AFRL-IF-RS-TR-2006-277
In-House Final Technical Report
September 2006

A REVIEW OF POLYPHASE FILTER BANKS AND THEIR APPLICATION

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2006-277 has been reviewed and is approved for publication.

APPROVED: /s/

                                GERALD C. NETHERCOTT
                                Chief, Multi-Sensor Exploitation Branch

FOR THE DIRECTOR: /s/

                                JOSEPH CAMERA, Chief
                                Information & Intelligence Exploitation Division
                                Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
SEP 2006	Final	Jun 05 – May 06

4. TITLE AND SUBTITLE

A REVIEW OF POLYPHASE FILTER BANKS AND THEIR APPLICATION

5a. CONTRACT NUMBER
In-House

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Daniel Zhou

5d. PROJECT NUMBER
459E

5e. TASK NUMBER
PR

5f. WORK UNIT NUMBER
OJ

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/IFEC
525 Brooks Rd
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFEC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2006-277

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 06-633

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this report we review and compare numbers of different filter bank approaches which include discrete Fourier transform (DFT) filter bank, modified discrete Fourier transform (MDFT) filter bank, and cosine modulated filter bank. Our simulation results show that there is no one best overall method that gives the best performance. One method can perform better than others under one scenario, but not as well as others under different scenarios. Given a particular situation or specific application, the best method in terms of reconstruction and dealing with channel distortion may vary.

15. SUBJECT TERMS
Polyphase Filter, Filter Bank

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UL	36	Daniel Zhou
U	U	U			19b. TELEPONE NUMBER *(Include area code)*

Table of Contents

List of Figures

List of Tables

iii

Acknowledgements

I would like to thank my mentor, Dr. Andrew Noga, without his guidance I could not have written this report. His guidance allowed the understanding and proficiency of Polyphase filter banks. His ideas and suggestions were very important for implementing the polyphase filter banks. I am grateful to him for his patience and permanent support during this research period. Thanks is also given to AFRL/IFEC Ms Emily Krzysiak for management and technical oversight.

1. Introduction

Multirate digital signal processing (DSP) has attracted much attention over the past two decades due to the applications in subband coding of speech, audio and video, multiple carrier data transmission, etc. A key characteristic of multirate algorithms is their high computational efficiency. A multirate system can increase or decrease the sampling rate of individual signals before or while processing them. These signals then with different sampling rate can be simultaneously processed in various parts of the multirate system.

Digital filter banks are the most important applications of multirate DSP. A great amount of different filter bank approaches have been developed over last fifteen years. Among those filter banks, Cosine Modulated filter banks [1]-[3] are very popular because they are easy to implement and can provide perfect reconstruction (PR). The Discrete Fourier Transform (DFT) polyphase filter bank [4] is another popular filter bank that provides high computational efficiency, but suffers from the fact that it is not able to cancel alias components caused by subsampling the sub band signals. By introducing a certain modification to the DFT filter bank, we can overcome its disadvantage. The modified DFT (MDFT) filter banks [5]-[7] can also provide PR.

In this report, we will first study the structures of DFT, MDFT and Cosine-Modulated filter banks, and then demonstrate MATLAB simulation for these filter banks. In section 2, we will discuss some essential operations for multirate systems in order to understand filter banks. A brief review on these three filter banks and their MATLAB implementation results will be presented in Section 3 and Section 4 respectively, and finally the conclusions are given in Section 5.

1

2. Fundamentals of Multirate DSP

2.1   Sampling Rate Conversion

To understand the multirate systems, it is essential to understand how the sampling rate is changed. There are two basic sampling rate change operations: downsampling (decrease the sampling rate) and upsampling (increase the sampling rate).

Downsampling

Downsampling is the process of reducing the sampling rate. Downsampling a signal can be useful if the sampling rate was considerably greater than the bandwidth of the signal. It can reduce the calculation and/or memory required to implement a DSP system.

The operation of downsampling by a factor M is shown in Figure 2-1, where the downsampler takes an input signal $x[n]$ with high sampling rate F_{s_H} and produces a low sampling rate F_{s_L} output sequence $y[m]$ by keeping every M^{th} sample and discarding the rest. This downsampling process can be written as

$$y[m] = x[n]\big|_{n=mM} \quad \text{for } n = 0,1,2,\cdots N ; \quad m = 0,1,2,\cdots \frac{N}{M} \tag{2.1}$$

The relationship between sampling rates F_{s_H} and F_{s_L} can be described as below

$$F_{s\_L} = F_{s\_H} / M \tag{2.2}$$

where M is called the downsampling factor, and is simply the ratio of the input rate to the output rate.

$$x[n] \xrightarrow{\quad F_{s\_H} \quad} \boxed{\downarrow M} \xrightarrow{\quad y[m] \quad F_{s\_L}}$$

Figure 2-1   Downsampler

2

Upsampling

Upsampling increases the sampling rate of signal by inserting zero-valued samples between original samples. The main reason for upsampling is to increase the sampling rate at the output of one system so that another system operating at a higher sampling rate can process the signal.

Figure 2-2 illustrates the operation of upsampling a signal by a factor L, it takes an input signal $x[m]$ with a low sampling rate F_{s_L} and produces the output sequence $y[n]$ has high sampling rate F_{s_H} by inserted L-1 zeros between every sample of the input signal. This upsampling process can be written as

$$y[n] = x[m]\big|_{m=n/L} \quad \text{for } m \in \text{integer}, \quad y[n] = 0 \quad \textit{otherwise}$$

$$n = 0,1,2,\cdots N \ ; \quad m = 0,1,2,\cdots \frac{N}{L} \tag{2.3}$$

In this case the sampling rates F_{s_L} and F_{s_H} are related by an upsampling factor L, which is the ratio of the output rate to the input rate.

Figure 2-2   Upsampler

2.2   Decimation

Decimation is the process of filtering and downsampling a signal to decrease its effective sampling rate. If we downsample a signal by just throwing away the intermediate samples, we will get alisasing. To prevent this alisasing of the result from the downsampling, an anti-aliasing low pass filter $H(z)$ is employed before the downsampler as shown in figure 2-3.

3
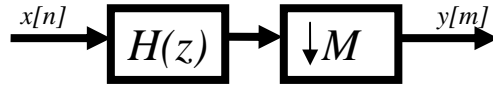
Figure 2-3   Decimator consisting of an anti-aliasing filter *H(z)* and a downsampler

2.3   Interpolation

Interpolation is the process of upsampling and filtering a signal to increase its effective sampling rate as shown in figure 2-4. To prevent extra spectral copies that might result from upsampling, an anti-image low pass filter $G(z)$ is employed after the upsampler.

Figure 2-4   Interpolator consisting of an upsampler and anti-image filter *G(z)*

2.4   Noble Identities

The noble identities describe the property of reverse ordering the filter and downsampling/upsampling. Figure 2-5 (a) and (b) show a pair of equivalent block diagrams, which describe the Noble identities for decimation and interpolation respectively. Note the FIR filter $H(z)$ is the M downsampled impulse response of $H(z^M)$ and $H(z^L)$ is the upsampled impulse response of $H(z)$.

(a)

(b)
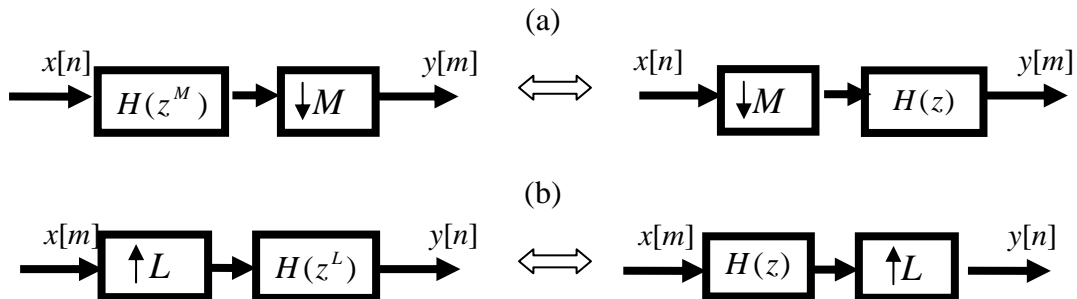
Figure 2-5   The Noble identities for (a) decimation and (b) interpolation

4

2.5   Polyphase Decimation

The standard decimation method in Figure 2-3, is computationally inefficient because it throws away the majority of the computed filter outputs. By using the Noble identity we can rearrange the structure in Figure 2-3 so that filter outputs are not discarded. In order to apply the Noble identity for decimation, first we must decompose the filter *H(z)* into its polyphase components.

$$H(z) = \sum_{i=0}^{M-1} z^{-i} H_i(z^M) \qquad (2.4)$$

Now we can transform Figure 2-3 into the polyphase components structure as shown in Figure 2-6.

Figure 2-6   Polyphase components structure

Then we can apply the Noble identity for decimation to Figure 2-6 yielding the process shown in Figure 2-7. The ladder of delays and downsamplers on the left of Figure 2-7 accomplishes a form of serial-to-parallel conversion.

