - i_p)\}$ and $|E(\omega_p)| \geq |\delta|$, set $\omega_p \rightarrow \hat{\omega}_\mu$

(d) If $\{E(\omega_a) > 0$ and $E(\omega_a) > E(\omega_a + i_a)\}$ or $\{E(\omega_a) < 0$ and $E(\omega_a) < E(\omega_a + i_a)\}$ and $|E(\omega_a)| \geq |\delta|$, set $\omega_a \rightarrow \hat{\omega}_{\mu+1}$

(e) If $\{E(\omega) > 0$ and $E(\omega - i_a) < E(\omega) > E(\omega + i_a)\}$ or $\{E(\omega) < 0$ and $E(\omega - i_a) > E(\omega) < E(\omega + i_a)\}$, set $\omega \rightarrow \hat{\omega}_{ka}$

(f) If $\{E(\pi) > 0$ and $E(\pi) > E(\pi - i_a)\}$ or $\{E(\pi) < 0$ and $E(\pi) < E(\pi - i_a)\}$ and $|E(\pi)| \geq |\delta|$, set $\pi \rightarrow \hat{\omega}_\xi$.

iv. Compute

$$Q = \frac{\max|E(\varpi_k)| - \min|E(\varpi_k)|}{\max|E(\varpi_k)|} \qquad \text{Where } k = 0, 1, \ldots \xi. \qquad (3.18)$$

v. Reject the $\xi - r$ frequencies $\hat{\omega}_k$, for which $|E(\hat{\omega}_k)|$ is lowest, and renumber the remaining passband frequencies from 0 to m and the remaining stopband frequencies from m+1 to r. Then update extremal frequencies by letting $\omega_k = \hat{\omega}_k$, for $k = 0, 1, \ldots, r$.

vi. If $Q > 0.01$, repeat from ii. Otherwise continue to vii.

vii. Compute $P(\omega)$ using the last set of extremal frequencies.

viii. Deduce the impulse response h(n) of the filter.

3.1.4.3 The Correction Filter

The choice of the halfband filter $H_I(z)$ is dependent on which filter bank is desired. If almost perfect reconstruction is desired, $H_I(z)$ has to be composed of a delay and a delayed allpass filter.

$$H_I(z) = \frac{1}{2}\left[z^{-2L} + z^{-1}D_L(z^2)\right] \qquad (3.19)$$

where L is the order of the allpass filter $D_L(z)$. Two filters of $H_I(z^M)$ and H(z) cannot be overlapped. In order to obtain two filters that do not overlap, the allpass filters in H(z) must have an approximate linear phase between 0 and $\pi/2$.

3.1.5 An Example of the Almost Perfect Reconstruction Filter Bank

An almost perfect reconstruction filter bank was designed with M of 32. An example of the almost perfect reconstruction filter bank would be an order of 11 for an allpass filter D(z) in $H_I(z)$, and 4 and 7 for an allpass filter in an analysis and a synthesis bank A(z) and B(z), respectively. The coefficients for filters A, B and D are shown in the Appendix Table C-1, Table C-2 and Table C-3. Since the first allpass filter (k = 1) is a pure delay, the coefficients are not listed in the table.

Figure 3-2 shows the amplitude response of the prototype of H(z) and $H_I(z^{32})$. It is noted that the transition bandwidth of the prototype is determined by the transition bandwidth of the correction filter $H_I(z^M)$. Thus the transition bandwidth of H(z) can be very wide. The stopband attenuation of the prototype is determined by the stopband attenuation of both H(z) and $H_I(z^M)$. The transition bandwidth of the halfband filter $H_I(z^2)$ is 0.1π, thus the transition bandwidth of $H_I(z^{32})$ is $0.1\pi/32$. The phase error of the allpass filters A(z) in H(z) are designed equiripple from 0 to 0.5π, resulting in a filter H(z) with a transition bandwidth of $0.5\pi/32$. The stopband attenuation of both H(z) and $H_I(z^{32})$ are proximately 90 dB. The allpass filters B(z) are designed given the phase response of A(z). The slope of the phase response of A(z)B(z) in the pass band and the transition band of $H_I(z)$ is approximately $-N_A-N_B+1$, where N_A and N_B are the orders of A(z) and B(z), respectively. This gives the prototype a minimal stopband attenuation of approximately -90 db and a transition band of $0.1\pi/32$.

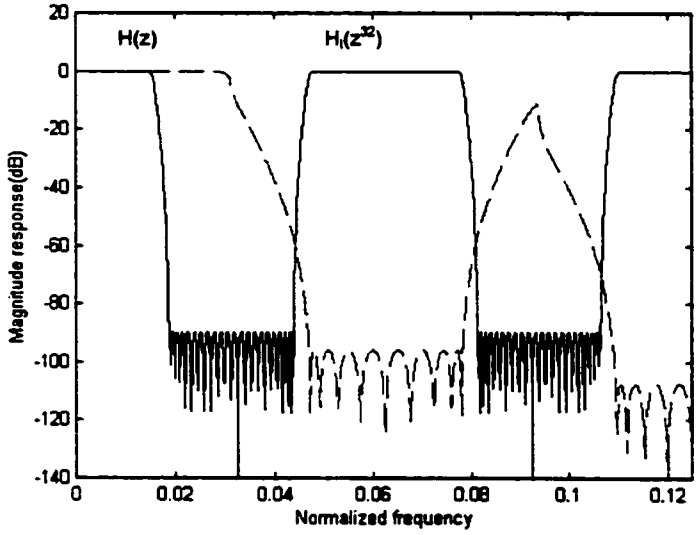

Figure 3-2. The amplitude response of H(z) and $H_I(z^{32})$

The amplitude response of prototype $R(z) = H_I(z^{32})H(z)$ is shown in Figure 3-3. The sharpness of the filter bank can be seen in Figure 3-4 from the amplitude response of the filter bank.

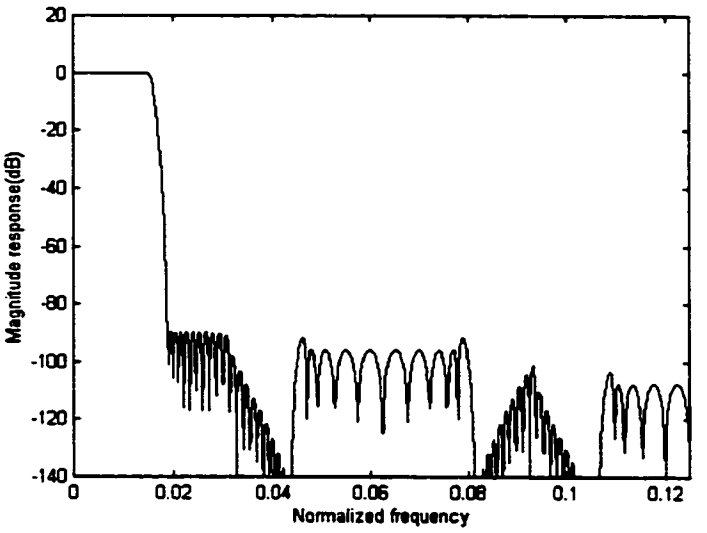

Figure 3-3. The amplitude response of the prototype $R(z) = H_I(z^{32})H(z)$

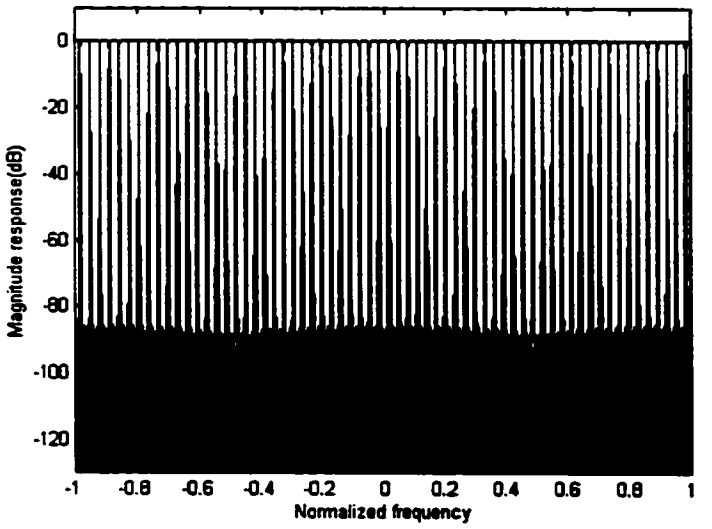

Figure 3-4. The magnitude response of the filter bank with M = 32.

The quality of an almost perfect reconstruction filter bank can be judged by amplitude distortion, phase distortion and peak aliasing distortion. The amplitude and phase distortion of the designed filter bank are shown in Figure 3-5. A maximum amplitude distortion is approximately –93.6 dB and a maximum phase distortion is approximately $\pm 3.8 \times 10^{-5}$ radians. Both type of maximum distortion occur at the transition frequencies. Figure 3-6 shows that the peak aliasing distortion is approximately –93 dB with the maximum values again occurring at the transition band.

Leest [17] used this example to prove that a very good performance of the filter bank can be obtained with minor complication.

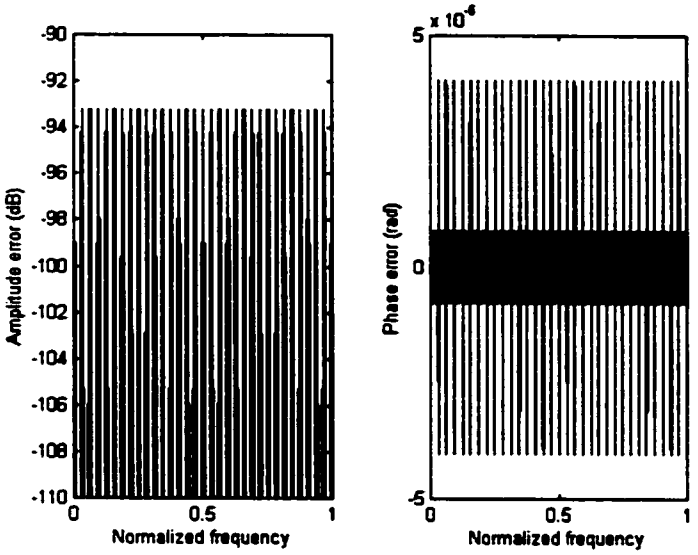

Figure 3-5. The amplitude error and phase error of the designed filter bank.

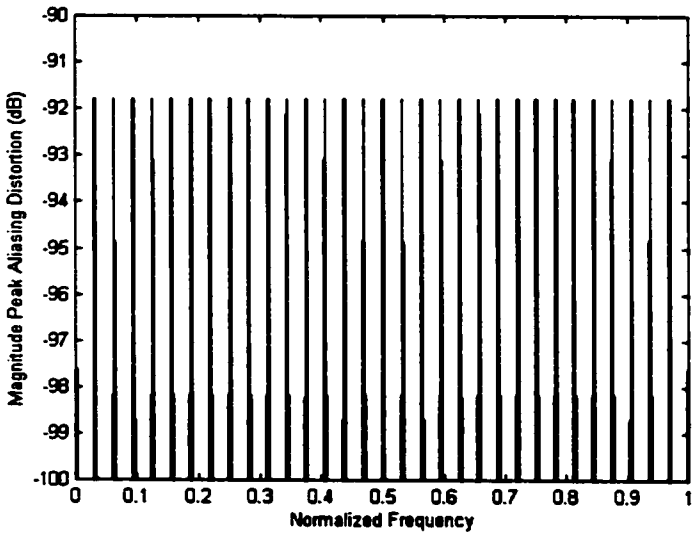

Figure 3-6. The peak aliasing distortion of the designed filter bank.

3.2 Implementation and Simulation

Leest et al.[17] designed an almost perfect reconstruction filter bank that has very good performance. However, they did not publish any simulation results. In this thesis, the almost

perfection reconstruction filter bank was implemented with Matlab (see Appendix) and an audio signal was used in the simulation.

3.2.1 Computer Equipment and Software

A Dell Dimension V400c PC-compatible computer equipped with a Pentium® II processor at 400MHz, 128MB of SDRAM memory and Window NT operating system was used for simulation.

A MatLab student version 5.3 was used to implement the filter bank and perform the simulation.

3.2.2 Implementation Procedure

In order to efficiently compute the almost perfect reconstruction filter bank, the structure from Figure 3-1 has been redrawn in Figure 3-7 based on notable identity. The filters $H_I(z^M)$ and $H_I(z^{-M})$ in the analysis bank can be combined and noted as $H(\pm)$ in Figure 3-7. The filter bank was implemented as follows.

Analysis Bank:

- delay and downsampling for M times
- pass through $H(\pm)$ filter
- pass through filter A(z) or A(-z).
- add or subtract two bands signals to form two new bands signal
- go through DFT modulation

Synthesis bank:

- go through Inverse Fast Fourier Transformation
- add or subtract two bands signals to form two new bands signal
- pass through filter B(z) or B(-z)
- add two bands together
- upsample the sample M time
- delay and add signal together to reconstruct signal

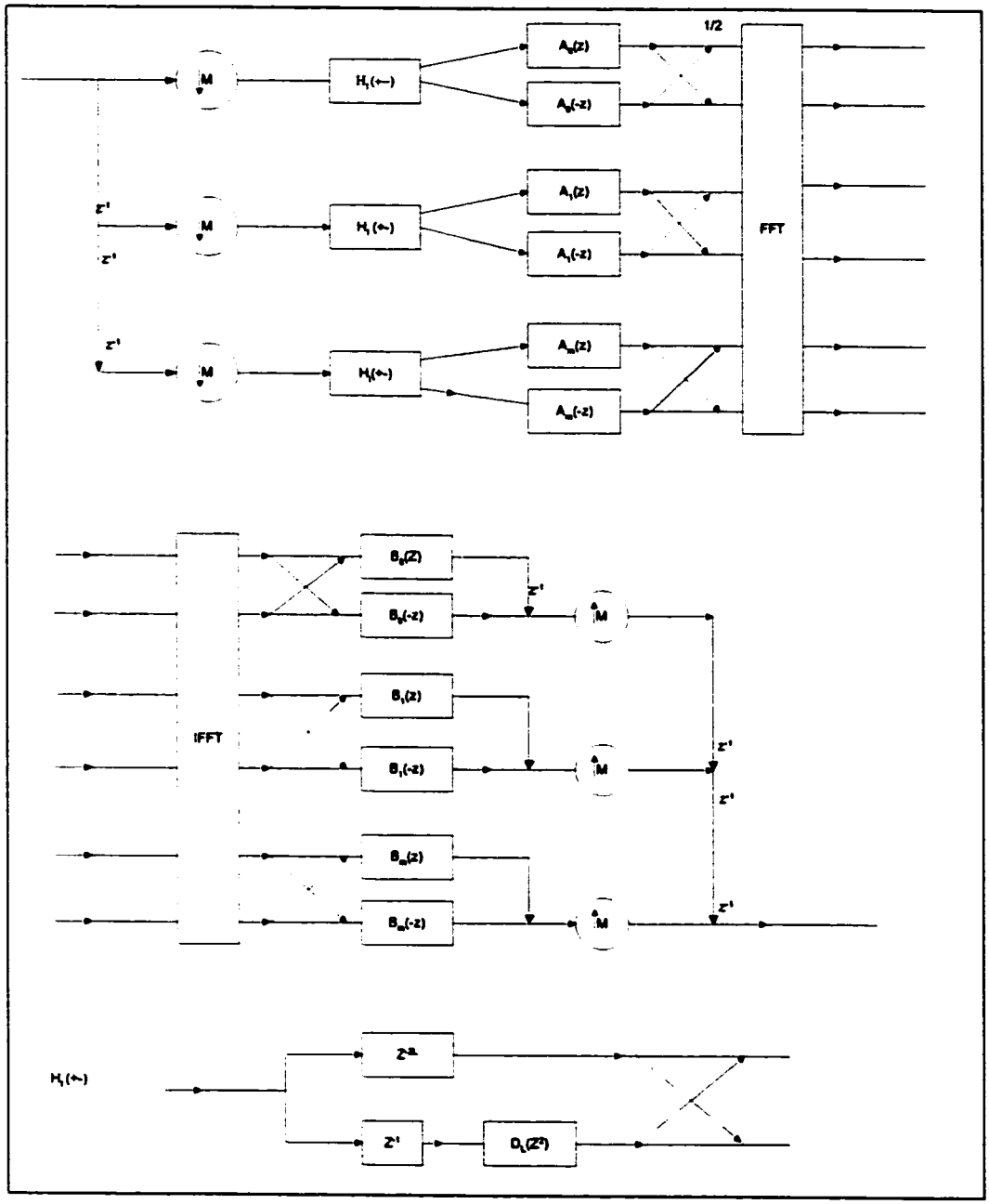

Figure 3-7. Filter bank structure used in simulation.

3.2.3 Simulation Environment

3.2.3.1 Test Samples

Sample 1: Chirp signal

The Chirp signal is a standard sound signal in the Matlab. In the time domain (see Figure 3-8), it gives regular sharp signal. In the frequency domain (see Figure 3-9), the majority of the signals are located in a high frequency range. Simulation with a chirp signal is used for testing the filter bank response to the high frequency signal with sharp property.

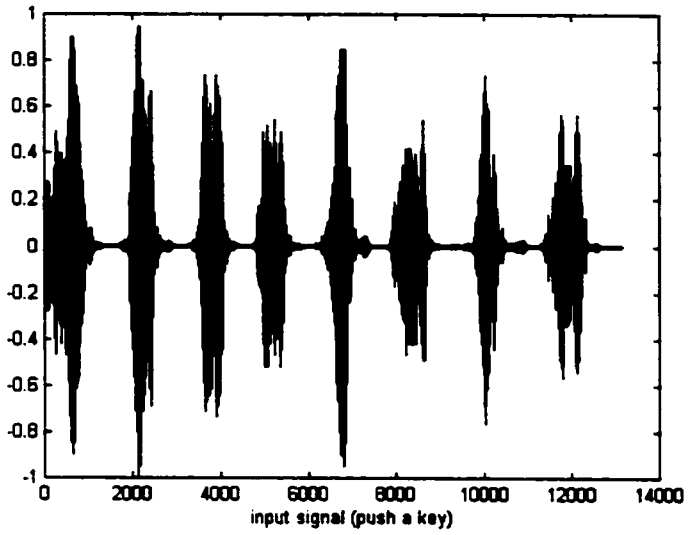

Figure 3-8. A chirp signal in the time domain from Matlab.

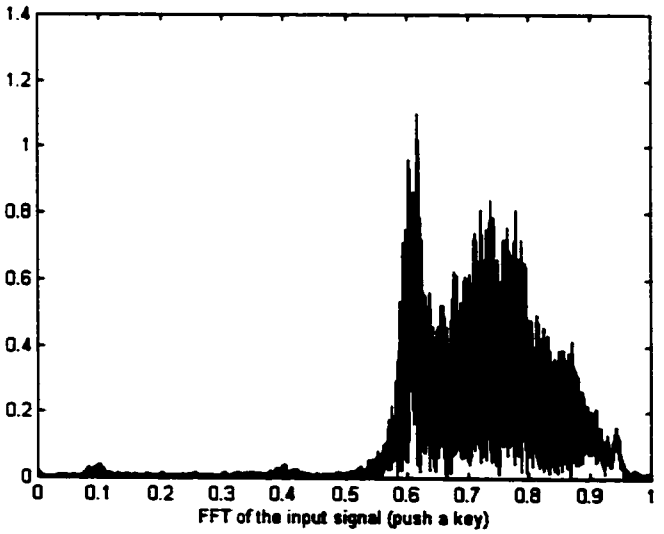

Figure 3-9. The spectrum of chirp signal in frequency domain.

Sample 2: a segment of people speech signals with some musical background

This signal was downloaded from internet in .wav format. A segment speech signal was recorded from a man's speech with some music in the background. The signal had some degrees of white noise. The signal in the time domain is shown in Figure 3-10 and in frequency domain is shown in Figure 3-11. This signal covers a much wider range of frequency than the chirp signal. It can be used to test filter bank frequency response with wide range frequency.

