Group Delay Reduction Methods in FIR Digital Filters

Hala Haseeb Al-Ani

Submitted to the
Institute of Graduate Studies and Research
In partial fulfillment of the requirements for the Degree of

`

Master of Science
in
Computer Engineering

Eastern Mediterranean University
October 2013
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Hasan Kömürcügil
Supervisor

Examining Committee

1. Prof. Dr. Hakan Altınçay

2. Prof. Dr. Hasan Kömürcügil

3. Assoc. Prof. Dr. Ekrem Varoğlu

ABSTRACT

Linear-phase finite impulse response (FIR) filters are widely used in digital signal processing applications because of their various advantages. These advantages include the absence of phase distortion, unrestricted stability, and lower filter-coefficient sensitivity. The most important shortcoming of linear-phase FIR filter is that the overall group delay is (N-1)/2 where N is the length of the filter. This quantity becomes large for higher order filters in communication applications.

Many algorithms have been proposed to reduce this delay and its distortion. Typically, block convolution techniques such as overlap-add method (OAM) and overlap-save method (OSM) are used for a long input sequence. Yet, with respect to input, by using these methods, the output sequence has a finite group delay.

In this thesis, the performance of enhanced modified overlap and save method is investigated. First, the impulse response is made causal and then it is shifted left (circular) by an amount of $(N-1)/2$ for N odd and $N/2$ for N even. Finally, the samples to be excluded from the final convolution are defined. It is expected that this results in a reduction in the causal delay and also in the group delay. Simulations are carried out by MATLAB. The performance of the method is compared with the results obtained from the OSM based filter.

Keywords: Impulse Response, FIR Filter, Linear Phase FIR Filter, Group Delay.

ÖZ

Doğrusal fazlı sonlu dürtü cevaplı (FIR) süzgeçler çok çeşitli avantajlarından dolayı, sayısal sinyal işleme uygulamalarında yaygın olarak kullanılmaktadırlar. Bu avantajlar, faz bozunumu içermeyen, sınırsız kararlılık ve süzgeç katsayılarına olan az duyarlılık olarak sıralanabilir. Ancak, doğrusal fazlı FIR süzgeçlerinin en önemli dezavantajı ise toplam grup gecikmesinin (N-1)/2 olarak ortaya çıkmasıdır. Buradaki N, süzgeçteki katsayı sayısını temsil etmektedir. Haberleşme uygulamalarında, toplam gecikmenin miktarı süzgeç katsayılarının sayıları ile doğru orantılı olarak artmaktadır.

Bu gecikmeyi ve ondan dolayı oluşan bozunumu azaltmak için birçok algoritma önerilmiştir. Tipik olarak, uzun giriş dizileri için blok konvolüsyon olarak bilinen üstüste binik toplama metodu (OAM) ve üstüste binik saklama metodu (OSM) kullanılır. Bu metodlar kullanıldığında, çıkış dizisinin girişe göre sonlu bir grup gecikmesi vardır.

Bu tezde, iyileştirilip modifiye edilmiş üstüste binik saklama metodunun performansı incelemiştir. İlk olarak, süzgeçin dürtü cevabı nedensel yapılmıştır ve bu dürtü cevabı daha sonra, N tek olduğu zaman (N-1)/2 kadar sola, çift olduğu zaman ise N/2 kadar yine sola kaydırılmıştır.

Son olarak, son konvolüsyon sonucundan dışlanacak olan örnekler tanımlanmıştır. Bunun, nedensel gecikme ve grup gecikmesinin azalmasına neden olacağı beklenmektedir. Benzetim çalışmaları MATLAB ortamında yapılarak, bu metodun performansı, OSM metodundan elde edilen sonuçlarla karşılaştırılmıştır.

Anahtar kelimeler: Dürtü Cevabı, FIR Süzgeç, Doğrusal Fazlı FIR Süzgeç, Grup Gecikmesi

I dedicate this thesis to my husband, mother, father, two sisters, uncles, aunts, cousins and to all my friends.

ACKNOWLEDGMENT

I sincerely acknowledge all the help and support that I was given by Prof. Dr. Hasan Kömürcügil whose knowledge, guidance, and effort made this research go on and see the light.

I would like to express my deep gratitude to my mother and father for the support, effort, pain, and patience and to whom I own the success of my life. Special thanks go to my husband Ayman and my uncle Nashwan and aunt Ban and my friends Liwaa Hussein, Omar Hayman, Humam Mohammed, Ahmed Hani, Hayder Mohammed, Anas Qasim, Mohammed AL-Sayed, Bessam Al-Jewad, Sinan Hazem and to all my friends for their help and support.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

Chapter 1

INTRODUCTION

Finite impulse response (FIR) filter is defined as the digital filter that performs mathematical operations on a piece of discrete-time signal in order to change some of its characteristics in a desired manner. Linear-phase FIR filters are preferred in digital signal processing applications because of their various advantages. These advantages include unconditional stability, absence of any phase distortion and lower filter-coefficient sensitivity.

The linear-phase FIR filter also has a main disadvantage which is the overall group delay $(N - 1)/2$ in specific applications, where N represents the filter length. It is obvious that this amount becomes larger when high order filters are considered. Also, such a large amount of group delay leads to untolerable echoes of the transmitted signals in communication applications. There are a lot of important applications of linear phase filters, in which the high group delay caused by linear phase is important (for instance in electrocardiography where the delay could modify the QRS complex location [1], in two-way speech communication systems calling for a low round-trip delay).

On the other hand, large delay in discrete-time control applications is also unacceptable. The group delay slows down the speed of processor. A multirate digital signal processing approach has been suggested to minimize this delay in active noise control systems [2].

A recent category of maximally nonsymmetrical flat FIR lowpass filter has been improved as in [3] to improve the performance of the filter design. In comparative manner, this improvement gives a constant of group delay, unlike the symmetric filter with no collapse of frequency response magnitude. A robust non-iterative algorithm has been proposed in [4] to design optimal minimum-phase digital FIR filters with arbitrary magnitude responses based on discrete Hilbert transform (DHT). When the DHT is extended to the complex case, the minimum-phase filters require less memory and less arithmetic calculations than linear-phase filters for satisfying the same constraints on delay and magnitude response. Here, by that algorithm, the magnitude spectrum of the truncated minimum-phase sequence is different from the actual magnitude spectrum. As a result, it is interesting to calculate the minimum delayed outcome response without making any change in the impulse response (IR) of the filter. It looks like unattainable according to the most common filtering algorithms, for instance overlap add method (OAM), and overlap save method (OSM) mentioned in [5] and this technique is utilized by the following convolution equation:

$$y(n) = \sum_{k=0}^{N} h(k)x(n-k) \tag{1-1}$$

Here x(n) represents the input signal which will be filtered, h(n) represents the filter impulse response, y(n) represents filtered signal, and N is the filter length.

The design of low delay FIR bandpass filters with maximally flat passband and equiripple stopband by doing successive projections approach was presented in [6]. It is well known that linear-phase FIR filter delay increases by increasing the filter length. A weighted least squares (WLS) technique has been proposed to design a near-equiripple FIR filter having variable fractional delay [7]. On the other hand, the design of arbitrary variable fractional-delay FIR filters has been achieved which is based on the complex version of WLS [8].

Apaydin showed a new technique for reducing the delay in FIR digital filters with equiripple passband, and peak constrained least squares stop-bands for real-time applications [9]. In this technique, the reduction of the group delay that can be reached is between 12-22 % in passband compared with the present techniques. Kene [10] proposed doing some adjustments to overlap-save method (OSM) and overlap-add method (OAM) algorithms to be used for filtering real signals by an N tap FIR filter. This technique requires two complex 2N-point discrete Fourier transform (DFT) in order to get two blocks of N samples, unlike the actual method which only needs four real 2N-point DFT to generate the exact data quantities. In this method, when it is compared with the classical algorithms, the time needed for processing the delay increases by N samples. Although the polyphase decomposition method reduces the delay considerably, it increases the computational complexity. A recent filtering algorithm has been designed using the DFT based circular convolution and the OSM method of block convolution when the data sequences are long and need to be filtered [11]. Zero group delay has been obtained by zero delay modified overlap-save method (ZDMOSM). Although this method achieves a zero group delay, it needs some changes in the acquisition duration in the process period, which is not likely preferred. Furthermore, J., S. Fouda [11] proposed a reduced delay modified overlap-save method (RDMOSM), in which the group delay reduction was achieved by a factor of 0.5 compared to OSM.

In this thesis, the RDMOSM technique has been enhanced by redefining the samples to be excluded from the final circular convolution result. According to the simulation results, the proposed technique is very close to that of the OSM with little changes in the delay. Subsequently, it can be compared to linear convolution. This thesis is organized as follows: Chapter 2 deals with the FIR filter structures and their types. In Chapter 3, the reduction of group delay will be explained with delay reduction methods such as RDMOSM and ZDMOSM and the proposed enhanced RDMOSM (ERDMOSM) technique will be discussed. In Chapter 4, results and discussions will be discussed. Finally, Chapter 5 gives conclusions of the work.

Chapter 2

FIR FILTERS

2.1 Structures of FIR Systems

A polynomial system function in z^{-1} for a basic FIR filter is:

$$\mathrm{H(z)}=\sum_{n=0}^{\mathrm{N}} h(n)z^{-n} \tag{2-1}$$

the transfer function of the FIR filter is $H(z)$, the impulse response is h(n), a delay of one sample time denoted by z^{-1}, N represents the filter length (number of coefficients) and n represents discrete time. For an input $x\ (n)$, the output is as follows:

$$\mathrm{y(n)}=\sum_{k=0}^{\mathrm{N}} h(k)x(n-k) \tag{2-2}$$

Equation (2-2) is identified as the convolution sum equation. Computation of this sum requires N additions and $(N\ +\ 1)$ multiplications for every n value.

2.1.1 Direct Form

The realization of equation (2-2) by using a tapped delay line method is shown in Fig. 1.

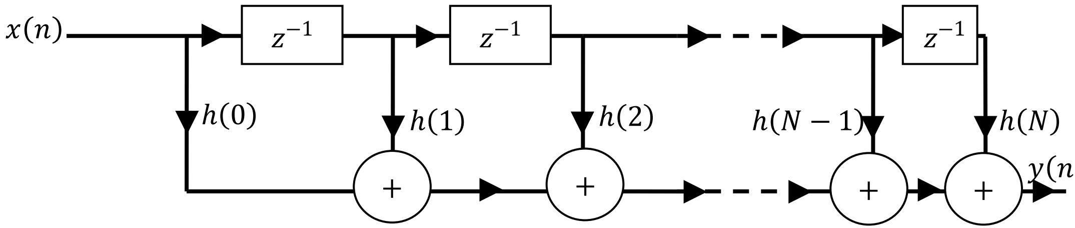


Figure 1: Direct Form Structure

Each output sample $\mathrm{y(n)}$ needs N additions, $\mathrm{N} + 1$ multiplications, and N delays. Otherwise, in the case of any similarity in the unit sample response, it is possible to reduce the multiplications number.

2.1.2 Cascade Form

For a basic FIR filter, the transfer function could be factored into first-order factors,

$$\mathrm{H(z)}=\sum_{n=0}^{\mathrm{N}} h(n)z^{-n} = A\prod_{k=1}^{N}(1-a_{\kappa}\, z^{-1}) \tag{2-3}$$

where a_k for $k = 1\ldots\ldots N$ are the zeros of $H\,(z)$. The complex roots of $H(z)$ happen in complex conjugated pairs if $h(n)$ is real and these conjugated pairs can be combined to form second-order factors with real coefficients,

$$\mathrm{H(z)}=\mathrm{A}\prod_{k=1}^{N_s}[1+b_k(1)\,z^{-1}+b_k(2)z^{-2}] \tag{2-4}$$

the structure of equation (2-4) is illustrated in Figure 2.

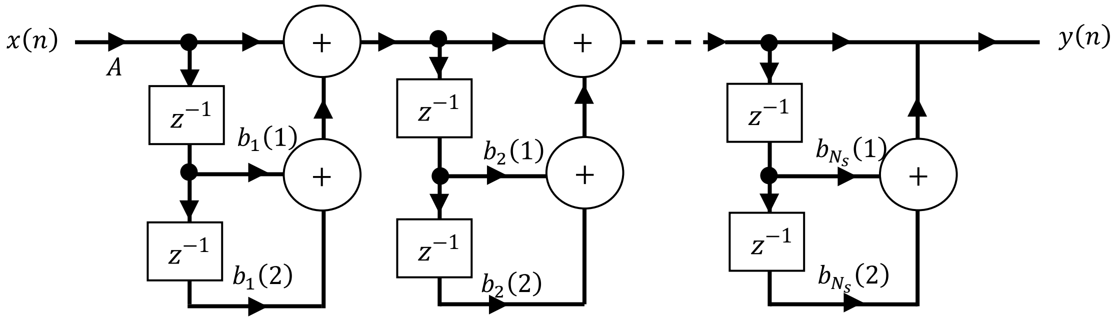


Figure 2: An FIR Filter Implemented as a Cascade of Second-order Systems

2.1.3 Linear Phase Filters

Impulse responses of linear phase filters are either symmetric or anti-symmetric

$$h(n) = h(N-n) \tag{2-5}$$

$$h(n) = -h(N-n) \tag{2-6}$$

respectively. This symmetry could be exploited to make the network structure simpler. For instance, if $h(n)$ is symmetric, and N is even (type I filter),

$$y(n) = \sum_{k=0}^{N} h(k)x(n-k) = \sum_{k=0}^{\frac{N}{2}-1} h(k)[x(n-k) + x(n-N+k)] + h\left(\frac{N}{2}\right)x\left(n-\frac{N}{2}\right) \qquad (2-7)$$

Consequently, making the sums $[x(n-k) + x(n-N+k)]$ before multiplying by $h(k)$ reduces the multiplications number. The out coming structure is in Fig. 3 (a). While, if N is odd and $h(n)$ symmetric (type II filter), the resulting structure is shown in Fig. 3(b). There are similar anti-symmetric structures (type III and IV) linear phase filters.