Figure 2-7   Polyphase structure for decimation

5

Figure 2-8 shows an equivalent structure of the polyphase decimation by using an input commutator to represent the splitting of input signal *x[n]* into the lower rate sub-sequences $x_0[m], x_1[m] \cdots x_{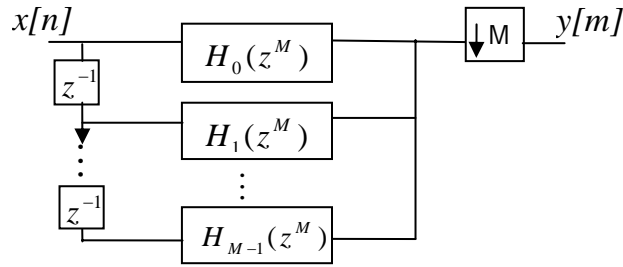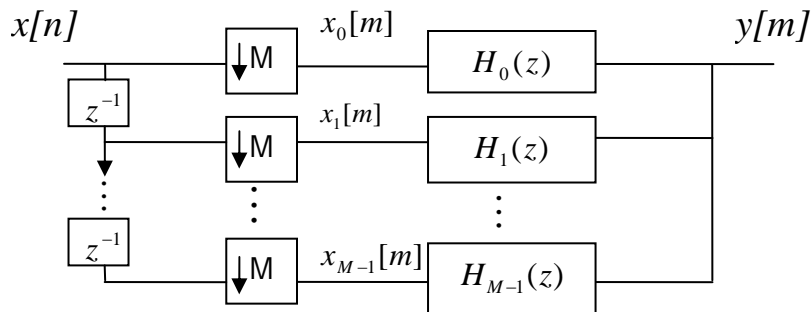M-1}[m]$ [4]. These two structures have trivial differences in their practical implementation. For the configuration in Figure 2-7, a group of *M* input samples are sent to the *M* sub filters at times *t=mM* by *M* downsamplers. For example at time *t=0 (m=0)*, a group of samples { $x_0[0] = x[0], x_1[0] = x[-1], \cdots x_{M-1}[0] = x[-M+1]$ } are sent to filters { $H_0(z), H_1(z), \cdots, H_{M-1}(z)$ }. At time *t=M (m=1)*, a group of samples { $x_0[1] = x[M], x_1[1] = x[M-1], \cdots x_{M-1}[1] = x[1]$ } are sent to the same filters, etc. On the other hand, the representation with an input commutator (Figure 2-8) gives the impression that the input samples pass to *M* sub filters one after the others. To get the output value *y[m]*, the commutator has to rotate counterclockwise to give input samples $x[t-M+1], \cdots, x[t-1], x[t]$ *(t=mM)* to the filters $H_{M-1}(z), \cdots, H_1(z), H_0(z)$. However these two representations lead to same result.
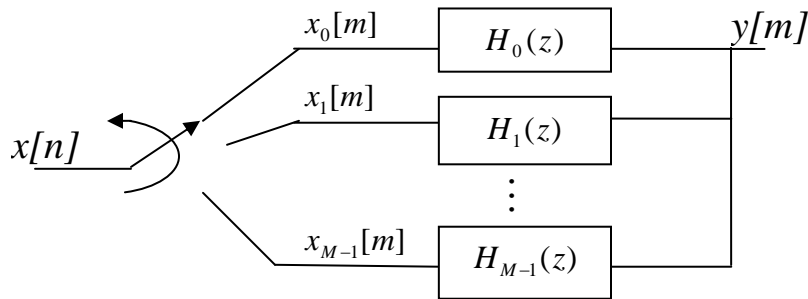

Figure 2-8   Polyphase decimator with an input commutator rotating counterclockwise

2.6   Polyphase Interpolation

The standard interpolation procedure illustrated in Figure 2-4 is also computationally inefficient since the lowpass filter operates on a sequence that is mostly composed of

zeros. We can use the same procedures as polyphase decimation to transform Figure 2-4 into an efficient implementation structure as shown in Figure 2-9 by using the Noble identity for interpolation.
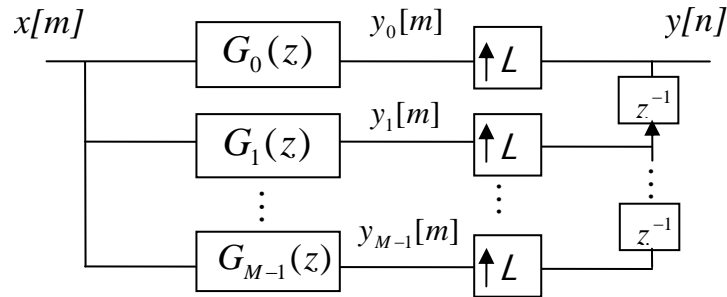
Figure 2-9   Polyphase structure for interpolation

In Figure 2-9, to produce the output signal $y[n]$, the sub-sequences $\{ y_0[m], y_1[m] \cdots y_{M-1}[m] \}$ must transform to polyphase components of the output signal by upsampling by a factor L and adding a delay z^{λ} $(0 \leq \lambda \leq M-1)$ at the higher sampling rate. This process can also be represented by using a commutator to combine subsequence into the output signal as illustrated in Figure 2-10. The commutator combines the output signal by taking sample by sample from sub-sequence $\{ y_0[m], y_1[m] \cdots y_{M-1}[m] \}$. So the output signal $y[n]$ has sample sequence as $\{ y_0[0], \cdots, y_{M-1}[0], y_0[1], \cdots, y_{M-1}[1], \cdots \}$. Again these two structures will produce same result.
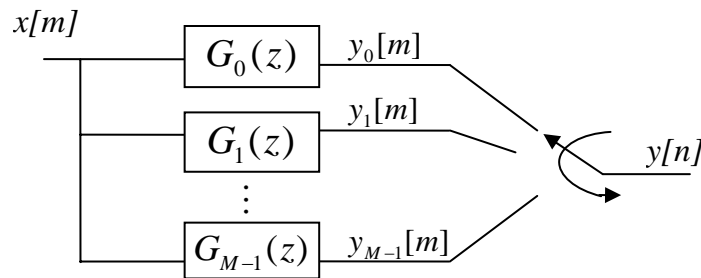
Figure 2-10   Polyphase interpolator with a counterclockwise output commutator

7

For many applications in signal processing, it is useful to separate a signal into different frequency ranges called sub-bands. The spectrum can be partitioned in the uniform manner as illustrated in Figure 3-1, where the sub-band width $\nabla k = \dfrac{2\pi}{M}$ is identical for each sub-band and the band centers are uniformly spaced at intervals of $\dfrac{2\pi}{M}$.
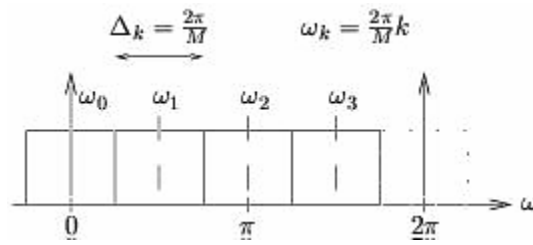
Figure 3-1   Uniform spaced spectrum *(Ref: Connexions, by Phil Schniter)*

The sub-bands can also have non-uniformly spacing. For our discussion, we will only focus on uniformly spaced sub-bands. The goal of separation into sub-band components is to make further processing more convenient. Some of the popular applications for sub-band decomposition are audio and video source coding with the purpose of efficient storage and/or transmission.

In typical applications, non-trivial signal processing takes place between a bank of analysis filters $\{H_0(z), H_1(z) \cdots H_{M-1}(z)\}$ and $\{G_0(z), G_1(z) \cdots G_{M-1}(z)\}$, a bank of synthesis filters as shown in Figure 3-2. For our discussion, we will focus on filter bank design rather than the sub-band processing that occurs between the filter banks.

8
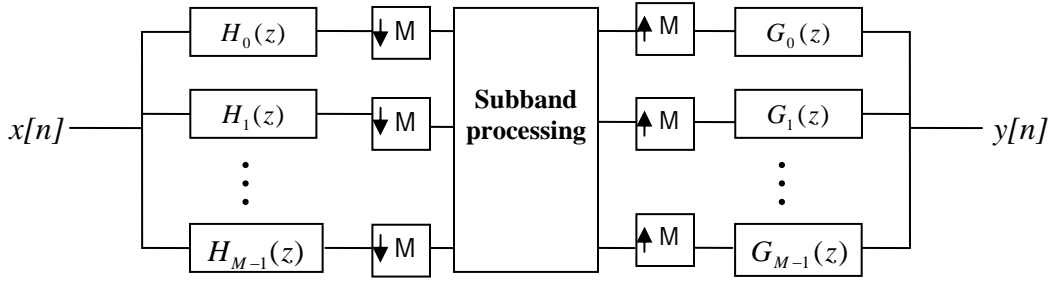
Figure 3-2   Standard filter bank structure

The main goals in filter bank design is to have good reconstruction (i.e., $y[n] \approx x[n-d]$ for some integer delay d) when the sub-band processing is lossless. This goal is motivated by the idea that the sub-band filtering should not limit the reconstruction performance when sub-band processing (e.g., the coding/decoding) is lossless or nearly lossless. For the following sections, we will discuss three types of polyphase filter bank design: DFT, Modified DFT (MDFT) and Cosine Modulated filter banks respectively.

3.1   DFT Filter bank

Among these three polyphase filter banks, the DFT filter bank is the easiest one to implement and it has the simplest structure, which is shown in Figure 3-3. The materials in section 2 are essential for us to understand the structure of the polyphase filter bank. For the standard filter bank shown in Figure 3-2, it performs filtering before downsampling. On the other hand, the DFT polyphase structure performs downsampling first and then filtering, which results in a huge computational saving. We know that filtering is very expensive in computation. With the structure of downsampling before filtering, we can save a huge amount of filtering operations. The properties that allow polyphase structures to perform the sampling and then filtering are known as the Noble identities, which we had discussed in section 2.

