Filter Bank Multicarrier Modulation
for Spectrally Agile Waveform Design

by

Harika Velamala

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Master of Science
in
Electrical and Computer Engineering
by

MONTH YEAR

APPROVED:

Professor Alexander Wyglinski, Major Advisor

Professor Lifeng Lai

Professor Thomas Eisenbarth

Abstract

In recent years the demand for spectrum has been steadily growing. With the limited amount of spectrum available, *Spectrum Pooling* has gained immense popularity. As a result of various studies, it has been established that most of the licensed spectrum remains underutilized. Spectrum Pooling or spectrum sharing concentrates on making the most of these whitespaces in the licensed spectrum. These unused parts of the spectrum are usually available in chunks. A secondary user looking to utilize these chunks needs a device capable of transmitting over distributed frequencies, while not interfering with the primary user. Such a process is known as *Dynamic Spectrum Access* (DSA) and a device capable of it is known as *Cognitive Radio*.

In such a scenario, multicarrier communication that transmits data across the channel in several frequency subcarriers at a lower data rate has gained prominence. Its appeal lies in the fact that it combats frequency selective fading. Two methods for implementing multicarrier modulation are non-contiguous orthogonal frequency division multiplexing (NC-OFDM) and filter bank multicarrier modulation (FBMC). This thesis aims to implement a novel FBMC transmitter using software defined radio (SDR) with modulated filters based on a lowpass prototype. FBMCs employ two sets of bandpass filters called analysis and synthesis filters, one at the transmitter and the other at the receiver, in order to filter the collection of subcarriers being transmitted simultaneously in parallel frequencies. The novel aspect of this research is that a wireless transmitter based on non-contiguous FBMC is being used to design spectrally agile waveforms for dynamic spectrum access as opposed to the more popular NC-OFDM. Better spectral containment and bandwidth efficiency, combined with lack of cyclic prefix processing, makes it a viable alternative for NC-OFDM. The main aim of this thesis is to prove that FBMC can be practically implemented for wireless communications. The practicality of the method is tested by transmitting the FBMC signals real time by using the Simulink environment and USRP2 hardware modules.

To my sister Anusha

Acknowledgements

I would like to take this opportunity to express my heartfelt gratitude to my advisor Prof.Alexander Wyglinski for his constant guidance and support through the course of my Masters. I thank him for giving me this opportunity and for always encouraging me to do better. His inputs were invaluable in furthering my research and the experience of working with him helped me grow professionally as well as academically. It was a pleasure working with him.

I would like to thank Prof.Lifeng Lai and Prof.Thomas Eisenbarth for being a part of my committee and for their valuable feedback which helped me improve my work. I would like to specially thank Amit Sail, Srikanth Pagadarai and Travis Collins for their support and advice. I would also like to thank my other WiLab members Di Pu, Sean Rocke, Si Chen, Jingkai Su, Zhu Fu, Le Wang, Nathan Olivarez and Bengi Aygun. The past few years spent at WPI, I have had the pleasure of meeting some truly wonderful people who have helped in making this a truly remarkable experience. I thank them all.

I am grateful for my amazing family for always being there and supporting me. I would specially like to thank my sisters, Madhuri, Indu, Anusha, Chandan and my brother Swaroop. Last but not the least, I am eternally grateful for my wonderful parents for their unending love and belief in me. Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Spectrum is a limited resource. Even with the intense regulations on spectrum allocation, as the number of devices dependent on radio frequency spectrum access increase, the strain on the already overtaxed spectrum also increases considerably, leading to spectrum scarcity. With the advent of 3G and 4G services globally, mobile broadband traffic has also grown exceptionally. Thus with increasing data traffic, spectrum usage is being pushed to the limit. According to the FCC, the United States will run out of spectrum by next year [1]. While looking for ways to deal with this spectrum scarcity issue, a detailed study of the spectrum has revealed that the entire spectrum is not used continuously [2]. Further, it has been found that spectrum is allocated in not the most efficient of ways.

It has been determined after widespread spectrum measurement and analysis that the licensed (primary) users do not use all their allocated frequencies all the time, leading to the availability of spectral whitespaces (unused spectrum) across bands of frequency [3]. This fact is reiterated by the spectrum occupancy survey conducted at Worcester Polytechnic Institute, Worcester, MA, USA on July 11, 2008 as shown in Figure 1.1.

These unused parts of the spectrum are usually available in slabs of frequency bands and

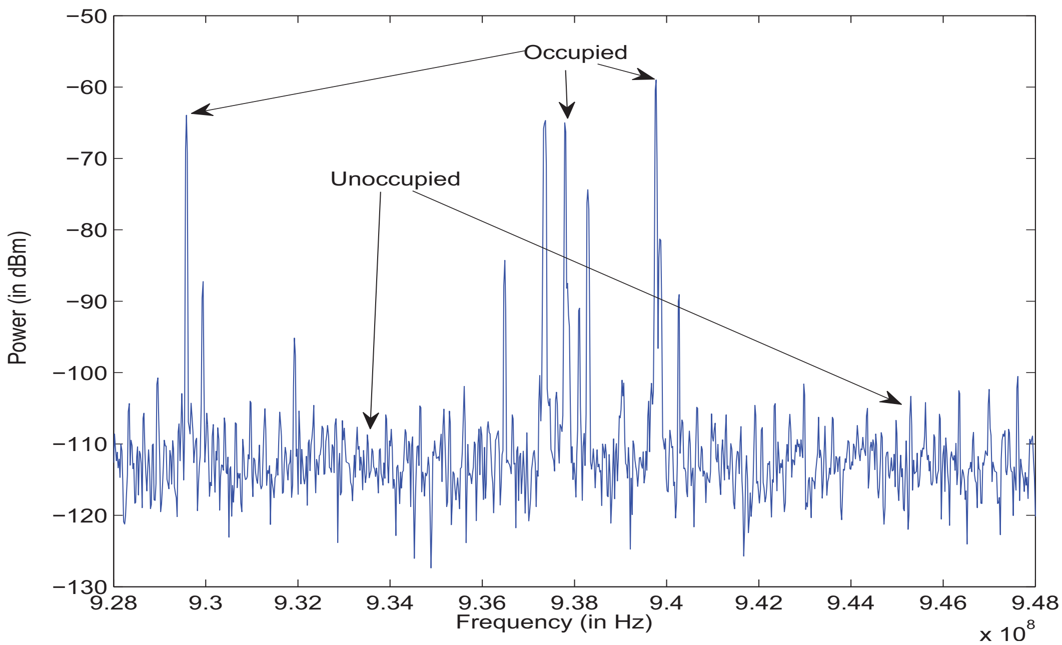
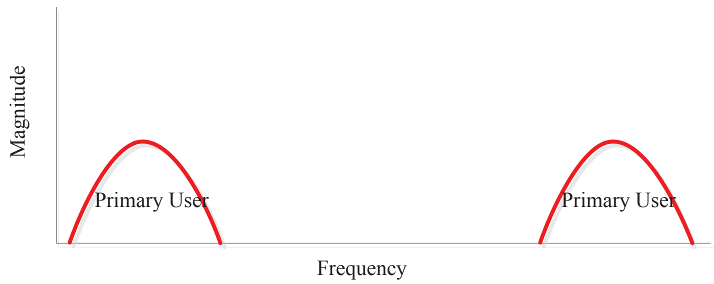
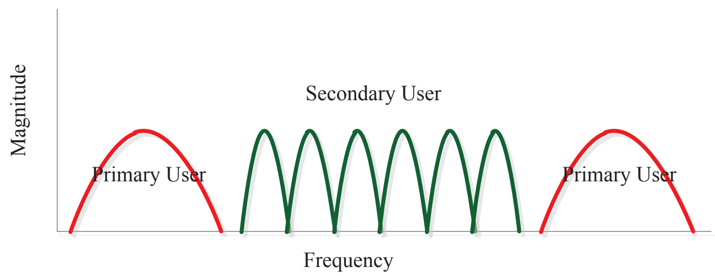


Figure 1.1: Measurements of Spectrum occupancy in the range 924 MHz to 948 MHz taken on 7/11/2008 in Worcester, MA, USA.

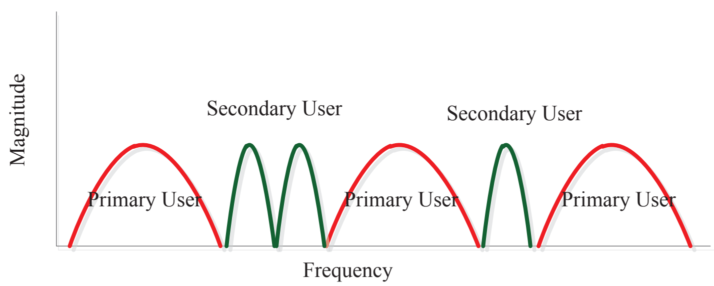
can be used by a secondary user for transmission. The location of these whitespaces may change with time, either periodically or aperiodically [4]. Therefore, this secondary user has to be capable of transmitting over non-adjacent frequencies while ascertaining not to interfere with the primary licensed user. Thus the secondary user effectively "borrows" the spectrum from the primary user temporarily. This ability is known as Dynamic Spectrum Access and the devices capable of performing it are known as cognitive radios. This process of making the most of whitespaces is known as Spectrum Pooling, which is illustrated in Figure 1.2.



(a) The primary user's frequency occupancy.



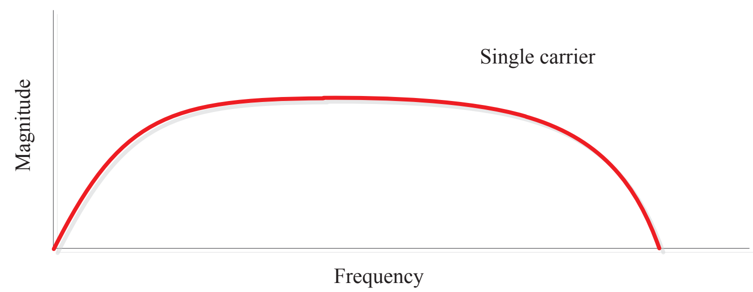
(b) The secondary user detects a whitespace and starts transmitting.



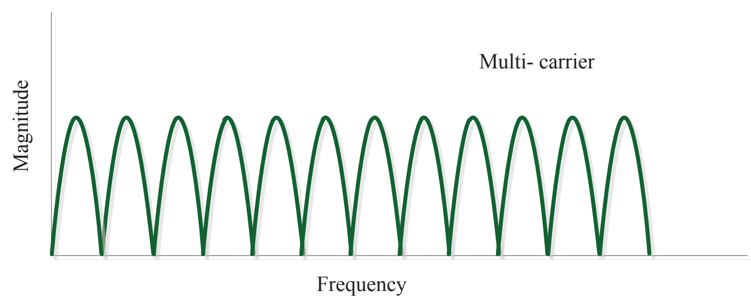
(c) The primary user becomes active in the previously unoccupied frequencies and the secondary user stops transmission in those frequencies.

Figure 1.2: Schematic Representation of the concept of Spectrum Pooling in the event of two active users.

The most popular method to perform Dynamic Spectrum Access is by multicarrier modulation [5, 2]. Multicarrier modulation (MCM) is a form of frequency division multiplexing, where data is transmitted across the channel in several frequency subcarriers at a lower data rate. Figure 1.3 illustrates single and multicarrier transmission, we see in Figure 1.3(b), that for the same frequency range, n carriers are employed instead of a single carrier. MCM systems follow a "divide and conquer" policy that makes it easier to deal with channel distortion, as it is done on a per-subcarrier basis thus improving the overall throughput [6]. MCM systems allow for overlapping adjacent subcarriers thus being spectrally efficient. They counteract the effects of ISI due to multipath fading channels, in addition to combating frequency selective fading [6]. Multicarrier modulation can be implemented in a variety of approaches, including orthogonal frequency division multiplexing (OFDM) and filter bank multicarrier modulation (FBMC) [5].



(a) Single carrier transmission



(b) Multicarrier transmission

Figure 1.3: Comparison of the single carrier and multicarrier transmissions in the frequency domain.

In recent years, OFDM has been the dominant technology for broadband multicarrier communications. However, in certain applications and circumstances, FBMC can be a more effective solution. Although FBMC methods were studied even before the invention of OFDM, it is only recently that they are receiving attention once again [7].

Filter bank multicarrier modulation is a multicarrier modulation method in which a set of analysis and synthesis filters are employed at the transmitter and receiver respectively [5, 8]. The filters are a set of bandpass filters, which are frequency shifted or modulated versions of a prototype lowpass filter. Here we mainly focus on modulated filter banks, in particular cosine modulated filter banks [9, 10, 11] and exponentially modulated filter banks [12, 13]. FBMC offers better spectral containment than OFDM as the filter bandwidth, and therefore selectivity, is a parameter that can be varied during lowpass prototype design [14, 9]. It also offers better bandwidth efficiency when compared to OFDM. FBMC eliminates the need for CP processing while attentuating interferences within and close to the used frequency band efficiently. FBMC systems are comparitively more resistant to narrowband noise effects [9]. All these reasons make FBMC a better option for cognitive systems.

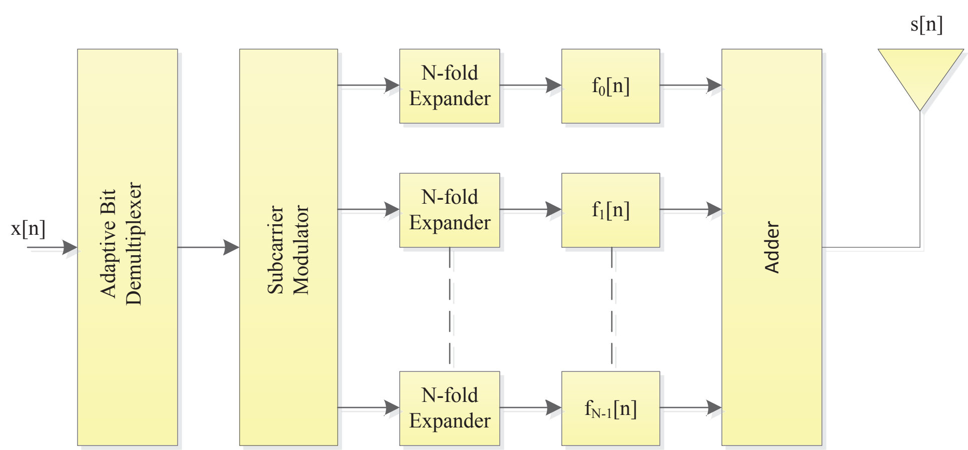


Figure 1.4: Graphical illustration of a generic Filter bank Multicarrier (FBMC) transmitter.

A typical FBMC system works as follows: At the transmitter, as shown in Figure 1.4, a high speed input signal is demultiplexed into N branches, which are then modulated by the same or different signal constellation as required. The subsequent modulated branches are then upsampled to give N copies. These upsampled data are sent through the set of synthesis filters $g_k(n)$, $k = 0, 1, \cdots, N-1$, and the outputs of all the filters are summed together to give the transmitted signal $s(n)$.

At the receiver, as depicted in Figure 1.5, the received signal $r(n)$ is passed through the bank of analysis filters $f_k(n)$, $k = 0, 1, \cdots, N-1$, to give N subcarriers of different center frequencies. The signal in each branch is then downsampled by N, demodulated and multiplexed to give the estimate of the original signal $x_r(n)$.

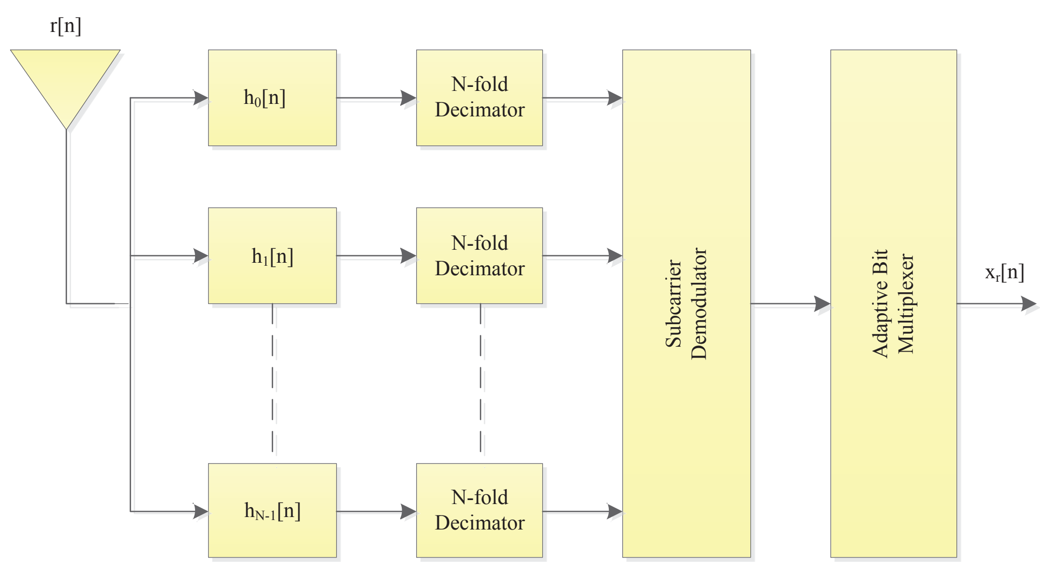


Figure 1.5: Graphical illustration of a generic Filter bank Multicarrier (FBMC) receiver.

1.2 Current State of Art

The concept of cognitive radios was first introduced in [15] [16]. Reference [16] also covers the concepts of spectrum pooling. Reference [17] was one of the first papers to talk in detail about SDRs. Once spectrum pooling became mainstream, the methods to implement it have been researched extensively. The two main contenders to design spectrally agile waveforms are OFDM and filter banks, the various methods being used in the industry are a variant of either one of these methods. Even though filter banks were first researched, for a very long time now OFDM has been the dominant method, with filter banks recently getting attention again.

The earliest multicarrier techniques used filter banks, which systems can be designed with small sidelobes making them a perfect choice for multiple access and cognitive radio applications [10]. After that initiation the work on filter banks continued further, as researchers found alternative and increasingly efficient ways of implementation. DFT based [18, 19, 20], cosine modulated and exponential modulated filter banks were some of the options that were studied. The different structures of DFT based filter banks and the conditions for perfect reconstruction were studied to find the optimal methods. In recent years the focus has shifted to more specific areas.

Oversampled and critically sampled DFT modulated filter banks with perfect reconstruction structure have been investigated [21]. An oversampled causal FIR NPR prototype has been constructed for DFT modulated filter banks based on the modified Newtons algorithm [22], this method is expected to allow low system delays. A DFT filter bank that exhibits high time-frequency (TF) concentration where the TF-translated versions of the prototype window constitute an orthogonal set has been presented that is expected to deliver maximal TF resolution [23].

A substantial amount of work has been done on the cosine modulated filter banks (CMFBs) [24, 9, 10, 25, 26, 11, 27, 28, 29] and exponential modulated filter banks (EMFBs)[13, 12, 30, 31, 32, 33]. The direct and polyphase implementation of these methods with perfect (PR) or near perfect reconstruction (NPR) has garnered a lot of attention. A scheme for developing a less complicated multichannel NPR CMFB is presented in [25]. In this paper it is suggested that the prototype filter be designed using the interpolated finite impulse response (IFIR) technique which guarantees reduced computational complexity and improved performance.

Reference [34] talks about designing a PR prototype filter in two stages using the variable neighborhood search - least mean-square error method which surpasses existing ones in reconstruction error and stopband attenuation. To further aid cognitive radio transmission, work has also been done on the spectrally agile version of multicarrier method the non-contiguous filter bank multicarrier transmission.

Non-uniform transmultiplexers that can be reconfigured have been developed based on fixed uniform modulated filter banks [35]. In this paper cosine modulated and discrete Fourier transforms filter banks have been considered while exploring the filter design, parameter selection and the transmultiplexer realization. Reference [35] proposes a transmultiplexer using a fixed uniform modulated filter bank structure where users are allowed to occupy different bandwidths according to requirement.

On the OFDM side, much more research has been accomplished in relation to achieving dynamic spectrum access. NC-OFDM which is the spectrally agile version of OFDM has been developed extensively [36, 37]. Further research is underway where mitigation of the interference caused by the high sidelobes has taken precedence with many.

Reference [38] talks about an innovative way to combat out of band emissions by mapping groups of paired input transmit symbols on to expanded constellation points such that subcarriers in each group are out of phase by 180 degrees. Spectrum shaping using windowing and active cancellation carriers to reduce interference due to out of band emissions has been discussed in [39]. References [5, 40] discuss the practicality of NC-OFDM and NC-NOFDM (non-contiguous nonorthogonal frequency division multiplexing) for opportunistic wireless access. Reference [5] also proposes employing modulated filter banks for sidelobe suppression, while [40] explores the frequency domain carrier cancellation technique which is using differently weighted subcarriers to negate the existing unnecessary subcarriers. In [41] the suggested algorithm uses mean square error (MSE) solution to reduce the interference power which works on the OFDM signal in both time and frequency domains. This algorithm is a combination of the CC algorithm [42] and the ATS algorithm [43]. Reference [44] introduces an enhanced active interference cancellation (AIC) method which selectively tunes the modulation of subcarriers to reduce interference. Reference [45] uses extended active interference cancellation (EAIC) signals for interference mitigation while shaping the spectrum with a cyclic prefix. OFDM pulse shaping is used in [46] to deal with the interference/OOB emission. Reference [47] proposes to use windowing and carrier cancellation techniques consecutively for optimal interference mitigation.

An innovative OFDM system carrier interferometry OFDM (CI/OFDM) is introduced in [48]. This system avoids the shortcomings of OFDM by using carrier interferometry (CI) spreading codes thus resulting in better BER performance and lower PAPR. Thus a non-contiguous version of it NCCI-OFDM is proposed as the new data transmission technique.

1.3 Challenges and Opportunities

With the ever increasing popularity of cognitive radios to deal with spectrum scarcity, the need to fashion existing transmission methods to be compatible with software defined radios has gained a new sense of urgency. The need for secondary devices to transmit while guaranteeing uninterrupted access to the incumbent devices has acclerated the growth of multicarrier modulation schemes. Since the spectral occupancy is fluctuating with time and frequency, the transmission schemes are required to be pliable for spectrally agile transmissions. It is crucial that the out-of-band emissions of the secondary device are suppressed as much as possible. An ideal transmission scheme was thus one which could transmit over fragmented bands of spectrum while keeping its out-of-band transmissions to a minimum. Thus the non-contiguous form of multicarrier modulation was formulated, where the subcarriers can be activated or deactivated according to requirement.

As far as practical implementation of transmission schemes with SDRs are concerned, OFDM and NC-OFDM have been implemented to varying degrees of success, whereas FBMC and its variants have been virtually untouched. The overwhelming popularity of OFDM over the past few years can be attributed to its relative ease of implementation. It employs orthogonal subcarrier signals which can be executed using fast Fourier transform (FFT) blocks. However, OFDM does have a crippling drawback especially when it comes to dynamic spectrum access in that it has substantial amount of OOB emissions which can interfere with adjacent transmissions. There has been considerable research on techniques for sidelobe suppression for OFDM. This scenario has been ideal for a revival of filter bank based multicarrier techniques, with some advocating that filter bank based techniques are more suitable for SDR purposes. Significant amount of papers have been written on filter bank implementation, but they weren't as many working hardware models.

The aim of this thesis is to implement a proof-of-concept filter bank multicarrier modulation (FBMC) using SDR hardware. One of the main challenges while implementing FBMC is the design of the lowpass prototype filter. It is a vital component on which the operation of the entire model depends. It is imperative to get the required amount of attenuation while making sure that the filter satisfies the given conditions. The ideal filter type should be chosen from the broad selection of filters. The decision of whether it should be an IIR or FIR filter, and further which type from all the options available like butterworth, chebyshev, etc. The advantages and disadvantages of using an IIR or FIR filter should be weighed carefully before making a decision.

After the selection of the prototype filter is finalized, the order of the center frequencies has to be decided. Here, we have the possibility of trying various types of filter bank modulations and determining which one works the best in all scenarios. Once a working FBMC model has been built, the next step would be implementing a non-contiguous version of it, as the aim is to achieve dynamic spectrum access. Implementing a non-contiguous version comes with its own challenges, when a filter is switched off it should provide good attenuation comparable to stopband attenuation.

1.4 Contributions

This thesis presents novel implementations of the FBMC and NC-FBMC based systems with software defined radios for dynamic spectrum access:

- **Filter bank multicarrier modulated transmitter using an SDR.** Since OFDM has been the popular method for implementing multicarrier modulation the research on developing working models using FBMC has been very restricted. Though the ultimate objective of this thesis is to implement an FBMC system that is capable of non-contiguous transmission for dynamic access, to enable that the first step is building an FBMC system compatible with a USRP, a commonly available SDR hardware platform. The FBMC systems are implemented using two variants, the cosine modulated FBMC and the exponentially modulated FBMC. The drafting process involves the design of the lowpass prototype which operates as the keystone in all of these implementations. The prototype is designed taking the attenuation, stability and performance into account. The effect of the selection of the bandwidth of the prototype and the modulating functions are examined. The designed lowpass prototype is modulated by a cosine and an exponential respectively. Various sets of center frequencies are analyzed with varying filter bandwidths to find the better performance. Both the cosine and the exponentially modulated filter bank transmitters perform equally with similar characteristics.

- **Polyphase implementation of the FBMC systems.** Complexity has always been a concern for FBMC systems. Therefore here we explore the probabilty of using the polyphase version of the system for cognitive transmission. A polyphase implementation of the FBMC system is realized here. The practicality and advantages, if any of using this version are examined while assessing its performance characteristics in a wireless transmission scenario using USRPs.

- **Non-contiguous filter bank multicarrier modulated transmitter using an SDR.** The FBMC system implemented in the earlier stage is enhanced to be capable of non-contiguous transmission. Several methods to achieve this are explored, like switching off the carriers in the unnecessary regions, and assigning filters only to the required frequencies. The methods are tested for their performance with various factors especially with bigger systems with increased number of carriers and thus filters. The NC-FBMC is implemented using both modulations and their performance is compared, and the better technique is determined.

1.5 Thesis Organization

Chapter 2 provides a literary overview of the several topics and concepts that will be covered over the course of this thesis. It contains an introduction to transmultiplexers and FBMC systems. It also deals with cognitive radios and SDR.

Chapter 3 presents the FBMC transmultiplexer models using the cosine and exponential modulation. The merits of both the implementations are compared to find the suitable one for non-contiguous implementation. This also deals with the polyphase implementation of the FBMC model and the experimental results for the implementations. Chapter 4 introduces the proposed NC-FBMC transmultiplexer model. It discusses the design and its SDR implementation in detail. It also includes the experimental over the air results. Chapter 5 gives an insight into the future work in this field and concludes the thesis.

Chapter 2

Overview of Multirate Systems and Filter Banks

Multirate systems are systems that employ multiple sampling rates. Reference [49] defines multirate systems as discrete time systems with different sampling rates at various parts of the system. Audio and video processing, communication systems, high-definition television and transform analysis are a few areas in which they find application. Changing the sampling rate of a signal ensures the increased efficiency of various signal processing operations. Increased computational efficiency, reduced transmission rate and space for storage are the benefits of multirate systems. Filter bank systems are a very good example of multirate digital signal processing systems.

Before going into detail about multirate systems, it is essential that we peruse some necessary terms and concepts that lay the base for filter bank systems.

2.1 Signals

A signal is a physical quantity that carries information, which is represented as a variation in time or space. Signals are mainly classified in one of the following three ways

[49]: continuous time and discrete time signals, analog and digital signals, deterministic and random signals.

Depending on the mathematical representation signals can be classified as continuous-time signals and discrete-time signals. A signal that is defined at every point in time is known as a **continuous-time signal**. It is represented as $s(t)$ where t is the time variable. These signals are also known as analog signals. A signal that is defined at only particular points in time is known as a **discrete-time signal**. It is represented as $s[n]$ where n is the time index. A discrete-time signal that is defined only by discrete values is known as a **digital signal**.

When the numerical representation of the physical quantities is taken into account is it either done by analog or digital representation. The definition of signals is the same as above, i.e. continuous-time signals are considered as **analog signals** while discrete-time signals and digital signals are classified as **digital signals**.

On a wider range signals can be classified as deterministic signals and random signals. When the future values of a signal can be predicted based on its past values, such a signal is known as a **deterministic signal**. The value at each point in time can be determined. When the future values of a signal cannot be predicted from its past values then it is known as a **random signal**. The value at each point in time obeys some type of probability distribution.

Here we deal with only discrete time signals and their systems. So let us study discrete-time signals and systems in more detail.

2.2 Discrete-time signals and systems

Discrete-time signals, more commonly known as discrete signals are represented by a sequence of numbers. They are represented as $s[n]$, where n is an integer that can range from $-\infty$ to $+\infty$. Discrete signals are usually obtained by sampling continuous signals or when collecting and recording data. The relation between a continuous signal, $s(t)$ and the discrete version of it can be represented as:

$$s[n] = s(nT), \tag{2.1}$$

where T is the sampling period and its reciprocal is the sampling frequency. Discrete signals are not defined for noninteger values of n. Let us now look at some widely used discrete signals or sequences.

Unit impulse: The unit impulse or unit pulse or unit sample sequence, as shown in Figure 2.1(a), is the most basic discrete-time signal and is defined as:

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

That is the unit impulse is zero everywhere except at $n = 0$ where it is unity [49, 50, 51].

Unit step: The unit step signal is given as:

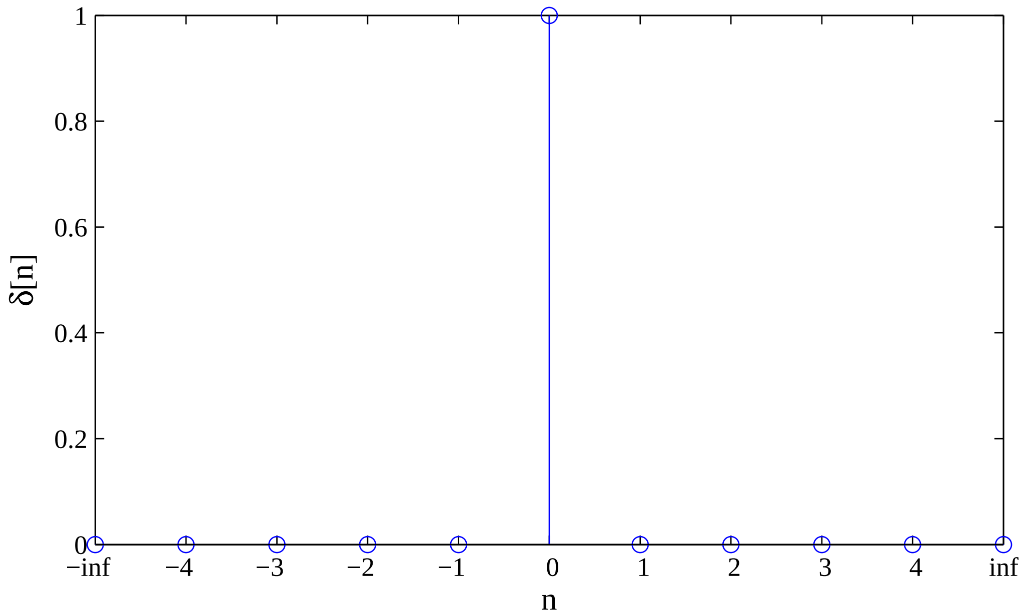
$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

Thus the unit step signal can be visualised as a sequence of unit impulses from $n = 0$ to $n = \infty$ [49, 50, 51], it is shown in Figure 2.1(b).

