Perfect Reconstruction Filter Bank Structure Based On Interpolated FIR Filter

Jorge E. Cadena

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

A. A. (Louis) Beex, chair
Harpreet S. Dhillon
William T. Baumann

May, 2016
Blacksburg, Virginia

Keywords: Filter Banks, Interpolated FIR Filters, Perfect Reconstruction

Perfect Reconstruction Filter Bank Structure Based On Interpolated FIR Filter

Jorge E. Cadena

ABSTRACT

State of the art filter bank structures achieve practically perfect reconstruction with very high computational efficiency. However, the increase in computational requirements due to the need to process increasingly wider band signals is paramount. New filter bank structures that provide extra information about a signal while achieving the same level of required efficiency, and perfect reconstruction properties, need to be developed. In this work a new filter bank structure, the interpolated FIR (IFIR) filter bank is developed. Such a structure combines the concepts of filter banks, and interpolated FIR filters. The filter design procedures for the IFIR filter bank are developed and explained.

The resulting structure was compared with the non-maximally-decimated filter bank (NMDFB), achieving the same performance in terms of the number of multiplications required per sample and the overall distortion introduced by the system, when operating with Nyquist prototype filters.

In addition, the IFIR filter is tested in both simulated and real communication environments. Performance, in terms of bit-error-rate, was found to not be degraded significantly when using the IFIR filter bank system for transmission and reception of QPSK symbols.

Perfect Reconstruction Filter Bank Structure Based On Interpolated FIR Filter

Jorge E. Cadena

GENERAL AUDIENCE ABSTRACT

In communication systems applications it is desired to utilize the frequency bands available in an efficient manner. For example, if data transmission (such as cellular or satellite) occurs at “X” MHz, and another at “Z” MHz, it would be of interest to place a third user at the “Y” MHz band without interfering with the other two. In other applications the shuffling of frequency bands used by multiple users can be useful for dynamic allocation of spectrum resources. This is of increasing interest as the frequency spectrum has an abundance of “white holes” that are not commonly used in the licensed spectrum bands.

Perfect reconstruction filter banks allow for bandwidth efficient ways to share the frequency spectrum. They allow for multiple users to operate in adjacent bands without interfering with one another. Filter banks are also efficient in the sense of less hardware components, and this translates in more time to process a signal and/or less cost and power required.

In this work a new filter bank structure that competes with the state of the art is developed and tested.

Acknowledgements

I want to thank my graduate advisor, Dr. A. A. (Louis) Beex. I will never forget how he opened many doors to a student who was just really excited about filters, nor all the time he invested in my learning about signal processing concepts, even before becoming my adviser. I have very dear memories, and have met very talented people in the DSPRL, and this is all thanks to Dr. Beex. I would also like to thank Dr. William T. Baumann and Dr. Harpreet Dhillon for graciously accepting to be part of my thesis committee.

Special thanks to my brother, Jose, for always being the best example of an older brother that one can hope for. Thank you for always being there for me. You will always be my role-model.

I dedicate this thesis to my mother, Narciza. None of this would be possible without her, and my accomplishments will always be her accomplishments as well.

Contents

List of Abbreviations

AFB	Analysis Filter Bank
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CMFB	Cosine Modulated Filter Bank
CPFB	Cascaded Polyphase Filter Bank
DFT	Discrete Fourier Transform
DPLL	Discrete Phase Lock Loop
EbNo	Energy per Bit to Noise Power Density Ratio
EMFB	Exponential Modulated Filter Bank
FB	Filter Bank
FIR	Finite Impulse Response
IDFT	Inverse Discrete Fourier Transform
IFIR	Interpolated Finite Impulse Response
IFIR-FB	IFIR Filter Bank
IR	Image Rejection
ISI	Inter-Symbol Interference
MAC	Multiply-Accumulate
MDFT	Modified DFT Filter Bank
NMDFB	Non-Maximally-Decimated Filter Bank
PR	Perfect Reconstruction
PRFB	Perfect Reconstruction Filter Bank
PRQMFB	Perfect Reconstruction Quadrature Mirror Filter Bank
QMF	Quadrature Mirror Filter
QPSK	Quadrature Phase Shift Keying
Rx	Receiver
SFB	Synthesis Filter Bank
SNR	Signal to Noise Ratio
SPS	Samples Per Symbol
SQR-Nyquist	Square-root Nyquist
SRRC	Square-root Raised Cosine
Tx	Transmitter

Chapter 1

1 Introduction

Multirate signal processing has attracted a lot of attention in the past decades for its uses in speech coding, image processing, channelization of signals, and their transmission in digital communications, etc.

A digital system becomes a *multirate* system when multiple sampling rates are used within the system. The need to change the sampling rate of a signal is sometimes unavoidable, for example when there is a need to synchronize data from multiple systems that do not work at compatible sampling rates. At other times, multirate processing is desirable, for example when signals can be represented at lower sampling rates. Multirate processing allows for very efficient filter structures.

Multirate processing also allows for processing many signals simultaneously at lower sampling rates, or to divide the processing workload of a wide-band signal by dividing, or *analyzing* it into sub-bands that can be processed at slower rates. Analyzing a signal comes with the challenge of being able to *synthesize*, that is being able to reconstruct the wide-band signal from its sub-bands with the least amount of distortion possible. These properties bring forth the concepts of filter banks, which have been studied extensively in the last 40 years.

1.1 Filter Banks

Filter banks (FBs) are a collection of filters that work in parallel on a common input to produce multiple outputs, or to produce a common output from multiple inputs. An *analysis* FB takes an input $x(n)$ and produces M outputs $y_k(n)$ as in Figure 1-1a. In contrast, a *synthesis* FB takes M inputs and combines them into a single output as seen in Figure 1-1b.

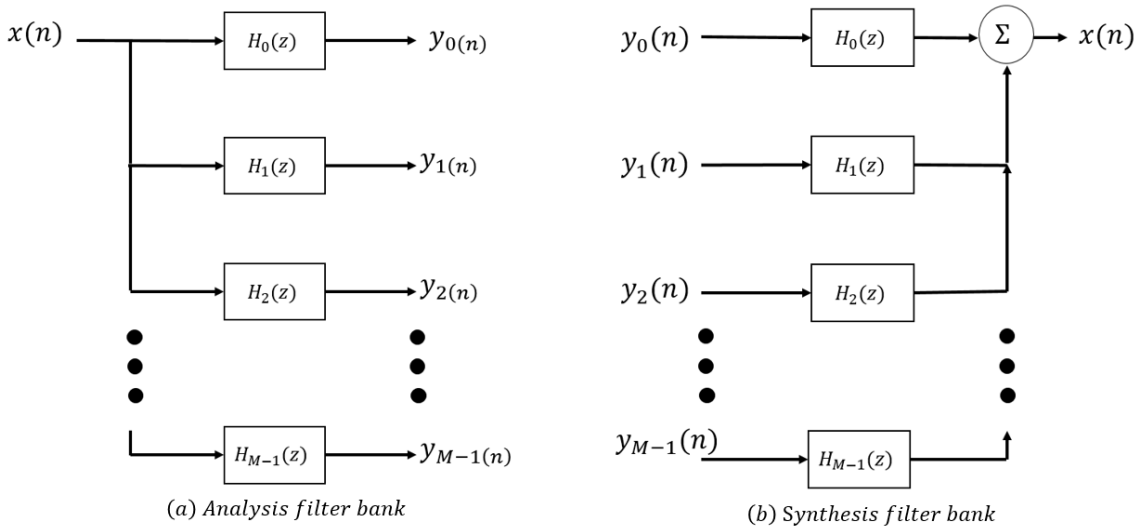


Figure 1-1 Synthesis and analysis filter banks

The filters in the analysis filter bank operate on the signal $x(n)$ to obtain the sub-band signals $y_k(n)$, $0 \le k \le M-1$. This is illustrated in Figure 1-2 in the frequency domain (with period 2π). Note that uppercase letters are used for z-transformation of a time sequence.

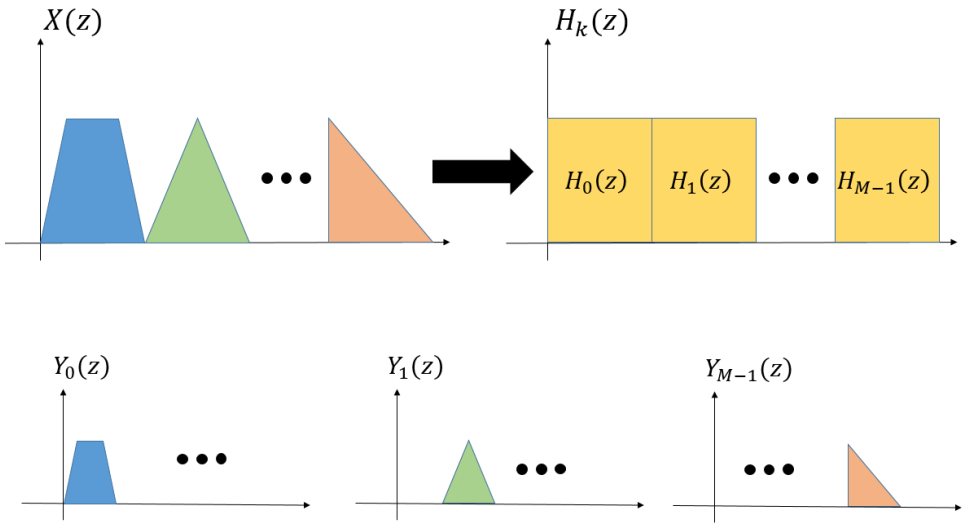


Figure 1-2 Illustration of the analysis filter bank

In contrast, the synthesis filter bank extracts the sub-band signals $y_k(n)$ and combines them in order to obtain $x(n)$. This is shown in Figure 1-3. Note that the sub-band signals may be subject to noise and/or some other source of interference, hence the need for the synthesis filters.

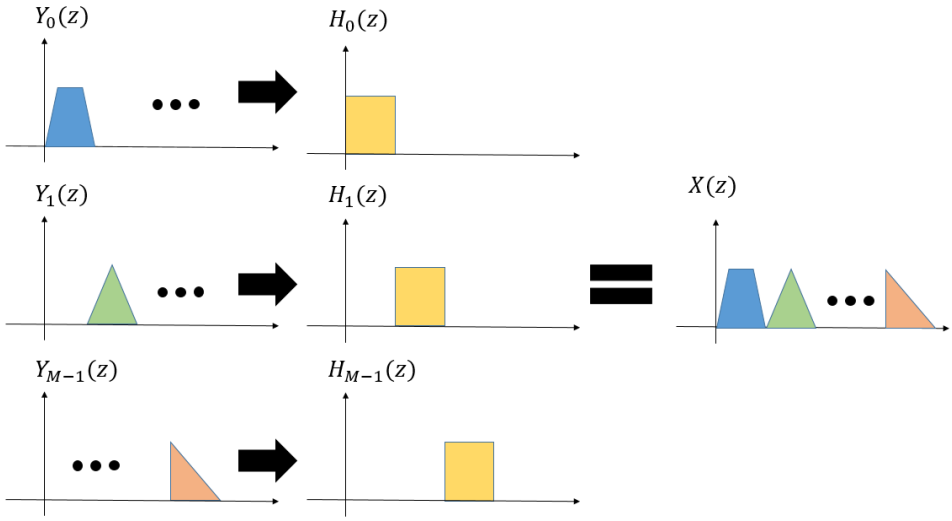


Figure 1-3 Illustration of synthesis filter bank

The sub-band signals $y_k(n)$ have a bandwidth M times smaller than the original $x(n)$ which, by the Nyquist sampling theorem, means that they can be sampled at a rate M times lower. This reduction in sampling rate translates to the processing time available per sub-band being increased by a factor M.

One issue with separating and reconstructing $x(n)$ is that real filters are not 'brick-wall' filters, that is, they are not band-limited and they overlap with one-another. Figure 1-4 shows a more realistic 8-channel analysis filter bank.

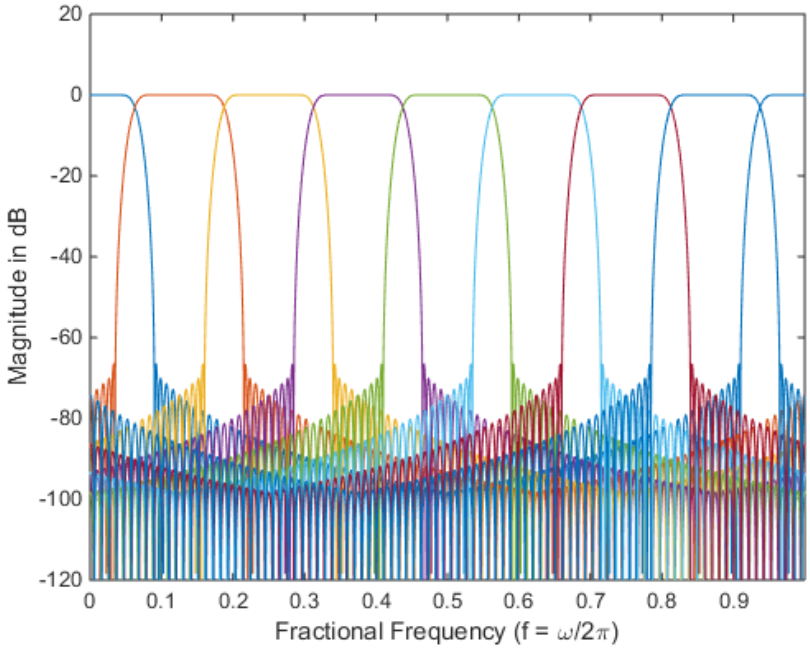


Figure 1-4 Realistic analysis filter bank

This suggests that special considerations are needed in the design of the analysis and synthesis filters, which translate into the perfect reconstruction conditions for filter banks.

A filter bank is said to have the **perfect reconstruction** (PR) property if the signal $x(n)$ can be recovered without distortion from its sub-band components $y_k(n)$. There has been extensive research in the topic of perfect reconstruction filter banks (PRFB). For the case of 2-channels (QMF filter banks) the conditions for perfect reconstruction have been shown [1] to be satisfied with very simple relationships between analysis and synthesis filters.

For the *M*-channel case, the conditions are not so simple, and the classical concept of *lossless* matrices [2, 3] (also called the "paraunitary property" [4], which is the term adopted in this work) is necessary to design these filters with relative ease. These filter bank structures and concepts are reviewed in Chapter 2.

1.2 Applications of Multirate Processing and Filter Banks

Multirate processing and filter banks have a wide variety of applications. Some of these include:

1. *Arbitrary sample rate change:* As mentioned before, multirate systems allow for sampling rate change between sub-systems, which can be either a necessity or just for convenience. For example, audio signals are commonly sampled at a rate of 44.1 kHz. In the digital-to-analog conversion of these signals, an analog filter with a cutoff frequency of 22 kHz is needed. What is often done is to increase the sampling rate of the audio signal before conversion, to 128 kHz. This relaxes the analog filter requirements to have a cutoff frequency of 64 kHz, and consequently a much larger transition band.

2. *Sub-band coding of speech and image signals:* Practical signals, such as speech, often have most of their power/information concentrated in a specific frequency band. If it is desired to represent signals digitally, then it is of interest to allocate more bits where most of the application-relevant information can be found. Filter banks allow for the separation of the sub-bands, and for speech this separation is often done in frequency octaves by using the tree-structure filter bank [5]. This is sometimes called the generalized discrete time wavelet transform. Both speech and image signals have most of their power in the lower octave range, and with the use of filter banks, the compression of these signals can be done more efficiently.

3. *Transmultiplexers:* In multicarrier communications it is desired to modulate symbol streams with different carrier frequencies. This can be achieved with the synthesis filter bank, where N bit (or symbol) streams are modulated into a single signal with N times higher sampling rate [6] and then transmitted. At the receiver side the corresponding analysis filter bank separates the symbol streams and recovers the information. The transmitter and receiver modulation and demodulation can be done very efficiently, with fewer multiplications per sample, thanks to multirate processing blocks.

1.3 Motivation and Outline

The efficient allocation of bandwidth has become essential with the increasing need of sharing communication channels among users. Increasing data rates reduces the time allocated to signal

processing routines such as matched filtering, carrier synchronization, and equalization. This may result in signal processing engines saturating even with the simplest filtering operation. While filter banks have been highly effective in this regard, more efficient filter bank structures are needed if processing of very wideband signals is desired.

The goal of this work is to develop a new structure that combines two efficient FIR filter implementations: filter banks, and interpolated finite impulse response (IFIR) filters. Analysis and synthesis filter bank structures are made efficient thanks to the polyphase representation of filters, a milestone concept in multirate processing. The IFIR system is formed by the cascade of two filters that achieve the same performance as the FIR version, but with a pronounced reduction in the number of multiplications required per sample. To combine these two methods, the problem of combining two filters via their polyphase implementation needs to be addressed.

This thesis is organized as follows. In Chapter 2 the basic building blocks of multirate processing are reviewed, working up towards efficient structures to achieve rate changes. These are the basic ingredients to start talking in depth about filter banks, and satisfaction of which conditions is required for perfect reconstruction. The IFIR filter design and properties are also introduced. In Chapter 3 the IFIR-based filter bank is constructed, starting from the problem of cascading two filters in terms of their polyphase components. In Chapter 4 the performance of the IFIR filter bank is compared to other state of the art filter bank structures, in terms of multiplications per sample and how closely the perfect reconstruction conditions are met. Chapter 5 deals with the performance of the IFIR filter bank under real conditions, that is, with practical communication signals. The conclusions and observations of this work are given in Chapter 6, where possibilities for future work on the IFIR filter bank are given as well.

Chapter 2

2 Background

In this chapter the basics of multirate processing are reviewed as necessary for a deeper understanding of filter banks, including different filter bank structures and the conditions for perfect reconstruction. The concept of interpolated finite impulse response (IFIR) filters, which is one of the building blocks in the research effort described here, is reviewed also.

2.1 Multirate Processing Blocks

In this section the basic building blocks of multirate processing are explored, eventually leading to perfect reconstruction filter banks.

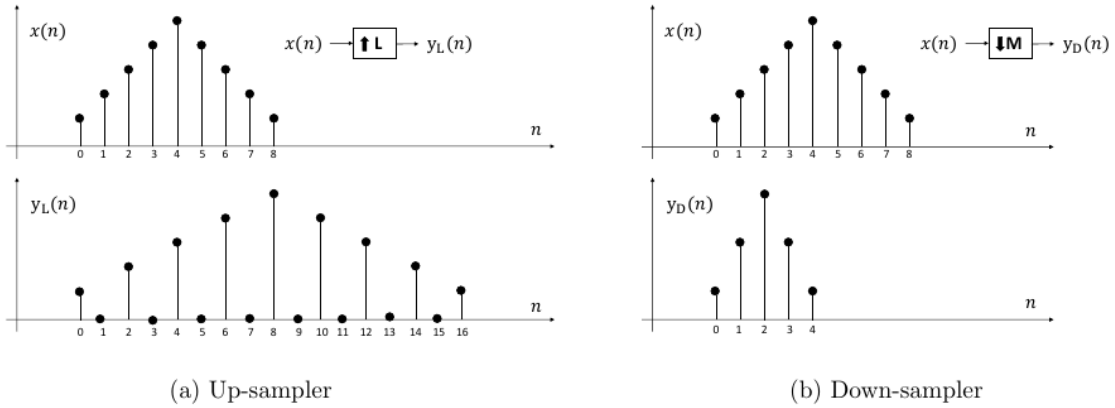
2.1.1 Down-samplers and Up-samplers

Down-samplers and up-samplers are the most basic blocks in multirate processing.

An L-fold **up-sampler** takes the input signal $x(n)$ and produces the output sequence

$$y_L(n) = \begin{cases} x\left(\frac{n}{L}\right) & n = 0, \pm L, \pm 2L, ..., \pm kL \\ 0 & otherwise \end{cases} \quad (1)$$

In other words, the up-sampler produces a longer sequence that has $L-1$ zeros in between each two samples of $x(n)$, as illustrated in Figure 2-1a for $L = 2$.



(a) Up-sampler (b) Down-sampler

Figure 2-1 Up/down-sampler effect in time domain

An *M*-fold **down-sampler** takes an input signal $x(n)$ and produces the output sequence

$$y_M(n) = x(Mn), \tag{2}$$

where $y_M(n)$ retains only the M^{th} sample of $x(n)$. Figure 2-1b illustrates the effect of the down-sampler on a time domain signal for $M = 2$.

The effect of the up/down-sampler can also be expressed in the frequency domain by taking the z-transform of $y_L(n)$ and $y_M(n)$. Starting with the up-sampler, the output can be written as

$$\begin{aligned} Y_L(z) &= \sum_{n=-\infty}^{\infty} y_L(n) z^{-n} \\ &= \sum_{n=0,\pm L,\ldots} y_L(n) z^{-n} \\ &= \sum_{k=-\infty}^{\infty} y_L(kL) z^{-kL} \\ &= \sum_{k=-\infty}^{\infty} x(k) z^{-kL} \end{aligned} \tag{3}$$

Thus, by substituting $z = e^{-j\omega}$

$$Y_L(\omega) = X(\omega L) \tag{4}$$

In a similar manner, but with a more involved proof (see [4, 7]) an expression for the down-

sampled output $Y_M(\omega)$ is derived

$$Y_M(z) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{\frac{1}{M}} W_M^k\right), \tag{5}$$

where W_M is the M^{th} root of unity defined as (subscript will be dropped when it is implicit from context)

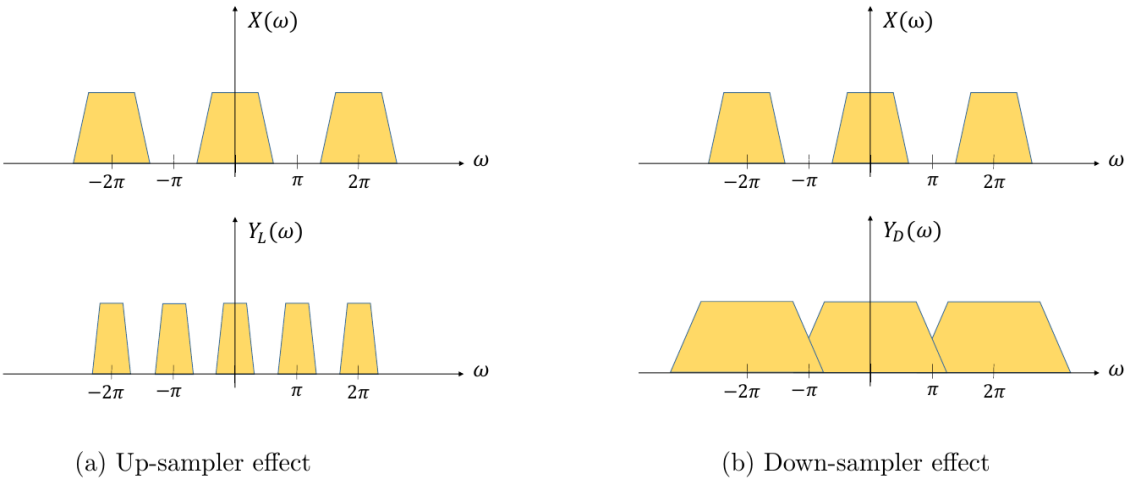
$$W_M = e^{-j\frac{2\pi}{M}} \tag{6}$$

Substituting (6) into (5), $Y_M(\omega)$ can be written as

$$Y_M(\omega) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{j\frac{\omega-2\pi k}{M}}\right) \tag{7}$$

$Y_L(\omega)$ in (4) shows that the output of the up-sampler is a frequency compressed version of the input. Thus, if we look at $Y_L(\omega)$ for $0 \le \omega \le \pi$ (the Nyquist interval) we will see compressed images of $X(\omega)$ at intervals of $2\pi / L$. This *imaging effect* is illustrated in Figure 2-2a for $L = 2$

On the other hand, the down-sampled output in (7) refers to the sum of stretched (by *M*) and shifted versions of $X(\omega)$. In this case the sampling rate is being decreased, and there is the potential of spectral folding and aliasing, which will result in a loss of information about $X(\omega)$. Figure 2-2b illustrates the stretching in frequency, and the potential for aliasing if $X(\omega)$ is not bandlimited to a bandwidth of $\omega = \pi / M$ before going through the down-sampler.



(a) Up-sampler effect (b) Down-sampler effect

Figure 2-2 Up/down-sampler effect in frequency

Note from Figure 2-2a that rejecting the passband images in $Y_L(\omega)$ that were created *after* the up-sampler would effectively produce a signal that is sampled L times faster than before. In contrast, bounding $X(\omega)$ to have a bandwidth of π / M *before* the down-sampler results in a signal that is sampled M times slower, and not corrupted by aliasing. These two operations are called ***interpolation*** and ***decimation***, both depicted in Figure 2-3. The interpolator consist of an up-sampler followed by an interpolation filter, with system function $H_L(z)$, that rejects the signal images at the high sampling rate. The decimator consists of a decimation filter $H_D(z)$, that makes sure aliasing does not occur, followed by the signal being down-sampled.