9
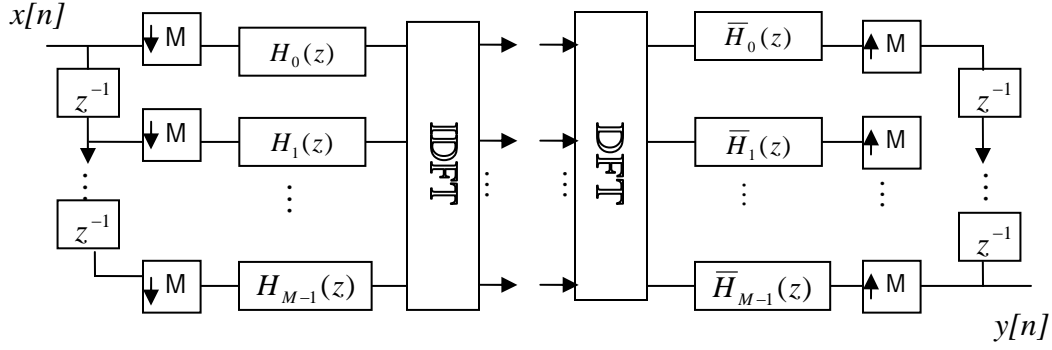
Figure 3-3   Structure of DFT filter bank

In Figure 3-3, the DFT filter bank consists of analysis and synthesis filter banks, and the sub-filters within these banks are generated from same prototype filter. For the analysis bank, we need to perform polyphase decimation. In contrast, we implement polyphase interpolation in the synthesis bank. Filters in the analysis bank $\{H_0(z), H_1(z) \cdots H_{M-1}(z)\}$ and filters in the synthesis bank $\{\overline{H}_0(z), \overline{H}_1(z) \cdots \overline{H}_{M-1}(z)\}$ are called the polyphase components of the prototype filter. The impulse responses of these filters are related as

$$\overline{h}_k(n) = h_k(N-1-n), \qquad 0 \le n \le N-1 \tag{3.1}$$

If the sub-band processing between IDFT and DFT operations is lossless, then the output signal *y[n]* should be identical to *x[n]* with some delay. But that is not the case, when we decompose the prototype filter into its polyphase components; it creates aliasing among these sub-filters, which prevents us from reconstructing the signal perfectly. DFT filter bank structure is very simple and it doesn't contain any aliasing cancellation structure; it suffers from the aliasing effect within the banks, but is useful for channelization.

10

3.2   Modified DFT Filter Bank

The disadvantage of DFT polyphase filter banks can be overcome by introducing a certain modification to the DFT filter bank. It is called the Modified DFT (MDFT) filter bank [5], where a structure inherent alias cancellation can be obtained, yielding nearly PR. MDFT filter banks are modified complex modulated, critically sub-sampled filter banks based on DFT filter banks.
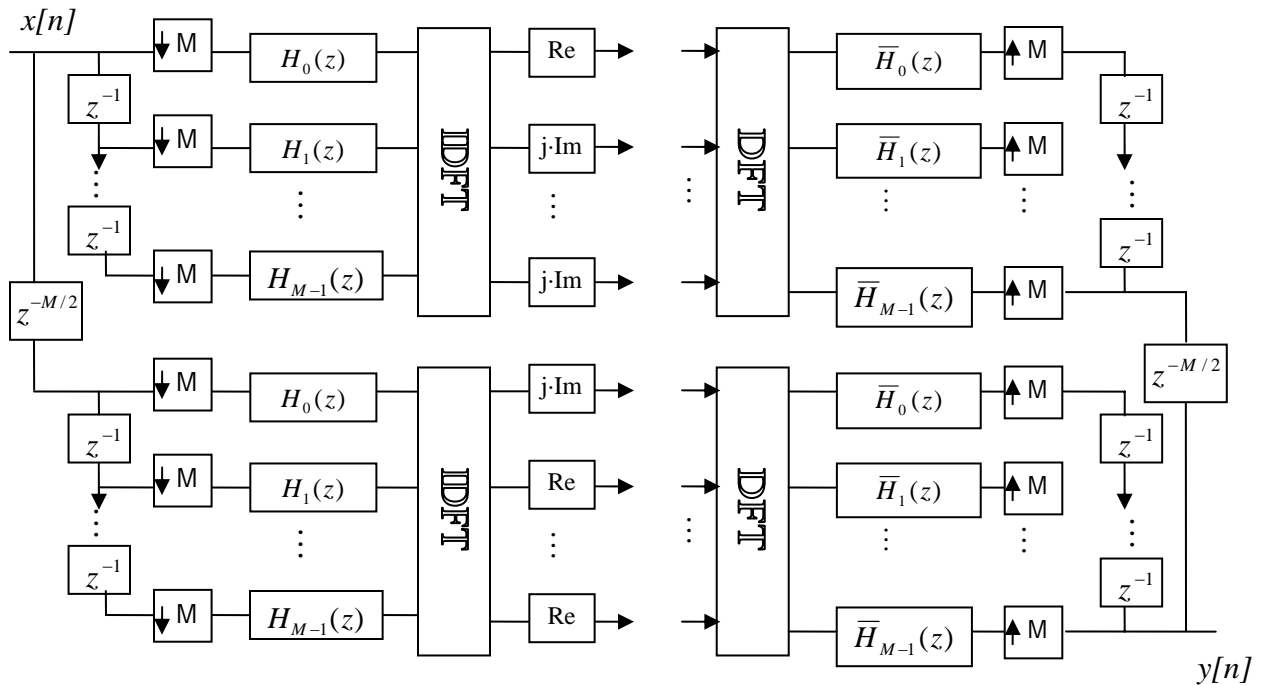
Figure 3-4   MDFT filter bank realized by two DFT polyphase filter banks

The structure of MDFT filter bank can be realized by means of two DFT polyphase filter banks, where one is delayed by *M/2* samples as shown in Figure 3-4. We also notice from the structure that it is taking alternately the real and imaginary part of the sub-band signals. For the upper DFT filter bank, the even sub-channels take the real parts, and the odd sub-channels take the imaginary parts of the sub-band signals. For the lower DFT

11

filter bank, it performs the opposite. Even sub-channels take imaginary parts, and odd sub-channels take the real parts. The implementation cost of this MDFT structure is twice the implementation cost of the DFT filter bank but can be further reduced [6].
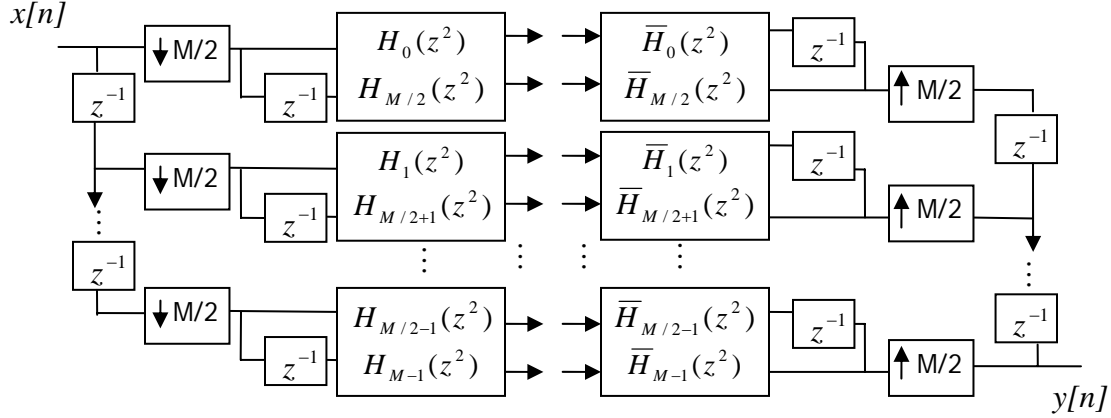
Figure 3-5   Computational efficient realization of the MDFT Filter Bank

A simpler and computational efficient realization of the MDFT filter bank is shown in Figure 3-5. (Detailed information can be found in [5]). In this structure, the input signal *x[n]* will be decomposed into *M/2* polyphase components and the prototype filter will have *M* polyphase components. Each of the *M/2* polyphase components of input signal will filter with a pair of the sub-filters alternately without and with a delay. The MDFT filter bank guarantees PR if each paired polyphase components of the prototype filter $H_k(z)$ and $H_{k+M/2}(z)$ satisfy following condition,

$$H_k(z)\overline{H}_k(z) + H_{k+M/2}(z)\overline{H}_{k+M/2}(z) = \frac{2}{M}z^\alpha \ \text{ for } \ k = 0, \cdots, M/2-1 \qquad (3.1)$$

This is equivalent to the PR condition for Cosine Modulated filter banks which will be introduced in the following section.