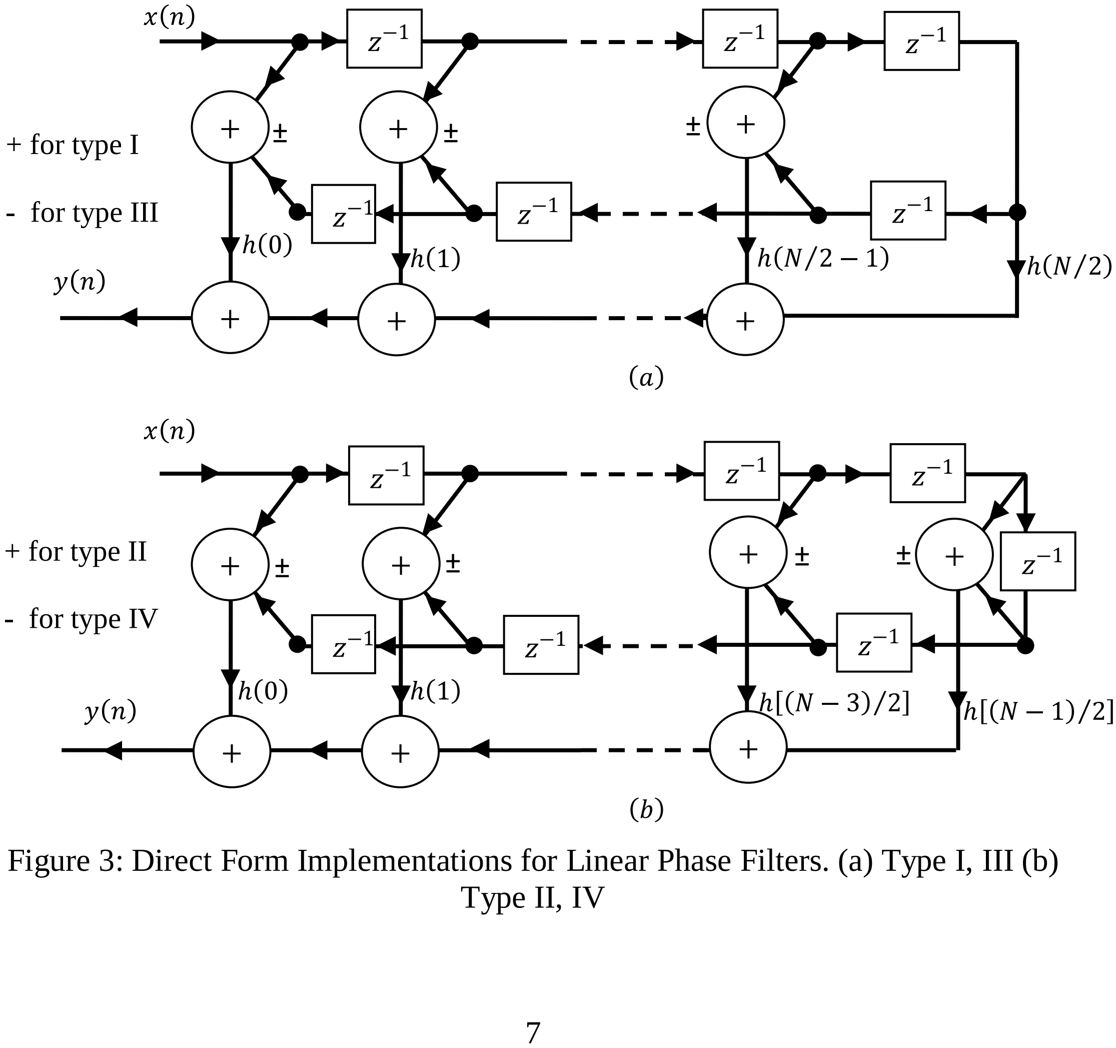


Figure 3: Direct Form Implementations for Linear Phase Filters. (a) Type I, III (b) Type II, IV

2.1.4 Frequency Sampling

A filter is parameterized after the implementation of frequency sampling structure in terms of its discrete Fourier transform (DFT) coefficients. Particularly, if $H(k)$ is the $\mathrm{N}-$point DFT of an FIR filter with $h(n) = 0$ for $0 < n > N$, then the unit sample response of the filter written as:

$$h(n) = \frac{1}{N}\sum_{k=0}^{N} H(k)e^{j2\pi nk/N} \tag{2-8}$$

The transfer function can be written as

$$\left.\begin{aligned} H(z) &= \sum_{n=0}^{N-1} h(n)z^{-n} = \sum_{n=0}^{N-1}\left[\frac{1}{N}\sum_{k=0}^{N-1} H(k)e^{j2\pi nk/N}\right]z^{-n} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} H(k)\sum_{n=0}^{N-1} e^{j2\pi nk/N}z^{-n} \end{aligned}\right\} \tag{2-9}$$

Computing the sum over n gives

$$H(z) = \frac{1}{N}(1 - z^{-N})\sum_{k=0}^{N-1}\frac{H(k)}{1 - e^{j2\pi k/N}z^{-1}} \tag{2-10}$$

which corresponds to an FIR filter cascade $\frac{1}{\mathrm{N}}(1 - \mathrm{z}^{-\mathrm{N}})$ with one-pole parallel network filters:

$$H(z) = \frac{H(k)}{1 - e^{j2\pi k/N}z^{-1}} \tag{2-11}$$

For a filter in narrowband that has the majority of its DFT coefficients equal to zero, the structure of the frequency sampling shall be an efficient implementation. The structure of the frequency sampling is given in Figure 4. If $h(n)$ is real, $H(k) = H^{*}(N - k)$, the structure could be simplified . For instance of $H(k)$, if N is even [12]

$$H(z)=\frac{1}{N}(1-z^{-N})\left[\frac{H(0)}{1-z^{-1}}+\frac{H(N/2)}{1+z^{-1}}+\sum_{k=1}^{N/2-1}\frac{A(k)-B(k)z^{-1}}{1-2\cos(2\pi k/N)z^{-1}+z^{-2}}\right] \quad (2\text{-}12)$$

where

$$A(k)=H(k)+H(N-k) \quad (2\text{-}13)$$

$$B(k)=H(k)e^{-j2\pi k/N}+H(N-k)e^{j2\pi k/N} \quad (2\text{-}14)$$

On the other hand, when N is odd, same definition results can be obtained.

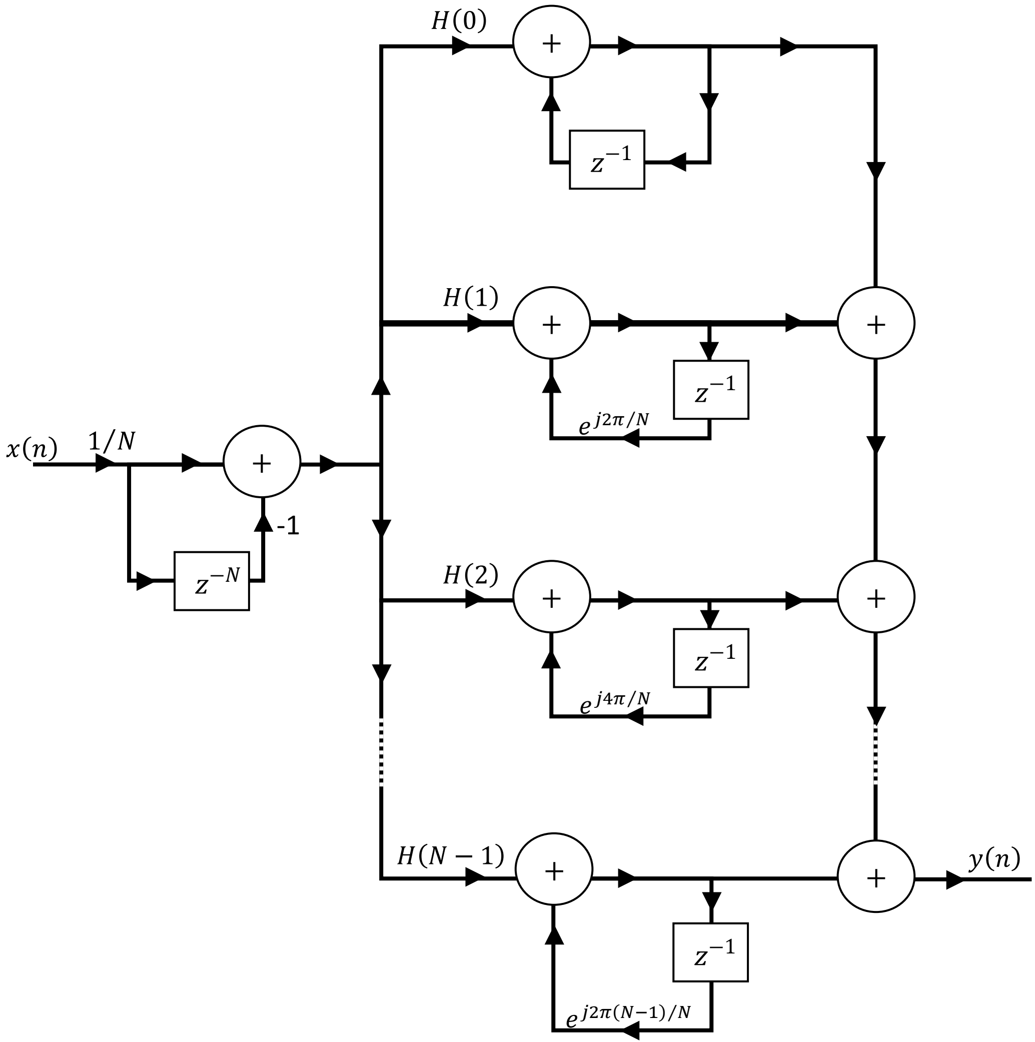


Figure 4: Frequency Sampling Filter Structure

2.2 Linear Phase Response

Linear phase response is one of the most important properties of FIR filters. When an input signal is applied to filter, it appears at the output of the filter with modifications done in amplitude and/or phase. The extent of this modification depends on the amplitude and phase characteristics of the filter. The phase delay or group delay of the filter gives important information about how the filter makes this modification in the phase of the signal. The phase delay of the filter is defined as the amount of time delay each frequency component of the input signal suffers in going through the filter. On the other hand, the group delay is the average time delay the composite signal suffers at each frequency. Mathematically, the phase delay can be defined as the negative of the phase angle divided by frequency and the group delay is defined as the negative of the derivative of the phase with respect to frequency.

If a filter has nonlinear phase characteristics, it causes a phase distortion in the signal passing through it. Such a distortion is undesired in many applications such as music, data transmission, video, and biomedicine. Therefore, the filters having linear phase characteristics are widely used in these applications. A linear shift-invariant system has a linear phase response if it is written in the following form

$$H\left(e^{j\omega}\right) = \left|H(e^{j\omega})\right|e^{-j\alpha\omega} \tag{2-15}$$

where α can be a real number which defines the group delay,

$$\tau_h(\omega) = \alpha \tag{2-16}$$

A system has a generalized linear phase when the frequency response gets the form

$$H\left(e^{j\omega}\right) = A\left(e^{j\omega}\right)e^{-j(\alpha\omega-\beta)} \tag{2-17}$$

where α and β are constants. Now, consider the FIR system with an impulse response

$$h(n) = \begin{cases} 1 & n = 0,1, \dots N \\ 0 & else \end{cases} \qquad (2\text{-}18)$$

The frequency response is

$$H(e^{j\omega}) = e^{-jN\omega/2} \frac{\sin(\frac{N+1}{2}\omega)}{\sin(\frac{\omega}{2})} \qquad (2\text{-}19)$$

Thus, this system has generalized linear phase, with α = N / 2 and β = 0.

For a causal system with a rational transfer function to own linear phase, the impulse response have to be finite in length. An FIR filter that has a real-valued impulse response of length N + 1, has generalized linear phase if its impulse response is symmetric,

$$h(n) = h(N - n) \qquad (2\text{-}20)$$

In this case, α = N / 2 and β = 0 or π. Another sufficient condition is that h(n) be antisymmetric [12]

$$h(n) = -h(N - n) \qquad (2\text{-}21)$$

which corresponds to case through which α = N/2 and β = π/2 or 3π/2.

2.3 Types of Linear Phase FIR Filters

Let us consider the special types of FIR filters where the coefficients $h(n)$ of the transfer function

$$H(z^{-1}) = \sum_{n=0}^{N} h(n) z^{-n} \qquad (2\text{-}22)$$

are supposed to be symmetric or anti-symmetric. Since the order of the polynomial in both of these two types can be either odd or even, there are four types of filters with diverse properties, which will be explained next [13].

Type I. Coefficients are symmetric [$h(n) = h(N - n)$], and the order N is even.

In general, coefficients can be expressed in some other forms. Let us assume that the order is even. The transfer function in equation (2-22) can be expanded as:

$$H(z^{-1}) = \sum_{n=0}^{N} h(n)z^{-n}$$

$$= h(0) + h(1)z^{-1} + h(2)z^{-2} + \cdots + h(N-1)z^{-N+1} + h(N)z^{-N} \tag{2-23}$$

For type I filter with N order, as shown in Fig. 5, it is noted that $h(0) = h(N)$, $h(1) = h(N-1), \dots, h(n) = h(N-n)$. Applying these relationships in the equation above, we get

$$H(z^{-1}) = h(0)[1 + z^{-N}] + h(1)[z^{-1} + z^{-N+1}] + \cdots + h(\frac{N}{2})z^{\frac{-N}{2}} \tag{2-24}$$

This can also be shown as in the following form

$$H(z^{-1}) = z^{\frac{-N}{2}}\left\{h(0)\left[z^{\frac{N}{2}} + z^{\frac{-N}{2}}\right] + h(1)\left[z^{-1+\frac{N}{2}} + z^{-N+1+\frac{N}{2}}\right] + \cdots + h\left(\frac{N}{2}\right)\right\} \tag{2-25}$$

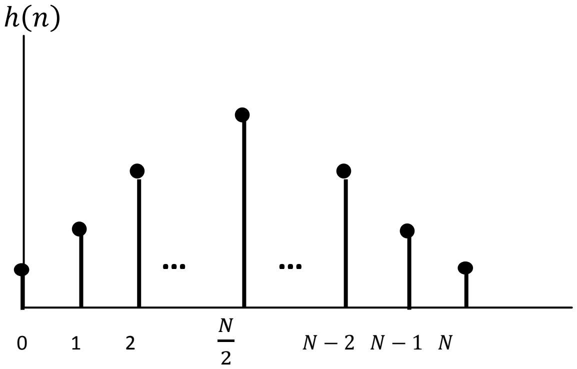


Figure 5: Unit Impulse Responses of the Type I FIR Linear Phase Filters

The frequency response of equation (2-25) is given by

$$H\left(e^{-j\omega}\right) = e^{j\theta\omega}\{H_R(\omega)\} \tag{2-26}$$

In this formula, the term $H_R(\omega)$ is a real-valued function; however it can be negative or positive at any specific frequency, therefore while transforming from a positive value to a negative one, the angle of the phase changes by π radians $(180°)$. The angle of the phase $\theta(\omega) = -3\omega$ is a linear function of ω, and the group delay τ is the same as three samples. Remember that the group delay is three samples on the normalized frequency basis, but the real the group delay is $3T$ seconds, where T denotes the sampling period.

In general, $H(e^{j\omega})$ can be expressed in some other forms

$$\left.\begin{aligned} H\left(e^{j\omega}\right) &= \sum_{n=0}^{N} h(n)e^{-jn\omega} \\ &= h(0) + h(1)e^{-j\omega} + h(2)e^{-2j\omega} + \cdots + h(N-1)e^{-j(N\omega)} \\ &= e^{-j[(N/2)\omega]}\left\{2h(0)\cos\left(\frac{N\omega}{2}\right)\right. \\ &\qquad + 2h(1)\cos\left(\left(\frac{N}{2}-1\right)\omega\right) + 2h(2)\cos\left(\left(\frac{N}{2}-2\right)\omega\right) + \cdots \\ &\qquad \left. + h\left(\frac{N}{2}\right)\right\} \end{aligned}\right\} \text{(2-27)}$$

and now in a more compact form:

$$H\left(e^{j\omega}\right) = e^{-j[(N/2)\omega]}\left\{h\left(\frac{N}{2}\right) + 2\sum_{n=1}^{N/2} h\left[\frac{N}{2}-n\right]\cos(n\omega)\right\} = e^{j\omega}\{H_R(\omega)\} \qquad \text{(2-28)}$$

The whole the group delay is constant $\tau = N/2$ in the general case, for a type I FIR filter.

Type II. Coefficients are symmetric $[h(n) = h(N - n)]$, and the order N is odd.

$$H(z^{-1}) = \sum_{n=0}^{N} h(n)z^{-n} = h(0) + h(1)z^{-1} + h(2)z^{-2} + \cdots + h(N)z^{-N} \qquad (2\text{-}29)$$

and due to symmetry

$$h(0) = h(N), h(1) = h(N-1), h(2) = \ h(N-2), \ldots, h(n) = h(N-n) \qquad (2\text{-}30)$$

Now, if we consider symmetric coefficients with N odd, the impulse response is shown in Figure 6.

The frequency response is in the type II filter for general case can be written as

$$\left.\begin{aligned} H\left(e^{-j\omega}\right) &= \sum_{n=0}^{N} h(n)e^{-jn\omega} = e^{j\theta(\omega)}\{H_R(\omega)\} \\ &= e^{-j\left(\frac{N}{2}\omega\right)} \left\{ \sum_{n=1}^{(N+1)/2} 2h\left[\frac{N+1}{2} - n\right] \cos\left(\left(n - \frac{1}{2}\right)\omega\right) \right\} \end{aligned}\right\} \qquad (2\text{-}31)$$

which demonstrates a linear phase $\theta(\omega) \ = \ -[(N/2)\omega]$ and a constant group delay $\tau = \ N/2$ samples.

