



**NEAR EAST UNIVERSITY**  
**FACULTY OF ENGINEERING**  
**DEPARTMENT OF ELECTRICAL AND**  
**ELECTRONIC ENGINEERING**

**DESIGNING OF FIR FILTERS USING MATLAB**  
**TECHNIQUES**

**GRADUATION PROJECT**  
**EE – 400**

**STUDENT: Nauman Shaukat (960456)**

**SUPERVISOR: Prof. Dr. Fakerddin Mehmedov**

**NICOSIA 2000**



## ACKNOWLEDGEMENT

I am grateful to all the people in my life who have, supported me, advised me, taught me and who have always encouraged me to follow my dreams and ambitions. My dearest parents, my brother and sister, my friends and my tutors. They have taught me that no dream is unachievable.

As in the words of Walt Disney “If you can dream it, you can do it.”

I wish to thank my advisor, Prof. Dr. Fakhraddin Mamedov, for intellectual support, encouragement, and enthusiasm, which made this project possible, and his patience for correcting both my stylistic and scientific errors.

My sincerest thanks must go to my friends, Jamal Fatih, Rabyan Gayas, Kashif Malik and Waqar Ahmed who shared their suggestions and evaluations throughout the completion of my project. The comments from these friends enabled me to present this project successfully.

## ABSTRACT

The project attempts to put together the theories, techniques, and procedures which can be used to design finite impulse response filters. The mathematical tools for the designing of digital filters are introduced briefly, and then the designs are carried out using matlab. It's divided into three sections.

Sections 1 gives a brief introduction of filtering operation, the ideal models that can be used to design any type of filter depending on the specifications of the application. At the end designing method for FIR filters is briefly explained. It gives the preliminaries and steps that should be followed to design a finite impulse response filters.

Section 2 start with the properties of linear phase filters, then it explains the impulse response and frequency response of FIR filters. A brief mathematical design of FIR filter under symmetric and antisymmetric impulse responses and with even and odd order in given. Then designing techniques for the windows like, Rectangular, Bartlett, Hamming, Hanning, Hamming, Kaiser, Naive and Blackman are discussed. Then frequency design techniques are discussed, the chapter ends with the discussion of optimization of FIR filers.

Section 3 deals with the matlab implementation of FIR filters. First the design of four types FIR filters is given using matlab, which are symmetric and antisymmetric, both with even and odd orders. Then the designing of windows which are mentioned above is given.

The chapter ends with conclusion, which compare four types of FIR filters and discuss the advantages of FIR filters over IIR filters and its drawbacks.

# CONTENTS

Acknowledgements	i
Abstract	ii
<b>1. Introduction</b>	<b>1</b>
1.1 Introduction to Filters	1
1.2 Filter Designing Models	3
1.3 FIR and IIR Filters	7
1.4 Preliminaries	7
1.4.1 Absolute Specifications	9
1.4.2 Relative (dB) Specifications	9
1.4.3 Problem Statement	10
<b>2. FIR Filter Design</b>	<b>12</b>
2.1 Properties of Linear-Phase FIR Filters	12
2.1.1 Impulse Response	13
2.1.2 Frequency Response	15
2.1.2.1 Type-1 L-P FIR filters: Symmetrical IR, M odd	15
2.1.2.2 Type-2 L-P FIR filter: Symmetrical IR, M even	16
2.1.2.3 Type-3 L-P FIR filter: Antisymmetric IR, M odd	17
2.1.2.4 Type-4 L-P FIR filter: Antisymmetric IR, M even	18
2.1.3 Zero Locations	18
2.2 Window Design Techniques	20
2.2.0.1 Basic Window Design Idea	23
2.2.1 Rectangular Window	24
2.2.2 Bartlett Window	27

2.2.3	Hanning Window	27
2.2.4	Hamming Window	28
2.2.5	Blackman Window	29
2.2.6	Kaiser Window	29
2.3	Frequency Sampling Design Techniques	31
2.3.0.1	Basic Idea	32
2.3.1	Optimum design method	33
2.3.2	Optimal Equiripple Design Technique	34
2.4	Development of the Minimax Problem	35
2.4.1	Constraint on the Number of Extrema	38
2.4.2	Theorem 1: Alternation Theorem	39
2.4.3	Parks-McClellan algorithm	39
2.4.4	MATLAB Implementation	40
<b>3.</b>	<b>EXAMPLES OF MATLAB</b>	<b>42</b>
3.1	Simple Example of FIR Filters	42
3.1.1	Fir1	42
3.1.2	Fir 2	43
3.1.3	Fir 3	44
3.2	Linear-phase FIR Filters	45
3.2.1	Hr-type1	45
3.2.2	Hr-type2	46
3.2.3	Hr-type3	48
3.2.4	Hr-Type4	49
3.3	Window Design Techniques	50
3.3.1	Hanning window	51

3.3.2	Hamming window	52
3.3.3	Blackman Window	54
3.3.4	Kaiser Window	56
3.3.5	Naive window	58
	<b>CONCLUSION</b>	61
	<b>References</b>	62
	<b>Index</b>	63

# 1. INTRODUCTION

## 1.1 Introduction to Filters:

Filtering is a process by which the frequency spectrum of signal can be modified, reshaped, or manipulated according to some desired specification. It may entail amplifying or attenuating a range of frequency components, rejecting or isolating one specific or attenuating a range of frequency component, etc. The uses of filtering are manifold, e.g., to eliminate signal contamination such as noise to remove signal distortion brought about by an imperfect transmission channel or by inaccuracies in measurement, to separate two or more distinct signals which were purposely mixed in order to maximize channel utilization, to resolve signals into their frequency components, to demodulate signals, to convert discrete-time signals into continuous-time signals, and to band-limited signals. [1].

The digital filter is a digital system that can be used to filter discrete-time signals. It can be implemented by mean of software (computer programs) or by means of dedicated hardware, and in either case it can be used to filter real-time signals or non-real-time (recorded) signals.

Software digital filters made their appearance along with the first digital computer in the late forties, although the name digital filter did not emerge until the midsixties. Early in the history of the digital computer many of the classical numerical analysis formulas of Newton, Starling, Everett, and others were used to carry out interpolation, differentiation,

and integration of function (signals) represented by mean of sequences of numbers (discrete-time signals). Since interpolation, differentiation, or integration of a signal represents a manipulation of the frequency spectrum of the signal, the subroutines or programs constructed to carry out these operations were essentially digital filters. In subsequent years, many complex and highly sophisticated algorithms and programs were developed to perform a variety of filtering tasks in numerous applications, e.g., data smoothing and prediction, pattern recognition, electrocardiogram processing, and spectrum analysis. In fact, as time goes on, interest in the software digital filter is becoming progressively more intense while its applications are increasing at an exponential rate.

A band-limited continuous-time signals can be transformed into a discrete-time signals by means of sampling. Conversely, the discrete-time signals so generated can be used to regenerate the original continuous-time signals by means of interpolation, by virtue of Shannon's sampling theorem. As a consequence, hardware Digital's filters can be used to perform real-time filtering tasks, which in the not too distant past were performed almost exclusively by analog filters. The advantages to be gained are the traditional advantages associated with digital systems in general:

1. Component tolerances are uncritical.
2. Component drift and spurious environmental signals have no influence on the system performance.
3. Accuracy is high.
4. Physical size is small.

5. Reliability is high.

A very important additional advantage of digital filters is the ease with which filter parameters can be changed in order to change the filters characteristics. This feature allows one to design programmable filters, which can be used to perform a multiplicity of filtering tasks. Also one can design new types of filters such as adaptive filters. The main disadvantage of hardware Digital's at present is their relatively high cost. However, with the tremendous advancements in the domain of large-scale integration, the cost of hardware digital filters is likely to drop drastically in the not too distant future, When this happens, hardware Digital's filters will replace analog filters in many more applications.[1,4]

## **1.2 Filter Designing Models:**

While the importance of analog filters is continuously being reduced by their digital counterparts, they remain an important study, if for no other reason than they provide a gateway to the study of digital filters. The design of a contemporary analog filter, in many cases, remains today as it was during the early days of radio. The design objective of the radio engineers was to shape the frequent-spectrum of a received or transmitted signal using modulators, demodulators, and frequency-selective filters. The frequency-selective filters were defined in terms of a mathematical ideal. The ideal models represent lowpass, high-pass, band-pass, band-stop, and all-pass filters. These are graphically interpreted in Figure 1-1. Their

shape represents the steady-state magnitude-frequency response of a filter with a transfer function of  $H(\Omega) = H(s) |_{s=j\Omega}$  where  $\Omega$  denotes an

analog frequency measured in radians per second. The mathematical specification of each ideal filter is summarized as,

$$\text{Ideal Low-pass} \quad |H(\Omega)| = \begin{cases} 1 & \text{if } \Omega \in [-B, B] \\ 0 & \text{otherwise} \end{cases} \quad (1-1)$$

$$\text{Ideal High-pass} \quad |H(\Omega)| = \begin{cases} 0 & \text{if } \Omega \in [-B, B] \\ 1 & \text{otherwise} \end{cases} \quad (1-2)$$

$$\text{Ideal Band-pass} \quad |H(\Omega)| = \begin{cases} 1 & \text{if } \Omega \in [-B_2, -B_1] \text{ or } \Omega \in [B_1, B_2] \\ 0 & \text{otherwise} \end{cases} \quad (1-3)$$

$$\text{Ideal Band-stop} \quad |H(\Omega)| = \begin{cases} 0 & \text{if } \Omega \in [-B_2, -B_1] \text{ or } \Omega \in [B_1, B_2] \\ 1 & \text{otherwise} \end{cases} \quad (1-4)$$

$$\text{All-pass} \quad |H(\Omega)| = 1 \text{ for all } \Omega \in [-\infty, \infty] \quad (1-5)$$