3.3 Cosine Modulated Filter bank

The theory and design of M-channel Cosine Modulated filter banks have been studied extensively in the past [1]-[3]. The Cosine Modulated filter banks emerged as an attractive choice for filter banks due to its simple implementation and the ability to provide PR. In this system, the impulse responses of analysis filters $h_k(n)$ and synthesis filters $f_k(n)$ are the Cosine Modulated versions of a single prototype filter $h(n)$ [2]. Therefore the design of the whole filter bank reduces to that design for the prototype filter. The filter bank has perfect reconstruction if the polyphase components of the prototype satisfy a pair-wise power complementary condition,

$$\tilde{H}_k(z)H_k(z) + \tilde{H}_{M+k}(z)H_{M+k}(z) = \frac{1}{2M} \tag{3.2}$$

The detail design of the prototype filter can be found in [1], where the optimization of the prototype filter coefficients is given.

Several efficient methods have been proposed to facilitate the design of prototype filter. In [8], Creusere and Mitra proposed a very efficient prototype design method without using nonlinear optimizations. Instead of a full search, it is limited to the class of filters obtained using the Parks-McClellan algorithm. As a result, the optimization can be reduced to that of a single parameter. In the Kaiser Window method of prototype filter design for Cosine Modulated filter banks [9], the design process is reduced to the optimization of the cutoff frequency in the Kaiser Window. Another design method in [10] is based on windowing, which varies the value of 6-dB cutoff frequency of the prototype filter so that final prototype filter has its 3-dB cutoff frequency located approximately at $\pi/2M$.

13
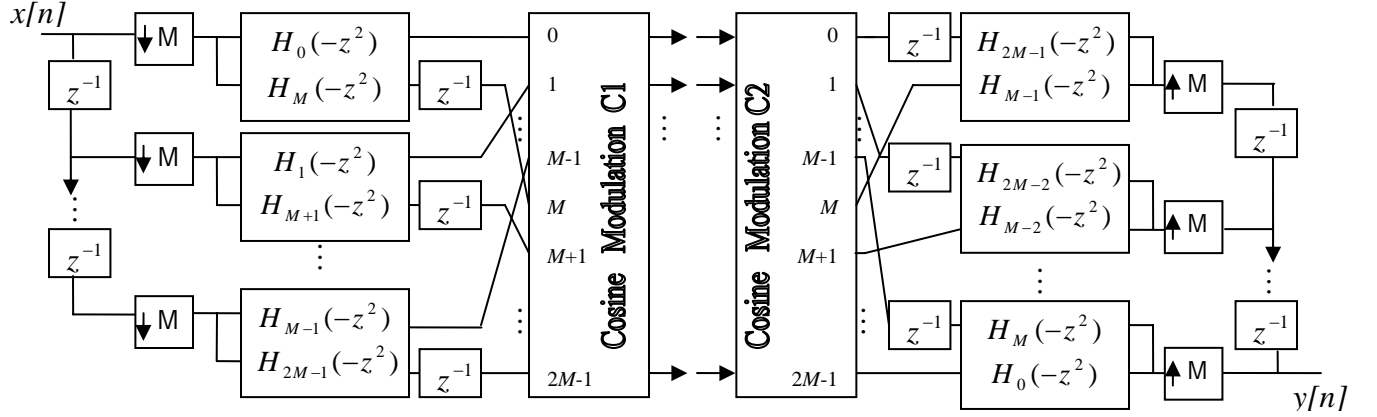
Figure 3-6   Structure of Cosine Modulated polyphase filter bank

The structure of the Cosine Modulated polyphase filter bank is shown in Figure 3-6 [11]. In this structure, the input signal *x[n]* will be decomposed into *M* polyphase components and the prototype filter will have 2*M* polyphase components. Each of the polyphase components of the input signal will be filtered with a pair of sub-filters that satisfy the pair-wise power complementary condition as shown in equation (3.2). The elements of the Cosine Modulation matrix C_1 and C_2 in Figure 3-6 can be calculated by use following equation,

$$[C_1]_{k,l} = 2\cos[\frac{\pi}{M}(k+0.5)(l-\frac{N}{2})+(-1)^k\frac{\pi}{4}],$$
$$[C_2]_{k,l} = 2\cos[\frac{\pi}{M}(k+0.5)(2M-1-l-\frac{N}{2})+(-1)^k\frac{\pi}{4}]$$
$$0 \le l \le 2M-1, \qquad 0 \le k \le M-1 \qquad (3.3)$$

4.   MATLAB Simulation

In section 3, we have reviewed the structures of DFT, MDFT and Cosine Modulated filter banks. In this section, we are going to show the MATLAB implementation results for these three filter banks. The structures of these filter banks used in our simulation are

based on the previous section, and we use various types of test signals (an audio signal, a chirp signal, and a step frequency signal) in our simulation. For demonstration purposes we are going to show only the result of the chirp input signal in our report. Since it is hard to compare and distinguish the difference between communication signals, we are going to perform both visual and subjective aural comparison. We not only plotted and compared the input and output signals, but also played them as sound waves to hear and find out how good the signal reconstruction is for each filter bank. We also try to introduce some sub-band channel distortion (e.g. zero out one sub-channel) to see its effect on filter bank reconstructions.

DFT Filter Bank

As we have mentioned before, the DFT polyphase filter bank is the easiest one to implement among these three filter banks, but it has the disadvantage of lack in ability of aliasing cancellation. The prototype filter we used is a very simple FIR (finite impulse response) filter with a cutoff frequency of $1/M$ (M is the number of sub-channels). Figure 4-1 shows the implementation result of DFT filter bank with a chirp input signal and sixteen sub-channels. Plot (a) displays the input signal in blue, output signal in red and error signal in cyan on the same axis. We can immediately observe that there is a great amount of error between input and output signals from the cyan color signal in this plot. Comparing the spectrogram of input signal and spectrogram of output signal in plot (b), we notice the occurring of some aliasing components in certain frequencies. This is mainly due to the aliasing effects produced by subsampling the sub-band signals. If we look at the frequency responses of these sub-filters, they are overlapping with each other, which creates interference among the filter banks and results in aliasing distortion. We

15

can easily hear the aliasing noise when we try to listen and compare the input and output
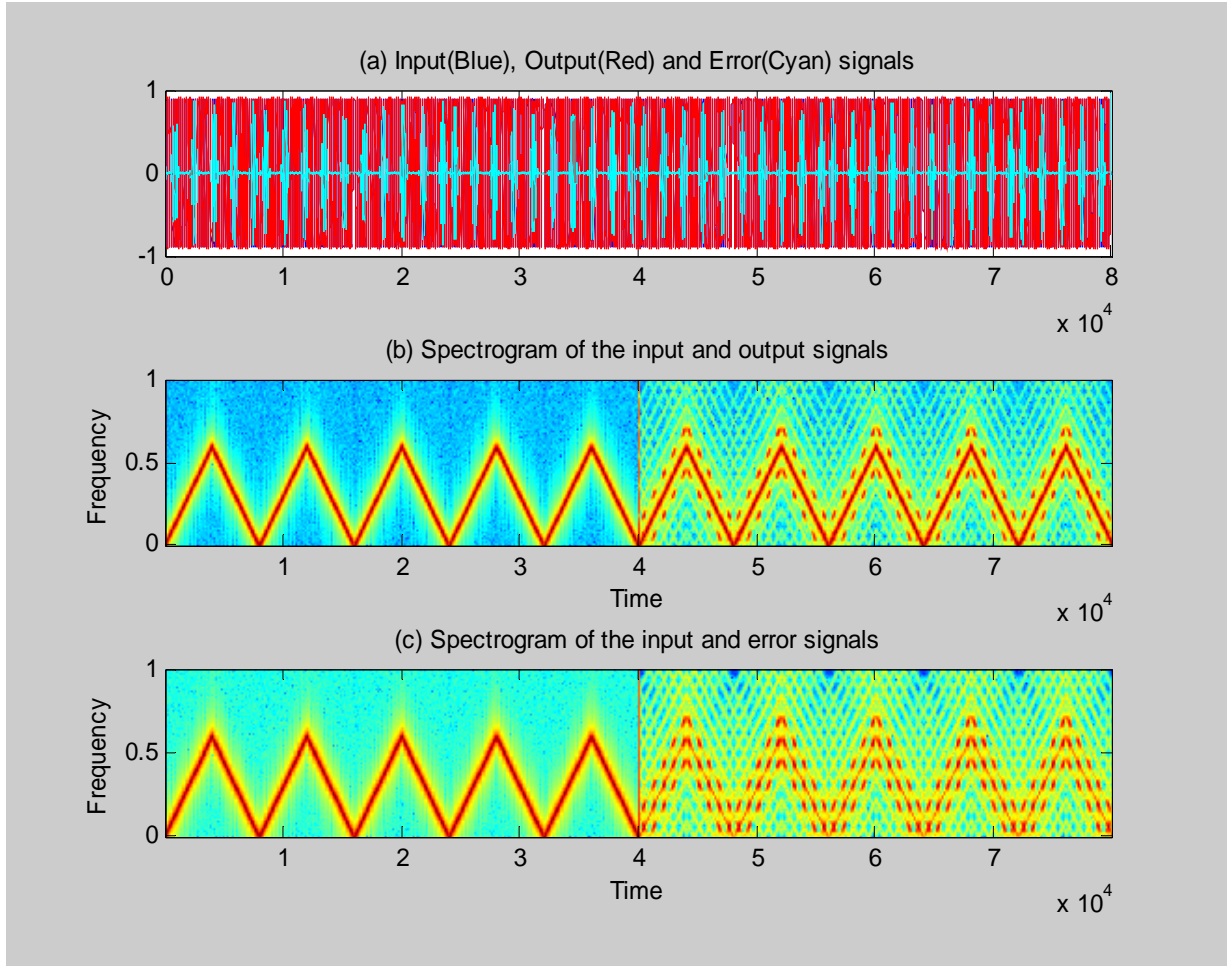
signals.