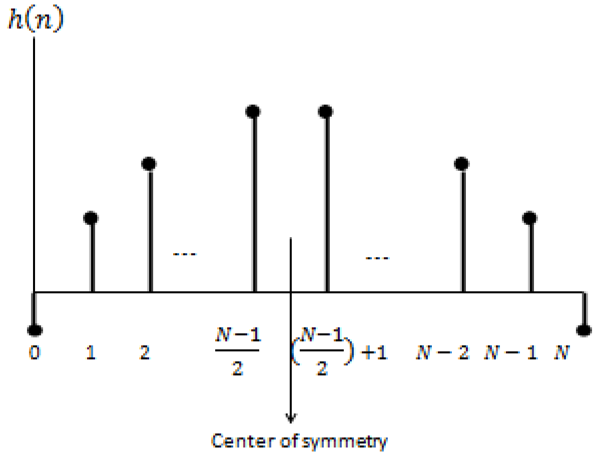


Figure 6: Unit Impulse Responses of the Type II FIR of Linear Phase FIR Filters

Type III. The coefficients are anti-symmetric $[h(n) = -h(N-n)]$, and the order N is even. Figure 7 shows that $h(0) = -h(N), h(1) = -h(N-1), h(2) = -h(N-2)$ and $h(N/2)$ = 0 to preserve anti-symmetry for these samples:

$$H(z^{-1}) = h(0)[1 - z^{-N}] + h(1)[z^{-1} - z^{-N+1}] + \cdots + h(\frac{N}{2})z^{\frac{-N}{2}} \tag{2-32}$$

This can also be shown as in the following form

$$H(z^{-1}) = z^{\frac{-N}{2}}\left\{h(0)\left[z^{\frac{N}{2}} - z^{\frac{-N}{2}}\right] + h(1)\left[z^{-1+\frac{N}{2}} - z^{-N+1+\frac{N}{2}}\right] + \cdots + h\left(\frac{N}{2}\right)\right\} \tag{2-33}$$

Here if we place $z = e^{j\omega}$, and $e^{j\omega}$ – $e^{-j\omega} = 2j\, sin(\omega) = 2e^{j(\pi/2)}\, sin(\omega)$, we get the frequency response in the general case as

$$H\left(e^{-j\omega}\right) = e^{-j[(N\omega-\pi)/2]}\left\{2\sum_{n=1}^{N/2} h\left[\frac{N}{2} - n\right]\sin(n\omega)\right\} \tag{2-34}$$

and it has a linear phase $\theta(\omega) = -[(N\omega - \pi)/2]$ and the group delay τ = *N/*2 samples.

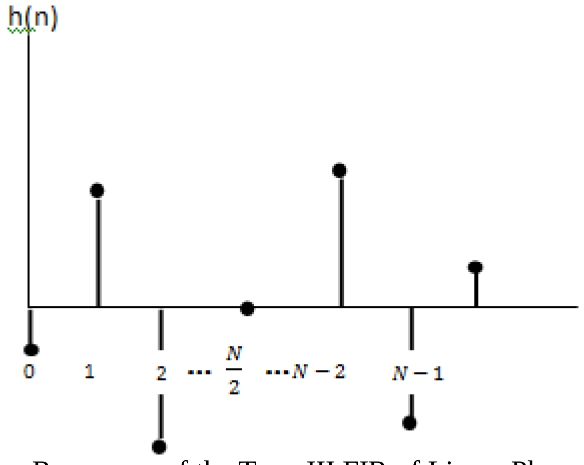


Figure 7: Unit Impulse Responses of the Type III FIR of Linear Phase FIR Filters

Type IV. Coefficients are anti-symmetric $[h(n) \; = \; -h(N \; - \; n)]$, and the order N is odd. As in Figure 8, in which $h(0) = -h(N),\; h(1) = -h(N-1), h(2) = -h(N-2), \dots,\; h(n) = h(N-n)$. Its transfer function can be written as

$$H(z^{-1}) = h(0)[1 - z^{-N}] + h(1)[z^{-1} - z^{-N+1}] + \cdots + h(\frac{N-1}{2})z^{\frac{-N-1}{2}} \tag{2-35}$$

The frequency response of the transfer function of the type IV linear phase filter is usually given by

$$H\left(e^{-j\omega}\right) = e^{-j[(N\omega-\pi)/2]}\left\{2\sum_{n=1}^{(N+1)/2} h\left[\frac{N+1}{2} - n\right]\sin\left(\left(n-\frac{1}{2}\right)\omega\right)\right\} \tag{2-36}$$

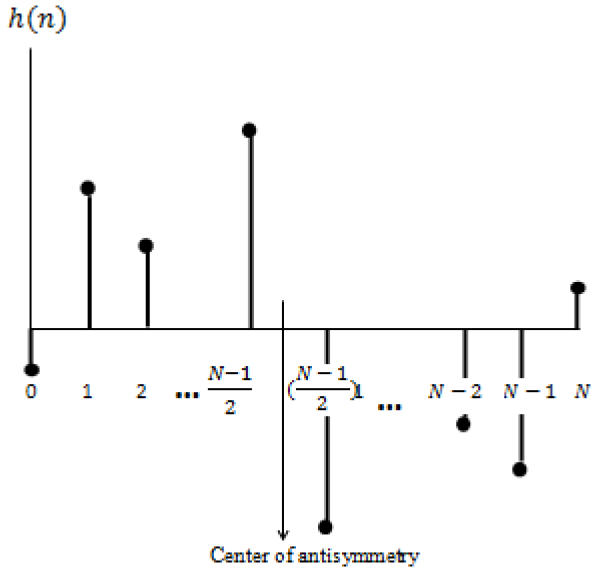


Figure 8: Unit Impulse Responses of the Type IV FIR of Linear Phase FIR Filters

2.4 Properties of Linear Phase FIR Filters

The discussion of the types of FIR filters revealed that FIR filters with symmetric or anti-symmetric coefficients offer equivalently constant group delay or (linear phase); these coefficients represent the impulse response samples. It was noticed before that an FIR filter with symmetric or anti-symmetric coefficients has a linear phase and as a result a constant group delay. Theoretically, [14] confirmed that an FIR filter with a constant group delay is required to have symmetric or anti-symmetric coefficients. These properties are practical in designing a FIR filters and their applications. The magnitude response of standard FIR filters with linear phase have been calculated to observe some extra properties of these four filter types [13] as shown in Figure 9.

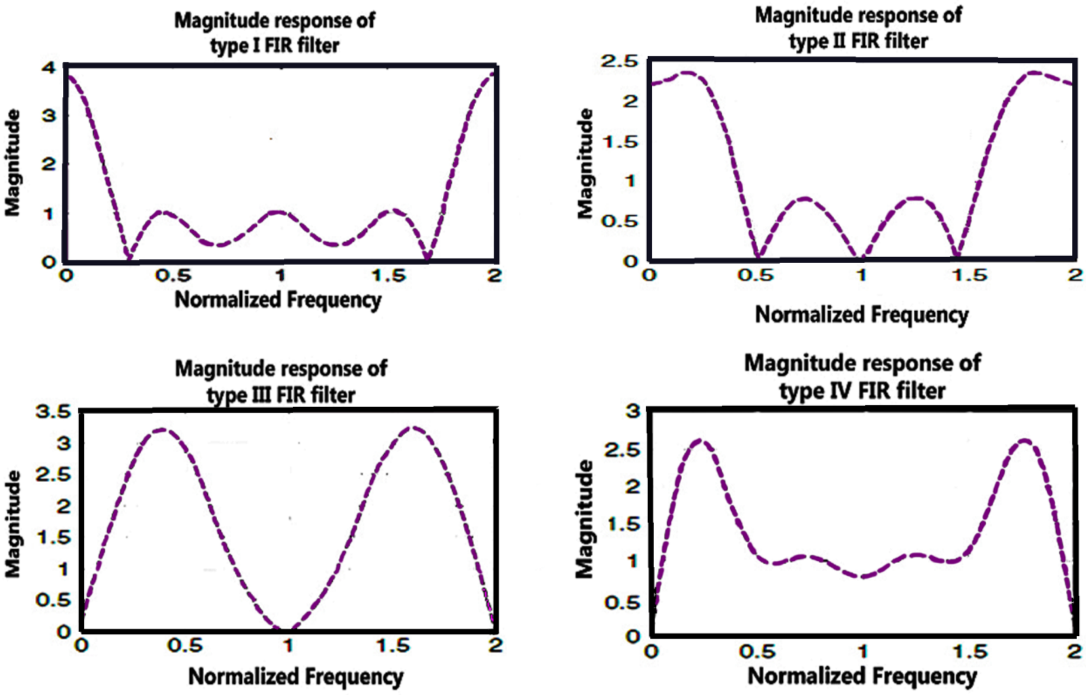


Figure 9: Magnitude Responses of the Four Types of Linear Phases FIR Filter

The following explanations about these standard magnitude responses will be useful in creating suitable alternatives at the beginning of their design. For example, type I filters have a non-zero magnitude at ω = 0 as well as a non-zero value at the normalized frequency $\omega/\pi = 1$ (corresponding to the Nyquist frequency), while type II filters have non-zero magnitude at $\omega = 0$ however a zero value at the Nyquist frequency. Therefore, these filters are clearly not appropriate for the design of bandpass and highpass filters, while both types are appropriate for lowpass filters.

Type III filters have zero magnitude at $\omega = 0$ as well as at $\omega/\pi = 1$, thus they are appropriate for the design of bandpass filters nevertheless it is not appropriate for lowpass and bandstop filters. Whereas, type IV filters have zero magnitude at $\omega = 0$ and a non-zero at $\omega/\pi = 1$. They are not appropriate for the design of lowpass and bandstop filters however they can be used for bandpass and highpass filters.

In Figure 10 (a), the linear relationship is shown by plotting filter phase response (type I). The phase response shows a jump discontinuity of π radians at the corresponding frequency when the transfer function has a zero on the unit circle in the z plane, and the plot applies a jump discontinuity of 2π every time the phase response goes beyond $\pm\pi$ so that the overall phase response stays within the principal range of $\pm\pi$. If there are no zeros on the unit circle, that is, if there are no jump discontinuities of π radians, the phase response, when it is unwrapped, becomes a constant function of ω. The result of unwrapping the phase (as shown in Figure 10 (a)) is to eliminate the jump discontinuities in the phase response in such way that the phase response stays within $\pm\pi$ (as shown in Fig. 10 (b)). Its group delay is an integer multiple of samples equal to $N/2$ samples if the order N of the FIR filter is even. Whereas, when the order N is odd, the group delay is equal to (an integer plus half) a sample.

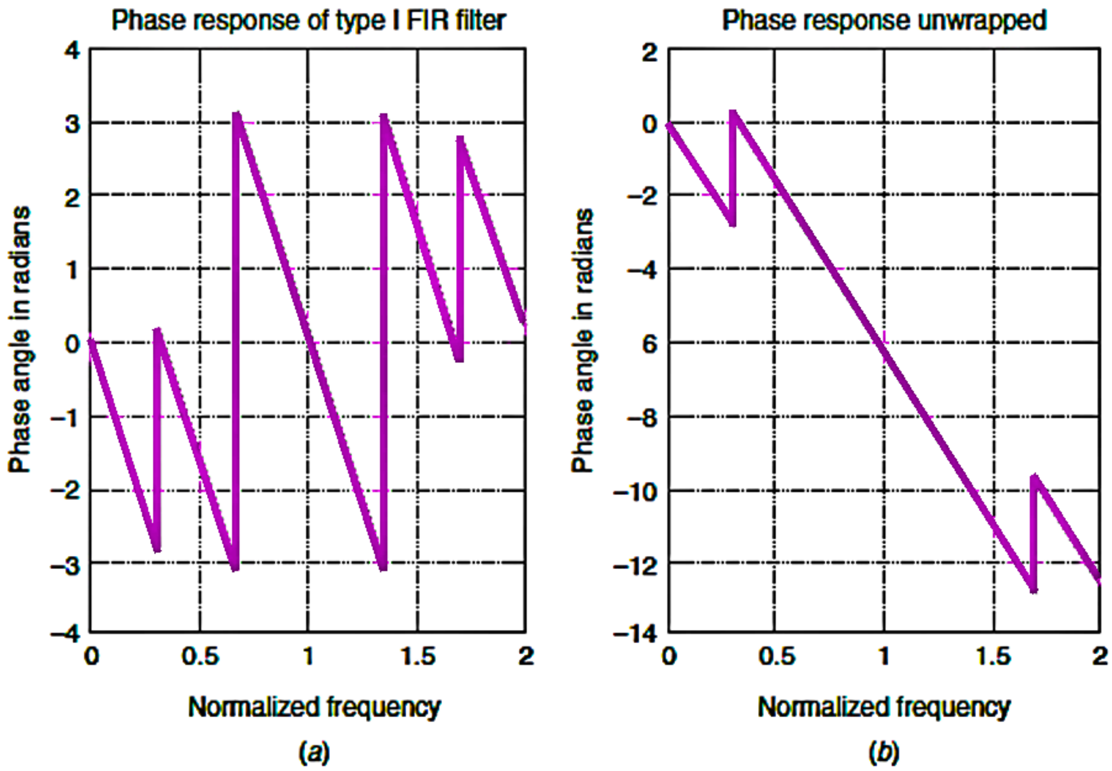


Figure 10: Linear Phase Responses of Type I FIR Filter

Chapter 3

GROUP DELAY REDUCTION METHODS

3.1 General Overview

Fast Fourier Transform (FFT) based circular convolution is used by Overlap-Save Method (OSM) to construct equivalent results as in the linear convolution. After that, the aliasing that occurs because of the circular convolution could be eliminated only following zero padding after the last nonzero impulse response (IR) sample. The convolution length for two signals having size N and M is $L = N + M - 1$. Therefore, for an IR with size N, the minimum number of added zeros is $M - 1$. For a considered piece period of signal with duration M, $N - 1$ unwanted samples are to be deleted. While the OSM is in use, these samples are first on the circular convolution result provided by $X(k)\,H(k)$. OSM result will provide a group delay equal to $(\mathrm{N} - 1)/2$ for N odd and $N/2$ for N even if $h(n)$ is a linear phase IR. Nevertheless, it is seen that Discrete Fourier Transform (DFT) of a samples series relies on its layout and the phase can be reduced for a certain layout [11]. To prove that, consider a linear phase IR of length N, denoted $h(n)$ (Figure 11). Its DFT $H(k)$ is given by:

$$H(k) = \sum_{n=0}^{N-1} h(n) W_L^{-kn} \tag{3-1}$$

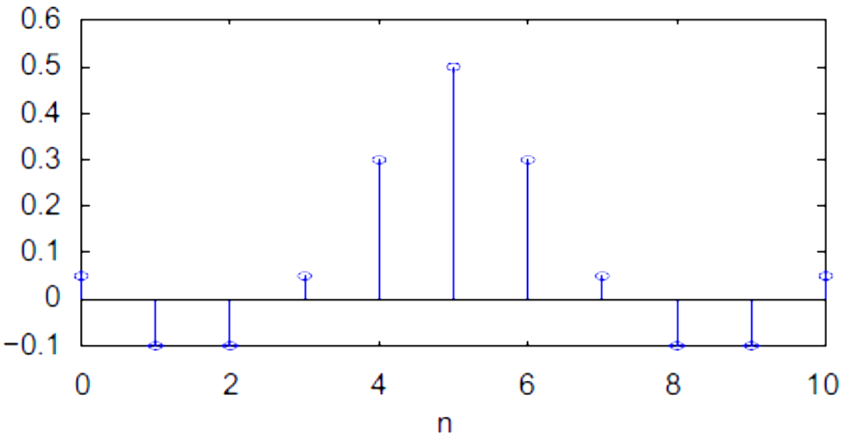


Figure 11: Linear Phase IR h(n)