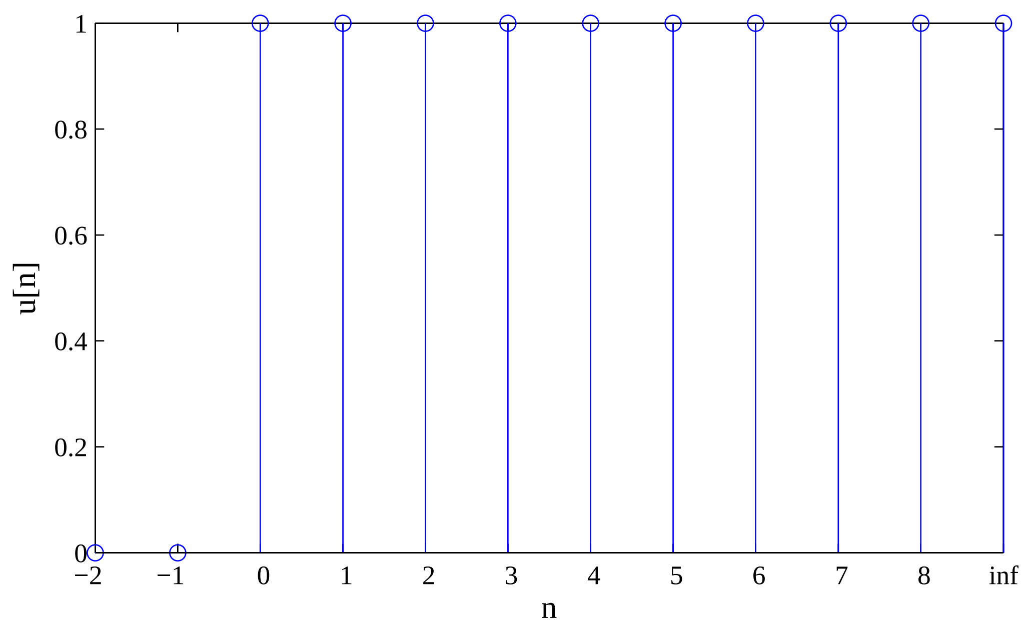
Exponential: The general form of an exponential sequence is given as:

$$x[n] = A\alpha^n, \quad \forall n, \tag{2.4}$$

where A and α can be real or complex, if A and α are real then the sequence is real. If $0 < \alpha < 1$ and $A > 0$ then the sequence is as in Figure 2.2(a). When $-1 < \alpha < 0$, then the sequence is as in Figure 2.2(c). In both these cases the sequence is decreasing with increasing n, while the first sequence has positive values and the second sequence oscillates between positive and negative values [49, 50, 51]. When $\alpha > 1$ or $\alpha < -1$ then the sequence is increasing with increasing n and the respective sequences are shown in Figures 2.2(b) and 2.2(d), where α is represented by a.

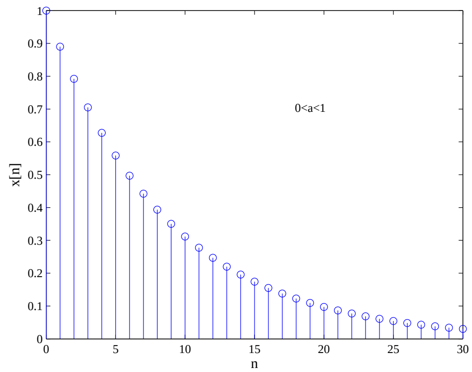


(a) Unit impulse sequence

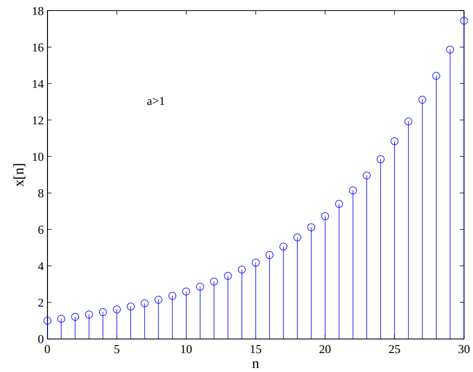


(b) Unit step sequence

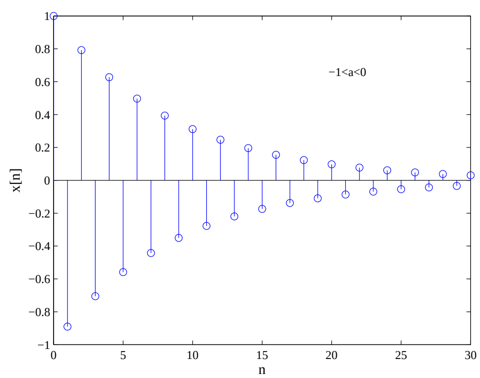
Figure 2.1: Time domain representation of the impulse and step functions.



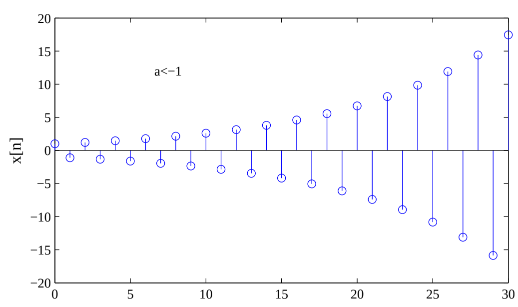
(a) Exponential sequence when $0 < a < 1$ decreases with n.



(b) Exponential sequence when $a > 1$ increases with n.



(c) Exponential sequence when $-1 < a < 0$ decreases while oscillating, with n.



(d) Exponential sequence when $a < -1$ increases while oscillating, with n.

Figure 2.2: The exponential sequence which increases and decreases depending on the various values of “a”.

Sinusoidal: When the sequence is of the form:

$$x[n] = A\cos(\omega_0 n + \phi), \tag{2.5}$$

where A is the amplitude, ω_0 is the angular frequency given as $\omega_0 = 2\pi f_0$ where f_0 is the frequency and ϕ is the phase shift, then it is known as a sinusoidal sequence [49, 50, 51].

Unit Ramp: The unit ramp signal is represented as [50]:

$$u_r[n] = \begin{cases} n, & n \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

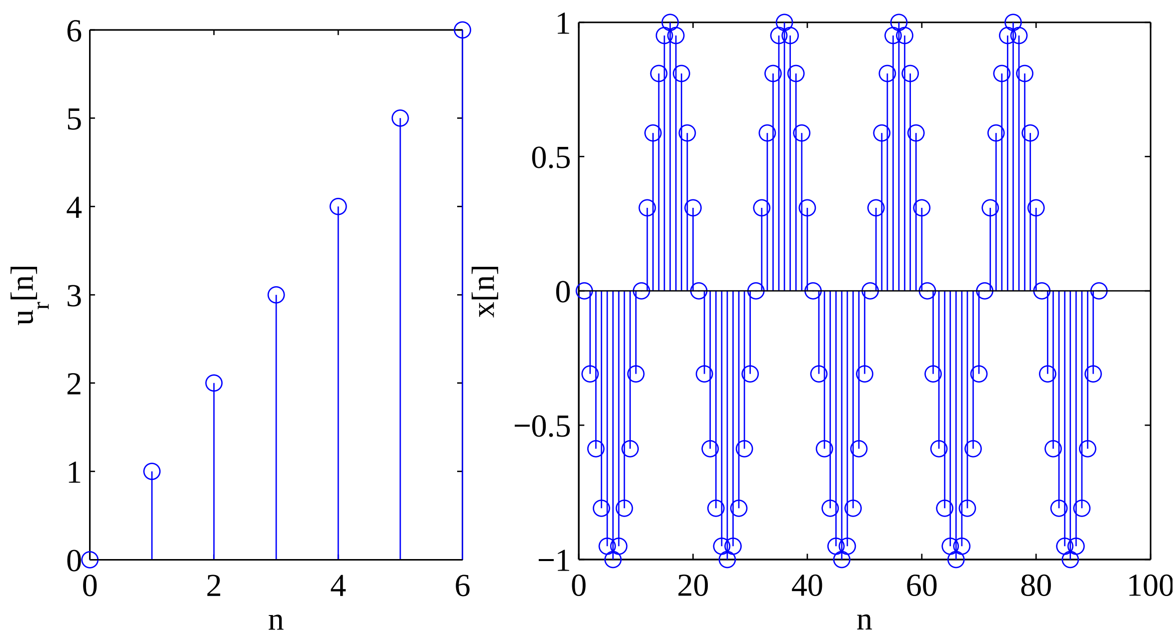


Figure 2.3: Unit ramp and Sinusoidal sequences

When we have a discrete-time sequence:

$$x[n] = x[N + n], \quad \forall n, \tag{2.7}$$

then it is known as a periodic sequence. Thus dicrete-time sequences can be periodic or aperiodic. They can also be symmetric or antisymmetric sequences [50].

A **discrete-time system** transforms or operates on a discrete-time input signal $x[n]$ to give a discrete-time signal output $y[n]$. Some inportant properties of discrete-time systems are [49, 50, 51, 52]

- **Linearity:** Systems which satify the superposition principle are linear systems and the systems which do not satisfy it are non linear.
- **Time Invariance:** If the characteristics of a system are constant with time then it is a time invariant system. If they vary with time then it is a time variant system.
- **Causality:** If the output of a system does not depend on the future values of the input then the system is said to be causal, else it is known as noncausal or anticausal.
- **Stability:** If a bounded input to a system produces a bounded output then the system is known as a stable system.

A discrete-time system that satisfies the properties of both linearity and time invariance is known as a **Linear time-invariant** or **LTI system**. LTI systems are distinguished by their impulse response $h[n]$. *Impulse response* is the output $y[n]$ of a discrete time system when the input $x[n]$ is a unit pulse function $\delta[n]$. It is also known as unit sample response. The input output relation of an LTI system is:

$$y[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m] \tag{2.8}$$

This operation is the convolution sum and can also be expressed as:

$$Y(z) = H(z)X(z) \tag{2.9}$$

where $H(z)$ is the z-transform of $h(n)$ and is known as the transfer function of the system. In difference equation form the system function of a discrete-time system is given as:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^- k}{\sum_{k=0}^{N} a_k z^- k} \tag{2.10}$$

Therefore the input output relation of the system can be given as:

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \tag{2.11}$$

where we consider $a_0 = 1$ for simplification. Thus an LTI system is represented by a linear constant coefficient difference equation [49, 52], where r-input p-output LTI systems are characterized by $p \times r$ transfer matrices $\mathbf{H}(z)$. These can be used to characterize filter banks [51]. The impulse response $h(n)$ of an LTI system may be of finite or infinite duration, according to which LTI systems are classified as finite impulse response (FIR) systems and infinite impulse response (IIR) systems.

If the impulse response has a limited number of nonzero samples or has a finite duration then it is known as a **finite impulse response (FIR) system**. The input output relation of an FIR system is given as:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k], \tag{2.12}$$

where $x[n]$ is the input to the system and $y[n]$ is the output from the system. An FIR system has no feedback from the output and thus the output of the system at any time only depends on the value of the input signal at different times. It is thus a non-recursive system [49, 52]. They are stable, all zero systems with finite memory, which are implemented in any one of the following ways: Direct form, Direct form linear phase, Cascade form, Frequency sampling or Lattice structures.

If the impulse response has unlimited number of nonzero samples or has an infinite duration it is known as an **infinite impulse response (IIR) system**. The input output relation of an IIR system is given as:

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \tag{2.13}$$

Thus an IIR system has a feedback from its output, so the output of the system at any time depends on the past outputs of the system as well as the input signal at different times.

It is a recursive system [49, 52]. They have infinite memory and it is harder to achieve stability for IIR systems. When $b_k = 0$ for all M then we have an all pole IIR system. IIR systems are implemented in one of the subsequent ways: Direct form I, Direct form II, Transposed direct form, Cascade form, Parallel form and Lattice structures. These are all just methods to implement an IIR system structure.

A special type of LTI systems the discrete-time filters, also known as Digital Filters play an important role in this dissertation. Here we try to get a better understanding of the how filters work and the different type of filters.

2.3 Filters

Filters are frequency selective electric circuits that pass signals of designated band of frequencies while attenuating the signals of frequencies outside the band [53]. An ideal filter is a device that provides distortionless transmission over certain frequency bands and zero response at other frequencies [54]. They are used to eliminate unwanted components or features such as interference, noise and distortion products from a signal that is bearing information. Thus filters are capable of allowing and rejecting frequencies selectively. Alternatively, filters can be defined as electrical networks that modify the amplitude and phase components of a signal with respect to frequency.

Filters can be analog or digital, having different types and orders. Filters can be classified as passive or active depending on the components used to make them. A passive filter is made using passive components like resistors, capacitors and inductors, while active filters use operational amplifiers as the active component in addition to using resistors and capacitors. Digital filters are classified as Infinite Impulse response (IIR) or Finite Impulse response (FIR) filters. Here we are going to look in to digital filters.

Before we go into detail about the different types of digital filters, all filters are classified into certain basic types depending on their frequency response characteristics [50, 51]. The frequencies which are allowed to pass are known as the passband frequencies while the rejected frequenies are known as the stopband frequencies. The filters have constant gain in their passband and zero gain in the stopband. The constant gain is usually represented as unit gain. The boundaries of the passband frequencies or the frequencies where the filter gain has dropped by 3dB of its maximum voltage gain are known as the -3dB frequencies or the cutoff frequencies. The following Figure 2.4 illustrates the characteristics of a typical filter, here a lowpass filter characteristics are taken as reference.

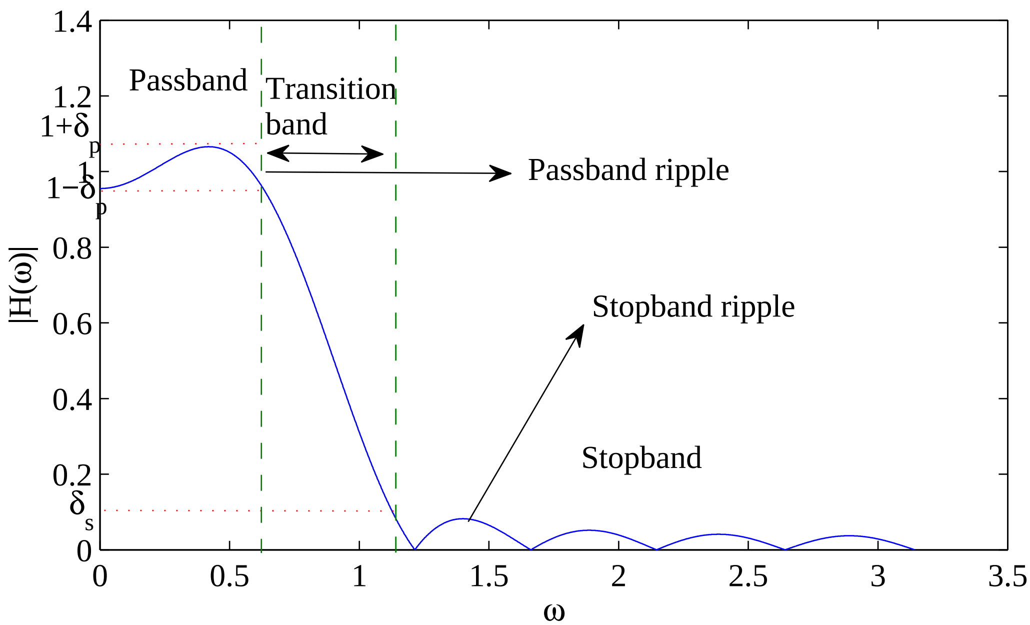


Figure 2.4: An illustration of the characteristics of a physical filter.

Some of the general terms associated with filters are discussed next. The edge frequency of the passband is denoted by f_p while the start of the stopband is denoted by f_s. The transition band or region of the filter is the change of the frequency response from the passband to the stopband. It is given as $f_s - f_p$ in a lowpass filter and $f_p - f_s$ for a highpass filter [50]. The range or extent of the passband of the filter is known as its bandwidth. In the passband the magnitude response should be unity with an allowable error of $\pm\delta_p$, where $\delta_p \ll 1$. In the stopband the magnitude response should be zero with an allowable error of $\pm\delta_s$, where $\delta_s \ll 1$. The values δ_p and δ_s are the peak ripple values in the passband and stopband respectively. [49, 52, 51]. Digital frequencies are represented by f or ω with their respective subscripts whereas analog frequencies are represented as Ω. The classification of different filter types are as follows:

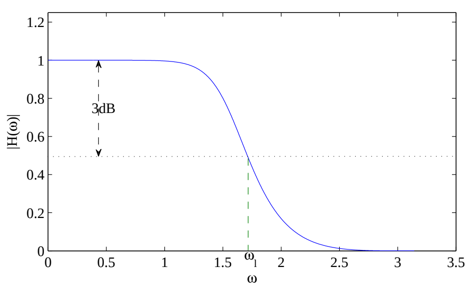
- **Low pass Filter:** A filter that allows low frequency signals but rejects frequencies above its cutoff frequency is known as a low pass filter. Therefore this filter has higher gain at low frequencies compared to high frequencies. Its cutoff frequency is represented by f_h. The frequency response of a low pass filter is represented in Figure 2.5(a).

- **High pass Filter:** A filter which attenuates the frequencies below its cutoff frequency, that is allows only high frequencies to pass through is known as a high pass filter. It is the reverse of a low pass filter. Its cutoff frequency is represented by f_l. In the frequency domain a high pass filter is represented as in Figure 2.5(b).

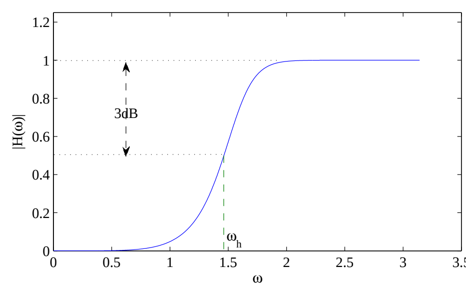
- **Band pass Filter:** The signal between a range of frequencies is permitted to pass without attenuation while everything out of that range is attenuated in a band pass filter. The filter characteristics are as in Figure 2.5(c), where the passband is given as $f_l < f < f_h$. The frequency response of a band pass filter has its highest or peak value at the center frequency given by $f_c = \sqrt{f_l f_h}$.

- **Band reject Filter:** Band reject or band stop filters are filters which allow all frequencies except some particular range of frequencies to pass through. It is the reverse of the band pass filter. The passband frequencies are represented as $f < f_l \quad and \quad f > f_h$. Notch filter is a special type of band stop filter which attenuates one particular frequency, i.e. has perfect nulls in its frequency response at that frequency. It is a narrow band reject filter [53]. The following Figure 2.5(d) show the frequency responses of a band reject filter.

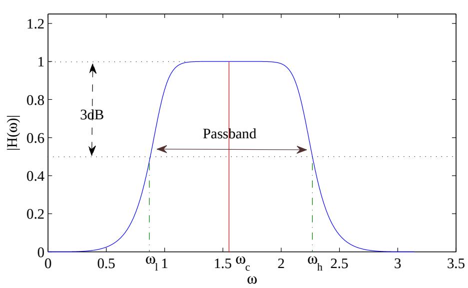
- **All pass Filter:** A filter which allows all frequencies to pass through it is known as an all pass filter. It is mainly used to change the phase of a signal without affecting its amplitude. In allpass filters, the phase relationship between frequencies is altered by adjusting the propagation delay with frequency. Allpass filters are used to balance phase shifts in systems or to convert mixed phase filters into minimum phase filters. The gain of an all pass filter is unity.



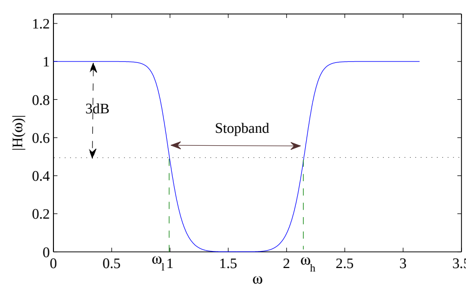
(a) Lowpass filter



(b) Highpass filter



(c) Bandpass filter



(d) Bandstop filter

Figure 2.5: Frequency response representations of filters with varying frequency ranges.

2.4 Digital Filters

Digital filters are classified as IIR and FIR filters depending on their characteristics and design method. [55]

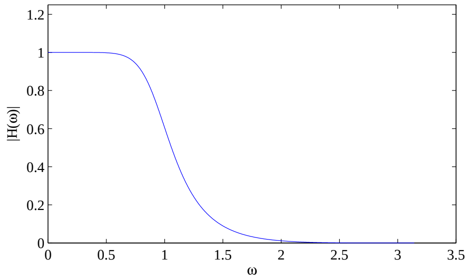
2.4.1 Infinite Impulse Response (IIR) Filters

A digital filter that has an infinite duration unit sample response is known as an IIR filter. The output of the IIR filter is a function of future, current and past values of input, as well as past values of outputs. IIR filters are also known as recursive filters as they have a feedback from the output. [56]. Due to this feedback any error can be fedback and accumlate. This feedback error can oscillate and becomes a problem for long sequences, i.e. for equipment that is always on. IIR filters are normally not stable and have nonlinear phase. They have a start up transient that decays exponentially.

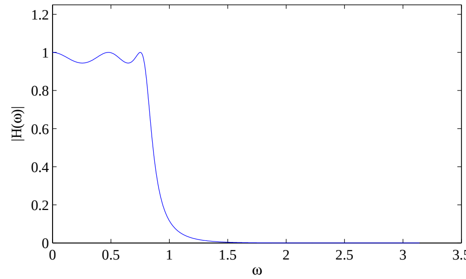
Digital IIR filters are usually derived from their analog counterpart prototypes. [56, 49]. The design of IIR filters is therefore done step by step. The given design specifications are converted to their analog form, an analog filter is then designed which is converted to the digital form by using a transformation.

- **Butterworth Filters:** Butterworth filters are filters that are monotonic in both the passband and the stopband. Their design involves determining the filter order N for a desired maximum stopband ripple, δ_s, by solving a set of formulaic equations. They have a distict frequency response. They are all pole filters. They can be completely defined by the N,δ_s and Ω_s/Ω_p parameters.

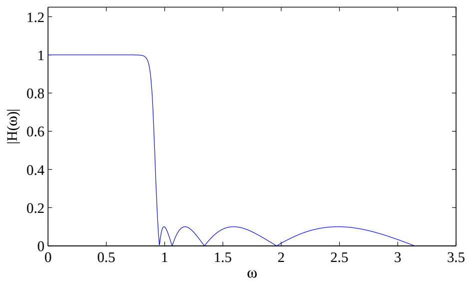
- **Chebyshev Type I Filters:** Type I Chebyshev filters are equiripple in the passband and monotonic in the stopband. They are also all pole filters.



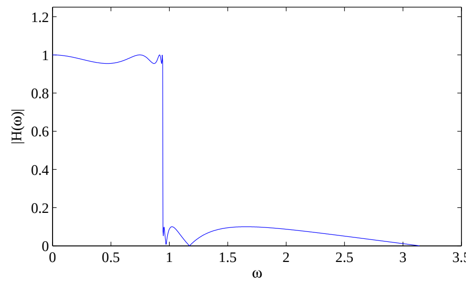
(a) Butterworth filter



(b) Type I Chebyshev filter



(c) Type II Chebyshev filter



(d) Elliptic filter

Figure 2.6: Frequency response representation of various IIR filters.

- **Chebyshev Type II Filters:** Type II Chebyshev filters are monotonic in the passband and equiripple in the stopband. They contain both zeros and poles. Their design involves solving a set of formulaic equations for required specifications to find N. If we consider the Chebyshev and the Butterworth filters, for the given specifications the designed Chebyshev filter has the lower order and transition bandwidth as compared to the Butterworth.

- **Elliptic Filters:** Also known as Cauer filters. Elliptic filters are equiripple in passband and stopband. They have both zeros and poles. Depending on even or odd N their magnitude characteristics have unique features. They are the most efficient filters as they yield the smallest order for the given specifications and have the smallest transition bandwidth. One drawback of elliptic filters is their nonlinear phase in the passband. For given specifications, N is determined using the analytic formulae.

- **Bessel Filters:** Bessel filters have linear phase in the passband. They are all pole filters. Their tranfer function is defined by the $N-th$ order Bessel polynomial.

Once the analog or continuous signal filters are designed, they are converted into the digital filters by one of the following two methods

- **Impulse Invariance Method:** The impulse invariance method of transformation involves the corresponding steps. From the frequency response $H(s)$ of the analog filter, the impulse response $h(t)$ is found. The continuous time impulse response $h(t)$ is sampled to get $h[n]$, the z transform of which gives the frequency response of the digital filter $H(z)$ and thus the digital filter is obtained. The conversion of the frequency response from the analog to the digital domain can also be done directly by using the relation between s and z, i.e. $z = e^{sT}$. This technique is only effective with bandlimited filters, i.e. lowpass and bandpass. Highpass and bandstop filters face a problem of aliasing.

- **Bilinear Transformation:** Bilinear transformation method involves direct substitution of $s = \frac{2}{T}\frac{1-z^-1}{1+z^-1}$ in $H(s)$ to get $H(z)$. This is a nonlinear mapping, thus there is no periodic replication and thus can be used for all filter types.

  Once the digital filter has been designed, it is usually the lowpass model which is then frequency transformed to get the respective filter type. [56].

2.4.2 Finite Impulse Response (FIR) Filters:

A digital filter that has a finite duration unit sample response is known as an FIR filter. The output of the FIR filter is a function of only the sequence of input values . It is usually also known as a nonrecursive filter, but there are a few exceptions. [56]. There are no feedback errors. They are stable filters with linear phase. They have a finite start up transient.

Direct design methods are used for FIR filters. With the given filter specifications, the design of FIR filters involves finding an approximate polynomial frequency response function and implementing it using certain algorithms. Certain constraints are placed on the filters during their design like causality, linear phase and that $h[n]$ is real. For FIR filters $b_k = h[k]$ for all M. It is a finite duration sequence, therefore it can be defined by N samples of its Fourier Transform. For that reason finding the N samples of its frequency response or the impulse response co-efficients an FIR filter can be designed. Some methods to design FIR filters are [49]:

Windowing: FIR filters are characterized by a finite length impulse response, which can be obtained by truncating an infinte length impulse response. This process of truncating an infinite duration sequence to get a finite response is known as *windowing* and the function used is known as the *window function*, represented as $w[n]$. Consider $H_d(e^{j\omega})$ is the required ideal frequency response, then the corresponding impulse response will be $h_d[n]$ [52]. However, $h_d[n]$ has infinite duration, we need a truncated version of this, which we can define as:

$$h[n] = \begin{cases} h_d[n] & o \leq n \leq N-1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.14}$$

We can then write this as:

$$h[n] = h_d[n]w[n]. \tag{2.15}$$

where $w[n]$ is the window function defined as:

$$w[n] = \begin{cases} 1 & o \leq n \leq N-1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.16}$$

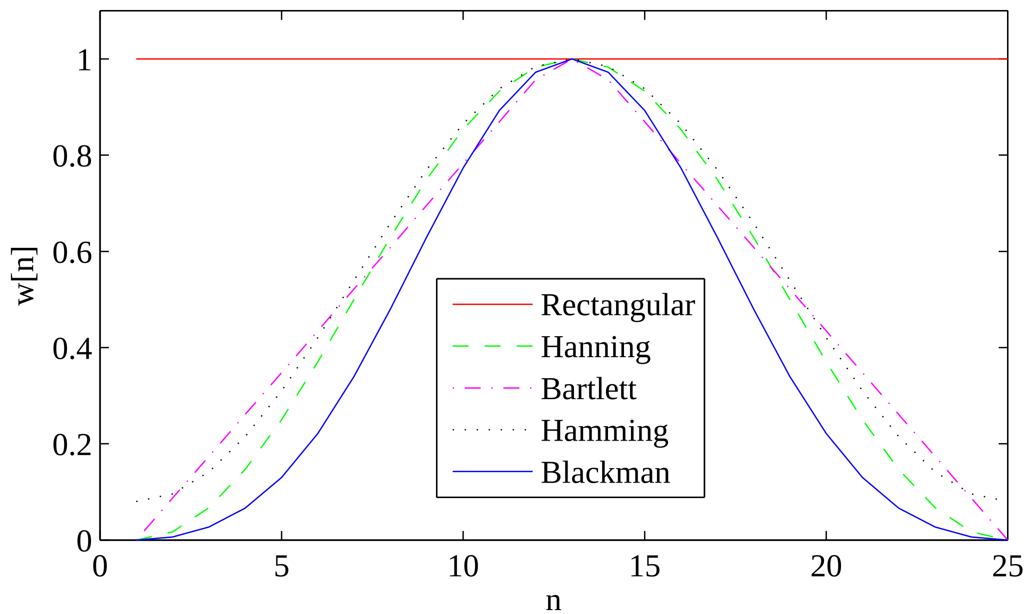
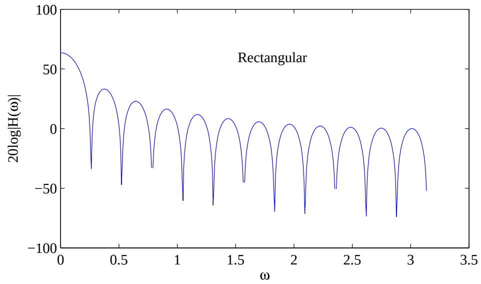


Figure 2.7: Time domain characteristics of typical windows

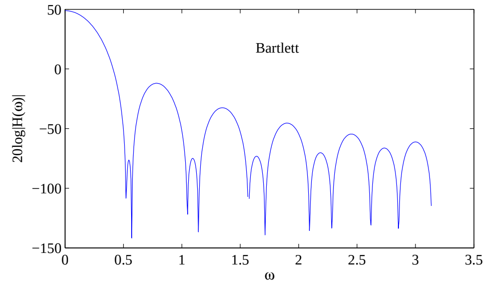
Different types of window functions or *windows* can be used depending on required output characteristics. Some commonly used windows are shown in Figure 2.8 and defined in Table 2.1. The rectangular window is also known as *boxcar* or *Dirchlet* window.

Table 2.1: Some commonly used window functions.

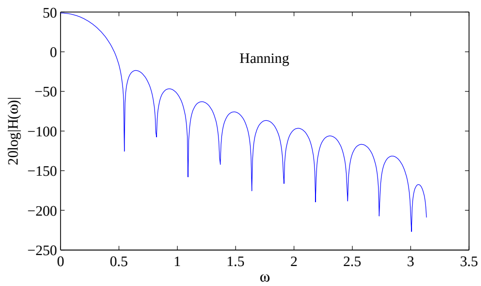
Type of Window function	Definition
Rectangular Window	$w[n] = 1, \qquad 0 \le n \le N-1$
Bartlett Window	$w[n] = \begin{cases} \frac{2n}{N-1} & o \le n \le \frac{N-1}{2} \\ 2 - \frac{2n}{N-1} & \frac{N-1}{2} \le n \le N-1 \end{cases}$
Hanning Window	$w[n] = \frac{1}{2}[1 - \cos(\frac{2\pi n}{N-1})], \qquad 0 \le n \le N-1$
Hamming Window	$w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), \qquad 0 \le n \le N-1$
Blackman Window	$w[n] = 0.42 - 0.5\cos(\frac{2\pi n}{N-1}) + 0.08\cos(\frac{4\pi n}{N-1}), \qquad 0 \le n \le N-1$



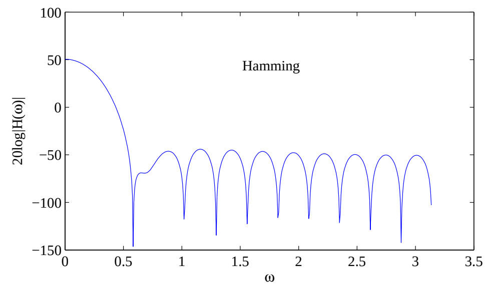
(a) Rectangular window



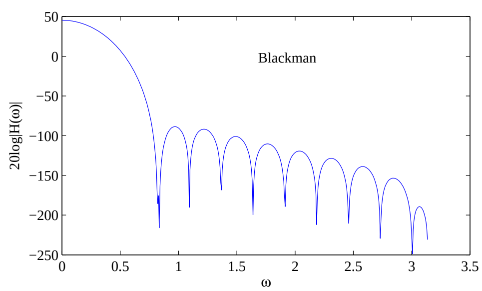
(b) Bartlett window



(c) Hanning window



(d) Hamming window



(e) Blackman window

Figure 2.8: Frequency domain characteristics of some commonly used windows.

Frequency Sampling: Since a finite duration sequence can be represented by the discrete fourier transform, an FIR filter can be represented by the frequency samples. We define the filter in terms of the samples of one period of the required frequency response.

$$H\tilde{(}k) = H_d(e^{j(2\pi/N)k}), \qquad k = 0, 1, \cdots, N-1 \tag{2.17}$$

The frequency response so obtained can be smoothened by interpolation. This method is highly beneficial for designing narrow band frequency selective filters [52].