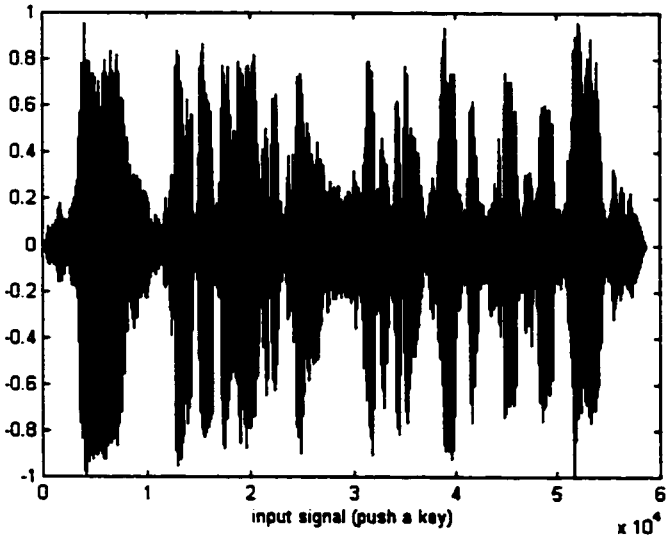

Figure 3-10. A segment of speech signal with some music background in the time domain.

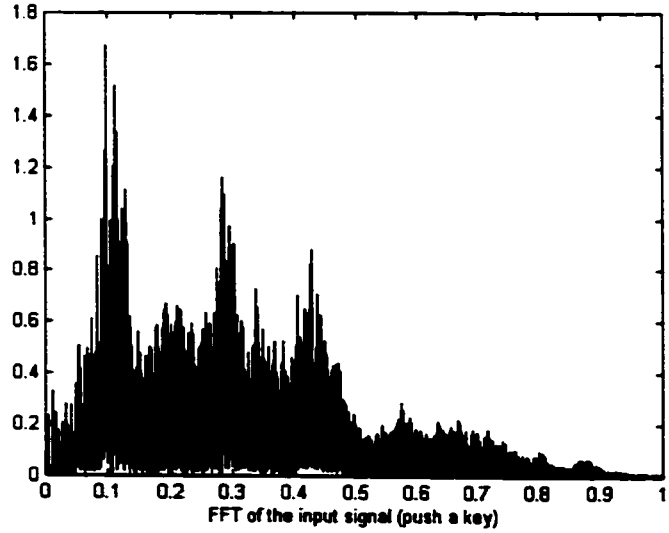

Figure 3-11. A segment of speech signal with some music background in the frequency domain.

Sample 3: a segment of people speech signal recorded from telephone line

This sample was obtained from Newbridge. It was a recorded 5 minutes long telephone conversation. Only a portion of the signal was used in the simulation. The signal frequency bandwidth (see Figure 3-13) is quite similar to that of sample 2. In the time domain (see Figure 3-12), the signal in amplitude response is cleaner when compared to sample 2. Studying this sample will provide some information about the possibility of using this filter bank in telecommunication.

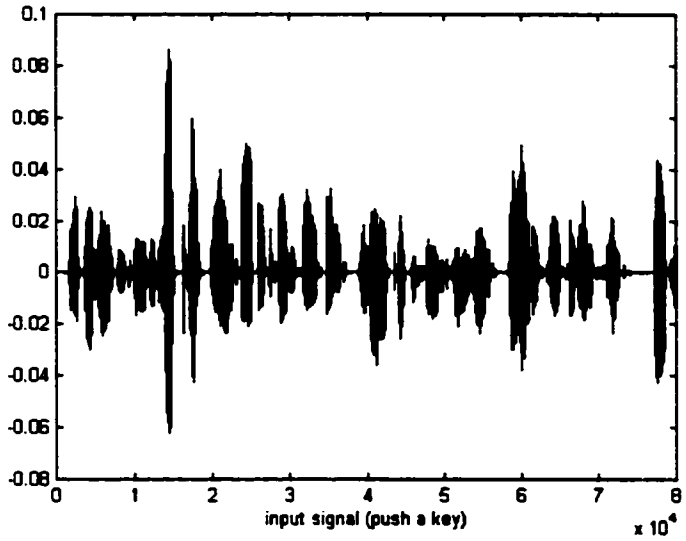

Figure 3-12. A segment of speech signal in time domain.

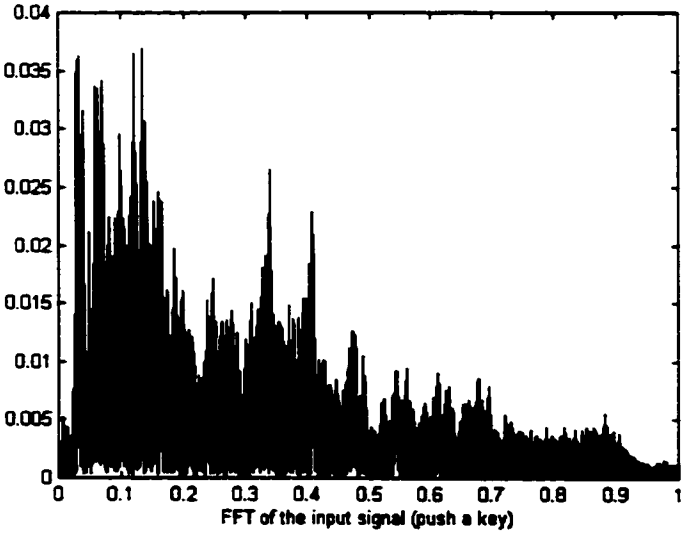

Figure 3-13. A segment of speech signal in frequency domain.

3.2.3.2 *Filter Bank Used in the Simulation*

The filter bank used in this simulation has an almost perfect reconstruction, which was designed in 3.1.5 with M of 32. The order of coefficient for correction filter is 11. The order of coefficient for the analysis and synthesis bank is 4 and 7, respectively. The transition bandwidth is $0.1\pi/32$ and the stopband attenuation is approximately 90 dB. The maximum amplitude and phase distortion of the filter bank is approximately –93.6 dB and $\pm 3.8 \cdot 10^{-5}$ radians, respectively. The peak aliasing distortion is approximately –93 dB.

3.2.3.3 *Errors*

Since the ideal filter bank must have a pure delay, which means that there are only delays between the output signal and the input signal, comparing the output signals with the delayed input signals will yield some information about the quality of the filter bank with respect to a certain signals. Two errors were used in this study.

(1) Square error:

A square error is in the time domain. It is calculated by using a point-to-point base as follows:

$$\sigma_{square} = \left(\hat{x} - x_d \right)^2 \qquad (3.20)$$

where $\hat{x}$ is a reconstructed signal and x_d is a delayed input signal. The plot of σ_{square} as a function of time shows the error in the time domain.

(2) Amplitude error:

An amplitude error is calculated in the frequency domain. It is calculated as follows:

$$\sigma_{amplitude} = abs\left\{ 1 - abs\left(\frac{\hat{F}}{F_d} \right) \right\} \qquad (3.21)$$

where $\hat{F}$ is a reconstructed signal in the frequency domain and F_d is a delayed input signal in the frequency domain.

A plot of $20\log_{10}(\sigma_{amplitude})$ as a function of normalized frequency will demonstrate the error in the frequency domain and give the frequency response of filter bank.

3.2.4 Simulation Results and Discussions

3.2.4.1 Simulation with Sample 1

Simulation was performed with sample 1. The amplitude error and square error for the simulation results are shown in Figure 3-14. In the time domain, the square error is approximately 1.2×10^{-8}. The signals with the large error also had the high amplitude. In the frequency domain, the amplitude errors are located in -40 dB to -80 dB ranges. Most of the signals energy from sample 1 are located at high frequency ranges. The error in the high frequency range is smaller than in the low frequency range. This result demonstrates that the error is small, both in the time domain and in the frequency domain. The filter bank has a good frequency response with signals located mainly at higher frequencies.

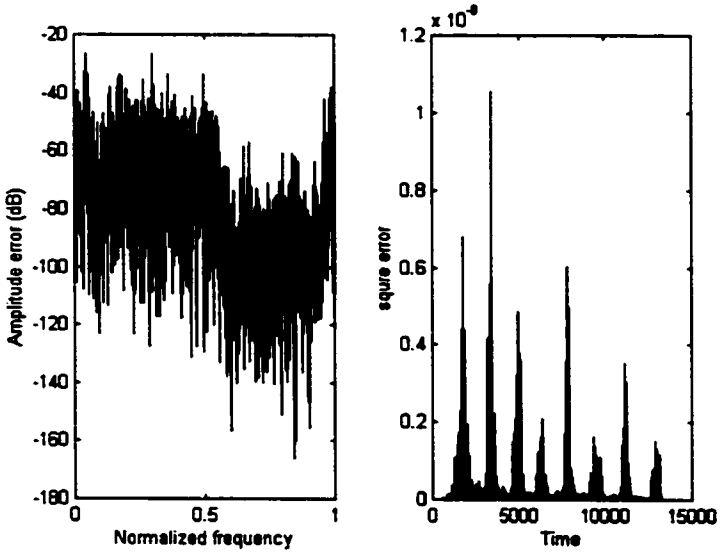

Figure 3-14. Simulation results with sample 1

3.2.3.2 Simulation with Sample 2

Since sample 2 is a real speech signal with some musical background, simulation was performed, and the reconstructed signal was stored on a disk in .wav format and played on the computer. The difference between the input and the reconstructed signal cannot be distinguished by the human ear. The simulation result with this signal is shown in Figure 3-15. The square error in the

time domain is also less than 10^{-8}. The average amplitude error is around –80 dB. This simulation results prove that the filter bank offers a good frequency response with small errors.

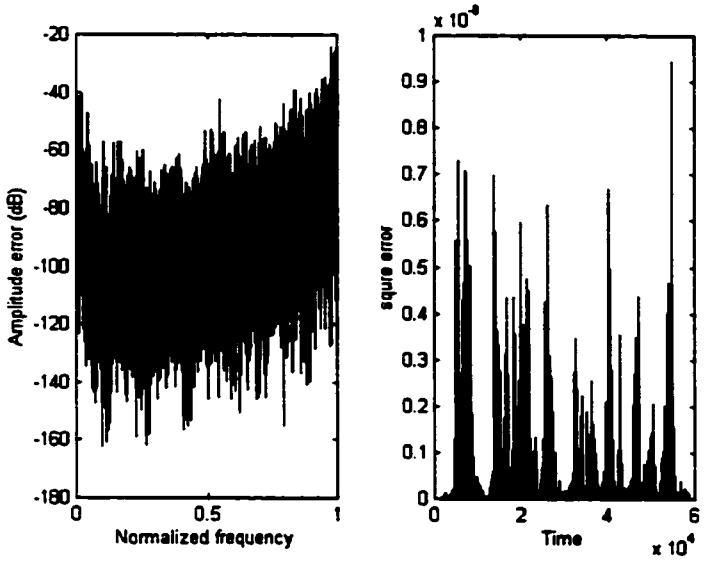

Figure 3-15. Simulation result with sample 2

3.2.3.3 *Simulation with Sample 3*

The simulation results of sample 3 are shown in Figure 3-16. Sample 3 is a segment of speech signal recorded from telephone line. The reconstructed signals were also saved as an .au file. No difference can be distinguished by the human ear for both the reconstructed signal and the original signal. Figure 3-16 shows that the maximum square error in the time domain is less than 2.5 x 10^{-11} which is much less than square error from sample 1 and sample 2. The amplitude error in the frequency domain is around –90 dB. This proves again that even the filter bank did not give perfect reconstruction, but the error is quite small. The filter bank has the potential to be used in the telecommunications applications. However, the problem is that it has delay of over 1000 samples, which means that the delay will be about 132 ms if the filter bank is used in the telephone line. Humans cannot accept this kind of delay during conversation. In the next chapter, we will optimize the parameters that we used in design in order to reduce this delay while maintaining an acceptable error range.

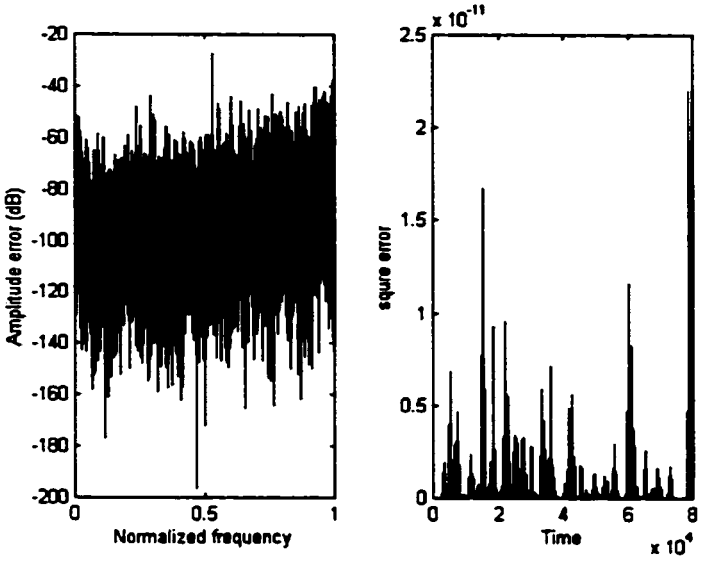

Figure 3-16. Simulation result with sample 3

Chapter 4: Optimization of DFT based Modulated Filter Bank

4.1 The Effect of order of Allpass Filter

The filter bank designed in Chapter 3 has a delay of 1055, which is about 132 ms. The tolerable minimum delay for human ear is around 200 samples [oral communication with Dr. Eric Verreault from Newbridge], which is about 25 ms.

The delay of the filter bank can be calculated by Equation 4.1, based on the structure of the filter bank.

$$\text{Delay} = M(N_A + N_B - 1) + N_H 2M + (M - 1) \quad (4.1)$$

If we rearrange Equation 4.1 to obtain Equation 4.2, delay is proportional to the $(N_A + N_B + 2N_H)$ and to the number of bands (M).

$$\text{Delay} = M(N_A + N_B + 2N_H) - 1 \quad (4.2)$$

where M is the number of band, N_A, N_B and N_H is the order of coefficients for the IIR filters of A,B and H. In order to reduce the delay, the order of coefficients and/or the number of bands must be reduced.

Since the order of coefficients of filters A, B and H cannot be reduced independently, it is necessary to determine the same stopband attenuation for filters A and H. The stopband attenuation for A and B have been studied with respect to the order of coefficients. The results are summarized in Figure 4-1.

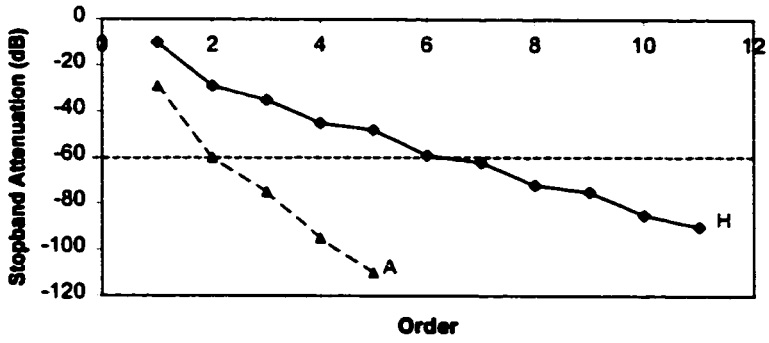

Figure 4-1. The relationship between stopband attenuation and the order of coefficients for allpass filter A and H.

The stopband attenuation is decreased by increasing the order of coefficients for A and H. However, the decrease rate is much faster for A than H. The order of filter A is increased from 1 to 5, so that stopband attenuation drops from –30 dB to –110 dB. However, to produce similar stopband attenuation in filter, the order of coefficients needs to be from 3 to over 12. In order to reach the stopband attenuation of –90 dB, the orders of coefficient need to be 4 for A and 11 for H. For the subband sound compression in telephone systems, the stopband attenuation needs to be at least –60 dB [Oral communication with Newbridge Engineer: Dr. Eric Verreault]. Therefore, the order of coefficients can be reduced to 2 for A and 6 for H. The prototype of the 32 subbands filter bank with an order of coefficients of 2 for A and 6 for H is shown in Figure 4-2. Compared to the original design with the order of coefficients of 4 and 11 in Figure 3-3, the only difference with the prototype is a stopband attenuation from –90 dB to –60dB.

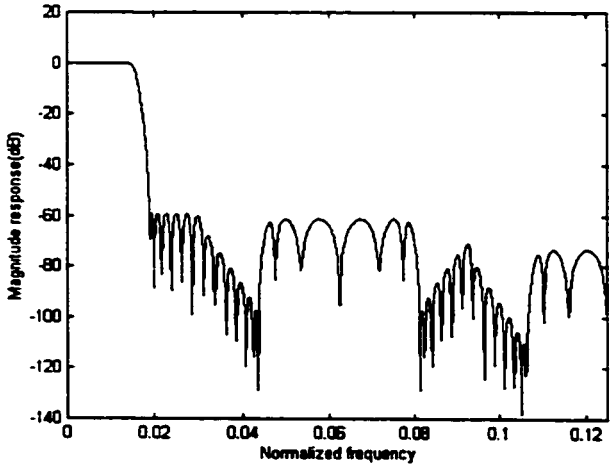

Figure 4-2. The amplitude response of the prototype with an order of coefficients of 2 for A and 6 for H with 32 subbands.

Matching the stopband attenuation of A and H is one way to reduce the coefficient of A and H and still keep good filter bank quality. The only way to find an optimized number of orders for filter B given A and H is to find reasonable phase, peak aliasing and amplitude errors. The phase and peak aliasing errors can be calculated, but the amplitude errors can only be produced in simulation. Since the order of coefficients for filters A and H is 2 and 6, respectively, the phase and peak aliasing errors have been calculated with respect to the order of B. The phase error and peak aliasing response to the order of coefficients for B is shown in Figure 4-3 and in Figure 4-4, respectively. The phase error and aliasing distortion error show the same trend with respect to the order of coefficients for B. They have no significant difference between the order of coefficients of 4 and 5 for filter B. However, from 3 to 4, there is not a big difference in phase error, but quite a substantial one in aliasing. Therefore, to reach a stopband attenuation of –60 dB and peak aliasing distortion of -60 dB, the order of coefficients for B has to be at least 4. The results of the studies suggest that the best the order of coefficient are 2, 4 and 6 for filter A , B and H, respectively, with stopband attenuation around –60 dB.

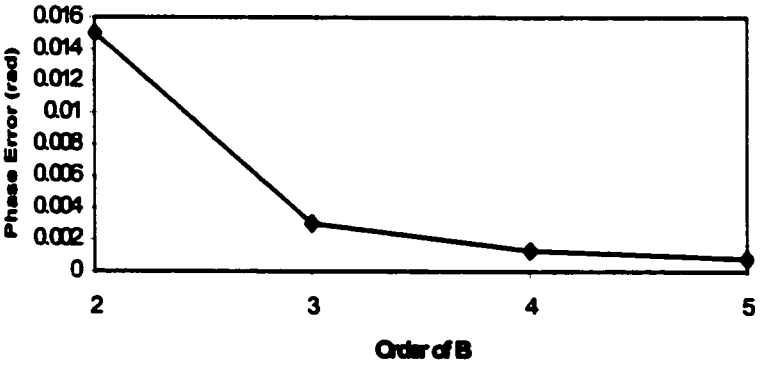

Figure 4-3. Peak aliasing distortion versus the order of allpass filter B.

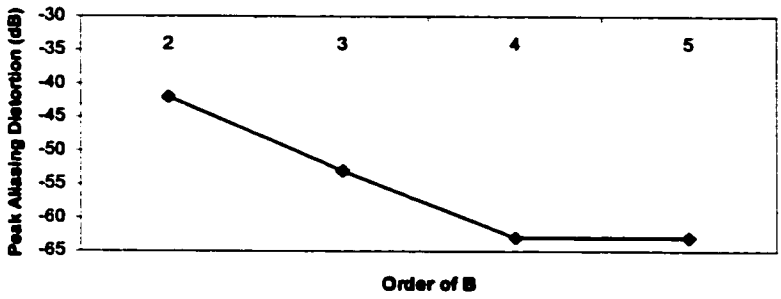

Figure 4-4. Phase error versus the order of allpass filter B.