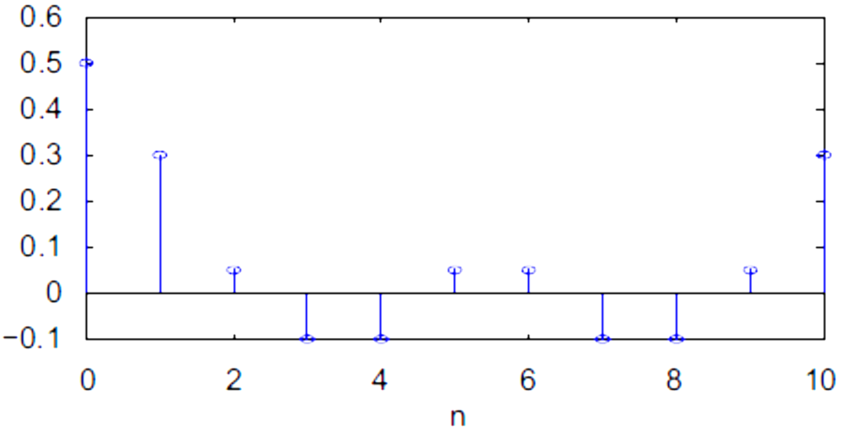


Figure 12: Circularly Shifted Linear Phase IR h_1(n)

with $W_L^{kn} = e^{j2\pi kn/L}$, $L = N + M - 1$. During the calculation of the DFT, it is assumed that the samples series are periodic, and are positioned on a circle whose origin synchronizes with the first sample of $h(n)$. The sort of $h(n)$ being $N - 1$, its phase is given by

$$\varphi = -2\pi \frac{N-1}{2} f \qquad (3\text{-}2)$$

Therefore the group delay is *(N −1)/*2. Now, by shifting the IR of $(N - 1)/2$ samples circularly toward the left the origin is changed as in Fig. 12. The DFT of the new IR is represented by $h_1(n)$ will be:

$$H_1(k) = \sum_{n=0}^{N-1} h_1(n) W_L^{-kn} \tag{3-3}$$

Equation (3-3) may be written also as below

$$H_1(k) = \sum_{n=0}^{N-1} h\left(n + \frac{N-1}{2}\right) W_L^{-kn} \tag{3-4}$$

The relation in equation (3-4), according to the DFT characteristics, lastly can be written as follows:

$$H_1(k) = W_L^{k(N-1)/2} \sum_{n=0}^{N-1} h(n) W_L^{-kn} \tag{3-5}$$

Equation (3-5) demonstrates that the phase of $h_1(n)$ compared to that of $h(n)$, is given by

$$\varphi_1 = 2\pi \frac{N-1}{2} f \tag{3-6}$$

Which shows the cancellation of the dephasing φ because $h(n)$. Consequently, $h_1(n)$ shows a zero phase if *N* is odd and its phase is $\varphi_1 = \pi f$ if N is even. This type which is named as zero phase IR could be utilized in the FFT based circular convolution to achieve zero group delay in filtering. However, this kind of operation is impossible since OSM and overlap-add method (OAM) techniques which are based on the basics of using the DFT, advise zero padding at the IR end. Thus, the DFT phase of the IR $h_2(n)$ acquired after zero padding to $h_1(n)$ would become, as in Fig. 13, unspecified. Fig. 13 shows the frequency response of $h_1(n)$ after zero padding [11]. As a result, it appears essential to develop a new algorithm based on the utilization of DFT, which keeps the zero phase IR properties [15].

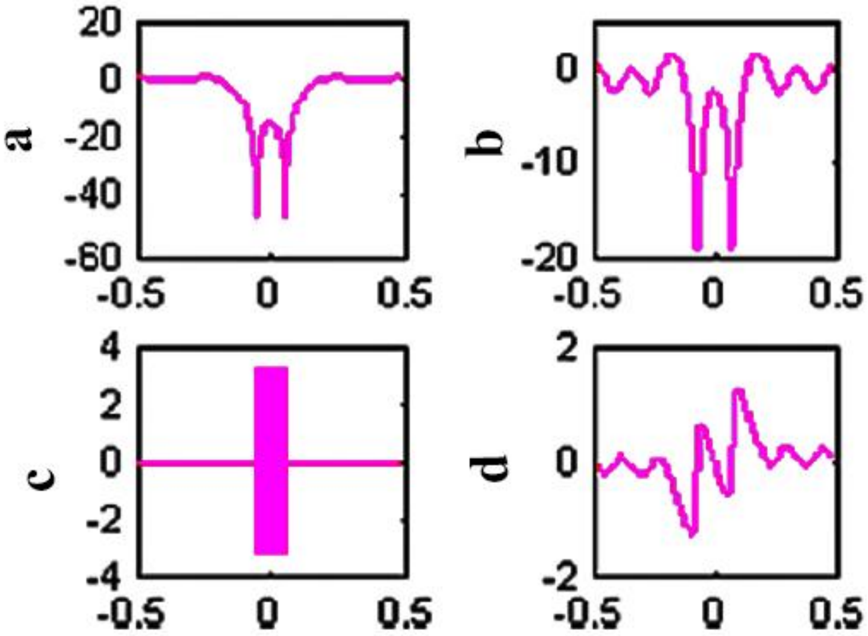


Figure 13: Frequency Response of $h_1(n)$ after zero padding: (a) magnitude response of $h_1(n)$, (b) magnitude response of $h_2(n)$, (c) phase response of $h_1(n)$, (d) phase response of $h_2(n)$.

3.2 Modified Overlap and Save Method (MOSM)

In this section, a new algorithm is presented for filtering with the use of the FFT based circular convolution and the OSM. The idea here is to make it easier to realize the zero group delay in the filtering. Here, no zero padding occurs after the zero phase IR which means that it is possible to maintain its spectral properties [16]. Let $y(n)$ be the circular convolution result of $h(n)$ and $x(n)$, its DFT will be written as

$$Y(k) = X(k)H(k) \qquad (3\text{-}7)$$

According to the OSM, if the size of $y(n)$ is $\mathrm{L} = \mathrm{N} + \mathrm{M} - 1$, N being the size of $h(n)$, then the $N - 1$ first samples of $y(n)$ should be omitted. By determining the circular convolution $y_1(n)$ of $h_1(n)$ and $x(n)$, one obtains

$$Y_1(k) = W_L^{k(N-1)/2} X(k)H(k) \tag{3-8}$$

This formula demonstrates that the result obtained with the zero phase IR is circularly shifted of $(N\ -1)/2$ samples to the left like $h_1(n)$. As a result, $(N\ -1)/2$ samples of the $N-1$ to be deleted should be suppressed by each of the two extremities of the circular convolution result. Next, it seems that after the $N\ -\ 1$ first samples of $y(n)$ are deleted by the OSM they will be unsuitable to the use of the zero phase filters. Thus, we name the MOSM whose idea is to delete $(N\ -1)/2$ samples on both sides of the circular convolution result extremities. It can be noticed that this convolution result rotation does not modify the group delay produced by the filter, thus the MOSM offers the same results like the other conventional filtering techniques. In order to reduce the group delay, the redefinition of the samples to be retained is required.

3.3 Zero Delay MOSM (ZDMOSM)

It is possible to perform zero delay filtering by the use of the MOSM. Therefore, rather than adding $N\ -\ 1$ zero samples to the beginning of file to be filtered as suggested by the OSM, as explained in Fig. 14, only $(N-1)/2$ samples will be considered [11]. Next, by deleting $(N\ -1)/2$ samples on both sides of the filtered signal, the group delay is suppressed. This method is named as the zero delay MOSM (ZDMOSM). An example of a filtered signal without group delay is shown in Fig. 15 [17].

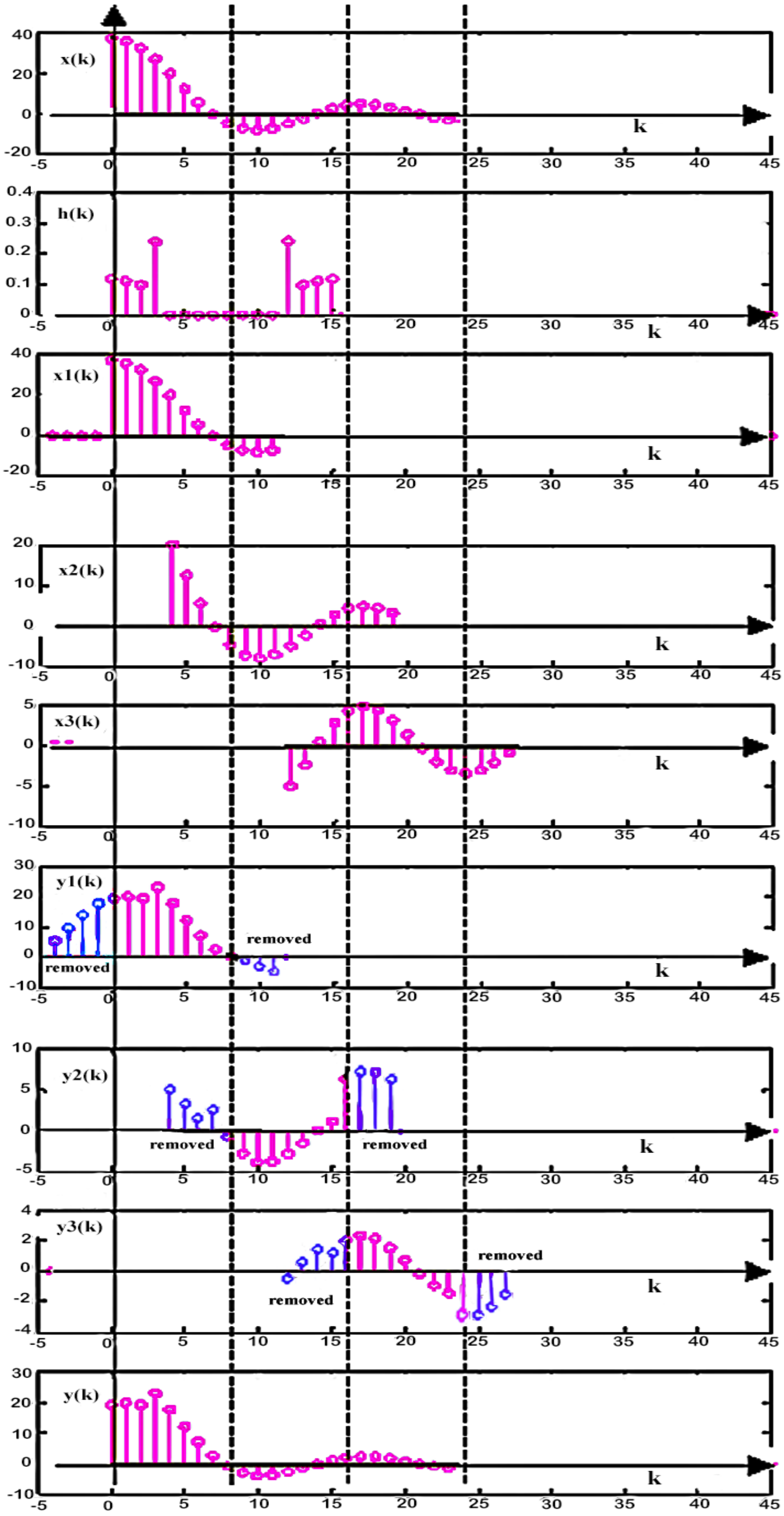
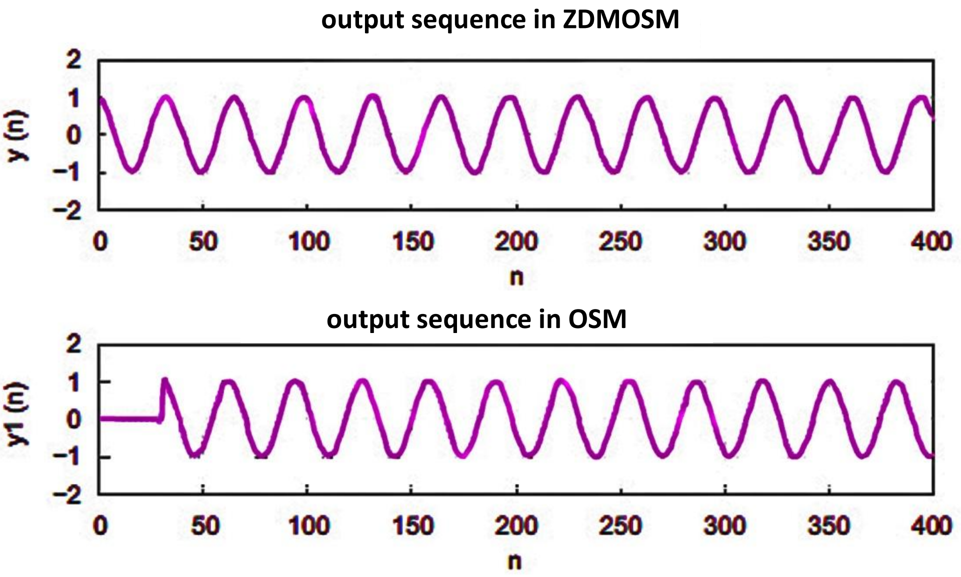


Figure 14: Principle of the ZDMOSM


output sequence in ZDMOSM

output sequence in OSM

Figure 15: Zero Group Delay Filtering using the ZDMOSM

3.4 Reduced Delay MOSM (RDMOSM)

Here ZDMOSM method cannot be applied since it would need altering the acquisition time during the processing, which is not easy. Next, when starting filtering as in the case of the OSM, $\mathrm{N}-1$ zero samples are taken into consideration. In general, the $N-1$ deleted samples present some ripples. These ripples, when emerging on the filtered signal can create an awkward effect (for example in audio filtering). But, their amplitude reduces slowly from left to right as seen in Fig. 16. In the real time processing, the decrease of the group delay might be considered by deleting, following circular convolution with the zero phase filter, rather than $(N-1)/2$ samples each sides as explained in the MOSM, however $3(N-1)/4$ on the left end and only $(N-1)/4$ on the right end of the result. This method maintains at the right end $(N-1)/4$ of the samples which usually should have been deleted, according to the OSM. Whereas, $(N-1)/4$ of the samples carrying the group delay is deleted, this reduces the group delay into a half. This method is named as the reduced delay

MOSM (RDMOSM). Fig. 17 shows a filtered signal example (in real time) in which the groups delay was reduced [11].