Equiripple Approximation: The above two methods suffer from lack of control of the critical frequencies ω_p and ω_s, even though they are comparatively easier methods to design FIR linear phase filters. This method is a formulation of the Chebyshev approximation problem. The weighted approximation error between the desired frequency response and the actual frequency response is distributed equally across the passband and the stopband of the filter, which in turn reduces the maximum error. These filters have ripples in both the passband and the stopband. They are equiripple except at $\omega = 0, \pi$. The solution to the Chebyshev problem was given as applying the Alternation theorem to it. According to it, the error function alternates with ω in the passband and the stopbands. For lowpass and highpass filters the maximum error occurs at the edges of the passband and the stopband. This was proposed by Parks and McClellan who implemented it using the Remez exchange algorithm, because of which this method of designing filters is known as the Parks McClellan method. It is an iterative algorithm that successfully improves the location of the extreme frequencies [50].

One of the main considerations while designing Filter bank systems is whether to choose an FIR or an IIR filter. IIR filters are computationally advantageous while FIR filters offer phase linearity and greater adjustability in filter characteristics. IIR filters accomplish superior amplitude response by sacrificing phase.

2.5 Multirate Filter Banks

Now that we have familiarized ourselves with filters and their operation it is time to apply this knowledge in developing filter banks. Filter banks are multirate systems, which typically consist of a bank of filters on the transmitter and receiver side respectively with a bunch of sample rate converters and multiplexers to form a multicarrier system.

2.5.1 Sample Rate Converters

Multirate systems in general employ two types of sample rate conversion decimation and interpolation. [57]. Thus the converters can be classified as follows:

- **Upsampler:** An upsampler is a sampling rate expander. It increases the sampling rate of a signal by upsampling, i.e., inserts an integral number of samples (zeroes) between consecutive samples of the input signal. It is known by various names such as expander and interpolator and can be represented diagrammatically as in Figure 2.9.

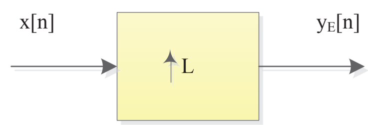


Figure 2.9: Block diagram of an expander.

The input output relation is given as:

$$y_E[n] = \begin{cases} x[n/L], & n = mL \\ 0, & \text{otherwise} \end{cases} \tag{2.18}$$

where L is the upsampling factor and m is an integer, i.e. n should be an integral multiple of L. [51, 49]. In this case $(L-1)$ zeroes will be inserted between every two samples of $x[n]$ to get $y_E[n]$, as shown in Figure 2.10 which gives the time domain representation of the upsampling process. The expander when used with a filter performs interpolation. There is no loss of information and it is always possible to recover input.

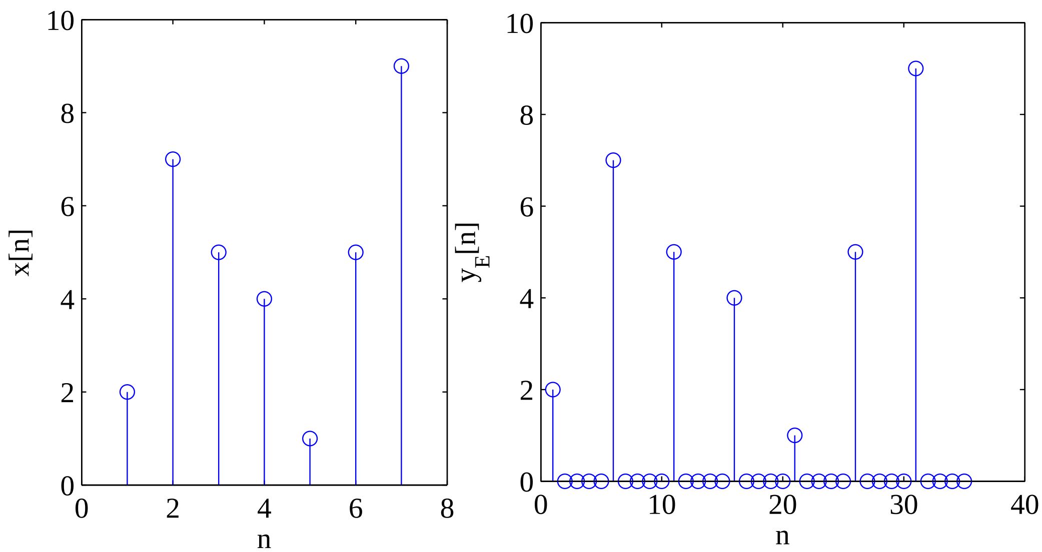


Figure 2.10: Time domain representation of the process of upsampling.

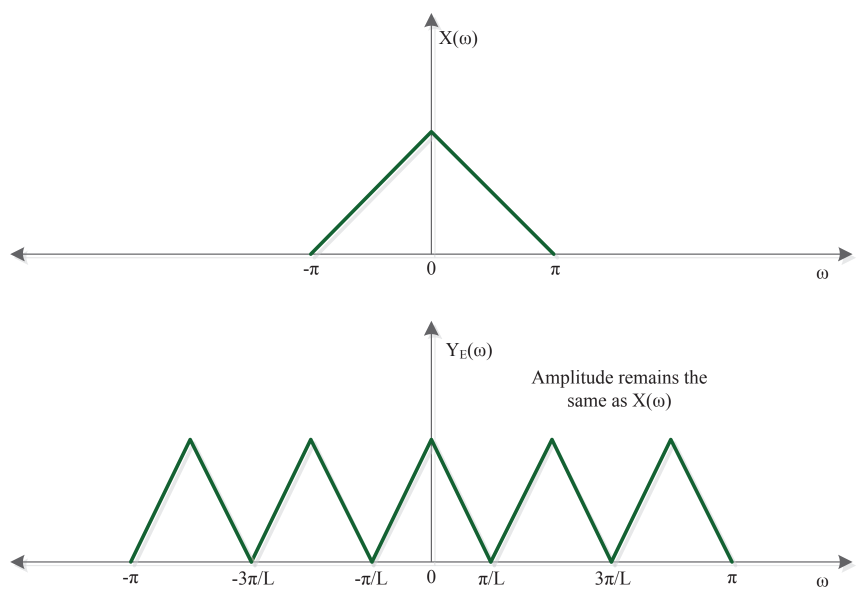


Figure 2.11: Frequency domain representation of the upsampling process.

If we look at the effect of the expander in the frequency domain, represented in Figure 2.11, we see that multiple images of the original signal are formed, thus it creates an imaging effect. Thus the spectrum gets contracted by the factor L as a result. An interpolator can be defined as an upsampler that is followed by a filter that passes one of the images and stops the others.

- **Downsampler** A downsampler on the other hand is a sampling rate compressor. It decreases the sampling rate of a signal by downsampling, i.e. it retains only one sample out of an integral number of samples of the input signal. It is also known as a compressor or subsampler usually represented as in Figure 2.12.

Figure 2.12: Block diagram of a decimator.

The process is represented as follows:

$$y_D[n] = x[Mn] \tag{2.19}$$

where M is the downsampling factor. [51, 49]. Here for every M samples only one sample is retained in the output. So we are actually losing some samples and thus information, as shown in Figure 2.13.

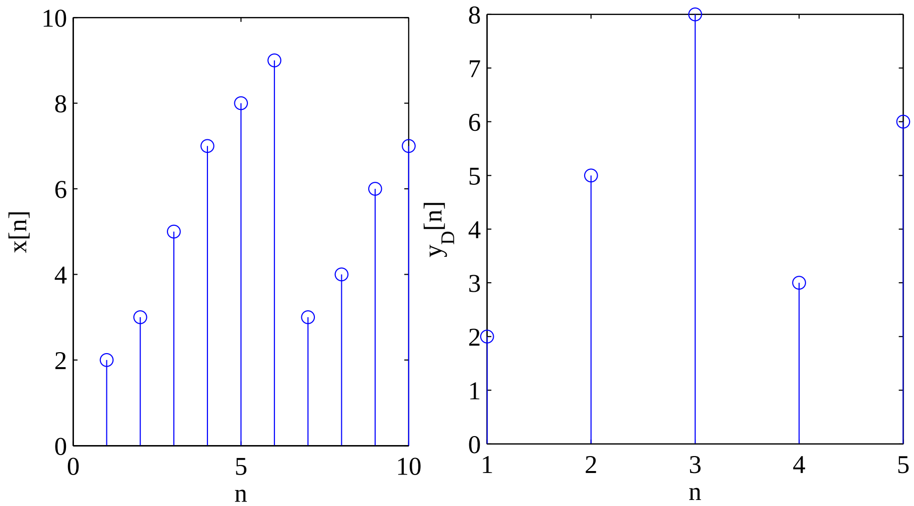


Figure 2.13: Time domain representation of the process of downsampling.

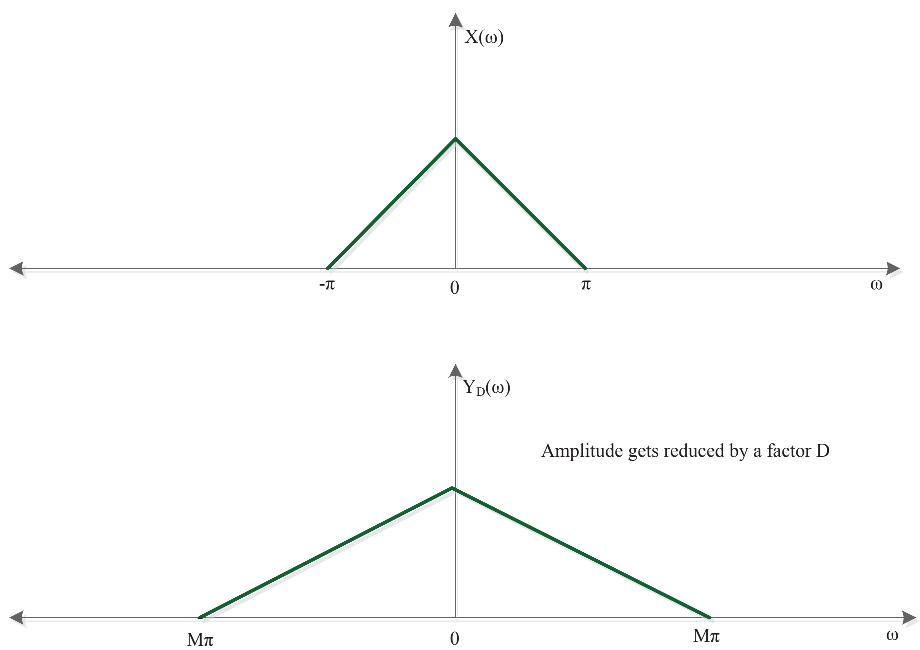


Figure 2.14: Frequency domain representation of the downsampling process.

Downsamplers perform decimation, which may also results in aliasing, because of which we may not be able to recover the input. If we look at the effect of decimation in the frequency domain, Figure 2.14, we see that the spectrum has expanded by a factor M and it may result in aliasing. Therefore decimation usually includes lowpass filtering prior to downsampling to avoid aliasing.

2.5.2 Filter banks

A filter bank is a system that comprises of a group of filters which process a common input or result in a common output. The filter banks either break down an input signal to form subband component signals or combine the subband signals to form the output signal. The decomposition process performed by the filters is known as *analysis* while the reconstruction process in known as *synthesis*. Based on the operation carried out, there are two types of filter banks:

- **Analysis Filter bank** This section of filters decomposes a signal into its subband components. It is also known as separating filter bank. It is usually represented as $h_k[n]$ or $H_k(z)$.
- **Synthesis Filter bank** This section of filters is responsible for reconstructing an approximation to the onput signal from the subband components. It is known as combining filter bank. It is usually represented as $f_k[n]$ or $F_k(z)$.

Filter banks systems can be implemented by two basic architectures

2.5.3 Transmultiplexers

A transmultiplexer is a multiple-input, multiple output (MIMO) system. It consists of a filter bank on the transmitter and a receiver side called the synthesis and analysis bank respectively. In a transmultiplexer N inputs are combined to get a single aggregate signal that is transmitted and the received signal is then treated to get back the original

N outputs. The N inputs to the transmitter are obtained by sampling a continuous signal and demultiplexing consecutively. Thus we have an N band system. The N signals are then upsampled by a factor M, which creates M images of each signal. The signals are then passed through a bank of filters. The filters are so designed that they are all centered at different frequencies and thus select and pass the image of the respective signal at that frequency. All the signals are added to give the aggregate signal that is transmitted. If we look at the transmitted signal we see that it is nothing but the frequency division multiplexed form of the input signals, as each signal is allocated equal parts of the available bandwidth, the only difference here being that the filtered signals of adjacent bands are allowed to overlap to a certain extent.

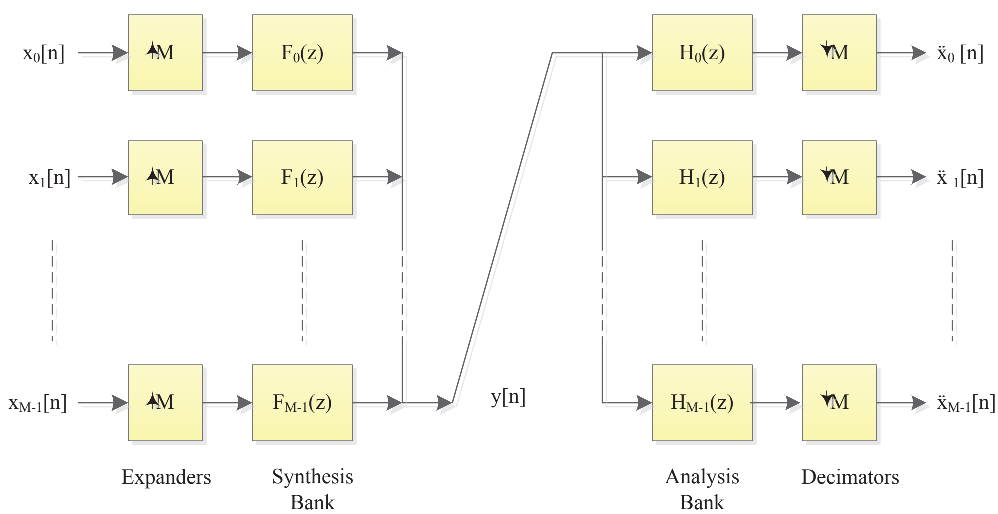


Figure 2.15: Schematic illustration of a generic transmultiplexer.

The received aggregate signal is then passed through a bank of filters that effectively separate out the signal to give N signals, which are then downsampled to get the original N signals. The filters on the receiver side or the analysis filters have the same center frequencies as their transmitter counterparts and thus filter out the signal in that frequency getting back the image of the original N^{th} signal. The N signals are now multiplexed to give the output signal. The signal is in discrete format and is converted to get the continuous signal [51, 57]. In the given Figure 2.15, we set $N = M$.

Thus a transmultiplexer converts from the time division multiplexing (TDM) format to frequency division multiplexing (FDM) and then again to TDM. They are used for data transmission. [58].

2.5.4 Sub-band systems

A sub-band system, as shown in Figure 2.16, is a single-input single-output system. In a sub-band system the analysis filter bank is on the transmitter side and the synthesis bank on the receiver side. The input to the sub-band system is divided into N parts by the analysis filter bank. Here also, every filter has different center frequencies and thus filters output different components of the input signal. Each component is then downsampled and the decimated signal is then coded and transmitted.

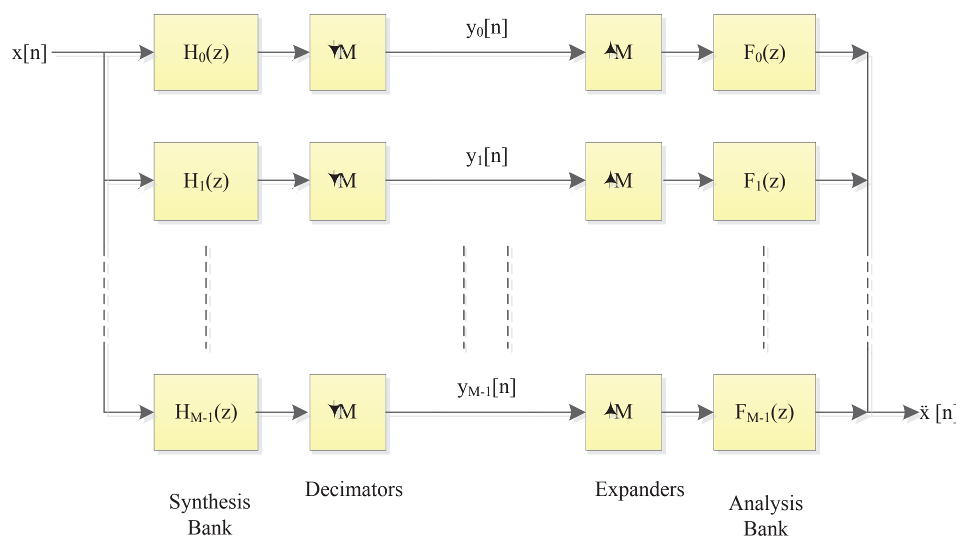


Figure 2.16: Schematic illustration of a generic sub-band system.

The receiver decodes the received signals, which then undergo upsampling which produces images of the signal while restoring the original sample rate. The signals are then treated by the synthesis filter bank, which eliminates the images and are then added to get the output version of the original signal [51, 57].

Sub-band system on the other hand converts from the FDM format to the TDM format and then to FDM. They are used in speech and image coding. [58]. These two systems are complementary in nature. In literature, filter bank subband systems are referred to as filter banks whereas the transmultiplexers are referred to as filter bank transmultiplexers.

Depending on the relation between the number of inputs or channels N and the sampling factor M, filter banks are classified as:

- **Critically sampled Filter banks** When $N = M$ then the system is said to be critically sampled. They are also known as maximally decimated filter banks. These filter banks are the most preferred as they preserve information and are not data expensive. [55, 28].

- **Oversampled Filter banks** For subband systems they are said to be oversampled when $N > M$ and the oversampling ratio is given as $L = N/M$. Transmultiplexers are said to be oversampled when $M > N$ and the oversampling ratio is given as $L = M/N$. Oversampled filter banks provide greater design freedom and noise immunity, but are computationally complex. [27, 28]

- **Undersampled Filter banks** Subband systems are said to be undersampled when $N < M$ and transmultiplexers are said to be undersampled when $M < N$. It is the converse of the oversampled filter banks.

The bank of filters can be implemented independently, but are preferably employed as modulated filter banks. When all the filters are frequency shifted versions of a lowpass prototype then they are known as modulated filter banks. They are obtained by multiplying the prototype with a modulating function. As a result we get a group of bandpass filters centered at different frequencies. They are also known as uniform filter banks as all of them have the same bandwidth, magnitude response and they are equally spaced spectrally. Depending on the type of modulating function used and a few other characteristics filter banks are classified into different types. Some of them are discussed in the following sections.

2.5.5 Paraunitary Perfect Reconstruction Filter Banks

Paraunitary filter banks are filter banks that satisfy the paraunitary property. Consider the analysis bank is described by an $M \times 1$ transfer matrix $\mathbf{H}(z)$, synthesis bank by a $1 \times M$ transfer matrix $\mathbf{F}^T(z)$. These transfer matrices can also be expressed as:

$$\mathbf{H}(z) = \mathbf{E}(z^M)\mathbf{e}(z) \tag{2.20}$$

$$\mathbf{F}^T(z) = z^{-(M-1)}\tilde{\mathbf{e}}(z)\mathbf{R}(z^M) \tag{2.21}$$

where $\mathbf{E}(z)$ and $\mathbf{R}(z)$ are the polyphase matrices of the analysis and synthesis banks respectively. If $\mathbf{E}(z)$ satisfies the paraunitary and lossless property then the filter bank so obtained satisfies the perfect reconstruction property.

A $p \times r$ causal transfer matrix $\mathbf{H}(z)$ is lossless when each element $H_{km}(z)$ is stable and $H(e^{j\omega})$ is unitary. The unitary property is given as:

$$\mathbf{H}^H(e^{j\omega})\mathbf{H}(e^{j\omega}) = d\mathbf{I}_r, \qquad \forall\omega, \tag{2.22}$$

when $p = r = 1$ it becomes the allpass property. For rational tranfer functions the unitary property becomes:

$$\tilde{\mathbf{H}}(z)\mathbf{H}(z) = d\mathbf{I}, \qquad \forall z, \tag{2.23}$$

which is the paraunitary property. A causal, stable paraunitary system is a lossless system. The columns of $\mathbf{H}(z)$ are mutually orthogonal, i.e.,

$$\tilde{\mathbf{H}}_k(z)\mathbf{H}_m(z) = 0 \qquad for \quad k \neq m. \tag{2.24}$$

Moreover, each column is a set of p power complementary filters:

$$\tilde{\mathbf{H}}_k(z)\mathbf{H}_k(z) = d. \tag{2.25}$$

In a square matrix, every row is power complementary and any pair of rows orthogonal. The properties of paraunitary systems are as follows:

1. When $p = r$ $det\mathbf{H}(z)$ is allpass, i.e. it is a constant. If $\mathbf{H}(z)$ is FIR then

   $$det\mathbf{H}(z) = az^{-}K, \qquad K \geq 0, \quad a \neq 0 \tag{2.26}$$

   its determinant is a delay. Here *det* stands for the determinant.

2. Submatrices of $\mathbf{H}(z)$ are also paraunitary. Consider a $M \times 1$ transfer matrix of a filter bank.

   $$\mathbf{H}(z) = [H_0(z) \quad H_1(z) \cdots H_{M-1}(z)]^T \tag{2.27}$$

   The paraunitary property implies the power complementary property. Therefore the channel filters are power complementary.

   $$\sum_{k=0}^{M-1} |H_k(\exp j\omega)|^2 = c, \qquad \forall\omega \tag{2.28}$$

When $\mathbf{E}(z)$ is paraunitary, it implies

$$\mathbf{E}^{-1}(z) = \tilde{\mathbf{E}}(z)/d \qquad \forall z \quad and \quad d > 0 \tag{2.29}$$

then $\mathbf{R}(z)$ becomes

$$\mathbf{R}(z) = cz^{-k}\tilde{\mathbf{E}}(z) \tag{2.30}$$

which satisfies the perfect reconstruction condition. According to this relation, synthesis filter bank is the paraconjugate of the analysis filter bank. Since the paraconjugate is the inverse of a paraunitary filter matrix it is exactly what we need for perfect reconstruction. Therefore each synthesis filter is simply the flipped and conjugated version of the analysis filter.

$$f_k(n) = ch_k{}^*(L - n), \qquad 0 \leq k \leq M - 1, \tag{2.31}$$

here $L = M - 1 + MK$. This equation implies

$$F_k(z) = cz^{-L}\tilde{H}_k(z), \qquad 0 \leq k \leq M-1, \tag{2.32}$$

which is the hermitian image of $H_k(z)$. When $\mathbf{H}(z)$ is FIR then $\tilde{\mathbf{H}}(z)$ is also FIR, which is nothing but the tranfer function of the synthesis filters. Therefore in this case the synthesis and analysis filters have the same length. FIR analysis and synthesis filters in paraunitary filter banks have the same amplitude response. Paraunitary filter banks are allowed to scale and/or delay the input signal. In a paraunitary system, every filter has the same energy.

2.5.6 Linear Phase Perfect Reconstruction Filter Banks

Filter banks in which the analysis filters $H_k(z)$ are constrained to have linear phase are Linear phase filter banks. In order to design linear phase perfect reconstruction filter banks it is necessary to discard the power complementary property. Thus we are also forgoing the paraunitary property. For perfect reconstruction, the *distortion transfer function* $T(z)$ must be a delay. This is possible when the condition

$$det\mathbf{E}(z) = \alpha z^{-}K, \qquad K \geq 0, \quad \alpha \neq 0 \tag{2.33}$$

is satisfied, where *det* is the determinant, α is a constant and K is an integer. Linear phase perfect reconstruction filter banks can be generated by nontrivial analysis filters which means that they are not just the sum of two delays.

2.5.7 Exponentially Modulated Filter Banks

Exponentially modulated filter banks (EMFB) are obtained by exponential modulation of the prototype filter. Due to exponential modulation, the resulting analysis and syntheis filters have single sided magnitude responses that divide the whole frequency range uniformly. [12]. Exponentially modulated filter banks can be implemented independently or derived from a combination of cosine modulated filter banks and sine modulated filter bank. They can also be implemented using extended lapped transforms (ELT) [30].

One of the independent implementation of exponential filter banks is as follows Consider the prototype filter $H_0(z)$. The analysis filters $H_k(z)$ are related to $H_0(z)$ as $H_k(z) = H_0(z{W_M}^k)$. Where $W_M = e^{-j2\pi/M}$. Thus $h_k[n]$ are obtained by exponential modulation of $h_0[n]$.

$$h_k[n] = h_0[n]\exp\left(j2\pi kn/M\right) \tag{2.34}$$

thus the coefficients of $h_k[n]$ are complex even when $h_0[n]$ is real, which results in the possibility of the output being complex even when the input was real. These are also known as complex modulated or DFT filter banks when they are implemented using the DFT transform. Thus DFT filter banks are a type of exponentially modulated filter banks which are implemented using polyphase and FFT structures. A point of note here is that even though DFT filter banks can be called a type of EMFBs, they differ from EMFBs in the way they are sampled and their channels are stacked.

The more popular implementation of exponentially modulated filter banks is given below. Consider $h_p(n)$ as the lowpass prototype filter. Then the synthesis and analysis filter banks for a $2\times$ oversampled filter bank are defined as

$$f_k^e(n) = \sqrt{\frac{2}{M}}h_p(n)\exp\left(j(n+\frac{M+1}{2})(k+\frac{1}{2})\frac{\pi}{M}\right) \tag{2.35}$$

$$h_k^e(n) = \sqrt{\frac{2}{M}}h_p(n)\exp\left(-j(N-n+\frac{M+1}{2})(k+\frac{1}{2})\frac{\pi}{M}\right) \tag{2.36}$$

where $n = 0, 1, \cdots, N$ and $k = 0, 1, \cdots, 2M-1$ [12, 31, 32]. If we consider an oversampled exponentially modulated filter bank it can be implemented using cosine and sine modulated filter banks as follows:

$$f_k^e(n) = \begin{cases} f_k^c(n) + jf_k^s(n), & k \in [0, M-1] \\ -f_{2M-1-k}^c(n) + jf_{2M-1-k}^s(n), & k \in [M, 2M-1] \end{cases} \tag{2.37}$$

$$h_k^e(n) = \begin{cases} h_k^c(n) - jh_k^s(n), & k \in [0, M-1] \\ -h_{2M-1-k}^c(n) - jh_{2M-1-k}^s(n), & k \in [M, 2M-1] \end{cases} \tag{2.38}$$

where:

$$f_k^c(n) = \sqrt{\frac{2}{M}} h_p(n) \cos\left((n + \frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M}\right) \tag{2.39}$$

$$f_k^s(n) = \sqrt{\frac{2}{M}} h_p(n) \sin\left((N - n + \frac{M+1}{2})(k + \frac{1}{2})\frac{\pi}{M}\right) \tag{2.40}$$

where $k = 0, 1, \cdots, M-1$ and $n = 0, 1, \cdots, 2KM-1$. The analysis filters are obtained as:

$$h_k^c(n) = f_k^c(N-n), \qquad h_k^s(n) = f_k^s(N-n). \tag{2.41}$$

The cosine and sine modulated filters are related as:

$$h_k^s(n) = (-1)^{k+K} f_k^c(n), \qquad f_k^s(n) = (-1)^{k+K} h_k^c(n). \tag{2.42}$$

2.5.8 Cosine Modulated Filter Banks

Filter banks that are based on cosine modulation are known as cosine modulated filter banks. In this system, the analysis filters are obtained from the prototype filter by cosine modulation. The frequency responses of the filters are uniformly shifted versions of the prototype. The advantages of this method are: the cost of implementation reduces to that of one filter, the prototype plus the modulation overhead. The parameters to be optimized also reduce as only the prototype needs to be optimized. They can be implemented as near perfect reconstruction (NPR) or perfect reconstruction (PR) systems. The NPR systems are also known as pseudo systems. References [59, 60] deal with pseudo QMF filter banks in detail. In the NPR systems, the design of the analysis and synthesis filters is such that only adjacent channel aliasing is cancelled and the distortion function is not a pure delay. On the other hand the PR systems are paraunitary systems that retain all the ease of the NPR system with perfect reconstruction. References [26, 11, 61] discuss PR cosine modulated filter bank systems.

One way of implementing cosine modulated filter banks is by modifying exponential modulated filter banks. In the above section we defined the implementation of exponential filter banks and also the possibility of complex output even when the input is real. Cosine modulation on the other hand results in filters with real co-efficients. Using exponential modulation, we design $2M$ complex filters and combine them appropriately to get real filters. Consider a prototype filter $P_0(z)$, whose impulse response is real and magnitude response is symmetric. It is a lowpass filter with $f_c = \pi/2M$. The polyphase components of $P_0(z)$ are $G_k(z), 0 \leq k \leq 2M-1$. The other filters are given as:

$$P_k(e^{jw}) = P_0(e^{j(\omega - k\pi/M)}), \tag{2.43}$$

the magnitude responses of these filters are the shifted versions of the prototype. Now if we consider:

$$U_k(z) = c_k P_0(zW^{k+0.5}) \tag{2.44}$$

$$V_k(z) = {c_k}^* P_0(zW^{-(k+0.5)}) \tag{2.45}$$

where $W = W_{2M} = e^{-j2\pi/2M} = e^{-j\pi/M}$. Now we define the analysis filters as:

$$H_k(z) = a_k U_k(z) + {a_k}^* V_k(z), \qquad 0 \leq k \leq M-1. \tag{2.46}$$

The prototype is of the form:

$$P_0(z) = \sum_{n=0}^{N} p_0[n] z^{-n}, \tag{2.47}$$

which is an N^{th} order FIR filter, which then gives the FIR analysis filters:

$$H_k(z) = \sum_{n=0}^{N} h_k[n] z^{-n}, \qquad 0 \leq k \leq M-1 \tag{2.48}$$

where $h_k[n]$ are real, as the coefficients of $P_0(z)$ are real which implies that the co-efficients of $V_k(z)$ and $U_k(z)$ are conjugates of one another. This method results in an NPR system.

Cosine modulated systems can be implemented using discrete cosine tranform (DCT) and discrete sine transform (DST) matrices when the filter length $N+1$ and the number of channels M are related as:

$$N + 1 = 2lM, \qquad l \quad \text{is an integer} \tag{2.49}$$

The DCT matrix is represented as $\mathbf{C}$ and DST as $\mathbf{S}$. There are four types of these matrices but here we only consider *Type 4*. They are $M \times M$ matrices whose elements are defined as:

$$c_{kn} = \sqrt{\frac{2}{M}} \cos\Big(\frac{\pi}{M}(k+0.5)(n+0.5)\Big), \tag{2.50}$$

$$s_{kn} = \sqrt{\frac{2}{M}} \sin\Big(\frac{\pi}{M}(k+0.5)(n+0.5)\Big), \tag{2.51}$$

The matrices $\mathbf{C}$ and $\mathbf{S}$ are real, symmetrical and orthonormal. They are related as:

$$\mathbf{C} = \mathbf{\Gamma S J} \tag{2.52}$$

where:

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots (-1)^{M-1} & \end{bmatrix} \tag{2.53}$$

and $\mathbf{J}$ is the anti-diagonal or reversal matrix.

The cosine modulation matrix $\mathbf{T}$ in the above method can be defined using DCT and DST matrices as:

$$T = [\mathbf{A}_0 \mathbf{A}_1], \tag{2.54}$$

where $\mathbf{A}_0$ and $\mathbf{A}_1$ are $M \times M$ matrices defined as:

$$\mathbf{A}_0 = \sqrt{M}\mathbf{\Lambda}_c(\mathbf{C} - \mathbf{\Gamma S}), \quad \mathbf{A}_1 = -\sqrt{M}\mathbf{\Lambda}_c(\mathbf{C} + \mathbf{\Gamma S}) \quad \text{when } m \text{ even} \tag{2.55}$$

$$\mathbf{A}_0 = \sqrt{M}\mathbf{\Lambda}_s(\mathbf{C} + \mathbf{\Gamma S}), \quad \mathbf{A}_1 = \sqrt{M}\mathbf{\Lambda}_s(\mathbf{C} - \mathbf{\Gamma S}) \quad \text{when } m \text{ odd} \tag{2.56}$$

where $\mathbf{\Lambda}_c$ and $\mathbf{\Lambda}_s$ are:

$$[\mathbf{\Lambda}_c]_{kk} = \cos(\pi(k+0.5)m), \quad [\mathbf{\Lambda}_s]_{kk} = \sin(\pi(k+0.5)m) \tag{2.57}$$

of size $M \times M$.