Simulation with the sound signal #3 has been successfully performed so that the filter bank will give good quality signal reconstruction for the real signal. However, it was found that the filter bank with an order of coefficients of 2, 4 and 6 gives poor reconstruction. The result of the square error is 7×10^{-3}. The simulations were performed against to the order of coefficient for B of 3 and 5. It is surprising that filter bank with the order of 3 gave much better results than the order of 4 for B. The maximum number of square errors is 4×10^{-8}. Therefore, the order of B needs to be an odd number. To prove this, simulation with the order of 6 for B was performed. Both of the square error and amplitude error were greater than the one with the order of 3 for B. Therefore, the order of coefficients for B can only be 3 or 5. The simulation results are summarized in Table 4-1, with the order of 2,3 and 6 for A, B and H for filter bank #1 and the order of 2,5 and 6 for A, B and H for filter bank #2.

Table 4-1. Properties of the filter bank with respect to the order of coefficients

Filter bank	Order	Delay	Peak aliasing (dB)	Amplitude error (dB)	Phase distortion	Max. Square error
#1	2,3,6	543	-54	-55	4×10^{-3}	4×10^{-8}
#2	2,5,6	607	-63	-63	1×10^{-3}	2.5×10^{-9}

Filter bank #1 in Table 4-1 has a delay of 543, and filter bank #2 has delay of 607. The maximum peak aliasing distortion of the modulated filter bank is –54 dB for filter bank #1 and -63 dB for filter bank #2. The phase distortion is 4×10^{-3} and 1×10^{-3} for filter bank #1 and filter bank #2, respectively. Summarizes the various properties of the prototype with respect to the filters used are in Table 4-2.

Table 4-2. Delay of the filter bank with respect to the order of coefficients for each allpass filter

Order(A, B, H)	Delay	Max. Phase error(rad)	Stopband Attenuation(dB)	Passband Ripple
4,7,11	1055	4e-5	-90	3.4e-9
3,5,9	831	3e-2	-76	1.2e-7
3,5,8	767	3e-2	-72	2.0e-7
2,3,7	607	3e-3	-62	4.6e-6
2,3,6	543	3e-3	-60	5.8e-6
2,3,4	415	5e-3	-49	3.3e-5
2,2,3	319	4e-3	-45	7.1e-5
1,2,2	223	4e-2	-29	5.5e-3
1,1,2	191	0.3	-29	5.5e-3

In order to reach stopband attenuation to –60 dB, the orders of the coefficients for A, B and H have to be 2, 3 and 6 or 2, 5 and 6, respectively. For the purpose of easy representation, the filter bank is represented as 2-3-6. Although for the order of 2-3-6 the delay is 543 samples for 32 subbands, this delay is still too long in telecommunications applications. Judging from Table 4-2, in order to meet the delay of less than 200 samples, the order of coefficients would have to be 1-1-2. But if this were the case, the stopbank attenuation would be only –29 dB. Therefore, the filter bank has to be optimized in other parameters to meet the delay requirement.

4.2 The Effect of Transition Band

The sharpness of the filter bank is dependent on the sharpness of allpass filter H from Chapter 3. If the application does not require a filter bank with such sharpness, the order of coefficients for H can be reduced. As Equation 4.2 demonstrates, the H filter produces a delay that is greater than either A or B. The delay of the filter bank can therefore be reduced by increasing transition bandwidth.

A series of filter banks were designed with the transition bandwidth of $H_I(z^2)$ ranging from 0.025π to 0.2π. The transition bandwidth of filter bank can be calculated by transition bandwidth of $H_I(z^2)$ divide by number of bands. In this chapter, when we mention the transition bandwidth, if means the transition bandwidth of $H_I(z^2)$.

The only requirement for this design is that the stopband's attenuation for filter A and H are in the same range. The relationship between the stopband attenuation and the order of H with respect to each transition band is shown in Figure 4-5.

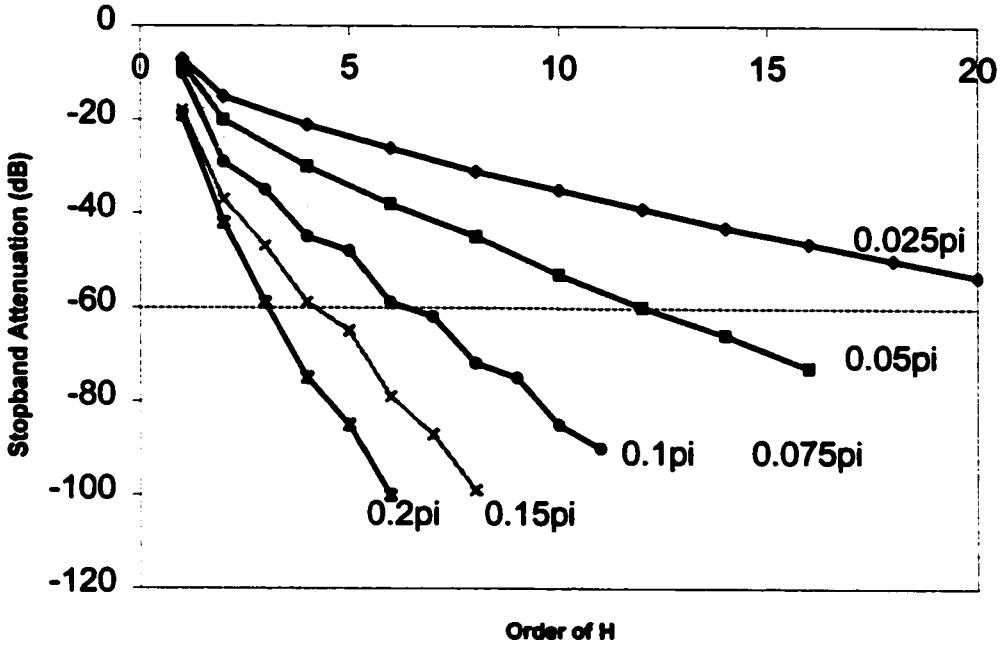

Figure 4-5. The stopband attenuation versus the order of H with respect to each transition bandwidth.

Figure 4-5 demonstrates that the sharper the filter bank, the higher the order of coefficients H is required. To meet a stopband attenuation of –60 dB, the order of allpass filters A and H are listed in Table 4-3. The order of coefficient H needs to be 12 for a transition bandwidth of 0.05π and 3 for transition bandwidth of 0.2π, respectively. The order of coefficients for A can be kept as a constant of 2.

Table 4-3 Order of allpass filter A and H to reach the stopband attenuation of –60dB

Transition bandwidth	0.025π	0.05π	0.075π	0.1π	0.15π	0.2π
Order of A	2	2	2	2	2	2
Order of H		12	8	6	4	3

The order of H with respect to the transition band with a given stopband attenuation of –60 dB is also shown in Figure 4-6. The order of H decreases exponentially with respect to the transition bandwidth.

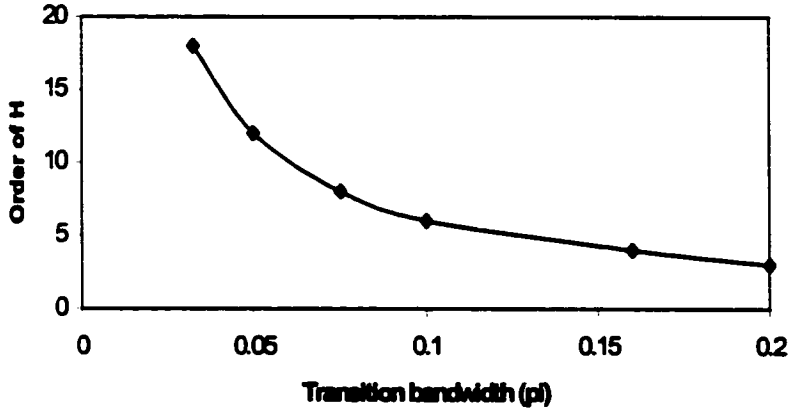

Figure 4-6. The order of allpass filter H versus the transition bandwidth with a stopband attenuation of –60dB.

The order of coefficients for B are dependent on the peak aliasing distortion error, phase error and amplitude error. The order of A and H were kept as in Table 4-2, and the peak aliasing distortion and phase errors were computed by varing the order of B. Figure 4-7 shows the peak aliasing distortion versus the order of allpass filter B with respect to the transition bandwidth. Figure 4-8 shows the phase error versus the order of allpass filter B with respect to the transition bandwidth.

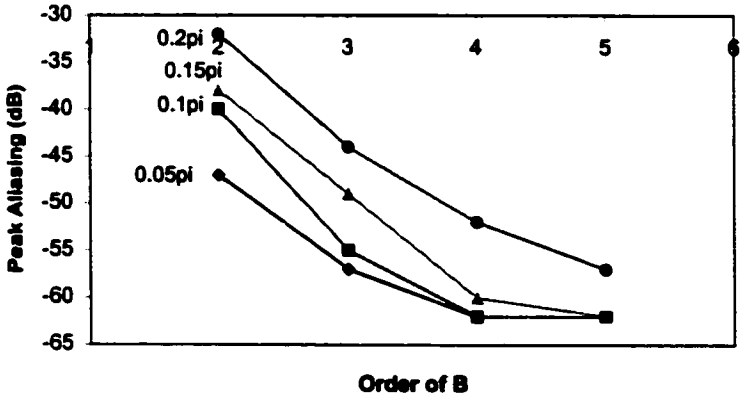

Figure 4-7. The peak aliasing distortion versus the order of allpass filter B with respect to the transition bandwidth.

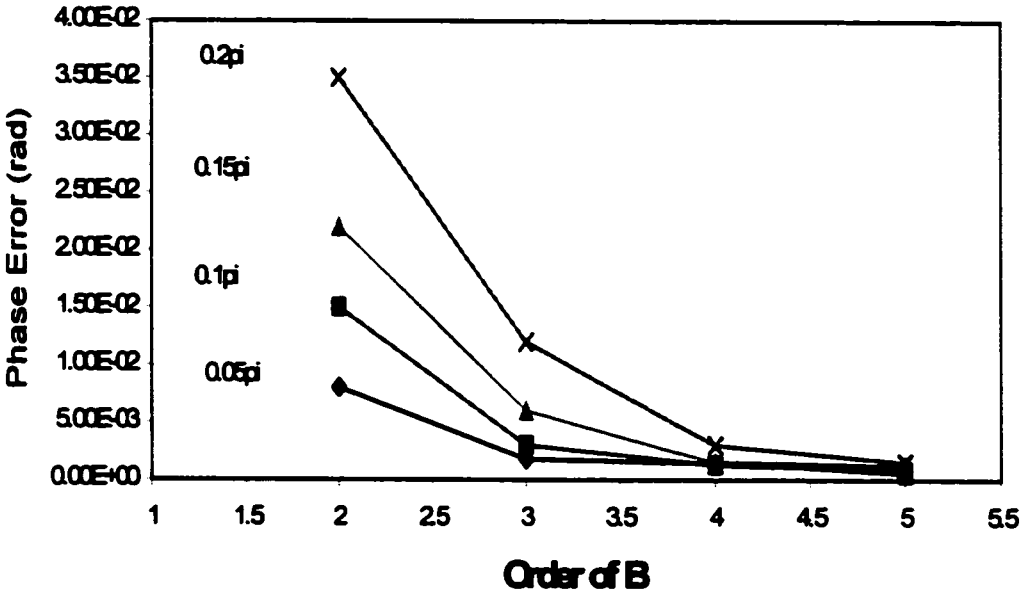

Figure 4-8. The phase error versus order of allpass filter of B respect to the transition bandwidth.

Both the peak aliasing distortion and phase errors have the same trend. When the order of coefficient C increases, the error decreases. When the transition bandwidth increases, the errors increase because there is more overlapping between subbands. To produce a reasonably good performance filter bank, the order of coefficients for B needs to be at least 3 or more. The maximum square errors for the various transition bands are summarized in Table 4-4. Table 4-4 also demonstrates that the order of coefficients has to be an odd number. Therefore, the order of allpass filter of B can be 3 or 5. The delays for each combination were calculated and are shown in Table 4-5. The delay of filter bank can be reduced to 351 if a transition band of 0.2π is used.

Table 4-4. The maximum square error of simulation with signal #3

Order B	0.05π (2, 12)	0.1π(2,6)	0.15π(2,4)	0.2π(2,3)*
2	7.00E-03	7.00E-03	7.00E-03	7.00E-03
3	1.40E-08	1.00E-07	5.00E-07	2.50E-06
4	7.00E-03	7.00E-03	7.00E-03	7.00E-03
5	1.20E-09	1.20E-09	1.60E-09	1.60E-08

- The numbers in brackets are the order of A and H (A,H)

Table 4-5. The delay of the filter bank with respect to the transition bandwidth and the combination of the order of allpass filters A, B and H.

Order (A, B, H)	0.05π	0.1π	0.15π	0.2π
2,3,12	927			
2, 5, 12	991			
2, 3, 6		543		
2, 5, 6		607		
2, 3, 4			415	
2, 5, 4			479	
2, 3, 3				351
2, 5, 3				415

The delay versus of the prototype transition bandwidth is shown in Figure 4-9. The sharper filter with less transition bandwidth gives a longer delay.

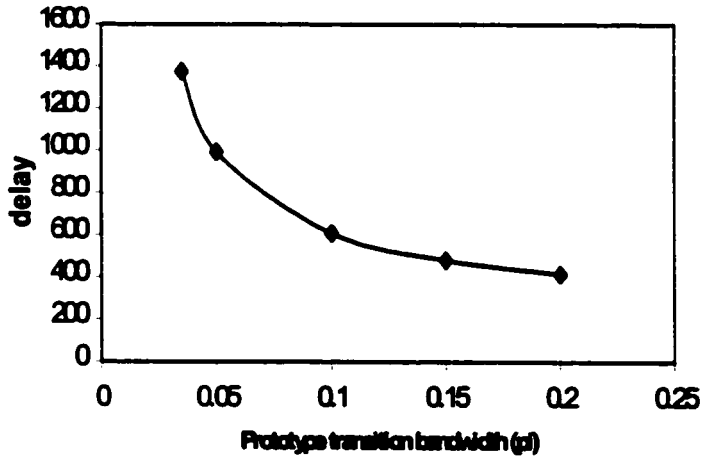

Figure 4-9. The relationship between delay and prototype transition bandwidth (π).

The magnitude responses of the 32 subband filter bank with respect to various transition bands of 32 are shown in Figure 4-10. Only the first four of the 32 subbands are displayed bellow. Figure 4-10 demonstrates the sharpness of the filter bank if no limitation for the sample delay is required. This design therefore produces "brick wall-like" filter bank.

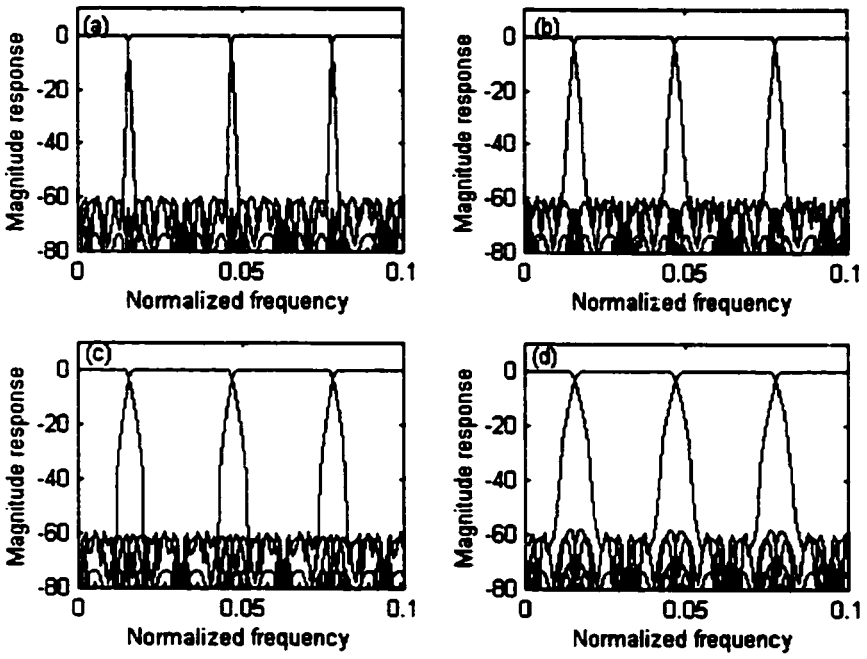

Figure 4-10. The analysis filter bank with transition bandwidth (a) 0.05π, order of 2, 5, 12; (b) 0.1π, order of 2, 5, 6; (c) 0.15π, order of 2, 5, 4 and (d) 0.2π, order of 2, 5, 3.

In order to find the limits of this design method, the transition bandwidth must be reduced in increments of 0.01π when the transition bandwidth is less than 0.05π. The minimum transition bandwidth with the order of H is 0.0325π. For a transition band such as 0.025π, the stopband attenuation can reach –53 dB. The magnitude response of the two filters with transition bands of 0.025π and 0.0325π are shown in Figure 4-11. Therefore, for off-line application where the delay of the filter bank is not so much of a concern, an almost "brick-wall" like filter bank can be designed with a transition band of 0.035π.

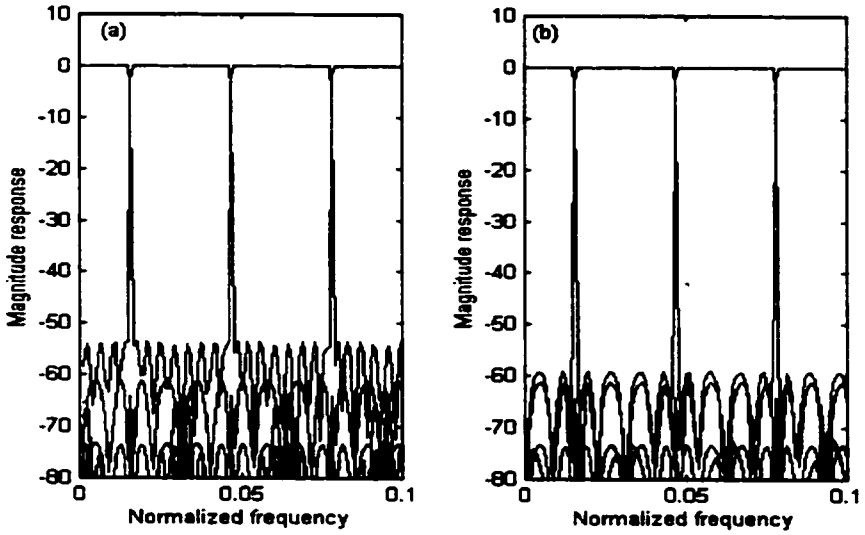

Figure 4-11. The analysis filter bank with sharpest transition band (a) 0.525π, order of 2,5,20 and (b) 0.5325π, order of 2, 5, 18.

4.3 The effect of filter bank structure

The delay of the modulated filter bank is proportional to M, or the number of bands, as shown in Equation 4.2. If a two layered tree-structure filter bank is used, in which M equals to 4 the tree structure filter bank should cascade this small system into a bigger system. The total M should be 4 + 4 = 8. The total delay would be half of a regular modulated filter bank. However, since the filter bank is IIR based, a different delay may occur.

4.3.1 Even Split Tree Structure

The first experiment using a two-layered tree structure is shown in Figure 4-12. After the input signals pass through an 4-band analysis bank, it splits into 8 bands from frequency -π to π. Then each band will pass through another 4-band analysis bank, causing the signal to split into eight groups of 8 bands (2M) each, which equals a total of 64 bands. At the receiver end, each group will pass through the synthesis bank and reconstruct the signal. Then these 8 reconstructed signals will pass through another synthesis bank and produce a final reconstructed signal. Since the signals goes through the first layer of the analysis bank and passes through the second layer sub filter bank, the output signals from the first layer of the analysis bank are complex if DFT modulation are used. These complex output signals will be input signals for second layer filter bank, they cause the

problem due to real part and imaginary part of signal go through analysis bank. Therefore, the DFT modulation cannot be used due to a complex output signal. Therefore, another modulation technique such as DCT modulation or Hartley modulation,(which each of them will be discussed in Chapter 5), must be used.