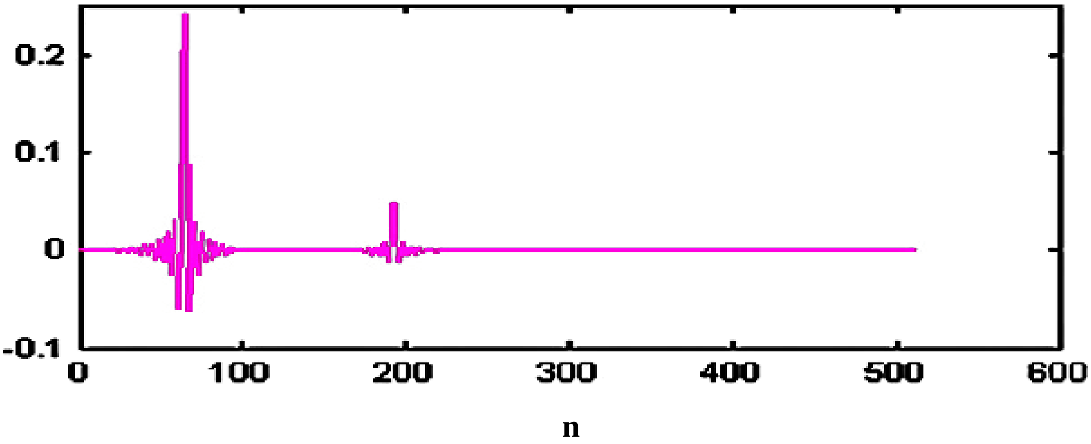


Figure 16: Evolution of the Ripples Amplitude

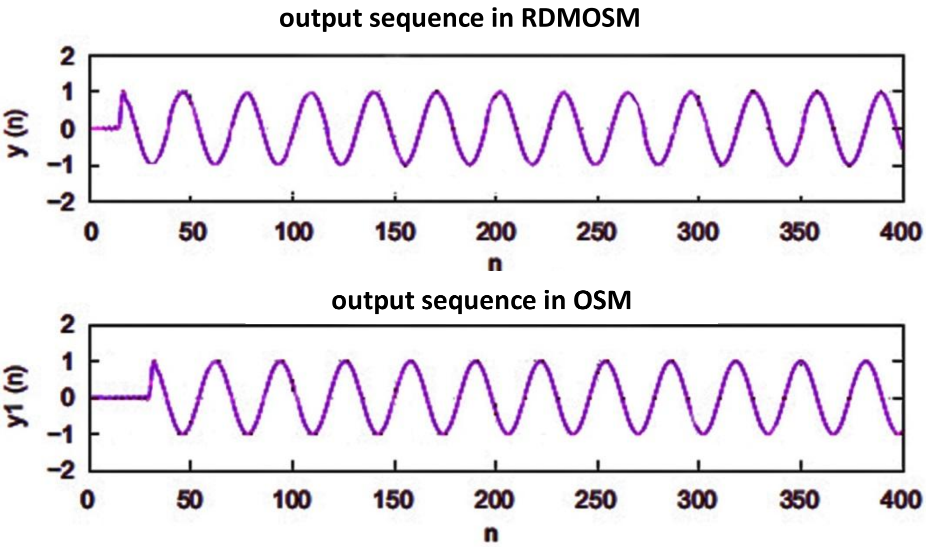


Figure 17: The use of RDMOSM to Reduce Group Delay Filtering

3.5 Enhanced RDMOSM (ERDMOSM)

In this section, the new proposed filtering algorithm via DFT based circular convolution and the OSM will be examined.

The main aim of this method is to realize zero group delay filtering. In this method, there exists no zero padding to zero phase IR. Let us consider $y(n)$ as the result of circular convolution of $h(n)$ and $x(n)$. Now, the DFT of $y(n)$ can be written as follows

$$Y(k) = X(k)H(k) \tag{3-9}$$

According to the OSM, if the size of $y(n)\ is\ L = N + M - 1$ where N is the size of $h(n)$, after that leftmost $N - 1$ samples of $y(n)$ have to be deleted. By evaluating the circular convolution $y_1(n)$ of $h_1(n)$ and $x(n)$, we are able to show that:

$$\left.\begin{aligned} Y_1(k) &= H_1(k)X(k) \\ &= e^{j(2\pi/L)k((N-1)/2)}\,H(k)X(k) \\ &= W_L^{k((N-1)/2)}H(k)X(k) \end{aligned}\right\} \tag{3-10}$$

Taking inverse DFT gives

$$y_1(n) = \left(y\left(n + \frac{N-1}{2}\right)\right) \tag{3-11}$$

This equation demonstrates that the result obtained by the zero phase IR is a circularly shifted version of $y(n)$ shifted by $(N-1)/2$ samples to the left. Therefore, $(N-1)/2$ samples out of $N - 1$ samples to be deleted have to be suppressed on each side of the circular convolution result extremities. After that, it is found that the OSM which deletes the $N - 1$ first samples of $y(n)$ is not appropriate to be utilized in zero phase filters. As a result, the MOSM method was defined and its working principle is to delete $(N-1)/2$ samples on both ends of the circular convolution result. The convolution result rotation $y\ (n)$ does not change the group delay generated by the filter. Thus, the MOSM offers the same result as obtained from the OSM or other usual filtering method. Afterward, it is noted that a reduction in the group delay can be obtained by re-defining the samples to be kept from last rotated result of the circular convolution.

In case of RDMOSM, a reduction in the group delay can reach half. This technique has been extended to one where reduction in the group delay can be obtained using a factor more than half. In performing DFT based circular convolution the $N-1$ zero samples should be considered to be filtered as in the case of the OSM. Following circular convolution of $x(n)$ with $h_1(n)$, $4(N-1)/5$ and $9(N-1)/10$ of the samples are deleted from the left end side. Similarly, $(N-1)/5$ and $(N-1)/10$ of the samples are deleted from the right end side of the result. These are named as enhanced RDMOSM, ERDMOSM1 and ERDMOSM2, respectively. This process preserves at the left end side $(N-1)-4(N-1)/\ 5=(N-1)/5$ samples in ERDMOSM1 case and $(N-1)-9(N-1)/10=(N-1)/10$ samples in the ERDMOSM2 case, which would usually be deleted according to the OSM. Whereas, $4(N-1)/5$ or $9(N-1)/10$ samples carrying the group delay are deleted, correspondingly, for the ERDMOSM1 or ERDMOSM2 cases. These lead to the reduction of the group delay by a factor given by

$$\left(1-\frac{\frac{N-1}{5}}{\frac{N-1}{2}}\right)=\frac{3}{5} \qquad (3\text{-}12)$$

and

$$\left(1-\frac{\frac{N-1}{10}}{\frac{N-1}{2}}\right)=\frac{4}{5} \tag{3-13}$$

Figures 18 and 19 show an example of a filtered signal (in real time) in which the reduction in the group delay happened by a factor 3/5 and 4/5, correspondingly, in ERDMOSM1 and ERDMOSM2 cases.

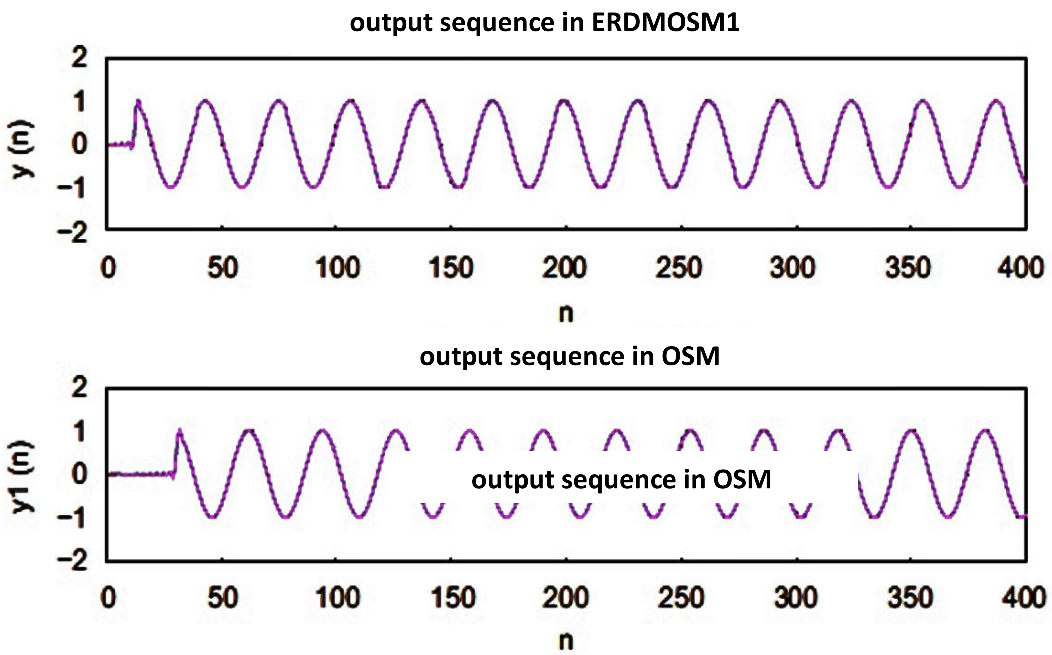


Figure 18: Reduced Group Delay Filtering using ERDMOSM1

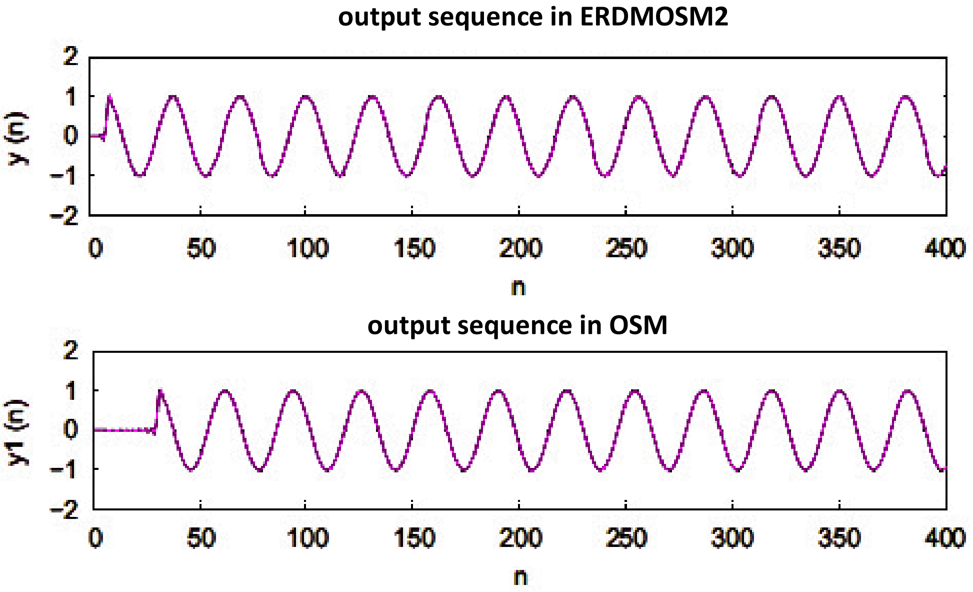


Figure 19: Reduced Group Delay Filtering using ERDMOSM2

It is observed that for a filter of order 60, the result of the output signal begins from 31^{st} sample in the OSM case (see Fig. 17) or linear convolution (the group delay is $(N-1)/2 = 30$ samples), where in the RDMOSM case (see Fig. 17), the output sequence begins from the 16^{th} sample (i.e., the group delay is $\frac{1}{2} * \ (N-1)/2 = 15$ samples) and in ZDMOSM, the output sequence begins from the 1^{st} sample meaning that there is no group delay. On the opposing side, for the same input signal, the output sequence of ZDMOSM varies a lot from the OSM filtered output which shows a deviation from linear convolution result. In the present ERDMOSM1 case (see Fig. 18), the filtered output sequence begins from the 13^{th} sample (i.e., the group delay is $\frac{2}{5} * \frac{N-1}{2} = 12$ samples) and for $\mathrm{ERDMOSM2}$ case (see Fig 19), the filtered output sequence begins after 6 samples, from 7^{th} sample (i.e, the group delay is $\frac{1}{5} * \ (N-1)/2 = 6$ samples).

The full algorithms are shown in Figures 20 and 21. The results obtained from each of these algorithms are better than the usual RDMOSM.

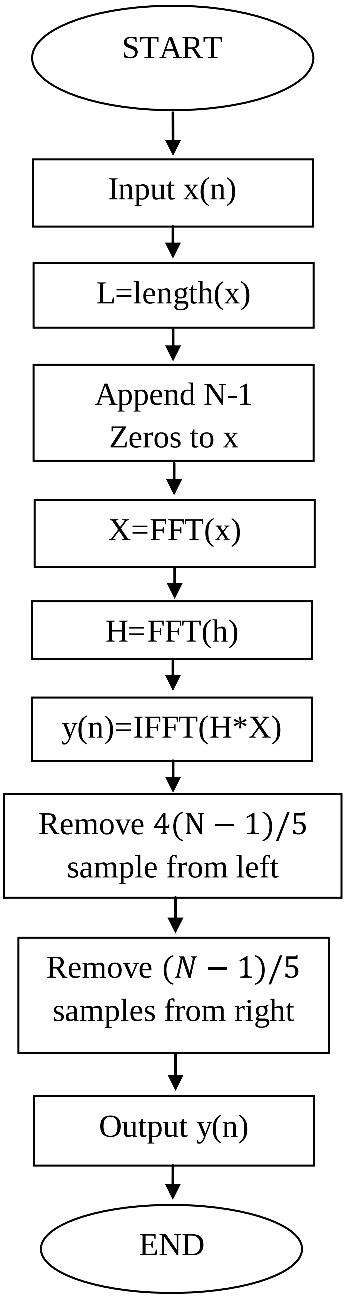


Figure 20: Algorithm for ERDMOSM1

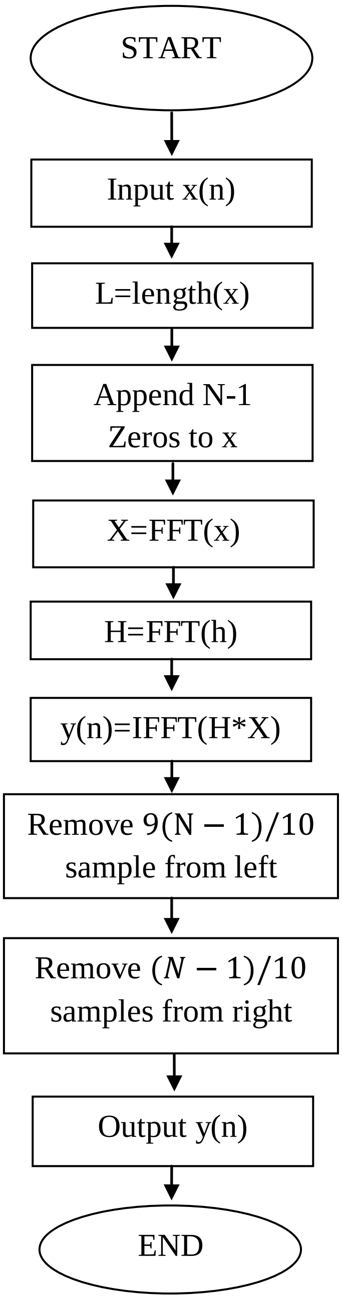


Figure 21: Algorithm for ERDMOSM2

In our present work, it has been observed that if the movement in the number of samples is increased from the left side of the circular convolution result, there will be a reduction in the group delay that increases with the increase of the movement of the number of samples at the cost of increased ripple amplitude. If $N-1$ samples are deleted from the left of the circular convolution result side, the group delay will be totally suppressed since no sample carrying the group delay will remain, which will increase the deviation of the resulting filtered sequence from the OSM result. Thus, the result will be invalid. Therefore, samples have to be deleted from each sides of the circular convolution result to attain better performance.

The reason of removing $(N-1)/5, (N-1)/10$ numbers of samples from the right and $4(N-1)/5, 9(N-1)/10$ numbers of samples from the left can be explained as follows. Three important conditions should be followed in order to re-define the samples to be retained after circular convolution:

- The overall number of samples to be deleted from left and right side of final circular convolution result of length $L = (M + N - 1)$ have to be $(N-1)$ as the original input length is M.
- More samples are to be deleted from left instead of right end of the circular convolution result since the reduction factor of the group delay depends on how many samples are deleted from the left.
- Samples should be deleted from each sides of the circular convolution result to obtain better performance [17].

Chapter 4

COMPUTER SIMULATIONS

In this chapter, two cases will be presented to show the effectiveness of group delay reduction to the recently suggested algorithms defined in chapter 3. In order to be able to compare these filtering techniques with the traditional OSM, the results with reduced delay are every time compared to those obtained with the OSM.

where $y_1(n)$ and $y(n)$ represent the filtered output OSM and the filtered output regarding each filtering technique, respectively. The input signal that is to be filtered is denoted by $x(n)$ and the zero phase impulse response is represented by $h(n)$. In the simulation studies, sine and random waves are considered as the input.