From this a PR filter bank can be implemented as follows:

$$\mathbf{h}(z) = \mathbf{T}\begin{bmatrix}\mathbf{g}_0(z^{2M}) & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_1(z^{2M})\end{bmatrix}\begin{bmatrix}\mathbf{e}(z) \\ z^{-M}\mathbf{e}(z)\end{bmatrix} = \mathbf{T}\begin{bmatrix}\mathbf{g}_0(z^{2M}) \\ z^{-M}\mathbf{g}_1(z^{2M})\end{bmatrix}\mathbf{e}(z) \tag{2.58}$$

where $\mathbf{e}(z)$ is the delay chain vector:

$$\mathbf{e}(z) = [1z^{-1}\cdots z^{-(M-1)}]^T \tag{2.59}$$

and:

$$[\mathbf{g}_0(z)]_{kk} = G_k(-z), \quad [\mathbf{g}_1(z)]_{kk} = G_{M+k}(-z). \tag{2.60}$$

Now cosider $\mathbf{h}(z) = \mathbf{E}(z^M)\mathbf{e}(z)$, comparing we get:

$$\mathbf{E}(z) = \mathbf{T}\begin{bmatrix}\mathbf{g}_0(z^2) \\ z^{-1}\mathbf{g}_1(z^2)\end{bmatrix} \tag{2.61}$$

which is:

$$\mathbf{E}(z) = [\mathbf{A}_0 \quad \mathbf{A}_1]\begin{bmatrix}\mathbf{g}_0(z^2) \\ z^{-1}\mathbf{g}_1(z^2)\end{bmatrix} \tag{2.62}$$

For perfect reconstruction $\mathbf{E}(z)$ must be paraunitary, [28],[51] it is paraunitary if $G_k(z)$ if pairwise power complementary:

$$\tilde{G_k}(z)G_k(z) + \tilde{G}_{M+k}(z)G_{M+k}(z) = \alpha, \qquad 0 \le k \le M-1, \quad \alpha > 0. \tag{2.63}$$

Now let us look at one of the one most common and popular ways to implement cosine modulated filter banks. The prototype filter chosen is an FIR lowpass filter, $h_p(n)$. The analysis filters are then defined as:

$$h_k^c(n) = 2h_p(n)\cos\left((2k+1)\frac{\pi}{2M}(n-\frac{N}{2}) + (-1)^k\frac{\pi}{4}\right) \tag{2.64}$$

The synthesis filters are given as:

$$f_k^c(n) = 2h_p(n)\cos\left((2k+1)\frac{\pi}{2M}(n-\frac{N}{2}) - (-1)^k\frac{\pi}{4}\right) \tag{2.65}$$

for $k = 0, \cdots, M-1$ and $n = 0, \cdots, N$. We see that:

$$f_k^c(n) = h_k^c(N-n) \tag{2.66}$$

i.e., by flipping and shifting the analysis filters we obtain the synthesis filters. [9, 29, 24].

2.5.9 Modified Discrete Fourier Transform Filter Banks

One of the earliest, efficient implementation of filter banks was achieved by complex modulation. It is known as DFT filter bank and was accomplished by means of polyphase filters and IDFT/DFT transforms. They are highly popular due to their computational efficiency, but suffer from aliasing due to subsampling. To overcome this drawback an improved version of the DFT filter banks has been designed which is known as a modified DFT (MDFT) filter bank [62]. This is achieved by oversampling and modifying the subsampling stages of the original DFT filter banks [32, 63, 64]. MDFT filter banks have linear phase in both the analysis and synthesis parts [62]. The prototype chosen is normally a symmetrical, zero phase, lowpass filter [64].

The subband signals acquired after decimation by M are further subjected to a two step decimation. Therefore each of the $2M$ subbands is decomposed into complex valued even and odd polyphase components. The real part of one polyphase component and the imaginary part of another are taken in each subband and the order alternates from subband to subband. On the syntheis side the reconstruction proceeds first from the binary polyphase components and is then processed by the synthesis filter bank of the oversampled DFT [63].

If we consider an MDFT transmultiplexer having even number of channels, we see that the real and imaginary parts of the signals to the transmultiplexer are offset, and their assignment to the channels also alternates. Thus the structure of the MDFT systems is planned so as to cancel crosstalk [33].

Based on the method of selection of the real and imaginary parts of the subband polyphase components and the modulation phase MDFT filter banks are classified as Type I and Type II.

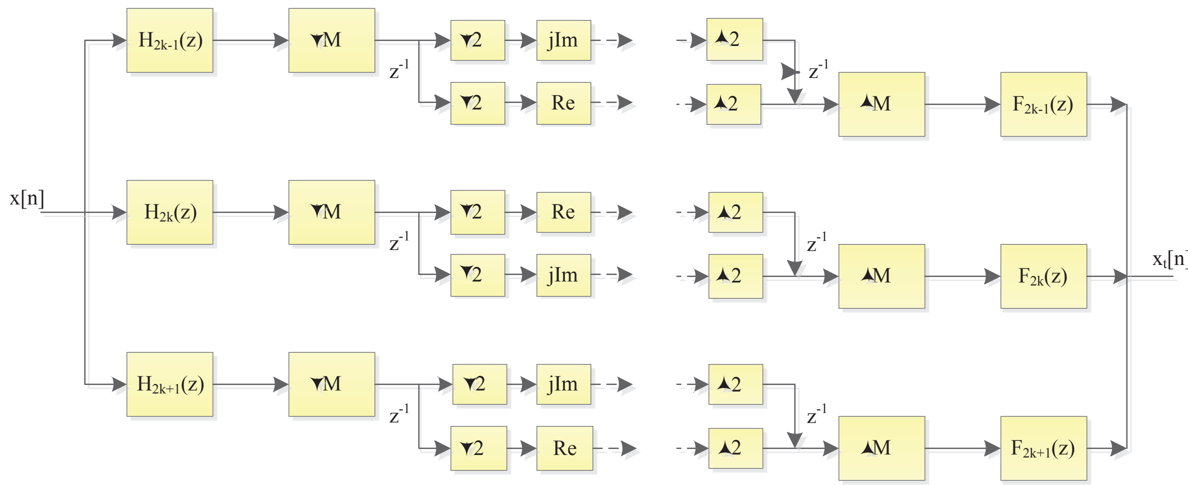


Figure 2.17: Diagrammatic representation of the Type I MDFT filter banks.

In a *Type I* MDFT filter bank, shown in Figure 2.17, the analysis and synthesis filters are given as:

$$h_k[n] = \sqrt{2} h_p[n] \exp\left(\frac{j2\pi k}{2M}(n - \frac{D}{2})\right) \tag{2.67}$$

$$f_k[n] = \sqrt{2} h_p[n] \exp\left(\frac{j2\pi k}{2M}(n - \frac{D}{2})\right) \tag{2.68}$$

here D is system delay and $k = 0, 1, \cdots, 2M - 1$. After being processed by the analysis filters each subband signal is divided into odd and even polyphase components. The real part of the even subband polyphase component and the imaginary part of the odd polyphase component is taken for even indexed subband signals and vice versa for odd indexed subband signals.

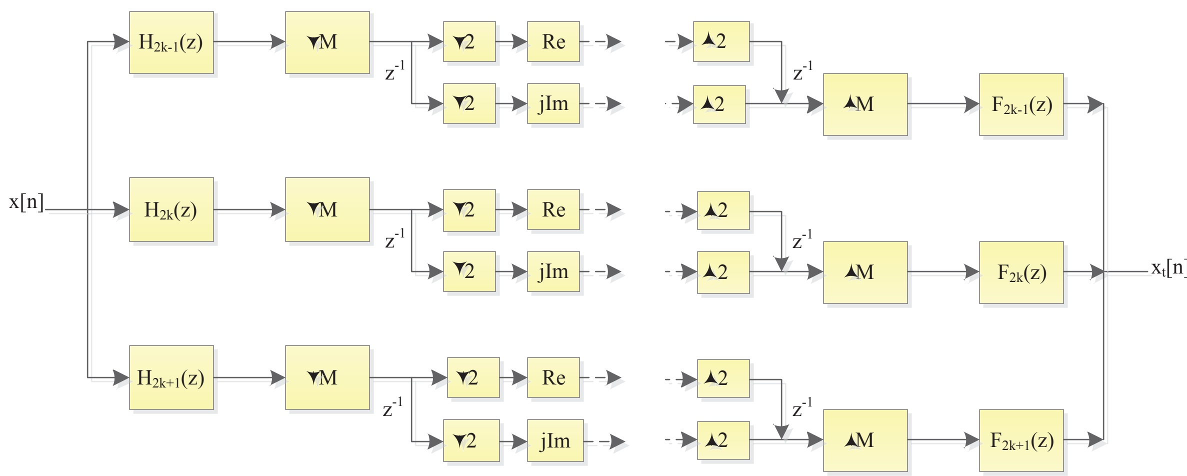


Figure 2.18: Diagrammatic representation of the Type II MDFT filter banks.

Conversely, a *Type II* MDFT filter bank, shown in Figure 2.18, filters are defined as:

$$h_k[n] = \sqrt{2}h_p[n] \exp\left(\frac{j2\pi k}{2M}(n - \frac{D+M}{2})\right) \tag{2.69}$$

$$f_k[n] = \sqrt{2}h_p[n] \exp\left(\frac{j2\pi k}{2M}(n - \frac{D-M}{2})\right) \tag{2.70}$$

where $k = 0, 1, \cdots, 2M - 1$. Here the real part in the direct branch of the subband and the imaginary part in the delayed branch are chosen. They find application in subband coding and block filtering [65].

2.6 Cognitive Radio

When the idea of spectrum pooling first came into existence, the need for a device that could realize that idea emerged. This has led to the development of the cognitive radio.

A radio that adapts to its environment and tranmits data accordingly, while making decisions on the fly is known as a Cognitive Radio (CR). As the term 'cognitive' indicates a cognitive radio is capable of perceiving, processing and reasonably deciding the course of action to take in any kind of working environment. Cognitive radios conform their behaviour according to external factors and make independent decisions while following some predefined guidelines. One of the main functions of the cognitive radio is to sense the spectrum for activity. This sensing can be through active interaction with the other uses or may be passive, in which case it involves some decision making. While detecting the spectrum for activity, if it recognizes a whitespace it transmits while making sure that the primary transmission is not corrupted. This is easier said than done as the primary transmission is not constant and changes with time, at which point the cognitive features of the radio come into action [66].

The term cognitive radio was coined by Joseph Mitola III [67]. Based on operation requirements and conditions, an SDR that can reconfigure its operating parameters through perception is known as cognitive radio [68]. In simple words, cognitive radio can be defined as a radio that can think for itself. The spectrum access is dealt on a negotiated and opportunistic basis among the users through cognitive radio technology. They are capable of operating in a range of frequecny bands through different modulation schemes. The features that make cognitive radios really special are that they are capable of interoperability and dynamic spectrum access (DSA).

Cognitive radios have the ability to communicate with radios working on different standards and protocols. Since cognitive radios are usually implemented using software defined radios they can nimbly adopt any configuration and arrange themselves to communicate with any kind of radio. They can also act as a interface between two incompatible radio systems. They are expected to operate effectively across a broad range of heterogenous environments.

The whitespaces in a licensed primary spectrum can be utilized by a secondary user who ‘borrows’ the spectrum while ascertaining that they are not interfering in any way with the primary transmission. This process of spectrum usage is known as Dynamic Spectrum

Access. With cognitive radios this can be achieved. Some scenarios in which cognitive radios can be used to improve spectrum efficiency are A primary user can employ the cognitive radio technology internally within their own network to increase its efficiency. They make the existense of secondary markets possible, where the primary licensee can enter an agreement with a third party to share spectrum. It makes spectrum access possible to unlicensed devices in emergencies or in areas where there is no licensed spectrum available. It also facilitates frequency coordination among licensees of co-primary services. It aids interoperability between different kinds of radio systems and thus in formation of mesh networks [68, 69].

2.7 Software Defined Radio (SDR)

A software defined radio or SDR is as the name suggests a radio whose elements are designed or defined by software rather than the prevalent hardware implementation. This feature of the SDR makes it highly versatile by providing a standard platform over which the radio can be conformably used across various fields by changing or making the necessary upgrades to the software as needed, without undergoing the hassle of replacing or reconfiguring the hardware components. It is also highly cost effective which makes the SDR more desirable. It is also the radio of choice for implementing cognitive radios.

A single radio can work effectively at different frequencies and communication protocols [68]. In software defined radios, only the RF front end is implemented by hardware while its working is governed totally by software making it a dynamic system no longer dependent on its circuits. The FCC defines a software defined radio as [70]:

“a radio that includes a transmitter in which the operating parameters of frequency range, modulation type or maximum output power (either radiated or conducted), or the circumstances under which the transmitter operates in accordance with Commission rules, can be altered by making a change in software without making any changes to hardware components that affect the radio frequency emissions.”

Software defined radios are expected to be highly beneficial in the field of mobile communications for subscribers, mobile network operators and handset and base station manufacturers [71].

In 1984 a team at the Garland Texas Division of E-Systems Inc.(now Raytheon) coined the term “software radio”. The term “software defined radio” was coined by Joseph Mitola in 1992. The requirement for a radio transceiver system which can be reconfigured remotely has driven the development of the SDR [72]. Some of the first work on software defined radios was undertaken by the Department of Defence (DoD) through the “SPEAKeasy” and “Joint Tactical Radio Systems”(JTRS) programs [73]. One of the first implementation of the software defined radio transceiver design was achieved by Peter Hoeher and Helmuth Lang at the German Aerospace Research Establishment (DLR, formerly DFVLR) in Oberpfaffenhofen, Germany, in 1988 [74]. Nowadays, radios where most of the functionality of a standard transmitter and receiver can be accomplished via software programs embedded on microprocessors or digital signal processors have been built. The input and output of these radios is in the analog form, which is then converted to its digital form for processing which is done by software. They perform baseband processing - modulation/demodulation, error correction coding, and compression entirely in software [68, 75].

To get a clearer understanding of how all this works lets look at the basic block diagram of a radio and then identify the parts that can be implemented by software.

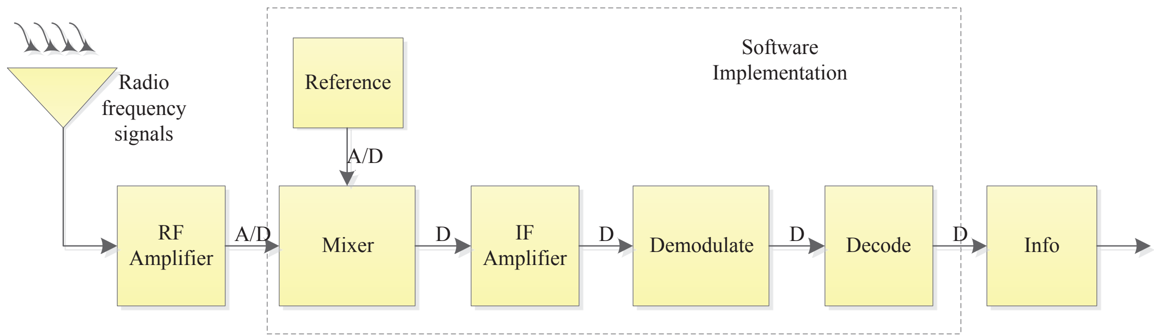


Figure 2.19: Block diagram representation of a standard radio receiver.

As the Figure 2.19 depicts, the antenna and RF amplifier of a basic radio are still hardware implemented in an SDR, whereas the processing parts like the mixing, amplification, demodulation and decoding can be achieved by means of software, thus making it a radio whose functionality is controlled by software.

Software defined radios (SDRs) are usually characterized by a hardware platform used in conjuction with a software architecture. The common components of the hardware platform are radio-frequency (RF) parts, one or more of ASICs (application specific integrated circuits), FPGAs (field programmable gate arrays), DSPs (digital signal processors) or GPPs (general purpose processors) and links to communicate with the host system which contains the software based signal processing elements [76]. Some of the prevailing SDR hardware platforms are Universal Software Radio Peripheral 2 (USRP2), Rice Wireless Open Access Research Platform (WARP), Berkeley Emulation Engine (BEE3), Kansas University Agile Radio (KUAR), Small Form Factor Software Defined Radio (SSF-SDR) and Intelligent Transport System (ITS).

Table 2.2: Commonly used SDR Hardware Platforms.

Model	Architecture	RF Bandwidth (MHz)	RF Range (GHz)	RF Channels	Connectivity	ADCs	DACs	Power (volts)	Price ($)
USRP2	FPGA (Xilinx Spartan III) GPP (AeMB Processor)	100	2.4&5 (multi M/G-Hz)	2	Gig Ethernet (National Semiconductor)	14 bit, 100MS/s (LTC2284)	16 bit, 400MS/s (AD9777)	6	1,400
KUAR	FPGA (Xilinx Virtex II) GPP (Pentium M)	30	2.4&5 ISM/UNII	16	USB 2.0, Gig Ethernet	14 bit, 105MS/s (LTC2284)	16 bit, 100MS/s (AD9777)	12	4,000
WRAP v2.2	FPGA (Xilinx Virtex-4) GPP (PowerPC)	30	2.4&5 ISM/UNII	4	USB 2.0/ Gig Ethernet (Marvel)	14 bit, 65MS/s (AD9284)	16 bit, 160MS/s (AD9777)	12	6,500
SSF-SDR	FPGA (Xilinx Virtex-4) GPP (ARM926) DSP (TMX320xx)	22	0.2-1.0, 1.6-2.2, 2.5/3.5 (WiMAX)	2	RS232/ USB 2.0, Ethernet	14 bit, 125MS/s (TI ADS5500)	16 bit, 500MS/s (TI DAC5687)	12	9,900
BEE3	4 FPGAs (Xilinx Virtex-5) Quad-Core OpenSPARC	100	Ultra wide band (multi GHz)	16	RS232/USB, Gig Ethernet (Broadcom)	8 bit, 3GS/s (BEE3-ADC-D3G)	12 bit, 2GS/s (BEE3-DAC-D2G)	12	20,000
NICT-ITS	FPGA (Xilinx Virtex-4) GPP (ARM11)	5	2,5 and VUHF	5	RS232C, USB, Ethernet	12 bit, 170MS/s (ADCs N/K)	12 bit, 500MS/s (DACs N/K)	3.3	N/A

GNU Radio, Open-Source SCA Implementation-Embedded (OSSIE), Wireless Open-Access Research Platform for Network (WARPnet), Cognitive Radio Open Source System (CROSS) and MATLAB/Simulink are the popular choices for software architecture implementation [77].

2.8 Universal Software Radio Peripheral

The Universal Software Radio Peripheral is a family of computer-hosted hardware for making software radios [78]. The USRPs are products of Ettus Research LLC, developed by Matt Ettus and his company. It is an inexpensive hardware device facilitating the building of a software radio. The hardware connects to a host computer through a high-speed USB or Gigabit Ethernet. Their range is from DC to 6GHz. The USRPs are the preferred SDR platforms as they are highly cost effective, while being operational over a vast frequency scope due to the ample choice of available radio frequency (RF) front end modules [75].

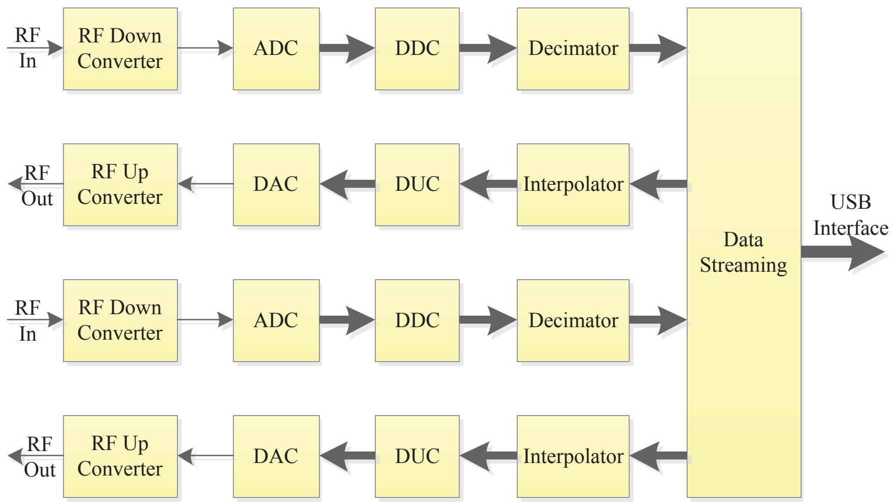


Figure 2.20: Schematic diagram of a general USRP board depicting the input/output paths.

A basic USRP RF transceiver system comprises of the mother board and a few daughter boards. The mother board is the main engine which oversees the analog-digital conversion, baseband and digital signal processing. It constitutes of the analog to digital converters (ADCs), digital to analog converters (DACs) and the FPGA. Two ADCs and two DACs result in four possible transmit and receive paths. The FPGA performs the signal processing and conversion, like up-down conversion, decimation and interpolation. It also performs the conversion from baseband signals to streamed data so it is compatible with the host system, to which the data is sent through the interface. The choice of the RF daughterboards control the frequency range over which the SDR is operational. These serve as the RF front end devices which are responsible for signal transmission, reception and frequency conversion between RF and intermediate frequency (IF) signals. The USRP platform can be programmed and controlled through a software package, such as MATLAB/Simulink, GNU radio and LabVIEW [79].

The USRP2 is the second generation of the USRPs. It is interfaced with a host system

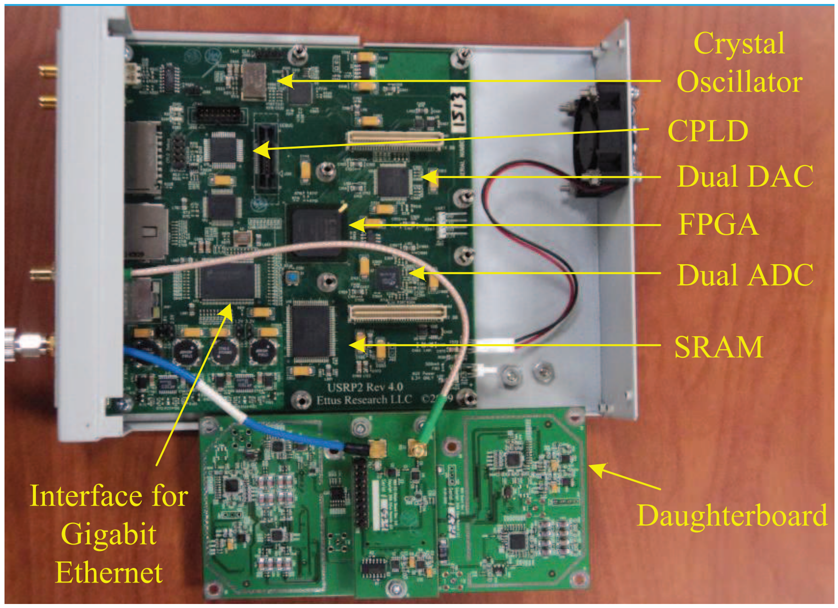


Figure 2.21: USRP2 internal view.

through Gigabit Ethernet. This connection enables host-based software to control the USRP and prepare signals for transmission or reception. USRP2 is the later version of USRP, its internal view is shown in Figure 2.21. It consists of a Xilinx Spartan 3-2000 FPGA, a SD card reader , two 100 MS/s, 14 bit, analog to digital converters and two 400 MS/s, 16 bit, digital to analog converters. It has two daughtercards, XCVR2450, Dual-band Transceiver, 100+mW output at 2.4-2.5 GHz and 50+mW output 4.9-5.85 GHz which serve as the RF front end [80, 77, 66].

2.9 MATLAB and Simulink

Since the main aim of this thesis is to show the practicality of FBMC standard with cognitive radios, it is imperative for us to create a software defined radio to this purpose. The SDR is fashioned for testing purposes by using USRPs with a personal computer as the hardware platform in conjuction with MATLAB and Simulink on the software end.

Simulink along with MATLAB is the development environment for creating the working model of the FBMC system. They are products developed by MathWorks.

2.9.1 MATLAB

MATLAB, MATrix LABoratory is a language for technical computing. It can be used for algorithm development, data analysis, visualization and numerical computation. It is a tool for effortless handling of mathematical calculations. An interactive system, its appeal lies in that fact that it allows you to plot and thus visualize data in numerous ways. Working with matrix algebra, polynomials and integration functions is very easy with MATLAB. It is a computational and programming toolbox with several tools that facilitate the exploration and resolution of problems ultimately improving the learning process. [81]. Complex numerical problems are solved easily and in a fraction of the time required with a programming language such as FORTRAN or C, which make it so desirable. MATLAB is discrete in nature, and thus all inputs and outputs of MATLAB codes are discrete.

MATLAB has a good collection of built in functions which make the job all the more easier for the user, as the need for defining every little function is avoided. The functions range over a wide criterion, some of which are basic math, linear algebra, statistics, random numbers, numerical integration, differential equations, fourier analysis, interpolation and so on. It also has the alternative of user defined functions, which can be used as any other built-in function once defined. All these qualities make it a highly viable choice for working in the areas of signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology.

It has numerous options and defined functions for building digital filters, which can than be parameterized according to user choice. This and other built in signal processing functions, along with its comprehensive data analytical tools make it a good choice for implementing this thesis. It is built to work hand in hand with Simulink and integration between the two is highly beneficial in furthering the simulation and analysis process [82, 83].

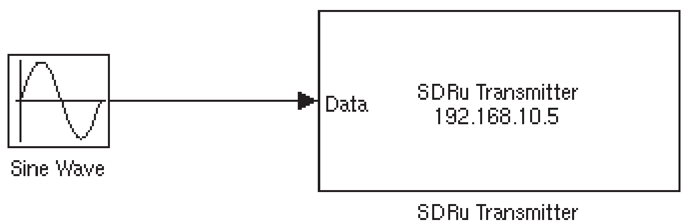
2.9.2 Simulink

Simulink is a graphical user interface program that uses block diagrams for simulating various systems. It is highly interactive and can be used to study the behavior of systems under different conditions and parameters [84]. The merits of Simulink lie in the fact that it can model and simulate dynamic systems, while also providing an option to generate code for the diverse digital signal processing hardware being replicated. It is possible to realize and model any complex system. In Simulink, the signal processing blocks are mapped to functional blocks which are then interlinked and parameterized to form complete systems.

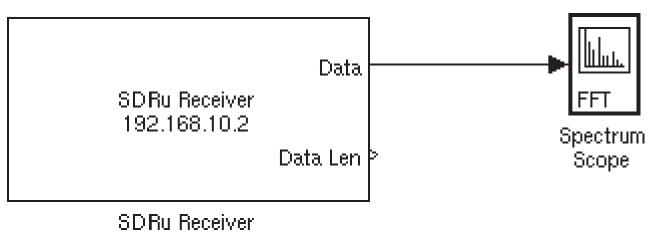
Simulink has an extensive library comprising of predefined blocksets encompassing areas such as digital signal processing systems, communication systems, computer vision systems, control systems, neural networks and so on apart from the basic blocks such as math, logic and bit operators, structural blocks, sinks and sources. Simulink supports both continuous and discrete blocks. It allows the user to build hierarchical models, where subsystem contents can be hidden under a mask if needed. It has the option where MATLAB code can be integrated into its models to simulate blocks that do not exist in the library. These code blocks can be customized according to requirement. It also provides an option to extract simulation results to MATLAB for further examination.

Simulink has both signals which are time-varying data and parameters the coefficients defining system dynamics and behaviour. Their attributes like data type, dimensions, range, initial values, units and type(real or complex) can be altered by the user. For simulation of models, Simulink is equipped with solvers, fixed-step and variable-step ODE (Ordinary Differential Equation) solvers, that work with a wide range of systems. The type and properties of the solver, start and stop times can be designated by the user. Thus most of the modeling and design can be customized and controlled by the user making it a very useful tool for implementation [85, 76].

Simulink has special plug-ins which allow interaction between the USRP2 and the host computer. It has built-in SDRu Transmitter and SDRu Receiver blocks which convert baseband signals to passband signals of transmission frequency and passband signals to baseband signals respectively. Once the transmitter block converts the signals to passband frequencies they are given to the USRP2s through the Ethernet cable for transmission. Figure 2.22(a) shows a simple transmitter model with a sine wave generator and the SDRu transmitter block. As shown, the SDRu transmitter block has one input port and no output ports. Figure 2.22(b) depicts the receiver model with an SDRu receiver block and a spectrum scope block. The SDRu receiver block has no inputs and two outputs, the output data and the data length. The data length output is optional and can be left unconnected.



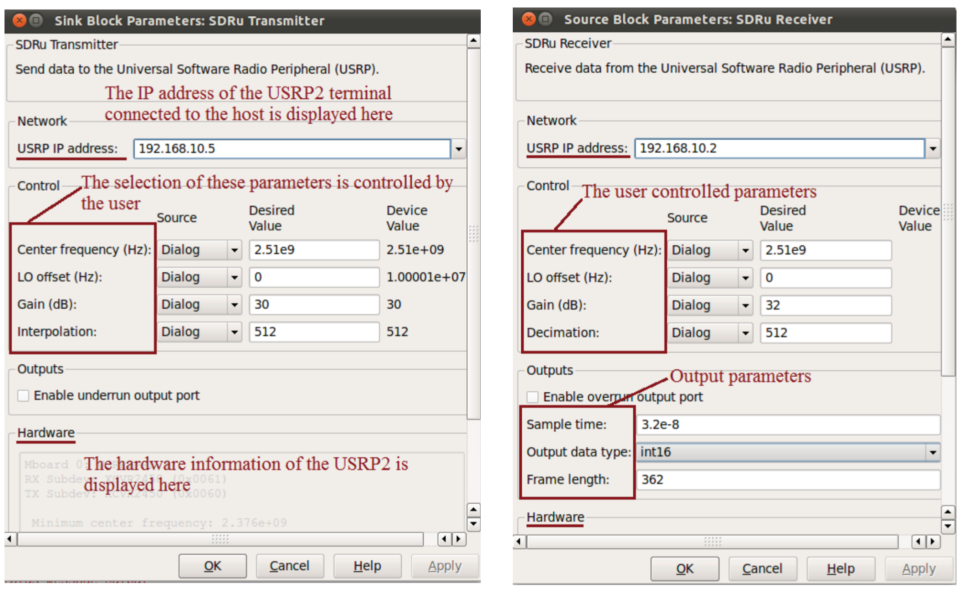
(a) Simulink model of a basic sine wave transmitter.



(b) Simulink model of a basic receiver.

Figure 2.22: Simulink SDRu blocks that enable the host to interface with the USRP2.

Figure 2.23 shows the parameters of the SDRu blocks. They have an USRP IP address section, where the IP address of the USRP connected to the host is entered. This address is usually detected by the host computer when connected to the USRP and can be selected from the drop list. The control section consists of the parameters that can be regulated by the user. These control parameters are almost the same for both the blocks, with the exception that the transmitter block has the interpolation and receiver block has the decimation option. The SDRu receiver has an additional outputs section where the sample time, output data type and the frame length can be specified by the user. The last section in both blocks is the hardware section, which contains the hardware properties of the USRP connected to the host computer. These properties are fixed and cannot be changed.



(a) SDRu transmitter block parameters. (b) SDRu receiver block parameters.

Figure 2.23: Parameters of the SDRu blocks.

2.10 Chapter Summary

This chapter gives an overview of the basics of multirate systems. It discusses the classification of signals, systems and then delves in detail into filters. The different types of filters and multirate systems, especially filter banks are covered. Paraunitary, linear phase, exponentially modulated, cosine modulated and modified DFT filter banks are explained. This chapter then introduces cognitive radios, software defined radios and reviews the implementation platforms USRPs, MATLAB and Simulink. The next chapter talks about the proposed implementations to perform FBMC transmissions.

Chapter 3

Proposed Implementations of Filter bank Multicarrier Modulation

The ultimate goal of this thesis is to prove that filter bank multicarrier modulation is a viable option for wireless cognitive transmission. For this, it should be capable of spectrally agile waveform production and this is possible through the non-contiguous FBMC implementation. To realize non-contiguous FBMC transmission, it is first necessary to build an FBMC transmitter compatible with cognitive radios, therefore the first step in this thesis is the implementation of a FBMC transmitter with an SDR. It is important to note that this thesis involves accomplishing a filter bank transmultiplexer and focuses solely on the transmitter side of the operation.