(a) Interpolator (b) Decimator

Figure 2-3 Decimator and interpolator blocks

Details in the design procedure of $H_L(z)$ and $H_D(z)$ can be found elsewhere [4. 7].

2.1.2 The Noble Identities

The Noble identities are two of the most important and valued identities used in multirate processing. These identities facilitate commuting down-samplers and up-samplers with filters (provided their powers of z are integers after said commutation) with very simple relationships,

even given the time-varying nature of up-samplers and down-samplers. Figure 2-4 depicts the Noble identities used to commute both the up-sampler/down-sampler with an LTI system.

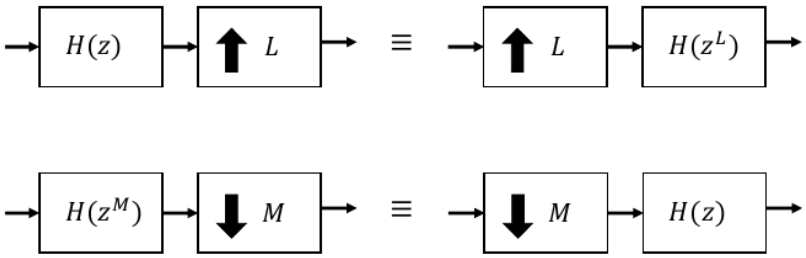


Figure 2-4 Noble identities

It should be noted that up/down-samplers are also commutable with memoryless operations such as addition and multiplication, as shown in Figure 2-5 for the up-sampler case.

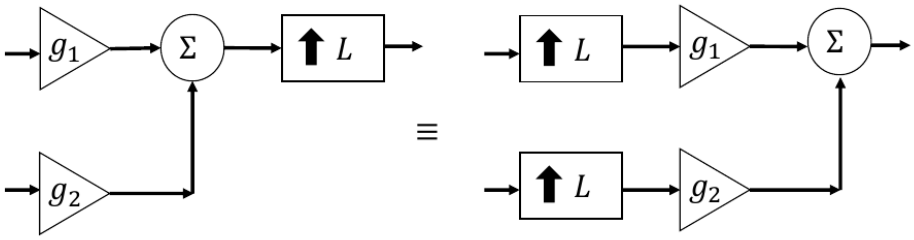


Figure 2-5 Commutation with adders and gains

These relationships are helpful in the analysis of efficient filter banks in future sections.

2.1.3 The Polyphase Representation

Consider a filter $H(z)=\sum_{n=0}^{\infty}h(n)z^{-n}$ with transfer function

$$H(z)=1+2z^{-1}+3z^{-2}+4z^{-3}+... \tag{8}$$

It is possible to separate (8) into even and odd powers of z as

$$H(z)=\left(1+3z^{-2}+5z^{-4}+...\right)+\left(2+4z^{-2}+6z^{-4}+...\right)z^{-1} \tag{9}$$

or

$$H(z)=H_0\left(z^2\right)+H_1\left(z^2\right)z^{-1} \tag{10}$$

$H_0(z)$ and $H_1(z)$ are called *polyphase components* of $H(z)$ and, for this 2-fold decomposition, are defined as:

$$H_0(z) = \sum_{n=0}^{\infty} h(2n) z^{-n} \tag{11}$$

$$H_1(z) = \sum_{n=0}^{\infty} h(2n+1) z^{-n} \tag{12}$$

Generalizing this result, $H(z)$ can be decomposed into M polyphase components as

$$H(z) = \sum_{k=0}^{M-1} H_k\left(z^M\right) z^{-k} \tag{13}$$

where

$$H_k(z) = \sum_{n=0}^{\infty} h(Mn+k) z^{-n}, \quad 0 \le k \le M-1 \tag{14}$$

A signal flow graph of the polyphase implementation of a filter is shown in Figure 2-6. An input $X(z)$ is separated into M branches by a delay line. Each branch is filtered by the k^{th} up-sampled polyphase filter component of $H(z)$, and then added to form the filtered output $X(z)H(z)$.

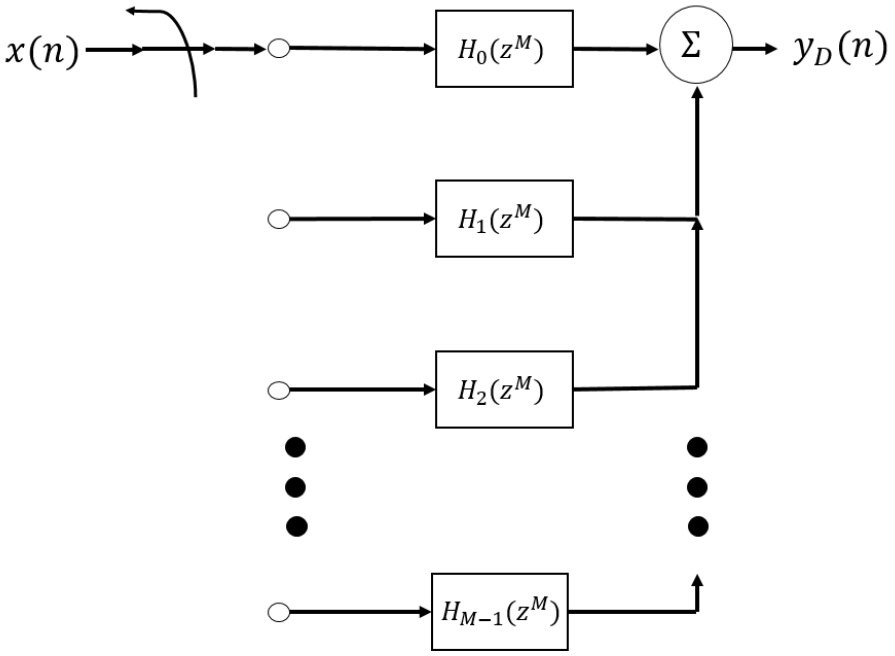
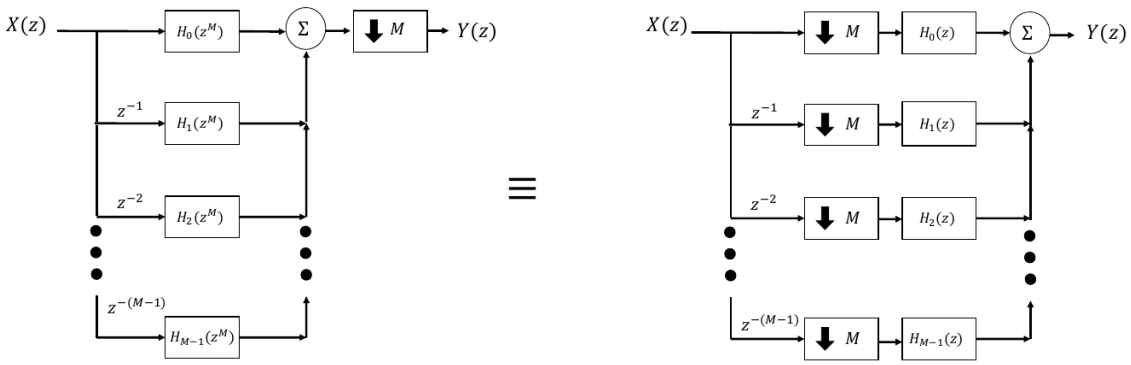


Figure 2-6 Polyphase implementation of a filter

Polyphase decompositions were first used for sample rate changes [8], and are without a doubt one of the greatest advances in multirate processing due to allowing for very efficient filter structures for decimation and interpolation by the use of the Noble identities.

Consider the decimator in Figure 2-3b. The decimation filter $H_D(z)$ is implemented using the polyphase structure as in Figure 2-7a; note that the number of polyphase components of $H_D(z)$ is the same as the decimation factor, which – as will become clear shortly – is a deliberate choice. By the commutative property between down-samplers and adders seen in Figure 2-5, as well as the Noble identities in Figure 2-4, an equivalent structure is arrived at and shown in Figure 2-7b below.



(a) Decimator using polyphase decomposition (b) Using commutative and Noble identities

Figure 2-7 Polyphase decimator identity

The down-samplers are now positioned before the filters, which means that the filters in Fig. 7b are operating at the *lower rate* as opposed to in Fig. 7a. This is an important result as lower rate operation means that there is more time available to operate on a signal, which translates into either better decimation filters or processing of larger signal bandwidth.

Down-samplers and delays in a signal flow graph are only interchangeble when the delays are multiples of M ; however, one last identity can be used to obtain the decimator structure in Figure 2-8. The commutator (or switch) at the input signal can also be seen as a serial-to-parallel conversion, which lowers the rate of operation of the system by a factor of M .

A similar analysis for the interpolator system can be done to obtain the efficient interpolator structure in Figure 2-9. Once again, the filters operate at the low rate, before the sampling rate increase by serialization of the branches.

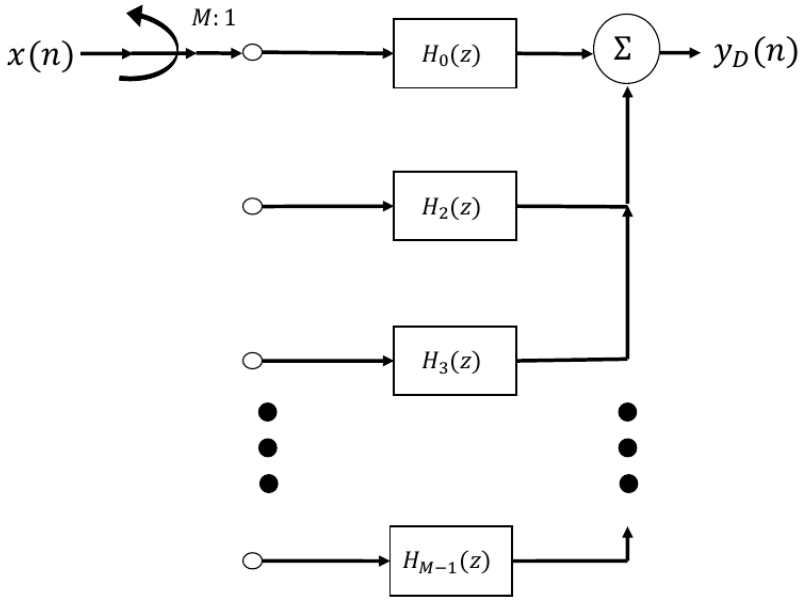


Figure 2-8 Efficient decimator

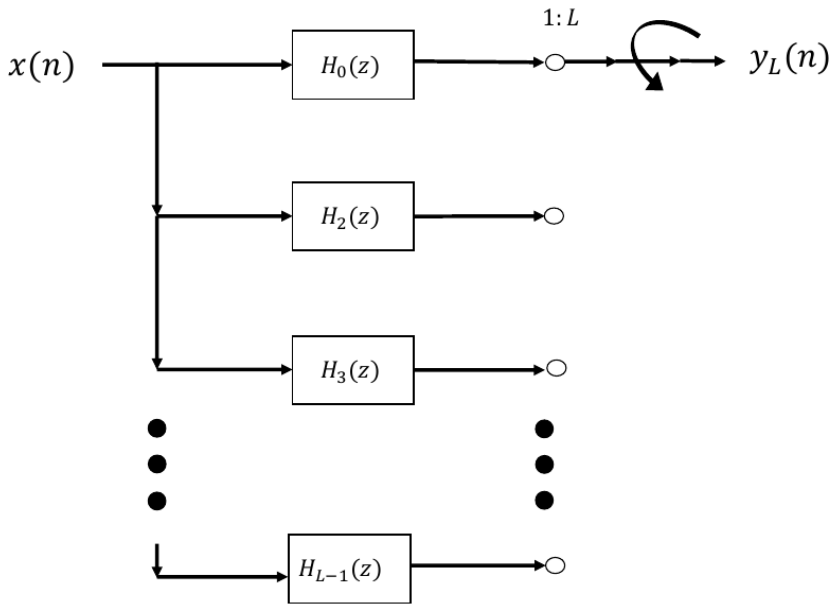


Figure 2-9 Efficient interpolator

Note that the decimator structure (Figure 2-8) and the interpolator structure (Figure 2-9) are transposed versions of each other, and together can be used to achieve arbitrary rate change in an efficient manner.

The efficiency of these structures comes from the length of each polyphase component $H_k(z)$ being $1/M$ times the length of $H(z)$, each polyphase component working on one input sample (a total of M samples) in a parallel manner.

2.2 Filter Bank Structures and Perfect Reconstruction

The fundamental multirate blocks, as well as efficient decimation and interpolation structures, are the ingredients for different filter bank structures. Conditions can then be defined that identify what is needed to achieve a perfect reconstruction filter bank (PRFB). Perfect reconstruction (PR) is a desirable property for many applications, such as speech coding [9, 10], image processing and compression [11], and digital communications [12].

The concepts above are reviewed in the next sections, starting with the DFT filter bank which will serve as an example to talk about the PR conditions. Other structures such as cosine modulated and non-maximally decimated filter banks (NMDFB) are also covered.

2.2.1 The DFT Filter Bank

A filter bank can be constructed using the M-point IDFT matrix defined in (15), where $(.)^*$ denotes complex conjugation.

$$\mathbf{W}_{r,c}^* = W_{r,c}^* = e^{j\frac{2\pi rc}{M}} \tag{15}$$

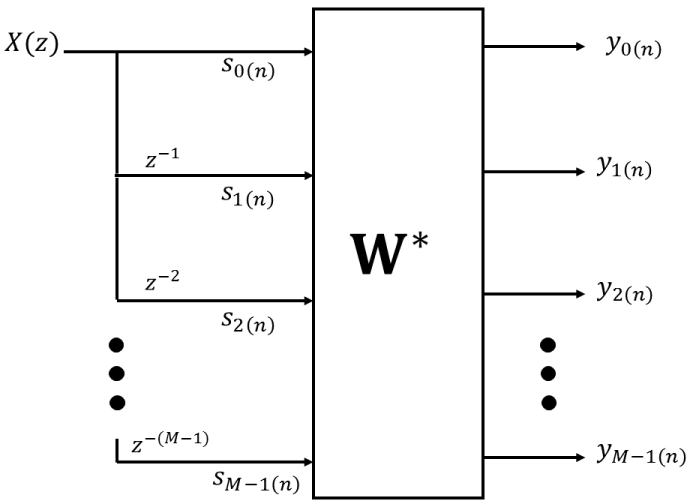


Figure 2-10 DFT Filter bank

Figure 2-10 shows the DFT *analysis* filter bank. The DFT input signals $s_k(n)$ can be written in terms of $x(n)$ as

$$s_k(n) = x(n-k) \tag{16}$$

Then by definition of the IDFT (ignoring the 1/*M* factor)

$$\begin{aligned} y_k(n) &= \sum_{i=0}^{M-1} s_i(n) e^{j\frac{2\pi ik}{M}} \\ &= \sum_{i=0}^{M-1} s_i(n) W^{-ik} \end{aligned} \tag{17}$$

Thus

$$y_k(n) = \sum_{i=0}^{M-1} x(n-i) W^{-ik} \tag{18}$$

Rewriting in terms of z

$$\begin{aligned} Y_k(z) &= \sum_{i=0}^{M-1} z^{-i} W^{-ik} X(z) \\ &= \sum_{i=0}^{M-1} \left(zW^{k}\right)^{-i} X(z) \end{aligned} \tag{19}$$

It is now possible to rewrite each output of the filter bank in the standard form of

$$Y_k(z) = X(z) H_k(z) \tag{20}$$

with

$$H_k(z) = H_0\left(zW^k\right) \tag{21}$$

where

$$H_0(z) = 1 + z^{-1} + z^{-2} + z^{-3} + ... + z^{-(M-1)} \tag{22}$$

From (21) and (22) note that the filter bank is formed by frequency shifted versions of $h_0(n)$, each one shifted by $\omega = 2\pi k / M$, with unit pulse response

$$h_k(n) = h_0(n) e^{-j\frac{2\pi nk}{M}} \tag{23}$$

Note that the frequency response of all $H_k(z)$ is completely determined by $H_0(z)$, and therefore $h_0(n)$ is referred to as the *prototype filter.*

For the DFT filter bank, the prototype filter has a rectangular shape in time given by

$$h_0(n) = \begin{cases} 1 & n = 0, 1, 2, ..., M-1 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

with magnitude response

$$\left|H_0(\omega)\right| = \left|\frac{\sin(M\omega/2)}{\sin(\omega/2)}\right| \tag{25}$$

Figure 2-11 shows the *analytical* magnitude response of $H_0(z)$ as expressed in (25), and the *observed* magnitude response of $Y_0(z) = \sum_{i=0}^{N} y_0(n) z^{-i}$, where $y_0(n)$ is the first output sequence, at the first branch of the DFT filter bank output as seen in Figure 2-10, with $x(n) = \delta(n)$. The size of the DFT was chosen as $M = 16$, and the frequency axis is shown as a normalized frequency

axis (where $\omega = 2\pi$ corresponds to a (normalized) sampling frequency of 1).

The magnitude responses of the output signals $Y_k(z)$ of the DFT filter bank are shown in Figure 2-12. Note that each filter is of length M, and that neighboring filters intersect at their -3dB points, which occur exactly at multiples of $\omega = \pi / M$.

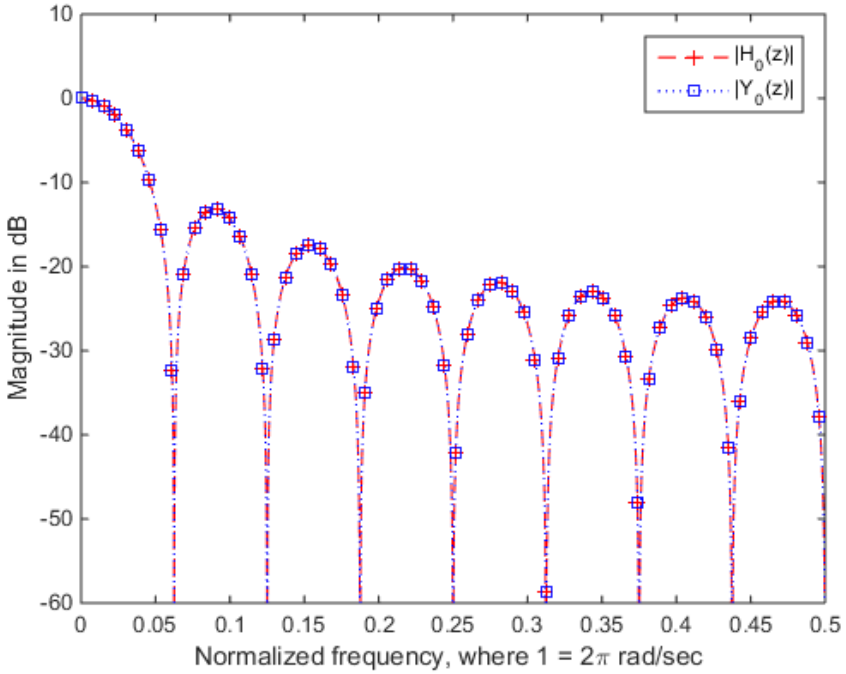


Figure 2-11 - DFT filter bank prototype filter magnitude response

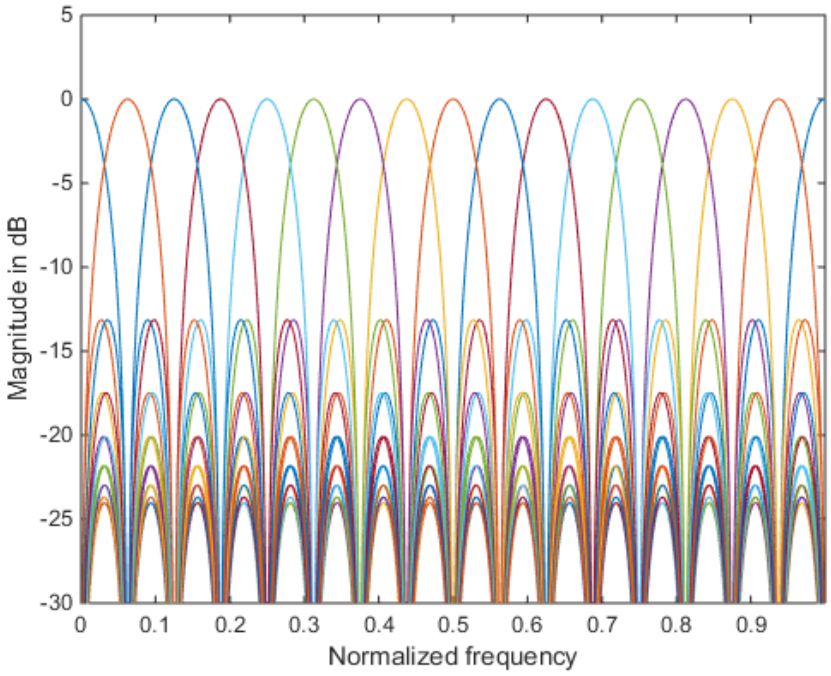


Figure 2-12 - DFT filter bank

From the figure above, and from (20), observe that the output of each branch is a sub-band of $X(z)$, each with a 3-dB bandwidth of $2\pi/M$. Considering that each sub-band is of complex nature, each one can be represented with a sampling rate equal to its bandwidth, that is, a sampling rate of $2\pi/M$ instead of 2π. To use this property, down-samplers are inserted in each output branch to obtain the DFT filter bank in Figure 2-13.

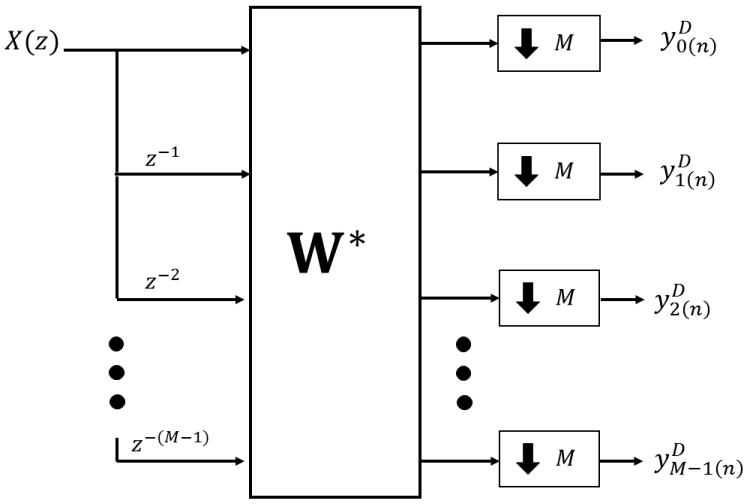


Figure 2-13 DFT filter bank with down-samplers (with *M*=*D*)

It was shown in the down-sampler analysis in Section 2.1.1 that down-samplers produce aliasing for non-bandlimited signals, such as will be the case here, where only the 3-dB bandwidth is being considered. However, the aliasing components in the signal $y_k^D(n)$ in Figure 2-13 can be cancelled because **the DFT filter bank is a perfect reconstruction filter bank**. Perfect reconstruction means that $x(n)$ can be perfectly reconstructed from $y_k^D(n)$ (for all k) using the *synthesis* DFT filter bank. Figure 2-14 shows the flat magnitude response of $Y_k^D(z)$ for all k. Note that with the DFT filter bank each sub-band of $X(z)$ can be maximally decimated without distortion.

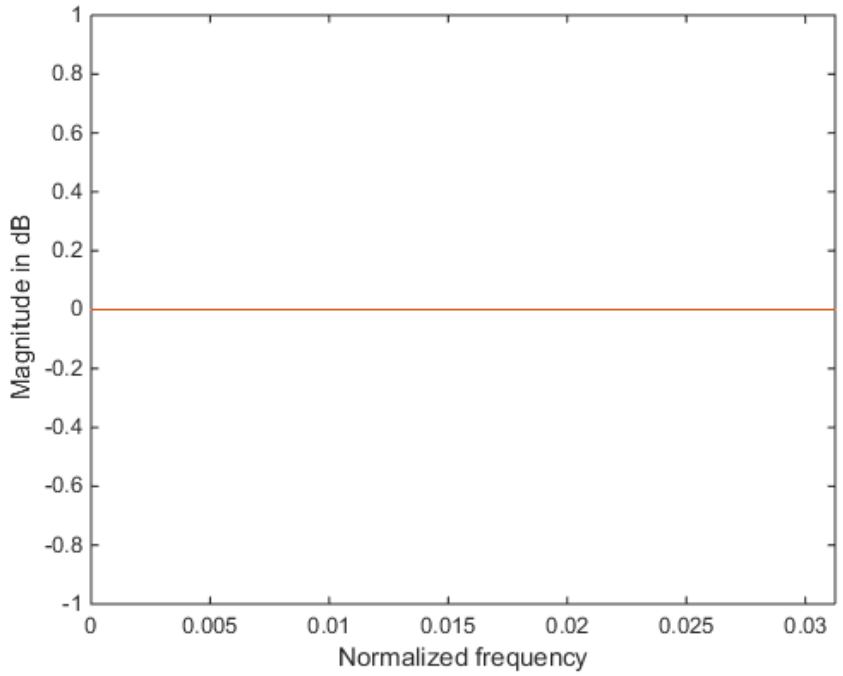


Figure 2-14 Maximally decimated DFT filter bank response

Conditions for perfect reconstruction will be given in the next section.