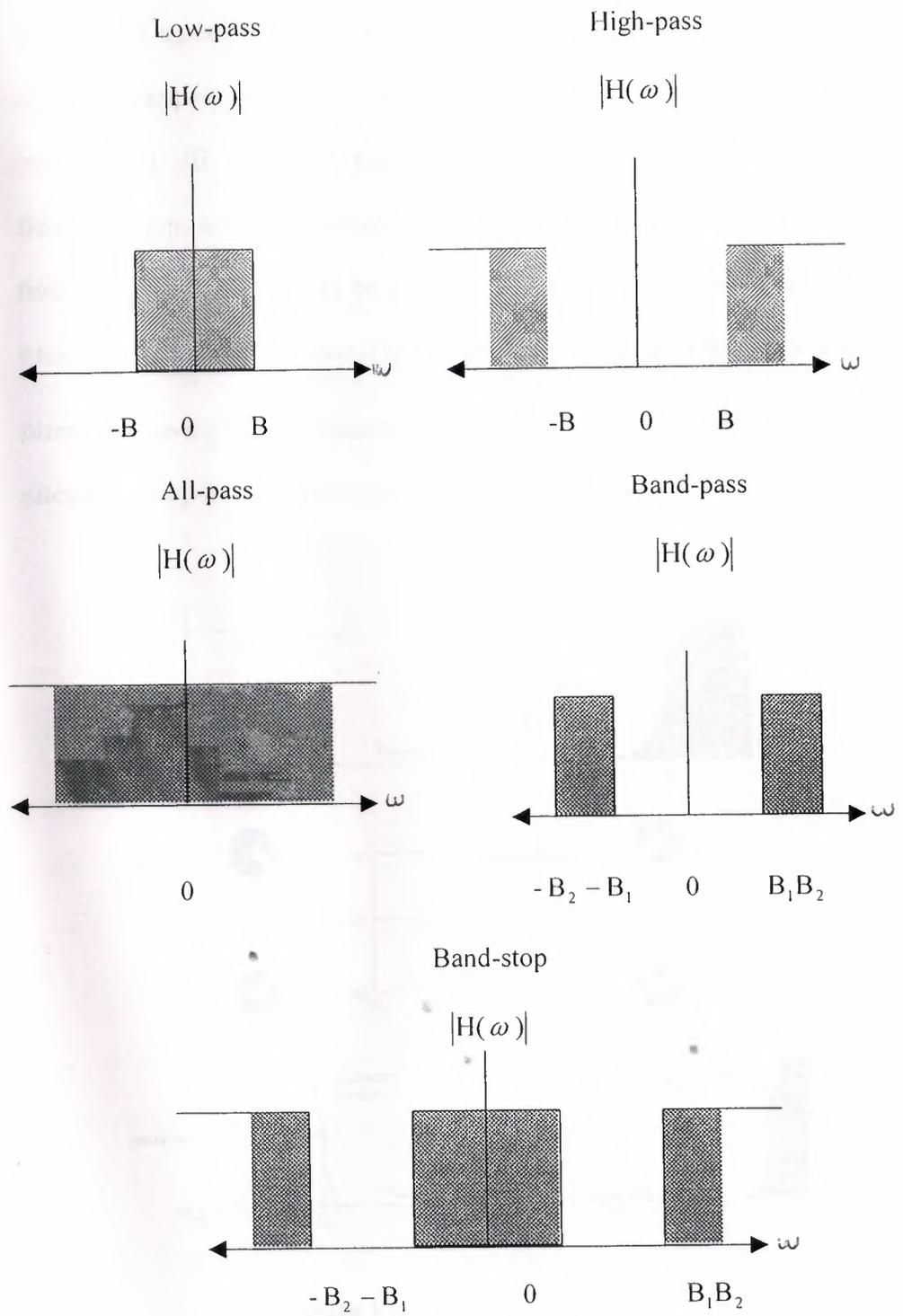


Figure 1-1 Basic ideal filter types.

To standardize the design procedure, a set of normalized analog prototype filter models was agreed upon and reduced to tables, charts, and graphs. These models, called prototypes, were all developed as low-pass systems having a known gain (typically -1 dB or -3 dB pass-band attenuation) at a known critical cut-off frequency (typically 1 radian/second). The transfer function of an analog prototype filter, denoted  $H_p(s)$ , would be encapsulated in a standard table as a function of filter type and order. The prototype filter  $H_p(s)$  would then be mapped into a final filter  $H(s)$  having critical frequencies specified by the designer. [2]. The mapping rules, frequency-frequency transform, as shown in figure.

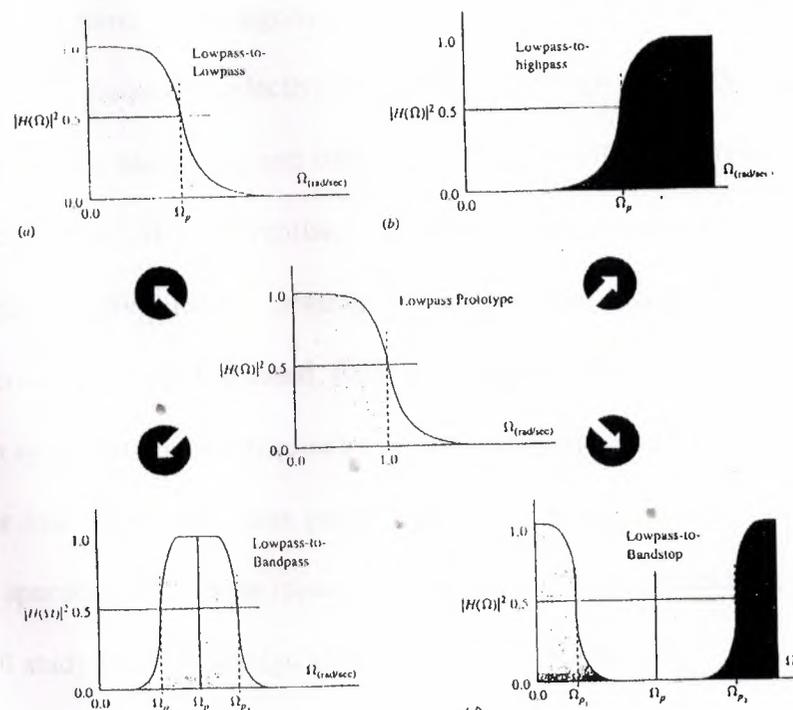


Figure 1-2. Band transformation

### 1.3 FIR and IIR Filters

We now turn our attention to the inverse problem of designing systems from the given specifications. It is an important as well as a difficult problem. In digital signal processing there are two important types of systems. The first type of systems perform signal filtering in the time domain and hence are called digital filters. The second type of systems provide signal representation in the frequency domain and are called spectrum analyzers. In this and the next chapter we will study several basic design algorithms for both FIR and IIR filters. These designs are mostly of the frequency selective type; that will design primarily multi-band low-pass, high-pass, band-pass, and band-stop filters. In FIR filter design we will also consider systems like differentiators or Hilbert transformers, which, although not frequency-selective filters, nevertheless follow the design techniques being considered. More sophisticated filter designs are based on arbitrary frequency-domain specifications and require tools that are beyond the scope of this book.

We first begin with some preliminary issues related to design philosophy and design specifications. These issues are applicable to both FIR and IIR filter designs. We will study FIR filter design algorithms in this chapter.[2,3]

#### 1.4 Preliminaries

The design of a digital filter is carried out in three steps:

- Specifications.' Before we can design a filter, we must have some specifications.

These specifications are determined by the applications.

- Approximations: Once the specifications are defined, we Use various concepts and mathematics that we studied so far to come lip with a filter description that approximates the given set of specifications. This step is the topic of filter design.
- Implementation: The product of the above step is a filter description in the form of either a difference equation, or a system function  $H(z)$ , or an impulse response  $h(n)$ . From this description we implement the filter in hardware or through software on a computer.

In this chapter we will discuss in detail only the second step, which is the conversion of specifications into a filter description.

In many applications like speech or audio signal processing, digital filters are used to implement frequency-selective operations. Therefore, specifications are required in tile frequency-domain in terms of the desired magnitude and phase response of the filter. Generally a linear phase response in the pass-band is desirable In the case of FIR filters, it is possible to have exact linear phase. In the case of IIR filters a linear phase in the pass-band is not achievable. Hence we will consider magnitude-only specifications.

The magnitude specifications are given in one of two ways. The first approach is called absolute specifications, which provide a set of requirements on the magnitude response function  $|H(e^{j\omega})|$ . These specifications are generally used for FIR filters. IIR filters are specified in a somewhat different way. The second approach is called relative specifications, which provide requirements in decibels (dB),given by

$$dB(scale) = -20 \log_{10} \frac{|H(e^{j\omega})|_{\max}}{|H(e^{j\omega})|} \geq 0$$

This approach is the most popular one in practice and is used for both FIR and IIR filters. To illustrate these specifications, we will consider a low-pass filter design as an example.

### 1.4.1 Absolute Specifications

A typical absolute specification of a low-pass filter is shown in Figure 7.1 a, n which

- Band  $[0, \omega_p]$  is called the pass-band, and  $\sigma_1$  is the tolerance (or ripple) that we are willing to accept in the ideal pass-band response,
- Band  $[\omega_p, \pi]$  is called the stop band, and  $\sigma_2$  is the corresponding tolerance (or ripple), and
- Band  $[\omega_p, \omega_s]$  is called the transition band, and there are no restrictions on the magnitude response in this band.

### 1.4.2 Relative (dB) Specifications

A typical absolute specification of a low-pass filter is shown in Figure 7.1 b, m which

- $\delta$  is the pass-band ripple in dB, and
- $A_s$  is the stop-band attenuation in dB

The parameters given in the above two specifications are obviously related. Since

$$R_p = -20 \log_{10} \frac{1 - \delta_1}{1 + \delta_1} \gg 0 (\approx 0)$$

$|H(e^{j\omega})|_{\max}$  in absolute specifications is equal to  $(1 + \sigma_1)$ , we have

and

$$A_s = -20 \log_{10} \frac{\delta_2}{1 + \delta_1} \gg 0 (\gg 1)$$

The above specifications was given for a low-pass filter. Similar specifications can also be given for other types of frequency-selective filters, such as high-pass or band-pass- However, the most important design parameters are frequency- band tolerances (or ripples) and band-edge frequencies.

Whether the given band is a pass-band or stop-band is a relatively minor issue. Therefore in describing design techniques, we will concentrate on a low-pass filter. In the next chapter we will discuss how to transform a low-pass filter into other types of frequency-selective filters. Hence it makes more sense to develop techniques for a low-pass filter so that we can compare these techniques. However, we will also provide examples of other types of filters. In light of this discussion our design goal is tile following.

### 1.4.3 Problem Statement

Design a low-pass filter (i.e., obtain its system function  $H(z)$  or its difference equation) that has a pass-band  $[0, \omega_p]$  with tolerance  $\sigma_1$  (or  $R_p$  in dB) and a stop-band  $[\omega_s, \pi]$  with tolerance  $\sigma_2$  (or  $A_s$  in dB).

In this chapter we turn our attention to the design and approximation of FIR digital filters. These filters have several design and implementational advantages.

- The phase response can be exactly linear.
- They are relatively easy to design since there are no stability problems.
- They are efficient to implement.
- The DFT can be used in their implementation.

We are generally interested in linear-phase frequency-selective FIR filters. Advantages of a linear-phase response are:

- Design problem contains only real arithmetic and not complex arithmetic;
- Linear-phase filters provide no delay distortion and only a fixed amount of delay;
- For the filter of length  $M$  (or order  $M - 1$ ) the number of operations are of the order of  $M/2$  as we discussed in the linear-phase filter implementation.