This chapter gives a comprehensive insight into the proposed approaches to achieve filter bank multicarrier (FBMC) modulation for cognitive radio use. Out of the many available options, it introduces three alternate techniques to realize this goal, each with their own advantages. The first approach employs cosine modulated filters to achieve FBMC modulation. The second approach uses exponentially modulated filters to implement FBMC modulation. These two techniques are very closely associated with each other. The prototype filters used in both the techniques as well as the center frequencies are the same. The selection of the prototype filter as well as the particular set of center frequencies is explained. The final technique detailed here can be considered as a modified version of the former methods and can be applied to either. It is the polyphase implementation of the filter banks and is computationally less complex than the others.

3.1 Cosine Modulated Filter bank Implementation

This section presents the first proposed method to implement filter bank multicarrier (FBMC) modulation, by using cosine modulated uniform filters for the synthesis bank in the transmultiplexer transmitter. In cosine modulated filter bank modulation, a common lowpass prototype is translated to the required center frequencies by modulating it with a cosine function. Before going into details about the proposed design it is important to understand what a simple modulated filter bank transmultiplexer system entails. The practical realization of filter banks involves numerous steps which are going to be studied here. Let us look at a simple block diagram of a transmitter of the modulated filter bank transmultiplexer.

We observe that it contains the adaptive bit demultiplexer, subcarrier modulator, n-fold expanders, filters and adder blocks. The preference of modulation for the subcarriers is usually PAM (Pulse Amplitude Modulation) or QAM (Quadrature Amplitude Modulation). Pulse amplitude modulation is a modulation technique in which the carrier is modulated according to the data signal. It is usually used together with vestigial sideband modulation (VSB), where only a part of the signal spectrum is transmitted which effectively represents the entire information. Quadrature amplitude modulation is a linear modulation scheme where the data signals are modulated by two carrier signals, that are $90°$ out-of-phase with each other, it is a double sideband modulation (DSB) technique.

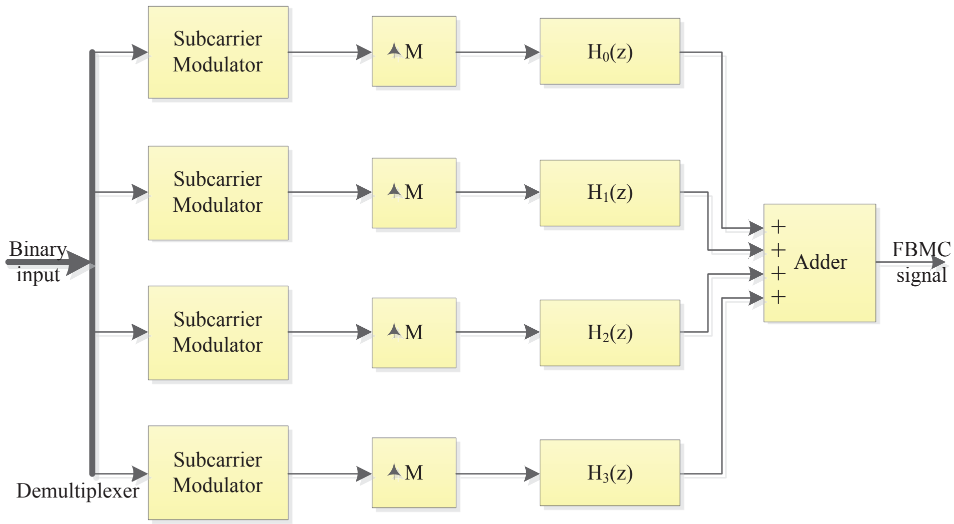


Figure 3.1: Graphical illustration of the transmitter of a simple transmultiplexer.

QAM modulation technique is dealt in more detail in Appendix A. After the modulation stage is interpolation, the upsampler in series with the filter forms the interpolator. We employ maximally decimated filter banks. Therefore the interpolation factor M is always equal to the number of channels N. The filters form the synthesis bank and are uniform modulated filters based on the same lowpass prototype. The design of the synthesis bank of filters will be discussed in this section. We are going to divide the designing process into two stages for better comprehension.

First comes the design of the lowpass prototype. The prototype design is governed by some prerequisites, which will be specified here. For the prototype a lowpass filter is preferred, as it is the base from which all the filters are constructed and modulating to different center frequencies becomes straightforward. For FBMC systems, FIR filters are preferred over IIR a few reasons being, FIR filters have linear phase whereas IIR do not. Moreover IIR filters may cause a synchronization problem later on even though they are of a lower order. Notwithstanding the existing preferences, different types of FIR and IIR filters, have been tested to see which gives the maximum performance results in a cognitive scenario. The results have been considered, and a decision has been made taking the overall advantage of the parameter set in account.

Filter banks usually depend on a unique lowpass prototype, but there exist systems that depend on two prototype filters. Since we will deal with the former, our discussion is also limited to them. The prototype filter is generally restricted to be of the order $O = 2kN - 1$ where k is an integer called the overlapping factor and N is the number of channels. The lowpass prototype is subjected to a bandwidth constraint, which becomes necessary to ensure that the spectral overlap occurs only between adjacent channels. It also guarantees the utilization of the whole $[0, \pi]$ range. If we think of the general scenario, for N filters with distinct center frequencies to function in the complete frequency range $[0, \pi]$ without interference (no overlap) from their neighbours, is to have a bandwidth of π/N. Here the center frequencies are usually odd multiples of $\pi/2N$, since overlap is allowed between adjacent filters the minimum bandwidth that can be allocated becomes $\pi/2N$. Therefore for a lowpass prototype, the stopband frequency ω_s should lie in the range $[\pi/2N, \pi/N]$ [57].

Filter bank systems exist where the center frequencies are repeated, in such systems all the frequencies except for 0 or π are used for transmission by two different signals. In such a system the bandwidth constraint varies slightly from the earlier case. Here to have filters which do not overlap the bandwidth allowed would be $2\pi/N$. But since overlap is allowed the minimum bandwidth for the prototype is π/N.

Selection of center frequencies is another major consideration in filter bank design. There are a bunch of possible frequency sets, in such frequency sets, the frequencies are uniformly spaced in the range $[0, \pi]$. The possible sets are comprised of either distinct frequencies or repeated frequencies. For the set with the distinct frequencies, the center frequencies are usually odd multiples of $\pi/2N$ given as:

$$\frac{\pi}{2N}, \frac{3\pi}{2N}, \frac{5\pi}{2N}, \frac{7\pi}{2N}, \cdots, \pi - \frac{\pi}{2N}. \tag{3.1}$$

Another possibility for distinct center frequencies is,

$$0, \frac{\pi}{N}, \frac{2\pi}{N}, \frac{3\pi}{N}, \frac{4\pi}{N}, \cdots, \frac{(N-1)\pi}{N}. \tag{3.2}$$

We notice that the frequencies are π/N apart from each other, thus the entire range is utilized.

For the case of repeated frequencies two set of frequencies are likely. The first set consists of 0, π and multiples of $2\pi/N$ and is given as:

$$0, \frac{2\pi}{N}, \frac{2\pi}{N}, \frac{4\pi}{N}, \frac{4\pi}{N}, \cdots, \pi - \frac{2\pi}{N}, \pi - \frac{2\pi}{N}, \pi. \tag{3.3}$$

The other set is collection of the odd multiples of π/N given by:

$$\frac{\pi}{N}, \frac{\pi}{N}, \frac{3\pi}{N}, \frac{3\pi}{N}, \cdots, \pi - \frac{\pi}{N}, \pi - \frac{\pi}{N}. \tag{3.4}$$

The first and the last two frequency sets were proposed by Ramchandran in his dissertation [57].

Second is the type of modulation that is used for the synthesis bank. Though there are multiple options, here we look at both cosine modulation and exponential modulation as possible options for FBMC. In the following section we discuss the design specifications along with the model schematic.

3.1.1 Schematic of the FBMC Transmitter based on Cosine Modulated Filter Bank Modulation

The proposed structure for implementing cosine modulation for FBMC modulation is illustrated in Figure 3.2. The primary blocks are the same as mentioned in the previous discussion of a transmultiplexer system, in particular the transmitter. This system accepts binary input data which is then processed through the various blocks.

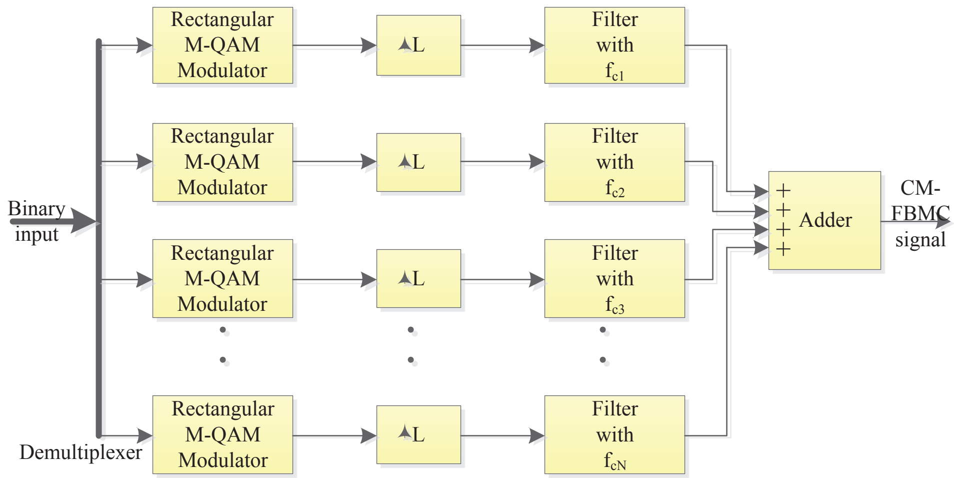


Figure 3.2: Schematic of the proposed implementation of cosine modulated filter bank multicarrier modulated transmitter.

The binary input is produced by a random binary generator, which is then diverted into the N branches using a demultiplexer. The demultiplexer allocates n bits to each branch to serve as input to the data modulators. From this point on there are N parallel branches operating simultaneously. The number of bits n allocated to each branch depend on the type of modulation being used. Since our model is a uniform filter bank transmultiplexer transmitter, all the modulation schemes are the same and hence all the branches obtain n bits. The modulation scheme employed in our design is $M - QAM$ modulation, where $M = 2^n$. The modulated data in each branch is then upsampled, where the upsampling factor L is equal to the number of branches N, since the designed model is a critically sampled version. The upsampled data is then filtered by the filters in each branch. The filtered output from each branch is then combined to give the FBMC signal that is to be transmitted.

Proposed Implementation

The proposed implementation uses a cosine function to modulate the prototype to give the filters in the filter bank. A lowpass filter built using the least-squares error minimization technique is used as the prototype. It is a linear phase filter. $16-QAM$ modulation is the employed modulation scheme, hence each branch has a 4 bit input to the modulator from the demultiplexer.

First a 4-filter bank system is executed, and as such the interpolation factor is taken to be 4, since we are implementing maximally decimated filter banks. One of the purposes of this implementation is to find the best set of parameters for cognitive use, therefore two sets of frequencies are tested, along with a choice of filter bandwidth. The model is later enhanced to a 16 channel structure to ascertain its ability for practical transmission. We test the system with two sets of frequencies, the first set is given by the frequencies $\frac{\pi}{8}, \frac{3\pi}{8}, \frac{5\pi}{8}, \frac{7\pi}{8}$. The second set that is tried consists of the frequencies $0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$.

Depending on the choice of the center frequencies, the manner in which the upsampled "copy" of the signal is filtered also varies. The second set of frequencies filter a complete "copy" of the upsampled signal, while the first set filter vestiges of the signal, which when added together give information about the actual signal. However, the problem with the second set is that the amplitude of the cosine modulating function for the frequencies $\frac{\pi}{4}, \frac{3\pi}{4}$ should be twice that of the others. Both the sets give similar results for FBMC generation, therefore we select the first set for ease of operation. Since it is a 4 bank system the stopband of the prototype should lie in the range $[0.125, 0.25]$. The actual implemented model looks like given in the Figure 3.3.

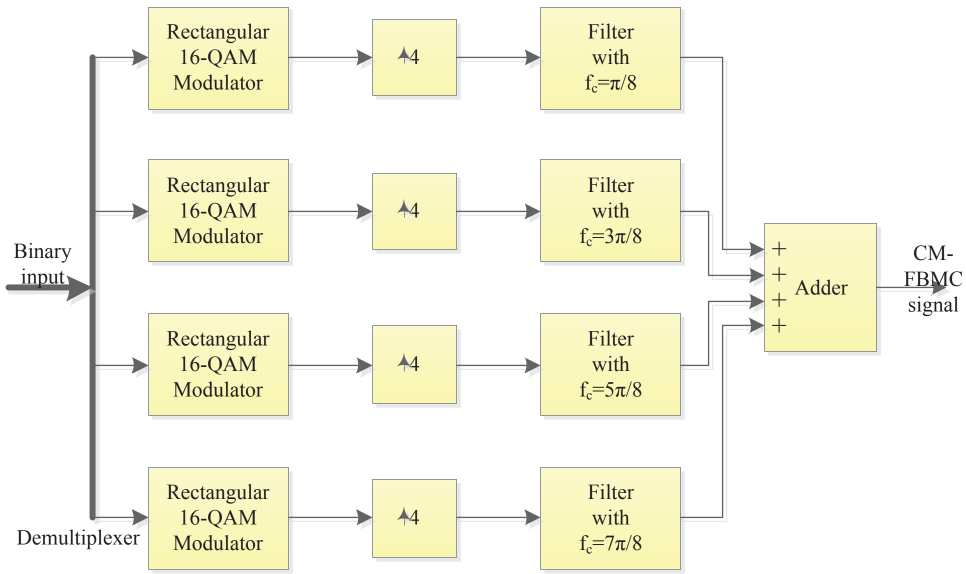


Figure 3.3: The implemented 4-bank system of cosine modulated filter bank multicarrier modulated transmitter.

SDR Implementation: The SDR implementation for this technique is quite straightforward, it is achieved by building Simulink models that are run using USRPs. The blocks are implemented as shown in the Figure 3.4. The binary data is produced by a binary generator block, which is then stored in a buffer whose length depends on N and is given as $l = 4N$, here $N = 4$ and $l = 16$ since it is a 4-bank system. This block of data is then given to the demultiplexer which then diverts it into N channels, each 4 bits in size as input to the 16-QAM modulator blocks. The modulated signals are upsampled by a factor of 4 and given to the MATLAB function blocks. The filters are implemented by means of MATLAB programs that are imported into the Simulink models through these function blocks. The outputs from these blocks are given to the matrix sum block which adds the signals together to give the FBMC signal. This composite signal is then given to the SDRu transmitter block to be converted to a passband signal and sent to the USRP2 for transmission.

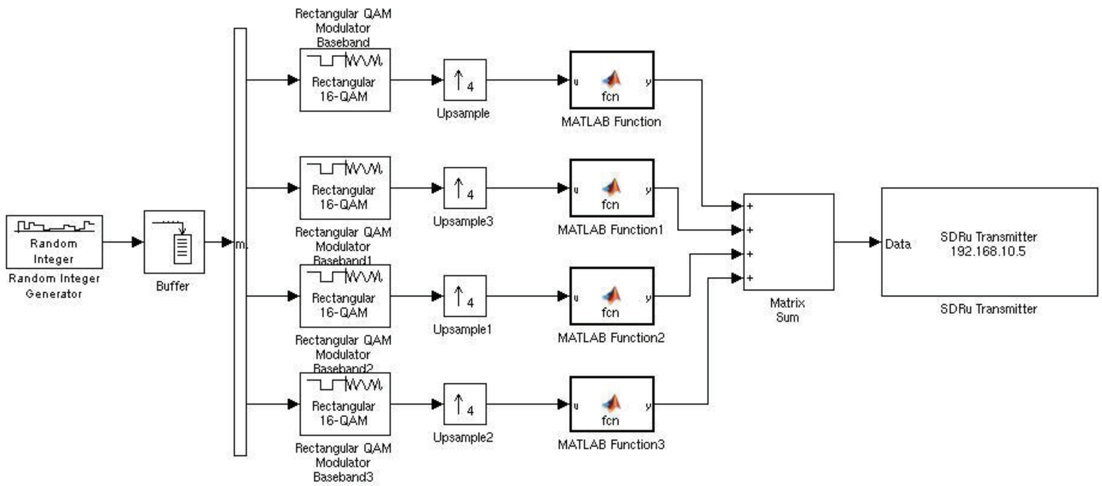


Figure 3.4: SDR implementation of the 4-bank cosine modulated filter bank multicarrier modulated transmitter.

3.2 Exponentially Modulated Filter bank Implementation

This section details the next approach that has been considered for achieving FBMC modulation which is to use exponential modulation for modulating the filters in the synthesis filter banks. Exponentially modulated filter banks and cosine modulated filter banks are very closely related to each other and are the most popular techniques to implement FBMC systems. One of the main difference between the systems comes in their filtering characteristics, as exponentially modulated filters have single sided frequency responses while cosine modulated filters have double sided frequency responses.

3.2.1 Proposed Implementation

The structure for the exponentially modulated filter bank transmitter is fundamentally similar to the cosine modulated transmitter except for the difference in the design of the filter banks. The lowpass prototype is modulated by an exponential function that centers the filters at a unique frequency.

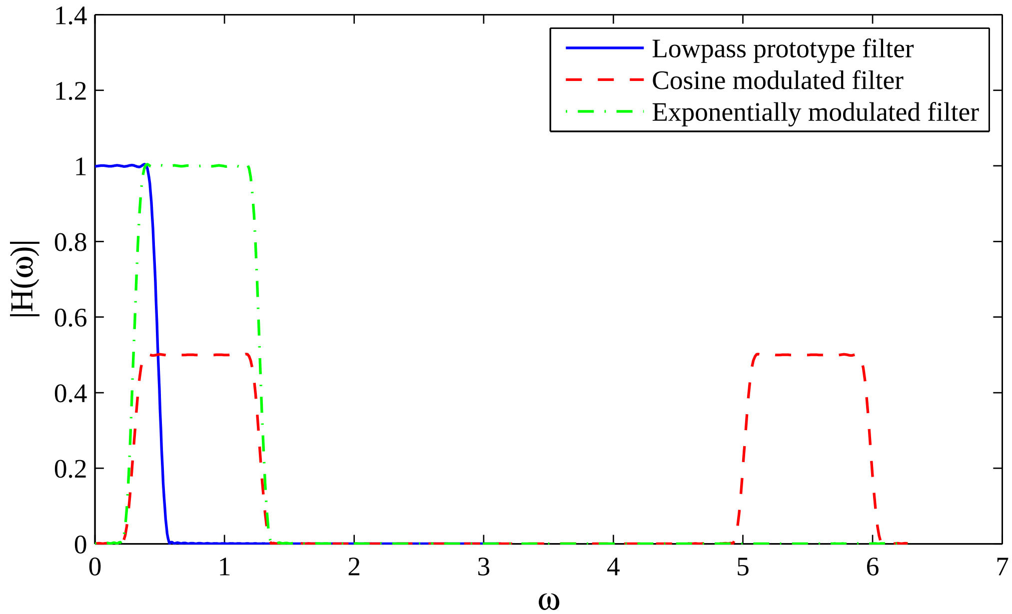


Figure 3.5: Comparison of the frequency response characteristics of cosine modulated and exponentially modulated prototype filter.

The principle difference in the effect of modulating a filter with an exponential function from that of a cosine is in its frequency response characteristics, as shown in Figure 3.5. The modulation with an exponential functions restricts the frequency response to a one-sided response, the significance of which will become apparent in the next section. Here, it is adequate to know the performance of this design with a choice of center frequencies and bandwidths. This structure was first executed as a 4 channel system and later upgraded to a 16 channel version. The choice of center frequencies and prototype bandwidth is the same as the earlier case.

The SDR implementation of this case is also straightforward with all the blocks implemented as shown in the earlier section, as in Figure 3.6, excluding the filter MATLAB functions which now have an exponential function modulating the prototype filter. The performance of both the systems, which are sometimes commonly referred to as complex modulated filter bank techniques, is mostly similar. They are each advantageous in certain scenarios depending on the required specifications. Exponentially modulated filter banks in particular are highly desirable for implementing non-contiguous FBMC systems, which will be discussed in more detail in the next chapter.

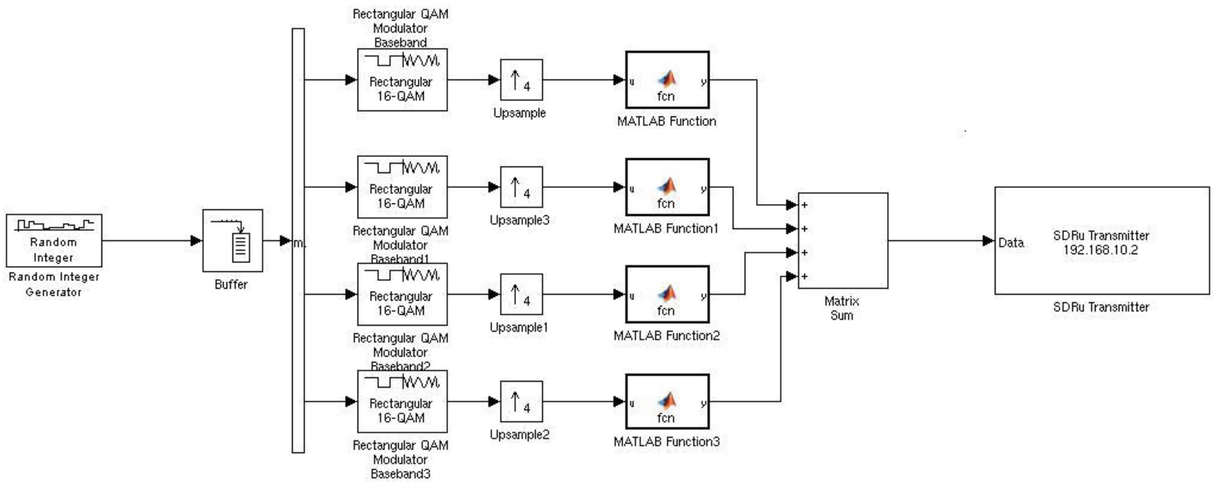


Figure 3.6: SDR implementation of the 4-bank exponentially modulated filter bank multicarrier modulated transmitter.

3.3 Proposed Polyphase Implementation

This section provides a complete description of the third and final suggested technique to perform filter bank multicarrier (FBMC) modulation which is by employing polyphase implementation. The polyphase method can be implemented with various models and before going further into the design architecture it is important to acquire a more thorough understanding of polyphase implementation.

3.3.1 Introduction to Polyphase Decomposition

While optimizing digital processing, it is necessary to minimize the computation rate and thus the sampling frequency of the signal. Consider filtering, one of the trademark processes of digital processing. Filtering usually results in loss of some data, for instance consider a low pass filter, it doesn't allow some of the input signal components which are lost. Similarly if the input and output filter sampling frequencies are the same there is repetition in the processing. Now consider a phase shifter where there is absolutely no loss of data, and minimal sampling frequency at the input is maintained at the output. Through this, efficient sample rate alteration can be obtained, and using these an optimal digital process can be created.

In general, filtering is nothing but a process in which unnecessary signal components are removed. This can be alternatively accomplished by cancelling the undesired signal components at the output of a summation device connected to a set of phase shifters. The phase shifters are all pass devices. They are capable of manipulating the component phases, so all the components that are required are rotated to the same phase so that they add constructively at the summer, while the unnecessary components cancel out due to their different phases. There are various types of phase shifter sets which can achieve filtering, one such group is the polyphase networks.

Let us see how a set of all pass phase shifters followed by a summation device are capable of performing the filtering operation. Consider a polyphase network of order N, defined by a set of N phase shifters $\phi n(n = 0, 1, \cdots, N-1)$, whose phase is defined as:

$$\phi n(f) = -\frac{2\pi}{N} n[\frac{f}{f_r} + \frac{1}{2}] \qquad \text{where } f_r \text{ is the reference frequency and [x] is the greatest integer in x.} \tag{3.5}$$

Apply a signal as input concurrently to all ϕn. From the N output signal components, the ones in the frequency band $(0, f_r/2)$ are in phase and add constructively at the summation point. The frequency band $(f_r/2, 3f_r/2)$ has components from $\phi 1$ which are $-2\pi/n$ out of phase of components coming from $\phi 0$ and similarly components of ϕn rotated to out of phase of $-n2\pi/n$ where $n = 2, \cdots, N-1$. Thus all these components eventually cancel each other out which also repeats in the bands $(3f_r/2, 5f_r/2)(5f_r/2, 7f_r/2)$ etc. In the end only the components in the bands $[i(N - \frac{1}{2})fr, i(N + \frac{1}{2})fr]$ where i is an integer remain.

The overall system has a frequency response equivalent to that of a lowpass filter with cutoff frequency $f_r/2$and periodicity Nf_r on the frequency axis [86].

Though this looks like a complicated way of performing filtering it succeeds in optimizing the processing operation which will be explained in a later section.

3.3.2 Polyphase Decomposition

Polyphase decomposition is one of the main factors that aided the increasing popularity of multirate processing. With the implementation of polyphase decomposition the overall computational complexity of any multirate system decreases drastically as the computations are performed at the least possible rate. To get a better understanding of polyphase decomposition and what it entails let us first learn about polyphase representation.

3.3.3 Polyphase Representation

Consider a sequence $x[n]$, whose Z-transform $X(z)$ is:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \tag{3.6}$$

This can be written as:

$$X_k(z) = \sum_{k=0}^{M-1} z^{-k} X_k(z^M), \tag{3.7}$$

where:

$$X_k(z) = \sum_{n=-\infty}^{\infty} x_k[n]z^{-n} = \sum_{n=-\infty}^{\infty} x[Mn+k]z^{-n}, \quad 0 \leq k \leq M-1. \tag{3.8}$$

The subsequences $x_k[n]$ are the polyphase components of the given sequence $x[n]$. They are related as:

$$x_k[n] = x[Mn+k], \quad 0 \leq k \leq M-1. \tag{3.9}$$

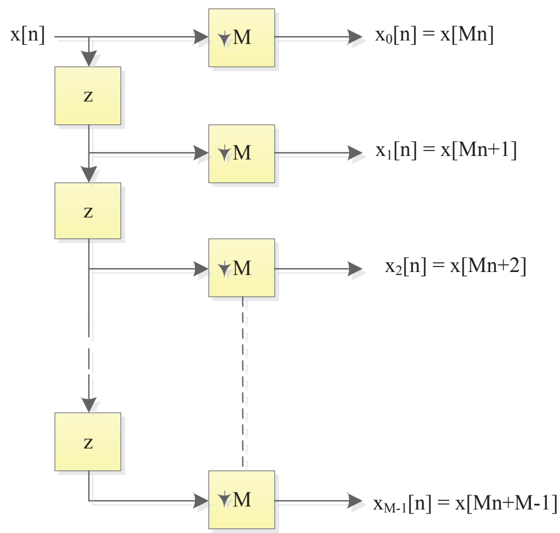


Figure 3.7: Block diagram representation of the relation between the main sequence and the subsequences.

which is diagrammatically represented in Figure 3.7. The Z-transforms of $x_k[n]$, $X_k(z)$ are the polyphase components of $X(z)$. Their relation can be represented as:

$$X(z) = \begin{bmatrix} 1 & z^{-1} & \dots & z^{-(M-1)} \end{bmatrix} \begin{bmatrix} X_0(z^M) \\ X_1(z^M) \\ \vdots \\ X_{M-1}(z^M) \end{bmatrix} \tag{3.10}$$

Now consider a system $h[n]$, whose Z-transform is $H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n}$. This system's polyphase representation is obtained as follows. First we separate the even indexed coefficients of $h[n]$ from the odd indexed coefficients, so the Z-transform becomes:

$$H(z) = \sum_{n=-\infty}^{\infty} h[2n]z^{-2n} + z^{-1} \sum_{n=-\infty}^{\infty} h[2n+1]z^{-2n}, \tag{3.11}$$

we now define:

$$E_0(z) = \sum_{n=-\infty}^{\infty} h[2n]z^{-n}, \quad E_1(z) = \sum_{n=-\infty}^{\infty} h[2n+1]z^{-n}, \tag{3.12}$$

so that $H(z)$ becomes:

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2). \tag{3.13}$$

Here we have decomposed the system by a factor by 2, considering a more general factor M the system can be decomposed as:

$$H(z) = \begin{array}{c} \sum_{n=-\infty}^{\infty} h[nM]z^{-nM} \\ +z^{-1}\sum_{n=-\infty}^{\infty} h[nM+1]z^{-nM} \\ \vdots \\ +z^{-(M-1)}\sum_{n=-\infty}^{\infty} h[nM+M-1]z^{-nM} \end{array} \tag{3.14}$$

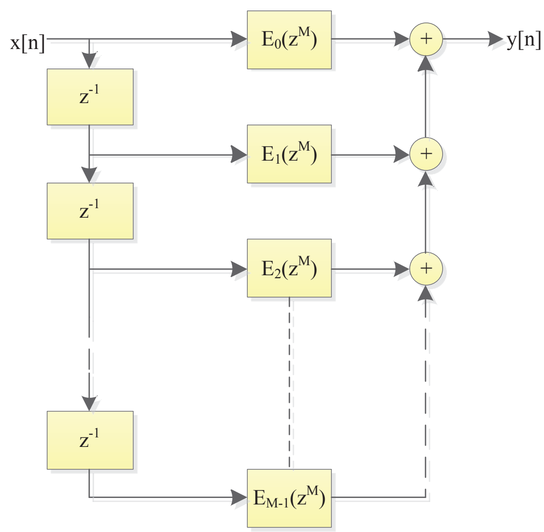


Figure 3.8: Block diagram representation of the Type I polyphase representation.

Succintly written as:

$$H(z) = \sum_{l=0}^{M-1} z^{-l}E_l(z^M), \tag{3.15}$$

here:

$$E_l(z) = \sum_{n=-\infty}^{\infty} e_l(n)z^{-n}. \tag{3.16}$$

The components $e_l[n]$ are related to the system $h[n]$ as:

$$e_l[n] \triangleq h[Mn + l], \quad 0 \leq l \leq M - 1. \tag{3.17}$$

$E_l(z)$ are the polyphase components of $H(z)$, they are governed by the value of M chosen. This type of polyphase representation is known as *Type I polyphase representation*, schematically represented as in Figure 3.8.

Now consider:

$$R_l(z) = E_{M-1-l}(z), \tag{3.18}$$

where $R_l(z)$ are the permutations of $E_l(z)$. Using this relation $H(z)$ becomes:

$$H(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} R_l(z^M). \tag{3.19}$$

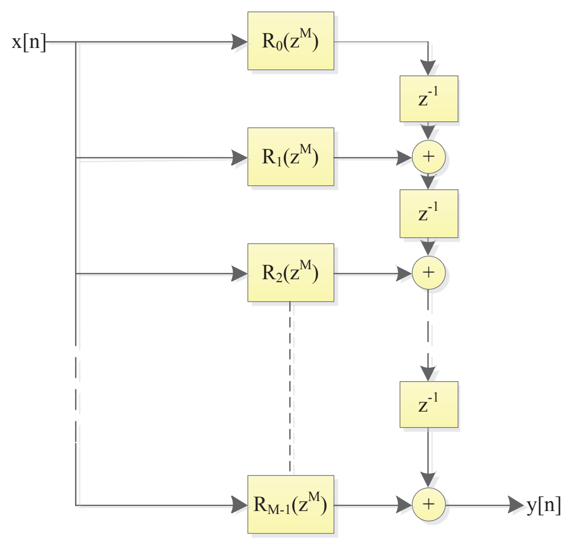


Figure 3.9: Block diagram representation of the Type II polyphase representation.

This representation of the system is known as the *Type II polyphase representation* and is shown in Figure 3.9. It is a modification of the transpose of Type I representation and is also known as reverse polyphase decomposition. Here the powers of z progress in the opposite direction.

While dealing with polyphase implementation we come across various connections between blocks. They are a few simple rules which when followed can simplify the overall structure and make it easier to follow. The rules can be classified as a few general identities and the noble identities. Here we will look at the noble identities and deal with the general identities in Appendix B.

3.3.4 Noble Identities

The noble identities are a set of rules which explain when it is viable to reverse the order of sampling rate alteration and filtering.

(a) Diagrammatic representation of the first noble identity.

(b) Diagrammatic representation of the second noble identity.

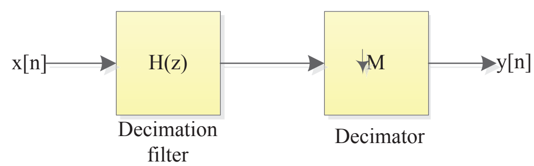
Figure 3.10: The multirate system noble identities.