2.2.2 Perfect Reconstruction

Consider the cascade of analysis and synthesis banks in Figure 2-15, with analysis and synthesis *transfer matrices* $\mathbf{A}(z)$ and $\mathbf{S}(\mathbf{z})$ respectively. A filter bank is said to have the perfect reconstruction (PR) property [7] if

$$\hat{x}(n) = c\,x(n-d) \tag{26}$$

for a constant gain *c* and an integer delay *d*. That is, if the output of the cascaded system is a scaled and delayed version of the input.

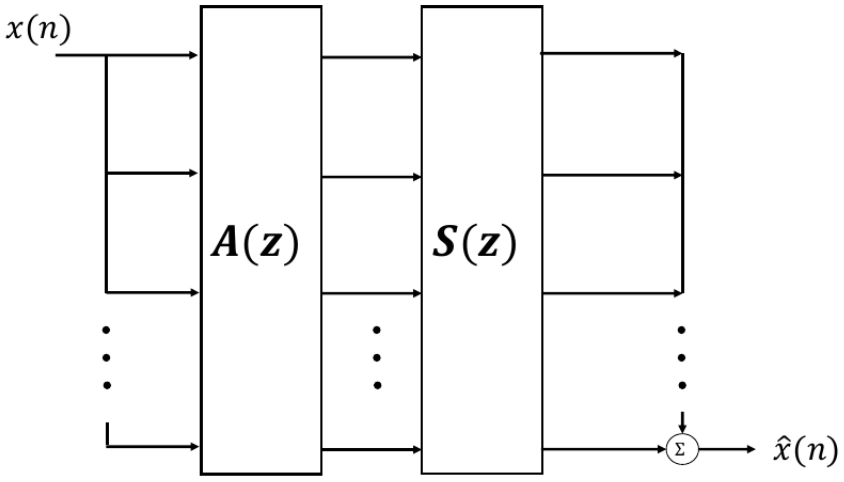


Figure 2-15 Cascade of analysis and synthesis filter bank

For the scaling and delay to happen, in the most general case, the condition in (27) must hold

$$\mathbf{S}(z)\mathbf{A}(z) = c\,z^{-d}\mathbf{I}_M \tag{27}$$

where $\mathbf{I}_M$ is the *M* x *M* identity matrix.

As an example, consider the simple perfect reconstruction filter bank (PRFB) shown in Figure 2-16. Using the expression obtained in (5) for the output of a down-sampler, $X_k^D(z)$ can be written as

$$X_k^D(z) = \frac{1}{M}\sum_{i=0}^{M-1}\left(z^{\frac{-k}{M}}W^{-i}\right)X\left(z^{\frac{1}{M}}W^{i}\right) \tag{28}$$

By (4) the output of the up-samplers will be

$$X_k^L(z) = \frac{1}{M}\sum_{i=0}^{M-1}\left(z^{-k}W^{-i}\right)X\left(zW^{i}\right) \tag{29}$$

The output of the filter bank is the (delayed) sum over *k* of (29), thus

$$\begin{aligned}\hat{X}(z) &= \frac{1}{M}\sum_{k=0}^{M-1} z^{-(M-1-k)} z^{-k} \sum_{i=0}^{M-1} W^{-i} X\left(zW^{i}\right) \\ &= \frac{1}{M} z^{-(M-1)} \sum_{i=0}^{M-1} X(z)\end{aligned} \tag{30}$$

$$\hat{X}(z) = z^{-(M-1)} X(z) \tag{31}$$

and in the time domain

$$\hat{x}(n) = x(n-M+1) \tag{32}$$

which is identical to (26) with $c=1$ and $d=M-1$. Note that if the up/down-samplers are taken away from the structure, leaving only the delays, the result is still a PRFB.

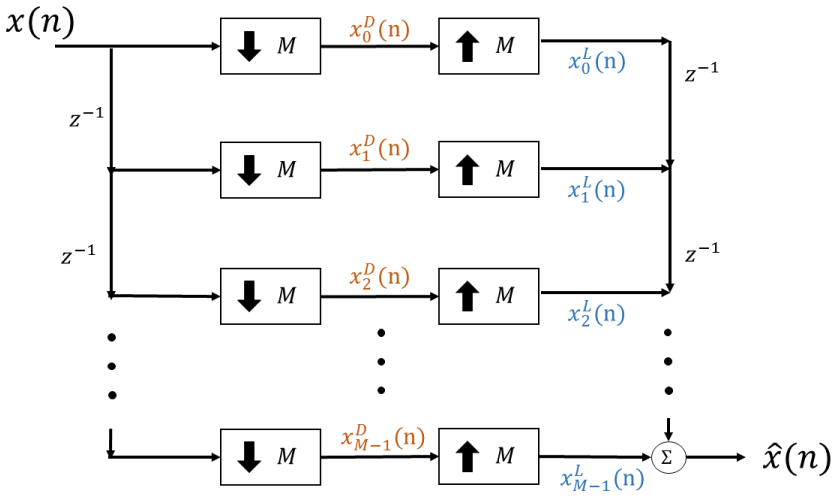


Figure 2-16 Simple PR filter bank

Now, let us insert an IDFT matrix before the down-samplers, and a DFT matrix after the up-samplers, as seen in Figure 2-17. Note that the analysis portion of the filter bank is exactly the same as the one depicted in Figure 2-13.

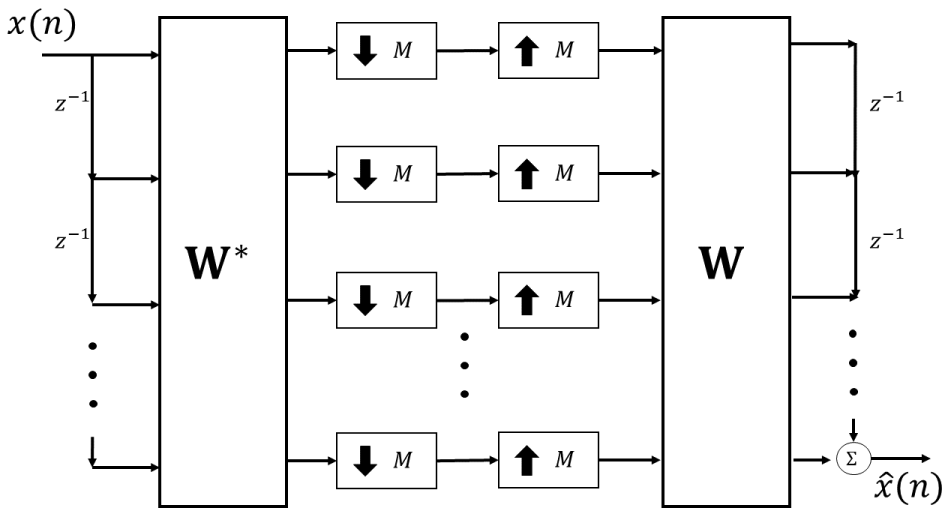


Figure 2-17 DFT cascaded filter bank

The IDFT and DFT matrices are constant matrices, they do not change with z or n. Because of this, it is possible to invoke the commutative properties between up/down-samplers and memoryless operators to obtain the structure in Figure 2-18.

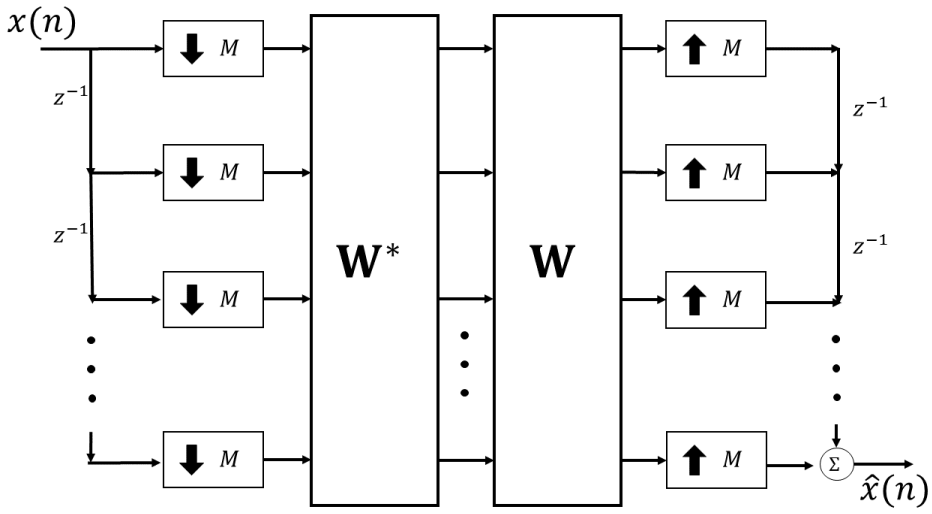


Figure 2-18 DFT filter bank with decimators in front

It is well known that applying an M-point IDFT followed by an M-point DFT results in an identity matrix.

$$\mathbf{W}_M \mathbf{W}_M^* = \frac{1}{M}\mathbf{I}_M \tag{33}$$

Thus, the cascade of analysis and synthesis filter bank in Figure 2-18 is still a PRFB.

As seen in Section 2.2.1, the DFT analysis filter bank results in equally spaced filters that do not offer very good stopband rejection. It is often desired to have better analysis/synthesis filters that still achieve perfect reconstruction. However; the DFT filter bank gives a good introduction to the concept of **paraunitary matrices**.

A matrix is said to be *paraunitary* when its inverse is the *paraconjugated* version of itself, that is [2]

$$\mathbf{H}(z)\mathbf{H}(z) = c\,\mathbf{I} \tag{34}$$

where c is a non-zero constant, and the operator $(.)$ colloquially means "conjugation of coefficients and replacement of z by z^{-1}." Formally

$$\mathbf{H}(z) = \mathbf{H}^{*}\left(z^{-1}\right) \tag{35}$$

The relationship between the IDFT and DFT matrices in (33) may seem like an obvious result, but is in fact the same as (34) with $\mathbf{H}(z) = \mathbf{W}_M$, $\mathbf{H}(z) = \mathbf{W}_M^{*}$, and $c = 1/M$. This means that the DFT matrix is a paraunitary matrix.

2.2.3 Condition for Perfect Reconstruction *M*-Channel Cascaded Filter Bank

Consider the filter bank structure in Figure 2-19, with analysis and synthesis transfer matrices $\mathbf{E}(z)$ and $\mathbf{G}(z)$ respectively. If $\mathbf{E}(z)$ is constrained to be paraunitary, then perfect reconstruction results [13] from choosing

$$\mathbf{G}(z) = \beta z^{-N}\mathbf{E}(z) \tag{36}$$

where β is an arbitrary constant different from zero, and z^{-N} is a delay that ensures the causality of $\mathbf{G}(z)$.

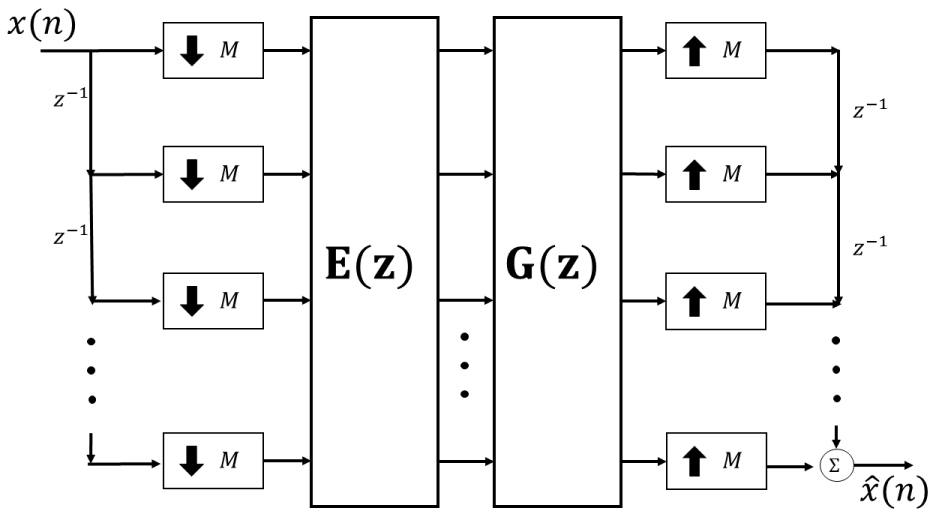


Figure 2-19 Filter bank illustrating the PR condition

Considering only M x 1 (paraunitary) transfer matrices (meaning that only one transfer function operates on each branch), $\mathbf{E}(z)$ and $\mathbf{G}(z)$ can be decomposed as

$$\mathbf{E}(z) = \begin{bmatrix} H_0(z) & H_1(z) & \dots & H_{M-1}(z) \end{bmatrix}^T \tag{37}$$

$$\mathbf{G}(z) = \begin{bmatrix} G_0(z) & G_1(z) & \dots & G_{M-1}(z) \end{bmatrix}^T \tag{38}$$

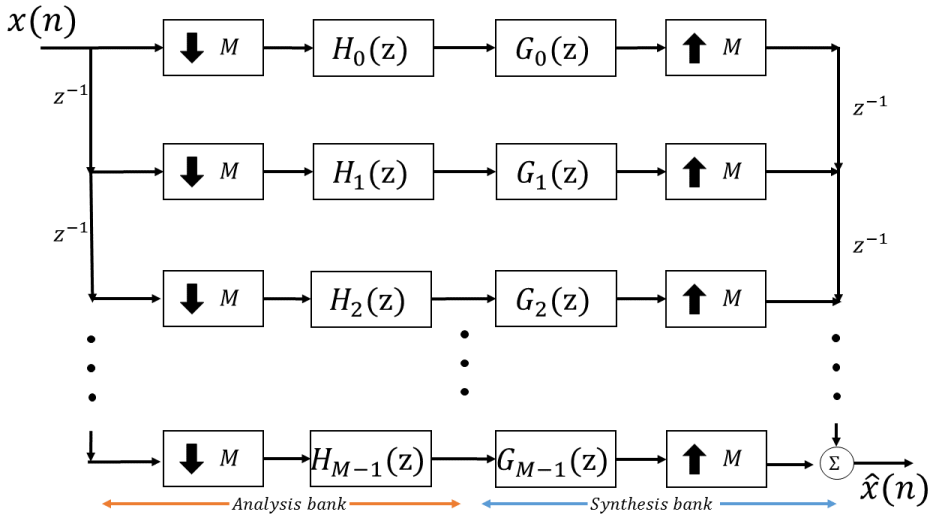


Figure 2-20 *M*x1 Transfer matrix cascaded filter bank

Some properties of paraunitary matrices include:

1. If $\mathbf{E}(z)$ is paraunitary, then so is $\mathbf{E}(z^M)$. This property is important in cases where it is desired to interchange the order of operation of up/down-samplers with the analysis and synthesis matrices, similar to what was done in the DFT filter bank.

2. The cascade of two paraunitary systems results in a paraunitary system. An example of this property is seen from the insertion of the (paraunitary) IDFT/DFT matrices in the simple PRFB of Figure 2-16.

3. If $\mathbf{E}(z)$ is an *M* x 1 paraunitary matrix as defined in (37), then this implies the power complementary property of the analysis filters (and vice-versa):

$$\sum_{i=0}^{M-1}\left|H_k\left(e^{j\omega}\right)\right|^2 = c \tag{39}$$

   The latter property relates perfect reconstruction to the power symmetric property and Nyquist *M*-band filters discussed in the next section.

4. By the choice of synthesis filters in (36), $H_k(z)$ and $G_k(z)$ have the same magnitude response for all *k*. Also, (36) implies

$$g_k(n) = \beta h_k^*(N-n), \tag{40}$$

   with *N* usually being the length of the $h_k(n)$ filters.

All of these properties and relationships are seen in the DFT bank example.
In the next sections the PRFB structures that are relevant to this work are briefly reviewed and related to communication systems and PR conditions.

2.2.4 QMF Banks

The quadrature mirror filter (QMF) bank [1, 14] is the special case when *M*, the number of channels in the filter bank, is 2, that is, it is desired to separate a signal into two sub-bands and then perfectly

reconstruct the input signal.

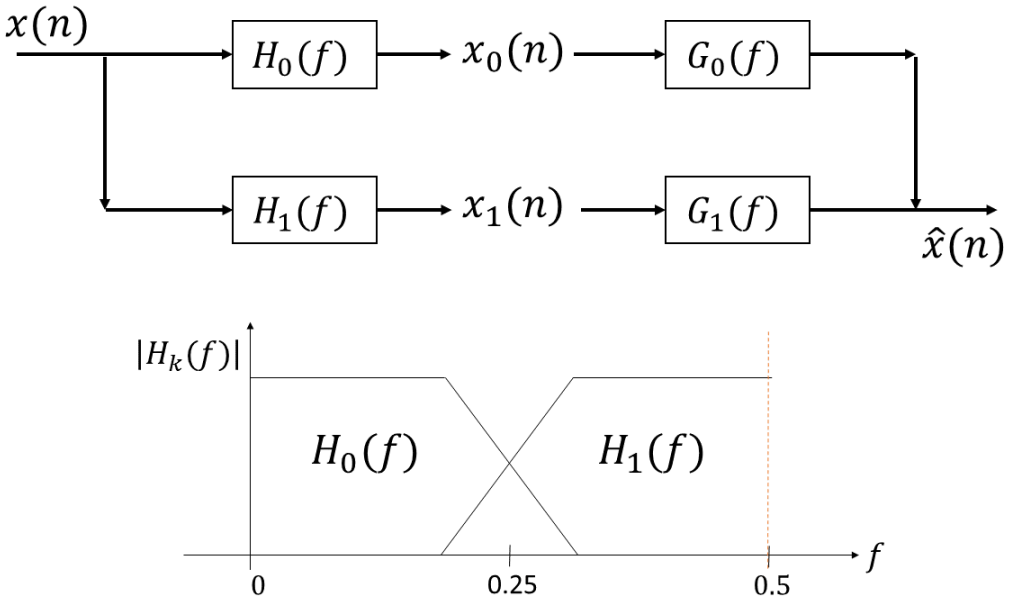


Figure 2-21 QMF bank structure and analysis filter magnitude responses

Figure 2-21 shows the QMF bank structure and a representation of the magnitude responses of the analysis filters.

It is possible to obtain a perfect reconstruction QMF bank (PRQMFB) provided that $H_0(z)$ is constrained to have the *power symmetric property* defined as

$$H_0(z)H_0(z)+H_0(-z)H_0(-z)=1 \tag{41}$$

which is achieved when $H_0(z)H_0(z)$ is a *half-band filter* (also called Nyquist filter, or Nyquist *M*-band filter with $M=2$). This property, also called the Nyquist criterion for zero inter-symbol interference (ISI) in digital communications [15], is defined for the general case for a filter $h_{Ny}(n)$ in the time domain as

$$h_{Ny}(nM)=\begin{cases} c & n=0 \\ 0 & n\neq 0 \end{cases} \tag{42}$$

That is, $h_{Ny}(n)$ is zero at all samples that are multiples of M except for $n=0$. In frequency this condition is manifested as

$$\sum_{k=0}^{M-1} H_{Ny}\left(zW^{k}\right) = c \tag{43}$$

which implies that the sum of M shifted versions of $H_{Ny}(z)$ by $2\pi k / M$ result in a flat frequency response. Note that (43) is exactly equal to the PRQMFB condition in (41) if $H_0(z)H_0(z)$ results in a half-band/Nyquist filter with $c = 1$.

Perfect reconstruction conditions dictate that the M x 1 transfer matrix $\left[H_0(z) \quad H_1(z)\right]^T$ is paraunitary, which – by properties of paraunitary matrices – implies that $H_0(z)$ and $H_1(z)$ are power complementary as expressed in (39). For the QMF bank this means

$$H_0(z)H_1(z) + H_0(-z)H_1(-z) = 1 \tag{44}$$

Relating (44) to the PRQMF condition observe that

$$H_1(z) = H_0(z) = H_0^*\left(z^{-1}\right) \tag{45}$$

The above suggests that $H_1(z)$ is the high-pass image of $H_0(z)$, symmetric about $f = 1/4$ (or $\omega = \pi/2$).

Invoking the PR property induced by the paraunitary property once more, shows that the choice of synthesis filters comes directly from the analysis filters (36). Then

$$G_0(z) = z^{-N} H_0(z) \tag{46}$$
$$G_1(z) = z^{-N} H_1(z) \tag{47}$$

For a real-coefficient filter $h_0(n)$ of length N, this simply means

$$h_1(n) = (-1)^n h_0(n) \tag{48}$$
$$g_0(n) = h_0(N-n) \tag{49}$$
$$g_1(n) = h_1(N-n) \tag{50}$$

and, provided $h_0(n)$ is also symmetric/anti-symmetric, replacement of $(N-n)$ by (n) is possible.

In digital communications the use of a shaping filter is often desired in order to reduce the bandwidth of the transmission. This *transmitter shaping filter* must meet the Nyquist zero ISI criterion when cascaded with a *receiver matched filter*. This ensures both zero ISI and maximum signal-to-noise (SNR) ratio at the sampling instants of the received symbol after being sent over a noisy channel. For an additive white Gaussian noise (AWGN) channel the matched filter at the receiver is simply a 'flipped' version of the transmitter filter. The transmitter-receiver filters are often designed to be *square root Nyquist* (srN) filters, with the square-root-raised cosine (SRRC) filter being the most popular shaping/matched filter [15]. Going back to the QMF bank, $h_0(n)$ and $h_1(n)$ can be thought of as two transmitter filters and $g_0(n)$, $g_1(n)$ as the matched/receiver filters for the two channels.

2.2.5 Cosine Modulated Filter Banks

The results obtained for the QMF bank can be extended to *M*-channel filter banks thanks to the relationships between power symmetry, paraunitary, and PR properties.

In many applications it is desired to have filters with a linear phase response. In audio/speech applications for example, phase response is not very important as humans do not perceive phase changes as well as magnitude changes. However, in digital communications the phase response of the transmit/receive filters is very important if the symbols are to be received with no errors.

Cosine modulated filter banks (CMFB) offer a way to design linear-phase FIR analysis and synthesis filters that have the PR property from a single prototype filter $p(n)$. Similar to the QMF bank, the analysis and synthesis filters will be shifted versions of the prototype filter. This shifting can be done with a simple cosine modulation operation. The analysis and synthesis filters for a CMFB are expressed in (51) and (52).

$$h_k(n) = 2p(n)\cos\left(\frac{\pi}{M}(k+0.5)\left(n-\frac{N}{2}\right)+\theta_k\right) \tag{51}$$

$$g_k(n) = 2p(n)\cos\left(\frac{\pi}{M}(k+0.5)\left(n-\frac{N}{2}\right)-\theta_k\right) \tag{52}$$

$$\theta_k = (-1)^k\left(\frac{\pi}{4}\right), \quad 0 \le k \le M \tag{53}$$

These expressions were derived to minimize aliasing and phase distortion (requirements for perfect reconstruction) at each branch of the filter bank, achieving perfect reconstruction under the condition that

$$P_k(z)P_k(z) + P_{k+M}(z)P_{k+M}(z) = c \tag{54}$$

where $P_k(z)$ are the 2*M* polyphase components of $P(z)$, and $P(z)$ is the z-transform of the prototype filter $p(n)$. In this case the prototype filter should be designed as a 2*M*-Nyquist filter. Also *2M* polyphase components are needed (see Figure 2-22) if having *M* channels is the goal. This factor of 2 comes from the analysis and synthesis filter having real coefficients, each one formed by two complex conjugate pairs.

The CMFB can be implemented efficiently using the $2M \times M$ cosine modulation matrix defined as

$$\mathbf{C}_{k.n} = 2\cos\left(\frac{\pi}{M}(k+0.5)\left(n-\frac{N}{2}\right)+\theta_k\right) \tag{55}$$

For simplicity only the analysis filters will be considered as the synthesis portion is obtained by transposition of the analysis filter bank.

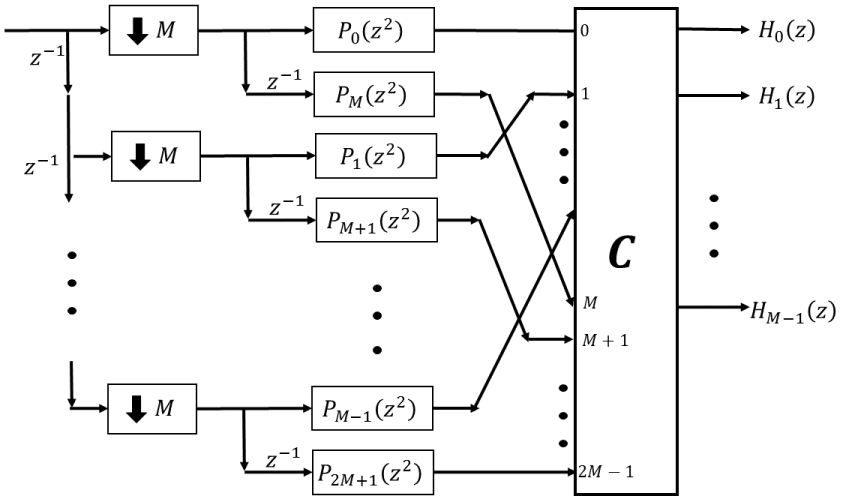


Figure 2-22 Efficient implementation of CMFB

Detailed derivation and implementation procedures of the CMFB can be found elsewhere [4, 16].