4.1 Simulation Study 1: Sine Wave

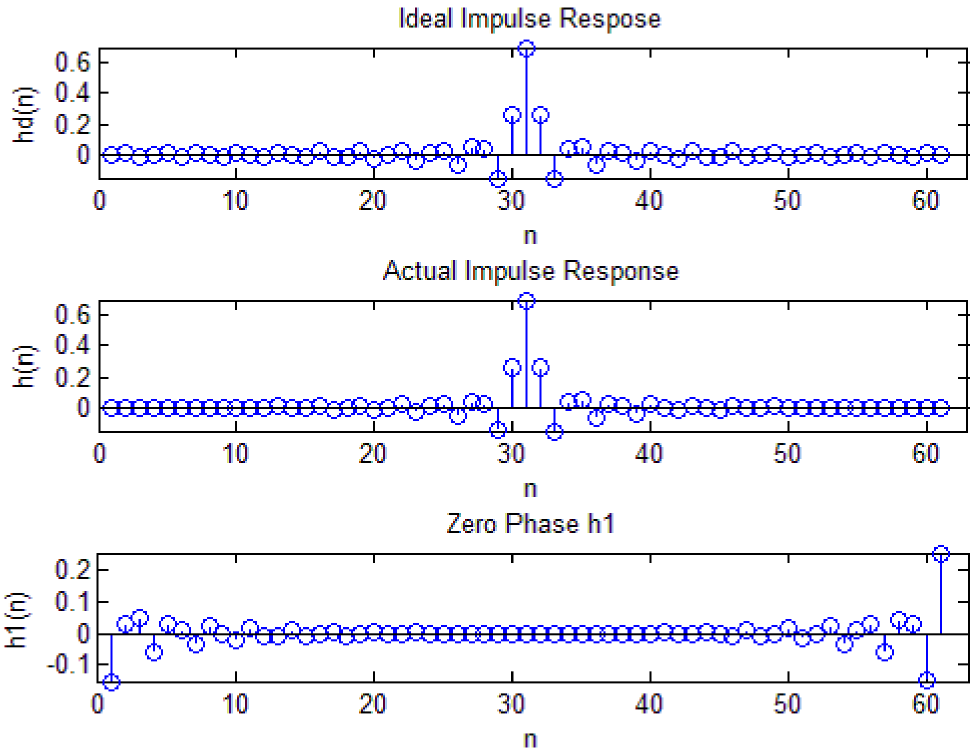


Figure 22: Impulse Response of FIR Filter Lowpass Equiripple Filter & Actual Impulse Response and Zero Phase $h_1(n)$

Fig. 22 shows the desired impulse response, actual impulse response and zero phase impulse response of FIR filter. It is clear that zero phase impulse response can be obtained by circularly shifting the impulse response of the filter by (N-1)/2 samples toward left.

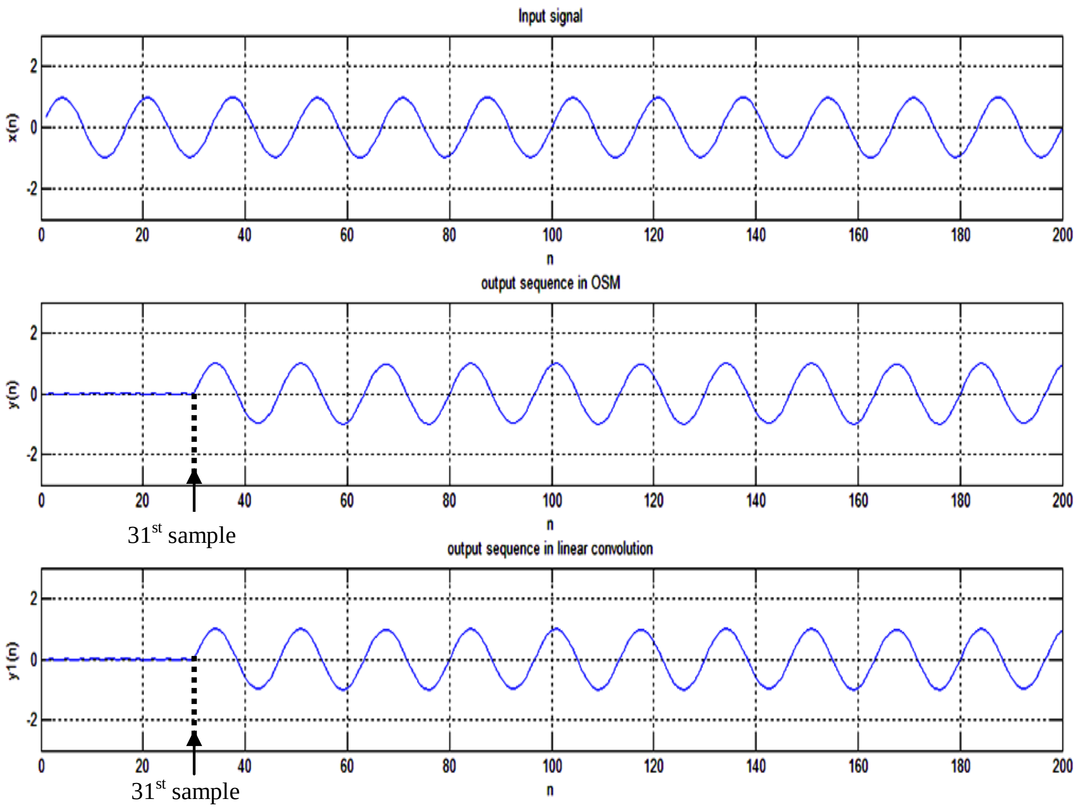


Figure 23: Comparison Between the Linear Convolution and OSM method

Fig. 23 shows the comparison between linear convolution and OSM method. Since two methods are equivalent, the OSM method can be considered as the reference in the other parts of this discussion. It has been observed that for a filter length of 60 coefficients, the resultant output signal starts from 31st sample in the case of OSM (or linear convolution) which shows that the group delay is (N-1)/2=30 samples.

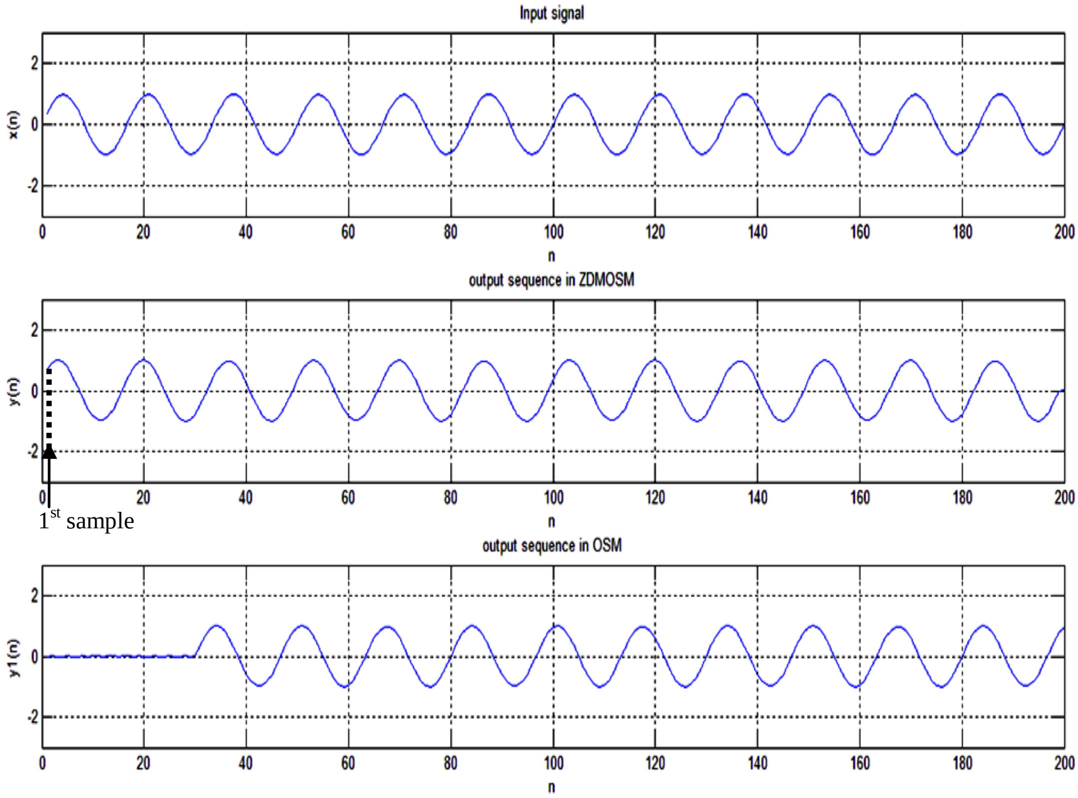


Figure 24: Comparison between ZDMOSM and OSM methods

Fig. 24 shows the output of the filter obtained by OSM and MOSM methods for the off-line processing case. The input signal, $x(n)$, the output of the filter obtained by the ZDMOSM, the output of the filter obtained by the OSM are illustrated, from top to bottom, in Fig. 24. It is clear the output obtained by ZDMOSM is different from that of obtained by the OSM.

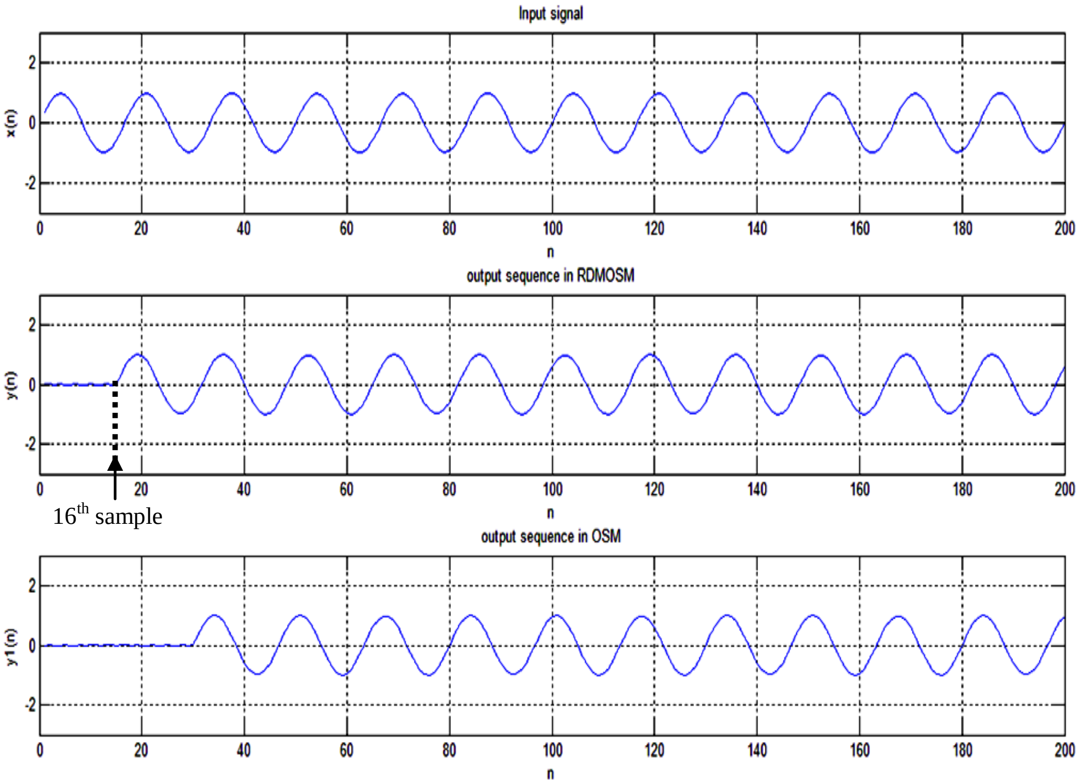


Figure 25: Comparison between RDMOSM and OSM methods

Fig. 25 shows the output of the filter obtained by OSM and RDMOSM methods for the same input shown in Fig. 24. If the output signals obtained by ZDMOSM in Fig. 24 and by RDMOSM in Fig. 25 are compared, one can easily see that the group delay has been reduced by a factor of 1/2.

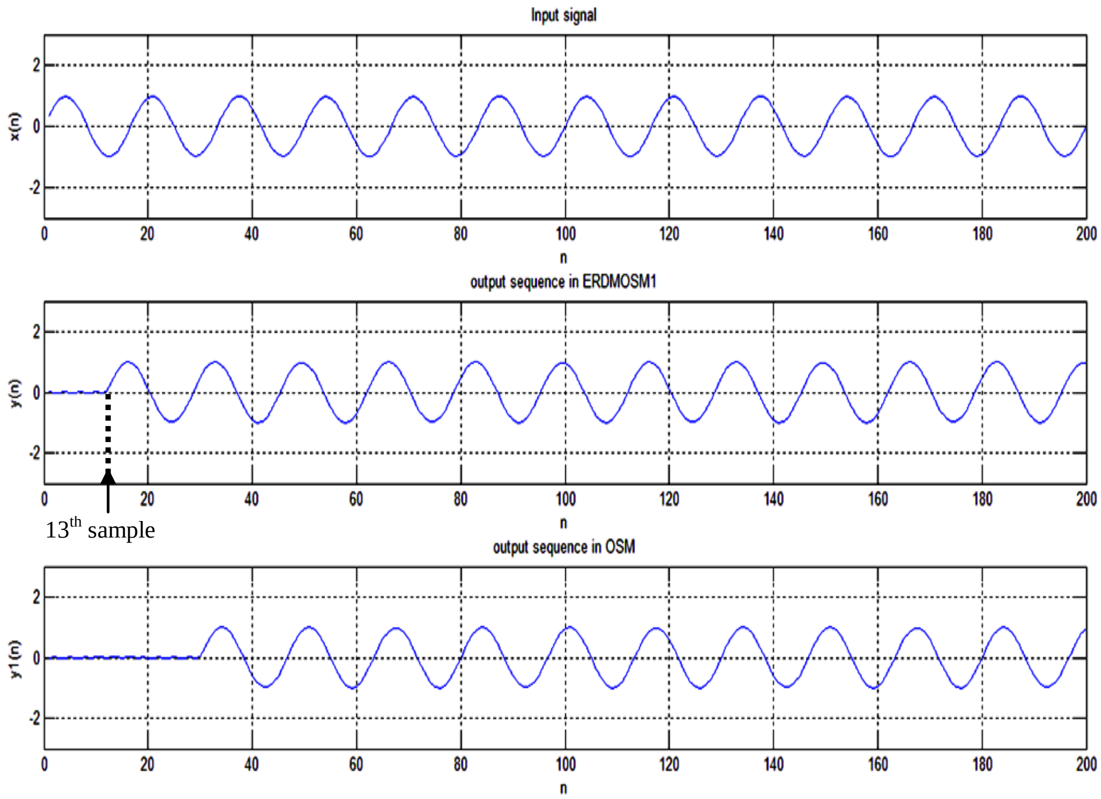


Figure 26: Comparison between ERDMOSM1 and OSM methods

Fig. 26 and 27 show the results obtained from the OSM and enhanced-RDMOSM (ERDMOSM1 and ERDMOSM2). In Fig. 26, the results obtained by OSM and ERDMOSM1 are compared. Similarly, in Fig. 27, the results obtained from the OSM and ERDMOSM2 are compared. It can be clearly seen from these figures that ERDMOSM1 and ERDMOSM2 methods result in better delay reduction. The delay reductions obtained by the ERDMOSM 1 and ERDMOSM2 are approximately 3/5 and 4/5, respectively.

The performances of the group delay reduction methods are compared and the results are reported in Table 1. Except ZDMOSM, it is obvious that ERDMOSM1 and ERDMOSM2 performs better than the other methods.