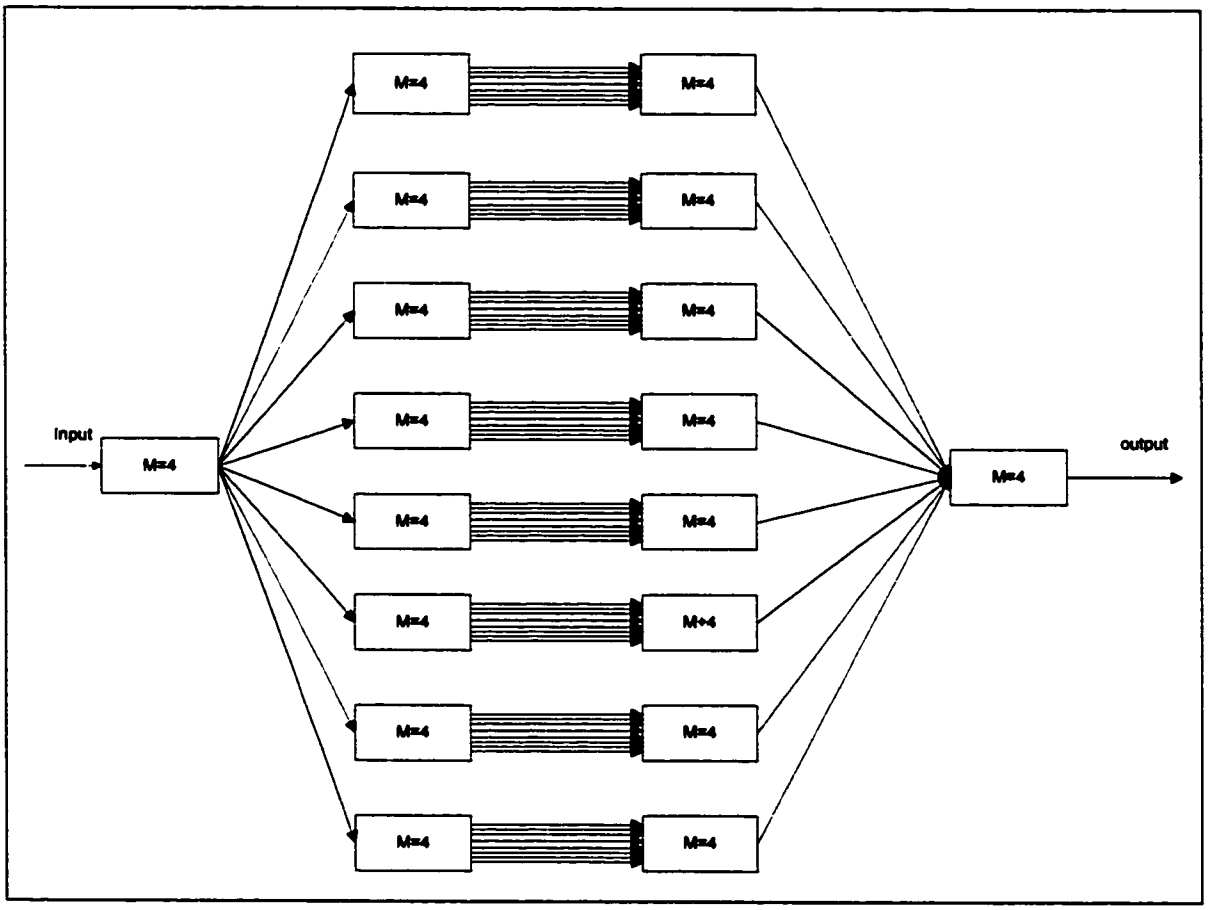

Figure 4-12. A two layered tree structure filter bank with even 4-4 split.

The frequency response of this filter bank is shown in Figure 4-13. The dotted line represents the frequency response of the first layer filter bank and the solid line represents the frequency response for the signal after it passes through the second layer. The smallest bandwidth is 62.5Hz for first and last band. Most of the middle bands have the bandwidth of 250 Hz.

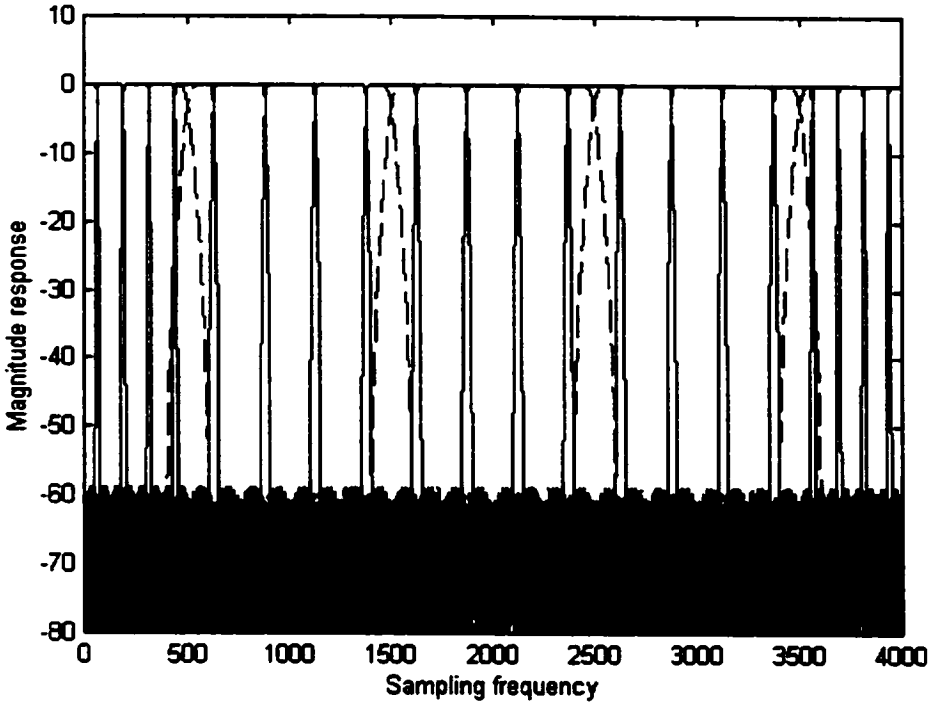

Figure 4-13. The magnitude response of a two layered tree structured filter bank versus a sample frequency with sampling frequency as 8K. The dotted line represents first layer and solid line represents second layer.

This two-layered tree structured filter bank was simulated using MatLab. The M=4 sub-filter bank has an order of coefficients of 2-5-6 with a delay of 75. A simulation with sound signal #3 was performed. Since the filter bank cascaded from a small filter bank, it was expected that the total delay would equal to two times the subsystem's delay. However, the output signal produced a much longer delay; it was five times the subsystems delay. Figure 4-14, demonstrates that (a) the input signal was delayed two times the delay of the subsystem and (b) the input signal was delayed 5 times the delay of the subsystem (c) and display the first 2000 output signals. This means that the total delay is M + 1 times the subsystem. The simulation results are shown in Figure 4-15. The square error and amplitude error show no significant difference from 2M modulated filter bank. The delay for an even split 2M modulated filter bank with 32 subbands can therefore be reduced from 607 to 375.

Another possible structure is an 8-2 split, which means that the signal first splits into 8 bands, then again into two bands. The total delay will be 8*(delay of 2 bands) + delay of 8, which is 447. This delay is even longer than that of the previous design. Therefore, the best split is still 4-4.

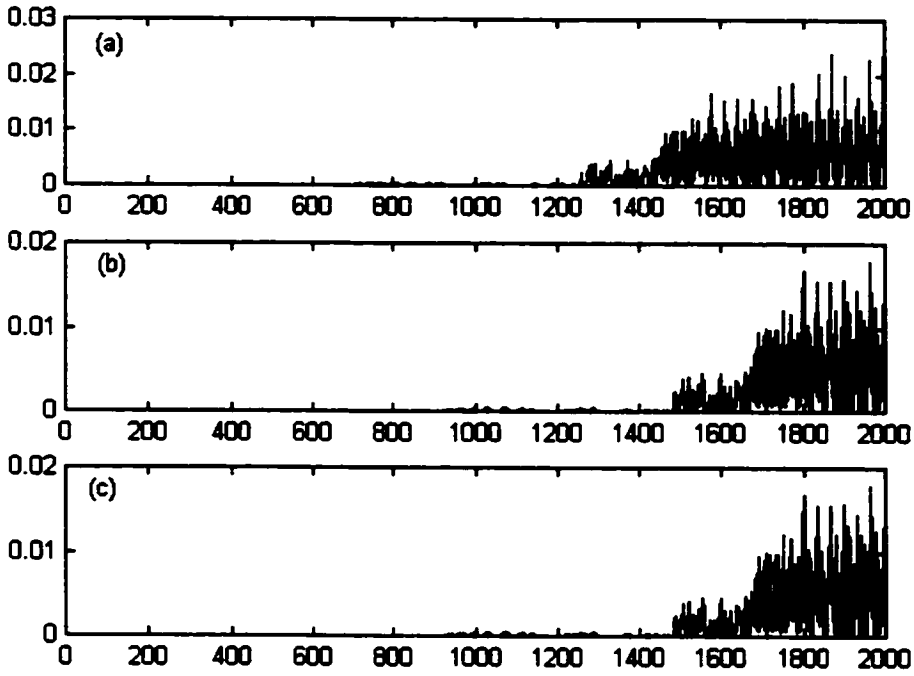

Figure 4-14. (a) input signal delayed for 2 times the subsystem's delay; (b) 5 times the subsystem's delay;(c) the first 2000 output signals.

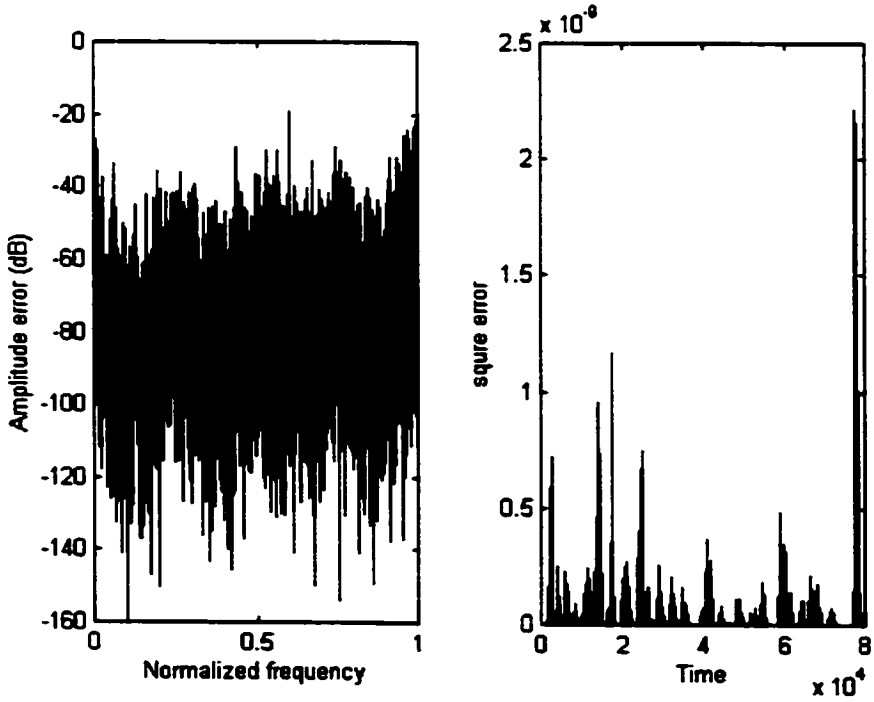

Figure 4-15. Simulation results with a two layered even split tree structured filter bank. The subsystem has order of coefficients are 2, 5, 6 and M of 4.

4.3.2 Non-even Split Tree Structure

In subband sound compression, human ears have a limited frequency response that varies in acuity from less than 100 Hz for the lowest audible frequencies to more than 4 kHz for the highest. Thus the audible spectrum can be partitioned into critical bands that reflect the resolving power of the ear as a function of frequency. The critical bandwidth is narrower for the lower frequencies and wider for the higher frequencies. In order to more efficiently utilize the bandwidth, the filter banks should have non-even bands. A non-even tree structured filter bank is shown in Figure 4-16. The signal first pass through a four subband modulated filter bank which produces eight bands to cover frequency range from $-\pi$ to π. The half of bandwidth of band 1 and 5 is in negative frequency range. The band 6, 7 and 8 are in negative frequency range too. Since DCT or Hartley transform used in tree structured filter bank, all bands (2M) have to be transmitted. Then two bands of the signals which range from 0 to 1500 Hz, pass through another 4 subband modulated filter bank. The last band with the highest frequency does not go through a second layer, but the middle two bands pass through a 2 subband modulated filter bank.

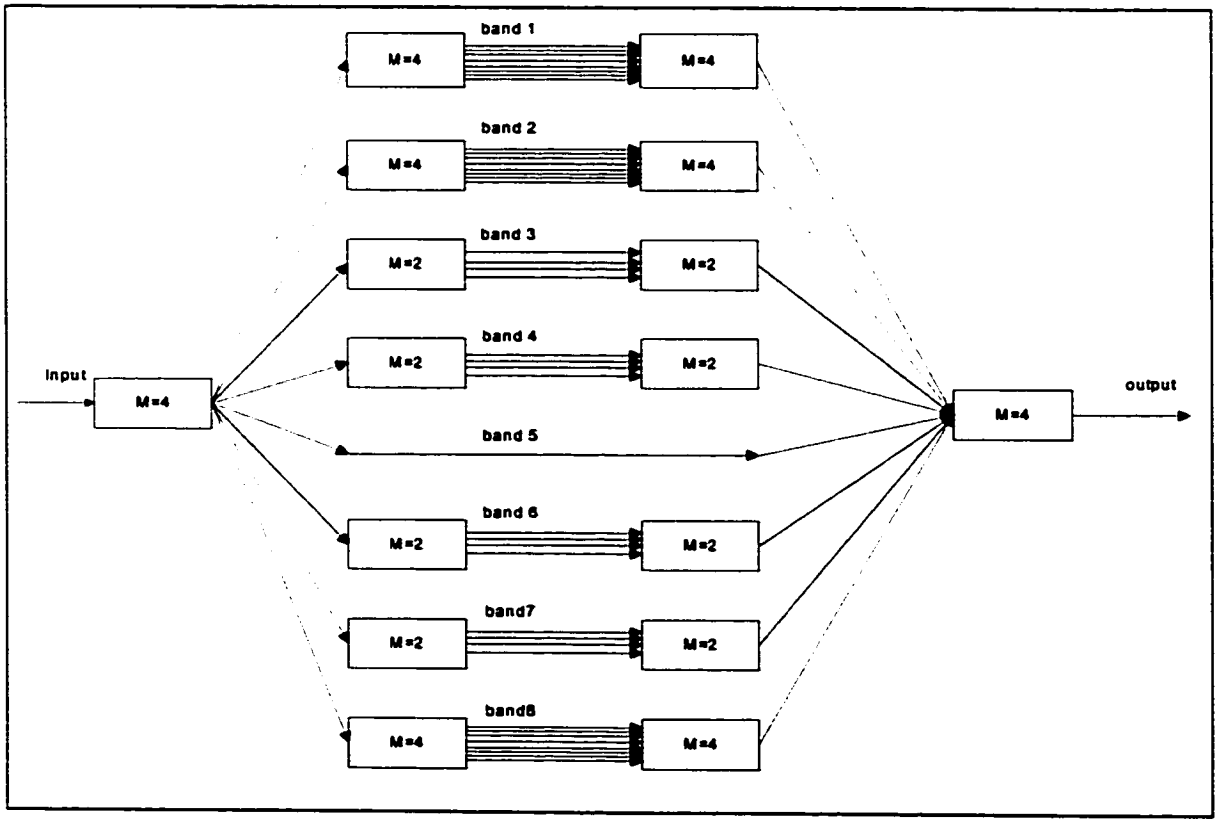

Figure 4-16. A two layered tree structure filter bank with a non-even split.

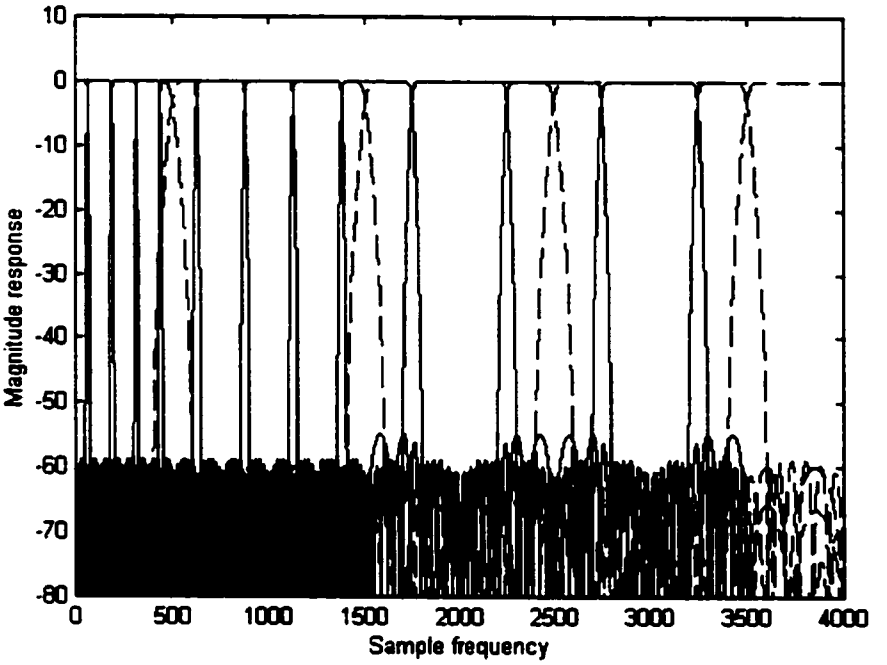

Figure 4-17. The magnitude response of a two layered non-even tree structure filter bank versus sample frequency with sampling frequency as 8K (-- first layer; _second layer).

The frequency response of the filter bank is shown in Figure 4-17. The bandwidth is small at lower frequencies and large in higher frequencies. The band boundaries of the tree-structured filter band and original 2M = 32 subband modulated filter bank are compared with the approximate critical band boundaries [1] in Table 4-6. The 2M modulated filter bank wasted a lot of bandwidth in the higher band frequency. The bandwidth of a tree structured filter bank is the same as a 2M modulated filter bank in a lower frequency, but it is much wider in a high frequency. The bandwidth of a tree structured filter bank meets the critical band requirement. The number of bands is reduced from 32 to 17 for the tree structured filter bank.

For 2M-band filter bank, when DFT was used as modulation technique, it generates 2M band with imaginary symmetry, which two bands are real signal only. Therefore, it only needs to transmit M real signal and M - 2 imaginary signals rather than 2M real signal and 2M-2 imaginary signals. For DCT and Hartley transformation, 2M real output bands are no symmetric relationship, therefore 2M bands need to be transmitted. Modulation techniques will be discussed in Chapter 5. For 17 tree structured filter bank, it needs to transmit a total of 41 bands. It is calculated as following:

$$3 \times 8 + 4 \times 4 + 1 = 41$$

where 8 from band 1, 2 and 8, 4 from 3, 4, 6 and 7, 1 from band 1. 41 is much less than 64 bands required by 2M modulated filter bank.