We first begin with a discussion of the properties of the linear-phase FIR filters, which are required in design algorithms. Then we will discuss three design techniques, namely the window design, the frequency Sampling design, and the optimal equiripple design techniques for linear-phase FIR filters.[2]

## 2. FIR FILTER DESIGN

### 2.1 Properties of Linear-Phase FIR Filters

In this section we discuss shapes of impulse and frequency responses and locations of system function zeros of linear-phase FIR filters. Let  $h(n)$ ,  $0 \leq n \leq M - 1$  be the impulse response of length (or duration)  $M$ . Then the system function is

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

which has  $(M - 1)$  poles at the origin  $z = 0$  (trivial poles) and  $(M - 1)$  zeros located anywhere in the  $z$ -plane. The frequency response function is

$$H(e^{j\omega}) = \sum h(n)e^{-j\omega n}, \pi < \omega < \pi$$

Now we will discuss specific requirements on the forms of  $h(n)$  and  $H(e^{j\omega})$  as well as requirements on the specific locations of  $(M - 1)$  zeros that the linear-phase constraint imposes.

#### 2.1.1 Impulse Response

We impose a linear-phase constraint

$$\angle H(e^{j\omega}) = -\alpha\omega, -\pi < \omega \leq \pi$$

where  $\alpha$  is a constant phase delay and  $h(s)$  must be symmetry that is,

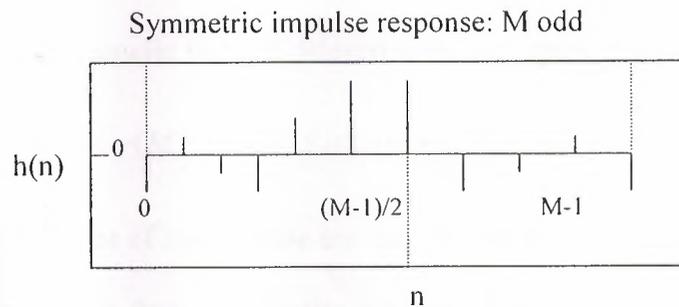
$$h(n) = h(M-1-n), \quad 0 \leq n \leq (M-1) \quad (2-1)$$

with

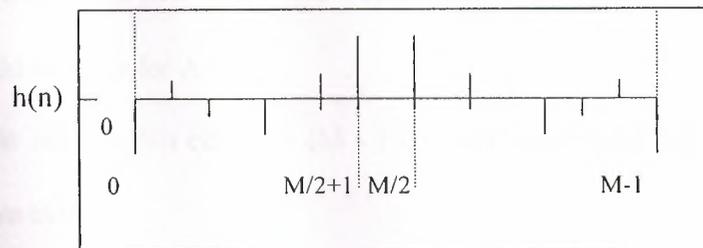
$$\alpha = \frac{M-1}{2}$$

Hence  $h(n)$  is symmetric about  $a$ , which is the index of symmetry. There are two possible types of symmetry:

- $M$  odd: In this case  $a = (M - 1)/2$  is an integer. The impulse response is as shown below.



- $M$  even: In this case  $a = (M - 1)/2$  is not an integer. The impulse response is as shown below.



Symmetric impulse response:  $M$  even

We also have a second type of “linear-phase” FIR filter if we require that the phase response

$$\frac{d\angle H(e^{j\omega})}{d\omega} = -\alpha$$

$\angle h(e^{j\omega})$  satisfy the condition.

which is a straight line but not through the origin. In this case  $a$  is not a constant phase delay, but

$$\angle H(e^{j\omega}) = \beta - \alpha\omega$$

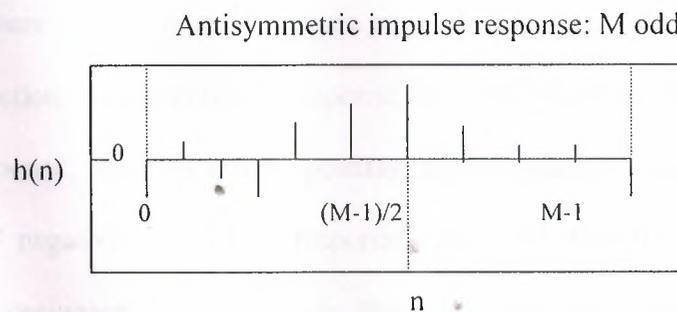
is constant, which is the group delay. Therefore  $\alpha$  is called a constant group delay. In this case, as a group, frequencies are delayed at a constant rate. But some frequencies may get delayed more and others delayed less.

$$h(n) = -(M-1-n), 0 \leq n \leq (M-1); \frac{M-1}{2}, \beta = \pm \frac{\pi}{2} \quad (2-2)$$

For this type of linear phase one can show that

This means that the impulse response  $h(n)$  is antisymmetric. The index of symmetry is still  $\alpha = (M-1)/2$ . Once again we have two possible one for  $M$  odd and one for  $M$  even

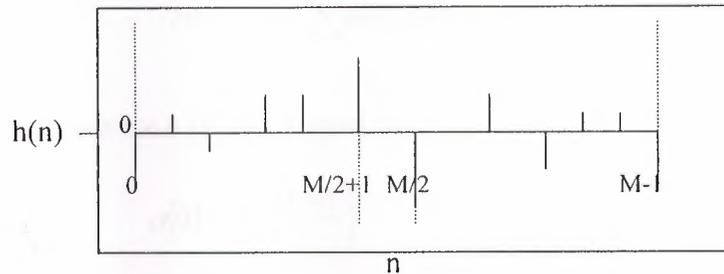
•  $M$  odd: In this case  $\alpha = (M-1)/2$  is an integer and the impulse response is as shown below



Note that the sample  $h(\alpha)$  at  $\alpha = (M-1)/2$  must necessarily be equal to zero. I.e.,

$$h((M-1)/2) = 0.$$

$M$  even; In this case  $\alpha = (M-1)/2$  is not an integer and the impulse response is as shown below.



Antisymmetric impulse response: M even

## 2.1. 2 Frequency Response

When the cases of symmetry and antisymmetry are combined with odd and even M, we obtain four types of linear-phase FIR filters. Frequency response functions for each of these types have some peculiar expressions and shapes. To study these responses, we write  $H(e^{j\omega})$  as

$$H(e^{j\omega}) = Hr(\omega)e^{j(\beta-\alpha\omega)}; \beta = \pm \frac{\pi}{2}, \alpha = \frac{M-1}{2} \quad (2-3)$$

Where  $Hr(\omega)$  is an amplitude response function and not a magnitude response function. The amplitude response is a real function, but unlike the magnitude response, which is always positive, the amplitude response may be both positive and negative. The phase response associated with the magnitude response is a discontinuous function, while that associated with the amplitude response is a continuous linear function, To illustrate the difference between these two types of responses.[2,4,8]

### 2.1.2.1 Type-1 Linear-Phase FIR filters: Symmetrical impulse response, M odd

In this case  $\beta = 0$ ,  $\alpha = (M-1)/2$  is an integer, and  $h(n) = h(M-1-n)$ ,  $0 \leq n \leq M-1$ .

$$H(e^{j\omega}) = \left[ \sum a(n) \cos \omega n \right] e^{-j\omega(M-1)/2} \quad (2-4)$$

Where sequence  $a(n)$  is obtained from  $h(n)$  as

$$a(0) = h\left(\frac{M-1}{2}\right) \quad (2-5a)$$

The middle sample is

$$a(n) = 2h\left(\frac{M-1}{2} - n\right), 1 \leq n \leq \frac{M-3}{2} \quad (2-5b)$$

Comparing (2-3) and (2-4), we have

$$Hr(\omega) = \sum_{n=0}^{(M-1)/2} a(n) \cos \omega n \quad (2-6)$$

### 2.1.2.2 Type-2 linear-phase FIR filter: Symmetrical impulse response, $M$ even

In this case again  $\beta = 0$ ,  $h(n) = h(M-1-n)$ ,  $0 \leq n \leq M-1$ , but  $\alpha = (M-1)/2$  is not an integer. Then we can show that

$$H(e^{j\omega}) = \left[ \sum b(n) \cos \left\{ \omega \left( n - \frac{1}{2} \right) \right\} \right] e^{-j\omega(M-1)/2} \quad (2-7)$$

Where

$$b(n) = 2h\left(\frac{M}{2} - n\right), \quad n = 1, 2, \dots, \frac{M}{2} \quad (2-8)$$

Hence

$$Hr(\omega) = \sum b(n) \cos \left\{ \omega \left( n - \frac{1}{2} \right) \right\} \quad (2-9)$$

Note: At  $\omega = \pi$  we get

$$Hr(\pi) = \sum_{n=1}^{M/2} b(n) \cos\left\{\pi\left(n - \frac{1}{2}\right)\right\} = 0$$

regardless of  $b(n)$  or  $h(n)$ . Hence we cannot use this type (i.e., symmetric  $h(n)$ ,  $M$  even) for high-pass or band-stop filters.

### 2.1.2.3 Type-3 linear-phase FIR filter: Antisymmetric impulse response, $M$ odd

In this case  $\beta = \pi/2$ ,  $a = (M - 1)/2$  is an integer,  $h(n) = -h(M-1-n)$ ,  $0 \leq n \leq M-1$ , and  $h((M - 1)/2) = 0$ . Then

$$H(e^{j\omega}) = \left[ \sum c(n) \sin \omega n \right] e^{j\left[\frac{\pi}{2} - \left(\frac{M-1}{2}\right)\omega\right]} \quad (2-10)$$

Where

$$c(n) = 2h\left[\frac{M-1}{2} - n\right], \quad n=1,2,\dots,\frac{M-1}{2} \quad (2-11)$$

And

$$Hr(\omega) = \sum_{n=1}^{(M-1)/2} c(n) \sin \omega n \quad (2-12)$$

**Note:** At  $\omega = 0$  and  $\omega = \pi$  we have  $Hr(\omega) = 0$ , regardless of  $c(n)$  or  $h(n)$ . Furthermore,  $e^{j\pi/2} = j$ , which means that  $jHr(\omega)$  is purely imaginary. Hence this type of filter is not suitable for designing a low-pass filter or a high-pass filter. However, this behavior is suitable for approximating ideal digital Hilbert

transformers and differentiators. An ideal Hilbert transformer [19] is an all-pass filter that imparts a 90° phase shift on the input signal it is frequently used in communication systems for modulation purposes. Differentiators are used in many analog and digital systems to take the derivative of a signal.[3,4,8]

#### 2.1.2.4 Type-4 linear-phase FIR filter: Antisymmetric impulse response, M even

This case is similar to Type-2.