The connections shown above in Figure 3.10, occur during the application of polyphase representation to the interpolation and decimation filters. In Figures 3.10(a)and(b) the diagrams on the right side can be redrawn as the ones on the left side of the equivalence. These relations are known as *Noble identities* and hold as long as $G(z)$ is rational. The noble identities play a major role in mitigating the complexity of the filter bank systems while using polyphase representation.

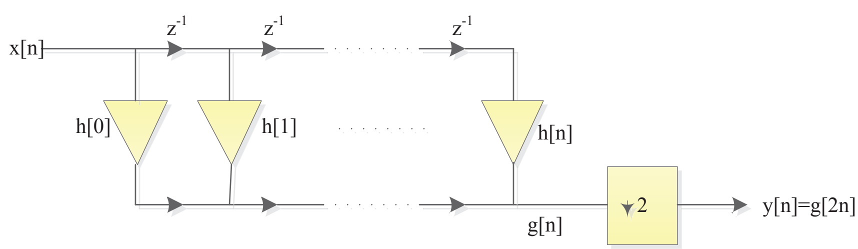
3.3.5 Decimation and Interpolation Filters

Consider a decimation filter $H(z)$ followed by a decimator, with factor $M = 2$ as in Figure 3.11(a). Using the Type I polyphase decomposition it can be represented as:

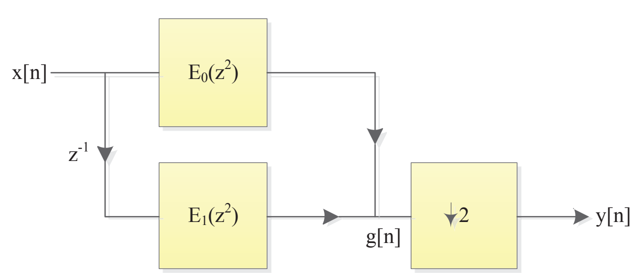
$$H(z) = E_0(z^2) + z^{-1}E_1(z^2). \tag{3.20}$$



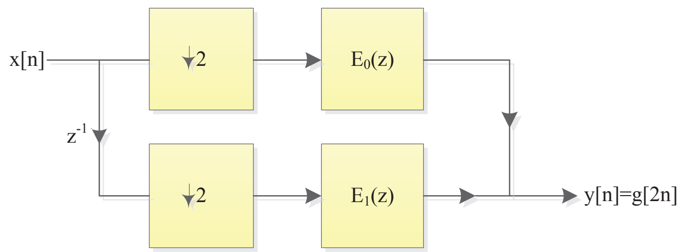
(a) Block diagram of the decimation filter



(b) Direct implementation



(c) Polyphase implementation



(d) Simplified polyphase representation

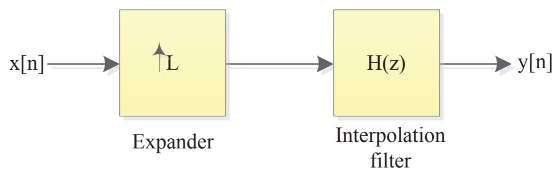
Figure 3.11: Schematic represention of the decimation filter, in its various equivalent models.

The application of polyphase decomposition results in a more efficient representation than the direct version. To better understand how consider an Nth order filter $H(z)$ followed by a decimator, with factor $M = 2$. At the output, only the even indexed samples are retained. This entire operation involves $N + 1$ multiplications and N additions, all of which are required to be completed in a unit of time. Therefore the speed of the operation is $N+1$ multiplications and N additions per unit time. In this operation the resources are not efficiently used as the structure is resting during the odd instants of time. On the other hand if we consider the polyphase implementation with $E_0(z)$ and $E_1(z)$ with orders n_0 and n_1, such that $N + 1 = n_0 + n_1 + 2$ we see that $E_l(z)$ requires $n_l + 1$ multiplications and n_l additions. The total number of operations is again $N + 1$ multiplications and N additions but since $E_l(z)$ operates at a lower rate only $(N + 1)/2$ multiplications and $N/2$ additions per unit time are needed. Therefore the multipliers and adders in the $E_l(z)$ filters have twice the amount of time for the same number of operations and also work continuously increasing the efficiency. This is diagrammatically represented in Figure 3.11.

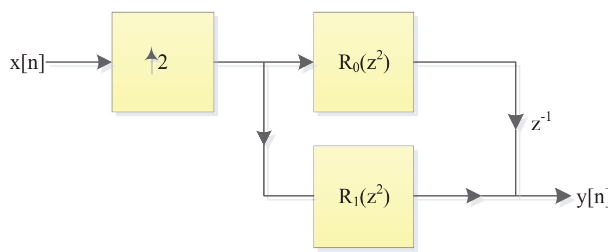
Now let us look at interpolation filters. In a direct implementation with $L = 2$, half of the input samples to $H(z)$ are zero. Thus only half of the multipliers $h(n)$ have nonzero input and the rest are idle, the working operators are also expected to complete the work in half unit of time. Applying Type II polyphase decomposition to the interpolator results, as shown in Figure 3.12 in a more economical design:

$$H(z) = R_1(z^2) + z^{-1} R_0(z^2). \tag{3.21}$$

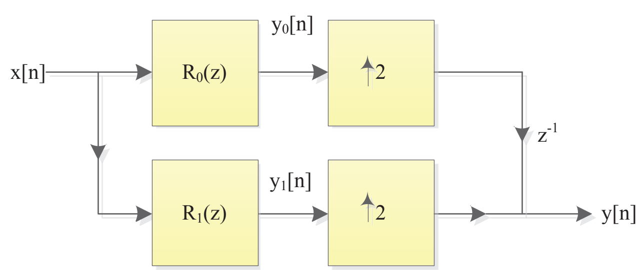
Here $R_l(z)$ work at the input rate, all the multipliers are active and finish their task in a unit of time.



(a) Block diagram of the interpolation filter



(b) Polyphase implementation



(c) Simplified polyphase implementation

Figure 3.12: Schematic represention of the interpolation filter, in its various equivalent models.

3.3.6 Polyphase Representation of Filter banks

We studied in the earlier chapters that the analysis filter bank of a system can be implemented by the Type I polyphase representation. Thus we can define the filters in general as:

$$H_k(z) = \sum_{l=0}^{M-1} z^{-l} E_{kl}(z^M). \tag{3.22}$$

This can then be interpreted in the matrix form as:

$$\begin{bmatrix} H_0(z) \\ \vdots \\ H_{M-1}(z) \end{bmatrix} = \begin{bmatrix} E_{00}(z^M) & E_{01}(z^M) & \cdots & E_{0,M-1}(z^M) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z^M) & E_{M-1,1}(z^M) & \cdots & E_{M-1,M-1}(z^M) \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(M-1)} \end{bmatrix} \tag{3.23}$$

and in an equation form as:

$$\mathbf{h}(z) = \mathbf{E}(z^M)e(z), \tag{3.24}$$

where:

$$\mathbf{E}(z) = \begin{bmatrix} E_{00}(z) & E_{01}(z) & \cdots & E_{0,M-1}(z) \\ E_{10}(z) & E_{11}(z) & \cdots & E_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z) & E_{M-1,1}(z) & \cdots & E_{M-1,M-1}(z) \end{bmatrix} \tag{3.25}$$

$$\mathbf{h}(z) = \begin{bmatrix} H_0(z) \\ H_1(z) \\ \vdots \\ H_{M-1}(z) \end{bmatrix} \tag{3.26}$$

and:

$$\mathbf{e}(z) = \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(M-1)} \end{bmatrix} \tag{3.27}$$

where $\mathbf{h}(z)$ represents the analysis bank and $\mathbf{e}(z)$ represents the delay chain. The matrix $\mathbf{E}(z)$ is known as the $M \times M$ Type I polyphase component matrix. This can be diagrammatically represented as in Figure 3.13.

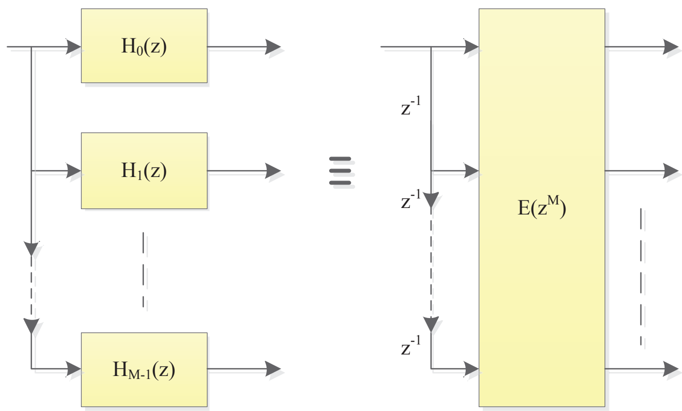


Figure 3.13: Schematic representation of the Type I polyphase representation of a general analysis filter bank.

The synthesis filters on the other hand can be characterized by the Type II polyphase representation as:

$$F_k(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} R_{kl}(z^M) \tag{3.28}$$

In matrix form, it becomes:

$$\begin{bmatrix} F_0(z) & \cdots & F_{M-1}(z) \end{bmatrix} = \begin{bmatrix} z^{-(M-1)} & z^{-(M-2)} & \cdots & 1 \end{bmatrix} \begin{bmatrix} R_{00}(z^M) & R_{01}(z^M) & \cdots & R_{0,M-1}(z^M) \\ R_{10}(z^M) & R_{11}(z^M) & \cdots & R_{1,M-1}(z^M) \\ \vdots & \vdots & \ddots & \vdots \\ R_{M-1,0}(z^M) & R_{M-1,1}(z^M) & \cdots & R_{M-1,M-1}(z^M) \end{bmatrix} \tag{3.29}$$

The equation form is:

$$\mathbf{f}^T(z) = z^{-(M-1)} \tilde{\mathbf{e}}(z) \mathbf{R}(z^M), \tag{3.30}$$

where

$$\mathbf{R}(z) = \begin{bmatrix} R_{00}(z) & R_{01}(z) & \cdots & R_{0,M-1}(z) \\ R_{10}(z) & R_{11}(z) & \cdots & R_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ R_{M-1,0}(z) & R_{M-1,1}(z) & \cdots & R_{M-1,M-1}(z) \end{bmatrix} \tag{3.31}$$

and

$$\mathbf{f}(z) = \begin{bmatrix} F_0(z) \\ F_1(z) \\ \vdots \\ F_{M-1}(z) \end{bmatrix} \tag{3.32}$$

Here $\mathbf{f}(z)$ is the transposed synthesis filter bank and $\mathbf{R}(z)$ is the $M \times M$ Type II polyphase component matrix [51]. This can be schematically represented as in Figure 3.14.

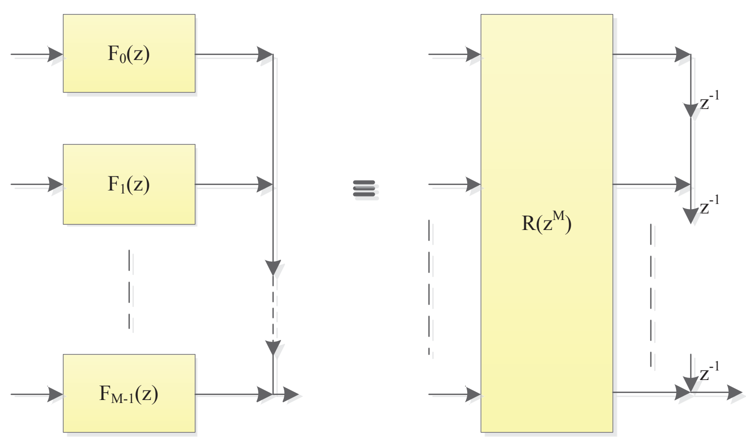


Figure 3.14: Schematic representation of the Type II polyphase representation of a general synthesis filter bank.

Now that we have acquired an idea about polyphase implementation, let us look at the proposed design.

3.3.7 Schematic of the Polyphase Implementation of a Modulated Filter bank

This section details the implementation of the polyphase representation of th filter bank multicarrier system. We are implementing the polyphase equivalent of a 4-bank filter bank system. The following Figure 3.15, gives the schematic of the proposed procedure.

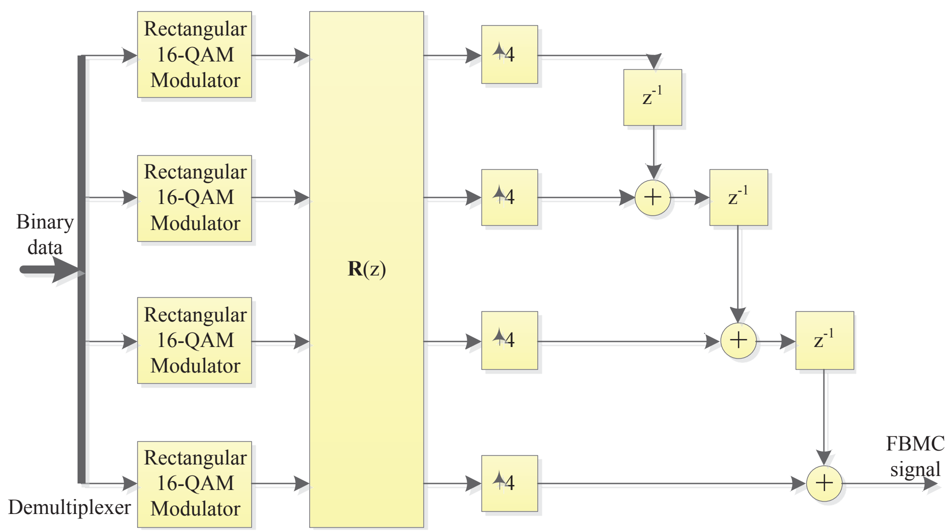


Figure 3.15: Schematic representation of the polyphase implementation of the FBMC transmitter.

The binary data is first processed by the demultiplexer which feeds the data into the 4 branches, 4 bits at a time as input to the 16-QAM modulators. The modulated data is then multiplied by the polyphase component matrix. Since we are implementing the transmitter of a transmultiplexer, we are basically implemeting the synthesis filter bank and thus use the Type II polyphase component matrix $\mathbf{R}(z)$ here. The data manipulated by the matrix $\mathbf{R}(z)$ is again branched into the 4 channels, which is then upsampled by a factor of 4. The upsampled data from the first branch is unit delayed and added to the data from the second branch. The output of this adder is further unit delayed and added to the output of the third branch. The output from this adder is unit delayed and added to the data from the last branch, the output of this adder gives the required FBMC signal.

3.3.8 Proposed Implementation

The main components in the polyphase representation of the filter banks are the polyphase matrix, sample rate converters and the delay elements. Since our main aim is to implement the transmultiplexer transmitter, we will deal only with synthesis filter banks and as such the Type II polyphase component matrix.

The demultiplexed and modulated binary data is multiplied by the polyphase matrix, after which it is processed by a series of upsamplers, delay elements and adders to give the final FBMC signal. This process as explained in the earlier sections basically has the same effect as having modulated filter banks in each branch. The merit of this technique is its ease of implementation in relation to the conventional method. Since the aim of the thesis is to test the practicality of the implementations in cognitive sense, this has been carried out using USRPs. The polyphase implementation has only been tested for a 4-bank system.

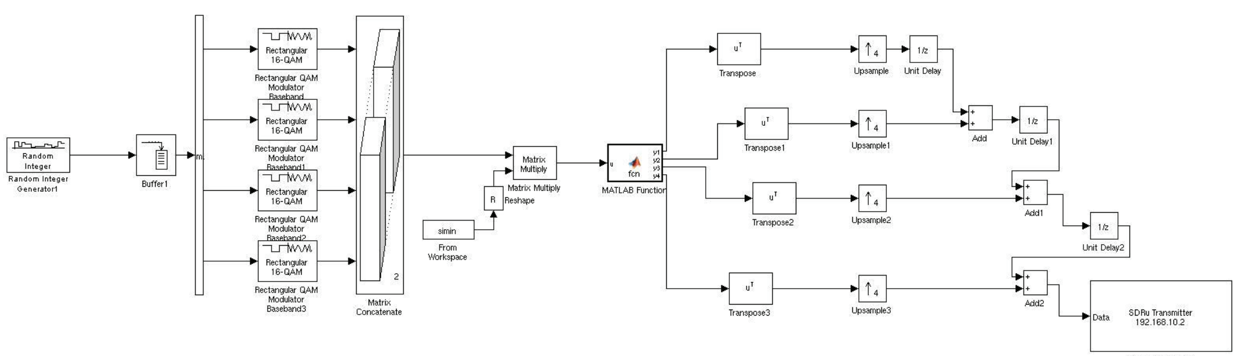


Figure 3.16: SDR implementation of the 4-bank cosine modulated filter bank multicarrier modulated transmitter.

SDR Implementation: The diagram in the earlier section shows the theorectical version of the concept to be applied and the actual implementation that is built to achieve it is shown in Figure 3.16 and it contains a few more blocks. The multiplication between the modulated input and the polyphase matrix is achieved practically as follows. The component matrix $\mathbf{R}(z)$ is created by a MATLAB program, the result of which is then imported into the Simulink model. The binary input after being demultiplexed, is 16-QAM modulated and then converted into matrix format by the matrix concatenate block. The output of this block is then matrix multiplied by the polyphase matrix that is imported. The output of matrix multiplication is then diverted into four branches by another MATLAB program, since the outputs of this program are in column format they are transposed to row format. The outputs are then upsampled, delayed and added respectively to give the FBMC signal. Another way of achieving this is by bypassing the QAM modulation process and giving the demultiplexed input directly to be multiplied by the polyphase matrix as shown in Figure 3.17. Here the signals effectively get PAM modulated and this process is proved to give smoother results than the earlier method as shown in the next section.

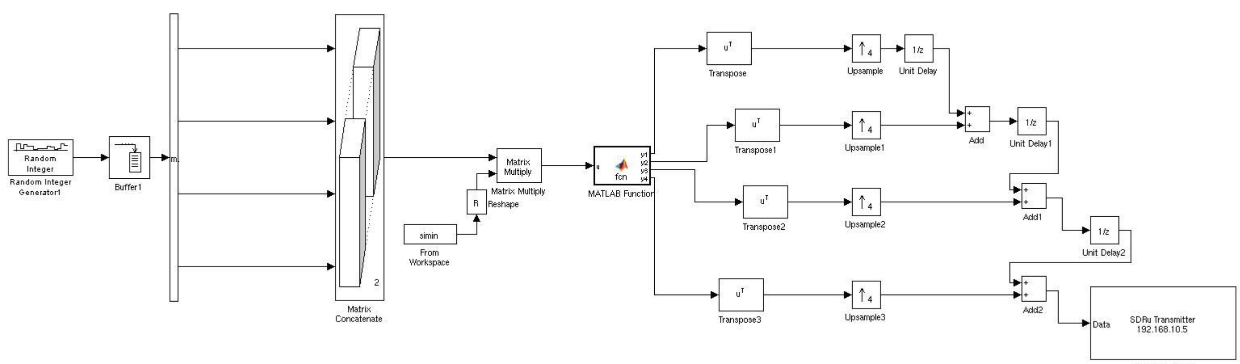


Figure 3.17: SDR implementation of the 4-bank cosine modulated filter bank multicarrier modulated transmitter.

3.4 Experimental Results

This section deals with the experimental over the air results obtained by implementing the models introduced in the earlier sections.

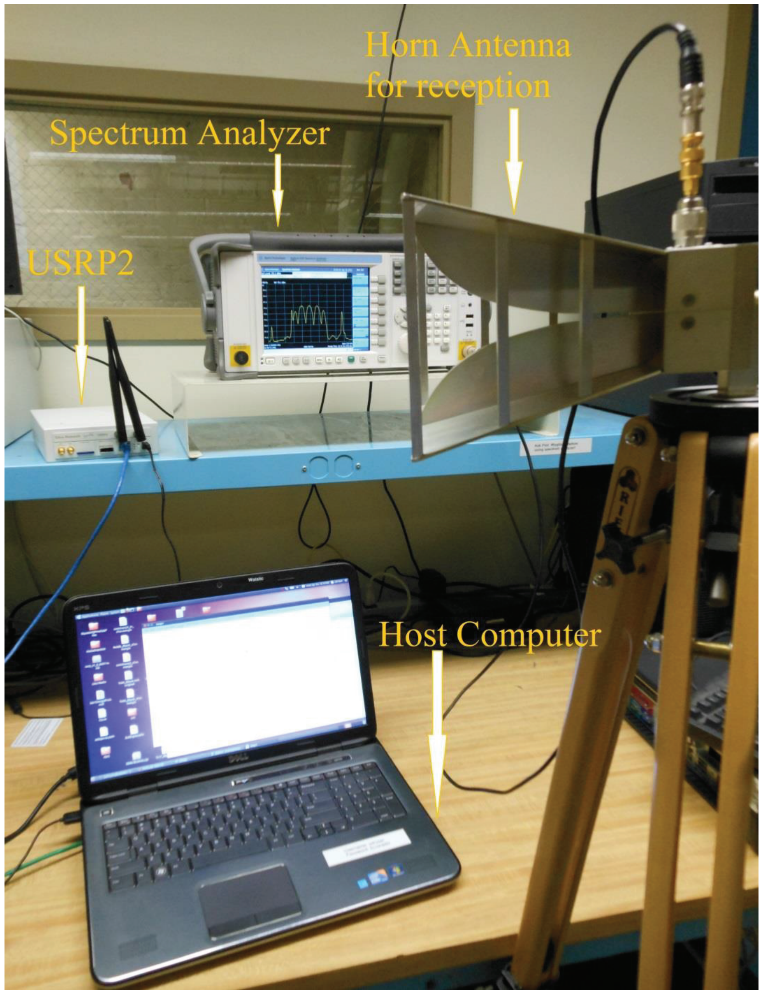
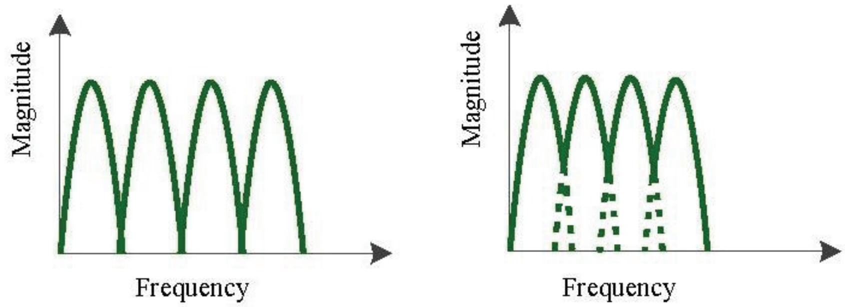


Figure 3.18: Experimental setup for the over-the-air transmissions.

Before looking at the actual results let us look at the experimental setup, shown in Figure 3.18, we have the USRP2 connected to the host computer, here a laptop, through an Ethernet cable. Here a horn antenna acts as the receiver and the data is displayed using the spectrum analyzer. It also discusses the simulated results and the performance of the implemented models when a few parameters are varied. The theorectical results of an FBMC signal system are shown in Figure 3.19, for non-overlapping and overlapping carriers.



(a) Non-overlapping carriers (b) Overlapping carrier

Figure 3.19: Theorectical results for a modulated filter bank transmitter.

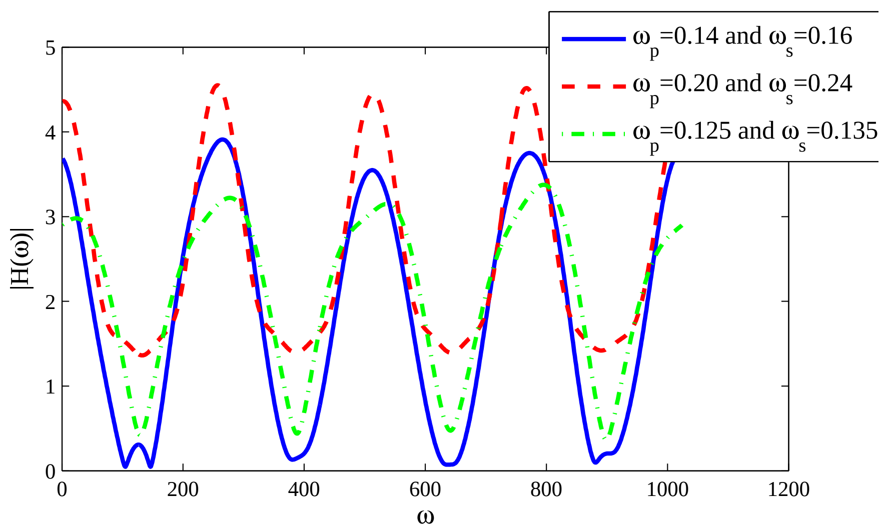


Figure 3.20: Simulated results for a modulated filter bank transmitter for the first set of carrier frequencies.

The simulated results for the modulated FBMC system are shown in Figure 3.20. The simulated results are shown for different bandwidths for the first set of carrier frequencies, $f_c = \pi/8, 3\pi/8, 5\pi/8, 7\pi/8$.

Let us now look at the experimental over-the-air results. As discussed in earlier sections, we consider the results for two sets of frequencies, where the first set is given as $[\frac{\pi}{8}, \frac{3\pi}{8}, \frac{5\pi}{8}, \frac{7\pi}{8}]$ and the second set is $[0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$. These experimental results are averaged over 200 samples. First we consider the results for cosine modulated filter banks. The Figure 3.21 shows the results for the first set of frequencies around the range $\omega_p = 0.125$ and $\omega_s = 0.135$. The frequency range can be changed to any value as long as the ω_s ranges from 0.125 to 0.25, this value depends on the number of subcarriers.

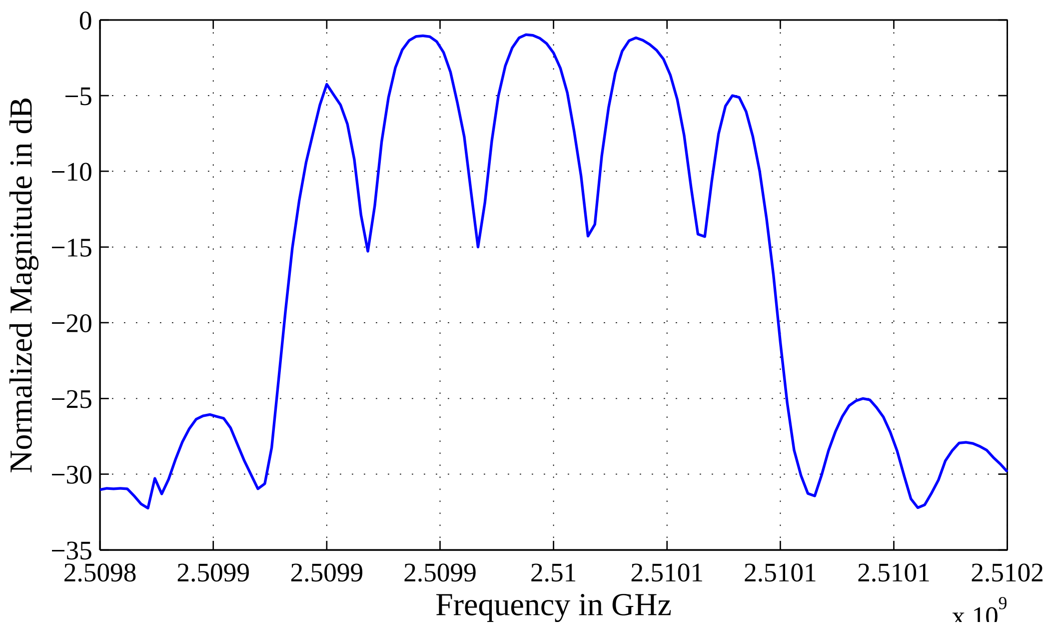


Figure 3.21: Over-the-air experimental results for cosine modulated FBMC transmitter with the first set of frequencies.

The Figure 3.22 shows the results for the second set of frequencies around the range $\omega_p = 0.20$ and $\omega_s = 0.24$. We also notice that Figure 3.21 has five peaks while Figure 3.22 has four. For a 4-bank system the expected number of peaks would be four due to the 4 carriers, but we see the extra peak in Figure 3.21 due to the selection of center frequencies which filter a portion of the upsampled signal and then add them together to give the output signal.

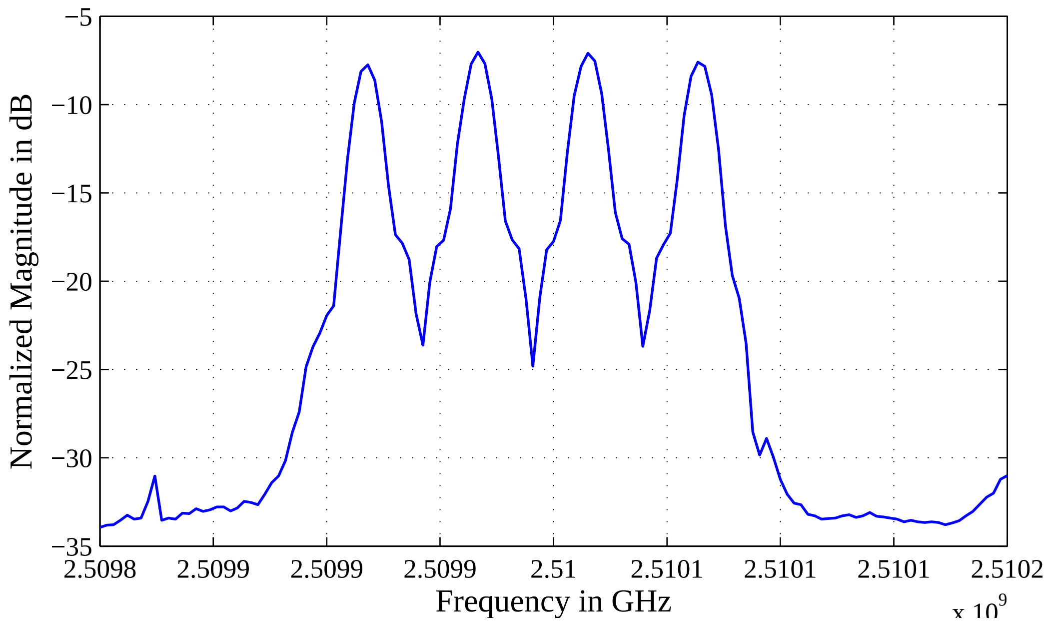


Figure 3.22: Over-the-air experimental results for cosine modulated FBMC transmitter with the second set of frequencies.

Next we consider the results for exponentially modulated filter banks which are also averaged over 200 samples, Figure 3.23 gives the results for the first set of frequencies and the Figures 3.24 for the second set of frequencies. If the transition range is reduced, keeping the stopband the same the subcarriers overlap more than usual and the result is as in Figure 3.24.

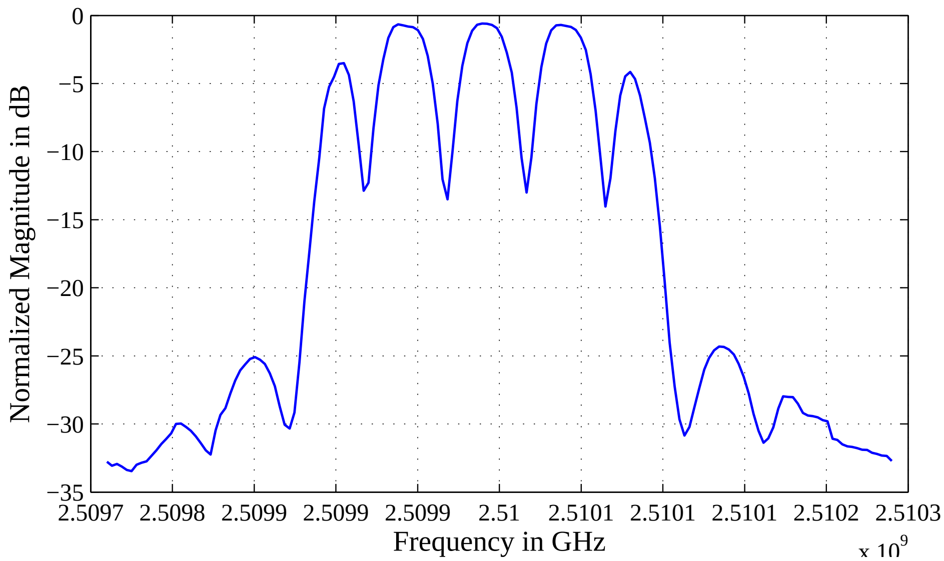


Figure 3.23: Over-the-air experimental results for exponentially modulated FBMC transmitter with the first set of frequencies.