Figure 4-1   Implementation result of DFT filter bank:
Chirp input signal and 16 sub-channels

To find out the effect of the channel distortion on the DFT filter bank, we zero out

one of the sub-channels, and the implementation result is shown in Figure 4-2. In the time

domain plot (a), we see great amount of error at some time due to the fact of losing one

sub channel data. In the spectrogram plot (b) and (c), we can look at it from a frequency

point of view. In plot (b), we see the discontinuity of the output signal occurring at the

particular frequency. On the other hand, we see significant amount spectrogram of error signal occurring at the same exact frequency range in plot (c). We can conclude from the figure that zeroing out one sub-channel in DFT filter bank has only impact on the particular sub-channel that has been zeroed out, but no significant impact on the other sub-channels within the filter bank. The reason we see two symmetric chirp signals on the upper and lower part of the spectrogram plot (b) and (c) is because it shows both the imaginary and real parts of the signal due to the fact that when we zero out one sub channel, the output signal becomes complex due to the IDFT and DFT operations.
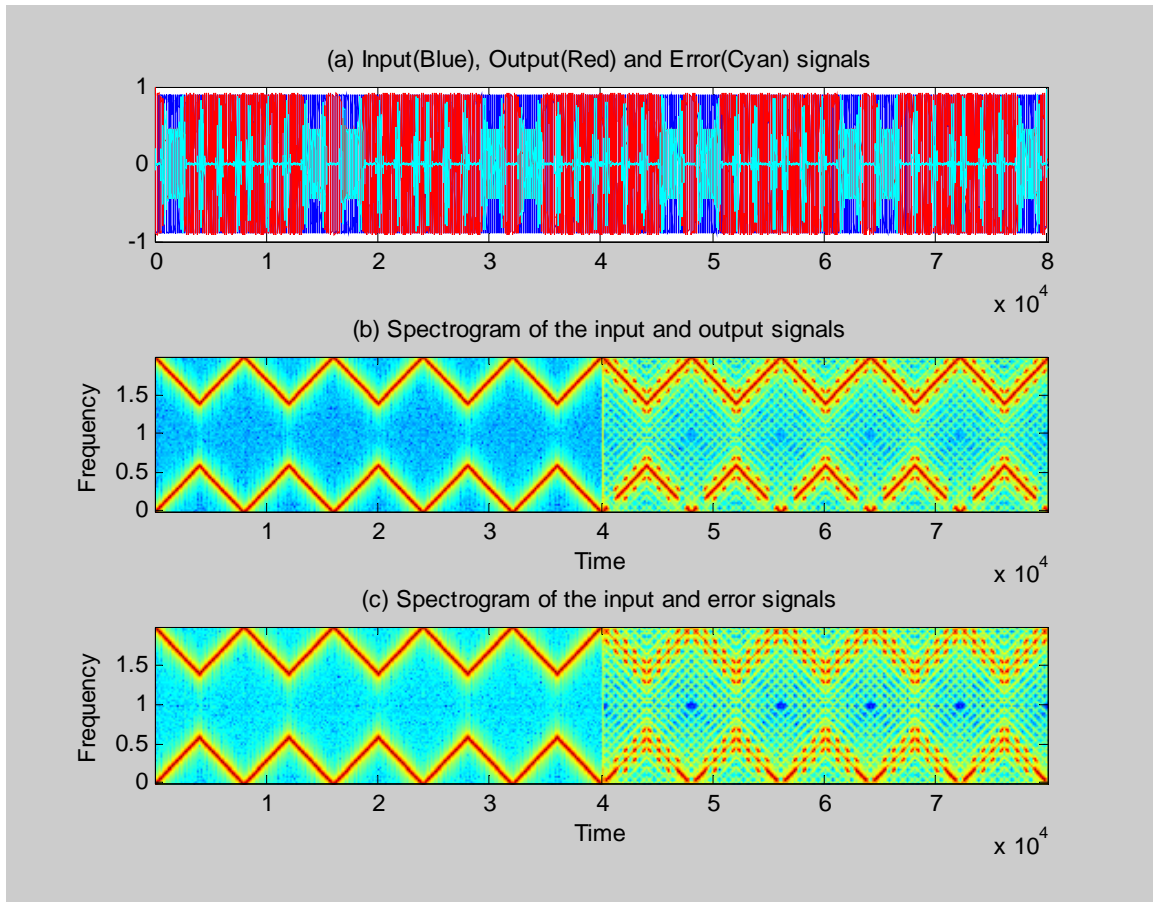

Figure 4-2   Implementation result of the DFT filter bank with one channel zeroed out: Chirp input signal and 16 sub-channels

17

MDFT Filter Bank

By introducing certain modifications to the DFT filter bank, we can improve the performance of filter bank significantly. MDFT filter bank has much better ability to handle the aliasing effects. The prototype filter design for MDFT filter bank can be done by either FIR method or the Kaiser Window method, which is the prototype filter design method for Cosine Modulated filter bank. Both methods will lead to very similar results. For the purpose of comparison, we use the FIR prototype filter design method for our MDFT filter bank implementation.

We have introduced two different MDFT filter bank structures in sections 3: one is the MDFT realized by 2 DFT filter banks; the other is the simplified and computational efficiency MDFT filter bank. The author of the MDFT filter bank only briefly mentions the MDFT realized by 2 DFT filter banks in his paper [5], and didn't go into detail discussion about its structure. We have spent some time to try to study and implement both structures of MDFT filter bank, and we found some interesting results.

Figure 4-3 illustrates the MDFT realized by 2 DFT filter banks' implementation result with the same prototype filter design as the DFT filter bank. Comparing with the performance of the DFT filter bank in Figure 4-1, we can clearly see the improvement on the signal reconstruction for this MDFT filter bank; the magnitude of the error signal in plot (a) is much less than the DFT filter bank's error signal, the spectrogram of output signal in plot (b) doesn't contain any significant aliasing components as the DFT filter bank does, and there are only few minimal errors in the spectrogram shown in the plot (c). We can also verify this improvement by listening to the output signal; we don't hear any aliasing noise in the signal background. However the performance of this MDFT

filter bank structure is much better than the DFT filter bank, but it is not as good as the performances of the simplified MDFT filter bank and the Cosine Modulated filter bank, which will be discussed in the following sections.
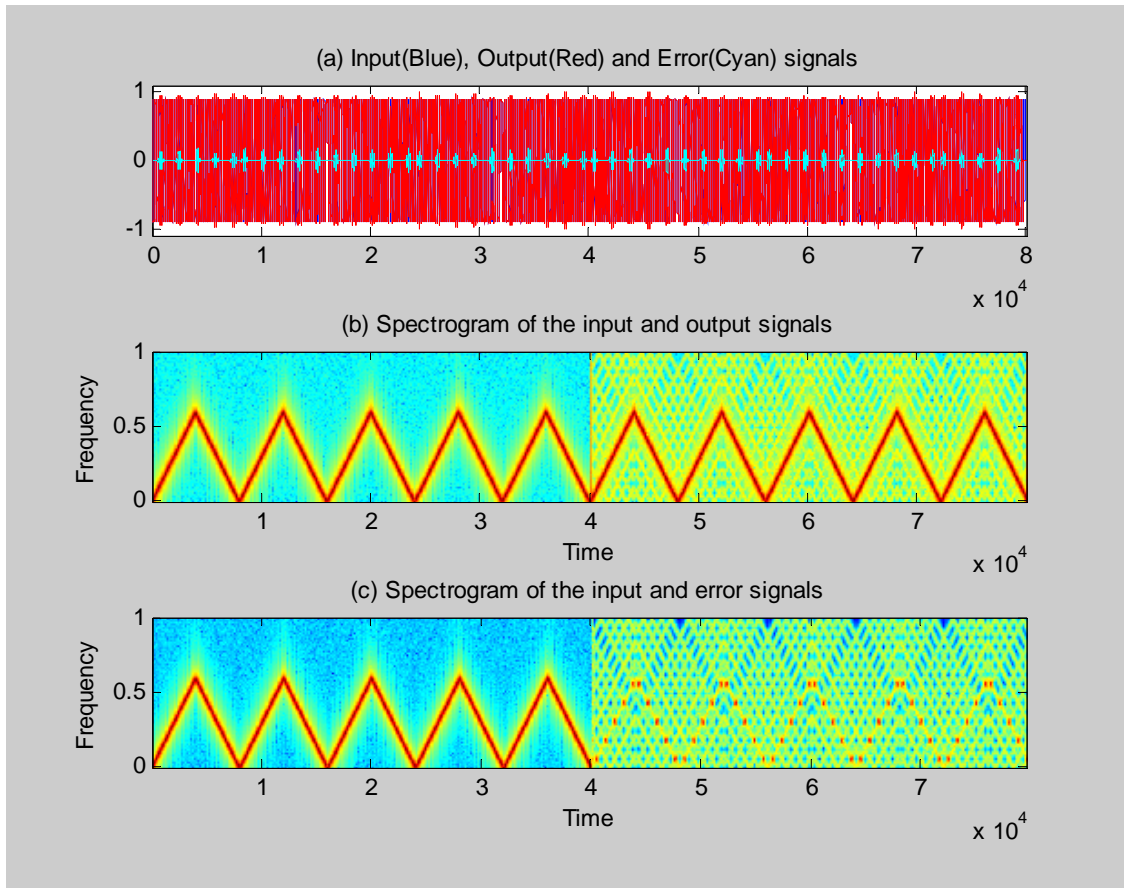
Figure 4-3   Implementation result of the MDFT realized by 2 DFT filter banks: Chirp input signal and 16 sub-channels

We are showing the implementation result of the second MDFT filter bank structure in Figure 4-4. For plot (a), the error between input and output signals is plotted at the same magnitude scale as the input and output signals seen as a straight line and it give us the impression that there is no error between input and output signals. In fact, the actual difference is of the order of 10^{-2}, which is hard to recognize without zooming in the plot.