2.2.6 Exponential Modulated Filter Banks

It is also possible to obtain the analysis filters of an exponential modulated filter bank (EMFB) by

$$h_k(n) = p(n)e^{j\frac{2\pi nk}{M}}, \quad 0 \le k \le M-1 \tag{56}$$

By defining $H_k(z) = \sum_{n=-\infty}^{\infty} h_k(n) z^{-n}$ as the z-transform of $h_k(n)$, the analysis filters can be expressed in the complex frequency domain as

$$H_k(z) = P\left(zW^k\right) \tag{57}$$

$$H_k(\omega) = P\left(e^{j\left(\omega - \frac{2\pi k}{M}\right)}\right) \tag{58}$$

The analysis filters are versions of the prototype filter shifted in frequency by $\omega = 2\pi / M$, intersecting one another around their -3dB point at $\omega = pi / M$. All of the filters conserve the power symmetric property, and the analysis transfer matrix has the power complementary property of (39), which implies the paraunitary and perfect reconstruction properties. This result was seen in both the DFT and QMF banks. The analysis EMFB is shown in Figure 2-23.

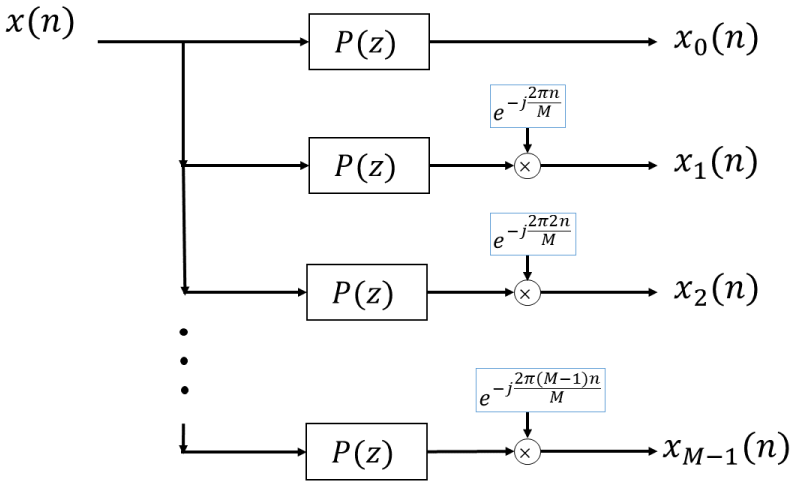


Figure 2-23 Exponential modulated filter bank

The outputs $x_k(n)$ can be written in terms of z as

$$X_k(z) = X(z)P(zW^k) \tag{59}$$

Writing $P_0(z)$ in its polyphase representation

$$P(z) = \sum_{i=0}^{M-1} z^{-i} P_i(z^M) \tag{60}$$

and substituting into (59) yields

$$X_k(z) = X(z)\sum_{i=0}^{M-1}(zW^k)^{-i} P_k(z^M W^{ikM}) \tag{61}$$

$$X_k(z) = X(z)\sum_{i=0}^{M-1} z^{-i} P_i(z^M) W^{-ik} \tag{62}$$

The expression (62) suggests that it is possible to implement the EMFB by means of the IDFT matrix operating on the outputs of the polyphase components of the prototype filter, as shown in Figure 2-24.

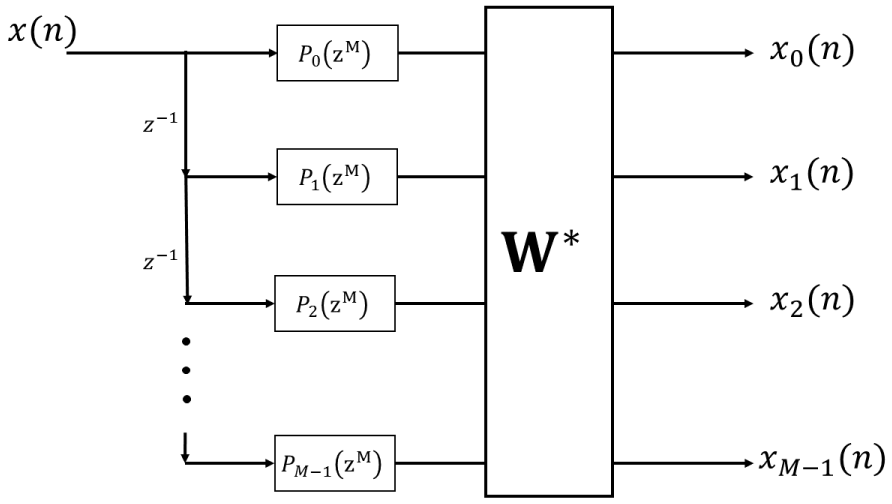


Figure 2-24 Exponential modulated filter bank with IDFT

Under the argument that signals $x_k(n)$ can be represented with a lower sample rate the IDFT output branches are down-sampled, as done for the DFT filter bank. Then, by using the Noble identities it is possible to obtain the structure in Figure 2-25. Note that this structure is much more efficient than the direct modulation in Figure 2-23 as the length of $P_k(z)$ is $1/M$ the length of $P_0(z)$, and the filters operate on M samples of $x(n)$ in parallel at $1/M$ of the rate.

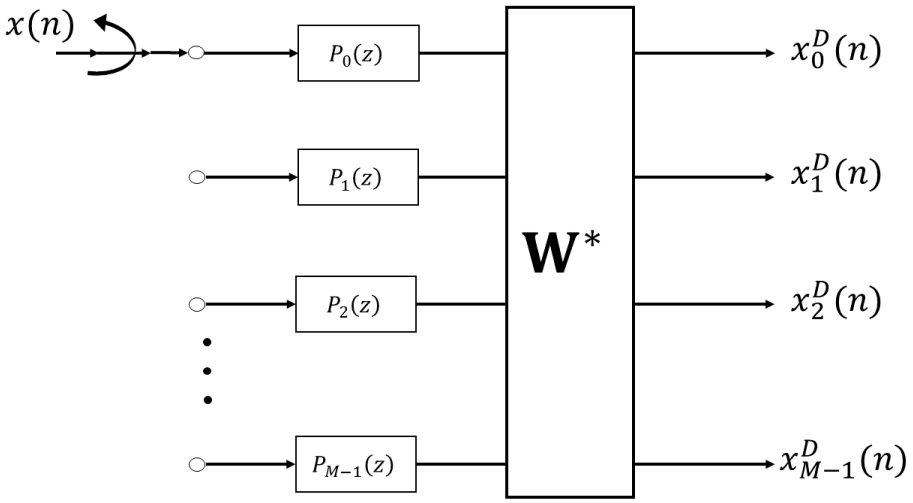


Figure 2-25 Efficient EMFB

While this structure is very efficient, **the PR property was lost** the moment $P_0(z)$ was decomposed into its polyphase components. Or rather, the conditions for PR now also include the

polyphase components of the prototype filter, $P_k(z)$, as well as the resulting filter bank modulated filters $H_k(z)$. In particular, the conditions for PR are now the same as for the CMFB, which are as shown in (54).

The PR property can be regained by modifying the EMFB into the modified DFT (MDFT) filter bank [17, 18] structure shown in Figure 2-26. The structure is obtained by decimating the incoming signal by $M/2$, followed by a second step decimation by 2, with and without a delay of z^{-1}. The alias spectra is cancelled with the MDFT bank.

Note that the MDFT comes from the combination of two EMFB, with a delay of $z^{-M/2}$ from one another.

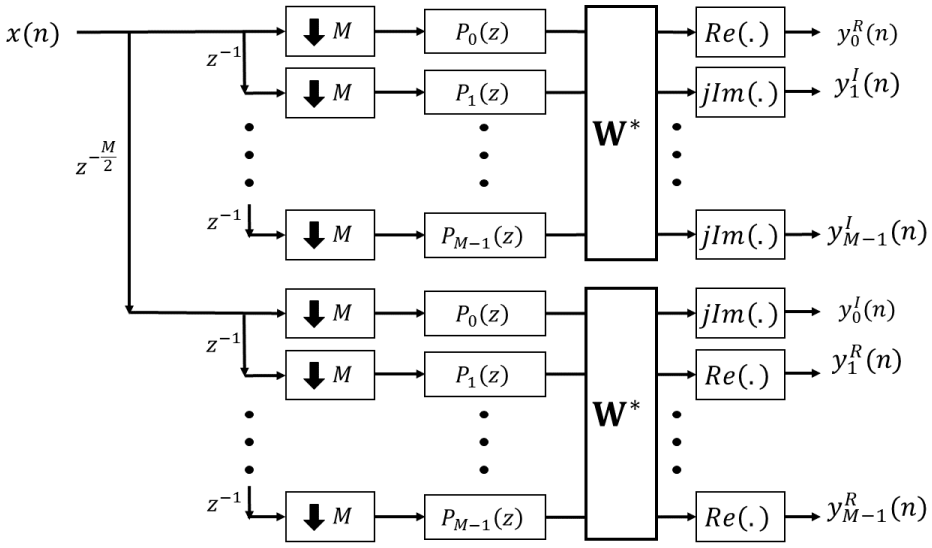


Figure 2-26 MDFT structure

The two *M*-point DFT operations in essence separate the in-phase and quadrature components of $x(n)$. If $x(n)$ and the prototype filter are real signals then half of the channels can be dropped since $\pm y_k^R(n) = \pm y_{M-k}^R(n)$ and $\pm y_k^I(n) = \mp y_{M-k}^I(n)$, or used for a second real signal.

2.2.7 Non-Maximally-Decimated DFT Filter Bank

The MDFT structure can be modified further to yield a form similar to the CMFB. This is shown in Figure 2-27.

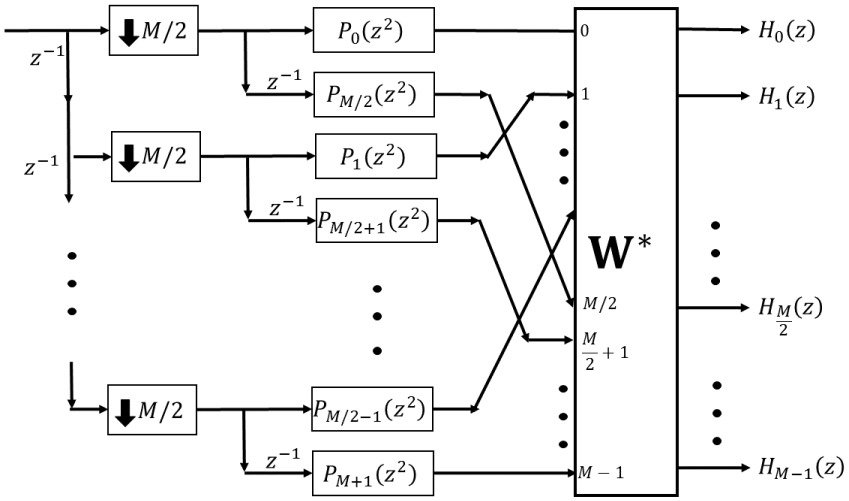


Figure 2-27 Efficient MDFT filter bank

This efficient structure [19] was later named a non-maximally decimated filter bank (NMDFB) [20] because of the *M*/2-factor instead of *M*-factor down-samplers. Note that the structure follows directly from down-sampling the output signals of the EMFB in Figure 2-24 by *M*/2, and therefore still equivalent to the MDFT in terms of perfect reconstruction.

Figure 2-28 shows the $H_k(z)$ outputs of the NMDFB (when a Dirac impulse $\delta(n)$ is applied as input). The non-maximal decimation makes the information of each decimated sub-band (aliased to baseband because of the down-samplers) reside in the range of $0 \le \omega \le \pi/2M$ in even-indexed channels, and $\pi/2M \le \omega \le \pi/M$ in odd-indexed channels, instead of in the complete decimated range of $0 \le \omega \le \pi/M$ seen with *M*-fold decimation (see the decimated output filters of the DFT bank in Figure 2-14). The prototype filter is an *M*-band square-root Nyquist filter with *M* = 16.

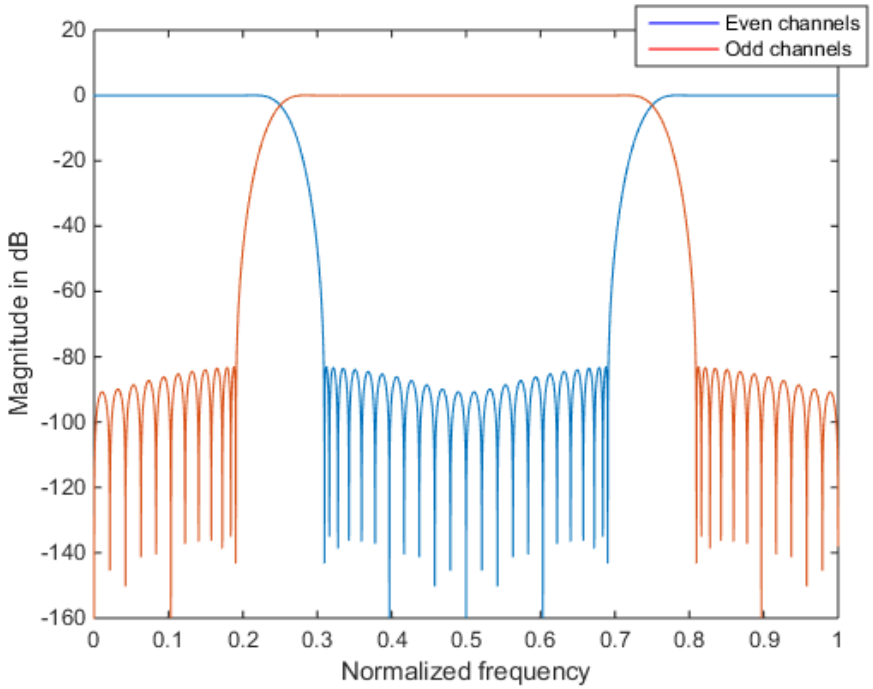


Figure 2-28 Non-maximally decimated FB filters after decimation

The NMDFB has been further modified [20] to "mirror" the frequency response of the odd-indexed channels by using a circular buffer before the IDFT, which is equivalent to alternating the signs of the odd samples coming out of the odd-indexed channels of the bank. The structure was also rearranged, and a commutator was added to take the place of the delays and down-samplers. The resulting structure is shown in Figure 2-29. The commutator distributes every sample to the top and bottom halves of the bank, and the circular buffer interchanges the top and bottom $M/2$ odd samples. The bottom half filters have a one sample delay, which can be absorbed into the filters instead. This is the structure referred to as h-NMDFB in future sections.

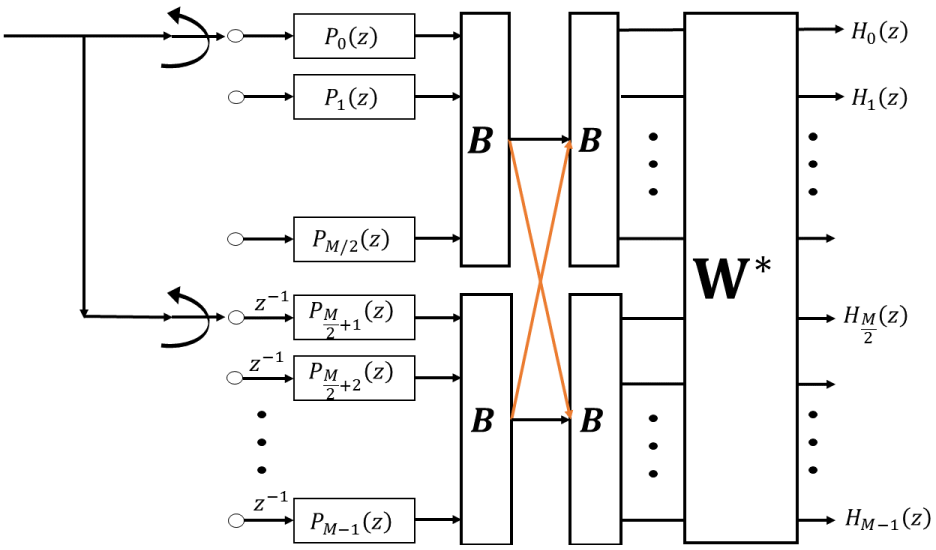


Figure 2-29 NMDBFB with circular buffer and commutator

Non-maximal decimation allows for a very interesting choice of analysis and synthesis filters as the decimated filters do not occupy all of the low-frequency range. With the h-NMDFB it is possible to achieve very close to perfect reconstruction even when the analysis filter matrix does not have the paraunitary property. This happens with the condition that the multiplication of analysis and synthesis matrices still results in an identity matrix. That is

$$\mathbf{E}(z)\mathbf{G}(z) = z^{-n_0}\mathbf{I} \tag{63}$$

This can be achieved by making the cascade of the k^{th} analysis and synthesis filter in the bank be Nyquist *M*-band filters.

$$H_k(z)G_k(z) = H_{Ny}(z) \tag{64}$$

This condition is also present under the paraunitary constraint, but the h-NMDFB allows for designing the filters such that the analysis filters $H_k(z)$ are Nyquist *M*-band filters, and the synthesis filters $G_k(z)$ have a wider passband so that (64) still holds. This of course translates into letting the prototype filter $P_k(z)$ be a Nyquist filter itself. The filter bank design procedure then goes as follows [21]:

Design a Nyquist *M*-band prototype filter for the analysis bank. This filter can simply be a sinc function of the form

$$p^A(n) = \frac{\sin(\pi n / M)}{\pi n} v(n), \quad -\frac{\gamma M}{2} \le n \le \frac{\gamma M}{2} \tag{65}$$

where γ will be the number of coefficients in each polyphase component of $P^A(z)$, and $v(n)$ is a Kaiser shaping window [22] with the Kaiser window parameter $\beta = 8.9$ [21]. The filter will have its -6dB point at the normalized frequency of $f = 1/(2M)$ at the rate of *M*.

Now, for the synthesis filter prototype $p^S(n)$ there is not a unique option. As long as the prototype has a passband of at least $1/(2M)$, and does not destructively alias on itself **over this range** when decimated, then the cascade of both filters will meet the condition in (64).

For example, the choice of a Nyquist *M*/2-band meets these requirements, as by definition it will have a flat passband over the *M*/2-decimated frequency range. Then:

$$p^S(n) = \frac{\sin(2\pi n / M)}{\pi n} v(n), \quad -\frac{\gamma M}{2} \le n \le \frac{\gamma M}{2} \tag{66}$$

Figure 2-30 shows both prototype filters at a rate of *M*, and Figure 2-31 illustrates the *M*/2-decimated analysis and synthesis filters. For both filters $\gamma = 12$ and $M = 64$.

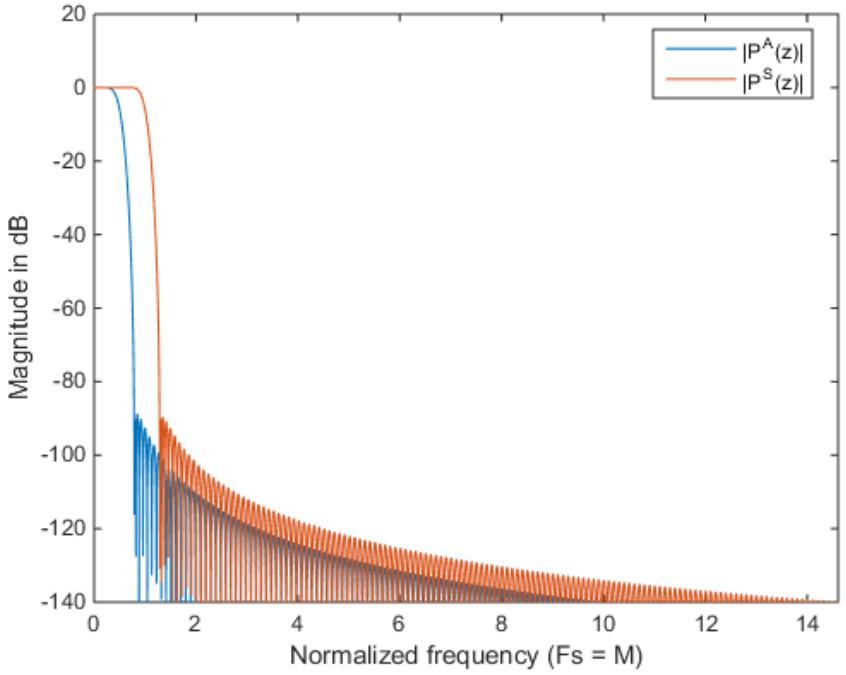


Figure 2-30 Analysis and synthesis prototype filters

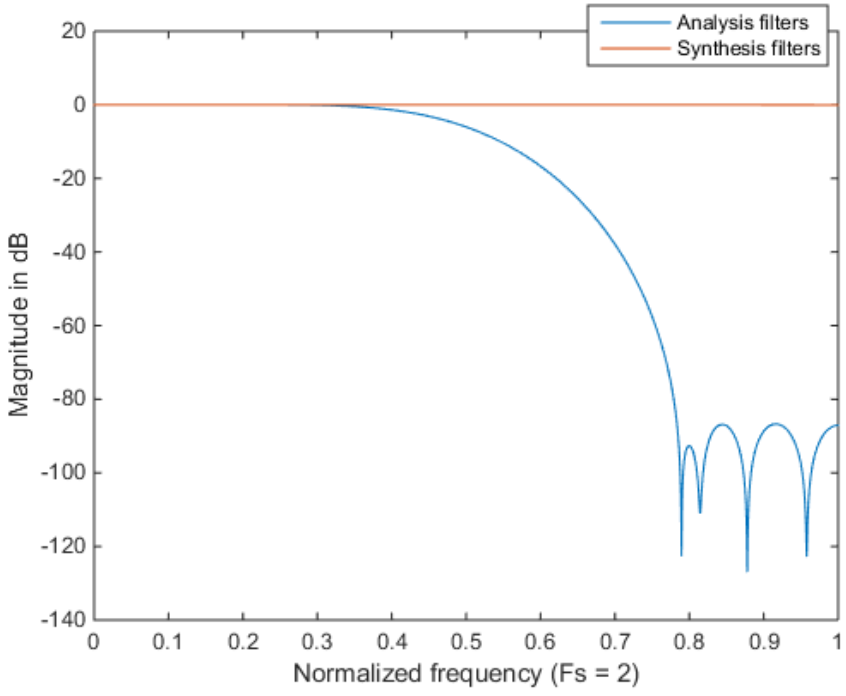


Figure 2-31 Aliased-to-baseband analysis and synthesis filters

The h-NMDFB can be used with either square-root Nyquist or Nyquist prototype filters. The first system meets the paraunitary property and the latter does not, and depending on the application one may be more favorable than the other. The choice of filters above violates the paraunitary

property, but the aliasing distortion can be made arbitrarily small by having filters with good stop-band attenuation. However, in a setting where noise is introduced between analysis and synthesis banks it may be more favorable to use square-root Nyquist filters in order to reject the out-of-band noise, meeting the PR property at the same time.

In terms of computational efficiency, let N_{FIR} be the length of the analysis/synthesis prototype filter. Assuming complex inputs and real coefficients, $(2)2N_{FIR}/M$ multiply-add (MAC) operations are required for one sample to propagate through the polyphase components. The *M*-point IDFT takes roughly $2M\log_2 M$ MACs by the FFT algorithm [23], which is $4\log_2 M$ operations per sample at the rate of 2. In total the analysis portion of the h-NMDFB would take $4\left(\frac{N_{FIR}}{M}+\log_2 M\right)$ MACs per complex sample operating at the low rate.

For the example above this is $4(12+5)=68$ MAC operations, or 136 for both the analysis and synthesis banks.

2.3 Interpolated FIR Filters

Interpolated finite impulse response (IFIR) filters [24] are an alternative to design narrowband filters efficiently. IFIR filters take advantage of the frequency compression of a signal (in this case of a filter) due to an up-sampler, as seen in Figure 2-2a.

An IFIR system is in fact the cascade of two FIR filters, an up-sampled *prototype* filter $G\left(z^L\right)$ with interpolation factor *L* (not to be confused with the prototype filter of a filter bank) and an *image reject* (IR) filter $H_{IR}(z)$. The IFIR system is depicted in Figure 2-32.

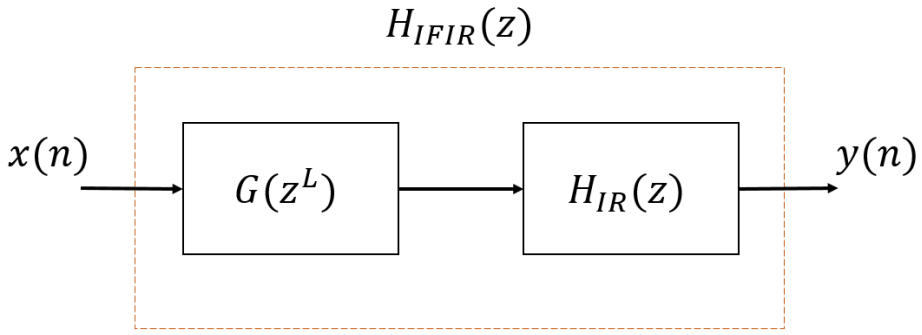


Figure 2-32 IFIR system

The design parameters for the design of a low-pass IFIR filter are shown in Table 2-1 for a set of filter specifications $\left(f_p, f_s, \delta_p, \delta_s\right)$ of passband and stopband frequencies (in normalized frequency units), and passband and stopband ripple in dB.