Table 4-6. The band boundaries of a 2M 32 subband modulated filter bank and tree structured filter band compared with the band boundaries of the critical band

Band Boundaries		
2M-modulated filter bank	Tree structured filter bank	Critical band frequency[1]
62.5	62.5	50
		95
187.5	187.5	140
		235
312.5	312.5	330
437.5	437.5	420
562.5	500	560
687.5	625	660
812.5	875	800
937.5		940
1062.5	1125	1125
1187.5		
1312.5	1375	1265
1437.5		
1562.5	1500	1500
1687.5	1750	1735
1812.5		
1937.5	2250	1970
2062.6		
2187.5		
2312.5	2500	2340
2437.5		
2562.5		
2687.5	2750	2720
2812.5		
2937.5		
3062.5		
3187.5	3250	3280
3312.5		
3437.5		
3562.5	3500	3840
3687.5		
3812.5		
3937.5		
4000.0	4000	4690

Since the filter bank has a non-even split, the delay has to be synchronized with the longest delay in all the bands. The total delay is the same as an 4-4 even split tree structure. In order to ensure that the signal is reconstructed properly, the errors were computed when the signal reconstructed after the second layer. The square error of the signal for each band is shown in Figure 4-18. The square errors in all bands are within 10^{-9} range.

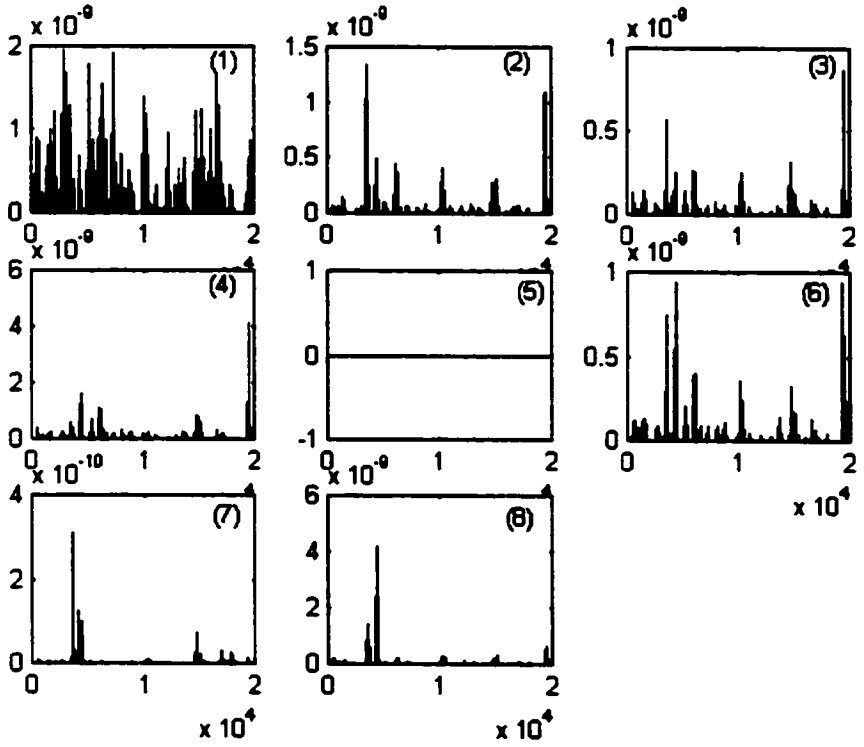

Figure 4-18. Square error of each band when signal reconstructed after second layer.

The amplitude errors and square errors are shown in Figure 4-19. These errors show no significant difference from the even split filter bank and the 2M modulated filter bank.

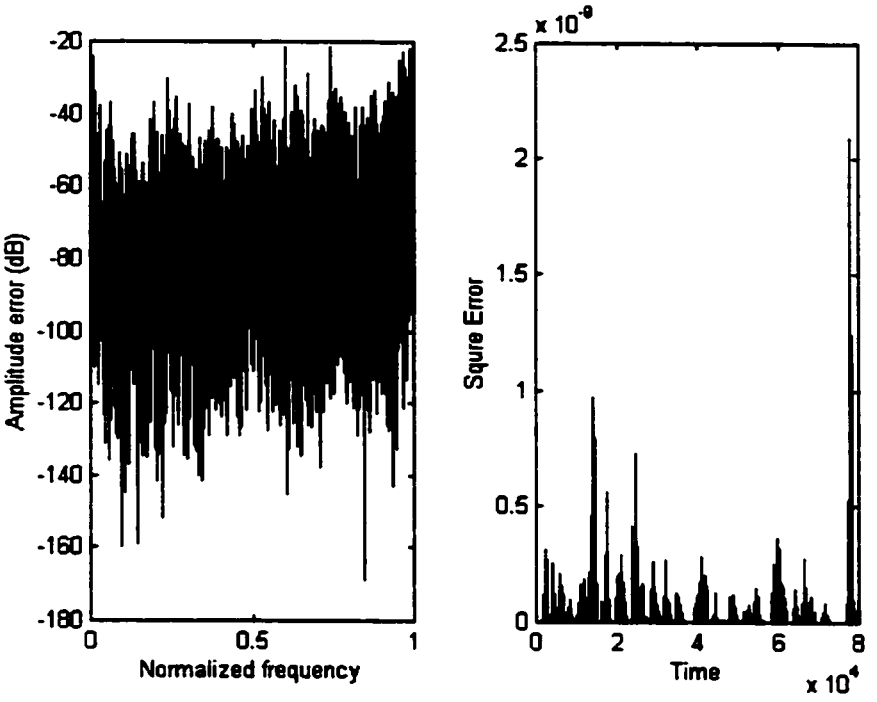

Figure 4-19. Simulation results with un-even split tree-structured filter bank.

4.4 The effect of band number

According to Equation 4.2, the delay of the modulated filter bank is proportional to the number of bands. The minimum delay for a 2M modulated filter bank is 607 with a stopband attenuation of –60dB and a prototype transition band of 0.6π. Increasing the prototype transition band to 0.7π reduces the delay to only 415. If a tree structure were used, the minimum delay can be reduced to 375. This delay is still too large for real time applications, which require a maximum of 200 delay. The only way to further reduce the delay is to reduce the number of bands. However, reducing the number of bands would affect the sharpness of the filter bank because the transition bandwidth of filter bank equals the prototype transition bandwidth divided by the number of bands. Figure 4-10 shows that the relationship between the delay and the transition bandwidth of filter bank with respected to the number of bands.

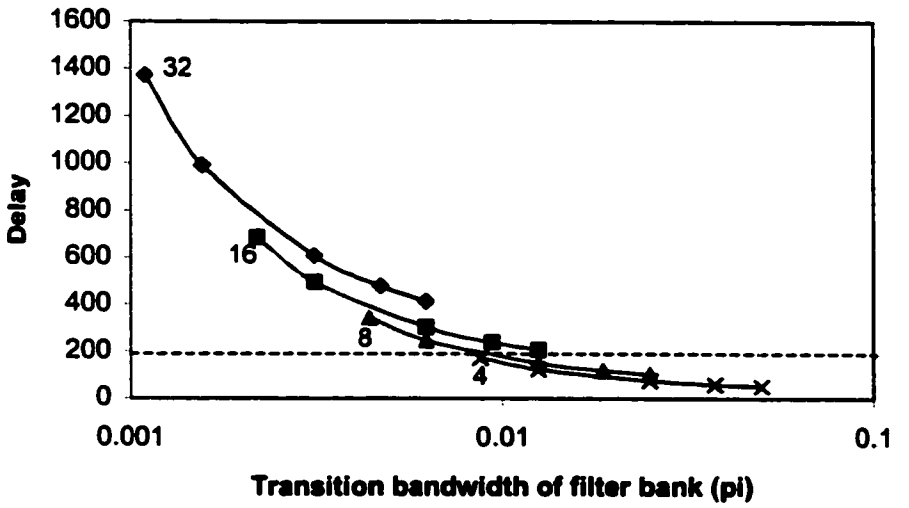

Figure 4-20. Relationship between the delay and the transition bandwidth of the filter bank with respect to the number of bands.

For a band number of 32, the minimum delay can reach only 415 with a transition bandwidth of $0.2/32\pi$. If the band number can be reduced to 16, the minimum delay is 207 with a transition bandwidth of $0.2/16\pi$. This delay is still slightly higher than 200. If the band number can be further reduced, the delay would have no problem meeting the requirement. The relationship between the delay, transition bandwidth and orders are summarized in Table 4-7.

Table 4-7 Filter bank properties with delay around 200.

Number of band	Order	Prototype Transition bandwidth(π)	Transition bandwidth(π)	Delay
16	2,5,3	0.2	0.0125	207
8	2,5,6	0.1	0.0125	151
4	2,5,12	0.05	0. 0125	123
4	2,5,18	0.035	0. 00875	171

The first three filter banks in Table 4-7 have the same transition bandwidths but number 4 has the sharpest filter. The magnitude responses of the filter banks are shown in Figure 4-21.

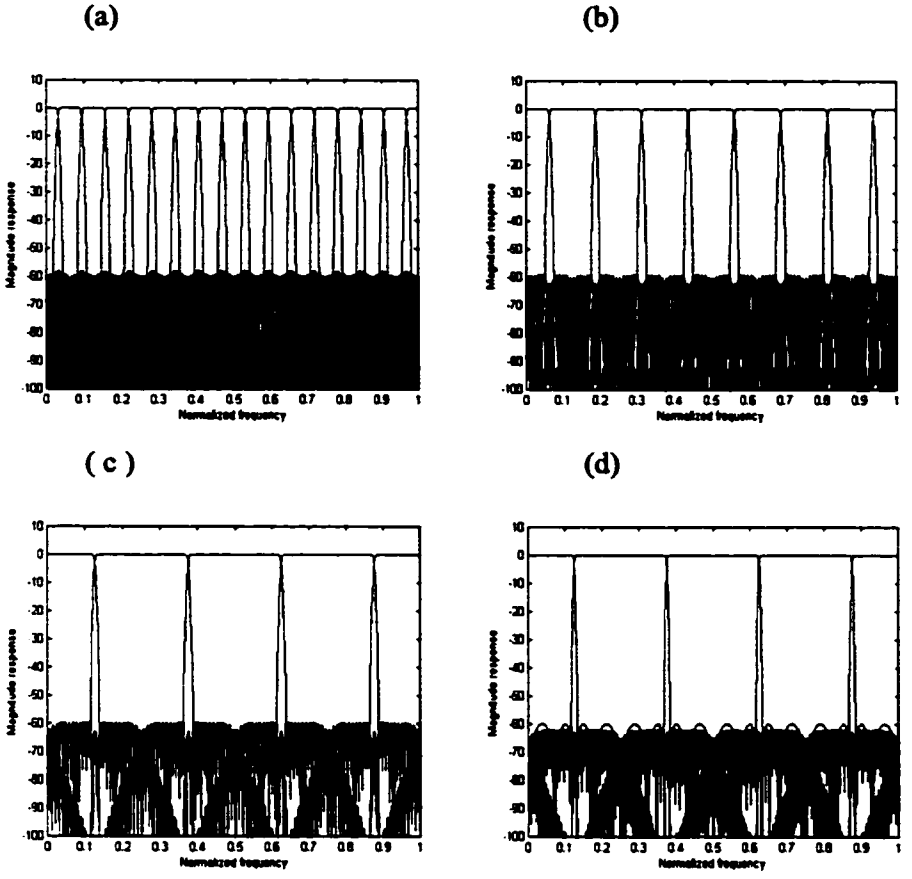

Figure 4-21. Magnitude response of the filter banks with (a) 16-band with an order of 2,5,3 and a prototype transition bandwidth of 0.2π; (b) 8-band with an order of 2, 5,6 and a prototype transition bandwidth of 0.1π; (c) 4-band with an order of 2, 5, 12 and a prototype transition bandwidth of 0.05π; (d) 4-band with an order 2, 5, 18 and a prototype transition bandwidth of 0.035π.

Using these four filter banks, the simulations were performed with sound signal #3. The amplitude errors and square errors are summarized in Figure 4-22. The square errors range from 10^{-8} to 10^{-10}. The smaller band numbers will give lower square errors when both the filter banks have the same transition bandwidth. Lower transition bandwidths give lower square errors when there is the same number of the bands in the filter bank.

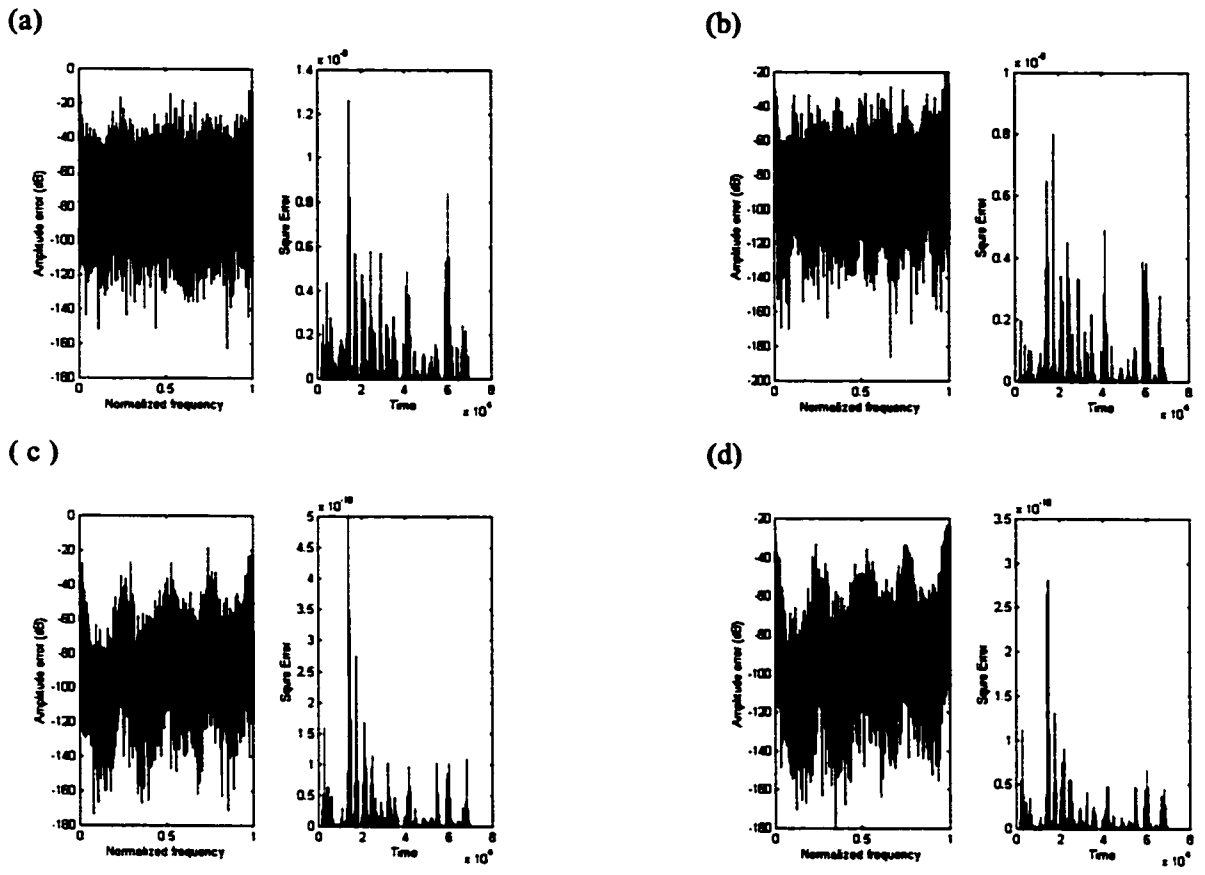

Figure 4-22. The simulation results with (a) 16-band with an order of 2,5,3 and a prototype transition bandwidth of 0.2π; (b) 8-band with an order of 2,5,6 and a prototype transition bandwidth of 0.1π; (c) 4-band with an order of 2,5,12 and a prototype transition bandwidth of 0.05π; (d) 4-band with an order of 2,5,18 and a prototype transition bandwidth of 0.035π.

4.5 Summary

The minimum delay for a 2M modulated filter bank is 607 with a stopband attenuation of –60dB and a prototype transition band of 0.6π. Increasing the prototype transition band to 0.7π reduces the delay to 415. If a tree structure were used, the minimum delay can be reduced to 375. This delay is still not good enough for real time applications. In order to reduce the delay to under 200 samples, the number of bands have to be reduced. The filter bank could have 16 bands or less. The effects of reducing the number of bands on data compression are yet unknown. Determining these effects requires more tests with respect to data compression, tests that (unfortunately) exceed the scope of this thesis.

Chapter 5: Other Design Considerations

5.1 Modulation Technique in Filter Bank

As we mentioned previously, to design a M-channel filter bank, the prototype filters of analysis and synthesis banks need to be designed first. Then the frequency needs to be shifted to obtain the M analysis filters and M synthesis filter by modulation. The greatest advantage of using modulation technique is the simplicity of its design and the speed of its implementation. Several techniques for modulation have been used in the literature. The most popular technique is DFT, which is used in both FIR and IIR based filter banks. DCT modulation is found only in FIR based filter banks. Hartly transform modulation has been used only in image processing with FIR based filter banks. In this thesis, both Hartly and DCT modulation are investigated.

5.1.1 DFT modulation

The original design from A. J. Leest et al.[17] used DFT modulation to modulate the prototype. In this design, the analysis filters were obtained by modulating the low pass prototype. The synthesis bank was modulated back by an inverse DFT matrix. The analysis filter bank is shown in Figure 5-1 where M equals to 4. In order to reduce the complexity of the calculation, the DFT matrix can be replaced by FFT.

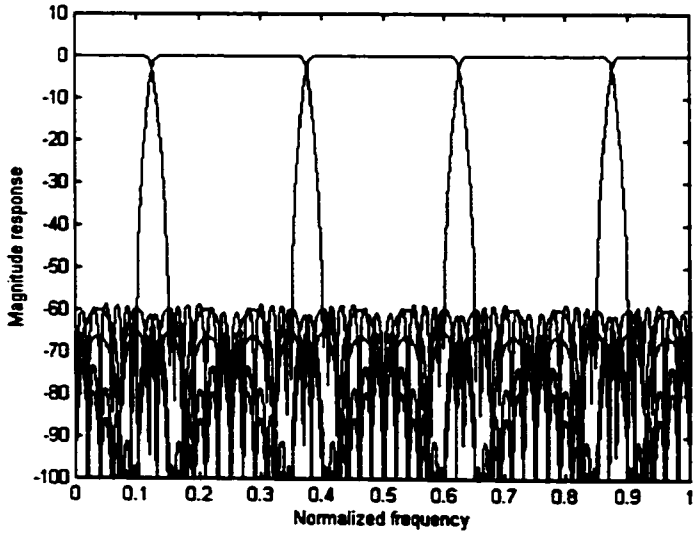

Figure 5-1. Filter bank using DFT modulation with M of 4 and an order of coefficients of 2-5-6.

The greatest problem with DFT modulation is that after the signal passes through the analysis bank, it becomes complex number. This means that double the bandwidth is needed to transmit the same amount of the signal.

5.1.2 Cosine transform modulation

Cosine modulation is widely used in FIR based filter banks. A DFT modulation occurs when the signal is modulated exponentially. A cosine modulation, on the other hand, means that the signal is modulated by a cosine function. Exponential modulation shifts the frequency in one direction, but since cosine modulation is the sum of two exponential modulations, it shifts the frequency in two directions. Four types of cosine modulation are presented in Equation 2.26. The most popular DCT modulation are type II and type IV as discussed in Chapter 2. Since the IIR filter is a feedback system, DCT modulation cannot be used directly. In the literature [16], only Nguyen designed a two-band IIR based cosine modulated filter bank, where the overlap of the filter bank was quite large.

First of all, we are using the cosine modulation method as one in FIR based filter banks. Since the prototype has a passband edge frequency of $\pi/2M$ and a bandwidth of π/M, it just satisfies the condition for an M-band cosine modulation. Therefore, M polyphase and Mth modulation have been tried with four types of cosine modulation techniques. But these techniques were not successful. The frequency response shows multiple bands for each band. Then, we found that the cosine modulated analysis and synthesis filters as defined in Equation 2.28 (pp. 24), can be rearranged into Equation 2.29 (pp.24) using an FIR filter. Since the IIR filter is a feedback system, we cannot simply rearrange equations from Equation 2.28 to generate Equation 2.29. Does this mean that we cannot use cosine modulation?