$$H(e^{j\omega}) = \left[ \sum d(n) \sin \left\{ \omega \left( n - \frac{1}{2} \right) \right\} \right] e^{-j(M-1)/2} \quad (2-13)$$

where

$$d(n) = 2h \left( \frac{M}{2} - n \right), \quad n=1,2,\dots,\frac{M}{2} \quad (2-14)$$

And

$$Hr(\omega) = \sum d(n) \sin \left\{ \omega \left( n - \frac{1}{2} \right) \right\} \quad (2-15)$$

Note: At  $\omega = 0$ ,  $Hr(0) = 0$  and  $e^{j\pi} = -1$ . Hence this type is also suitable for designing digital Hilbert transformers and differentiators

#### 2.1.3 Zero Locations

For a FIR filter there are  $(M - 1)$  (trivial) poles at the origin and  $(M - 1)$  zeros located somewhere in the z-plane. For linear-phase FIR filters, these zeros possess certain symmetries that are due to the symmetry constraints on  $h(n)$ . It can be shown that if  $H(z)$  has a zero at

$$z = z_1 = re^{j\theta}$$

then for linear phase there must be a zero at

$$z = \frac{1}{z_1} = \frac{1}{r} e^{-j\theta}$$

For a real-valued filter we also know that if  $z_1$  is complex, then there must be a conjugate zero at  $z_1^* = re^{-j\theta}$  which implies that there must be a zero at  $1/z_1^* = (1/r)e^{j\theta}$ . Thus a general zero constellation is a quadruplet

$$re^{j\theta}, \quad (1/r)e^{j\theta}, \quad re^{-j\theta}, \quad \text{and} \quad (1/r)e^{-j\theta}$$

If  $r = 1$  then  $1/r = 1$ , and hence the zeros are on the unit circle and occur in pairs

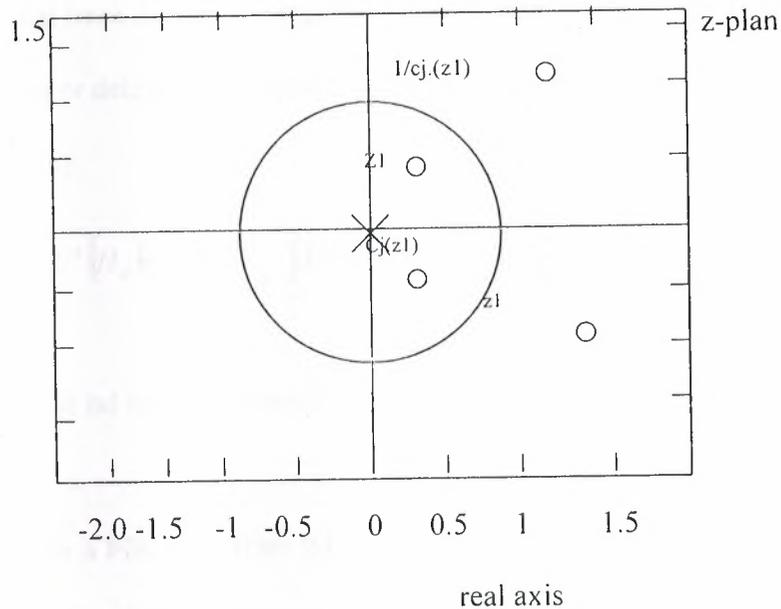
$$e^{j\theta} \quad \text{and} \quad e^{-j\theta}$$

If  $\theta = 0$  or  $\theta = \pi$  then the zeros are on the real line and occur in pairs

$$r \quad \text{and} \quad 1/r$$

Finally, if  $r = 1$  and  $\theta = 0$  or  $\theta = \pi$ , the zeros are either at  $z = 1$  or  $z = -1$ . These symmetries can be used to implement cascade forms with linear-phase sections.

Pole-Zero Plot



## 2.2 WINDOW DESIGN TECHNIQUES

The basic idea behind the window design is to choose a proper ideal frequency-selective filter (which always has a noncausal, infinite-duration impulse response) and then truncate (or window) its impulse response to obtain a linear-phase and causal FIR filter. Therefore the emphasis in this method is on selecting an appropriate windowing function and an appropriate ideal filter. We will denote an ideal frequency-selective filter by  $H_d(e^{j\omega})$ , which has a unity magnitude gain and linear-phase characteristics over its pass-band and zero response over its stop-band. An ideal LPF of bandwidth  $\omega_c < \pi$  is given by

$$h_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases} \quad (2-16)$$

Where  $\omega_c$  is also called the cutoff frequency and  $\alpha$  is called the sample delay

(note that from the DTFT properties,  $e^{j\alpha\omega}$  implies shift in the positive  $n$  direction or delay). The impulse response of this filter is of infinite duration and is given by

$$h_d(n) = L^{-1} [H_d(e^{j\omega})] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} 1 \cdot e^{-j\alpha\omega} e^{j\omega n} d\omega = \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)} \quad (2-17)$$

Note: that  $h_d(n)$  is symmetric with respect to  $\alpha$ , a fact useful for linear-phase FIR filters.

To obtain a FIR filter from  $h_d(n)$ , one has to truncate  $h_d(n)$  on both sides. To obtain a causal and linear-phase FIR filter  $h(n)$  of length  $M$ , we must have

$$h(n) = \begin{cases} h_d(n), & 0 \leq n \leq M-1 \\ 0, & \text{elsewhere} \end{cases} \quad \text{and} \quad a = \frac{M-1}{2} \quad (2-18)$$

This operation is called "windowing". In general,  $h(n)$  can be thought of as being formed by the product of  $h_d(n)$  and a window function  $\omega(n)$  as follows:

$$h(n) = h_d(n)\omega(n) \quad (2-19)$$

Where

$$\omega(n) = \begin{cases} \text{Some symmetric function with respect to} \\ \alpha \text{ over } 0 \leq n \leq M-1 \\ 0, \text{ otherwise} \end{cases}$$

Depending on how we define  $\omega(n)$  above, we obtain different window designs-  
For example, in (2-17) above

$$w(n) = \begin{cases} 1, & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} = R_M(n)$$

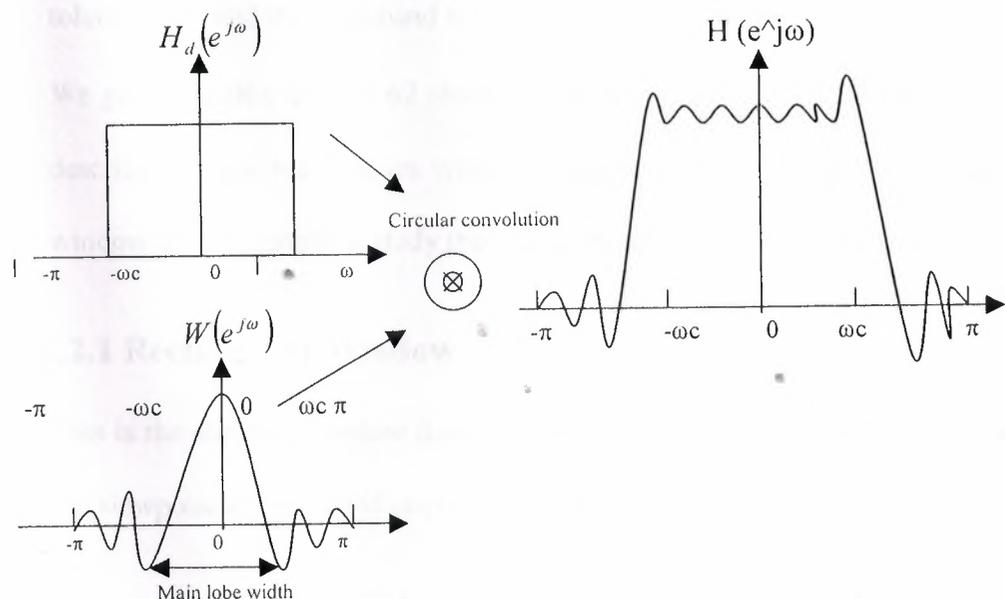
which is the rectangular window defined earlier.

In the frequency domain the causal FIR filter response  $H(e^{j\omega})$  is given by the periodic convolution of  $H_d(e^{j\omega})$  and the window response that is,

$$H(e^{j\omega}) = H_d(e^{j\omega}) \otimes W(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(e^{j\lambda}) H_d(e^{j(\omega-\lambda)}) d\lambda \quad (2-20)$$

This is shown pictorially in Figure 2-1 for a typical window response, from which we have the following observations:

1. Since the window  $w(n)$  has a finite length equal to  $M$ , its response has a peaky main lobe whose width is proportional to  $1/M$ , and has side lobes of smaller heights.



**Figure (2-1)** Windowing operation in the frequency domain

2. The periodic convolution (2-19) process a smeared version of the ideal response  $H_d(e^{j\omega})$ .
3. The main lobe produces a transition band in  $h(e^{j\omega})$  whose width is responsible for the transition width. This width is then proportional to  $1/M$ . The wider the main lobe, the wider will be the transition width.
4. The side lobes produce ripples that have similar shapes in both the pass-band and stop-band. [5,7]

### 2.2.0.1 Basic Window Design Idea

For the given filter specifications choose the filter length  $M$  and a window function  $w(n)$  for the narrowest main lobe width and the smallest side lobe attenuation possible. From observation 4 above we note that the pass-band tolerance  $\sigma_1$  and the stop-band tolerance  $\sigma_2$  cannot be specified independently. We generally take care of  $\sigma_2$  alone, which results in  $\sigma_2 = \sigma_1$ . We now briefly describe various well-known window functions. We will use the rectangular window as an example to study their performances in the frequency domain.[6]

### 2.2.1 Rectangular Window

This is the simplest window function but provides the worst performance from the viewpoint of stop-band attenuation. It is defined as:

$$w(n) = \begin{cases} 1, & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} \quad (2-21)$$

Its frequency response function is

$$W(e^{j\omega}) = \left[ \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right] e^{-j\omega \frac{M-1}{2}} \Rightarrow W_r(\omega) = \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

which is the amplitude response. From (2-20) the actual amplitude response  $H_r(\omega)$  is given by

$$H_r(\omega) \cong \frac{1}{2\pi} \int_{-\pi}^{\omega+\omega_c} W_r(\lambda) d\lambda = \frac{1}{2\pi} \int_{-\pi}^{\omega+\omega_c} \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} d\lambda, \quad M \gg 1 \quad (2-22)$$

This implies that the running integral of the window amplitude response (or accumulated amplitude response) is necessary in the accurate analysis of the transition bandwidth and the stop-band attenuation. Figure (2-2) shows the rectangular window function  $w(n)$ , its amplitude response  $W(\omega)$ , the amplitude response in dB, and the accumulated amplitude response (2-22) in dB. From the observation of plots in Figure (2-2) we can make several observations.