Table 2-1 Design parameters for IFIR filters

Specification frequency cutoffs	Frequency cutoffs for $G(z)$	Frequency cutoffs for $H_{IR}(z)$
f_p	$L\, f_p$	f_p
f_s	$L\, f_s$	$\frac{1}{L} - f_s$

For a given set of specifications the lengths of both filters, and of the IFIR system overall, will depend on the interpolation factor *L* only. This interpolation factor can be chosen to reduce the overall filter length [25] by optimization of the Kaiser filter order estimator [26]. The expression for the optimal *L* is given by

$$L_{opt} = \left\lceil \frac{1}{f_p + f_s + \sqrt{f_s - f_p}} \right\rceil \tag{67}$$

Figure 2-33 graphically shows how the IFIR system design works for a filter specification set $\left(0.0125, 0.0156, \pm 0.1313, -100\right)$. This is an *M*-band square-root Nyquist filter for *M* =32, designed with the PM algorithm. The optimum interpolation factor calculated from (67) yields

$L_{opt} = 12$.

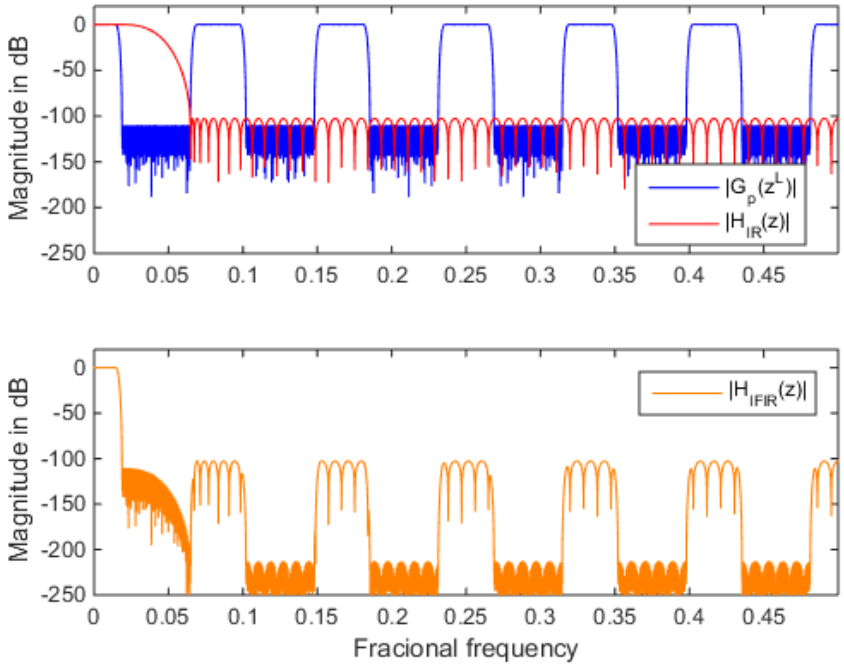


Figure 2-33 IFIR filters

The IFIR design is an efficient design in terms of the number of coefficients of the overall system, which translates to fewer multiply-adds in its implementation when compared to a common narrowband FIR. Table 2-1 shows the filter lengths (N_x) of both the prototype and the IR filters, as well as its sum, and the filter length of the direct FIR design seen in Figure 2-34. A 5-fold reduction in terms of the required number of multiplications per sample is realized in the IFIR design, relative to the direct FIR design for the implementation of a narrowband filter with the same specifications.

Table 2-2 Filter lengths

N_p	N_{IR}	$N_{IFIR} = N_p + N_{IR}$	N_{FIR}	$r = N_{FIR} / N_{IFIR}$
103	111	214	1147	5.35

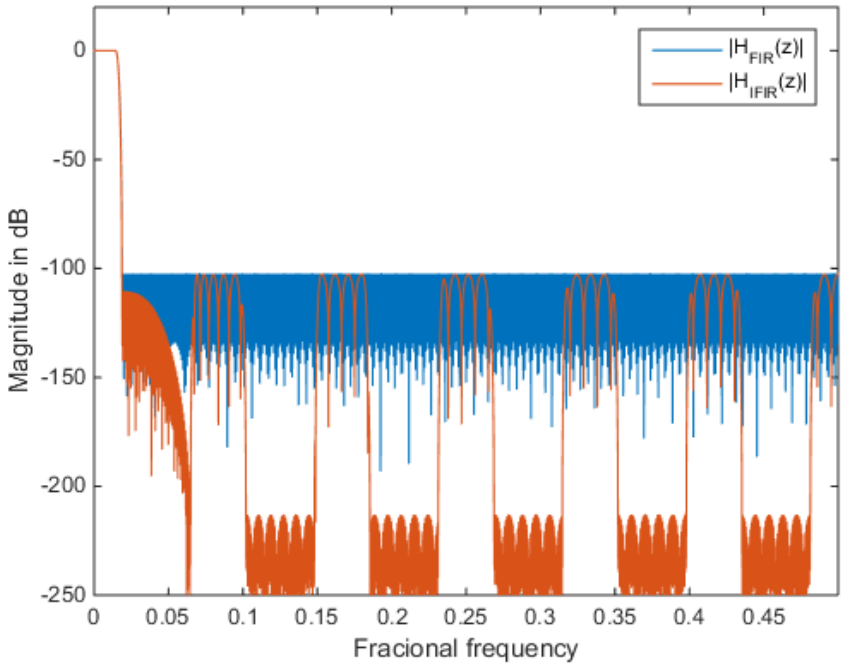


Figure 2-34 Comparison of FIR and IFIR filters

Note that **the IFIR implementation in Figure 2-32 <u>is not</u> a multirate system**. While the design procedure uses an up-sampler, both filters operate at a set sampling rate in real time operation.

Also, $G\left(z^{L}\right)$ and $H_{IR}(z)$ are **never** convolved to obtain $H_{IFIR}(z)$. This is a cascaded system, and if the filters are convolved then all of the efficiency is lost as the zero-coefficients in $G\left(z^{L}\right)$ will not be zeros anymore.

One can also construct an IFIR system with more stages and multiple interpolation factors [25], as well as narrowband high-pass and band-pass filters, but these designs are outside of the scope of this work.

2.4 Filter Design and Performance Metrics

From the PRFB analysis above, note that the analysis and synthesis filters are determined by a prototype filter only, which is often designed to be a square-root Nyquist filter. The performance of any of the banks is therefore dependent on how closely the prototype meets the power symmetric

property, which in many cases requires spectral factorization and/or non-linear coefficient optimization procedures [27].

The design approach taken in this work to design square-root Nyquist filters will be by using the Parks-McClellan algorithm [28] as implemented by the "firpm" function in Matlab. The frequency, amplitude, and weight vectors required for the use of the firpm function are given below [29].

$$F = \left[0, \quad \frac{(1-\alpha_{PB})}{M}, \quad \frac{1}{M}, \quad \frac{1}{M}, \quad \frac{(1+\alpha_{SB})}{M}, \quad 1 \right] \tag{68}$$

$$A = \begin{bmatrix} 1 & 1 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 0 \end{bmatrix} \tag{69}$$

$$\mathrm{W} = \begin{bmatrix} 2.453 & 1 & 1 \end{bmatrix} \tag{70}$$

α_{PB} and α_{SB} determine the length of the transition band, which in turn determines how closely $h_0(n)$ meets the power symmetric property. The best choice of α_{PB} and α_{SB} can be obtained by iterative methods [29] that minimize the objective function below

$$\sigma_\Psi = \int_0^{2\pi} \left| \Psi\left(e^{j\omega}\right) - 1 \right|^2 d\omega \tag{71}$$

with the *distortion function*, defined here as

$$\Psi(z) = \sum_{k=0}^{M-1} \left| H_{SQN}\left(zW^k\right) \right|^2 \tag{72}$$

that is, the squared magnitude response of the sum of shifted versions of the square-root Nyquist filter. In the case of a Nyquist filter the squaring operation is dropped, and the distortion function is

$$\Psi(z) = \sum_{k=0}^{M-1} \left| H_{Ny}\left(zW^k\right) \right| \tag{73}$$

In a filter bank, $\Psi(z)$ represents the magnitude and phase distortion caused by analysis and

synthesis filters *assuming aliasing cancellation*. The goal is for $\Psi(z)$ to be as close to 1 (0dB) in magnitude, and to have as linear a phase, as possible.

The distortion function has a periodicity of $2\pi/M$ [4], so it will be often shown over this range only.

For the design of Nyquist *M*-band filters, the windowed sinc function in (65) will be used.

Chapter 3

3 The IFIR Filter Bank

In this chapter the IFIR filter bank structure is analyzed. Such a system combines the concepts of IFIR filters and filter banks in order to get a combined processing gain. In previous work [30] [31], IFIR filter design has been used for the design of PR filter banks, and even for the prototype filter in transmultiplexer filter banks [32] with very low-order filters; however, none of these addressed the cascaded filter implementation in a filter bank strucuture, leading to the assumption that the implementation was done by direct modulation of the filters without fully combining their polyphase components, nor taking advantage of the multirate nature of the system.

In this chapter the problem of combining the polyphase components of two filters is addressed. This combination leads to a structure in which both PR filter banks and IFIR filters are combined.

3.1 The Cascaded Filter Bank

Consider the system $T(z)$ shown in Figure 3-1. The system is implemented by cascading two filters, $H(z)$ and $G(z)$.

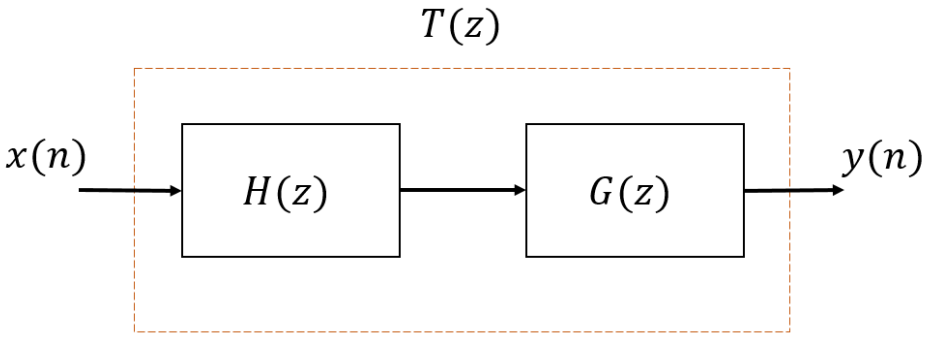


Figure 3-1 Cascaded system

Now, imagine that the resulting filter $T(z)$ is an *M*-band square-root Nyquist filter to be used in the PR filter bank. This $T(z)$ can be implemented in the cascaded filter bank shown in Figure 3-2. The cascaded filters in the bank are being directly modulated with exponentials, as seen in the

replacements of z by zW^m (they can also be modulated with cosines as in the CMFB).

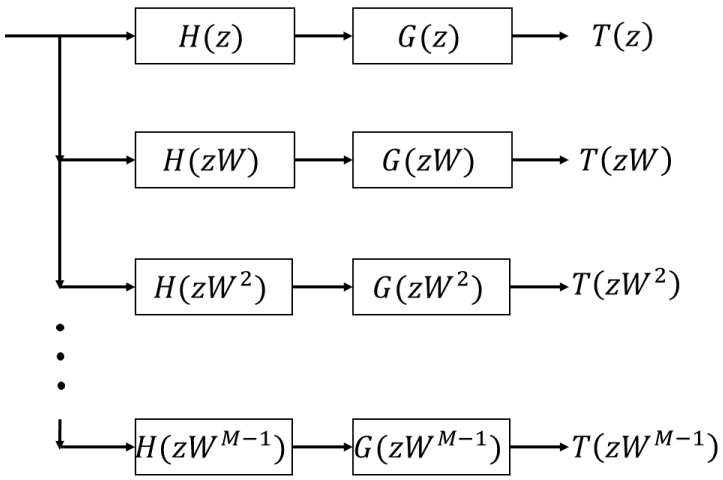


Figure 3-2 Cascaded filter bank with direct modulation of filters

As seen in Section 2.2.6 with the EMFB, it is possible to obtain a filter bank with modulated versions of a filter by decomposing said filter into its M polyphase components and applying an IDFT across the branches of the bank. If this is done for the first half of the cascaded bank, then the structure becomes the one shown in Figure 3-3 with $H_k(z)$ being the polyphase components of $H(z)$.

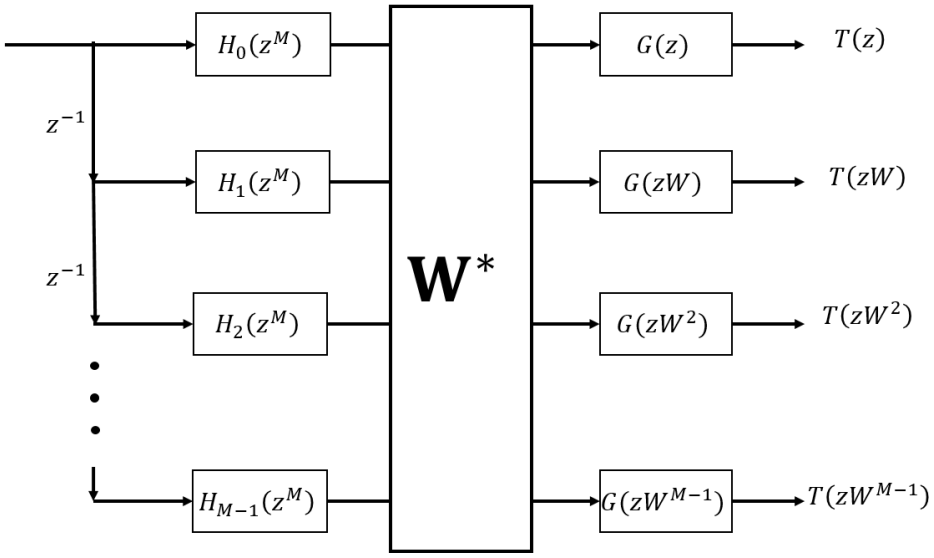


Figure 3-3 Cascaded filter bank with DFT on first branch

Define:

$$\mathbf{h}^d\left(z^M\right)=\begin{bmatrix} H_0\left(z^M\right) & z^{-1}H_1\left(z^M\right) & \cdots & z^{-(M-1)}H_{M-1}\left(z^M\right)\end{bmatrix}^T \tag{74}$$

$$\mathbf{g}^d\left(z^M\right)=\begin{bmatrix} G_0\left(z^M\right) & z^{-1}G_1\left(z^M\right) & \cdots & z^{-(M-1)}G_{M-1}\left(z^M\right)\end{bmatrix}^T \tag{75}$$

$$\mathbf{G}\left(z\right)=\begin{pmatrix} G\left(z\right) & 0 & \cdots & 0 \\ 0 & G\left(zW\right) & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & G\left(zW^{M-1}\right)\end{pmatrix} \tag{76}$$

$$\mathbf{t}\left(z\right)=\begin{bmatrix} T\left(z\right) & T\left(zW\right) & \cdots & T\left(zW^{M-1}\right)\end{bmatrix}^T \tag{77}$$

where $G_k\left(z\right)$ are the polyphase components of $G\left(z\right)$. Then, write the output of the filter bank in matrix form

$$\mathbf{t}\left(z\right)=\mathbf{G}\left(z\right)\mathbf{W}_M^*\mathbf{h}^d\left(z^M\right) \tag{78}$$

Inspecting the matrix in (76) observe that

$$\mathbf{G}\left(z\right)=diag\left(\mathbf{W}_M^*\mathbf{g}^d\right) \tag{79}$$

where $diag(.)$ denotes a diagonal matrix with the vector in $\left(.\right)$ as its diagonal. That is, the elements of the diagonal matrix $\mathbf{G}\left(z\right)$ are the outputs of an IDFT applied to the (delayed) polyphase components of the transfer function $G\left(z\right)$.

In order to express (78) completely in terms of polyphase components it is necessary to introduce the topic of **circulant** matrices [33]. Circulant matrices have the form in (80); they are determined by their first column $\mathbf{a}=\begin{bmatrix} a_0 & a_1 & \cdots & a_{M-1}\end{bmatrix}^T$, and subsequent columns are obtained by downward-circular-shift of the previous column.

$$\mathbf{C} = \begin{pmatrix} a_0 & a_{M-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & & \vdots \\ a_{M-1} & a_{M-2} & \cdots & a_0 \end{pmatrix} \tag{80}$$

Circulant matrices can be expressed in terms of their first column and a DFT/IDFT pair as follows

$$\mathbf{C} = \mathbf{W}_M \, diag\left(\mathbf{W}_M^* \mathbf{a}\right) \mathbf{W}_M^* \tag{81}$$

Going back to the cascaded polyphase filter bank in (78), by (79) $\mathbf{t}(z)$ can be expressed as

$$\mathbf{t}(z) = diag\left(\mathbf{W}_M^* \mathbf{g}^d\right) \mathbf{W}_M^* \mathbf{h}^d(z) \tag{82}$$

Adding a DFT/IDFT pair (which results in an identity) at the beginning of the expression yields

$$\mathbf{t}(z) = \mathbf{W}_M^* \mathbf{W}_M \, diag\left(\mathbf{W}_M^* \mathbf{g}^d\right) \mathbf{W}_M^* \mathbf{h}^d(z) \tag{83}$$

By the circulant matrix expression in (81)

$$\mathbf{t}(z) = \mathbf{W}_M^* \mathbf{C}(z) \mathbf{h}^d(z) \tag{84}$$

where $\mathbf{C}(z)$ is a circulant transfer matrix determined by the (delayed) polyphase components of $G(z)$, which can be written as

$$\mathbf{C}(z) = \begin{pmatrix} G_0(z^M) & z^{-(M-1)} G_{M-1}(z^M) & \cdots & z^{-1} G_1(z^M) \\ z^{-1} G_1(z^M) & G_0(z^M) & \cdots & z^{-2} G_2(z^M) \\ \vdots & \vdots & & \vdots \\ z^{-(M-1)} G_{M-1}(z^M) & z^{-(M-2)} G_{M-2}(z^M) & \cdots & G_0(z^M) \end{pmatrix} \tag{85}$$

This result makes apparent that implementing the cascaded polyphase filter bank requires a lot of work as there is now a transfer matrix involved. Every branch that goes into the IDFT is a linear combination of the cascade of outputs of the polyphase components of $H(z)$ and permutations of the polyphase components of $G(z)$. This cascaded polyphase filter bank (CPFB) is depicted in

Figure 3-4 for the simple case of $M = 4$. Note that the combination of polyphase components produces M^2 branches.

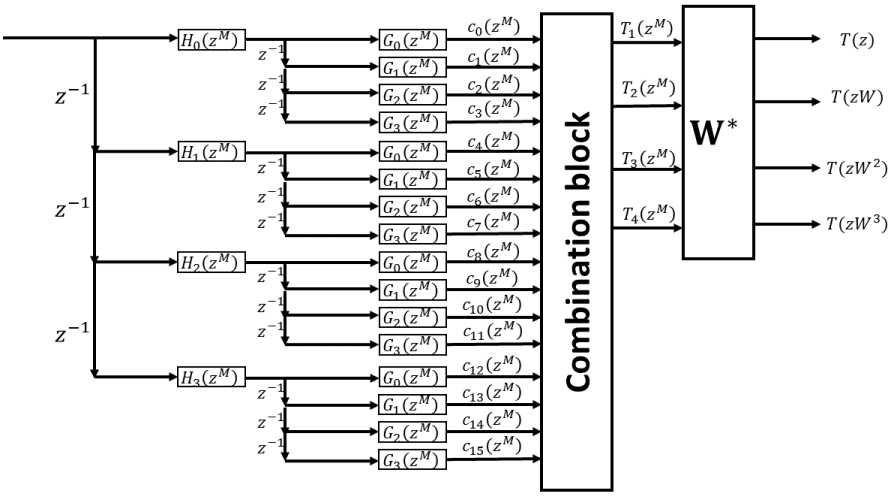


Figure 3-4 Cascaded filter bank in polyphase form

The combination block takes the M^2 branches as inputs and combines them to obtain the M inputs that go into the IDFT. This is expressed in (86), where the outputs of the combination block are the *M* polyphase components of the cascaded system $T(z)$.

$$T_k\left(z^M\right) = \sum_{i=0}^{M-1} c_{iM+\left[(k-i) \bmod k\right]}\left(z^M\right) \tag{86}$$

Just as in previous filter bank structures, it is desired to decimate the output signals of the bank for more efficient implementation. Figure 3-5 shows the result of non-maximally decimating the outputs of the bank.

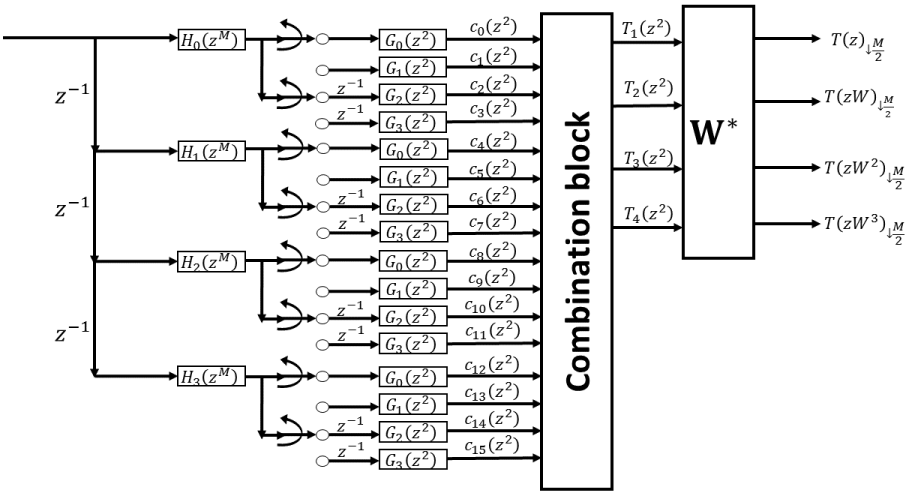


Figure 3-5 Non-maximally decimated cascaded filter bank

The decimators used to obtain the commutators in the above structure cannot be placed at the beginning of the structure as this would be an incorrect use of the Noble identities, which means that the first half of the bank – where $H(z)$ operates – is operating at the high rate.

Let N_H and N_G be the length of $H(z)$ and $G(z)$ respectively. Assuming complex inputs, every sample will take $2N_H$ MAC operations to propagate through the first filter into the *M* branches, at a rate of *M*. It then takes $\frac{(2)2N_G}{M}M = 4N_G$ MAC operations to propagate through the second filter into the M^2 branches, at a rate of 2. The *M*-point IDFT takes $4\log_2 M$ operations at the low rate, and the combination block contributes $2M$ complex additions. This is a total of $2N_H + 4(N_G + \log M)$ MACs and $2M$ adds per complex input for the implementation of the CPFB.

If instead the cascade of filters is directly modulated and implemented at each branch as in Figure 3-2, then $2M(N_H + N_G)$ MACs are needed, or $M(N_H + N_G)$ if the symmetry of the filters is taken into account. These operations happen at the high rate.

Consider the filter design in the IFIR example in Table 2-2, with $M = 32$, $N_H = N_p = 103$ and $N_G = N_{IR} = 111$. Using a direct modulation cascaded filter bank requires $32(103+111) = 365{,}856$

operations per complex sample at the high rate, while the CPFB can be implemented in $2(103)=206$ high-rate MAC operations and $4(111+5)=464$ operations at the lower rate. The computational effort comparison above is summarized in Table 3-1 in terms of the number of MACs required per second (MACPS), assuming an input rate of $M=32$ samples per second.

Table 3-1 Comparison between direct modulation and cascaded polyphase implementation

	Number of multiply-adds per second		
	High-rate operations (Rate = 32)	Low-rate operations (Rate = 2)	Total number of multiplications per second required
Direct modulation cascaded filter bank	$365,856\times 32$	-	$11,707,392$
Cascaded polyphase filter bank (CPFB)	206×32	464×2	7520

This is a ratio of 1556-to-1 more operations per second required. In the next section the CPFB is made more efficient by properties of the IFIR structure, and then compared with the filter banks studied in the previous chapter.

3.2 Non-Maximally Decimated FB Using IFIR Filters

While the CPFB is computationally efficient compared to the direct modulation implementation it may not be competitive with the $\frac{2N_{FIR}}{M}+4\log_2 M$ low-rate operations obtained for the h-NMDFB depending on the filter lengths N_G and N_H. Taking the example in Table 2-2 with $N_{IFIR}=1147$, the implementation of the h-NMDFB would take 91.68 MAC operations per complex input. This is not a good prospect for the CPFB.