19

Comparing with the implementation result of MDFT realized by 2 DFT filter banks in Figure 4-2, we also notice the improvement on the reconstruction in this MDFT filter bank structure. The spectrogram of the output signal in plot (b) doesn't show any significant aliasing components, and we only see some minor amount of error in plot (c). When we try to hear and distinguish the input and output signals, it's difficult to recognize the difference. So the performance of the MDFT filter bank in this case can be considered able to achieve nearly perfect reconstruction.
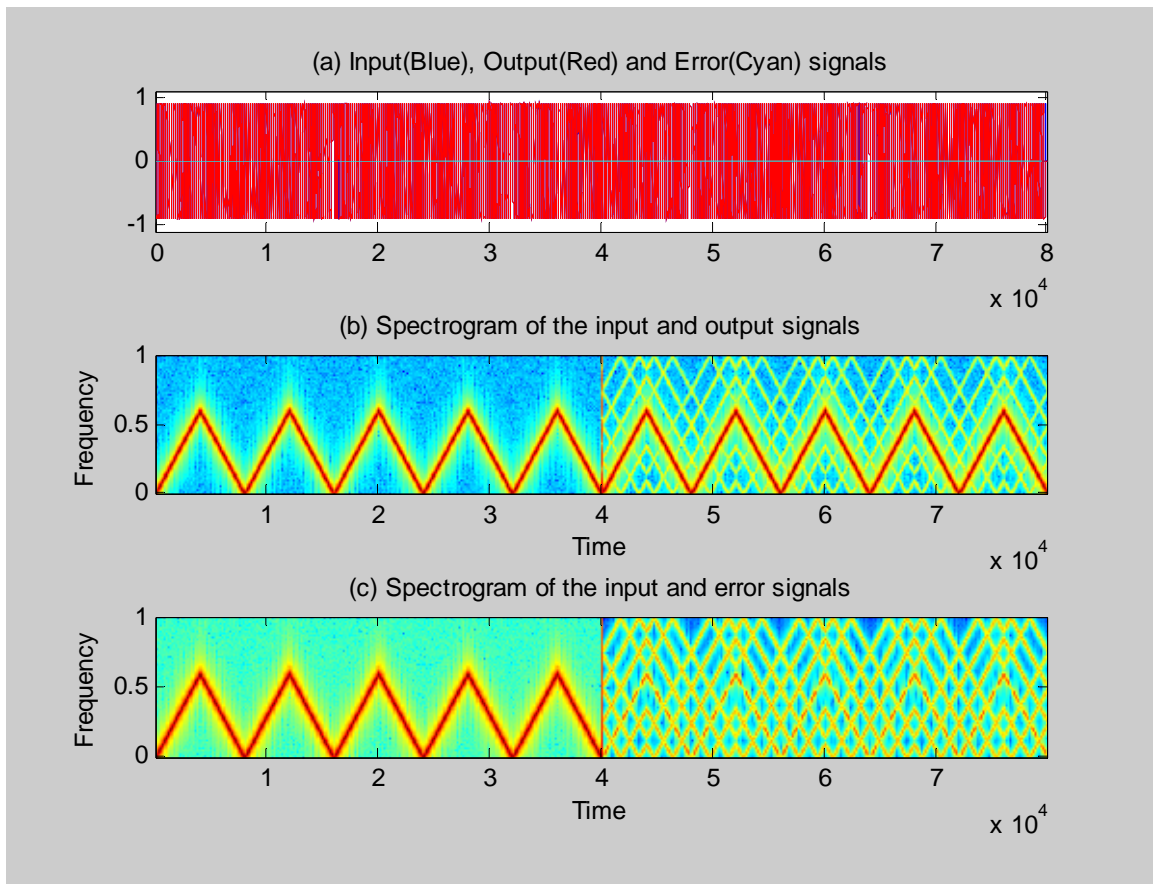

Figure 4-4   Implementation result of the simplified MDFT filter bank:
Chirp input signal and 16 sub-channels

The interesting result we found about the MDFT filter bank implementation is for the case when we introduce sub-band channel distortion by setting one sub-channel to zero

for both structures of the MDFT filter banks. It turns out that performance of the MDFT realized by 2 DFT filter banks in this situation is much better than the result for the simplified MDFT filter bank structure. The output results for these two MDFT filter bank structures are shown in Figure 4-5 and Figure 4-6 respectively. In Figure 4-5, we see a very similar result as the case of the DFT filter bank with one sub-channel zeroed out. The effect of the channel distortion in the structure of MDFT realized by 2 DFT filter banks only occurs on the particular channel that has been zeroed out; it doesn't have any significant impact on the other sub-channels within the filter bank.

Figure 4-5   Implementation result of MDFT realized by 2 DFT filter banks with one channel zeroed out: Chirp input signal and 16 sub-channels

21

In contrast, we can clearly see a huge amount of the error signal without needing to zoom in for the result of simplified MDFT filter bank in Figure 4-6. The effect of zeroing out one sub-channel is significant for this structure of the MDFT filter bank; it totally destroys the reconstruction for all the channels within the filter bank. We can easily hear the aliasing noise in the output signal, and we notice from the output signal's spectrogram in plot (b) that many significant aliasing harmonic components of the signal cross over with the actual output signal.
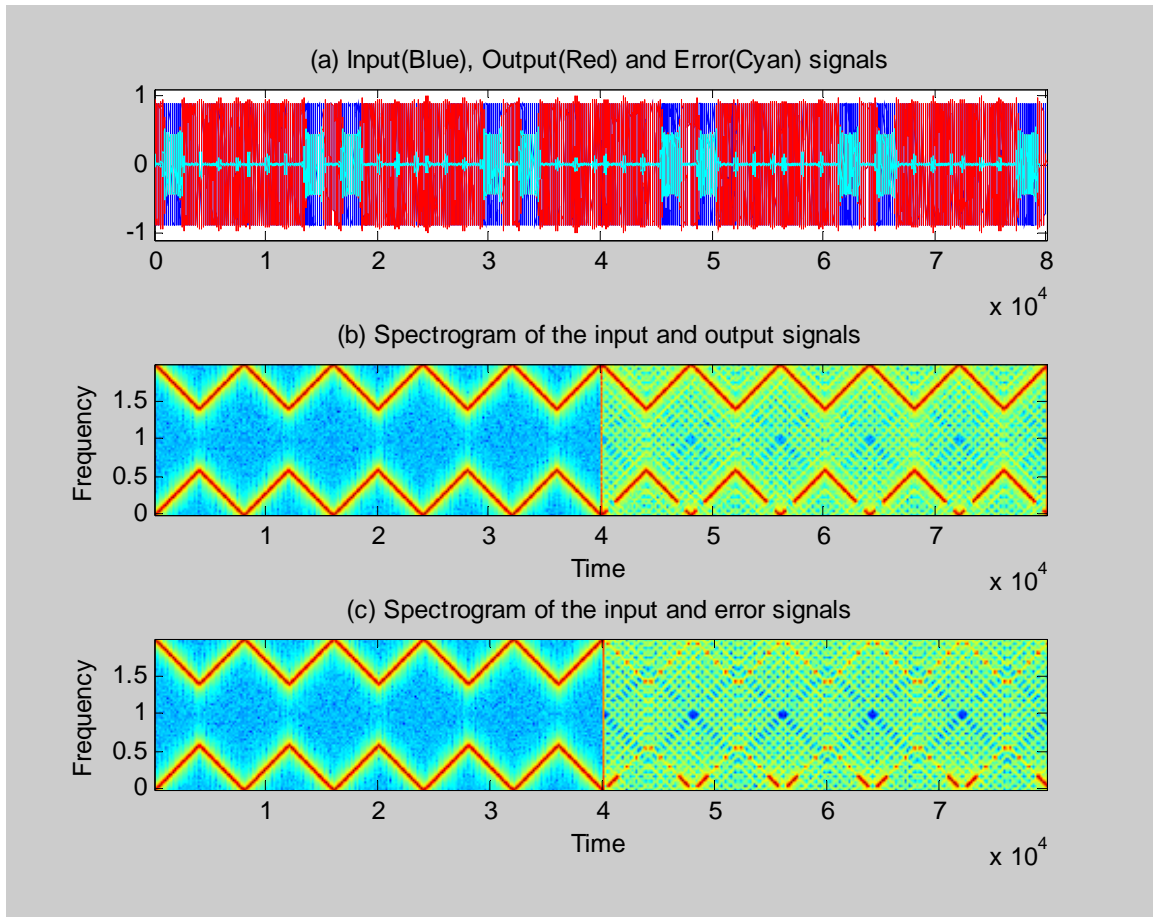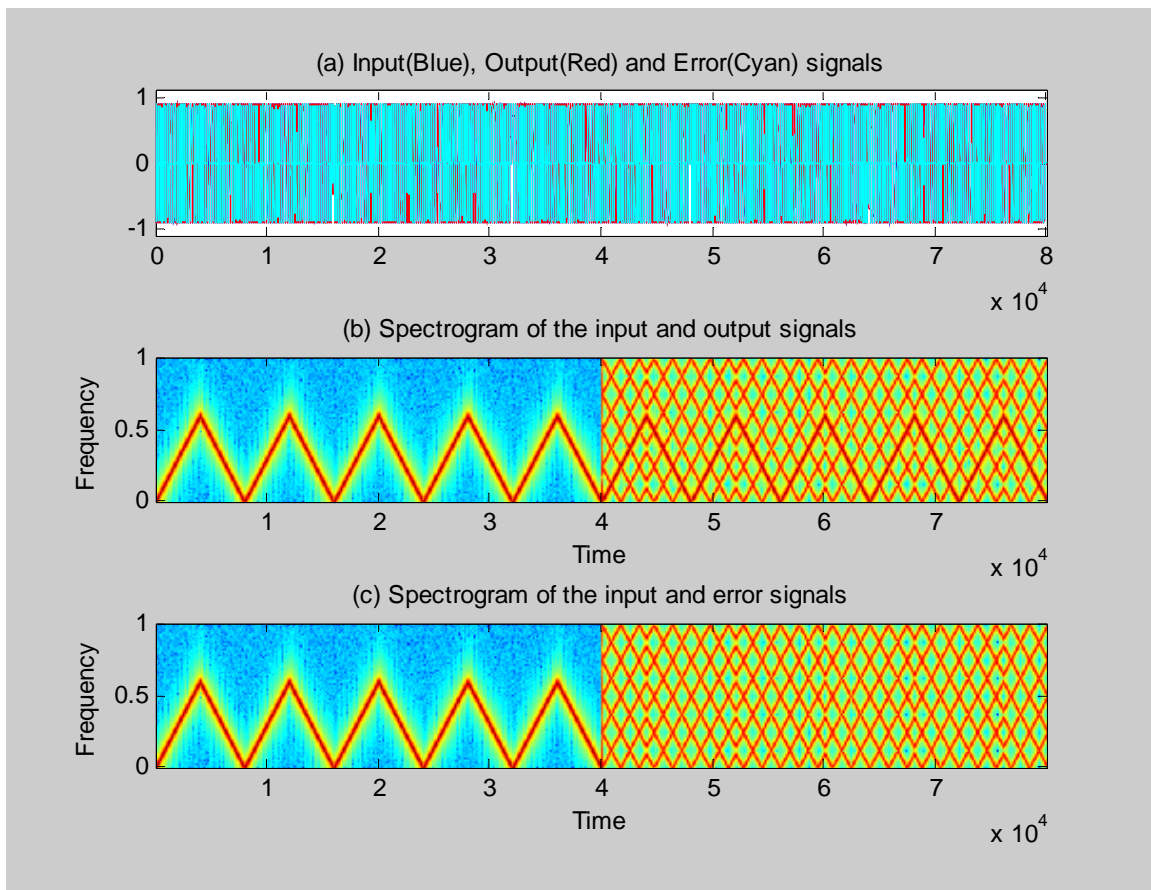