Since the prototype is designed for 2M polyphase components, we must go back to the 2M-band filter bank to see if we can use the cosine modulation. First of all, we use the cosine part of DFT modulation to modulate the prototype (Type I). This makes the frequency shift to two ways. In their book, Strang and Nguyen [18](p326) pointed out that the constant matrix for cosine modulation

needs to be orthogonal in the time domain. However, if we use the cosine part of DFT matrix, it is not orthorgonal. For example, if M = 2, the real part of the DFT matrix of 4 is

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}$$

And the maxtrix C x C' is not equal to I. The simulation results also produce a large big square errors and amplitude errors.

$$C \times C' = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 4 & 0 \\ 0 & 2 & 0 & 2 \end{bmatrix}$$

The magnitude response of each band using Type II DCT modulation is shown in Figure 5-2. Type IV gives similar results. Type III can make the frequency shift but has no M+1 bands.

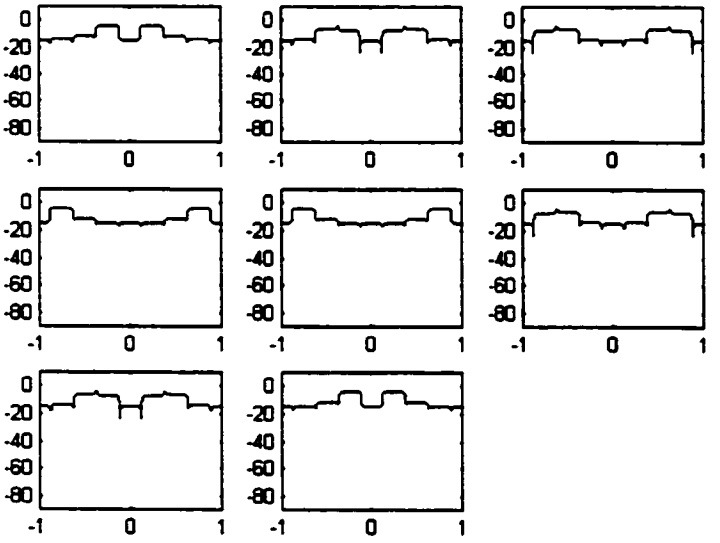

Figure 5-2. Type II DCT modulation.

When ¼ π was added to for Type I and Type III cosine functions, they both become orthogonal and shift frequency to both directions. Magnitude response versus normalized frequency for each band with modified Type I DCT modulation is shown in Figure 5-3.

Matlab codes for modified Type I DCT modulation are shown below.

```
% modified Type I DCT modulation

Ai=zeros(2*M,2*M);
for n=1:2*M,
   for k=1:2*M,
      Ai(n,k)=sqrt(2)*cos(pi*(n-1)*(k-1)/(M) + pi/4);
   end;
end;
```

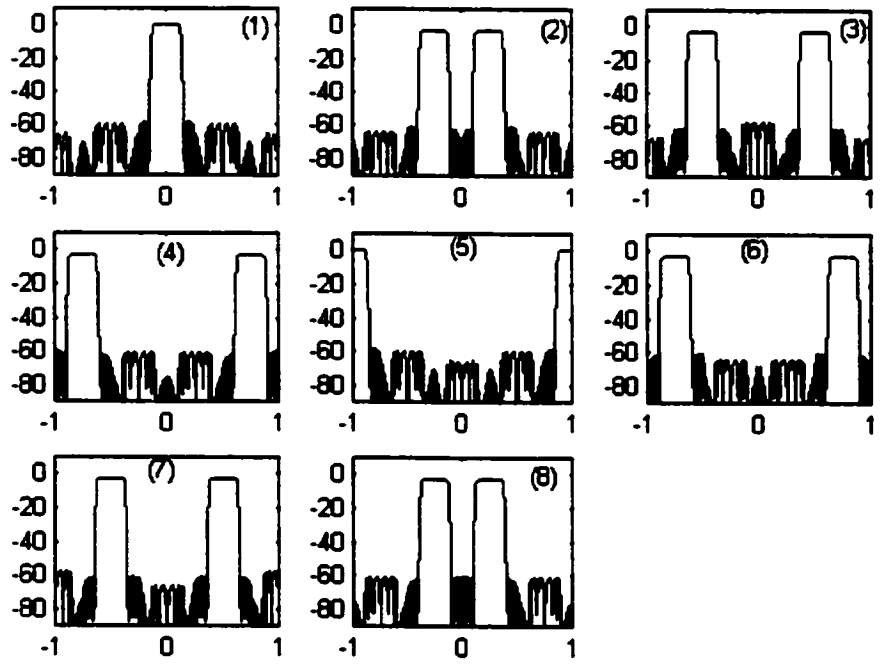

Figure 5-3. Magnitude response versus normalized frequency for each band with modified Type I DCT modulation and M = 4.

In Figure 5-3, band #2 and band #8 cover the same frequency bands. Adding band #2 and #8, band #3 and #7, and band #4 and #6 produce the same results as those produces by DFT modulation. If band #2 and #8 are the same, it is not necessary to transmit the 2M bands signal any more. A simulation has been performed with signal #3. The output signals are plotted in Figure 5-4, which demonstrates that though band #2 and #8 are quite similar their magnitudes are not exactly the same. Therefore, it is still necessary to transmit 2M bands signals.

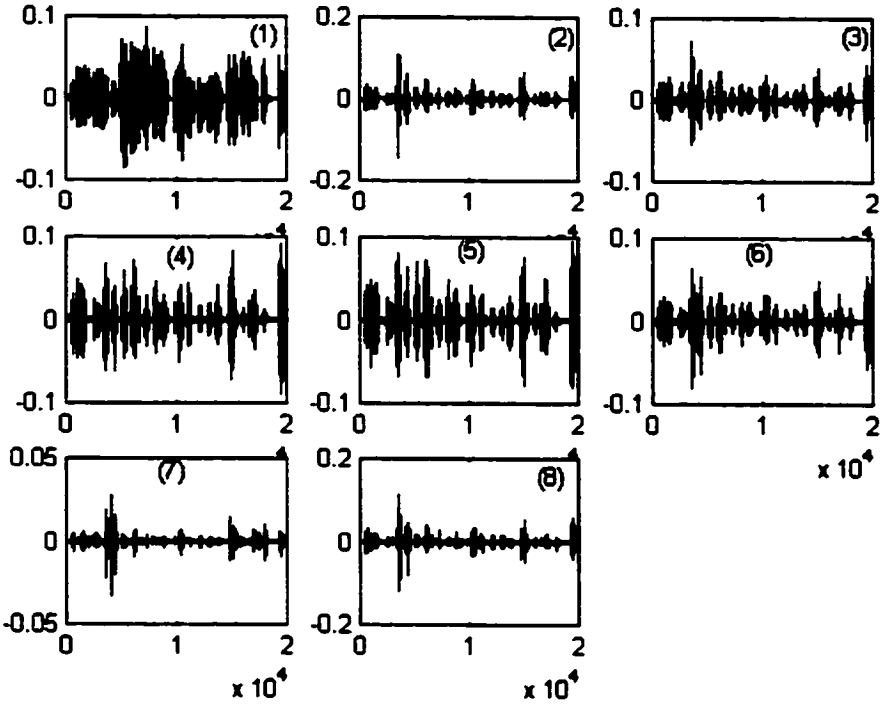

Figure 5-4. Output signals for each band after the signal has passed through analysis bank with modified Type I DCT modulation.

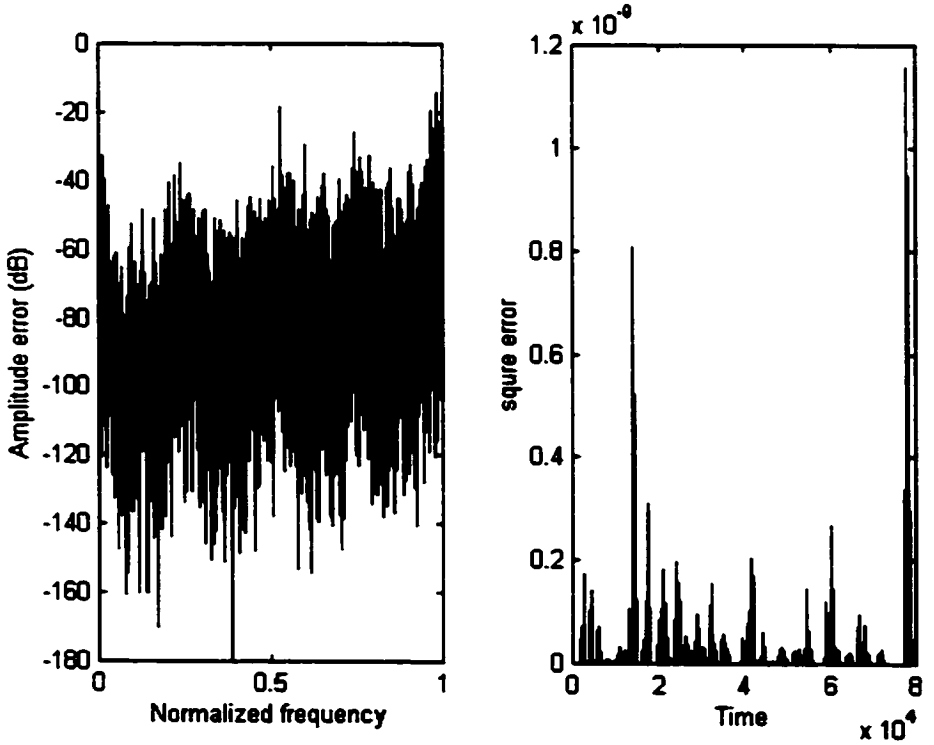

Figure 5-5. The simulation results with modified Type I DCT modulation with M = 4 and the order of coefficients is 2,5, 6.

The amplitude error and square error are shown in Figure 5-5. They are not significantly different from the errors produced by a DFT modulated filter bank. Since DCT modulation produces real output signals when the input signals are real, it can be used in tree structured filter banks, whereas DFT modulation cannot.

5.1.3 Hartley transform modulation

As we discussed in Chapter 3, Hartley transformation has been used in imaging processing [18]; however, no one has yet designed a Hartley modulated filter bank.

The Hartley transformation produces real output from a real input signal. If Hartley transform modulation can shift frequency, it would solve the problem introduced by DFT modulation, which generates complex signal from a real input signal. The Hartley transform can be expressed as $H(a) = \Re F(a) - \Im F(a)$. In Matlab, the Hartley transform can be derivated from FFT by the following:

```
%Hartley transform
 for k=1:R/M,
   YI(:,k)=(fft(YK(:,k)));
   YII(:,k)= real(YI(:,k) - imag(YI(:,k));
 end;
```

The inverse transform is the same function as above. The result of magnitude response with respect to normalized frequency for each band is displayed in Figure 5-6. Only 4 bands were used. The first band and M+1th (5^{th}) band are the same as FFT, the second and the 8^{th} bands shift to both sides, while the 3^{rd} and 7^{th} bands shift even further. Therefore, in Figure 5-6, the 2^{nd} and the 8^{th}, the 3^{rd} and the 7^{th}, and the 4^{th} and the 6^{th} bands cover the same frequency range. When we combine these bands in Figure 5-7, the 1^{st} and the 5^{th} bands have a magnitude response of 1 and the other bands have a response of less than 1. However, if we combine the remaining bands of the same frequency, we find that their magnitude responses all equal to 1, as Figure 5-8 demonstrates.

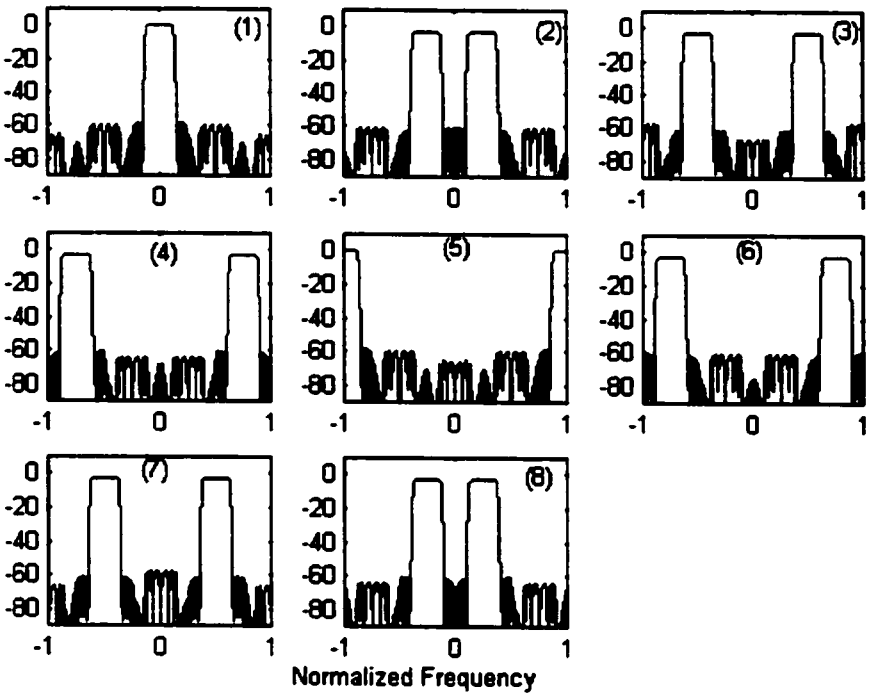

Figure 5-6. Individual bands of magnitude response versus normalized frequency with Hartley transform modulation.

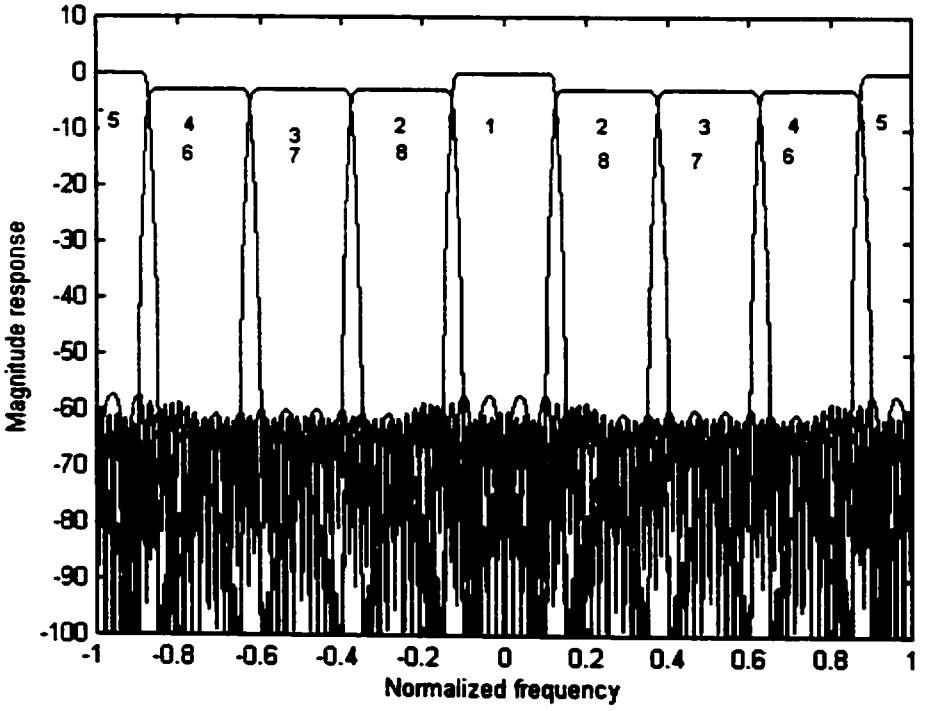

Figure 5-7. Modulated filter bank with Hartley transform.

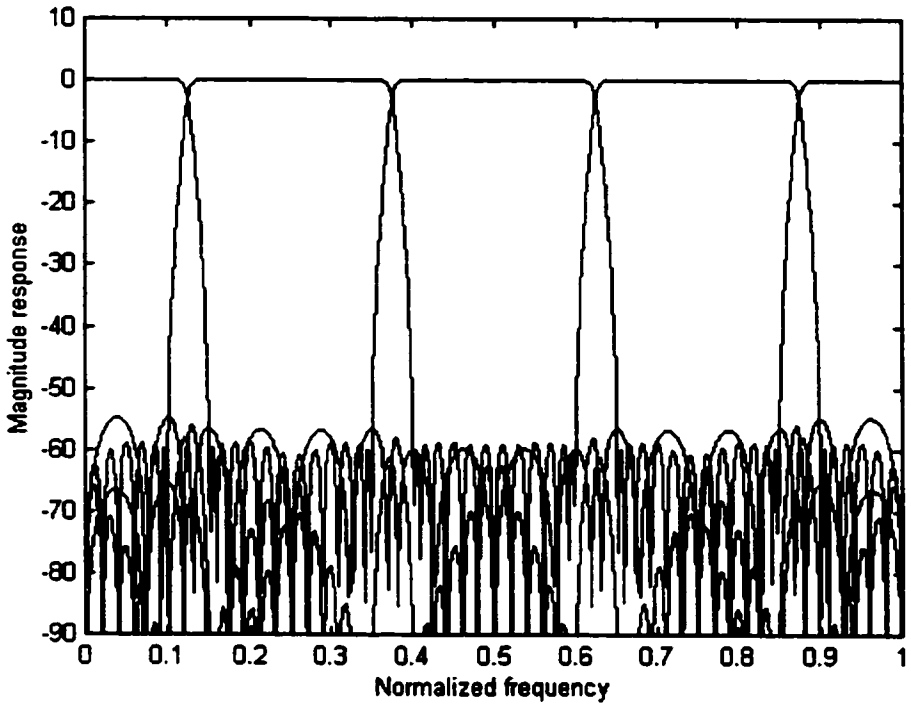

Figure 5-8. The modulated filter bank with Hartley transform, which combines the two bands with the same frequency range.

As Figure 5-7 demonstrates, the two bands that cover the same frequency range have the same magnitude response. If the two bands have the same signals, we can reduce the transmission to 2M bands to M bands.

A simulation has been performed with sound signal #3 and a filter bank with the order of coefficient of 2-5-6 and 4 subbands. The output signals are shown in Figure 5-9 for each band. Band #2 and #8 did not produce the same signal and although their patterns are quite similar, their magnitudes are different. Band #3 and #7, and #4 and #6 produce a pattern similar to band #2 and #8. Therefore, the number of bands for transmission still cannot be reduced. The filter bank still needs to transmit 2M bands signals.

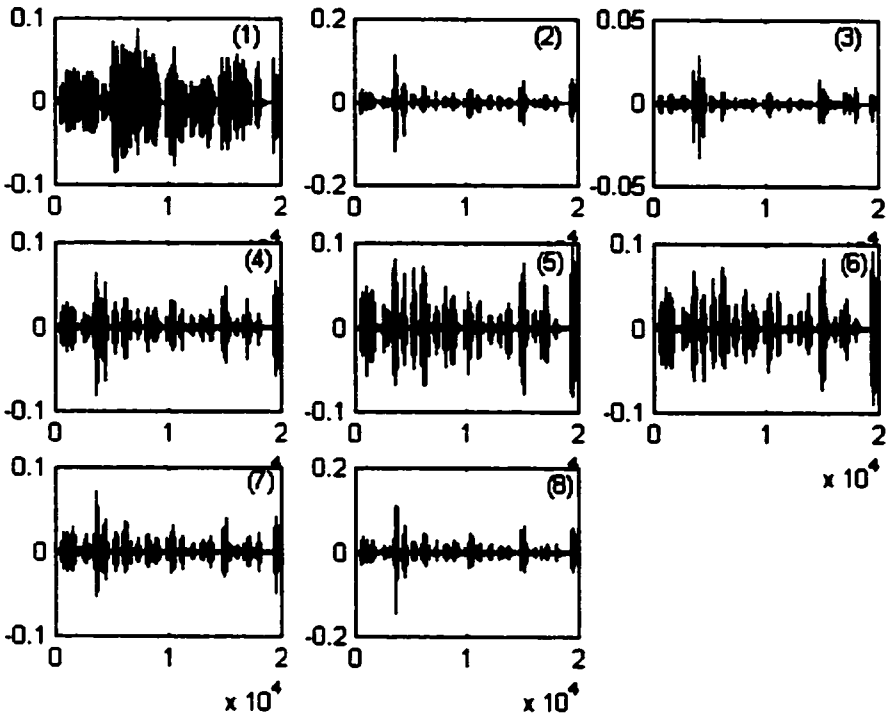

Figure 5-9. Output signals for each band after the signal has passed through analysis bank with Hartley modulation.