The computational efficiency of the structures studied so far has come from the commutators at the beginning of the analysis bank, or from the parallelization of the input into each branch at a lower rate operation. In the case of the CPFB this was not possible to achieve because the delays at the M^2 branches do not allow the decimators to continue their way to the beginning of the bank.

However, up to this point the 'interpolation factor' used in the IFIR design has not been considered in the CPFB.

Suppose that the IFIR system is implemented by placing the IR filter before the prototype filter as in Figure 3-6. This can be done because both filters are linear time invariant (LTI) systems (ignoring quantization noise).

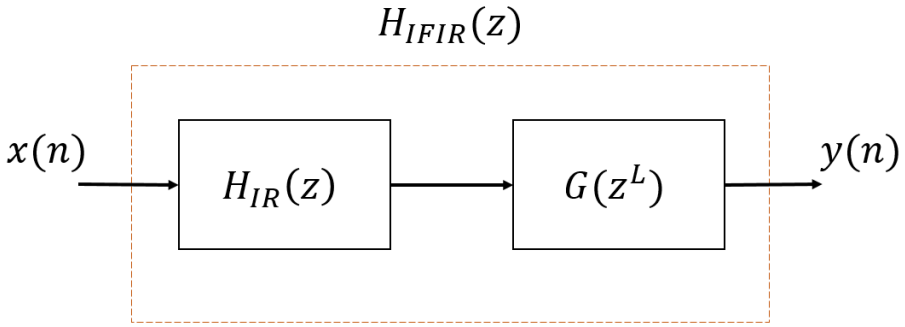


Figure 3-6 IFIR system with IR filter first

The optimal interpolation factor L_{opt} for the design of the IFIR system in terms of the number of MAC operations required was given in (67). For the CPFB L_{opt} is ignored, and instead the interpolation factor used is always

$$L = \frac{M}{2} \tag{87}$$

Under this condition, if $g(n)$ contains the prototype filter coefficients, and $g^L(n)$ is its up-sampled version, then the *M*-polyphase components of $G\left(z^L\right)$ are given by

$$G_k^L(z) = \begin{cases} \sum_n g^L(Mn+k)z^{-n} & k = 0, M/2 \\ 0 & otherwise \end{cases} \tag{88}$$

The above suggests that under the condition in (87) $G\left(z^L\right)$ will only have 2 non-zero polyphase components. Note that these two components are the result of the 2-polyphase component decomposition of the prototype filter $G(z)$.

Thus, by the definition of the polyphase representation

$$G\left(z^{L}\right)=G_{0}^{L}\left(z^{M}\right)+z^{\frac{-M}{2}}G_{\frac{M}{2}}^{L}\left(z^{M}\right) \tag{89}$$

or

$$G\left(z^{L}\right)=G_{0}^{P}\left(z^{M}\right)+z^{\frac{-M}{2}}G_{1}^{P}\left(z^{M}\right) \tag{90}$$

where $G_{m}^{P}\left(z\right),\ m=0,1,$ are the 2-polyphase components of $G\left(z\right)$.

Now suppose that the IFIR system results in a prototype filter to be used in a NMDFB with the PR property. This can be done using the CPFB structure from the previous section, as shown in Figure 3-7, where $H_{k}\left(z\right)$ are the polyphase components of the IR filter. Note that instead of M^{2} branches now only $2M$ branches are left. This is a result of the polyphase components of $G\left(z^{L}\right)$ being zero because of (87), which is a property of the cascade of filters being an IFIR system.

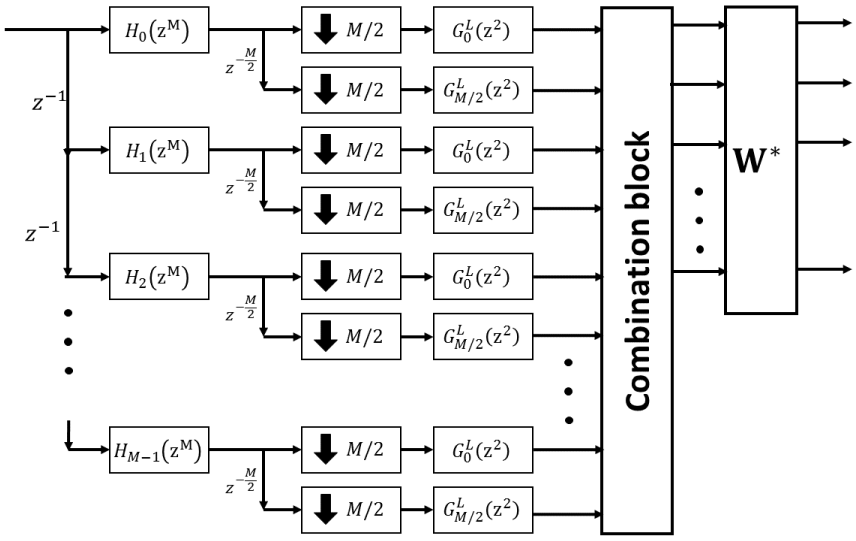


Figure 3-7 CPFB using an IFIR system

As seen above the down-samplers can now jump through the delays by use of the Noble identities, through the polyphase components of the IR filters, and be replaced by a double commutator at the beginning of the structure. The result is the non-maximally decimated IFIR filter bank (NMD-

IFIR-FB) in Figure 3-8.

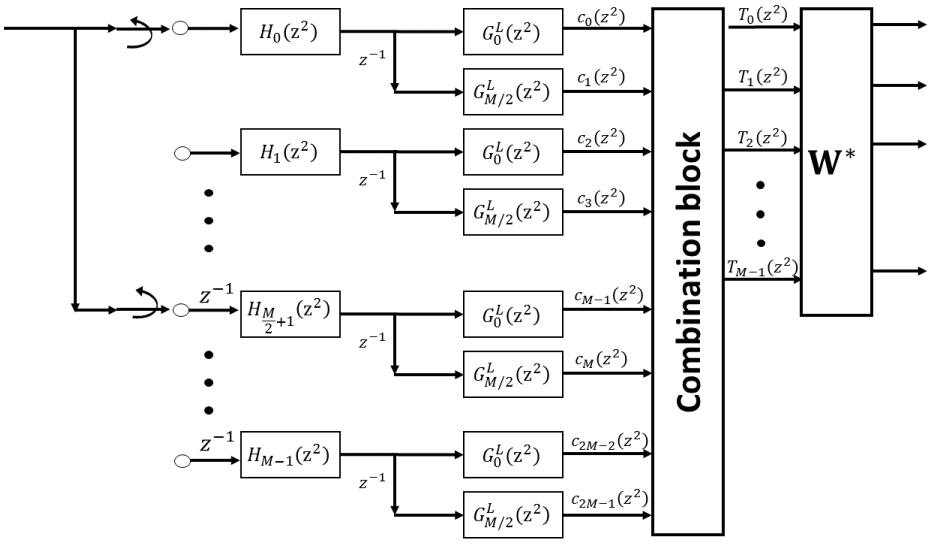


Figure 3-8 NMD-IFIR-FB

The combination block now takes 2*M* inputs and produces *M* outputs by the relationship:

$$T_k\left(z^2\right)=\begin{cases} c_{2k}\left(z^2\right)+c_{M+2k+1}\left(z^2\right), & k\le \dfrac{M}{2}-1 \\ c_{2k}\left(z^2\right)+c_{2k-M+1}\left(z^2\right), & k\ge \dfrac{M}{2} \end{cases} \tag{91}$$

Figure 3-9 shows the combination block for the case of $M=4$. The numbered blocks represent the indices of each branch. Note that the odd branches are subject to a $M/2$ circular shift before being combined with the even branches.

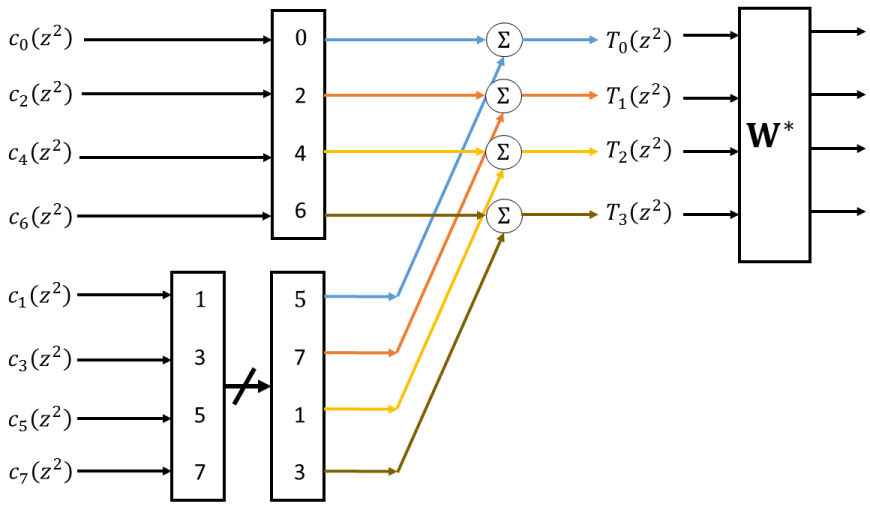


Figure 3-9 Combination block for M=4

To determine the computational complexity of the system, let N_H and N_G be the lengths of H_{IR} and $G(z)$ respectively. Each complex input takes $\frac{4N_H}{M}$ MACs to propagate through $H_k(z)$, it then takes $\frac{4N_G}{2}$ operations[1] to get to the $2M$ branches. Counting the IDFT and the combination block this is a total of $4\left(\frac{N_H}{M}+\frac{N_G}{2}+\log_2 M\right)$ MACs, and 2 additions per complex input sample.

$T(z)$ is the prototype filter for the filter bank, and because this is a non-maximally decimated system, there is the choice of designing $T(z)$ to be either a square-root, or Nyquist M-band filter to achieve perfect/close-to-perfect reconstruction. The design procedures for both choices, and the performance comparisons to the equivalent FIR filters $H_{FIR}(z)$, are discussed next.

3.2.1 Square-root Nyquist Filter Prototype Design

[1] The $1/2$ factor comes from the symmetry of the polyphase components $G_0^L(z^2)$ and $G_{\frac{M}{2}}^L(z^2)$, provided that the prototype filter has an odd number of coefficients. This symmetry on 2-band polyphase component decomposition will be assumed throughout this work.

For a square-root Nyquist filter prototype, designing $G(z)$ now becomes the design of a square-root Nyquist *half*-band filter, because of (87). That is, a filter that is power symmetric with respect to the half point of the Nyquist frequency range. Figure 3-10 illustrates such a filter.

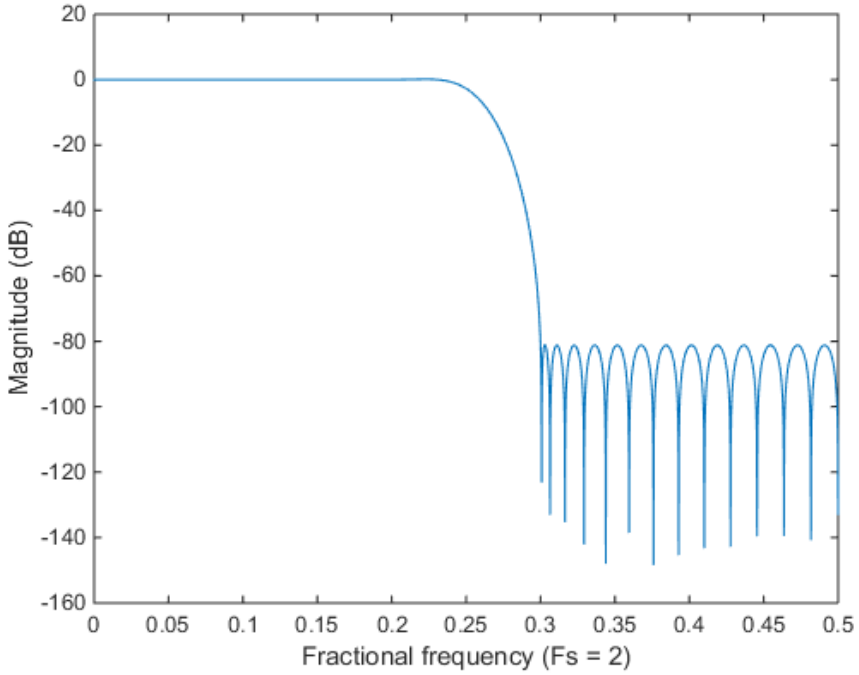


Figure 3-10 Square-root Nyquist half-band filter

The filter is then up-sampled by $L = M/2$ to obtain $G(z^L)$, which looks like a square-root Nyquist *M*-band filter with images at every integer multiple of $\omega = \frac{2\pi}{L} = \frac{2\pi}{M/2}$. The IR filter $H_{IR}(z)$ then needs to be designed to reject those images. Table 3-2 summarizes the frequency cutoffs for both filters in the IFIR system.

Table 3-2 Cutoff frequencies for square-root Nyquist M-band IFIR filter

Specification frequency cutoffs	Frequency cutoffs for $G(z)$	Frequency cutoffs for $H_{IR}(z)$
f_p	$\frac{(1-\alpha_{PB})}{4}$	$\frac{(1-\alpha_{PB})}{2M}$
f_s	$\frac{(1+\alpha_{SB})}{4}$	$\frac{2}{M} - \frac{(1+\alpha_{SB})}{2M}$

The cutoff frequencies above serve as inputs to an iterative search that reduces the objective function in (71). The goal is to obtain α values that make the IFIR system as close to a square-root Nyquist filter as possible.

This is analogous to the design procedure for an FIR square-root Nyquist filter, $H_{FIR-SQN}(z)$, as explained in Section 2.4. Both $H_{FIR-SQN}(z)$ and the IFIR cascaded system $T_{IFIR-SQN}(z)$ need to be designed to minimize the objective function in (71). The difference being that fewer coefficients need to be optimized for the IFIR filter system, which translates to a faster design time.

For the IFIR case, the distortion function is given by:

$$\Psi_{IFIR-SQN}(z) = \sum_{k=0}^{M-1} \left| T_{IFIR-SQN}\left(zW^k\right) \right|^2 \tag{92}$$

or equivalently

$$\Psi_{IFIR-SQN}(z) = \sum_{k=0}^{M-1} \left| H_{IR}\left(zW^k\right) G\left(z^L W^k\right) \right|^2 \tag{93}$$

Let N_{FIR} be the length of the FIR filter design. Figure 3-11 shows two cycles of the filter distortion functions $\Psi_{FIR-SQN}(z)$ and $\Psi_{IFIR-SQN}(z)$ of both $H_{FIR}(z)$ and $T_{IFIR}(z)$ respectively for $M = 64$, $N_H = 383$, $N_G = 23$, and $N_{FIR} = 767$.

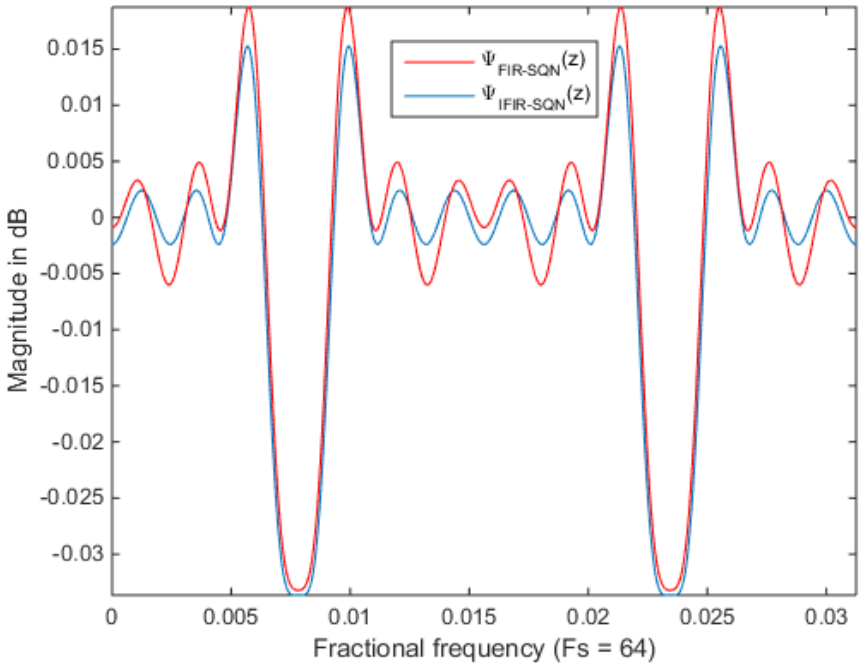


Figure 3-11 Filter distortion for both FIR and IFIR filter designs

Figure 3-12 shows the magnitude response of the filters. Both achieve a stopband attenuation of at least 75dB. $H_{FIR}(z)$ has less passband ripple; however both filters have similar distortion.

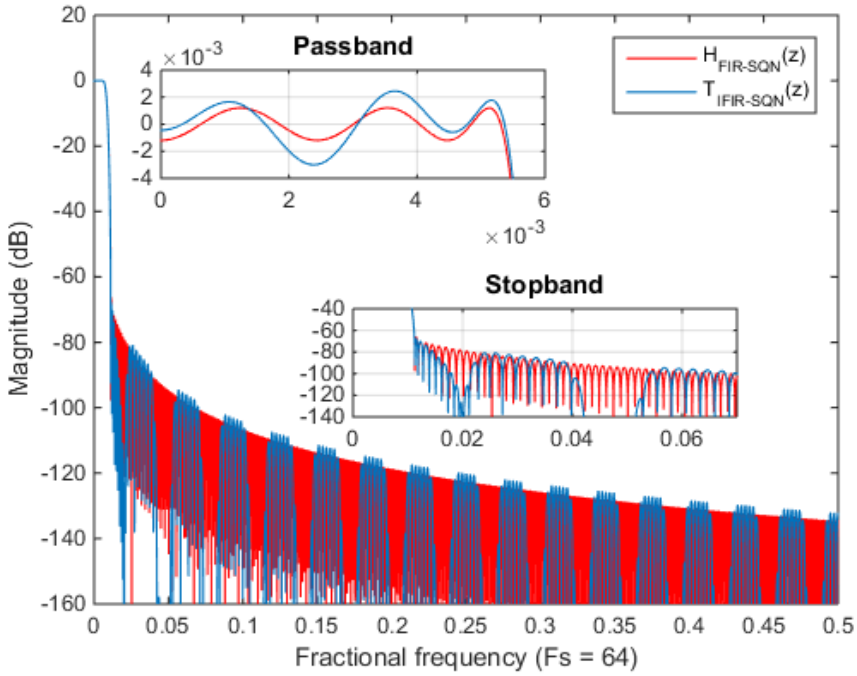


Figure 3-12 Square-root Nyquist prototype filters

As a filter performance measure, the power of the deviation from unit gain of $\Psi_x\left(e^{j\omega}\right)$ is defined:

$$\rho_{\Psi}^{FIR-SQN} = \int_0^{2\pi} \left|\Psi_{FIR-SQN}\left(e^{j\omega}\right) - 1\right|^2 d\omega \tag{94}$$

$$\rho_{\Psi}^{IFIR-SQN} = \int_0^{2\pi} \left|\Psi_{IFIR-SQN}\left(e^{j\omega}\right) - 1\right|^2 d\omega \tag{95}$$

These were found to be $\rho_{\Psi}^{FIR-SQN} = 7.0168\times10^{-7}$ and $\rho_{\Psi}^{IFIR-SQN} = 4.2263\times10^{-7}$. The distortion power is less in the IFIR case, and this is achieved with a reduced number of coefficients in the optimization process.

3.2.2 Nyquist Filter Prototype Design

The system being non-maximally decimated allows for letting $T_{IFIR}(z)$ be a Nyquist filter, just as with the h-NMDFB structure. This means that $G(z)$ will then be a Nyquist *half*-band filter. This can be easily achieved by letting $g(n)$ be a sinc function defined as

$$g(n) = \frac{\sin(\pi n/2)}{\pi n} v(n), \quad -\gamma \le n \le \gamma \tag{96}$$

where $v(n)$ is the same Kaiser window as in (65), and $\gamma = 12$.

Once again, $G(z)$ is up-sampled and the images in $G\left(z^L\right)$ need to be rejected by the IR filter. In this case the stopband cutoff frequencies are not nicely defined, but the same iterative approach to finding a suitable IR filter $H_{IR}(z)$ can be used. The same cutoff frequencies as in Table 3-2 can be used as a starting point to find the α's for the IR filter that minimize distortion power.

Now, comparing to the FIR design, Figure 3-13 shows two cycles of the filter distortion functions, $\Psi_{FIR-N}(z)$ and $\Psi_{IFIR-N}(z)$, with $M = 64$, $N_H = 383$, $N_G = 23$, and $N_{FIR} = 767$. Note that these are the same filter lengths as in the SQR-Nyquist case.

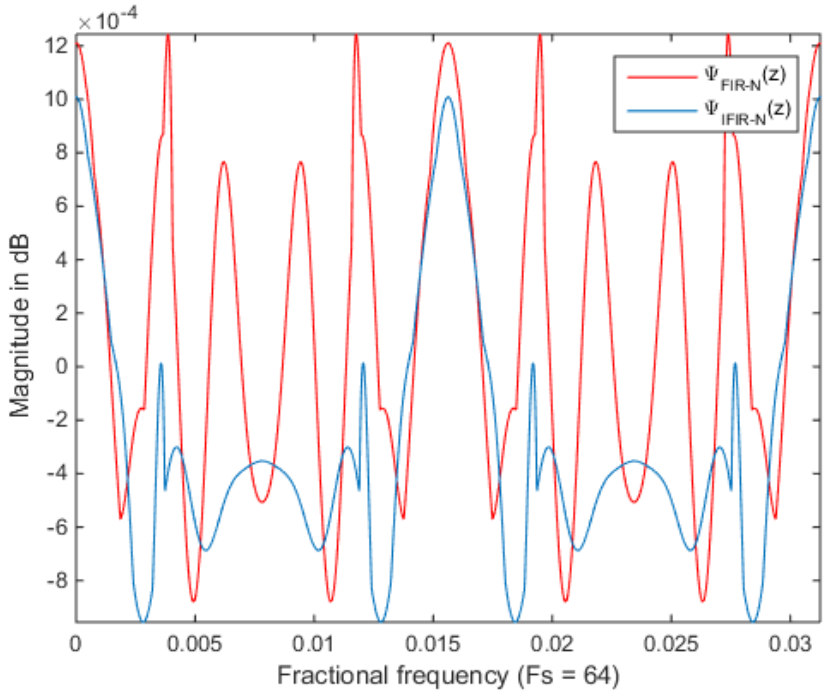


Figure 3-13 Filter distortion for both FIR and IFIR Nyquist filter designs

The filter distortion for the FIR case is given in (73), and for the IFIR case it can be written as

$$\Psi_{IFIR-N}(z)=\sum_{k=0}^{M-1}\left|H_{IR}\left(zW^{k}\right)G\left(z^{L}W^{k}\right)\right| \tag{97}$$

Note that the distortion in both cases is more than an order of magnitude (in dB) less than in the SQR-Nyquist filter case.

In terms of the distortion power, these were calculated to be $\rho_{\Psi}^{FIR-N}=4.9463\times10^{-9}$ and $\rho_{\Psi}^{IFIR-N}=4.2244\times10^{-9}$.

The Nyquist filter option also allows for the NMD-IFIR_FB structure in Figure 3-8 to be further simplified. The *M*-polyphase representation of $G\left(z^{L}\right)$, given that $G(z)$ is a half band Nyquist filter, is given by

$$G\left(z^{L}\right)=z^{-\gamma}0.5+z^{-\frac{M}{2}}G_{\frac{M}{2}}^{L}\left(z^{M}\right) \tag{98}$$

The above expression suggests that the first polyphase component can be implemented with a

delay and a shift operation only, no multiplies. The resulting structure is shown in Figure 3-14.

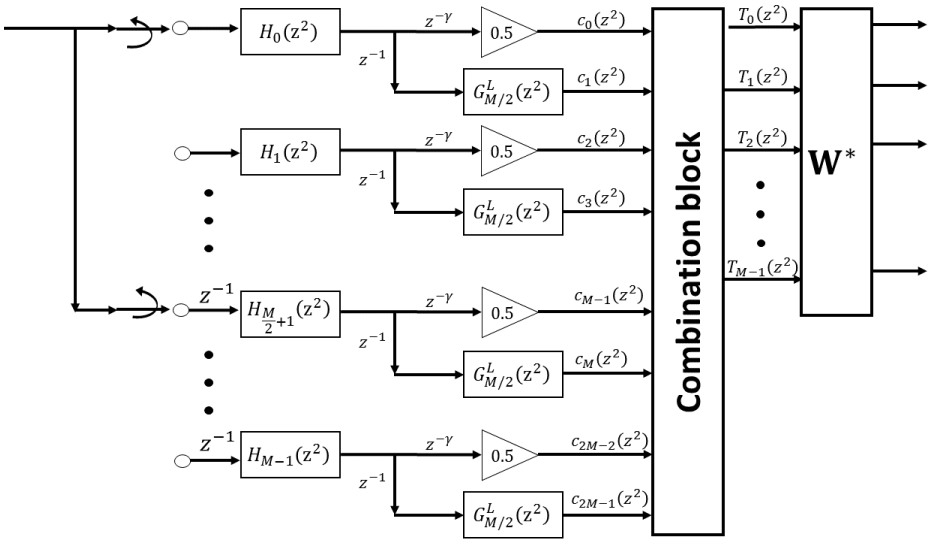


Figure 3-14 NMD-IFIR-FB structure with Nyquist filter prototype option

The computational complexity of the system then becomes $4\left(\frac{N_H}{M}+\frac{N_G}{4}+\log_2 M\right)$ MAC operations per complex sample.