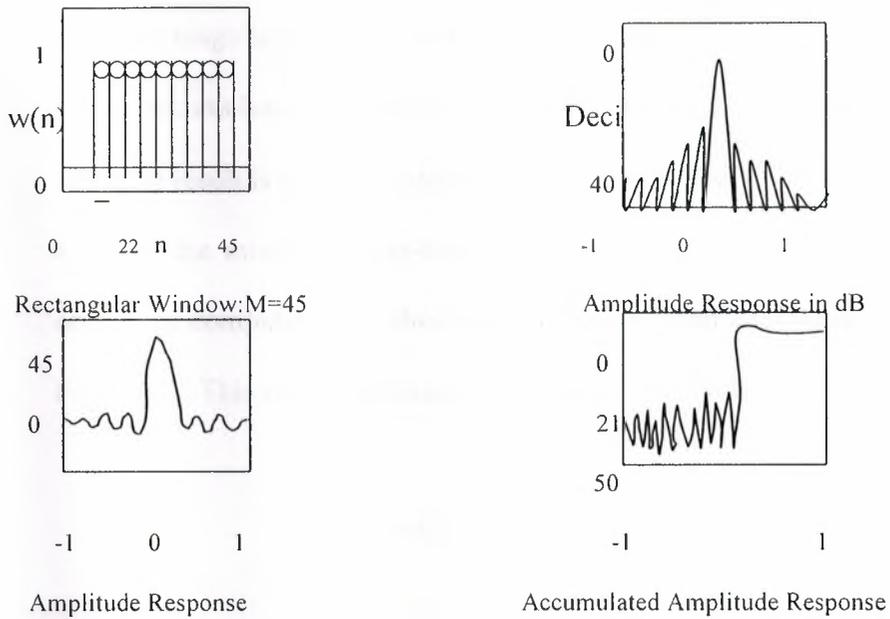


Figure (2-2) Rectangular windowing:  $M=45$

1. The amplitude response  $W_r(\omega)$  has the first zero at  $\omega = \omega_1$  where

$$\frac{\omega_1 M}{2} = \pi, \text{ or, } \omega_1 = \frac{2\pi}{M}$$

Hence the width of the main lobe is  $2\omega = 4\lambda/M$ . Therefore the approximated transition bandwidth is  $4\lambda/M$ .

2. The magnitude of the first side lobe is approximately  $\omega = 3\lambda/M$  and given by

$$\left| W_r\left(\omega = \frac{3\pi}{M}\right) \right| = \frac{\left| \sin\left(\frac{3\pi}{2}\right) \right|}{\left| \sin\left(\frac{3\pi}{2M}\right) \right|} \cong \frac{2M}{3\pi} \dots \dots \dots \text{for } M \gg 1$$

Comparing this with the main lobe amplitude, which is equal to  $M$ , the peak side lobe magnitude is of the main lobe amplitude.

3. The accumulated amplitude response has the first side lobe magnitude at 21 dB. The result is minimum stop-band attenuation 21dB.

4. Using the minimum stop-band attenuation, the transition bandwidth can be accurately computed. It is shown in the accumulated amplitude response plot in Figure 2-2. This computed exact transition bandwidth is

$$\omega_s - \omega_p = \frac{1.8\pi}{M}$$

which is about half the approximate bandwidth of  $4\lambda/M$ .

Clearly, this is a simple window operation in the time domain and an easy function to analyze in the frequency domain. However, there are two main problems. First, the minimum stop-band attenuation of 21 dB is insufficient in practical applications. Second, the rectangular windowing being a direct truncation of the infinite length  $hd(n)$ , it suffers from the Gibbs phenomenon. If we increase  $M$ , the width of each side lobe will decrease, but the area under each lobe will remain constant. Therefore the relative amplitudes of side lobes will remain constant, and the minimum stop-band attenuation will remain at 21 dB. This implies that air ripples will bunch up near the band edges.

Since the rectangular window is impractical in many applications, we consider other window functions, many of which bear the names of the people who first proposed them. Although these window functions can also be analyzed similar

to rectangular window.[2]

### 2.2.2 Bartlett Window

Since the Gibbs phenomenon results from the fact that the rectangular window has a sudden transition from 0 to 1 (or 1 to 0), Bartlett suggested a more gradual transition in the form of a triangular window, which is given by

$$\omega(n) = \begin{cases} \frac{2n}{M-1}, & 0 \leq n \leq M-1 \\ 2 - \frac{2n}{M-1}, & \frac{M-1}{2} \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} \quad (2-23)$$

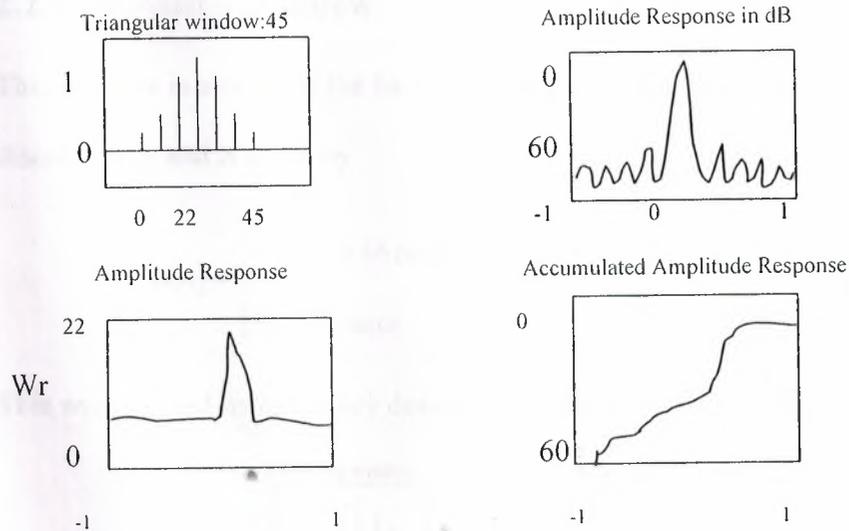


Figure (2-3) bartlett window

### 2.2.3 Hanning Window

This is raised cosine window function given by

$$\omega(n) = \begin{cases} 0.5 \left[ 1 - \cos\left(\frac{2\pi n}{M-1}\right) \right], & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} \quad (2-24)$$

This window and its frequency domain response are shown in figure (2-4)

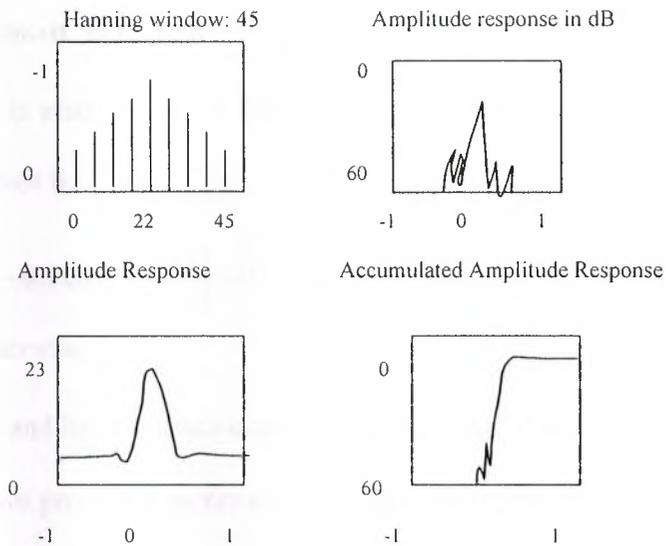


Figure (2-4) Hanning window

### 2.2.4 Hamming Window

This window is similar to the hanning window except that it has a small amount of discontinuity and is given by

$$\omega(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right), & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} \quad (2-25)$$

This window and its frequency domain response are shown in figure 2-5.

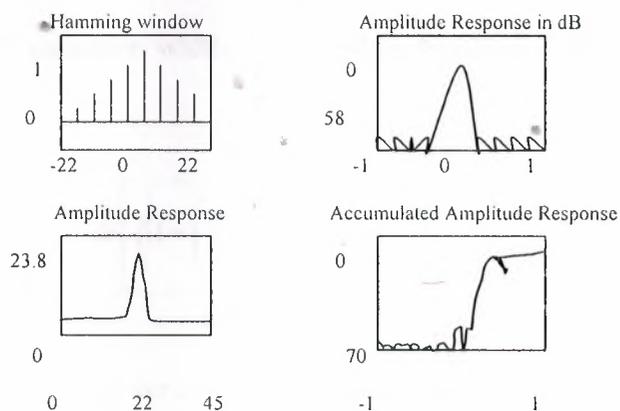


Figure (2-5) hamming window

### 2.2.5 Blackman Window

This window is also similar to the previous two but contains a second harmonic term and is given by

$$\omega(n) = \begin{cases} 0.42 - 0.5 \cos\left(\frac{2\pi n}{M-1}\right) + 0.08 \cos\left(\frac{4\pi n}{M-1}\right), & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} \quad (2-26)$$

This window and its frequency-domain responses are shown in Figure

In Table 7.1 we provide a summary of window function characteristics in terms of their transition widths (as a function of M) and their minimum stop-band attenuation in dB. Both the approximate as well as the exact transition bandwidths are given. Note that the transition widths and the stop-band attenuation increase as we go down the table. The Hamming window appears to be the best choice for many applications.[2,6,8]

### 2.2.6 Kaiser Window

This is one of the most useful and optimum windows. It is optimum in the sense of providing a large main lobe width for the given stop-band (1 attenuation, which implies the sharpest transition width. The window function is due to J. F. Kaiser and is given by

$$\omega(n) = \frac{I_0 \left[ \beta \sqrt{1 - \left(1 - \frac{2n}{M-1}\right)^2} \right]}{I_0[\beta]} \quad (2-27)$$

Where  $I_0[\cdot]$  is the modified zero-order Bessel function, and  $\beta$  is a parameter that depends on M and that can be chosen to yield various transition widths and near-

optimum stop-band attenuation. This window can provide different transition widths for the same M, which is something other window lack. For example,

•If  $\beta = 5.658$ , then the transition width is equal to  $7.8\pi/M$ , and the minimum stop-band attenuation is equal to 60 dB. This is shown in Figure 7.15.

If  $\beta = 4.588$ , then the transition width is equal to  $5.8\pi/M$ , and the minimum stop-band attenuation is equal to 50 dB.

Hence the performance of this window is comparable to that of the Hamming window. In addition, the Kaiser window provides flexible transition bandwidths.

Due to the complexity involved in the Bessel functions. The design equations for this window are not easy to derive. Fortunately,[2,8]

Kaiser has developed empirical design equations, which we provide below with out proof.