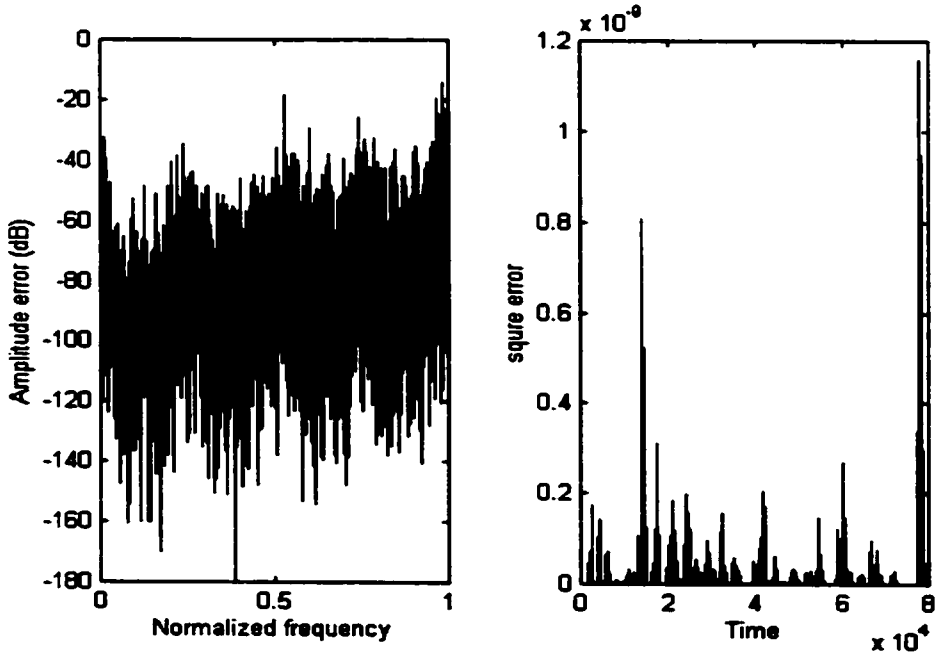

Figure 5-10. The simulation result with Hartley transform with M = 4 the order of coefficients of 2, 5, 6 for A, B, and H.

The amplitude and square error are shown in Figure 5-10. They are not significantly different from the errors produced by DFT modulation. Therefore, the Hartley transform could be used as a modulation technique in the filter bank.

When comparing Hartley modulation (Figure 5-6) with DCT modulation (Figure 5-3), we found that the results are similar. Even in time domain, as expressed in Figure 5-9 and Figure 5-4 the results are similar. The difference between the Hartley and DCT modulation is that the band #2 and #8, #3 and #7, and #4 and #6 are reversed. The simulation results in Figure 5-5 using DCT modulation are the same as those in Figure 5-10 using Hartley modulation. Thus, Hartley transformation could also be used in tree structured modulated allpass based filter banks.

5.2 Quantization Errors in Filter Bank Coefficients

In the real world, the filter bank has to be implemented by using a finite number of bits. Quantization errors usually include the signal quantization error and filter bank coefficients quantization error. In this chapter, we will discuss only the coefficient quantization error.

First of all, quantizing the filter coefficients into a finite number of bits will alter the position of the poles and zeros of H(z) in the z-plane. If the poles are too close to the unit circle, after quantization, they could deviate to the outside circle and thereby make the filter unstable. The fewer the number of bits used to represent the coefficients, the greater the deviation of the poles and zeros position. The filters used are stable real allpass filters since the poles are far from zero. More than likely, the number of bits would not make this type of filter bank unstable. However, the deviations in the locations of the poles and zeros could also lead to deviations in the frequency response. The changes in the passband are caused primarily by changes in the position of the poles, and those in the stopband by changes in the location of zeros.

As reference [19](p431) indicates the coefficients are quantized to B bits by rounding as follows:

$$Bl = \begin{cases} round(bl \times 2^{B-1} + 0.5)/2^{B-1} & if\ \ bl \geq 0 \\ round(bl \times 2^{B-1} - 0.5)/2^{B-1} & if\ \ bl < 0 \end{cases} \qquad (5.1)$$

where b1 is a real coefficient and B1 is quantized coefficient. However, when this method was used in quantizing the coefficients, the simulation results produced an over flow with 10 bits. When we try not to use ±0.5 before rounding the value, we only get an overflow with 6 bits. Therefore, the coefficients were quantized as follows

$$B1 = round(b1 \times 2^{B-1}) / 2^{B-1} \qquad (5.2)$$

A filter bank with M of 4 and an order of 2, 5,6 for allpass filters A, C, H is chosen as example. The coefficients are listed in Table 7.1-7.3 before and after quantization. The filters are implemented in the direct form. Since the first filter is a pure delay, the coefficient is equal to 0.

Table 5-1 The coefficients of allpass filter A before and after quantization

i	j	Ideal	6 bits	8 bits	16 bits
0	1	0.000000	0.000000	0.000000	0.000000
	2	0.000000	0.000000	0.000000	0.000000
1	1	0.203291	0.218750	0.203125	0.203278
	2	-0.032032	-0.031250	-0.031250	-0.032043
2	1	0.431296	0.437500	0.429688	0.431305
	2	-0.046585	-0.031250	-0.046875	-0.046570
3	1	0.692571	0.687500	0.695313	0.692566
	2	-0.038630	-0.031250	-0.039063	-0.038635

Table 5-2. The coefficients of allpass filter H before and after quantization

i	Ideal	6 bits	8 bits	16 bits
1	0.489436	0.500000	0.492188	0.489441
2	-0.109793	-0.125000	-0.109375	-0.109802
3	0.044843	0.031250	0.046875	0.044830
4	-0.020525	-0.031250	-0.023438	-0.020538
5	0.009080	0.000000	0.007813	0.009094
6	-0.004175	0.000000	-0.007813	-0.004181

Table 5-3. The coefficients of allpass filter B before and after quantization

I	J	Ideal	6 bits	8 bits	16 bits
0	1	1.000000	1.000000	1.000000	1.000000
	2	0.000000	0.000000	0.000000	0.000000
	3	0.000000	0.000000	0.000000	0.000000
	4	0.000000	0.000000	0.000000	0.000000
	5	0.000000	0.000000	0.000000	0.000000
1	1	0.786463	0.781250	0.789063	0.786469
	2	-0.119101	-0.125000	-0.117188	-0.119110
	3	0.043093	0.031250	0.046875	0.043091
	4	-0.008822	0.000000	-0.007813	-0.008820
	5	0.001433	0.000000	0.000000	0.001434
2	1	0.526672	0.531250	0.523438	0.526672
	2	-0.144701	-0.156250	-0.148438	-0.144714
	3	0.061103	0.062500	0.062500	0.061096
	4	-0.017767	-0.031250	-0.015625	-0.017761
	5	0.003434	0.000000	0.000000	0.003448
3	1	0.255319	0.250000	0.257813	0.255310
	2	-0.093791	-0.093750	-0.093750	-0.093781
	3	0.042921	0.031250	0.039063	0.042908
	4	-0.014537	0.000000	-0.015625	-0.014526
	5	0.003136	0.000000	0.000000	0.003143

The frequency responses for filters with unquantized and quantized (6, 8, 10, 12 and 16 bits) filter coefficients with 4 subbands are shown in Figure 5-11. With the small number of bits, the stopband attenuations are increased. Visually, the response for the 16 bit quantized filter was the same as the unquantized filter.

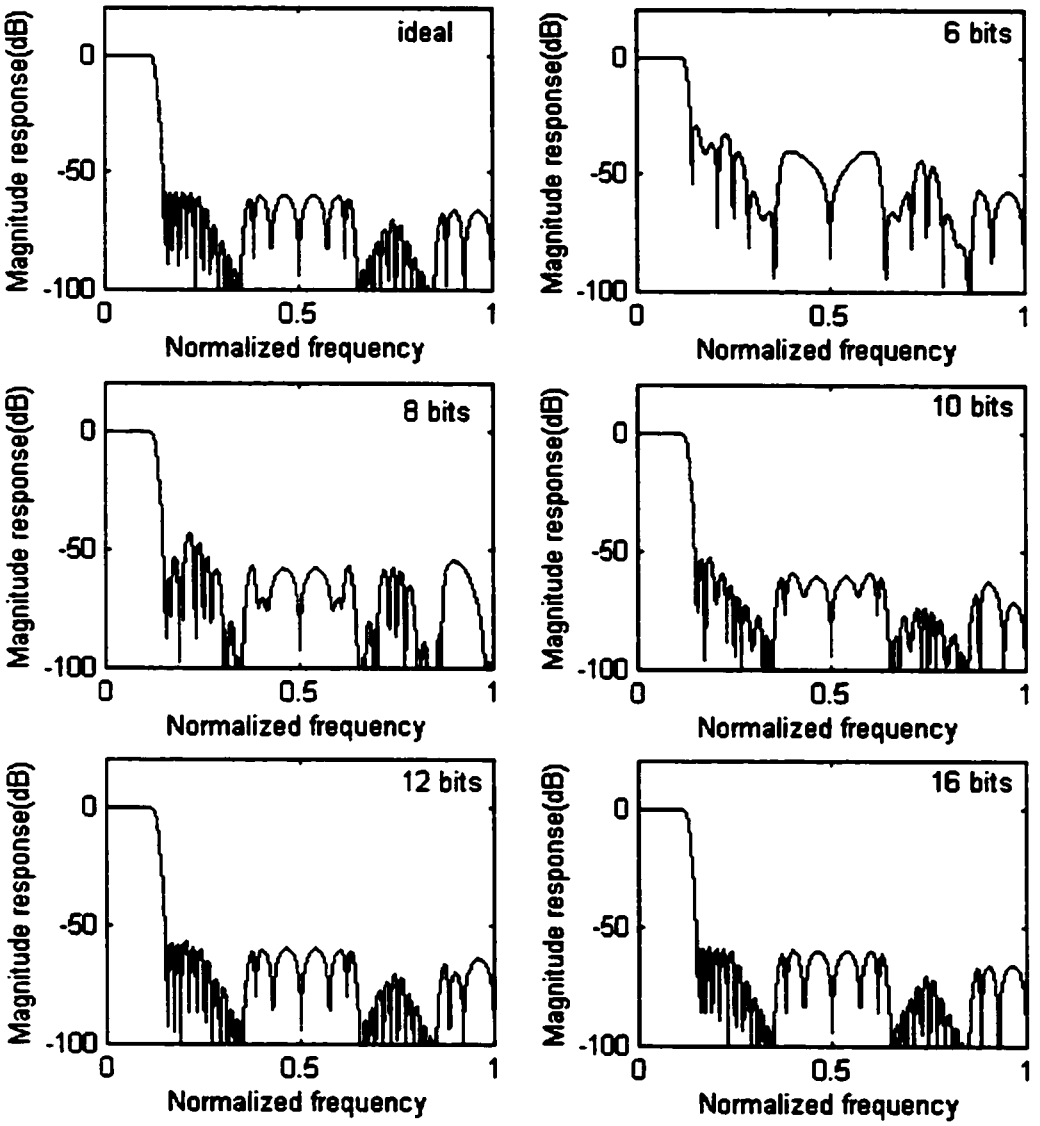

Figure 5-11. Practical effects of coefficient quantization on the frequency response, with M = 4 and the order of coefficients are 2, 5, 6 for allpass filters A, B and H.

The simulations with sound signal #3 were performed with unquantized and quantized filter coefficients, and the filters were implemented in direct form. The simulation results are shown in Figure 5-12 and Figure 5-13 where M equals 4 and 32, respectively. When M equals 4, the bit number increased and the square errors decreased. When M equals 32, 6 bits quantization produced overflow. 16 bit quantization produces the same result as the unquantized filter where M equals 4 or

32. Therefore, in order to satisfy the frequency response, 16 bits or more are required to implement a filter bank.

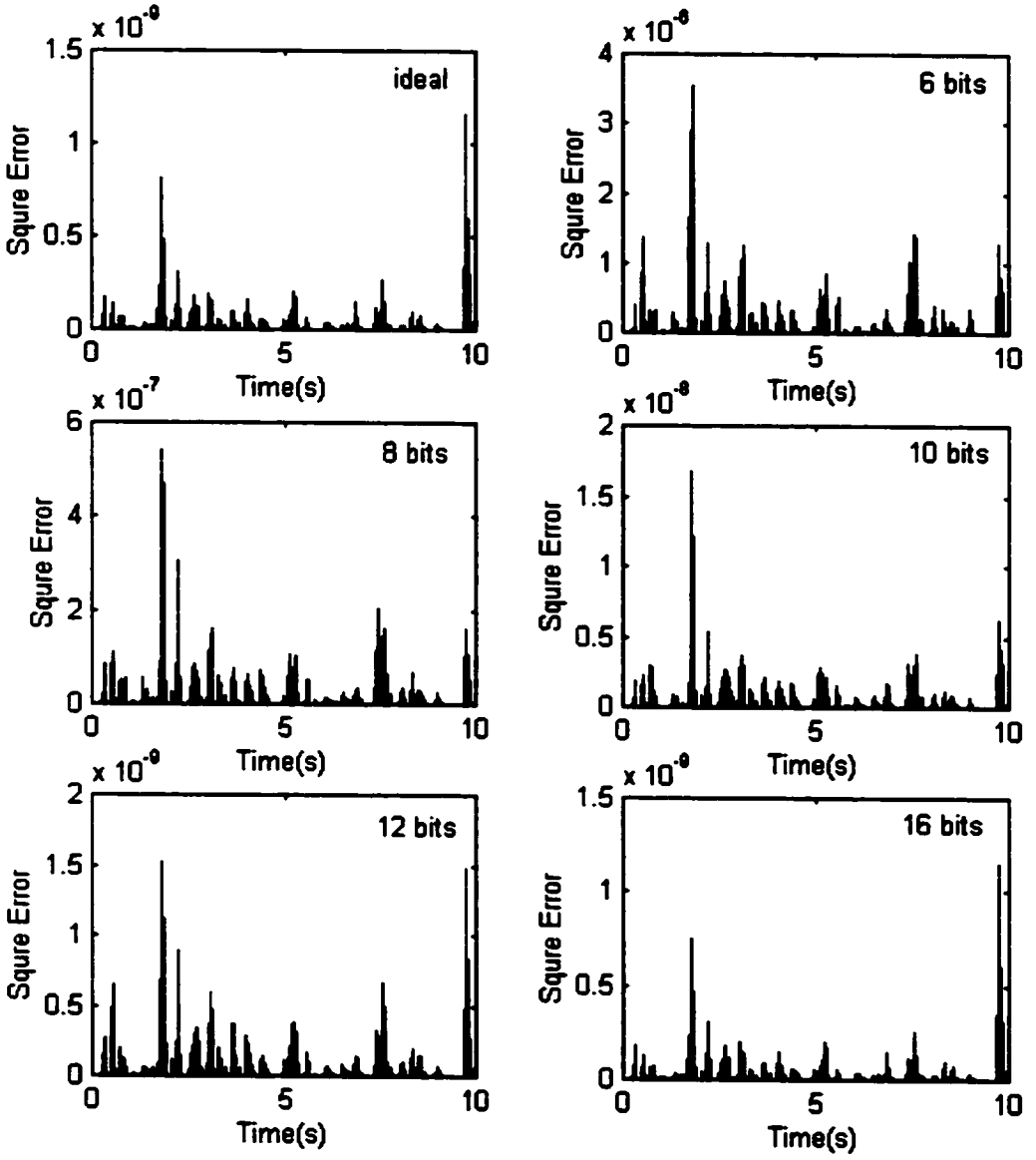

Figure 5-12. The practical effect of coefficient quantization on the simulation results of square error with M = 4 and the order of coefficients are 2, 5, 6 for allpass filters A, B and H.

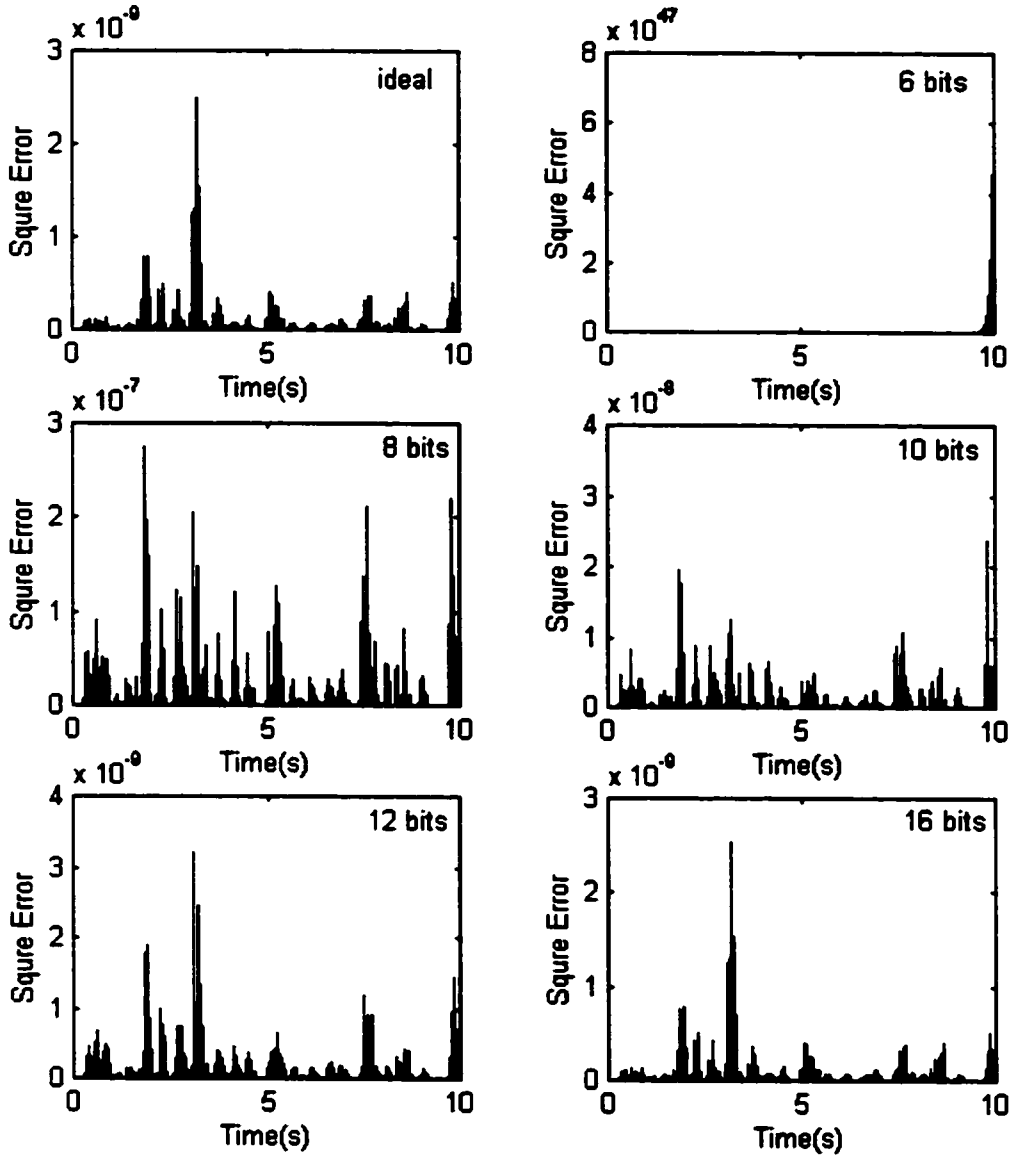

Figure 5-13. The practical effect of coefficient quantization on the simulation results of square error with M = 32 and the order of coefficients are 2, 5, 6 for allpass filters A, B and H.