In the next chapter the IFIR-based filter banks are compared with the filter banks studied in Section 2.2.

Chapter 4

4 IFIR Filter Bank Performance Comparisons

The h-NMDFB structure in Figure 2-29 is very computationally efficient. With the choice of a (sinc) Nyquist *M*-band prototype it can achieve nearly perfect reconstruction with very few real multiply-accumulate (MAC) operations per complex sample. This makes it a good candidate for comparison with the performance of the NMD-IFIR-FB.

The SQR-Nyquist and Nyquist filters used for the structures will be the same filters designed in Sections 3.2.1 and 3.2.2 for $M = 64$, keeping in mind that similar results are obtained for different values of *M*.

4.1 Performance Comparison with h-NMDFB

Define the notation:

$$\gamma_{\mathrm{X}}^{(\mathrm{M})} = \frac{N_{\mathrm{X}}}{\mathrm{M}} \tag{99}$$

as the ratio between the number of coefficients in filter $\mathrm{X}(z)$ and M. This represents the number of coefficients in each polyphase component of the M-polyphase decomposition of the filter $\mathrm{X}(z)$.

Also, denote the efficiency η_x as the number of MACs per complex input for the analysis filter bank structure "x". Then the efficiency of the h-NMDFB for the implementation of analysis and synthesis bank is:

$$\eta_{h-NMDFB} = 8\left(\gamma_{FIR}^{(M)} + \log_2 M\right) \tag{100}$$

Next the NMD-IFIR-FB and the h-NMDFB structures are compared for both square-root Nyquist and Nyquist prototype filter options.

4.1.1 Nyquist Filter Prototype Option

The efficiency of the NMD-IFIR-FB-SQN is given by

$$\eta_{IFIR-SQN} = 8\left(\gamma_{H_{IR}}^{(M)} + \frac{\gamma_G^{(2)}}{2} + \log_2 M \right) \tag{101}$$

Comparing the latter to (100), observe that computational advantage of IFIR-SQN will occur when

$$\gamma_{H_{IR}}^{(M)} + \frac{\gamma_G^{(2)}}{2} \le \gamma_{FIR}^{(M)} \tag{102}$$

The ratio of N_{FIR} to M will be picked to be 12 [20]. This suggests that to achieve the same performance with the IFIR filter the ratio of N_G to 2 also needs to be 12, as $G(z)$ is a half-band Nyquist filter. This is the choice that results in similar filter performance as seen in Section 3.2.1. The result then is:

$$\eta_{IFIR-SQN} = 8(6+6+5) = 136 \tag{103}$$

$$\eta_{h-NMDFB} = 8(12+5) = 136 \tag{104}$$

which indicates the same performance in terms of MACs per complex input.

Figure 4-1 shows the comparison between the input $x(n)$, and the outputs of both the h-NMDFB and NMD-IFIR-FB banks, $\hat{x}_{h-NMDFB}(n)$ and $\hat{x}_{IFIR-FB}(n)$. The time sequence $x(n)$ is formed by the sum of 10 sines and cosines with random frequencies following a uniform distribution over the Nyquist range. The signal power is normalized to 1. The constant *C* is chosen to normalize the signal power to 1.

$$x(n) = \frac{1}{C}\sum_{i=0}^{9}\left(\cos\left(2\pi f_{i,1} n\right) + \sin\left(2\pi f_{i,2} n\right)\right) \tag{105}$$

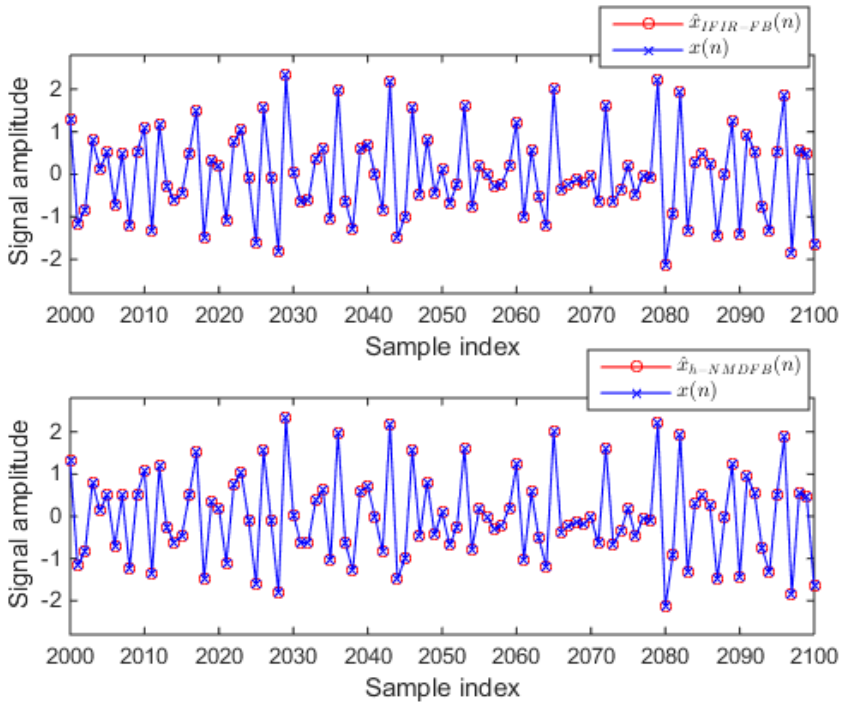


Figure 4-1 Comparison between input and output of the banks (Nyquist filter option)

The input signal has a total of 32512 samples, but a window of only 100 samples is shown for illustration and easy comparison. In Figure 4-1, note that the 'cross markers' corresponding to the input signal are aligned to the 'circle markers', which correspond to the outputs of the filter banks.

Figure 4-2 shows the difference (error or distortion) signal $d(n) = x(n) - x(n)$. All of the samples are shown in this case.

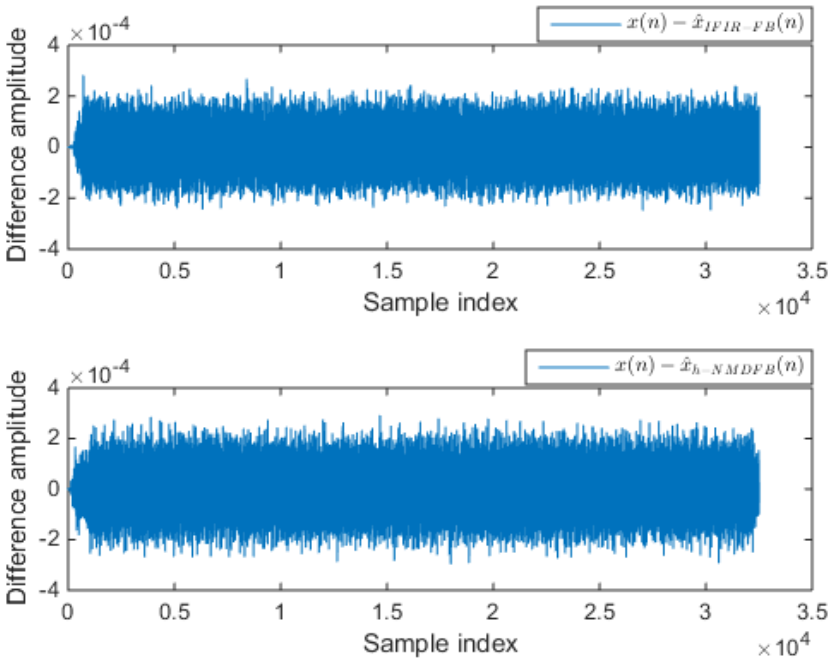


Figure 4-2 Difference of input and outputs of the banks (Nyquist filter option)

For better comparison, the variance of both error signals is computed, which gives an idea of the error power.

$$\sigma^2_{IFIR-FB} = 5.6442 \times 10^{-9} \tag{106}$$

$$\sigma^2_{h-NMDFB} = 6.4211 \times 10^{-9} \tag{107}$$

Note that the error power is very small, with a signal-to-distortion power ratio of 82.56 and 81.92 dB respectively. The error signal (difference between input and output) has an amplitude that is about 4 orders of magnitude smaller than for the input signal, which is achieved here with only 12 coefficients per polyphase branch. This error can be made even smaller by using more filter coefficients; for example, the experimental result of doubling and quadrupling the number of coefficients per branch yields further error power reduction by 1 and 2 orders of magnitude respectively. Consequently, it is reasonable to say that both structures are "practically PR filter banks" **relative to the number of coefficients used**.

4.1.2 Square-root Nyquist Prototype Filter Option

The complete analysis-synthesis filter bank structures of the h-NMDFB and the NMD-IFIR-FB

are shown in Figure 4-3 and Figure 4-4 respectively. Note that because of paraunitary system properties the synthesis polyphase components are "up-side-down flipped" versions of the polyphase components of the analysis filters.

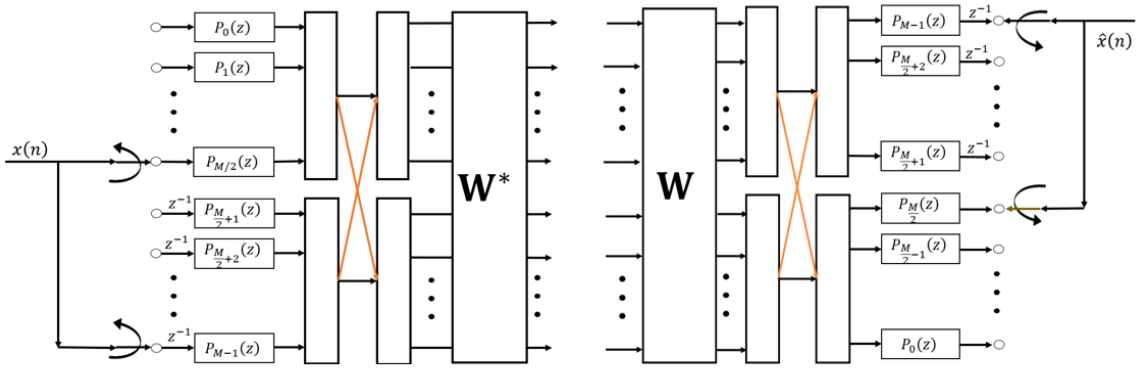


Figure 4-3 Complete h-NMDFB

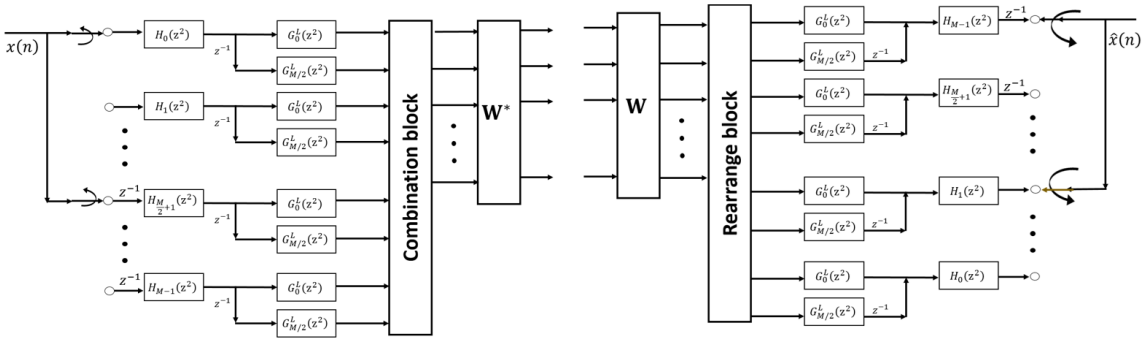


Figure 4-4 Complete NMD-IFIR-FB

The efficiency of the NMD-IFIR-FB in this case is given by

$$\eta_{IFIR-SQN} = 8\left(\gamma_{H_{IR}}^{(M)} + \gamma_G^{(2)} + \log_2 M\right) \tag{108}$$

Comparing this to (100), observe that computational advantage for IFIR-SQN will occur when

$$\gamma_{H_{IR}}^{(M)} + \gamma_G^{(2)} \le \gamma_{FIR}^{(M)} \tag{109}$$

If $\gamma_G^{(2)}$ is once again picked to be 12, then the computational advantage condition in (109) is automatically violated. However, this is the choice that results in similar filter performance as seen in Section 3.2.1. Then we have:

$$\eta_{IFIR-SQN} = 8\left(6+12+5\right) = 184 \tag{110}$$

$$\eta_{h-NMDFB} = 8\left(12+5\right) = 136 \tag{111}$$

Even with the reduced distortion power for the IFIR case, a more than 30% increase in the number of multiplications required per sample is not justifiable.

For completeness, Figure 4-5 shows the comparison between the input $x(n)$, and the outputs of both the h-NMDFB and NMD-IFIR-FB banks, $\hat{x}_{h-NMDFB}(n)$ and $\hat{x}_{IFIR-FB}(n)$. The input sequence $x(n)$ is formed by the sum of 10 sines and cosines with random frequencies following a uniform distribution over the Nyquist range. The constant *C* is chosen to normalize the signal power to 1.

$$x(n) = \frac{1}{C}\sum_{i=0}^{9}\left(\cos\left(2\pi f_{i,1}n\right) + \sin\left(2\pi f_{i,2}n\right)\right) \tag{112}$$

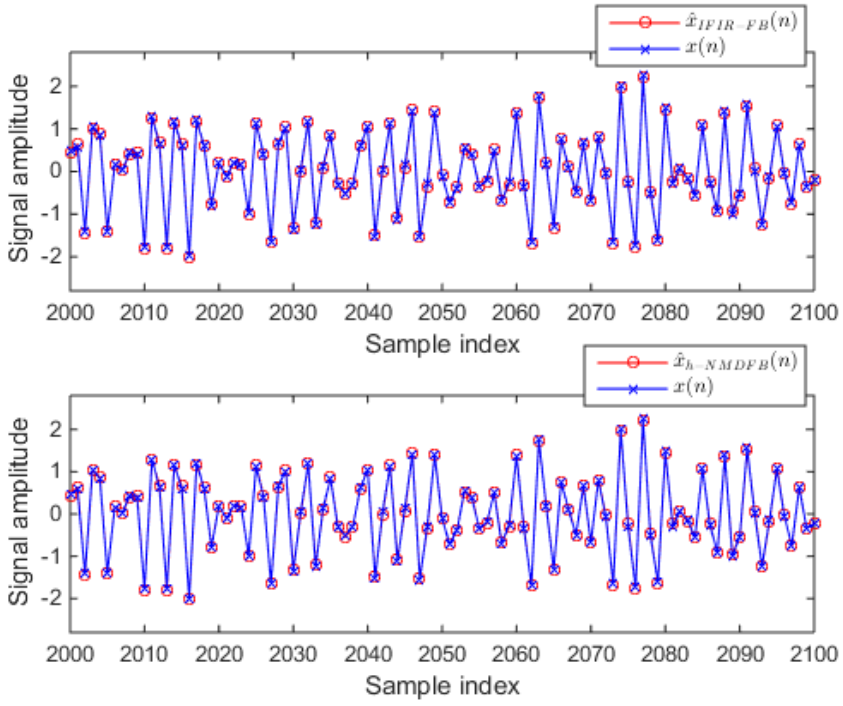


Figure 4-5 Comparison between input and output of the banks (square-root Nyquist filter option)

The input signal has a total of 32512 samples, but a window of only 100 samples is shown for illustration and easy comparison. In Figure 4-5, note that the 'cross markers' corresponding to the input signal are aligned to the 'circle markers', which correspond to the outputs of the filter banks.

Figure 4-6 shows the difference signal $d(n) = x(n) - x(n)$ for both banks. All of the samples are shown in this case.

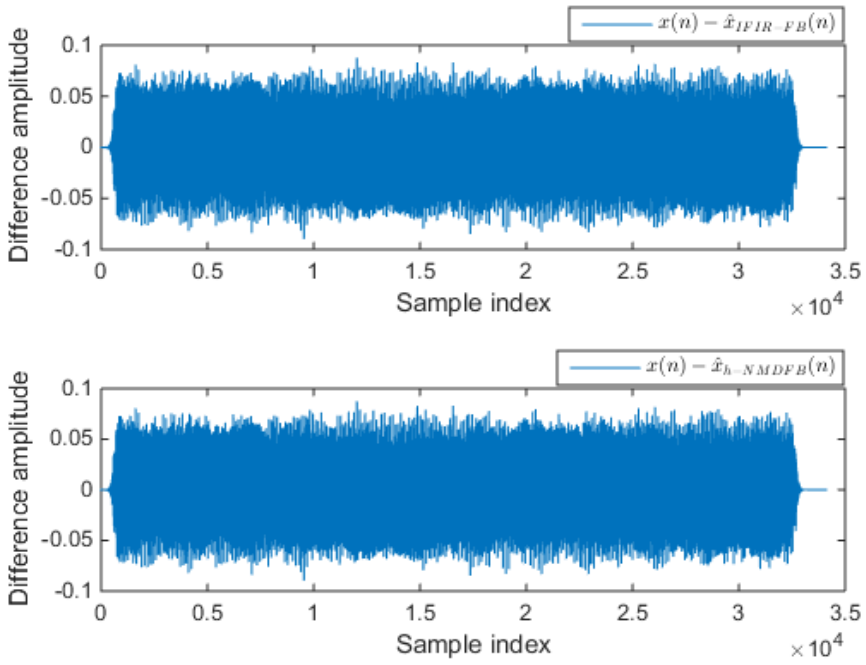


Figure 4-6 Difference of input and outputs of the banks (square-root Nyquist filter option)

Calculating the power (variance) of the difference

$$\sigma^2_{IFIR-FB} = 3.6398 \times 10^{-4} \tag{113}$$

$$\sigma^2_{h-NMDFB} = 3.6400 \times 10^{-4} \tag{114}$$

The power of the difference signal is very similar in both cases, with a signal-to-distortion power ratio of approximately 34.39 dB.

By increasing the number of filter coefficients it is possible to make the distortion power smaller. However, it has been empirically observed that – as the number of filter coefficients increases – with the Nyquist filter prototype option the distortion power minimizes "faster" than with the SQR-Nyquist filter option. Faster in the sense that an increase in the number of coefficients has a more pronounced effect in the reduction of the distortion power. Doubling and quadrupling the number of coefficients of a SQR-Nyquist prototype filter reduces the distortion power by factors close to 0.5 and 1 order of magnitude respectively. This is half of the distortion power reduction rate

observed in the Nyquist filter prototype case. This translates into fewer coefficients and more processing time per input for the Nyquist filter choice, for the same (or less) distortion power.

Chapter 5

5 IFIR Filter Bank in a Communication Environment

In this chapter the NMD-IFIR-FB (Nyquist filter option) will be used as a *transmultiplexer* (or transceiver) for transmission and reception of QPSK modulated symbols.

First, the symbols will be transmitted through a *simulated* additive white Gaussian noise (AWGN) channel for different signal-to-noise ratios (SNR). The performance of the system, measured by the bit-error-rate (BER), will be compared to the theoretical BER curve for QPSK transmission.

After this, the symbols will be transmitted between two independent transmitter (Tx) and receiver (Rx) systems, through a cable (audio) channel.

5.1 Simulated channel

The IFIR filter bank QPSK transmitter is shown in Figure 5-1. The symbols s_k^I and s_k^Q are the k^{th} in-phase (I) and quadrature (Q) symbols of each QPSK packet. Each packet is up-sampled by a factor of either 3 or 8, and shaped with a square-root-raised-cosine (SRRC) filter with a roll-off factor parameter $\beta = 0.515$. These up-sampling factors represent the samples-per-symbol (SPS) of each packet. The symbols are then up-converted from baseband to audio range using the synthesis IFIR-FB (SFB). The number of channels *M* was chosen to be 16, which means that the SFB will increase the sample rate of the symbols by a factor of 8. For this simulation the higher sampling rate F_s will be 48 kHz.

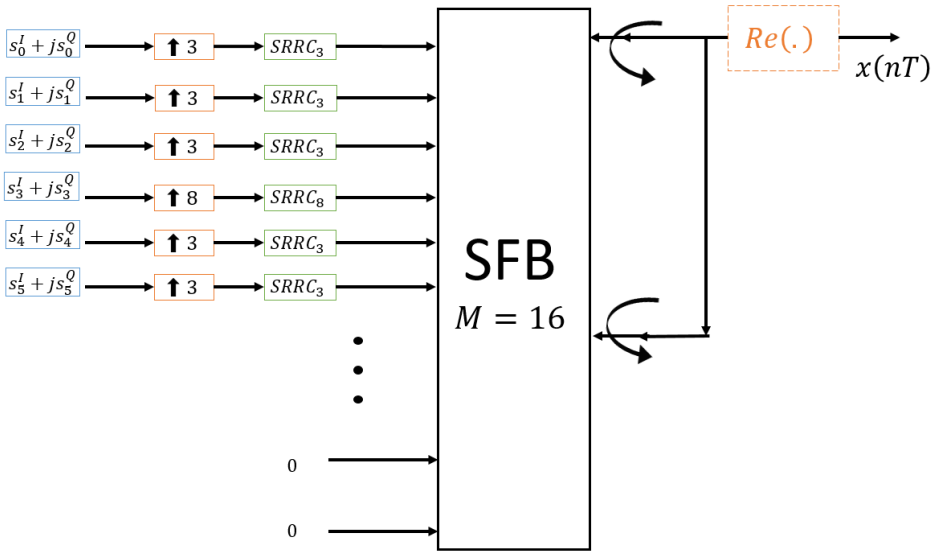


Figure 5-1 QPSK SFB transmitter

The (one-sided) spectrum of the resulting transmitted signal $x(nT)$ is shown in Figure 5-2. Note that the symbols that were up-sampled by 8 in Figure 5-1 correspond to the band centered at 12 kHz, the one with the narrower bandwidth.

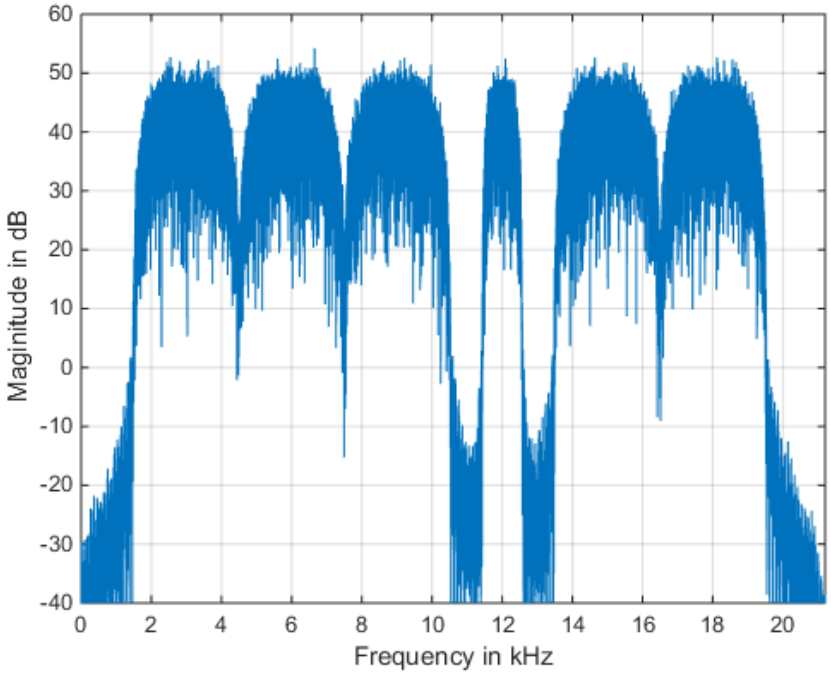


Figure 5-2 Output from SFB transmitter

The IFIR filter bank QPSK receiver is shown in Figure 5-3. The transmitted signal $x(nT)$ plus noise $z(n)$ from the channel/environment is down-converted to baseband by the analysis IFIR-FB (AFB). The down-converted channels are matched-filtered with the same SRRC filters (which maximizes SNR at the symbol sampling instant under a AWGN channel), and then 'sliced' or down-sampled to obtain the received I and Q symbols r_k^I and r_k^Q. Ideally, the received symbols r_k are exactly equal to s_k for successful transmission and zero bit errors.