### Design Equations

Given  $\omega_p, \omega_s, R_p$ , and  $A_s$

Norm. transition width =

$$\Delta f = \frac{\omega_s - \omega_p}{2\pi}$$

Filter order

$$M \approx \frac{A_s - 7.95}{14.36\Delta f} + 1$$

## 2.3 FREQUENCY SAMPLING DESIGN TECHNIQUES

In this design approach we use the fact that the system function  $h(z)$  can be obtained from the samples  $H(k)$  of the frequency response  $H(e^{j\omega})$ . Furthermore, this design technique fits nicely with the frequency sampling structure. Let  $h(n)$  be the impulse response of an  $M$ -point FIR filter,  $H(k)$  be its  $M$ -point DFT, and  $h(z)$  be its system function. Then we have

$$H(z) = \sum_{n=0}^{M-1} h(n)z^{-n} = \frac{1-z^{-M}}{M} \sum_{k=0}^{M-1} \frac{H(k)}{1-z^{-1}e^{j2\pi k/M}} \quad (2-28)$$

And

$$H(e^{j\omega}) = \frac{1-e^{-j\omega M}}{M} \sum_{k=0}^{M-1} \frac{H(k)}{1-e^{-j\omega}e^{j2\pi k/M}} \quad (2-29)$$

With

$$H(k) = H(e^{j2\pi k/M}) = \begin{cases} H(0), & k = 0 \\ H^*(M-k), & k = 1, 2, \dots, M-1 \end{cases}$$

For a linear-phase FIR filter we have

$$h(n) = \pm h(M-1-n), \quad n=0, 1, 2, \dots, M-1$$

Where the positive sign is for the Type-1 and Type-2 linear-phase filters, while the negative sign is for Type-3 and Type-4 linear-phase filters. Then  $H(k)$  is given by

$$H(k) = H_r \left( \frac{2\pi k}{M} \right) e^{j\angle H(k)} \quad (2-30)$$

where

$$H_r\left(\frac{2\pi k}{M}\right) = \begin{cases} H_r(0), & k = 0 \\ H_r\left(\frac{2\pi(M-k)}{M}\right), & k = 1, \dots, M-1 \end{cases} \quad (2-31)$$

And

$$\angle H_r(k) = \begin{cases} -\left(\frac{M-1}{2}\right)\left(\frac{2\pi k}{M}\right), & k = 0, \dots, \left[\frac{M-1}{2}\right] \\ +\left(\frac{M-1}{2}\right)\frac{2\pi}{M}(M-k), & k = \left[\frac{M-1}{2}\right] + 1, \dots, M-1 \end{cases}, (\text{type-1\&2}) \quad (2-32)$$

$$\angle H(k) = \begin{cases} \left(\pm\frac{\pi}{2}\right) - \left(\frac{M-1}{2}\right)\left(\frac{2\pi k}{M}\right), & k = 0, \dots, \left[\frac{M-1}{2}\right] \\ -\left(\pm\frac{\pi}{2}\right) + \left(\frac{M-1}{2}\right)\frac{2\pi}{M}(M-k), & k = \left[\frac{M-1}{2}\right] + 1, \dots, M-1 \end{cases}, (\text{Type-3\&4}) \quad (2-33)$$

Finally we have

$$h(n) = \text{IDFT}[H(k)] \quad (2-34)$$

Note that several textbooks (e.g., [19, 20, 16]) provide explicit formulas to compute  $h(n)$  given  $H(k)$ . We will use MATLAB's `ifft` routine to compute  $h(n)$ .

### 2.3.0.1 Basic Idea

Given the ideal low-pass filter  $H_d(e^{j\omega})$ , choose the filter length  $M$  and then sample  $H_d(e^{j\omega})$  at  $M$  equispaced frequencies between  $0$  and  $2\pi$ . The actual response  $H(e^{j\omega})$  is the interpolation of the samples  $H(k)$ . This is shown in Figure 2-5. The impulse response is given by (2-34). Similar steps apply to other frequency-selective filters. Furthermore, this idea can also be extended for approximating arbitrary frequency-domain specifications

From Figure 2-5. We observe the following.

1. The approximation error—that is, the difference between the ideal and the actual response is zero at the sampled frequencies.
2. The approximation error at other frequencies depends on the shape of the ideal response; that is, the sharper the ideal response the larger the approximation error.
3. The error is larger near the band edges and smaller within the band.

There are two design approaches. In the first approach we use the basic idea literally and provide no constraints on the approximation error:

that is, we accept whatever error we get from the design. This approach is called a naive design method. In the second approach we try to minimize error in the stop-band by varying the values of the transition band samples. It results in a much better design called an optimized design method.[2,3]

### **2.3.1 Optimum design method**

To obtain more attenuation, we will have to increase  $M$  and make the transition band samples free samples—that is, we vary their values to obtain the largest attenuation for the given  $M$  and the transition width. This problem is known as an optimization problem, and it is solved using linear programming techniques.

Clearly, this method is superior in that by varying one sample we can get a much better design. In practice the transition bandwidth is generally small containing either one or two samples. Hence we need to optimize at most two samples to obtain the largest stop-band attenuation. This is also equivalent to minimizing the maximum side lobe magnitudes in the absolute sense. Hence this optimization problem is also called a minimax problem. This problem can be solved in MATLAB, but it would require the use of Optimization toolbox.

### 2.3.2 OPTIMAL EQUI RIPPLE DESIGN TECHNIQUE:

-The last two techniques-namely, the window design and the frequency sampling design were easy to understand and implement. However, they have some disadvantages. First, we cannot specify the band frequency  $\omega_s$  and  $\omega_p$  precisely in the design; that is, we have to accept whatever values we obtain after the design. Second, we cannot specify both  $\sigma_1$  and  $\sigma_2$  and ripple factors simultaneously. Either we have  $\sigma_1 = \sigma_2$  in the window design method, or we can optimize  $\sigma_2$  only in the frequency sampling method. Finally, the approximation error that is, the difference between the ideal response and the actual response is not uniformly distributed over the band intervals. It is higher near the band edges and smaller in the regions away from band edges. By distributing the error uniformly, we can obtain a lower-order filter satisfying the same specifications. Fortunately, a technique exists that can eliminate the above three problems.

This technique is somewhat difficult to understand and requires a computer for its implementation.

For linear phase FIR filters it is possible to design a set of equation for which it can be proved that the design solution is optimal in the sense of minimizing the maximum approximation error (sometimes called the minimum or the Chebyshev error). Filters that have this property are called EQUI RIPPLE filters because the approximation error is uniformly distributed in both the pass-band and the stop-band. This results in lower-order filters

In the following we first formulate a minimax optimal FIR design problem and discuss the total number of maxima and minima (collectively called extrema) that

one can obtain in the amplitude response of a linear-phase FIR filter. Using this, we then discuss a general equiripple FIR filter design algorithm, which uses polynomial interpolation for its solution. This algorithm is known as the ParksMcClellan algorithm, and it incorporates the Remez exchange routine for polynomial solution. This algorithm is available as a subroutine on many computing platforms. In this section we will use MATLAB to design equiripple FIR filters.[1,4,6]

## 2.4 DEVELOPMENT OF THE MINIMAX PROBLEM

The four cases of linear-phase FIR filters can be written in the form

$$H(e^{j\omega}) = e^{j\beta} e^{-j\frac{M-1}{2}\omega} H_r(\omega)$$

where the values for  $\beta$  and the expressions for  $H_r(\omega)$  are given in Table

Linear-phase FIR Filter type	$\beta$	$H_r(e^{j\omega})$
Type-1: M odd, symmetric h(n)	0	$\sum_0^{(M-1)/2} a(n) \cos \omega n$
Type-2: M even, symmetric h(n)	0	$\sum_1^{M/2} b(n) \cos[\omega(n-1/2)]$
Type-3: M odd, antisymmetric h(n)	$\pi/2$	$\sum_1^{(M-1)/2} c(n) \sin \omega n$
Type-4: M even, antisymmetric h(n)	$\pi/2$	$\sum_1^{M/2} d(n) \sin[\omega(n-1/2)]$

**Table 2-1** Amplitude response and  $\beta$ -values for linear-phase FIR filters.

LP FIR Filter Type	$Q(\omega)$	$L$	$P(\omega)$
Type-1	1	$\frac{M-1}{2}$	$\sum_0^L a(n) \cos \omega n$
Type-2	$\cos \frac{\omega}{2}$	$\frac{M}{2} - 1$	$\sum_0^L \tilde{b}(n) \cos \omega n$
Type-3	$\sin \omega$	$\frac{M-1}{2}$	$\sum_0^L \tilde{c}(n) \cos \omega n$
Type-4	$\sin \frac{\omega}{2}$	$\frac{M}{2} - 1$	$\sum_0^L \tilde{d}(n) \cos \omega n$

**Table (2-2)**  $Q(\omega)$ , and  $P(\omega)$  for linear-phase FIR filters

Using simple trigonometric identities, each expression for  $H_r(\omega)$  above can be written as a product of a fixed function of  $\omega$  (call this  $Q(\omega)$ ) and a function that is a sum of cosines (call this  $P(\omega)$ ). Thus

$$H_r(\omega) = Q(\omega)P(\omega) \quad (2-35)$$

Where  $P(\omega)$  is of the form

$$P(\omega) = \sum_{n=0}^L a(n) \cos \omega n \quad (2-36)$$

And  $Q(\omega)$ ,  $L$ ,  $P(\omega)$  for the four cases are given in Table 7.3.

The purpose of this analysis is to have a common form for  $H_r(\omega)$  across all four cases- It makes the problem formulation much easier. To formulate our problem as a Chebyshev approximation problem, we have to define the desired amplitude response  $H_r(\omega)$  and a weighting function  $W(\omega)$ , both defined over pass-bands and stop-bands. The weighting function is necessary so that

we can have an independent control over  $\delta_1$  and  $\delta_2$ - the weighted error is defined

as

$$E(\omega) \triangleq W(\omega)[H_{dr}(\omega) - H_r(\omega)], \quad \omega \in S \triangleq [0, \omega_p] \cup [\omega_s, \pi] \quad (2-37)$$

These concepts are made clear in the following set of figures. It shows a typical equiripple filter response along with its ideal response.

Thus the maximum error in both the pass-band and stop-band is  $\delta_2$ . Therefore if we succeed in minimizing the maximum weighted error to  $\delta_2$  we automatically also satisfy the specification in the pass-band to  $\delta_1$ . Substituting  $H_r(\omega)$  from (2-34) into (2-36), we obtain

If we define

$$\begin{aligned} E(\omega) &= W(\omega)[H_{dr}(\omega) - Q(\omega)P(\omega)] \\ &= W(\omega)Q(\omega) \left[ \frac{H_{dr}(\omega)}{Q(\omega)} - P(\omega) \right], \quad \omega \in S \end{aligned}$$

and

$$\hat{W}(\omega) \triangleq W(\omega)Q(\omega)$$

then we obtain

$$\hat{H}_{dr}(\omega) \triangleq \frac{H_{dr}(\omega)}{Q(\omega)}$$

$$E(\omega) = \hat{W}(\omega) [\hat{H}_{dr}(\omega) - P(\omega)], \quad \omega \in S$$

Thus we have a common form of  $E(\omega)$  for all four cases.