Figure 4-6   Implementation result of simplified MDFT filter banks with one channel zeroed out: Chirp input signal and 16 sub-channels

To conclude the performance of MDFT filter bank from our implementation results, we see that these two difference structures of the MDFT filter banks have advantages and disadvantages over each other for different situations. One performs well on the reconstruction while the other performs well when dealing with channel distortion. It is hard to say one is better than the other, but they can be alternative choices when we try to decide which structure to use for a particular situation or specific application.

Cosine Modulated Filter Bank

As we had discussed in section 3, there are several different prototype filter design methods for Cosine Modulated filter bank. For our implementation, we take advantage of the Kaiser Window design method [9], which only needs to optimize one parameter (the cutoff frequency) during the design process in order to generate the prototype filter. We found out that the prototype filter produced by the Kaiser Window method works very well in our simulation.

The implementation results is shown in Figure 4-7, where the error between input and output signals shown in plot (a) is at the same magnitude scale as the input signal. It again looks like a straight line without zooming in, and the actual error signal is of the order of 10^{-3}. The spectrograms of input and output signals in plot (b) are identical to each other, and we don't see any aliasing in plot (b) but only some minor error in plot (c). We cannot tell much difference between input signal and output signal by either comparing the plots or hearing the signals. We can tell from comparison of Figure 4-7 and Figure 4-4 that the Cosine Modulated filter bank works a little bit better than the MDFT filter bank.

23
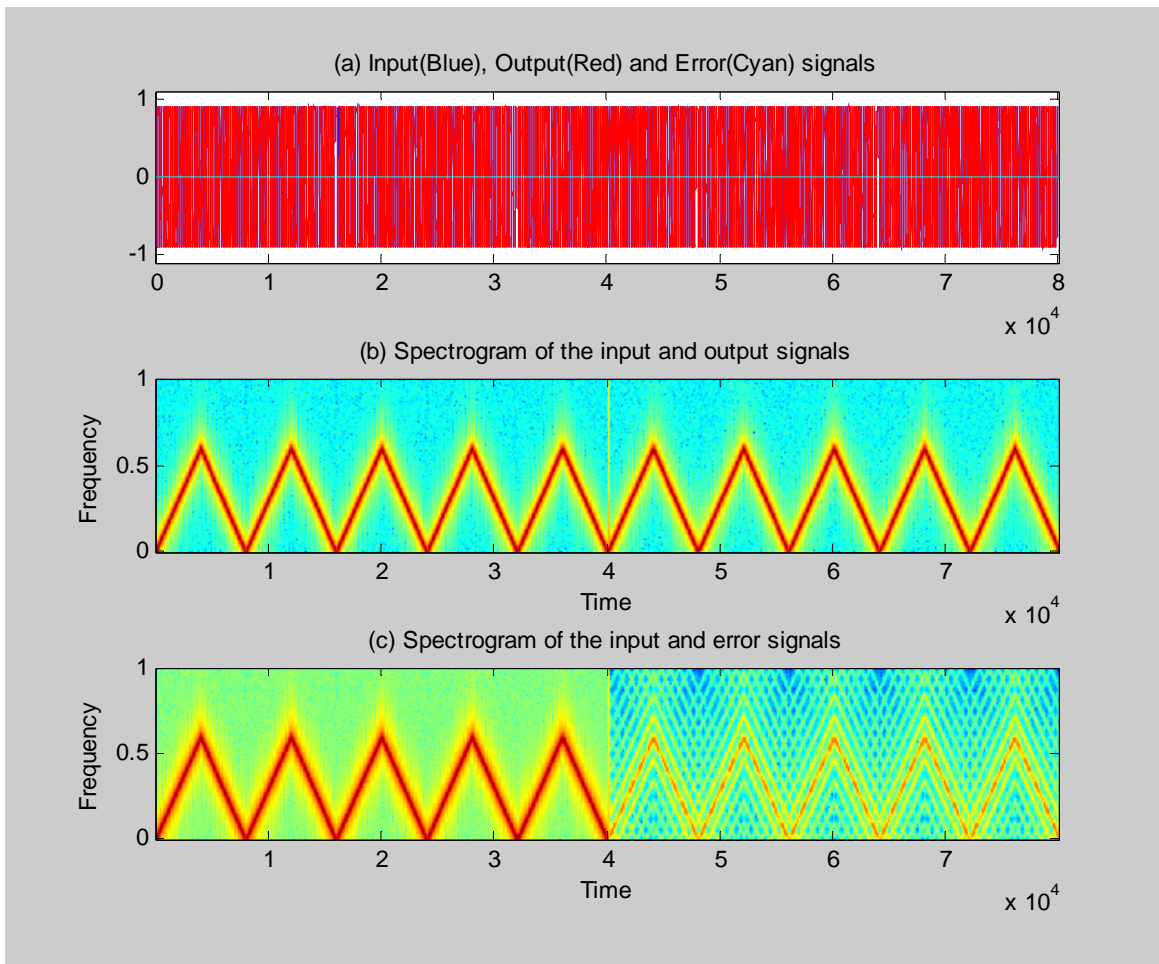
Figure 4-7   Implementation result of Cosine Modulated filter bank:
Chirp input signal and 16 sub-channels

Similar to MDFT filter bank, Cosine Modulated filter bank is sensitive to sub-band channel distortion. When we set one sub-channel to zero, the output signal of the simulation result is distorted as shown in Figure 4-8, we can clearly see the error signal in plot (a), and all the aliasing components are crossing over with the actual output signal in the spectrogram plot of output signal in plot (b). We can also clearly hear the aliasing noise in the background of output audio signal during playback. Comparing the response of simplified MDFT and Cosine Modulated filter bank to the sub-channel distortion, we notice that zeroing out one channel in MDFT filter bank leads to significant effect on all

the sub-channels within the bank, in contrast to the Cosine Modulated filter bank which has only a significant effect on the particular sub-channel that has been zeroed out and some minor effect on the neighboring sub-channels.