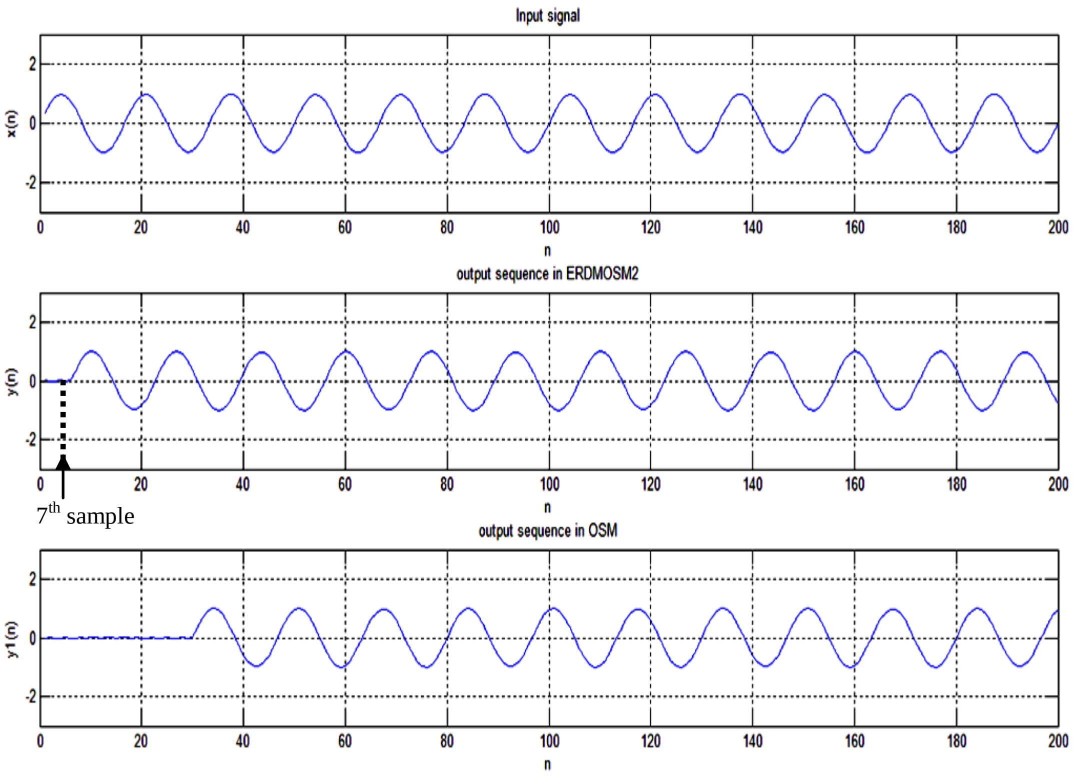


Figure 27: Comparison between ERDMOSM2 and OSM methods

Table 1: Comparison of Group Delay Reduction Methods

Methods	**Group Delay (samples)**
Linear Convolution	30
OSM	30
ZDMOSM	0
RDMOSM	15
ERDMOSM1	12
ERDMOSM2	6

4.2 Simulation Study 2: Random Wave

Another FIR, equiripple filter having passband frequency of 5 kHz, stopband frequency of 6 kHz, sampling frequency of 45 kHz, maximum passband ripple of 0.1 dB and minimum stopband attenuation of 60.1 dB has been considered.

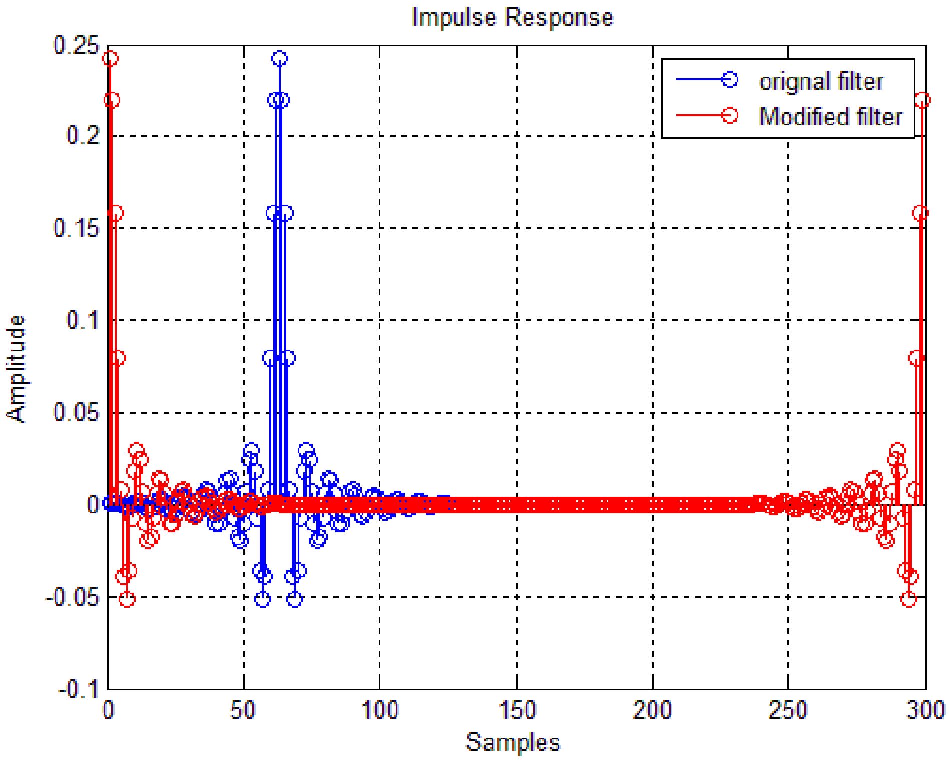


Figure 28: Impulse Response (blue and solid), the Equiripple Linear-phase Filters (red and dash) same Filter after Zero Padding and Circular Left Shifting

In this study, in order to make the filter length equal to the block length L=N+M-1, M-1 zeros have been added following the original filter coefficients. Also, the filter coefficients are circularly shifted to the left by an amount (N-1)/2. Fig. 28 shows the impulse response of the original filter and the modified filter. The parameters in this figure are: length of original filter (N)=121, data points in the segmented input sequence (M)=180, block length (L)=N+M-1=300. Thus, M-1=179 number of zeros have been added following the original filter coefficients. Also, the amount of the circular left shift is for (N-1)/2 = 60 number of samples.

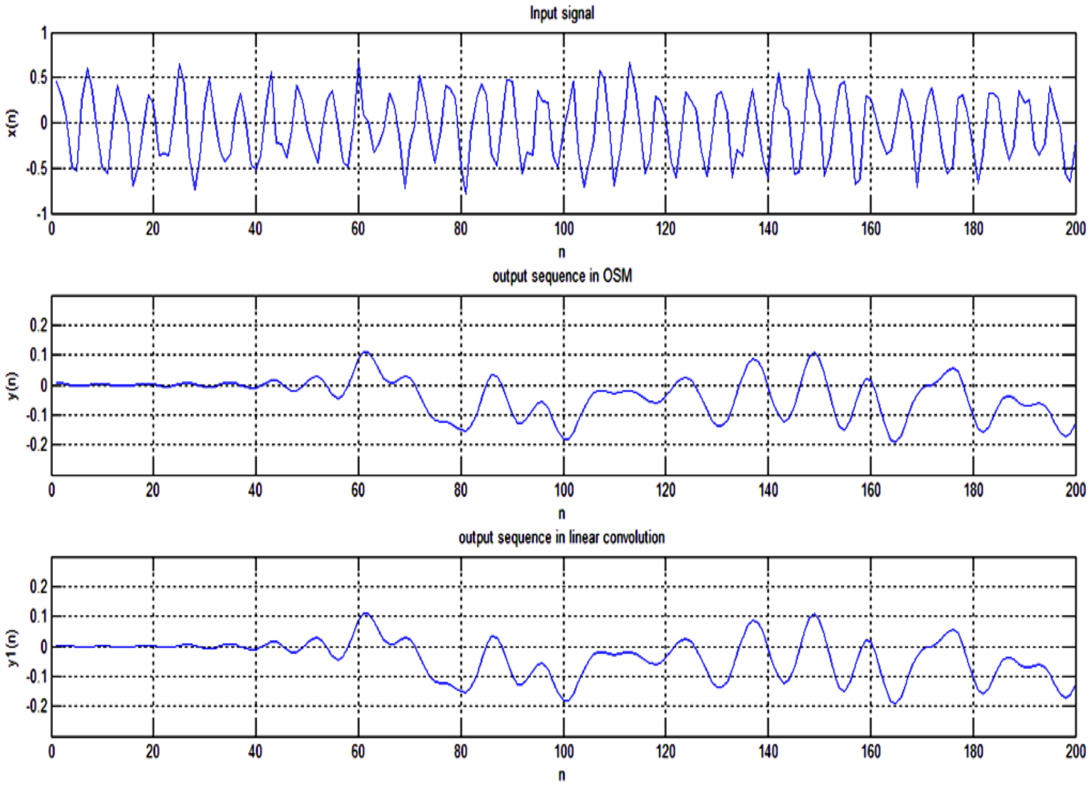


Figure 29: Comparison between OSM method and the Linear Convolution

Fig. 29 shows the comparison between linear convolution and OSM method. Since two methods are equivalent, the OSM method can be considered as the reference in the other parts of this discussion.

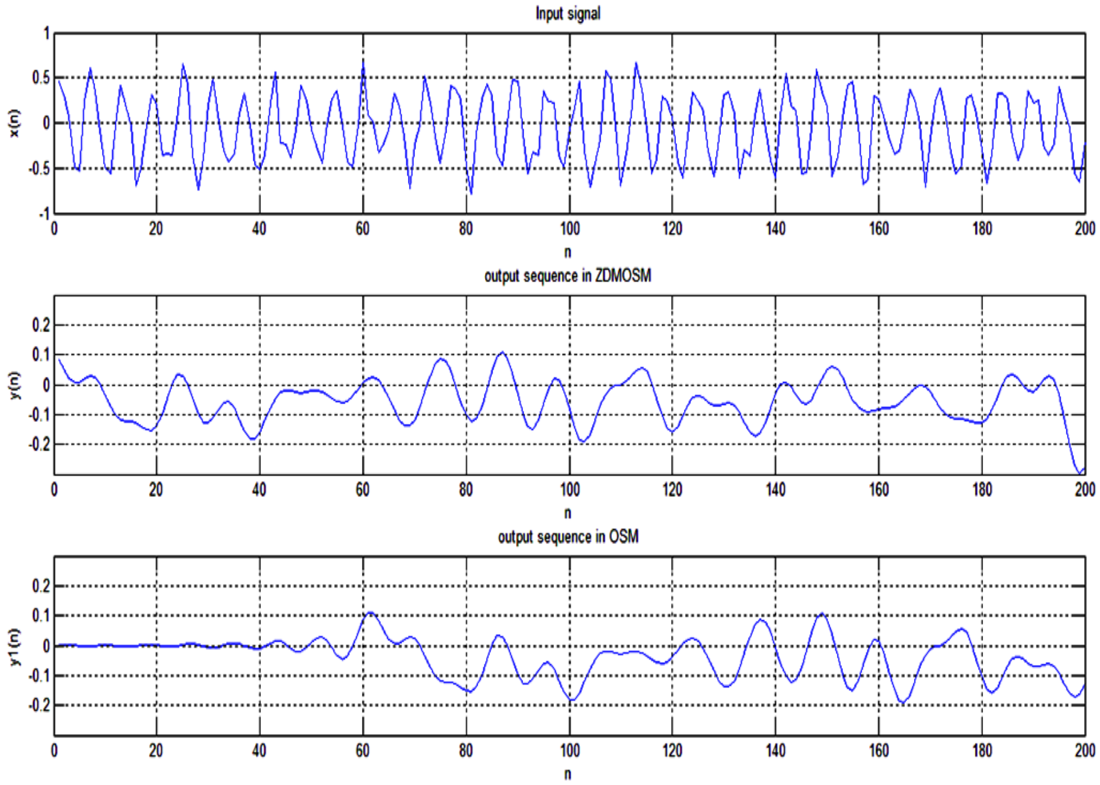


Figure 30: Comparison between ZDMOSM and OSM methods

Fig. 30 shows the output of the filter obtained by OSM and MOSM methods for the off-line processing case. The input signal, $x(n)$, the output of the filter obtained by the ZDMOSM, the output of the filter obtained by the OSM are illustrated, from top to bottom, in Fig. 30. It is clear the output obtained by ZDMOSM is different from that of obtained by the OSM.

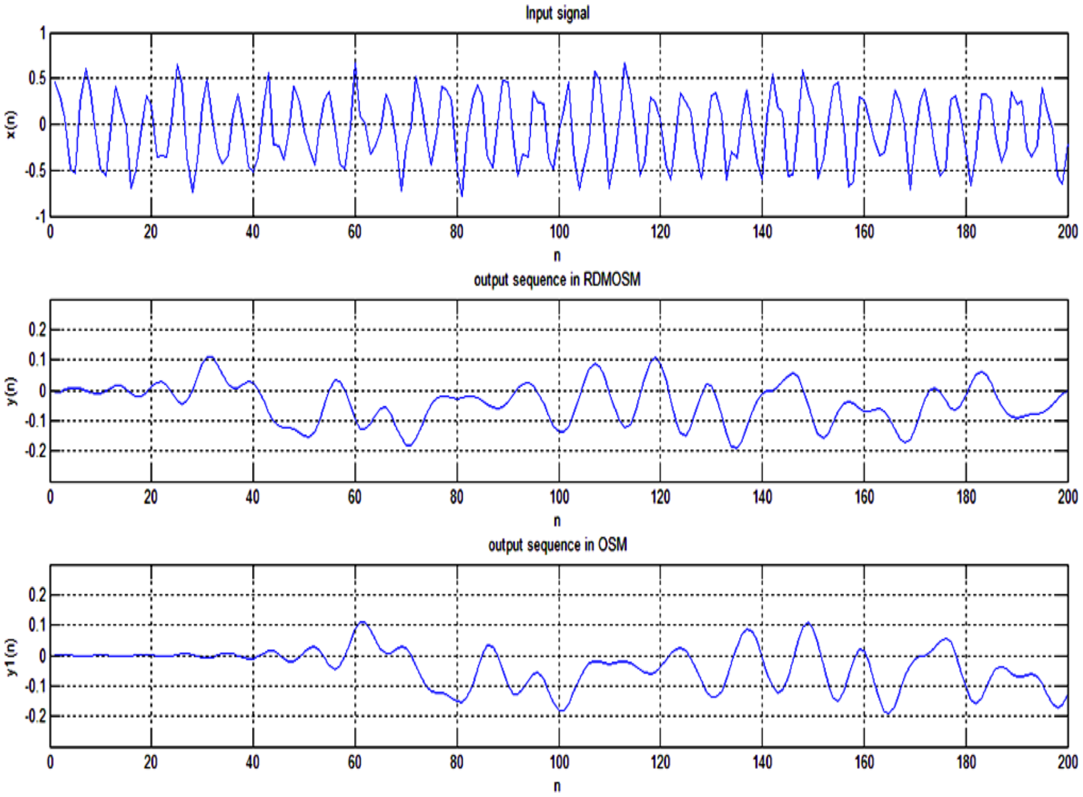


Figure 31: Comparison between RDMOSM and OSM methods.

Fig. 31 shows the output of the filter obtained by OSM and RDMOSM methods for the same input shown in Fig. 30. If the output signals obtained by ZDMOSM in Fig. 30 and by RDMOSM in Fig. 31 are compared, one can easily see that the group delay has been reduced by a factor of 1/2.