Determine the set of coefficients  $a(n)$  or  $b(n)$  or  $c(n)$  or  $d(n)$  [ or equivalently  $a(n)$  or  $b(m)$  or  $c(n)$  or  $4d(n)$ ] to minimize the maximum absolute value of  $E(\omega)$  over the pass-band and stop-band, i.e.,

$$\min_{\text{over...coeff.}} \left[ \max_{\omega \in S} |E(\omega)| \right]$$

Now we have succeeded in specifying the exact  $\omega_p, \omega_s, \delta_1$  and  $\delta_2$ . In addition the error can now be distributed uniformly in both the pass-band and stop-band.

### 2.4.1 CONSTRAINT ON THE NUMBER OF EXTREMA

Before we give the solution to the above problem, we will first discuss the issue: how many local maxima and minima exist in the error function  $E(\omega)$  for a given  $M$ -point filter? This information is used by the Parks McClellan algorithm to obtain the polynomial interpolation. The answer is in the expression  $P(\omega)$  From (7.41)

$$\cos(2\omega) = 2 \cos^2(\omega) - 1$$

$$\cos(3\omega) = 4 \cos^3(\omega) - 3 \cos(\omega)$$

$P(\omega)$  is a trigonometric function in  $\omega$ . Using trigonometric identities of the form  $P(\omega)$  can be converted to a trigonometric polynomial in  $\cos(\omega)$ , which we can write (2-35) as

$$P(\omega) = \sum_{n=0}^L \beta(n) \cos^n \omega$$

From (7-46) we note that  $P(W)$  is an  $L$ th-order polynomial in  $\cos(w)$ . Since  $\cos(w)$  is a monotone function in the open interval  $0 < w < \pi$ , then it follows that the  $L$ th-order polynomial  $P(w)$  in  $\cos(w)$  should behave like an ordinary  $L$ th-order polynomial  $P(x)$  in  $x$ . Therefore  $P(w)$  has at most (I-e., no more than)  $(L - 1)$  local extrema in the open interval  $0 < w < \pi$ .

Conclusion: The error function  $E(w)$  has at most  $(L + 3)$  extrema in  $S$ .

### 2.4.2 Theorem 1: Alternation Theorem

Let  $S$  is any closed subset of the closed interval  $[0, \pi]$ . In order that  $P(w)$  be the unique minimax approximation to  $H_d(\omega)$  on  $S$ , it is necessary and sufficient that the error function  $E(w)$  exhibit at least  $(L + 2)$  alternations" or external frequencies in  $S$ ; that is, there must exist  $(L + 2)$  frequencies  $w$  in  $S$  such that

$$E(\omega_i) = -E(\omega_{i-1}) = \pm \max_S |E(\omega)|$$

$$\stackrel{\Delta}{=} \pm \delta, \forall \omega_0 < \omega_1 < \dots < \omega_{L+1} \in S$$

Combing this theorem with our earlier conclusion, we infer that the optimal equiripple filter has either  $(L - 2)$  or  $(L - 3)$  alternations.

### 2.4.3 Parks-McClellan algorithm

The Parks-McClellan algorithm begins by guessing  $(L + 2)$  external frequencies  $\{\omega_i\}$  and estimating the maximum error  $\delta$  at these frequencies. It then fits  $L$ th-order polynomial (7.46) through points given in (7.47). Local maximum errors are

determined over a finer grid, and the external frequencies  $\{W_i\}$  are adjusted at this new external value. A new  $L$ th order polynomial is fit through these new frequencies, and the procedure is repeated. This iteration continues until the optimum set  $\{W_i\}$  and the global maximum error  $\delta$  are found. The iterative procedure is guaranteed to converge, yielding the polynomial  $P(w)$  from (7-46) coefficients  $\beta(n)$  are determined. Finally, the coefficients  $\beta(n)$  as well as the impulse response  $h(n)$  are computed. This algorithm is available in MATLAB as the `remez` function, which is described below.

Since we approximated  $M$ , the maximum error  $\delta$  may not be equal to  $\delta_2$ . If this is the case, then we have to increase  $M$  (if  $\delta > \delta_2$ ) or decrease  $M$  (if  $\delta < \delta_2$ ) and use the `remez` algorithm again to determine a new  $\delta$ . We repeat this procedure until  $\delta \leq \delta_2$ . The optimal equiripple FIR filter, which satisfies all the three requirements discussed earlier is now determined. [2]

#### 2.4.4 MATLAB Implementation

The Park-McClellan algorithm is available in MATLAB as a function called `remez`, the most general syntax of which is:

`[h] = remez(N,f,m,weights,type)`

There are several versions of this syntax: 1

- `[h] = remez(N,f,m)` designs an  $N$ th-order (note that the length of the filter is  $M = N + 1$ ) FIR digital filter whose frequency response is specified by the arrays `f` and `m`. The filter coefficients (or the impulse response) are returned in array `h` of length  $M$ . The array `f` contains band-edge frequencies in units of  $\lambda$ , that is,  $0.0 < f < 1.0$

These frequencies must be in increasing order, starting with 0.0 and ending with 1.0. The array  $m$  contains the desired magnitude response at frequencies specified in  $f$ . The lengths of  $f$  and  $m$  arrays must be same and must be an even number. The weighting function must in each band is equal to unity.

- $[h] = \text{remez}(N,f,m,\text{weights})$  is similar to the above case except that the array  $\text{weights}$  specifies the weighting function in each band..

- $[h] = \text{remez}(N,f,m,\text{ftype})$  is similar to the first case except when  $\text{ftype}$  is the string 'differentiators' or 'hibert', it designs digital differentiators or digital Hilbert transformers, respectively. For the digital Hilbert transform lowest frequency in the  $C$  array should not be 0, and the highest frequency should not be 1, For the digital differentiator, the  $m$  vector does not specify the desired slope in each band but the desired magnitude

- $[h] = \text{remez}(N,f,m,\text{weights.ftype})$  is similar to the above case except that the array  $\text{weights}$  specifies the weighting function in each band.

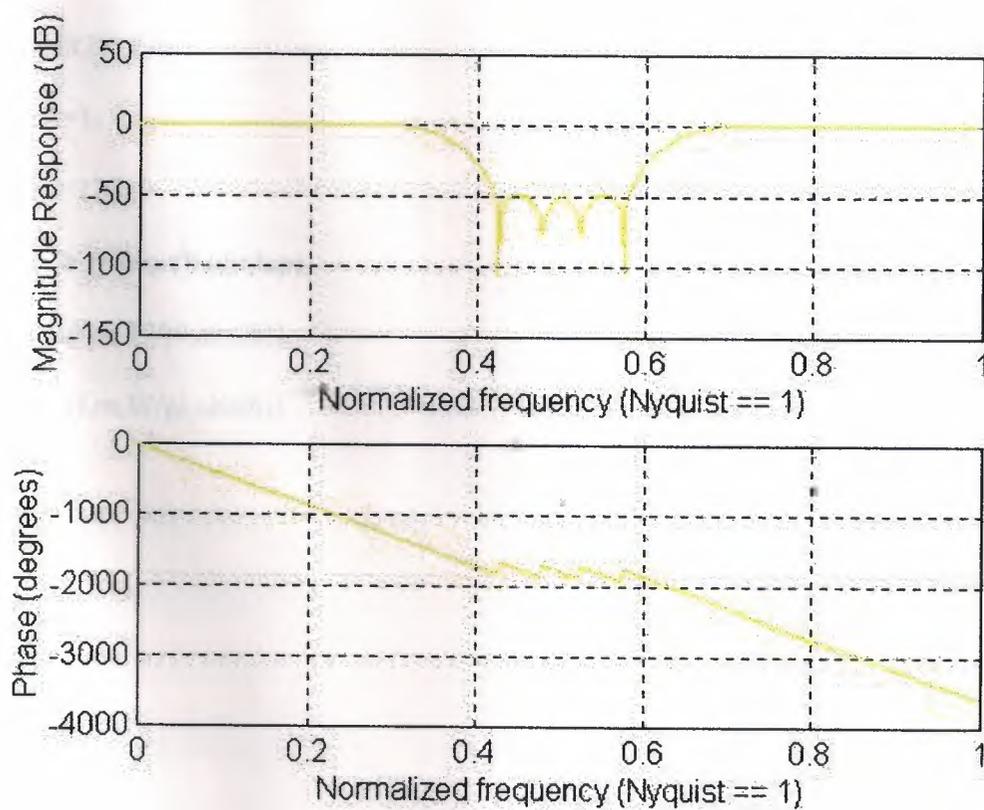
As explained during the description of the Park-McClellan algorithm, we have to first guess the order of the filter using (2-38) to use the routine `remez`. After we obtain the filter coefficients in array  $h$ .

### 3. EXAMPLES OF MATLAB

#### 3.1 Simple examples of FIR Filters

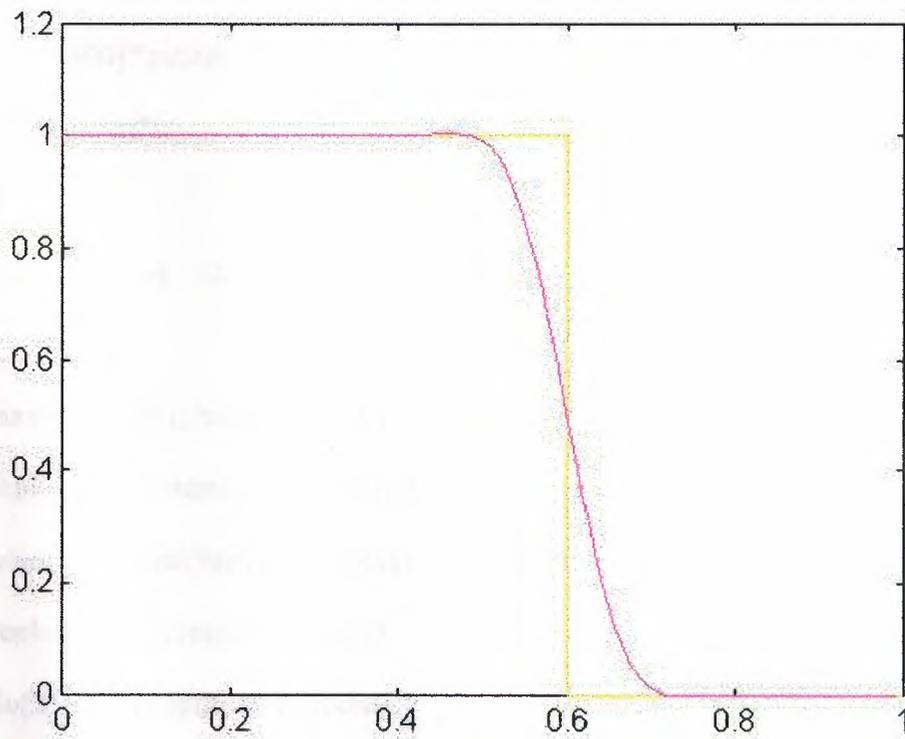
##### 3.1.1 Fir1

```
» n=48;  
» W1=0.35;  
» W2=0.65;  
» Wn=[W1 W2];  
» b=fir1(n,Wn,'nosale');  
» freqz(b,1,512)
```