Chapter 6: Conclusions and Future Works

6.1 Conclusions

A modulated filter bank based on allpass filters has been designed using Leest's [17] design. The filter banks have been implemented and simulated with sound/speech signals in Matlab code. The properties of the filter bank have been discussed. Since the delay of the filter bank using Leest's[17] design is too long to be used in real time applications such as telephone communications, the filter bank has been optimized to reduce that delay by increasing stopband attenuation and transition bandwidth. The relationships between the stopband attenuation and the order of coefficients have been studied. If the stopband attenuation is no more than -60 dB, the order of coefficients can be reduced to 2 for allpass filter A and 6 for allpass filter H. In order to reduce the order of coefficients of allpass filters in a synthesis bank, the phase and aliasing errors have been calculated and the simulations have been performed with sound signal #3 to obtain good reconstructed signals. The order of coefficients B has to be odd numbers; otherwise, the amplitude and square errors will produce large error in an simulation. The order of coefficients B can be reduced to 5. The delay of the filter bank after optimization is 601 with order of coefficients 2, 5 and 6 for allpass filters A, B and H, respectively. This delay is still too large to use in real time applications.

In order to further reduce the delay, the transition bandwidth needs to be increased; however, this will bring more aliasing errors. The relationship between delay and transition bandwidth have been studied. The delay still cannot be reduced to under 200. The only way to reduce the delay to under 200 is to reduce the number of subbands. With a delay of 200 samples, the subband numbers are 16, 8 or 4.

If the application is off-line, a near brick-wall filter can be designed with this technique, and the minimum transition bandwidth that can be reached is $0.035\pi/M$.

In order to avoid a complex output after a signal passes through an analysis filter bank, the DCT and Hartley modulation techniques have been investigated. Real output signals can be obtained from real input signals with both DCT and Hartley modulations, but the filter bank has to be kept at 2M bands for both modulations. With DCT and Hartley modulations, the tree structured filter banks

have been studied. The tree structured filter bank can produce a non-uniform band output, so it can be applied to subband coding based on psychoacoustic model. The delay can also be reduced to a certain degree.

The quantization error from quantizing coefficients has been investigated. At least 16 bits are required in order not to lose the quality of the filter bank.

6.2 Future Works

The filter bank designed using Leest's[17] method produces too long delay. Even after optimizing the order of coefficient, transition bandwidths, and filter bank structure, the delay is still too long. The only way to meet the delay requirement is to reduce the number of bands; however, this may cause some problems with sound compression. Further investigation needs to be done with respect to sound compression. Specifically, a filter bank needs to be designed that applies to real time sound compression.

References

1. D.Y. Pan, "Digital audio compression", *Digital Tech. J.*, pp. 1 – 14, 1993.

2. T. Q. Nguyen, "Near-Perfect-Reconstruction Pseudo-QMF Banks", *IEEE Trans. Signal Processing*, pp. 65 – 75, Jan. 1994.

3. W. P. Zhu, M. Q. Ahmad and M. N. S. Swamy, " An efficient approach for the design of nearly perfect-reconstruction QMF bands", *IEEE Trans. Circuits Syst II*, pp. 1161 – 1165, August, 1998.

4. A. Antoniou, "Accelerated procedure for the design of equiripple nonrecursive digital filters", *IEE Proc.*, pp. 1 – 10, Feb. 1982.

5. P. C. Millar, "Recursive Quadature Mirror Filters - Criteria Specification and Design Method", *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 413 – 420, April, 1985.

6. T. E. Tuncer and T. Q. Nguyen, " Interpolated IIR Mth-Band Filter Design with Allpass Subfilters", *IEEE Trans. Signal Processing*, pp. 1986 – 1990 August, 1986.

7. M. Renfors and T. Saramäki, " Recursive Nth-band digital filters-Part 1: Design and properties", *IEEE Trans. Circuits Syst.*, pp. 25 – 39. Jan. 1987.

8. M. Renfors and T. Saramäki, "Recursive Nth-band digital filters-Part II: Design of Multistage Decimators and Interpolators", *IEEE Trans. Circuits Syst.*, pp. 40 – 51 Jan. 1987.

9. T. Q. Nguyen, T. I. Laakso and R. D. Koilpillai, " Eigenfilter approach for the design of allpass filters approximating a given phase response", *IEEE Trans. Signal Processing*, pp. 2257 – 2263, Sept. 1994.

10. X. Zhang and H. Iwakura, "Design of IIR digital filters based on eigenvalue problem", *IEEE Trans. Signal Processing.*, pp. 1325 – 1333, June 1996.

11. X. Zhang and H. Iwakura, "Equiripple design of QMF banks using digital allpass networks", *IEICE Trans. Fundamentals*, pp. 1010 – 1015, August, 1995.

12. X. Zhang and H. Iwakura, "Design of digital allpass networks based on the eigenvalue problem", *Electronics and Communications in Japan*, pp. 99 – 109, 1994.

13. C. D. Creusere and S. K. Mitra, "Image coding using wavelets based on perfect reconstruction IIR filter bank", *IEEE Trans. Circuits Syst. Vidio Tech.*, pp. 447 – 458, Oct. 1996.

14. P. P. Vaidyanathan, P.A. Regalia and S.K. Mitra, "Design of doubly-complementary IIR digital filters using a single complex allpass filter, with multirate applications", *IEEE Trans. Circuits Syst.*, pp. 378 – 389, April, 1987.

15. F. Argenti and E. Del Re, "Rational sampling filter banks based on IIR filters", *IEEE Trans. Signal Processing*, pp. 3403 – 3408, Dec. 1998.

16. T. Q. Nguyen, T. I. Laakso and T. E. Tuncer, "On perfect-reconstruction allpass-based cosine modulated IIR filter bands", Proc. IEEE ISCAS., pp. 33 – 36, 1994.

17. A. J. van Leest, A. C. den Brinker and J. H. F. Ritzerfeld, " A modulated filter bank based on allpass filters", *Proc. ProRISC/IEEE Benelux Workshop on Circuits, Systems, and Signal Processing, Mierlo, The Netherlands*, pp. 221-226, Nov. 1996.

18. G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesly MA U.S.A. 1996.

19. E.C. Ifeachor and B.W. Jervis, *Digital Signal Processing – A Practical Approach*, Addison-Wesley, Workingham, England 1993.

20. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Inc., Englewood Cliffs, N. J. U.S.A., 1993.

21. P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks and applications: a tutorial", 1989.

22. M. G. Bellanger, G. Bonnerot and M. Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks", *IEEE Trans. Acoust., Speech, Signal Process.*, pp. 109 – 114, Apr. 1976.

23. H. G. Martinez and T.W. Parks, "A class of ininite-duration impulse response digital filters for sampling rate reduction", *IEEE Trans. Acoust., Speech, Signal Process.*, pp. 154 – 162, Apr. 1979.

24. R. Ishii and E. Tsutsui, "Some methods of designing polyphase networks by using all pass networks", in Proc. IEEE Int. Symp. Circuits Syst. (Houston, TX), pp. 183-186, Apr. 1989.

25. L. Taxen, "Polyphase filter banks using wave digital filters", *IEEE Trans. Acoust., Speech, Signal Process.*, pp. 423-428, June 1981.

26. R. A. Valenzuela and A. G. Constantinides, "Digital signal processing shemes for efficient interpolation and decimation", Proc. Inst. Elec. Eng. G, Electron. Circuits and Syst., pp. 225 – 235, Dec. 1983.

27. R. Ansari and B. Liu, "Efficient sampling rate alteration using recursive (IIR) digital fitlers", *IEEE Trans. Acoust., Speech, Signal Process*, pp. 1366 – 1373, Dec. 1983.

Appendix

A: Matlab code for implementation of 2M-channel modulated filter bank

The Matlab code of analytical bank without modulation is shown as following.

```
function YK=analbank(A,H,x);
% this function to simulation the analysis filter bank
% March 1999   by Min Chen
```

```
% YK is output signal without modulation

[M,N]=size(A); N=N-1;


%calculate order of H
NH=length(H(1,:))-1;
[L, R]=size(x);
 %demultiplexing
 Y=[];

for k=1:M,
    yz=delay(x,k-1);
    Y=[Y;yz];
end;

 % down sampling
YD=[];
for k=1:M,
   ydd=down(Y(k,:), M);
   YD=[YD;ydd];
end;

[n,d]=allpassm(H(2,:),2);

Y1=[];Y2=[]; YY1=[];
for k=1:M,
   f1=delay(YD(k,:),2*NH);
   x0=delay(YD(k,:),1);

   f2=filter(n,d,x0);

   y1=(f1+f2)/2;
   %y2: na Hi(-Z^M)
   y2=(f1-f2)/2; hold off

   YY1=[YY1;y1;y2];
   %y1 after A(Z)
   [na,da]=allpassm(A(k,:),1);

   res=filter(na,da,y1);
   Y1=[Y1; res];

   %y2 after A(-Z)

   [nan,dan]=allpassm(A(k,:),1,1);
   res=filter(nan,dan,y2);
   Y1=[Y1; res];
 end;
 YK=[];
 for k=1:M,
   ya=(Y1(2*k-1,:)+Y1(2*k,:))/2;
   YK=[YK;ya];
   ykn=(Y1(2*k-1,:)-Y1(2*k,:))/2;
```

```
   YK=[YK;ykn];
 end;
```

The synthesis bank without demodulation is as following:

```
function x1=synthbank(YII,B);

% simulation synthesis bank with inversted fft
% YI is from analysis bank
% B is coefficients of B from DESBANK
% Author: Min Chen
% Date: March 1999

[M,NC]=size(C); NC=NC-1;
[L, R]=size(YII);


%inversted K


YIK=[];
for k=1:M,
   ya=YII(2*k-1,:)+YII(2*k,:);

   ykn=YII(2*k-1,:)-YII(2*k,:);
   YIK=[YIK;ya;ykn];

end;
YIK=real(YIK);

YO=[];
% Y1 through C(Z)
for k=1:M,
   [n,d]=allpassm(C(k,:),1);
   Y2(2*k-1,:)=filter(n,d,YIK(2*k-1,:));

% Y2 through C(-Z)
   [nn,dn]=allpassm(C(k,:),1,1);

   Y2(2*k,:)=filter(nn,dn,YIK(2*k,:));
   YO(k,:)=Y2(2*k-1,:)+Y2(2*k,:);
end;


%up sampling
YU=[];
for k=1:M,
   YU(k,:)=up(YO(k,:),M);
end;

% delay

x1=zeros([1 R*M]);
```

```
for k=M:-1:1,
    x1=x1+delay(YU(M-k+1,:),k-1);
end;
```

B: The matlab code used in implementation the tree structured filter bank

B-1 Even Split Tree Structured Filter Bank

```
%DCT modulation constants
Ai=zeros(2*M, 2*M);
for n =1:2*M,
   for k=1:2*M;
      Ai(n,k)=sqrt(2)*cos(pi*(n-1)*(k-1)/(M) + pi/4);
   end;
end;

Y1 = analbank1(A,H,x);
YI=Ai*Y1;  % perform modulation


% second layer
Y11=[];
for k=1:2*M,
   Y2=analbank1(A,H,YI(k,:));
   Y2A=Ai*Y2;
   Y2B=Ai*Y2A/(2*M); %iDCT modulation
   x2=synthbank1(Y2B,C);
   Y11(k,:)=x2;
end;

Y1A=Ai'*Y11/(2*M);
x1 = synthbank1(Y1A,C);
% x1 is output signal and x is input signal
```

B-2 Uneven Split Tree Structured Filter Bank

```
Ai=zeros(2*M, 2*M);
for n =1:2*M,
   for k=1:2*M;
      Ai(n,k)=sqrt(2)*cos(pi*(n-1)*(k-1)/(M) + pi/4);
   end;
end;
Ai1=zeros(2*M1, 2*M1);
for n =1:2*M1,
   for k=1:2*M1;
      Ai1(n,k)=sqrt(2)*cos(pi*(n-1)*(k-1)/(M1) + pi/4);
   end;
end;
Y1 = analbank1(A,H,x);
YI=Ai*Y1;
```

```
% second layer
Y11=[];
for k=1:2*M,
   if k == 1|k==2|k==8
      Y2=analbank1(A,H,YI(k,:));
      Y2A = Ai*Y2;
      Y2B=Ai*Y2A/(2*M);
       x2=synthbank1(Y2B,C);
   end
   if (k > 2 & k < 8)
      if (k == 5)
         x2 = delay(YI(k,:), dly);
      else
         Y2=analbank1(A1,H1,YI(k,:));
        Y2A=Ai1*Y2;
        Y2B=Ai1*Y2A/(2*M1);
         x21=synthbank1(Y2B,C1);
         x2 = delay(x21, (dly - dly1));

      end
   end

   xd2=delay(YI(k,:),dly);
   dif2=abs(xd2)-abs(x2);
   subplot(3,3,k);
   plot(dif2.*dif2); pause;

   Y11(k,:)=x2;

end;
Y1A=Ai'*Y11/(2*M);
x1 = synthbank1(Y1A,C);
% x1 is output signal and x is input signal
```

C: The coefficients of A, B and D with order of 4, 7 and 11

Table C-1 The coefficient of A(z) with the M = 32

k	1	2	3	4
2	2.701e-002	-8.208e-003	2.277e-003	-3.820e-004
3	5.424e-002	-1.600e-002	4.377e-003	-7.276e-004
4	8.169e-002	-2.337e-002	6.304e-003	-1.038e-003
5	1.094e-001	-3.030e-002	8.059e-003	-1.315e-003
6	1.373e-001	-3.679e-002	9.644e-003	-1.558e-003
7	1.655e-001	-4.283e-002	1.106e-002	-1.770e-003
8	1.939e-001	-4.841e-002	1.232e-002	-1.952e-003
9	2.225e-001	-5.351e-002	1.341e-002	-2.104e-003
10	2.514e-001	-5.813e-002	1.434e-002	-2.228e-003
11	2.806e-001	-6.226e-002	1.511e-002	-2.324e-003
12	3.100e-001	-6.588e-002	1.573e-002	-2.395e-003
13	3.397e-001	-6.898e-002	1.620e-002	-2.441e-003

14	3.697e-001	-7.155e-002	1.652e-002	-2.463e-003
15	3.999e-001	-7.358e-002	1.670e-002	-2.463e-003
16	4.304e-001	-7.505e-002	1.673e-002	-2.442e-003
17	4.613e-001	-7.596e-002	1.663e-002	-2.401e-003
18	4.924e-001	-7.628e-002	1.639e-002	-2.341e-003
19	5.238e-001	-7.600e-002	1.603e-002	-2.263e-003
20	5.556e-001	-7.512e-002	1.553e-002	-2.169e-003
21	5.876e-001	-7.360e-002	1.492e-002	-2.059e-003
22	6.200e-001	-7.144e-002	1.419e-002	-1.936e-003
23	6.528e-001	-6.862e-002	1.335e-002	-1.799e-003
24	6.858e-001	-6.512e-002	1.240e-002	-1.651e-003
25	-6.093e-002	-6.093e-002	1.135e-002	-1.493e-003
26	7.530e-001	-5.602e-002	1.020e-002	-1.326e-003
27	7.871e-001	-5.039e-002	8.962e-003	-1.150e-003
28	8.216e-001	-4.399e-002	7.640e-003	-9.682e-004
29	8.565e-001	-3.683e-002	6.240e-003	-7.806e-004
30	8.918e-001	-2.887e-002	4.768e-003	-5.888e-004
31	9.274e-001	-2.009e-002	3.233e-003	-3.939e-004
32	9.635e-001	-1.048e-002	1.641e-003	-1.973e-004

Table C-2 The coefficient of B(z) with M = 32

K	1	2	3	4	5	6	7
2	9.728e-001	-1.791e-002	6.073e-003	-2.064e-003	4.719e-004	-2.833e-005	3.390e-006
3	9.450e-001	-3.459e-002	1.210e-002	-4.270e-003	1.062e-003	-1.066e-004	1.310e-005
4	9.165e-001	-4.992e-002	1.796e-002	-6.539e-003	1.736e-003	-2.248e-004	2.831e-005
5	8.873e-001	-6.383e-002	2.354e-002	-8.806e-003	2.461e-003	-3.733e-004	4.808e-005
6	8.575e-001	-7.629e-002	2.876e-002	-1.101e-002	3.210e-003	-5.432e-004	7.142e-005
7	8.269e-001	-8.726e-002	3.356e-002	-1.311e-002	3.957e-003	-7.265e-004	9.735e-005
8	7.959e-001	-9.677e-002	3.789e-002	-1.506e-002	4.684e-003	-9.158e-004	1.249e-004
9	7.643e-001	-1.048e-001	4.171e-002	-1.684e-002	5.372e-003	-1.105e-003	1.532e-004
10	7.323e-001	-1.115e-001	4.502e-002	-1.842e-002	6.009e-003	-1.289e-003	1.815e-004
11	6.999e-001	-1.168e-001	4.780e-002	-1.980e-002	6.584e-003	-1.463e-003	2.090e-004
12	6.673e-001	-1.209e-001	5.006e-002	-2.096e-002	7.089e-003	-1.623e-003	2.351e-004
13	6.345e-001	-1.237e-001	5.181e-002	-2.189e-002	7.519e-003	-1.768e-003	2.593e-004
14	6.015e-001	-1.254e-001	5.305e-002	-2.261e-002	7.870e-003	-1.893e-003	2.811e-004
15	5.686e-001	-1.259e-001	5.381e-002	-2.311e-002	8.141e-003	-1.999e-003	3.002e-004
16	5.356e-001	-1.254e-001	5.410e-002	-2.339e-002	8.330e-003	-2.083e-003	3.163e-004
17	5.026e-001	-1.240e-001	5.394e-002	-2.346e-002	8.438e-003	-2.145e-003	3.291e-004
18	4.697e-001	-1.216e-001	5.335e-002	-2.334e-002	8.465e-003	-2.183e-003	3.384e-004
19	4.370e-001	-1.183e-001	5.234e-002	-2.301e-002	8.413e-003	-2.199e-003	3.441e-004
20	4.044e-001	-1.143e-001	5.093e-002	-2.249e-002	8.284e-003	-2.192e-003	3.461e-004
21	3.720e-001	-1.094e-001	4.913e-002	-2.179e-002	8.078e-003	-2.161e-003	3.442e-004
22	3.398e-001	-1.038e-001	4.695e-002	-2.091e-002	7.797e-003	-2.108e-003	3.383e-004
23	3.078e-001	-9.743e-002	4.441e-002	-1.985e-002	7.444e-003	-2.031e-003	3.286e-004
24	2.759e-001	-9.042e-002	4.152e-002	-1.862e-002	7.018e-003	-1.931e-003	3.148e-004
25	2.444e-001	-8.275e-002	3.828e-002	-1.723e-002	6.522e-003	-1.809e-003	2.969e-004
26	2.130e-001	-7.445e-002	3.469e-002	-1.566e-002	5.955e-003	-1.664e-003	2.750e-004
27	1.819e-001	-6.553e-002	3.076e-002	-1.393e-002	5.319e-003	-1.496e-003	2.488e-004
28	1.510e-001	-5.602e-002	2.649e-002	-1.204e-002	4.612e-003	-1.306e-003	2.185e-004
29	1.204e-001	-4.593e-002	2.188e-002	-9.977e-003	3.836e-003	-1.092e-003	1.838e-004
30	8.995e-002	-3.528e-002	1.693e-002	-7.745e-003	2.988e-003	-8.553e-004	1.448e-004
31	5.975e-002	-2.406e-002	1.164e-002	-5.341e-003	2.067e-003	-5.948e-004	1.012e-004
32	2.977e-002	-1.230e-002	5.997e-003	-2.761e-003	1.072e-003	-3.099e-004	5.300e-005

Table C-3 The coefficient of D(z) in H with M = 32

order	coefficient
1	4.939e-001
2	-1.161e-001
3	5.185e-002
4	-2.745e-002
5	1.537e-002
6	-8.654e-003
7	4.749e-003
8	-2.471e-003
9	1.181e-003
10	-4.921e-004
11	1.752e-004