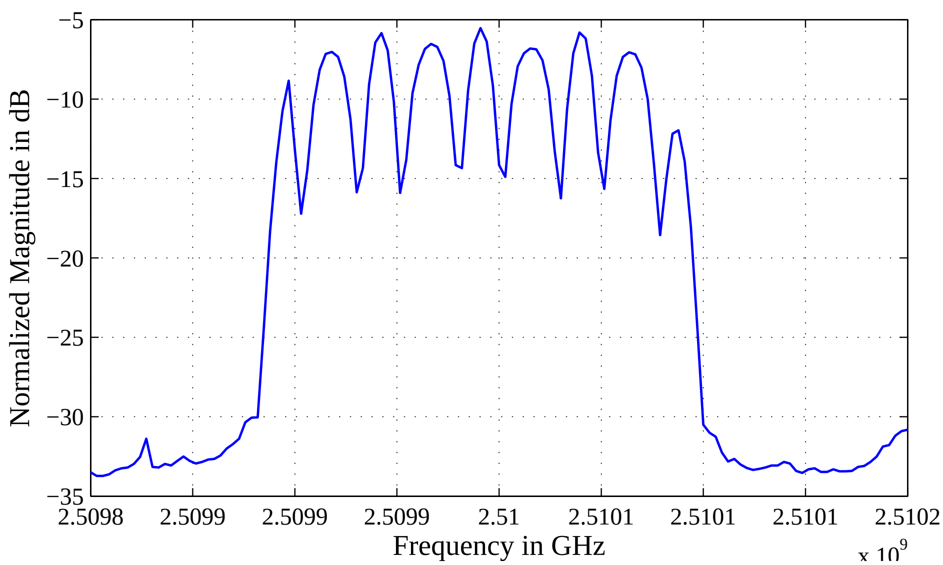


Figure 3.24: Over-the-air experimental results for exponentially modulated FBMC transmitter with the second set of frequencies, for reduced transition band.

We see that both the modulation schemes give similar results with the different frequency sets, but this doesn't hold true for NC-FBMC transmission as we will discuss in the next section. Now we look at the results for the polyphase implementation of the filter banks. Here we consider the results only for the first set of frequencies for an average of 200 samples. In Figure 3.25, the results for polyphase implementation with different modulation schemes like PAM and QAM modulation are shown.

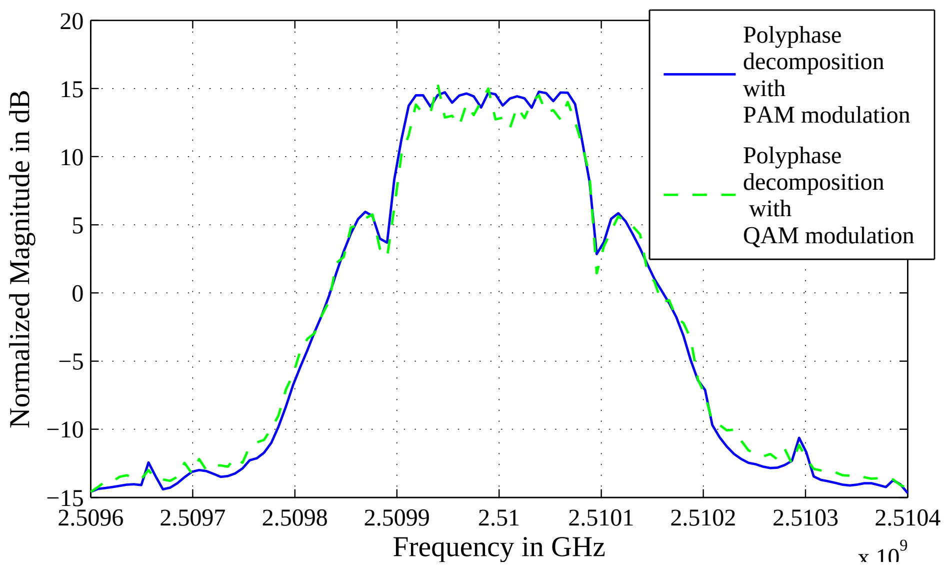


Figure 3.25: Over-the-air experimental results for polyphase implementation of the FBMC transmitter.

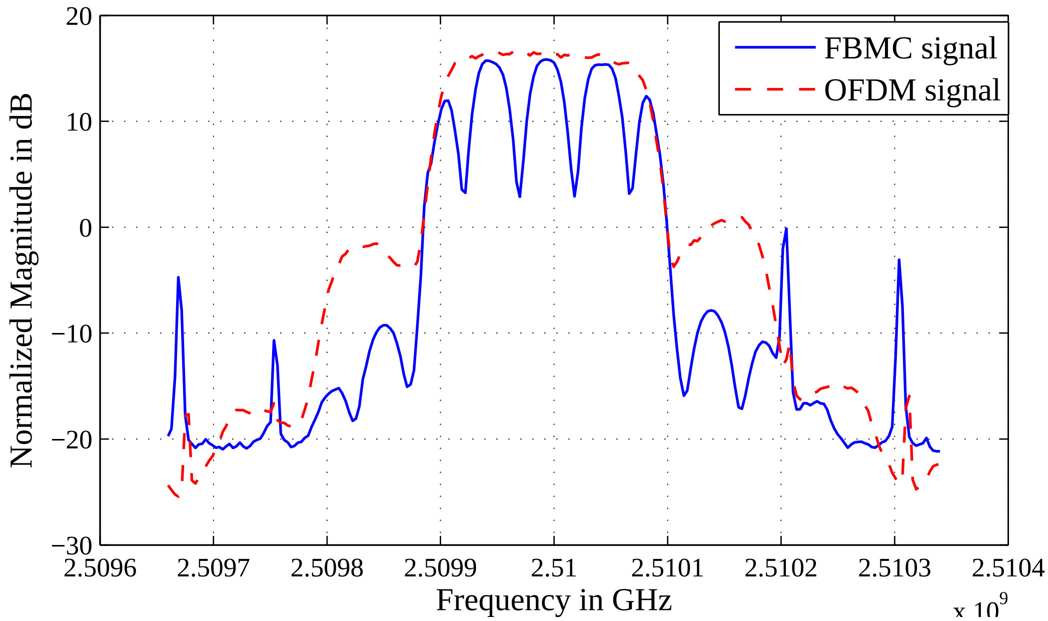


Figure 3.26: Over-the-air experimental results for over the air transmission of the FBMC and OFDM signals.

In Figure 3.26, we compare the performances of a 4-carrier FBMC and OFDM signal and we see that FBMC has better attenuation then the OFDM signal. Both the outputs were averaged over 200 samples. The spikes in the FBMC signal are caused due to external transmissions caused by the signals in the ISM band.

Polyphase implementation reduces complexity Here we are implementing the transmitter of an FBMC system using polyphase representation, so how does it actually reduce the complexity?. In our FBMC transmitter we have upsamplers followed by filters, so consider one such upsampler of factor L followed by a filter of order N. In direct implementation each sample is processed by the filter and requires $N+1$ multiplies to compute. Therefore, to get one output sample we need to perform $N+1$ multiplications. Now consider the polyphase implementation, here the filter is represented as L component filters, each of length $(N+1)/L$. Here a sample is processed by the L filters of length $(N+$

1)/L and hence $(N+1)$ multiplications are required, however each filter produces an output point. These output points from all the component filters pass through the expanders forming L output points, which after passing through the delay elements and the adder form a sequence of L output samples. Therefore we are performing $(N+1)$ multiplications to get L output samples, which means the computational complexity reduced by a factor of L to $(N+1)/L$ multiplications every output [87].

Table 3.1: Comparison chart of computational complexity of Multicarrier Modulation implementation methods.

Type	Number of Multiplications	Number of Additions
OFDM using FFT of length N ($N = 2^k$, where k is integer)	$\frac{N}{2}\log_2 N$	$N\log_2 N$
FBMC For a single filter - General filter of order N	$N+1$	N
Symmetrical N odd	$\frac{N}{2}+1$	N
N even	$\frac{N+1}{2}$	N
Polyphase Representation For a single filter L is the upsampling factor	$\frac{N}{L}$	N

Table 3.1 gives you a quantitative analysis of the computational complexity of the various approaches to implement multicarrier modulation. The data for the FBMC and polyphase implemetation are given for a single filter in a M-bank approach and thus accordingly the overall complexity changes. From the table, it is apparent that OFDM is the least computationally complex of all but for spectrally agile waveform transmission OFDM usually requires additional processing like cancellation carriers and filtering with windowing, in addition to the cyclic prefix processing. All these factors do dull the edge OFDM has over the others in computational complexity.

3.5 Chapter Summary

This chapter introduces and explains the proposed implementations for generating an FBMC signal by employing the cosine modulated and exponentially modulated filter bank systems. It discusses the Simulink models used for implementing the procedures. It also gives a detailed explanation about polyphase representation of systems and the proposed polyphase implementation of filter banks. It includes the experimental over the air results for the Simulink models discussed in this chapter.

Chapter 4

Proposed Implementation of Non-contiguous Filter bank Multicarrier Modulation

With the realization of the SDR implementation of filter bank multicarrier modulated transmitter, the subsequent procedure is to modify it to be capable of segregated agile transmission. This chapter demonstrates the adaptation of the proposed techniques to implement non-contiguous FBMC transmission for adaptable transmission by spectrally agile waveforms. The non-contiguous version of the transmitter should be capable of switching off unnecessary carriers, either to avoid other users or a patch of highly frequency selective fading whatever the impediment may be. The process in the development of this system is detailed in this section, including evaluation of the potential techniques of implementation. The desirability of cosine and exponentially modulated filter banks for non-contiguous transmission is examined.

4.1 Schematic of the proposed Non-contiguous FBMC implementation

This section gives the proposed schematic for achieving our objective of segregated transmission. The following diagrams gives the diagrammatic representation of the proposed design. We see that the schematic is largely similar to FBMC systems in the earlier chapters. The major difference lies in the center frequencies allocated or more specifically not allotted. Two versions of this technique have been implemented, they slightly differ in the way the filters are not allocated frequencies. The first version is as follows

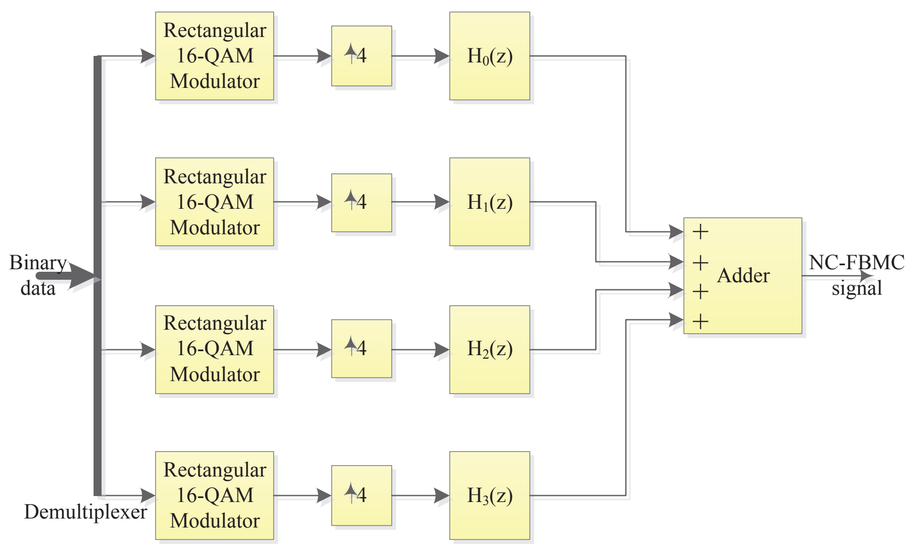


Figure 4.1: Schematic illustration of the first proposed implementation for achieving non-contiguous FBMC transmission.

In this version to achieve non-contiguous transmission, the filters with center frequencies that are not needed are effectively switched off. The second version is as follows In this version to achieve non-contiguous transmission, the filters are allocated only the required center frequencies and the unnecessary frequencies are not allocated to any filter. Therefore

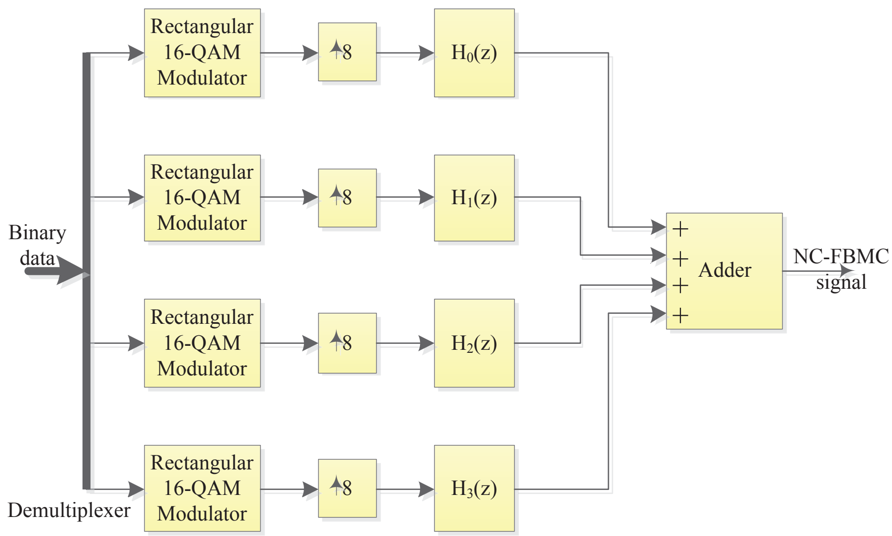


Figure 4.2: Schematic illustration of the alternate implementation for achieving non-contiguous FBMC transmission.

in this implementation an N-bank system may not have N branches but still implements an N-bank system. In this thesis, we mostly use this method for NC-FBMC transmission, as it was observed to give better performance.

4.1.1 Proposed Implementation

In non-contiguous FBMC transmission a matter of import is the modulation of filter banks. Even though, both cosine and exponential modulation have given similar results in conventional transmission, we find that it is not the same in this case. Consider cosine modulation, as discussed before the filtered signal has a two sided frequency response, so when a filter is switched off only half the signal is switched off in two sides of the signal spectrum. Therefore we fail to achieve perfect attenuation in the switched off frequency. On the other hand, exponential modulation has a one sided frequency response and thus when the filter is switched off the signal in the said frequency is totally switched off. Thus only exponential modulation is used here for non-contiguous signal transmission.

Now let us consider the first proposed implementation, here the design is similar to that of a conventional exponentially modulated transmultiplexer. It varies only in the design of the filter banks which we will now discuss. The filters with frequencies which are not needed are essentially switched off, this is achieved by making the filter response $H_k(z) = 0$. This ensures that no data passes through that frequency as it acts like a stopband filter. Thus any data in the channel with this filter gets nullified, thus effectively switching off this channel.

The alternate approach achieves switching off carriers in a slightly different manner, here filters are only allotted required frequencies. For example consider an 8 bank system like in the figure, consider the frequencies $\frac{3\pi}{16}, \frac{5\pi}{16}, \frac{11\pi}{16}, \frac{13\pi}{16}$ need to be switched off, then the filters are only allocated frequencies $\frac{\pi}{16}, \frac{7\pi}{16}, \frac{9\pi}{16}, \frac{15\pi}{16}$ that is the filters with unnecessary frequencies are not even included in the system. It is for this reason that in this implementation, an N-bank system does not always have N branches. The upsampling factor for this model depends on the actual system being implemented and not the number of channels in the implementation.

SDR implementation The SDR implementation for NC-FBMC system is direct and follows the same format as the schematic given. The filters are implemented by MATLAB functions as in earlier models.

4.2 Experimental Results

This section presents the experimental over the air results for the non-contiguous FBMC system. As discussed previously, we only employ exponentially modulated filter banks in this implementation and all the results are averaged over 200 samples. Now let us look at the obtained results while transmitting with a USRP2. We will observe the results for different number of switched off carriers as well as their positioning for a 16-bank NC-FBMC system.

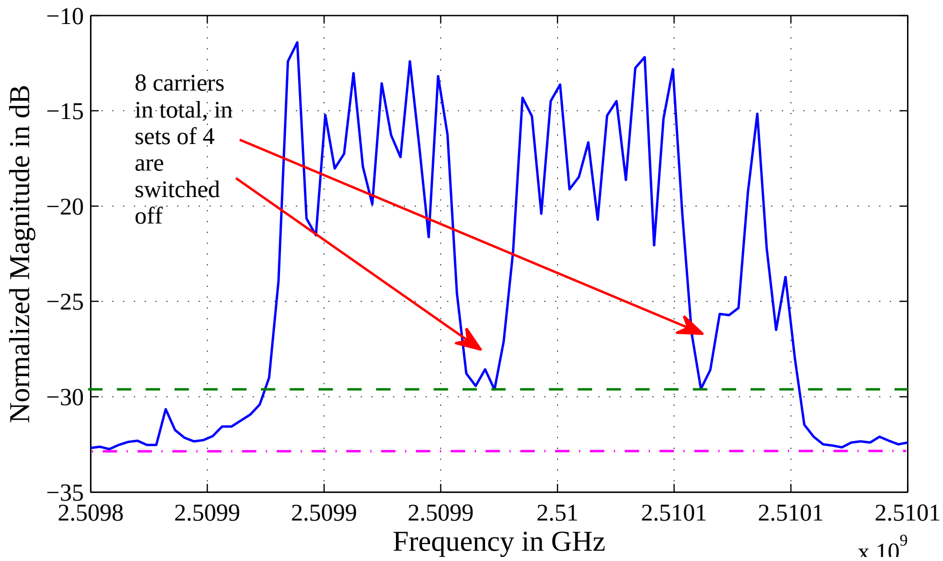


Figure 4.3: Over-the-air experimental results for exponentially modulated NC-FBMC transmitter with two sets of 4-carriers switched off.

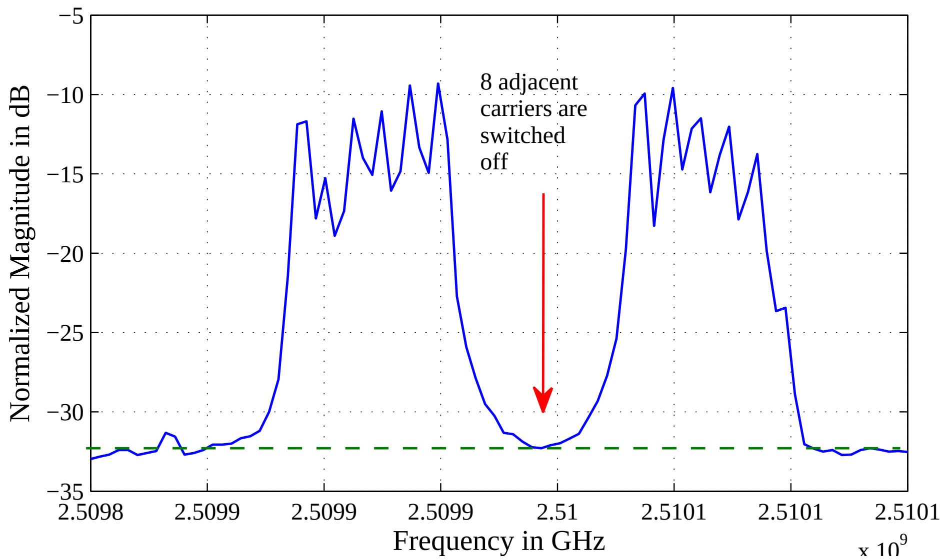


Figure 4.4: Over-the-air experimental results for exponentially modulated NC-FBMC transmitter with 8-carriers switched off.

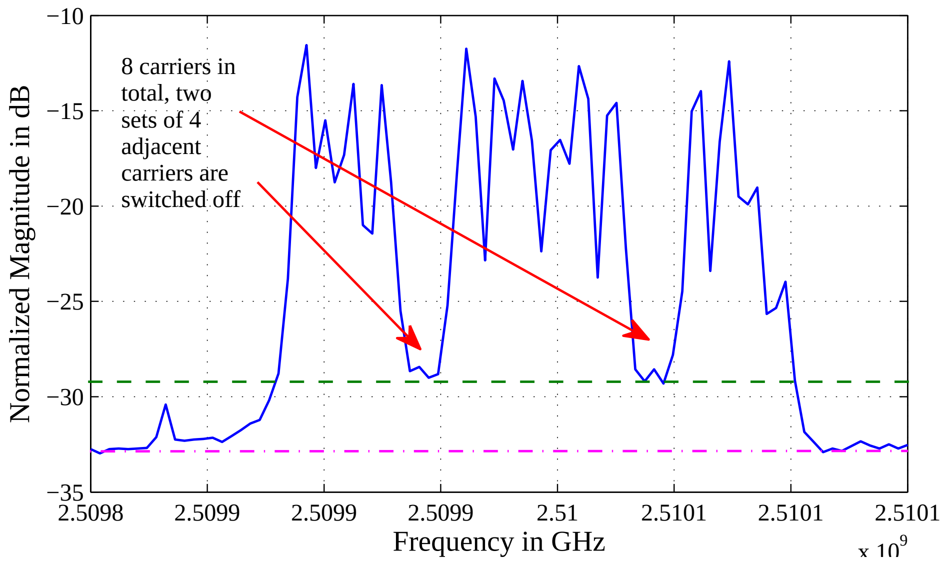


Figure 4.5: Over-the-air experimental results for exponentially modulated NC-FBMC transmitter with two sets of 4-carriers switched off.

Figures 4.3 and 4.5 depicts the results when two sets of 4-carriers, that is a total of 8 carriers are switched off. We notice that the positioning of the switched off carriers is also effecting how smooth the transition to the stopband is. The attenuation in both the cases is similar. In Figure 4.3, the effective out-of-band attenuation is around 21.21dB, while the effective attenuation achieved from switching off carriers is 18.21dB. In Figure 4.5, the effective out-of-band attenuation is 21.41dB, while the effective in-band attenuation is 17.75dB. All these results were obtained averaging 200 samples. The Figure 4.4 depicts the case when a bunch of adjacent carriers are switched off, here 8 carriers, the effective out-of-band attenuation in this case is 23.66dB and the effective attenuation from switching off carriers is 22.97dB. We notice that the attenuation is smoother and better with 8 carriers switched off, this attenuation improves with the number of carriers being switched off to an extent and then stabilizes. This is a good effect as in practical use the number of carriers would be far more than 16, which is the number we consider here, therefore the carriers switched off would also increase, thus we get good attenuation without the need for additional processing.

Let us now consider NC-FBMC and NC-OFDM signals averaged over 200 samples and compare their performances. Here we can only do a relative comparison, rather than an exact comparison due to the fact that both signals are obtained through different methods and as such all their parameters are different for example carrier frequencies, which would make the results from an attempt at exact comparison inaccurate. The NC-OFDM signal has been obtained from the method employed in [88]. The NC-FBMC signal is shown in Figure 4.6 and the NC-OFDM signal is as in Figure 4.7.

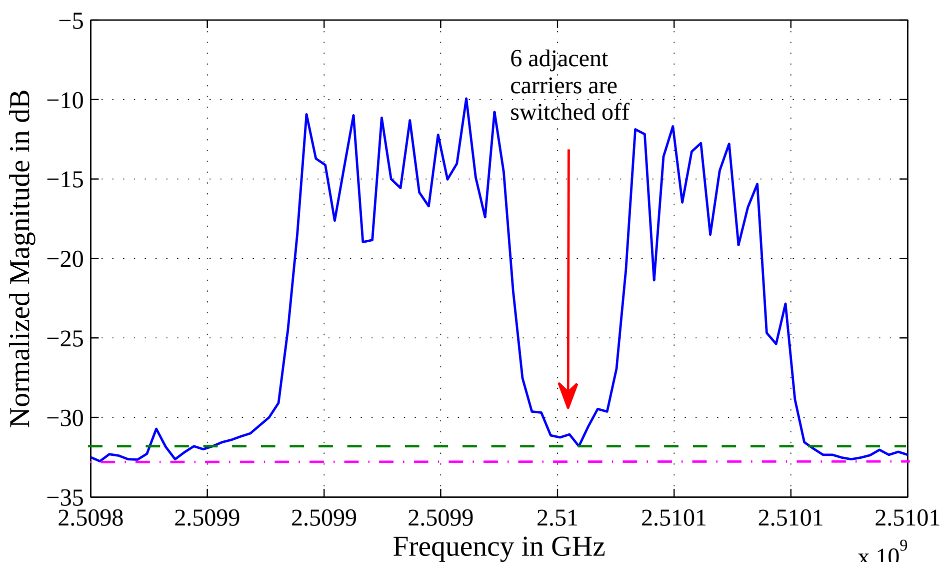


Figure 4.6: Over-the-air experimental results for exponentially modulated NC-FBMC transmitter with 6-carriers switched off.

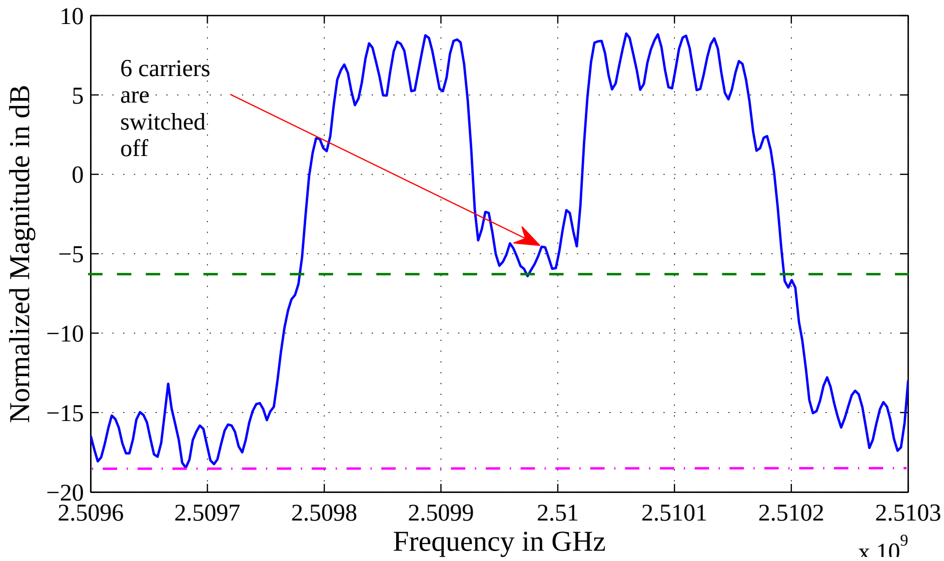


Figure 4.7: Over-the-air experimental results for NC-OFDM transmitter with 6-carriers switched off.

The NC-FBMC signal has an effective out-of-band attenuation of 22.56dB and an effective attenuation of 21.87dB where the carriers are switched off. The NC-OFDM signal has an effective out-of-band attenuation of 27.05dB and an effective attenuation of 14.98dB where the carriers are switched off. Here we notice that the relative attenuation achieved by the NC-FBMC transmitter is more than that achieved by the NC-OFDM transmitter. One reason for this is the high sidelobes of the OFDM signal, therefore, here if we notice the frequency band where the carriers where switched off, we see that NC-FBMC offers relatively better attenuation than NC-OFDM.

4.3 Chapter Summary

This chapter discusses the implementation of the non-contiguous version of the FBMC system. In this section only exponentially modulated filter banks are used for NC-FBMC signal generation. Two different procedures for implementing NC-FBMC transmission are covered. The method where filters are not assigned to unnecessary frequencies is chosen as the choice of implementation. The proposed procedure and the implemented model are discussed. The experimental over-the-air results for varying number of switched off carriers are included.

Chapter 5

Conclusion

In this thesis, we have aimed at the practical implementation of an FBMC system, including a non-contiguous version. We have explored various approaches to achieve it and discussed the performance in a real working environment. The overall achievements of this thesis are:

- An SDR framework for implementing a filterbank multicarrier-based transmitter was achieved using cosine modulated and exponentially modulated prototype lowpass filter techniques. The implementations were studied via over-the-air experimentation and compared with a conventional OFDM system, showing significant improvement in terms of OOB (out-of-band emission) reduction.

- This work was extended to non-contiguous scenarios that would be experienced in cognitive radio and dynamic spectrum access environments. A relative comparison of the results showed that the OOB reduction performance of the FBMC implementations exceeded that of the NC-OFDM transmitter implementation.

5.1 Future work

There is potential for further research in the areas discussed here and as well as related areas.

- Implementing a non-uniform version of the FBMC on an SDR platform. There are several methods to achieve this and each version provides a new area for research. Non-uniform FBMC can be achieved by using different modulation schemes for the N channels. It can also be achieved by using different versions of the same modulation scheme for example M-QAM with different bit allocations through an adaptive commutator.

- The non contiguous version of it would be the next step, as non-uniform modulation would be a good means to achieve greater efficiency. Modulation schemes with greater bit rate can be used for those channels with least noise corruption. Thus using the knowledge of the channel noise the least amount of data can be sent on the channel with most suppression.

- A variable power FBMC can be developed, in this version the power allocated to each subcarrier varies, thus making making it highly effective againt frequency selective attenuation in the channels. The power of each subcarrier is regulated so as to give the most throughput.

- The polyphase model can be further developed especially to implement the non-contiguous FBMC system. As the computational complexity of this model is considerably less than the conventional models it would further enhance the suitability of FBMC as the multicarrier modulation of choice for cognitive wireless transmissions.

- The NC-OFDM structure that has been proposed is controlled manually, therefore a way to automize it would be an area of further innovation. All these structure can be updated to include carrier sensing and automatic center frequency allocation, thus taking the whole process to the next level in spectrally agile waveform design.

Appendix A

Quadrature Amplitude Modulation

Quadrature amplitude modulation is a modulation scheme, where two carriers are modulated according to the data signal rather than the conventional single carrier. The two carriers are 90° out-of-phase with each other, therefore the term "quadrature". Even though here we are interested in the digital domain of things, the quadrature amplitude modulation can be discussed in the continuous and discrete sense to get a better understanding.

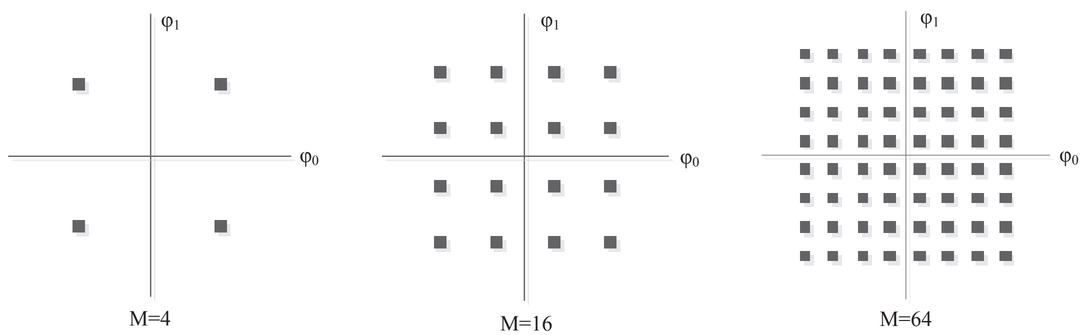
Quadrature amplitude modulation is a form of linear modulation, where the system can be considered to be constructed on a set of basis functions, which are tailored and added to give the required waveforms. Consider a set of M symbols that need to be transmitted, given by the signal set S.

$$S = \{s_0(t), s_1(t), \cdots, s_{M-1}(t)\} \tag{A.1}$$

The M symbols are mapped to M $\log_2 M$-bit patterns, which are then represented using the combination of basis functions. These signal sets are represented using signal spaces. The signal sets can depend on a single basis function or two basis functions. The signal set depending on two basis functions constitutes quadrature amplitude modulation, and the set of orthonormal basis functions can be described as

$$\phi_0(t) = \sqrt{2}p(t)\cos\omega_0 t \qquad \phi_1(t) = -\sqrt{2}p(t)\sin\omega_0 t \tag{A.2}$$

here $p(t)$ can be any pulse of unit energy. Thus the carriers are based on these sinusoids that are 90° out of phase with each other. An M waveform system described above is represented as an $M - ary$ system, in this case an $M - ary$ QAM system. The common QAM systems are $M = 2, 4, 8, 16, \cdots$, The QAM signal constellations is usually rectangular and square constellations are the most popular and commonly used. Some conventional signal constellations are



(a) (b) (c)

Figure A.1: Graphical representation of three commonly used QAM signal constellations.