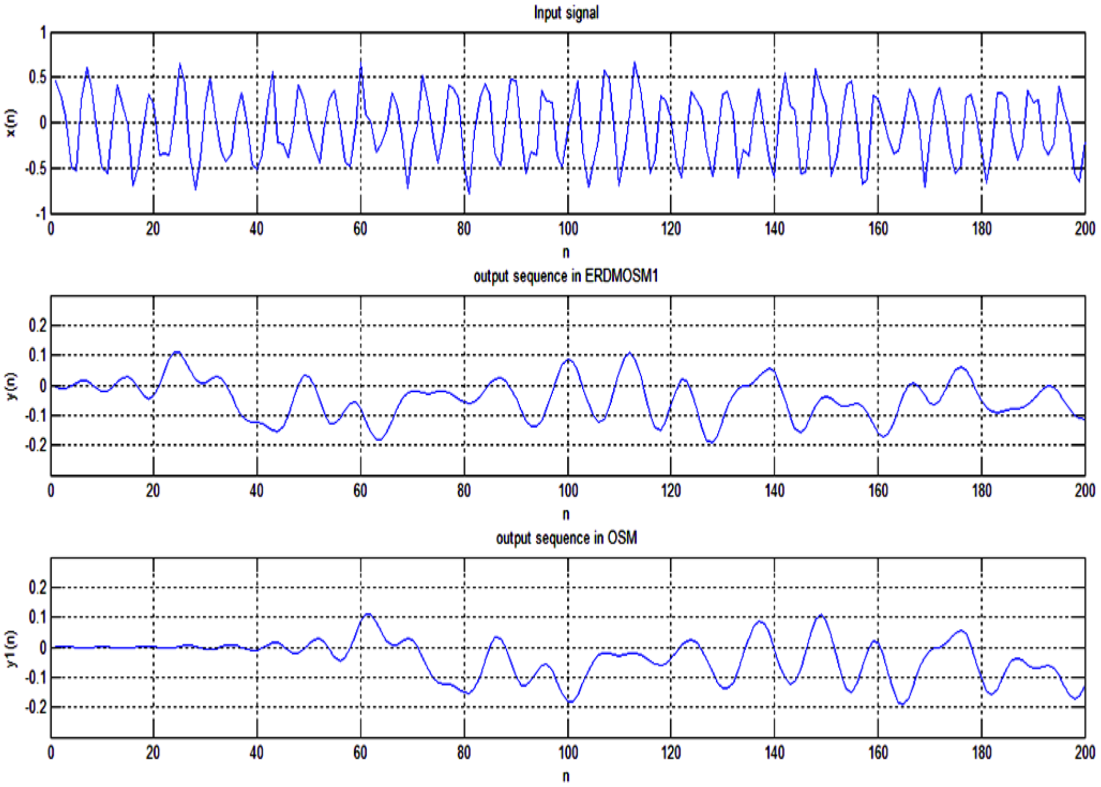


Figure 32: Comparison between ERDMOSM1 and OSM methods.

Fig. 32 and 33 show the results obtained from the OSM and enhanced-RDMOSM (ERDMOSM1 and ERDMOSM2). In Fig. 32, the results obtained by OSM and ERDMOSM1 are compared. Similarly, in Fig. 33, the results obtained from the OSM and ERDMOSM2 are compared. It can be clearly seen from these figures that ERDMOSM1 and ERDMOSM2 methods result in better delay reduction. The delay reductions obtained by the ERDMOSM 1 and ERDMOSM2 are approximately 3/5 and 4/5, respectively.

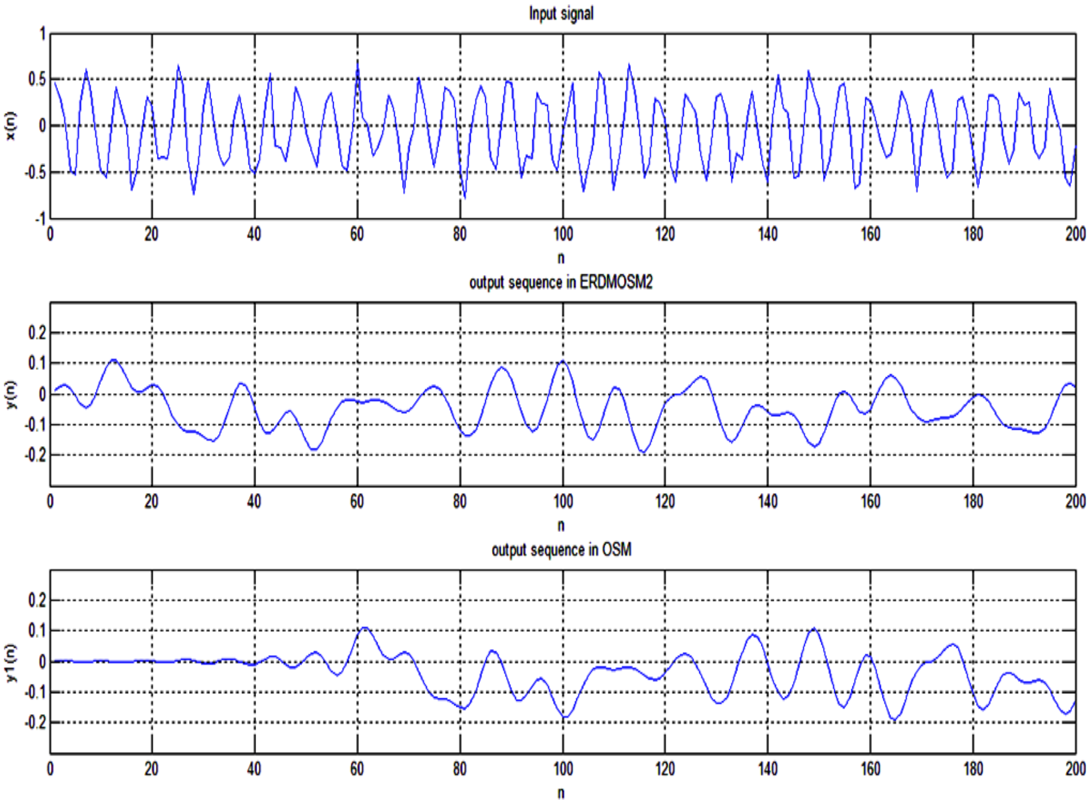


Figure 33: Comparison between ERDMOSM2 and OSM methods.

Chapter 5

CONCLUSION

Block convolution techniques such as overlap-add method (OAM) and overlap-save method (OSM) are usually used for a long input sequence to be filtered. In these methods, however, the output sequence has a finite group delay with respect to input. In this thesis, the performance of enhanced modified overlap and save method (ERDMOSM) has been investigated in reducing that group delay. In this method, the impulse response (IR) is made causal and then IR has been circularly shifted to the left by an amount of (N-1)/2 for N odd and N/2 for N even, where N is the length of the filter. This modified IR has been used in OSM based block convolution technique. Finally, the samples to be removed from the final convolution result have been defined. This leads to a minimized group delay. ERDMOSM1 permits a reduction of group delay by a factor of 3/5. On the other hand, ERDMOSM2 reduces the group delay by a factor of 4/5. Also, a compromise between the group delay reduction and the ripple amplitude is obtained. Although the group delay is reduced considerably, there exists some phase problems in passband which makes it ineffective for real time audio applications where group delay deviation in passband should be considered. However, the main advantage of this algorithm is in obtaining 83.33% group delay reduction devoid of using any complicated algorithm and keeping the magnitude response same as those of linear-phase filter.

Continually of this work, in future the following points can be taken in consideration:

i) Reduction of group delay deviation in pass band by using higher order filters, thereby giving rise to more selective filters.

ii) Generalization of the approach to all linear-phase IIR filters to obtain an effective real-time audio application filter.

iii) Application of the proposed approach to filter banks and modification of the method to reduce the overall group delay.

iv) Widening the consideration out to linear phase property of the stop band to optimize the overall delay performance of the filter.

REFERENCES

[1] E. J. Berbari and P. Lander, “Use of highpass filtering to detect late potentials in signal average ECG,” J. Electrocardiol, no. 225, p. 7–12, 1989.

[2] Y. Lin, M. R. Bai and J. Lai, “Reduction of electronic delay in active noise control systems-a multirate signal processing approach,” Electronic 111, pp. 916-924, 2002.

[3] C. S. Barrus and I. W. Selesnick, “Maximally flat lowpass fir filters with reduced delay,” IEEE Transactions on Circuits and Systems, no. 45, p. 53–68, 1998.

[4] B. L. Evans, N. Damera and S. R. McCaslin, “Design of optimal minimum-phase digital fir filters using discrete Hilbert transforms,” IEEE Transactions on Signal Processing, no. 48, p. 1491–1495, 2000.

[5] R. W. Schafer and A. V. Oppenheim, Discrete-Time Signal Processing, Pearson Education, 2006.

[6] N. Aikawa, A. Ogata and M. Sato, “A design method of low delay fir bandpass filters,” IEEE International Symposium on Circuits and Systems, pp. 28-31, 2000.

[7] S. J. You and J. C. Liu, “weighted least squares near equiripple approximation of variable fractional delay fir filter,” IET Signal Processing, no. 1, pp. 66-72, 2007.

[8] S. C. Pei and J. J. Shyu, “A generalized approach to the design of variable fractional-delay fir digital filters,” Signal Processing, no. 88, p. 1428–1435, 2008.

[9] G. Apaydin, “Realization of reduced-delay finite impulse response filters for audio applications,” Digital Signal Processing, 2009.

[10] J. D. Kene, “Extended overlap-save and overlap-add convolution algorithms for real signal,” in IET-UK International Conference on Information and Communication Technology in Electrical Sciences, 2007.

[11] M. Kom, J. S. Fouda, A. Tiedeu and S. Domngang, “Toward a group delay reduction in digital filtering,” Digital Signal Processing, no. 19, p. 22–32, 2009.

[12] H. H. Monson , Schaum's Outline of Theory and Problems of Digital Signal Processing, 1999, pp. 189-190, 289-291.

[13] B. A. Shenoi, Introduction to digital signal processing and filter design, 2006, pp. 251-258.

[14] L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, 1975.

[15] J. F. Kaiser and S. K. Mitra, Handbook for Digital Signal Processing, New York: Wiley, 1993.

[16] T. W. Parks, J. H. McClellan and L. R. Rabiner, “A computer program for designing optimum fir linear phase digital filters,” IEEE Transaction on Audio

and Electro acoustics, no. 21, p. 506–526, 1973.

[17] C. G. Boul, D. Priyanka and K. M. Amit, “Group delay reduction in FIR digital filters,” Signal Processing, no. 91, p. 1812–1825, 2011.

APPENDICES

Appendix A1: Simulation of Sine Wave

```
clear all;
clc;
wp=0.6*pi; ws=0.8*pi;
tr_width=abs(ws-wp);
%Use Hanning window,
M=63;
wc=0.5*(ws+wp); %Ideal LPF cutoff freq.
%========Compute Ideal filter response==============
alpha=(M-1)/2;
n=1:M;
MM=n-alpha+eps; %add smallest number to avoid divide by zero
hd=sin(wc*MM) ./ (pi*MM); %design LPF
%===================================================
figure(1);

h=hd.*hanning(M)';
subplot(3,1,1); stem(1:61,hd(1:61)); title('Ideal Impulse Respose');
axis([0 M -inf inf]), xlabel('n');ylabel('hd(n)');
subplot(3,1,2); stem(1:61,h(1:61)); title('Actual Impulse
Response');
axis([0 M -inf inf]),
xlabel('n'); ylabel('h(n)');
h1=circshift(h',(length(h)-1)/2)';
subplot(3,1,3); stem(1:61,h1(1:61)); title('Zero Phase h1');
axis([0 M -inf inf]), xlabel('n'); ylabel('h1(n)');

%=========Comparison between linear convolution and OSM
method=========
figure(2);

x=sin(2*pi*0.06*(1:200));%the signal to be filtered
subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(1:length(x));
subplot(3,1,2); plot(y); title('output sequence in OSM');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y); title('output sequence in linear
convolution');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y1(n)');
```

```
%=============Comparison between ZDMOSM and OSM method=========
figure(3);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round(length(h)/2):length(x)+round(length(h)/2)-1);
subplot(3,1,2); plot(y); title('output sequence in ZDMOSM');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y1(n)');

%=============Comparison between RDMOSM and OSM method=========
figure(4);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round(length(h)/4):length(x)+round(length(h)/4)-1);
subplot(3,1,2); plot(y); title('output sequence in RDMOSM');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y1(n)');

%=============Comparison between ERDMOSM1 and OSM method=========
figure(5);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round((3/5)*(length(h)/2)):length(x)+round((3/5)*(length(h)/2))-
1);
subplot(3,1,2); plot(y); title('output sequence in ERDMOSM1');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y1(n)');
```

```
%=============Comparison between ERDMOSM2 and OSM method=========
figure(6);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round((4/5)*(length(h)/2)):length(x)+round((4/5)*(length(h)/2))-
1);
subplot(3,1,2); plot(y); title('output sequence in ERDMOSM2');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -3 3]), xlabel('n'); ylabel('y1(n)');

%=======================End of example1===========================
```

Appendix A2: Simulation of Random Wave

```
clc;
clear all;
H = fdesign.lowpass('Fp,Fst,Ap,Ast',5000,6000,0.1,60.1,45000);
hd = design(H,'equiripple');
hd=hd.Numerator;
l=300;
h=[hd zeros(1,l-length(hd)-1)];
stem(hd);
hold on
h1=circshift(h',-round(length(hd)/2)+1)';
stem(h1,'r'); title('Impulse Response');grid;
xlabel('Samples'); ylabel('Amplitude');
leg=legend('orignal filter','Modified filter');
h=hd;

% % %=====Comparison between linear convolution and OSM
method=========
figure(2);

l=300;
x=rem((0.5*(sin(2*pi*0.17*(1:200))+rand(1,200)))-0.3,1);
h=[hd zeros(1,l-length(hd)-1)];
h1=circshift(h',-round(length(hd)/2)+1)';
h=hd;
subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -1 1]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,l-length(h)-1)];
x2=[x zeros(1,l-length(x)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(1:length(x));
subplot(3,1,2); plot(y); title('output sequence in OSM');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in linear
convolution');grid;
axis([0 length(x) -0.3 0.3]);
xlabel('n'); ylabel('y1(n)');

% % %=============Comparison between ZDMOSM and OSM method=========
figure(3);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -1 1]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round(length(h)/2):length(x)+round(length(h)/2)-1);
subplot(3,1,2); plot(y); title('output sequence in ZDMOSM');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -0.3 0.3]);
xlabel('n'); ylabel('y1(n)');
```

```
% % %=============Comparison between RDMOSM and OSM method=========
figure(4);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -1 1]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round(length(h)/4):length(x)+round(length(h)/4)-1);
subplot(3,1,2); plot(y); title('output sequence in RDMOSM');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y1(n)');

% % %=============Comparison between ERDMOSM1 and OSM
method=========
figure(5);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -1 1]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round((3/5)*(length(h)/2)):length(x)+round((3/5)*(length(h)/2))-
1);
subplot(3,1,2); plot(y); title('output sequence in ERDMOSM1');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y1(n)');

% % %=============Comparison between ERDMOSM2 and OSM
method=========
figure(6);

subplot(3,1,1); plot(x); title('Input signal');grid;
axis([0 length(x) -1 1]), xlabel('n'); ylabel('x(n)');
h2=[h zeros(1,length(x)-1)];
x2=[x zeros(1,length(h)-1)];
y=ifft(fft(h2).*fft(x2));
y=y(round((4/5)*(length(h)/2)):length(x)+round((4/5)*(length(h)/2))-
1);
subplot(3,1,2); plot(y); title('output sequence in ERDMOSM2');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y(n)');
y1=filter(h,1,x);
subplot(3,1,3); plot(y1); title('output sequence in OSM');grid;
axis([0 length(x) -0.3 0.3]), xlabel('n'); ylabel('y1(n)');

% % %===================End of example2============================
```