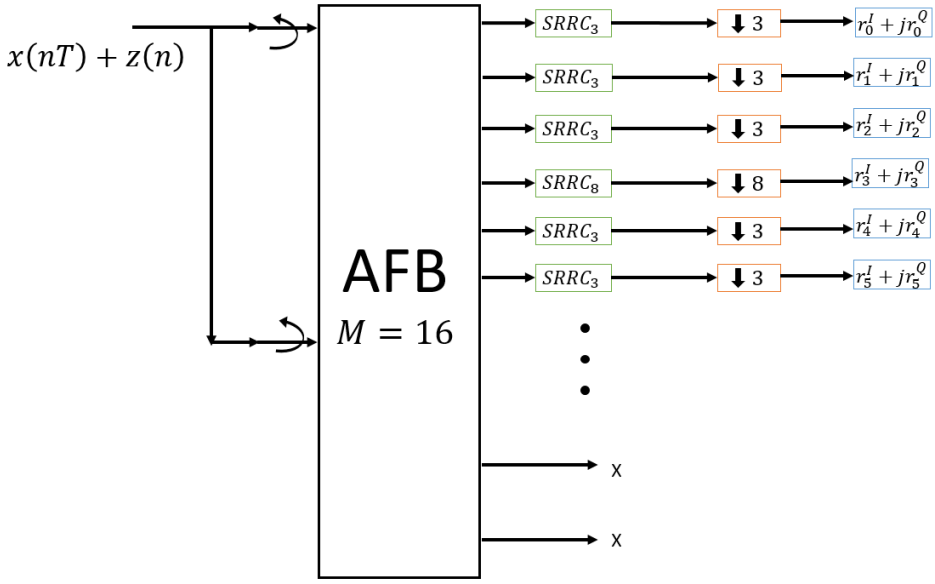


Figure 5-3 QPSK AFB receiver

Figure 5-4 shows the constellation plot of the received symbols r_k^I and r_k^Q, in the absence of channel noise. The blue crosses represent the symbols with SPS = 3, and the red circles the symbols with SPS = 8. Note that using more samples per symbol contributes to less distortion being introduced by the filter bank, as there is not as much adjacent channel interference.

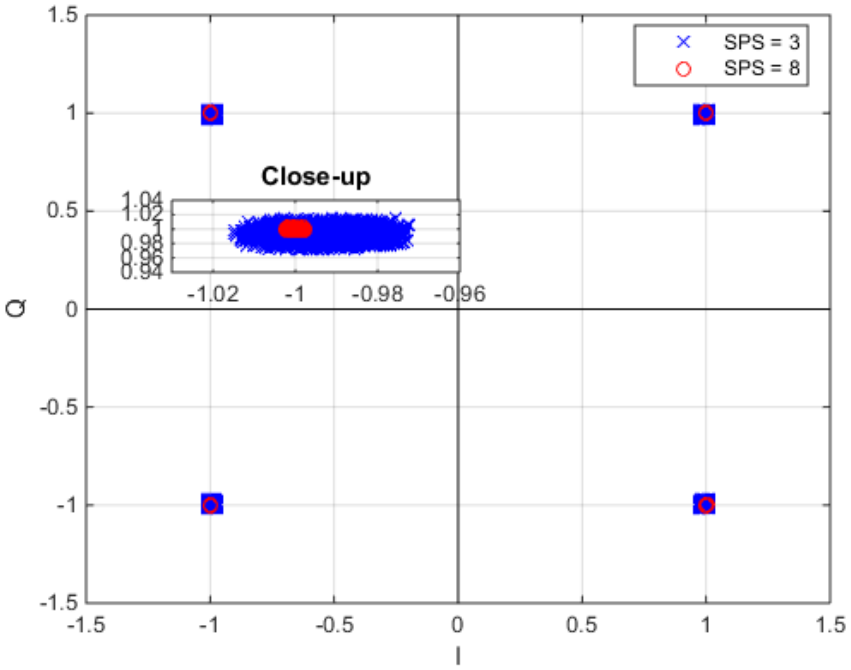


Figure 5-4 Constellation plot of received symbols (without channel noise)

While it is possible to achieve zero bit errors in the absence of channel noise, consider instead the simplified filter bank/AWGN communication channel in Figure 5-5. The channel adds (complex) Gaussian noise with a given energy per bit (Eb) over noise density (No) power ratio (EbNo).

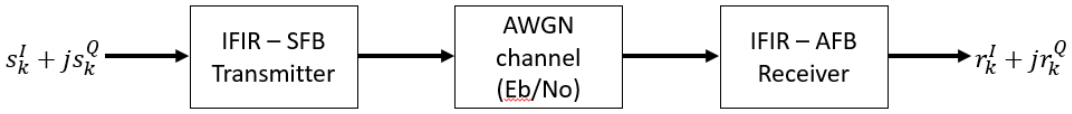


Figure 5-5 IFIR/AWGN communication channel

Figure 5-6 shows the BER for different levels of EbNo.

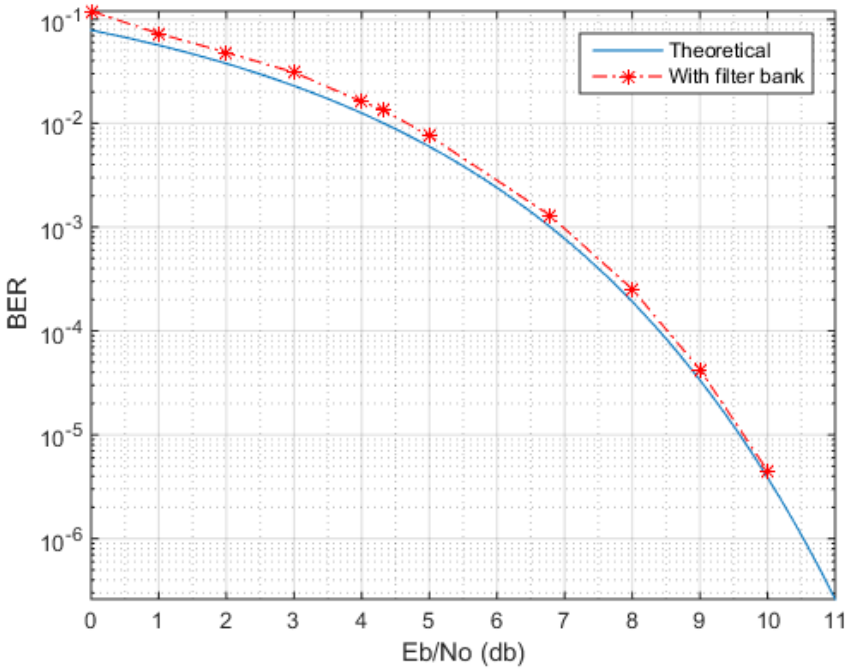


Figure 5-6 Theoretical and experimental QPSK BER curves

The blue curve represents theoretical values of BER for QPSK modulation [34]. The red curve represents the experimental BER values obtained from several simulation runs of the IFIR Tx/Rx system in Figure 5-5 for different EbNo values. Note that the experimental values follow the theoretical ones closely, being within half a dB after EbNo = 4 and closer as the signal to noise ratio increases. These results were obtained by taking all of the 6 channels used into consideration, with 10^5 symbols on every channel.

5.2 Cable Channel Transmission

In an effort to test the NMD-IFIR-FB in a more practical setting consider the system in Figure 5-7. The QPSK packets are created, pulse-shaped, up-converted to audio range, and converted to analog impulses via the audio codec of the Tx computer. The data (now analog) is transmitted through a cable channel into the audio codec of the Rx computer, where it is sampled at $F_s = 48\text{kHz}$, and stored. The received/sampled data is down-converted to baseband by the IFIR AFB. Each resulting sub-band is matched-filtered, and 'sliced' in order to obtain sampled symbols r_k^I and r_k^Q.

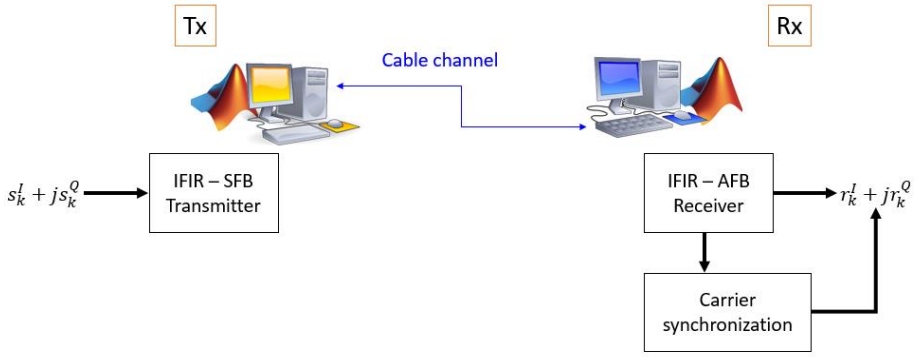
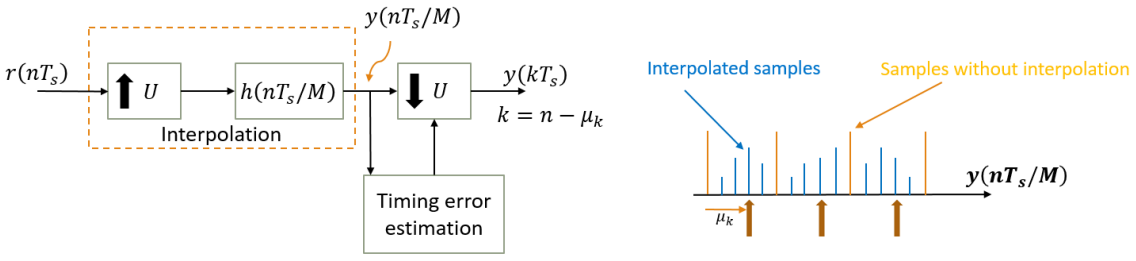


Figure 5-7 Cable-channel communication system

Because the Tx and Rx systems are operating independently, without any means of synchronizing their respective processor clocks with one another, carrier synchronization is required at the receiver side. There are many carrier/timing synchronization methods [35]; however, because there is no direct access to the oscillator in the audio codec of the Rx computer, an approach based on a digital polyphase phase locked loop (DPLL) is employed. In a DPLL each received data sequence $r(nT_s)$ is interpolated, a phase error μ_k is estimated, and the best interpolant that minimizes the error is picked [36]. A simplified model of the DPLL is shown in Figure 5-8a.

The *polyphase* DPLL takes advantage of the multiple phases in the polyphase implementation of an interpolator. This allows for the interpolator filter to be combined with the matched filter. More information about implementation procedures of the polyphase DPLL can be found elsewhere [37].



(a) DPLL simplified system (b) Interpolants representation

Figure 5-8 Polyphase DPLL

Figure 5-9 shows the spectrum of the transmitted and received QPSK symbols. Note that the last sub-band is the most affected by the cable channel in terms of magnitude response. Equalization may be required, but it is not performed in this experiment.

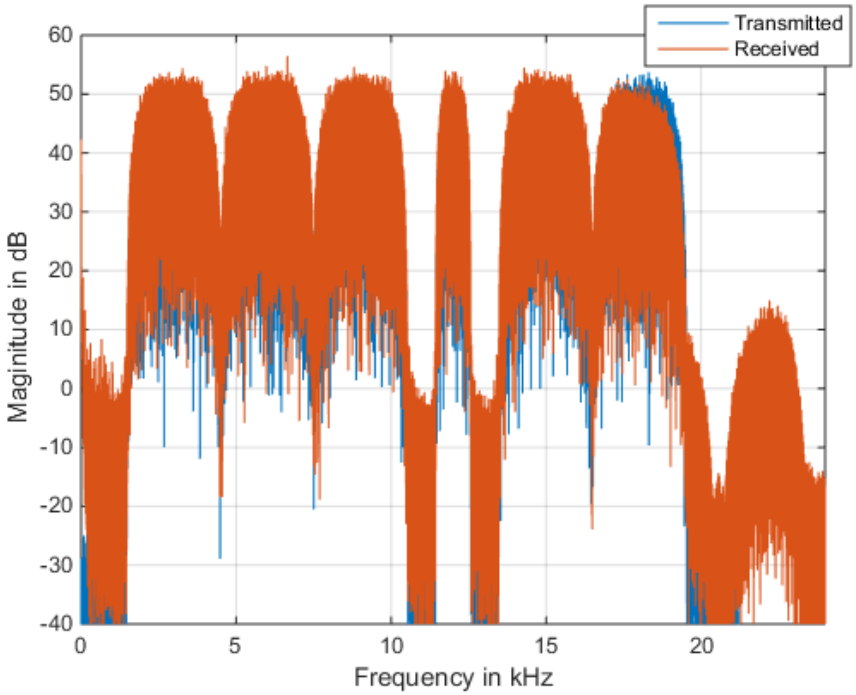


Figure 5-9 Spectrum of Tx and Rx signals

Each band is then down-converted to baseband by the AFB. An example of a down-converted, and down-sampled by a factor of $M/2=8$, sub-band is shown in Figure 5-10. This is sub-band #1 (centered at 3 kHz).

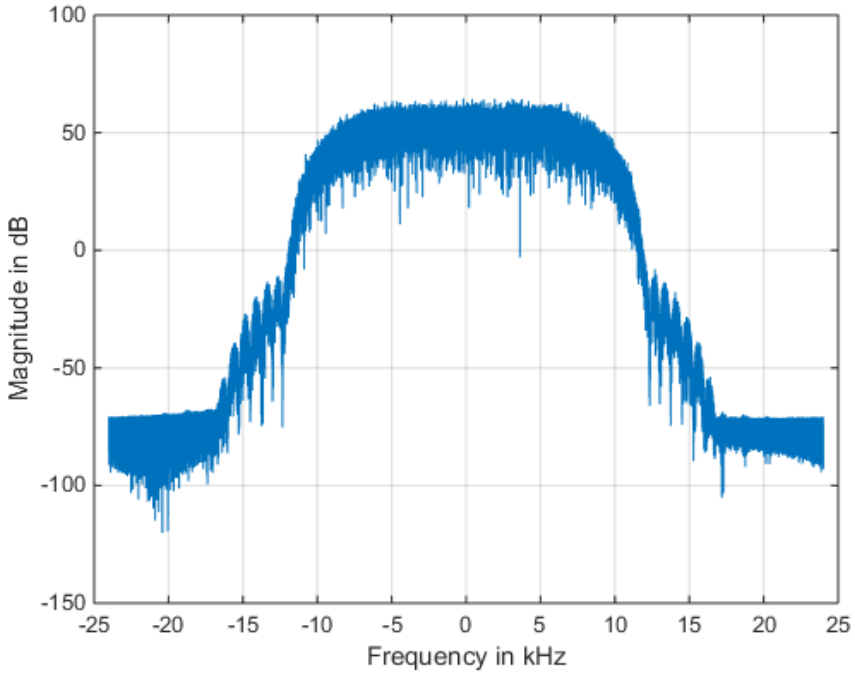


Figure 5-10 Down-converted sub-band

The matched filtering of each received packet occurs at the same time as the carrier synchronization. Figure 5-11 shows the constellation plot of all the received symbols r_k^I and r_k^Q after the best synchronization effort.

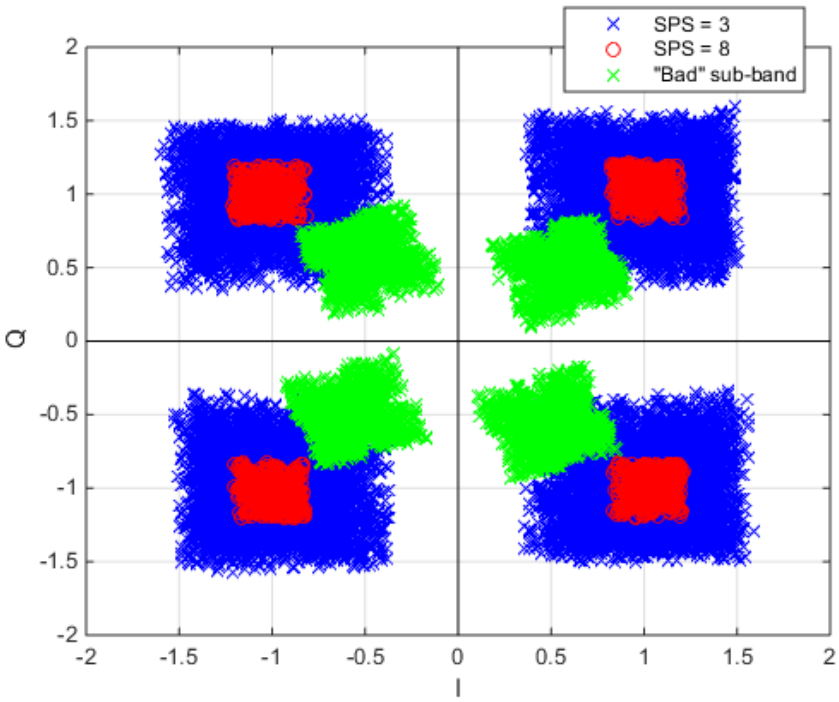


Figure 5-11 Constellation plot of received symbols

The blue crosses are the recovered bits from the packets with SPS = 3, red circles from the packets with SPS = 8, and the green crosses come from the last sub-band that was the most affected by the channel. Note that this Tx/Rx system based on the NMD-IFIR-FB resulted in a successful transmission with zero BER.

Chapter 6

6 Conclusion and Future Work

This work resulted in the development of a new filter bank structure based on IFIR filters. IFIR filters have been designed for filter banks in previous work, but these have never been implemented by their polyphase representation. The problem of combining the polyphase components of two filters was analyzed and used for the development of the IFIR filter bank. To get as close as possible to achieving the perfect reconstruction property, filter design procedures, based on optimization of a filter distortion function, were executed for the design of the IFIR filter bank prototype filter(s).

The resulting structure was compared to the very efficient h-NMDFB structure for both Nyquist and SQR-Nyquist prototype filter options. The IFIR-FB achieved similar performance to the h-NMDFB for the Nyquist filter case in terms of multiplications per complex input, and error power. However, a decrease in efficiency for the SQR-Nyquist case was found.

In a communication context, simulation results showed that the performance in terms of BER is not significantly degraded by the use of IFIR-FB to up/down-convert QPSK symbols. This means that efficiency can be greatly increased for almost the same BER performance.

The IFIR-FB was also put under test in the presence of a real communication channel, a cable channel between independent transmitter and receiver. The transmission was achieved successfully without bit errors.

The IFIR-FB needs to be studied further. In this work it was hinted that the IFIR-FB is possible only because of the IFIR filter properties (a lot of zeros that allow the Noble identities to be used for a more efficient structure), but the concept of cascading two arbitrary filters by their polyphase components is independent of these properties. The cascade of polyphase components as shown here has the potential to have properties that are useful in other applications of filter banks.

7 References

[1] A. Croisier, D. Esteban, and C. Galand, "Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques," in *Proceedings of the International Symposium on Information Circuits and Systems*, Patras, Greece, 1976.

[2] S. Darlington, "Synthesis of reactance 4-poles," *J. Math. Phys*, vol. 18, pp. 257 - 353, 1939.

[3] B. Belevitch, "Four-dimensional transformations of 4-pole matrices with applications to the synthesis reactance 4-poles," *IEEE Trans. Circuit Theory*, vol. 3, pp. 105 - 111, 1956.

[4] P. P. Vaidyanathan, Multirate Systems and Filter Banks, India: Pearson Education, 1993.

[5] S. N. Jayant and P. Noll, Digital Coding of Waveforms, Principles and Applications to Speech and Video, New Jersey: Prentice Hall, 1984.

[6] M. Vetterli, "Perfect transmultiplexers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1986.

[7] S. Mitra, Digital Signal Processing: a computer based approach, New York: McGraw-Hill Higher Education, 2006.

[8] M. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital Filtering by Polyphase Network: Application to Sample Rate Alteration Filter Banks," *IEEE Trans. on Acoust. Speech and Signal Proc.*, vol. 24, pp. 109-114, 1976.

[9] C. D. Creusere, "Subband coding of speech and audio," in *Signal Processing Conference (EUSIPCO)*, 1998.

[10] A. Satt and D. Malah, "Design of uniform DFT filter banks optimized for subband coding of speech," *IEEE Transactions on Acoustincs, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1672-1679, 1989.

[11] T. Tran, M. Ikehara, and T. Nguyen, "Linear phase paraunitary filter bank with filters of different lengths and its application in image compression," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2730 - 2744, 1999.

[12] P. Vaidyanathan, "Filter banks in digital communications," *IEEE Circuits and Systems Magazine*, vol. 1, no. 2, pp. 4 - 25, 2001.

[13] P. Vaidyanathan, "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having the perfect-reconstruction property," *IEEE Transactions on Acoustics, Speech, and Signal Processin*, vol. 35, no. 4, pp. 476 - 492, 1987.

[14] M. J. T. Smith and T. Barnwell III, "Exact reconstruction techniques for analysis/synthesis systems based on maximally decimatd filter banks," *IEEE Trans. on Acoustics, Speech and Signal Proc.*, pp. 434 - 441, 1986.

[15] J. G. Proakis, Digital Communications, 3rd Edition, McGraw-Hill Book Co, 1995.

[16] Y.-P. Lin and P. P. Vaidyanathan, "Linear phase cosine modulate maximally decimated filter banks with perfect reconstruction," *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2525 - 2539, 1995.

[17] N. J. Fliege, "Modified DFT polyphase SBC filter banks with almost perfect reconstruction," in *Internation Conference on Acoustics, Speech, and Signal Processing*, 1994.

[18] T. Karp and N. 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, 1999.

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

[20] X. Chen, F. J. Harris, E. Venosa, and D. Rao, "Non-Maximally Decimated Analysis/Synthesis Filter Banks: Applications in Wideband Digital Filtering," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 852 - 867, 2014.

[21] F. Harris, "Selectable bandwidth filter formed from perfect reconstruction polyphase filter bank," in *Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, 2010.

[22] J. F. Kaiser and F. Kuo, System Analysis by Digital Computer, New York: Wiley, 1966.

[23] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computations*, vol. 19, pp. 297 - 301, 1965.

[24] Y. Neuvo, C. Dong, and S. Mitra, "Interpolated Finite Impulse Response Filters," *IEEE Trans. on Acoust. Speech and Signal Proc*, vol. 32, pp. 563-570, 1984.

[25] A. Mehrnia and A. N. Willson, "On optimal IFIR filter design," in *Proceedings of the 2004 International Symposium on Circuits and Systems*, 2004.

[26] J. F. Kaiser, "Nonrecursive digital filter design using the Io sinh window function," in *IEEE International Symp. Circuits and Systems*, San Francisco, 1974.

[27] C. Creusere and S. Mitra, "A simple method for designing high quality prototype filters for M-band pseudo QMF banks," *IEEE Trans. Signal Proc.*, vol. 43, pp. 1005 - 1007, 1995.

[28] L. R. Rabiner, J. McClellan, and T. Parks, "FIR digital design techniques using weighted Chebyshev approximation," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 595 - 610, 1975.

[29] f. harris, S. Seshagiri, C. Dick, and K. Moerder, "An Improved square-root Nyquist shaping filter," in *Proceedings of the SDR Technical Conference and Product Exposition*, 2005.

[30] R. Chakrabarti, S. Dhabal, D. Misra, and P. Venkateswaran, "An IFIR approach for designing M-band NPR Cosine Modulated Filter Bank with CSD," in *IEEE Conference Publications*, 2012.

[31] S. Dhabal, L. Chowdhury, and P. Venkateswaran, "A novel low complexity multichannel Cosine Modulated Filter Bank using IFIR technique for Nearly Perfect Reconstruction," in *Recent Advances in Information Technology (RAIT)*, 2012.

[32] N. Rayavarapu and N. R. Prakash, "An Efficient IFIR Filter Based Prototype Filter Design for Cosine Modulated Transmultiplexers," in *International Conference on Signal Acquisition and Processing (ICSAP)*, 2010.

[33] P. J. Davis, Circulant Matrices, New York: Wiley, 1979.

[34] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport, and K. L. Kosbar, Principles of Communication Systems Simulation with Wireless Applications, Prentice Hall, 2004.

[35] W. H. Tranter, D. P. Taylor, R. E. Ziemer, N. F. Maxemchuk, and J. W. Mark, Carrier and Bit Synchronization in Data Communication: A Tutorial Review, Wiley - IEEE Press eBook Chapters, 2007.

[36] F. M. Gardner, "Interpolation in digital modems," *IEEE Transactions on Communications,* vol. 41, no. 3, pp. 501 - 507, 1993.

[37] F. J. Harris and M. Rice, "Multirate digital filters for symbol timing synchronization in software defined radios," *IEEE Journal on Selected Areas in Communications,* vol. 19, no. 12, pp. 2346 - 2357, 2001.

[38] P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques," *IEEE ASSP Magazie,* vol. 4, no. 3, pp. 4-20, 1987.

[39] M. Bellanger and J. Daguet, "TDM-FDM Transmultiplexer: Digital Polyphase and FFT," *IEEE Transactions on Communications,* vol. 22, no. 9, pp. 1199 - 1205, 1974.

[40] P. Chevillat and G. Ungerboeck, "Optimum FIR transmitter and receiver filters for data transmission over band-limited channels," *IEEE Trans. Commun.,* vol. 30, no. 8, pp. 1909 - 19015, 1982.

[41] B. Farhang-Boroujeny, "A square-root nyquist (M) filter design for digital communication systems," *IEEE Transactions on Signal Processing,* vol. 56, no. 5, pp. 2127 - 2132, 2008.