The general $M - QAM$ signal can be represented as

$$s(t) = \sqrt{2}\sum_k a_0(k)p(t - kT_s)\cos\omega_0 t - a_1(k)p(t - kT_s)\sin\omega_0 t \tag{A.3}$$

which can be written as

$$s(t) = I(t)\sqrt{2}\cos\omega_0 t - Q(t)\sqrt{2}\sin\omega_0 t \tag{A.4}$$

for convenience, where

$$\left. \begin{aligned} I(t) &= \textstyle\sum_k a_0(k)p(t-kT_s) \\ Q(t) &= \textstyle\sum_k a_1(k)p(t-kT_s) \end{aligned} \right| \tag{A.5}$$

are pulse amplitude modulated (PAM) pulse trains. They are known as the "inphase" and "quadrature" components of signal $s(t)$ respectively.

The k^{th} symbol of the signal

$$a_0(k)p(t-kT_s)\sqrt{2}\cos\omega_0 t - a_1(k)p(t-kT_s)\sqrt{2}\sin\omega_0 t \tag{A.6}$$

can also be declared as

$$\sqrt{a_0^2(k)+a_1^2(k)}p(t-kT_s)\sqrt{2}\cos{(\omega_0 t + \tan^{-1}\frac{a_1(k)}{a_0(k)})} \tag{A.7}$$

From the first equation, we see that the transmitted signal is a sum of sinusoids whose amplitudes are governed by the symbols $a_0(k)$ and $a_1(k)$ which depend on the data bits. This form is known as the rectangular form. In the second equation, it is dependent on a sinusoid whose phase and magnitude are regulated by the data bits and thus it is known as the polar form.

In the continuous domain the QAM modulator has the form given in Figure A.2.

A serial bit stream forms the input, with a new bit every T_b seconds. The serial to parallel converter partitions the bits into $\log_2 M$ bits which then act as a address at the lookup tables which contain the constellation symbol values. These values are generated every $T_s = T_b/\log_2 M$ seconds. A pulse shaping filter with an impulse train input in both arms, forms the weighted pulse trains for the basis functions. The mixers then multiply them with the sinusoids. $I(t)$ and $Q(t)$ are the inphase and quadrature pulse trains, which modulate the carriers to form the two components which make up the QAM signal [89].

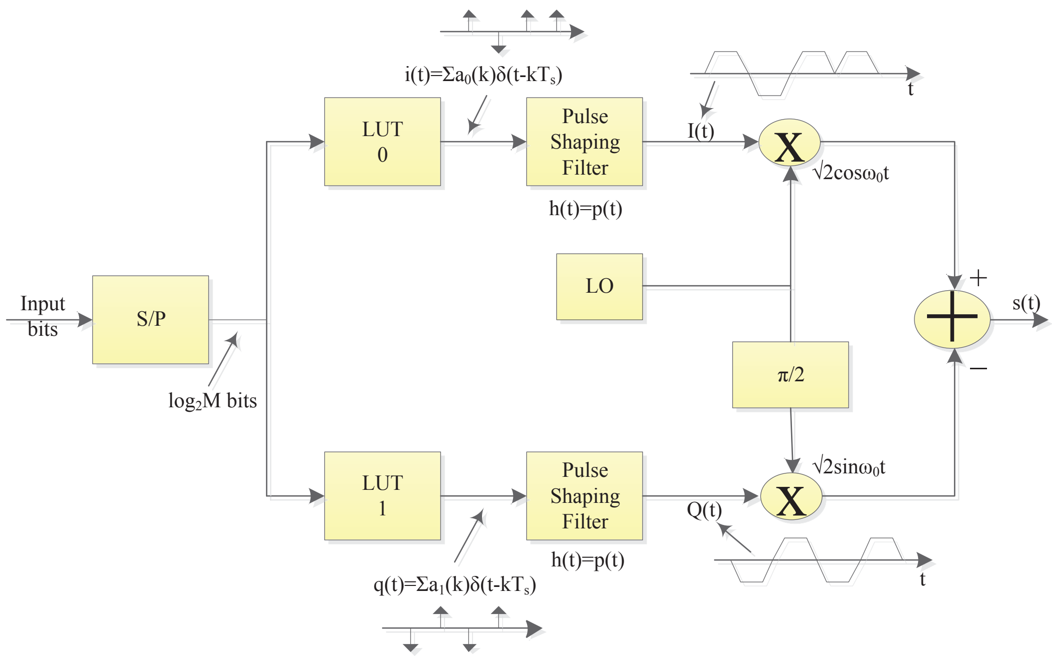


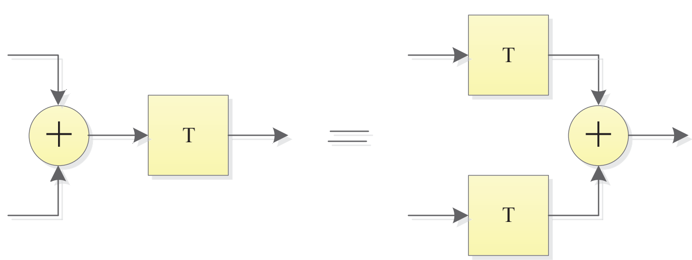
Figure A.2: Block diagram of a generic QAM modulator in the continuous domain.

Appendix B

Polyphase Representation

B.0.1 General Cascade Identities

Let us now look at some of the general identities and rules that are applied while simplifying complex interconnections of the blocks during the polyphase representations.



(a) An adder, sampler connection.

(b) A multiplier, sampler cascade.

Figure B.1: Adders and multipliers in a connection can be commuted across the sample rate converters.

It is important to remember that decimators and expanders are linear time variant operators, therefore their order of operation is crucial. Adders and multipliers which are memoryless operators can be moved across the expanders and decimators.

A series of sampling rate expanders or compressors can be alternately expressed as

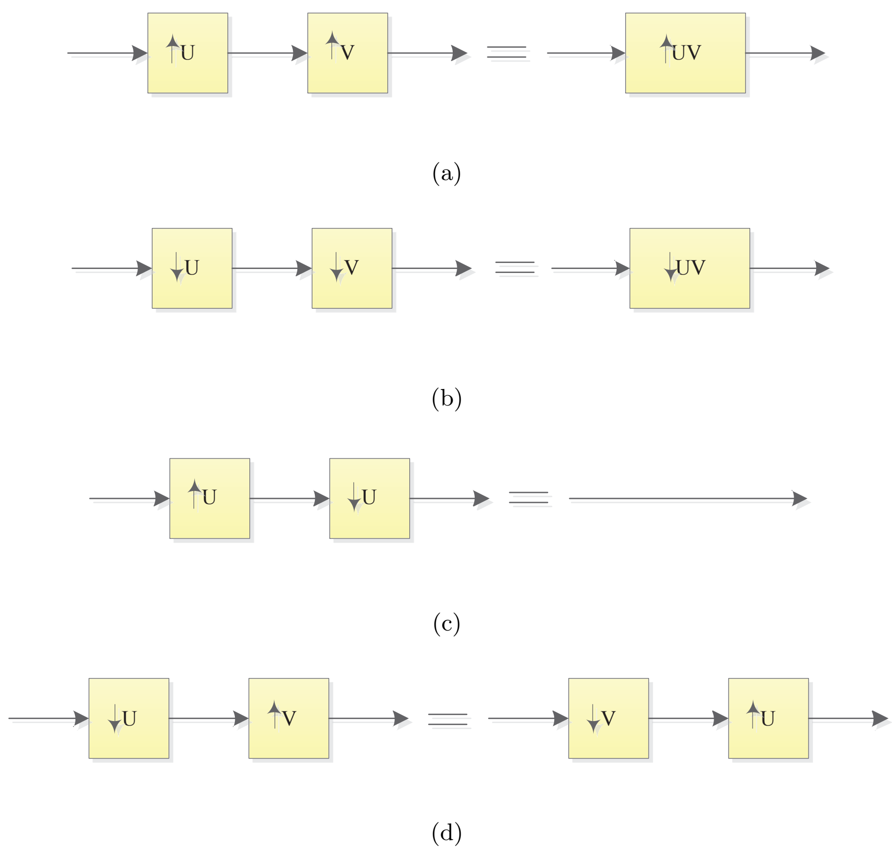


Figure B.2: Diagrammatic representation of some general cascade identities.

It is important to note that the following case doesn't have an equivalent representation

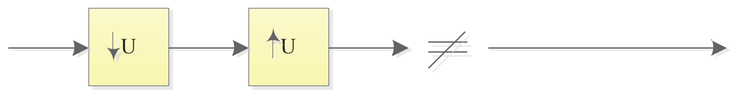
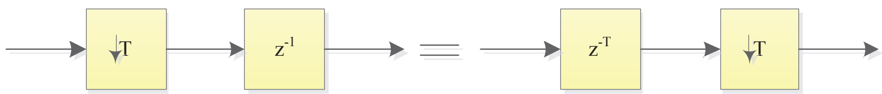
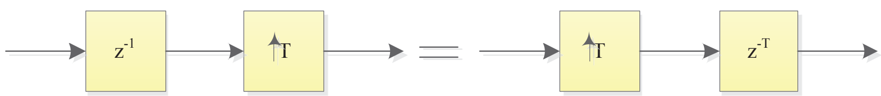


Figure B.3: A case of sampler interconnection that cannot be simplified.

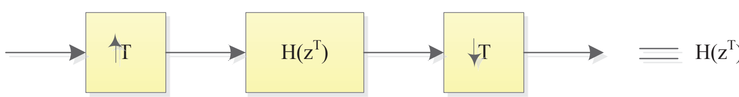
A series of blocks with delays can be also represented as



(a)



(b)



(c)

Figure B.4: Diagrammatic representation of some general cascade identities involving delays.

B.0.2 Why the name polyphase decomposition?

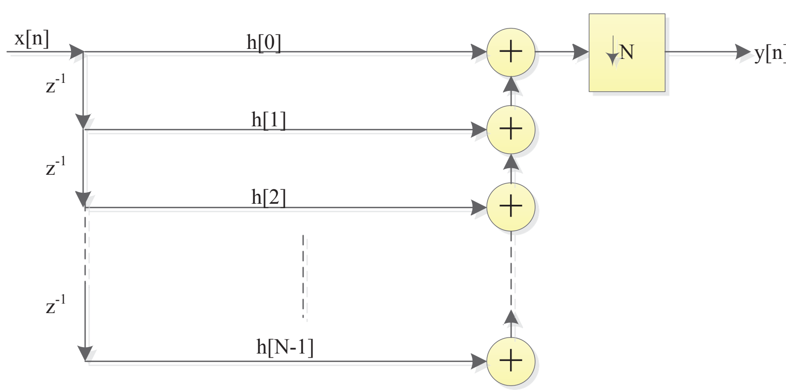
The impulse response $e_k(z)$ of a polyphase component $E_k(z)$ of a system $H(z)$ is the decimated version of $h(n+k)$. Thus the polyphase component $E_k(e^{j\omega})$ is the downsampled version of $e^{j\omega}H(e^{j\omega})$. These components are the aliased versions such that the overall function has the appearance of an allpass function scaled by $1/M$. In the passband region the M terms, each of magnitude $1/M$ sum together to almost unity. This can be given by the equation

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M) \tag{B.1}$$

This shows that the M terms are almost in phase, but for the Mth band filter $E_0(z)$ is constant. The phase responses of $z^{-k}E_k(z^M)$ are almost zero in the passband, which means $E_k(e^{j\omega})$ tries to approximate $e^{j\omega k/M}$. Therefore the phase response of the kth polyphase component $\phi_k(\omega)$, tries to approximate $\omega k/M$ for every k. Thus the term "polyphase" is used.

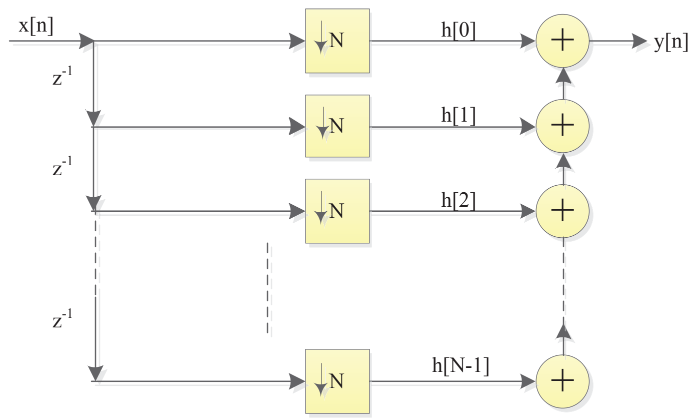
B.0.3 How it actually reduces complexity?

This section helps to better understand how polyphase representation of filter banks reduces computational complexity. Consider a cascade of a filter followed by a downsampler. We apply polyphase decomposition to the filter and thus it is now defined in terms of its polyphase components.



(a) The direct form implementation of a filter followed by a downsampler.

We notice that we use only one out of every N output samples of the filter due to the $N : 1$ downsampler, i.e. the other $N-1$ samples are redundant. Thus we can commute



(b) The implementation when the downsampler has been commuted into the filter structure.

Figure B.5: The diagram depicts the advantage offered by polyphase decomposition.

the downsampler through the adders, so now it is within the polyphase representation of the filter. In this arrangement the multipliers are operating at $1/N$ times the sampling frequency of the input signal, thus reducing the computational requirements by a factor of $1/N$.

Appendix C

MATLAB codes

C.1 MATLAB code for the filter function block in Cosine modulated FBMC SDR implementation

```
function y = fcn(u)
t = 1:1:100;
%freqs = [0 0.20 0.24 1];
freqs = [0 0.125 0.135 1];    %The range of the passband and the stopband can be
%freqs = [0 0.14 0.16 1];
amps = [1 1 0 0];
b = firls(99, freqs, amps);   %Lowpass FIR prototype filter
b1 = 4*b.*cos(2*pi*(1/8).*t); %Cosine modulation
y = filter(b1,1,u);
```

C.2 MATLAB code for the filter function block in Exponentially modulated FBMC SDR implementation

```
function y = fcn(u)
```

```
t = 1:1:100;
freqs = [0 0.1 0.11 1]; %Frequency selection of the lowpass prototype
amps = [1 1 0 0];
b = firls(99, freqs, amps); %Lowpass FIR prototype filter
b2 = 4*b.*exp(2j*pi*(1/4).*t); %Exponential modulation of the prototype
%b2 = 0;                  %This line can be used to switch off this filter
y = filter(b2,1,u);
%y = 0;                   %Alternate way of switching off the filter
```

C.3 MATLAB code for generating the Polyphase Matrix

```
%This code creates the polyphase matrix for the polyphase implementation of
%the exponentially modulated filter banks.
%The filter co-efficients of each filter become matrix elements as expained
%in section 3.3.6
clc;
clear all;
close all;
[e1,e2,e3,e4,d1,d2,d3,d4,f1,f2,f3,f4,g1,g2,g3,g4]=deal(zeros(1,25));
freqs = [0 0.2 0.25 1];
amps = [1 1 0 0];
t = 1:1:100;
b = firls(99,freqs,amps); %Lowpass FIR prototype filter
b1 = 4*b.*exp(2j*pi*(1/8).*t); %Exponentially modulated to f_c=pi/8
for i = 1:length(b1)/4;
    e1(i) = b1(4*(i-1)+1);
    e2(i) = b1(4*(i-1)+2);
    e3(i) = b1(4*(i-1)+3);
```

```
    e4(i) = b1(4*(i-1)+4);
end;
b2 = 4*b.*exp(2j*pi*(3/8).*t); %Exponentially modulated to f_c=3pi/8
for i = 1:length(b2)/4;
    d1(i) = b2(4*(i-1)+1);
    d2(i) = b2(4*(i-1)+2);
    d3(i) = b2(4*(i-1)+3);
    d4(i) = b2(4*(i-1)+4);
end;
b3 = 4*b.*exp(2j*pi*(5/8).*t); %Exponentially modulated to f_c=5pi/8
for i = 1:length(b3)/4;
    f1(i) = b3(4*(i-1)+1);
    f2(i) = b3(4*(i-1)+2);
    f3(i) = b3(4*(i-1)+3);
    f4(i) = b3(4*(i-1)+4);
end;
b4 = 4*b.*exp(2j*pi*(7/8).*t); %Exponentially modulated to f_c=7pi/8
for i = 1:length(b4)/4;
    g1(i) = b4(4*(i-1)+1);
    g2(i) = b4(4*(i-1)+2);
    g3(i) = b4(4*(i-1)+3);
    g4(i) = b4(4*(i-1)+4);
end;
simin.signals.values = [e4 d4 f4 g4 e3 d3 f3 g3 e2 d2 f2 g2 e1 d1 f1 g1];
simin.time = [];
simin.signals.dimensions = [1 400];
```

Bibliography

[1] K. Pretz, "Overcoming spectrum scarcity." `http://theinstitute.ieee.org/technology-focus/technology-topic/overcoming-spectrum-scarcity`, 2012.

[2] T. Luo, F. Lin, T. Jiang, M.Guizani, and W. Chen, "Multicarrier modulation and cooperative communication in multihop cognitive radio networks," *IEEE Wireless Communications*, vol. 18, pp. 38–45, Feb. 2011.

[3] S.Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE Journal on selected ares in Communication*, vol. 23, pp. 201–220, Feb. 2005.

[4] D. G. Dola Saha, Aveek Dutta and D. Sicker, "Blind synchronization for nc-ofdm - when channels are conventions, not mandates," *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks(DySPAN)*, pp. 552–563, May 2011.

[5] S. P. Hanna Bogucka, Alexander M.Wyglinski and A. Kliks, "Spectrally agile multicarrier waveforms for opportunistic wireless access," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 108–115, 2011.

[6] A. Wyglinski, *Physical Layer Loading Algorithms for Indoor Wireless Multicarrier Systems.* PhD thesis, McGill University, 2004.

[7] B. Farhang-Boroujeny, "Ofdm versus filter bank multicarrier," *IEEE Signal Processing Magazine*, vol. 28, pp. 92–112, May 2011.

[8] M. G.Bellanger, "Specification and design of a prototype filter for filter bank based multicarrier transmission," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2417–2420, 2001.

[9] J. A. Viholainen and M.Renfors, "Implementation of parallel cosine and sine modulated filter banks for equalized transmultiplexer systems," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3625–3628, 2001.

[10] B. F-Boroujeny and C. Yuen, "Cosine modulated and offset qam filter bank multicarrier techniques: A continuous-time prospect," *EURASIP Journal on Advances in Signal Processing*, Feb. 2010.

[11] R.D.Koilpillai and P.P.Vaidyanathan, "New results on cosine-modulated fir filter banks satisfying perfect reconstruction," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1793–1796, Apr. 1991.

[12] J.Alhava and M.Renfors, "Exponentially-modulated filter bank transmultiplexer with fine-coarse adaptive filtering," *International Symposium on Communications, Control and Signal Processing*, pp. 68–72, Mar. 2008.

[13] J.Alhava and M.Renfors, "Exponentially-modulated filter bank-based transmultiplexer," *International Symposium on Circuits and Systems (ISCAS)*, vol. 4, pp. 233–236, May 2003.

[14] A. Viholainen, T.Ihalainen, T.H.Stitz, M.Renfors, and M.Bellanger, "Prototype filter design for filter bank based multicarrier transmission," *17th European Signal Processing Conference*, pp. 1359–1363, Aug. 2009.

[15] J. III and J. G.Q.Maguire, "Software defined radio: Facets of a developing technology," *IEEE Personal Communications*, vol. 6, pp. 13–18, Aug. 1999.

[16] J. M. III, "Cognitive radio for flexible mobile multimedia communications," *IEEE Internal Workshop on Mobile Multimedia Communications*, pp. 3–10, 1999.

[17] J.Mitola, "The software radio architecture," *IEEE Communications Magazine*, vol. 33, pp. 26–38, May 1995.

[18] H. Maldi, "A novel structure for implementating dft-filter banks," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7, pp. 506–509, May 1982.

[19] K. Swaminathan and P. Vaidyanathan, "Theory and design of uniform dft, parallel, quadrature mirror filter banks," *IEEE Transactions on Circuits and Systems*, vol. 33, pp. 1170–1191, Dec. 1986.

[20] N. Fliege, "Modified dft polyphase sbc filter banks with almost perfect reconstruction," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 149–152, Apr. 1994.

[21] P.-L. Shui and J.-Z. Jiang, "Two-dimensional twice oversampled dft modulated filter banks and critically sampled modified dft modulated filter banks," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5597–5611, Nov. 2010.

[22] J.-Z. J. Peng-Lang Shui and Z.-J. Zhang, "Design of oversampled dft-modulated filter banks via modified newton's method," *IEEE Transactions on Signal Processing*, vol. 5, pp. 271–280, June 2011.

[23] U. R. T. Hunziker and D. Dahlhaus, "Spectrum sensing in cognitive radios: Design of dft filter banks achieving maximal time-frequency resolution," *International Conference on Information and Communications and Signal Processing*, pp. 1–5, Dec. 2011.

[24] Y-P.Lin and P.P.Vaidyanathan, "Linear phase cosine modulated maximally decimated filter banks with perfect reconstruction," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2525–2539, Nov. 1995.

[25] S. C. S. Dhabal and P. Venkateswaran, "A novel low complexity multichannel cosine modulated filter bank using ifir technique for nearly perfect reconstruction," *International Conference on Recent Advances in Information Technology (RAIT)*, pp. 208–213, Mar. 2012.

[26] T.A.Ramstad and J.P.Tanem, "Cosine-modulated analysis-synthesis filterbank with critical sampling and perfect reconstruction," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 1789–1792, Apr. 1991.

[27] H.Bolcskei and F.Hlawatsch, "Oversampled cosine modulated filter banks with perfect reconstruction," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, pp. 1057–1071, Aug. 1998.

[28] N.Rayavarapu and N.R.Prakash, "An efficient implementation of oversampled cosine modulated transmultiplexers," *International Journal of Advanced Engineering and Application*, pp. 310–315, Jan. 2010.

[29] L. B. P.S.R.Diniz and S.L.Netto, "Design of high-resolution cosine-modulated transmultiplexers with sharp transition band," *IEEE Transactions on Signal Processing*, vol. 52, pp. 1278–1288, May 2004.

[30] A. J.Alhava and M.Renfors, "Efficient implementation of complex exponentially-modulated filter banks," *International Symposium on Circuits and Systems*, vol. 4, pp. 157–160, May 2003.

[31] J. A.Viholainen and M.Renfors, "Efficient implementation of twice oversampled exponentially-modulated filter banks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, pp. 1138–1142, Oct. 2006.

[32] J. A.Viholainen and M.Renfors, "Efficient implementation of complex exponentially-modulated filter banks using cosine and sine modulated filter banks," *EURASIP Journal on Applied Signal Processing*, pp. 1–10, 2006.

[33] S.Mirabbasi and K.Martin, "Overlapped complex-modulated transmultiplexer filters with simplified design and superior stopbands," *IEEE Transactions on Circuits and Systems II:Analog and Digital Signal Processing*, vol. 50, pp. 456–469, Aug. 2003.

[34] Z. Y. H. Liu and Z. Cao, "Design perfect reconstruction cosine-modulated filter bank by variable neighbourhood search-least-meansquare error," *IET Signal Processing*, vol. 6, pp. 273–280, May 2012.

[35] H. J. A. Eghbali and P. Lowenborg, "Reconfigurable nonuniform transmultiplexers using uniform modulated filter banks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, pp. 539–547, Mar. 2011.

[36] S. Pagadarai, "Sidelobe Suppression for OFDM based Cognitive Radios in Dynamic Spectrum Access Networks," Master's thesis, University of Kansas, 2007.

[37] Z. Yuan, "Sidelobe Suppression and Agile Transmission Techniques for Multicarrier-based Cognitive Radio Systems," Master's thesis, Worcester Polytechnic Institute, 2009.

[38] S. Noreen and N.Z.Azeemi, "A technique for out-of-band radiation reduction in ofdm-based cognitive radio," *IEEE 17th International Conference on Telecommunications (ICT)*, pp. 853–856, Apr. 2010.

[39] A. M.S.El-Saadany and M.Abdallah, "Revisiting active cancellation carriers for shaping the spectrum of ofdm-based cognitive radios," *IEEE SARNOFF Symposium*, pp. 1–5, 2009.

[40] H. S.Pagadarai, A.Kliks and A.M.Wyglinski, "Non-contiguous multicarrier waveforms in practical opportunistic wireless systems," *IET Radar, Sonar and Navigation*, vol. 5, no. 6, pp. 674–680, 2011.

[41] N. Sokhandan and S. M. Safavi, "Sidelobe suppression in ofdm based cognitive radio systems," *International Conference on Information Sciences Signal Processing and their Applications (ISSPA)*, pp. 413–417, May 2010.

[42] I. S.Brandes and M.Schnell, "Reduction of out-of-band radiation in ofdm systems by insertion of cancellation carriers," *IEEE Communications Letters*, vol. 10, no. 6, pp. 420–422, 2006.

[43] H.A.Mahmoud and H.Arslan, "Sidelobe suppression in ofdm-based spectrum sharing systems using adaptive symbol transition," *IEEE Communications Letters*, vol. 12, pp. 133–135, Feb. 2008.

[44] S.-G. Huang and C.-H. Hwang, "Improvement of active interference cancellation: Avoidance technique for ofdm cognitive radio," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 5928–5937, Dec. 2009.

[45] Z. W. Daiming Qu and T. Jiang, "Extended active interference cancellation for sidelobe suppression in cognitive radio ofdm systems with cyclic prefix," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 1689–1695, May 2010.

[46] I. M. Paul Sutton, Baris Ozgul and L. Doyle, "Ofdm pulse-shaped waveforms for dynamic spectrum access networks," *IEEE Symposium on New Frontiers in Dynamic Spectrum*, pp. 1–2, Apr. 2010.

[47] H. B. Pawel Kryszkiewicz and A. M.Wyglinski, "Protection of primary users in dynamically varying radio environment: practical solutions and challenges," *EURASIP Journal on Wireless Communications and Networking*, Jan. 2012.

[48] J. W. Pei Gao and S. Li, “Non-contiguous ci/ofdm: A new data transmission scheme for cognitive radio context,” *IEEE International Conference on Wireless Information Technology and Systems (ICWITS)*, pp. 1–4, 2010.

[49] D. G.Manolakis and V. K.Ingle, *Applied Digital Signal Processing.* Cambridge University Press, 2011.

[50] J. G.Proakis and D. G.Manolakis, *Digital Signal Processing.* Prentice-Hall Inc, 1996.

[51] P.P.Vaidyanathan, *Multirate Systems and Filter Banks.* Pearson Education, Inc, 1993.

[52] A.V.Oppenheim and R.W.Schafer, *Digital Signal Processing.* Prentice-Hall Inc, 1975.

[53] D. Choudhary and S. B.Jain, *Linear Integrated Circuits*, ch. 7. New Age International Limited, Publishers, 2003.

[54] P. B. A.Bruce Carlson and J. C.Rutledge, *Communication Systems*, ch. 3. McGraw-Hill, 2002.

[55] V. K.Madisetti and D.B.Williams, *The Digital Signal Processing Handbook: DSP Fundamentals.* CRC Press, 2010.

[56] R. Kuc, *Introduction to Digital Signal Processing.* McGraw-Hill, 1988.

[57] R. P.Ramachandran, *Bandwidth Efficient Filter Banks for Transmultiplexers.* PhD thesis, McGill University, 1990.

[58] R.P.Ramachandran and P.Kabal, “Transmultiplexers: Perfect reconstruction and compensation of channel distortion,” *Signal Processing*, vol. 21, pp. 261–274, 1990.

[59] R. Cox, “The design of uniformly and nonuniformly spaced pseudoquadrature mirror filters,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, pp. 1090–1096, Oct. 1986.

[60] J.Masson and Z.Picel, “Flexible design of computationaly efficient nearly perfect qmf filter banks,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 10, pp. 541–544, Apr. 1985.

[61] R.D.Koilpillai and P.P.Vaidyanathan, “Cosine-modulated fir filter banks satisfying perfect reconstruction,” *IEEE Transactions on Signal Processing*, vol. 40, pp. 770–783, Apr. 1992.

[62] T.Karp and N.J.Fliege, “Modified dft filter banks with perfect reconstruction,” *IEEE Transactions on Circuits and Systems II:Analog and Digital Signal Processing*, vol. 46, pp. 1404–1414, Nov. 1999.

[63] T. P.N.Heller and T.Q.Nguyen, “A general formulation of modulated filter banks,” *IEEE Transactions on Signal Processing*, vol. 47, pp. 986–1002, Apr. 1999.

[64] T.Karp and N.J.Fliege, “Mdft filter banks with perfect reconstruction,” *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 744–747, May 1995.

[65] Y-P.Lin and P.P.Vaidyanathan, “Application of dft filter banks and cosine modulated filter banks in filtering,” *IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 254–259, Dec. 1994.

[66] R. V. R.Zhou, Q.Han and Z.Wu, “A software defined radio based adaptive interference avoidance tdcs cognitive radio,” *IEEE International Conference on Communications (ICC)*, pp. 1–5, May 2010.

[67] R. Electronics. `http://www.radio-electronics.com/analysis/receivers/2010-03/joe-joseph-mitola-cognitive-radio-future.php`.

[68] M. N. Alexander M.Wyglinski and Y. Hou, *Cognitive Radio Communications and Networks*, ch. 1. Elsevier, Inc, 2012.

[69] FCC. `http://www.cs.ucdavis.edu/~liu/289I/Material/FCC-03-322A1.pdf`.

[70] FCC. http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-01-264A1.doc.

[71] W. H.W.Tuttlebee, "Software defined radio: Facets of a developing technology," *IEEE Personal Communications*, vol. 6, pp. 38–44, Apr. 1999.

[72] L.S.Nagurney, "Software defined radio in the electrical and computer engineering curriculum," *39th IEEE Frontiers in Education Conference*, pp. 1–6, Oct. 2009.

[73] FCC. http://www.fcc.gov/help/public-safety-tech-topic-4-software-defined-radio.

[74] P.Hoeher and H.Lang, "Coded-8psk modem for fixed and mobile satellite services based on dsp," *ESA WPP-006*, pp. 117–123, Jan. 1990.

[75] L.T.Ong, "An usrp-based interference canceller," *IEEE International Conference on Communication Systems (ICCS)*, pp. 95–99, Nov. 2012.

[76] C. M.Schwall, S.Nagel and F.K.Jondral, "Model-based waveform design for the universal software radio peripheral with simulink," *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1–4, Aug. 2011.

[77] S. A.A.Tabassam, F.A.Ali and M.U.Suleman, "Building software-defined radios in matlab simulink - a step towards cognitive radios," *UKSim International Conference on Computer Modelling and Simulation (UKSim)*, pp. 492–497, 2011.

[78] E. R. LLC. http://www.ettus.com/.

[79] M. Z.Tong and C.F.Liau, "Wireless transmission using universal software radio peripheral," *International Conference on Space Science and Communication (IconSpace)*, pp. 19–23, Oct. 2009.

[80] D. J. C.Zhang, P.Xie and R.Yu, "Wireless home gateway: Software-defined radio architecture and applications," *IET International Conference on Communication Technology and Application (ICCTA)*, pp. 346–350, Oct. 2011.

[81] V. K.Ingle and J. G.Proakis, *Digital Signal Processing using MATLAB V.4.* Global Engineering, 1997.

[82] MathWorks. `http://www.mathworks.com/help/matlab/learn_matlab/product-description.html`.

[83] MathWorks. `http://www.mathworks.com/help/matlab/index.html`.

[84] A.Azemi and E.E.Yaz, "Utilizing simulink and matlab in a graduate nonlinear systems analysis course," *26th Annual Conference Proceedings of Frontiers in Education Conference*, vol. 2, pp. 595–598, 1996.

[85] MathWorks. `http://www.mathworks.com/products/datasheets/pdf/simulink.pdf`.

[86] G. M.G.Bellanger and M.Coudreuse, "Digital filtering by polyphase network: Application to sample-rate alteration and filter banks," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 2, pp. 109–114, 1976.

[87] D. B.Turek, "Design of Efficient Digital Interpolation Filters for Integer Upsampling," Master's thesis, Massachusetts Institute of Technology, 2004.

[88] A. Sail, "Hardware Implementation of Filtering Based Sidelobe Suppression for Spectrally Agile Multicarrier based Cognitive Radio Systems," Master's thesis, Worcester Polytechnic Institute, 2013.

[89] M. Rice, *Digital Communications: A Discrete-Time Approach.* Pearson Education, Inc, 2009.