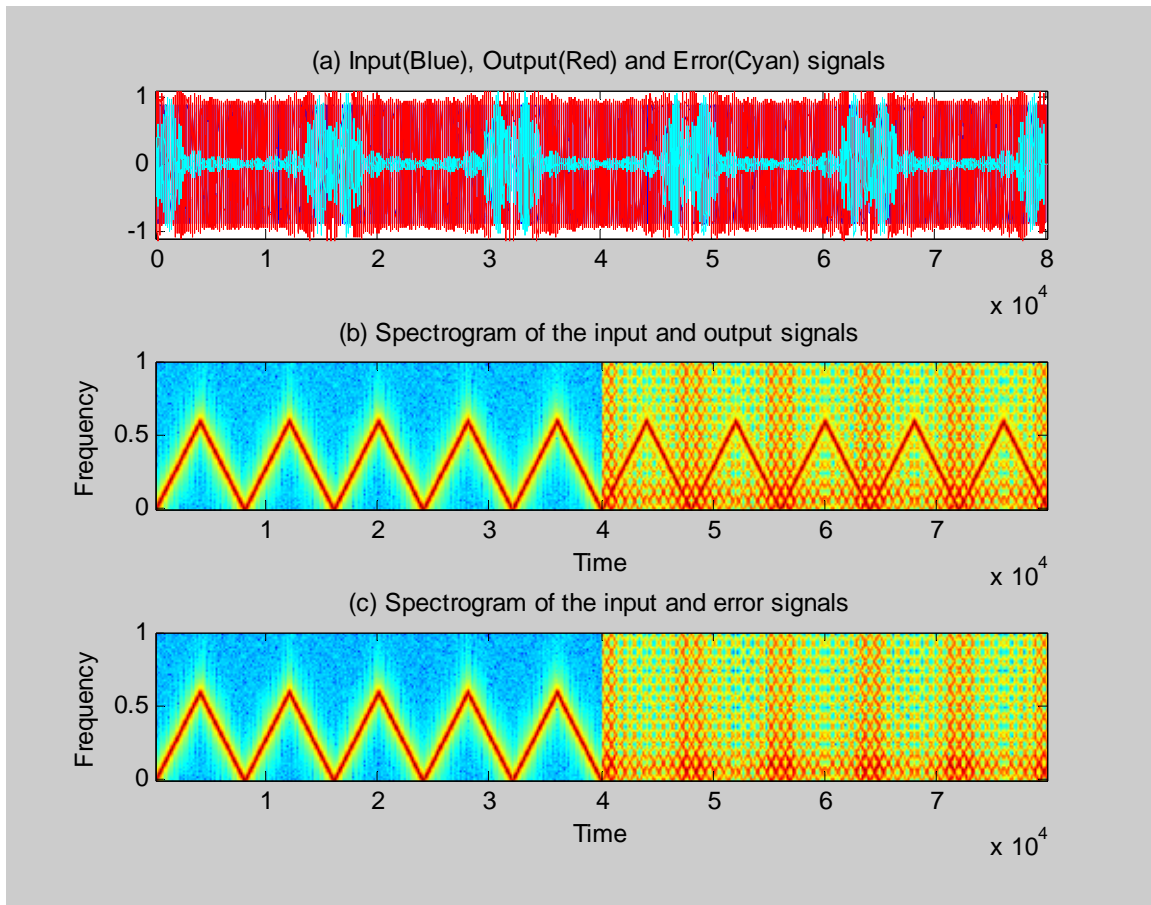Figure 4-8   Implementation result of Cosine Modulated filter bank with one channel zeroed out: Chirp input signal and 16 sub-channels

5.   Conclusions

In this report, we have reviewed three different types of polyphase filter banks: the DFT filter bank, MDFT filter bank and Cosine Modulated filter bank. For each of these filter banks, they consist of analysis and synthesis filters, which are derived from the

same prototype filter (PF). Summary of comparisons between these filter banks are listed in the two tables below.

Filter bank	PF Design Method	Length of PF	Number of Output Sample Delay
DFT	FIR	*M*N*	*Length(PF)-M*
MDFT realized by 2 DFT	FIR/Kaiser	*M*N*	*Length(PF)-M/2*
Simplified MDFT	FIR/Kaiser	*M*N*	*Length(PF)*
Cosine Modulated	Kaiser	*2*M*N*	*Length(PF)-M*

Table 1   Prototype filter comparison

Table 1 lists the comparison of prototype filter design for these filter banks, where *M* is the number of sub-channels in the filter bank and *N* is the number of samples in each polyphase component of the prototype filter. The prototype filter for DFT filter bank is a very easy FIR filter design with cutoff frequency of *1/M*, filter length of *M*N*, and the delay for the output signal is *length of prototype filter minus M* samples. The design of the prototype filter for Cosine Modulated filter banks is a little bit harder, as it involves an optimization process to find the cutoff frequency with Kaiser Window approach. It requires twice the length prototype filter for the same number of the sub-channels, and the delay for the output signal is also *length of prototype filter minus M* samples. The FIR prototype filter design of DFT filter bank can be used with the MDFT filter bank implementation with the same number of sub-channels. On the other hand, the Kaiser Window design of the Cosine Modulated filter bank with *M* sub channel can be used for the MDFT filter bank with *2M* sub-channels implementation. The delays for the outputs of two MDFT filter bank structures are different by *M/2* samples.

Table 2 shows the comparison for the performances of these filter banks in our simulation. DFT filter bank doesn't have a good reconstruction since it suffers from significant aliasing effects, but it doesn't cause huge impact to the neighboring channels when channel distortion occurs during the filter bank processing. Therefore DFT filter bank is good for channelization applications. The MDFT realized by 2 DFT filter banks has better performance than the DFT filter bank, with not only have the same advantage of minimal impact to neighboring channels from channel distortion as DFT filter bank, but also a lot of improvement on signal reconstruction. So it is suitable for both channelization and audio/image coding. Both simplified MDFT and Cosine Modulated filter banks are able to achieve nearly perfect reconstruction, but the Cosine Modulated filter bank has superior performance over the simplified MDFT filter bank in the channel distortion case as we have shown in our simulation.

Filter bank	Reconstruction	Channel Distortion	Application
DFT	Significant Aliasing	Minimal	Channelization
MDFT realized by 2 DFT	Minor Aliasing	Minimal	Audio/image coding Channelization
Simplified MDFT	Nearly Perfect	Significant	Audio/image coding
Cosine Modulated	Nearly Perfect	Significant or some	Audio coding

Table 2   Filter bank's performance comparison

A detailed comparison for these two filter banks can also be found in [5], which claims that the MDFT filter bank and Cosine Modulated filter bank have some similarities and some difference. The computational complexities of both kinds of filter banks are comparable, and the MDFT filter bank has certain advantages over Cosine Modulated filter bank in some applications.

6. References

[1] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, P T R prentice Hall, Englewood Cliffs, NJ, 1993.

[2] R.D. Koipillai and P.P. Vaidyanathan, "Cosine-Modulated FIR Filter Banks Satisfying Perfect Reconstruction", *IEEE transactions on Signal Processing*, vol. 40, pp. 770-783, April 1992.

[3] T.Q. Nguyen and R.D. Koilpillai, "The Theory and Design of Arbitrary-Length Cosine-Modulated Filter Banks and Wavelets, Satisfying Perfect Reconstruction", *IEEE Transactions on Signal Processing,* vol. 44, pp. 473-483, March 1996.

[4] N.J. Fliege, *Multirate Digital Signal Processing*, Chichester, U.K.: Wiley, 1994.

[5] T. Karp and N.J. Fliege, "Modified DFT Filter Banks with Perfect Reconstruction", *IEEE transactions on Circuits and Systems-II: Analog and digital signal processing,* vol. 46, pp. 1404-1414, November1999.

[6] T. Karp and N. J. Fliege, "Computational efficiency realization of MDFT filter banks," in *Proc. EURASIP European Signal Processing Conf.,* Trieste, Italy, Sept. 1996, PP. M83-1186.

[7] N.J. Fliege, "Modified DFT polyphase SBC filter banks with almost perfect reconstruction," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing,* vol. 2, pp. 149–152, April 1994.

[8] C. D. Creusere and S. K. Mitra, "A simple method for designing high quality prototype filters for M-band pseudo QMF banks," *IEEE Trans. Signal Processing*, vol. 43, pp. 1005–1007, April. 1994.

[9] Y.-P. Lin and P. P. Vaidyanathan, "A Kaiser window approach for the design of prototype filters of cosine modulated filter banks," *IEEE Signal Processing Lett.*, vol. 5, pp. 132–134, June 1998.

[10] F. Cruz-Roldan, P. Amo-Lopez, S. Maldonado-Bascon and S.S. Lawson, "An efficient and simple method for designing prototype filters for cosine-modulated Pseudo-QMF banks", *IEEE Signal Processing Lett.,* vol. 9, pp. 29-31, January 2002.

[11] R.B. Casey and T. Karp, "Performance analysis and prototype filter design for perfect-reconstruction cosine-modulated filter banks with fixed-point implementation", *IEEE transactions on circuits and systems-II*: express briefs, vol. 52, pp. 452-456, August 2005.

[12] B.W. Suter, *Multirate and wavelet signal processing,* San Diego, Acadmeic Press, 1998.

7. List of Acronyms

DSP   Digital Signal Processing

DFT   Discrete Fourier Transform

FIR   Finite Impulse Response

IDFT   Inverse Discrete Fourier Transform

MDFT   Modified Discrete Fourier Transform

PF   Prototype Filter

PR   Perfect Reconstruction