### 3.1.2 Fir 2

```
f1=0;  
» f2=0.6;  
» f3=0.6;  
» f4=1;  
» f=[f1 f2 f3 f4];  
» m1=1;  
» m2=1;  
» m3=0;  
» m4=0;  
» m=[m1 m2 m3 m4];  
» b=fir2(n,f,m);  
» npt=1;  
» lap=128;  
» [h,W]=freqz(b,npt,lap);  
» plot(f,m,W/pi,abs(h))  
» plot(f,m,W/pi,abs(h))
```



### 3.1.3 Fir 3

```
» h=[-4,1,-1,-2,5,6,6,5,-2,-1,1,-4];
```

```
» M=length(h); n=0:M-1;
```

```
» L=(M-1)/2;
```

```
» a=[h(L+1) 2*h(L:-1:1)];
```

```
» n=[0:1:L];
```

```
» h=[-4,1,-1,-2,5,6,6,5,-2,-1,1,-4];
```

```
» M=length(h); n=0:M-1;
```

```
» L=M/2;
```

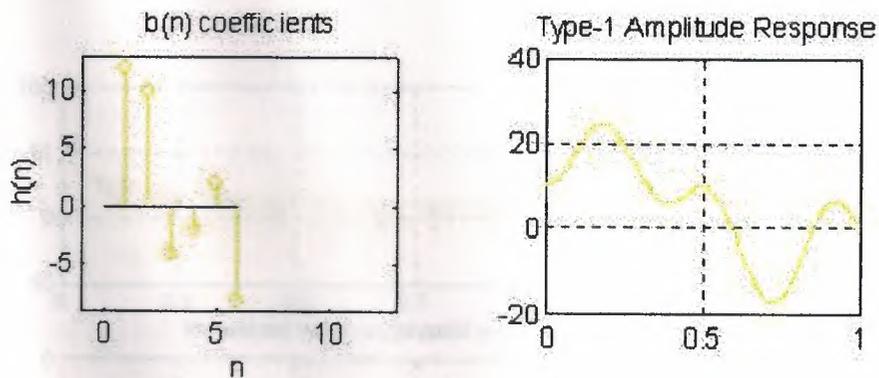
```
» b=2*[h(L:-1:1)];
```

```
» n=[1:1:L]; n=n-0.5;
```

```

» w=[0:1:500]*pi/500;
» Hr=cos(w*n)*b';
» b,L
b = 12 10 -4 -2 2 -8
L = 6
» bmax=max(b)+1; bmin=min(b)-1;
» subplot(2,2,3); stem(1:L,b); axis([-1 2*L+1 bmin bmax])
» xlabel('n'); ylabel('h(n)'); title('b(n) coefficients')
» subplot(2,2,2); plot(w/pi,Hr);grid
» title('Type-1 Amplitude Response')

```



## 3.2 Linear-phase FIR Filters

### 3.2.1 Hr-type1:

% Computes Amplitude response Hr(w) of a type-1 low pass filter

%[Hr,w,a,L] = Hr-type1(h)

% Hr = Amplitude Response

% w = 500 frequencies between [0 pi] over which Hr is computed

```
% a = type-1 LP filters impulse coefficients
```

```
% L = Order of Hr
```

```
% h = Type-1 LP filter impulse response
```

```
» h=[-4,1,-1,-2,5,6,5,-2,-1,1,-4];
```

```
» L=(M-1)/2;
```

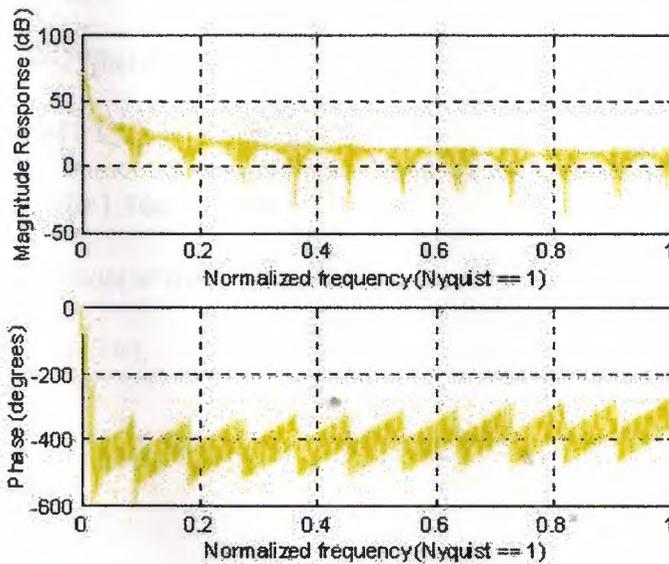
```
» a=[h(L+1) 2*h(L:-1:1)];
```

```
» n=[0:1:L];
```

```
» w=[0:1:500]*pi/500;
```

```
» Hr=cos(w*n)*a';
```

```
» freqz(Hr);
```



### 3.2.2 Hr-type 2

% Computes Amplitude response Hr(w) of a type-2 low pass filter

%[Hr,w,b,L] = Hr-type2(h)

% Hr = Amplitude Response

% w = 500 frequencies between [0 pi] over which Hr is computed

% b = type-2 LP filters impulse coefficients

% L = Order of Hr

% h = Type-2 LP filter impulse response

» h=[-5,-1,3,6,9];

» M=length(h);

» L=M/2;

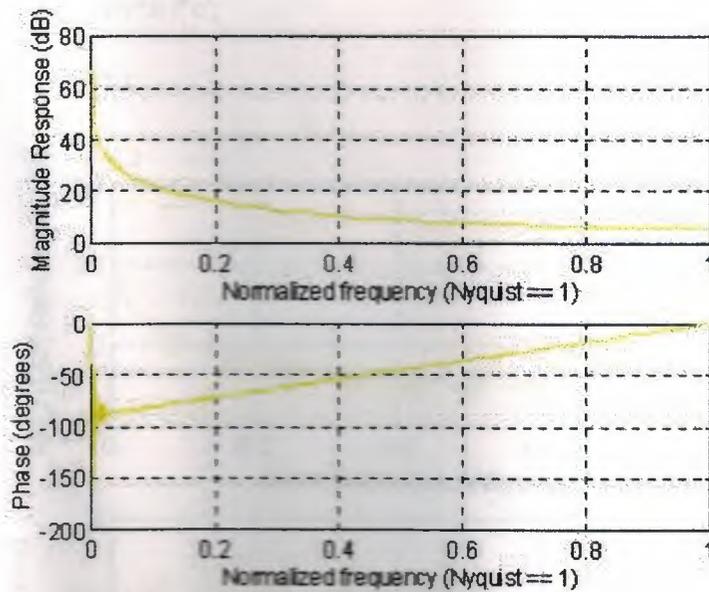
» b=2\*[h(L:-1:1)];

» n=[1:L]; n=n-0.5;

» w=[0:1:500]\*pi/500;

» Hr=cos(w\*n)\*b';

» freqz(Hr);



### 3.2.3 Hr-type 3

% Computes Amplitude response  $H_r(\omega)$  of a type-3 low pass filter

%  $[H_r, w, c, L] = \text{Hr-type3}(h)$

%  $H_r$  = Amplitude Response

%  $w$  = 500 frequencies between  $[0 \pi]$  over which  $H_r$  is computed

%  $c$  = type-3 LP filters impulse coefficients

%  $L$  = Order of  $H_r$

%  $h$  = Type-3 LP filter impulse response

»  $h = [-4, 1, -1, -2, -4];$

»  $M = \text{length}(h);$

»  $L = (M-1)/2;$

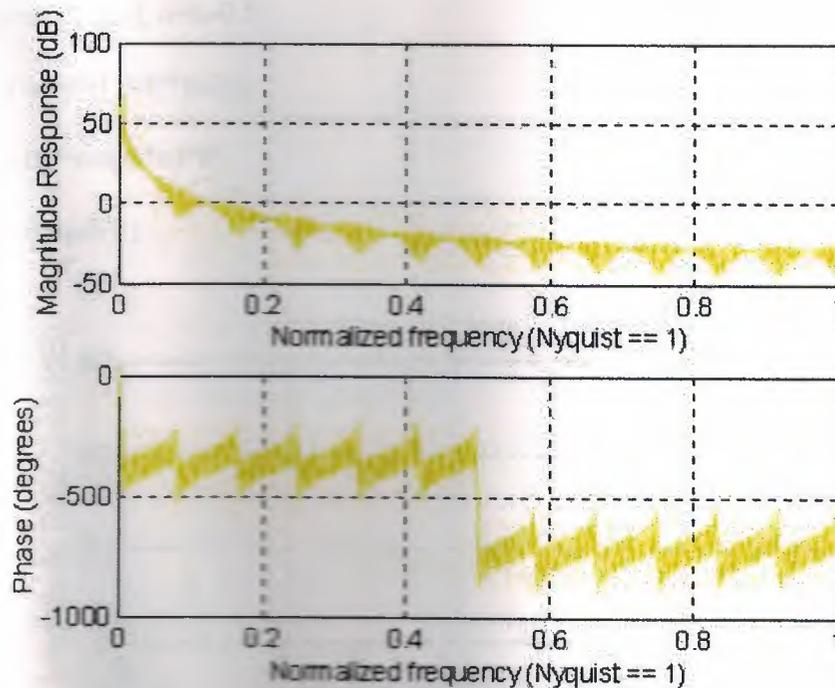
»  $c = [2 * h(L+1:-1:1)];$

»  $n = [0:1:L];$

»  $w = [0:1:500] * \pi / 500;$

```
» Hr=sin(w*n)*c';
```

```
» freqz(Hr);
```



### 3.2.4 Hr-Type-4

```
% Computes Amplitude response Hr(w) of a type-4 low pass filter
```

```
%[Hr,w,d,L] = Hr-type4(h)
```

```
% Hr = Amplitude Response
```

```
% w = 500 frequencies between [0 pi] over which Hr is computed
```

```
% d = type-4 LP filters impulse coefficients
```

```
% L = Order of Hr
```

```
% h = Type-4 LP filter impulse response
```

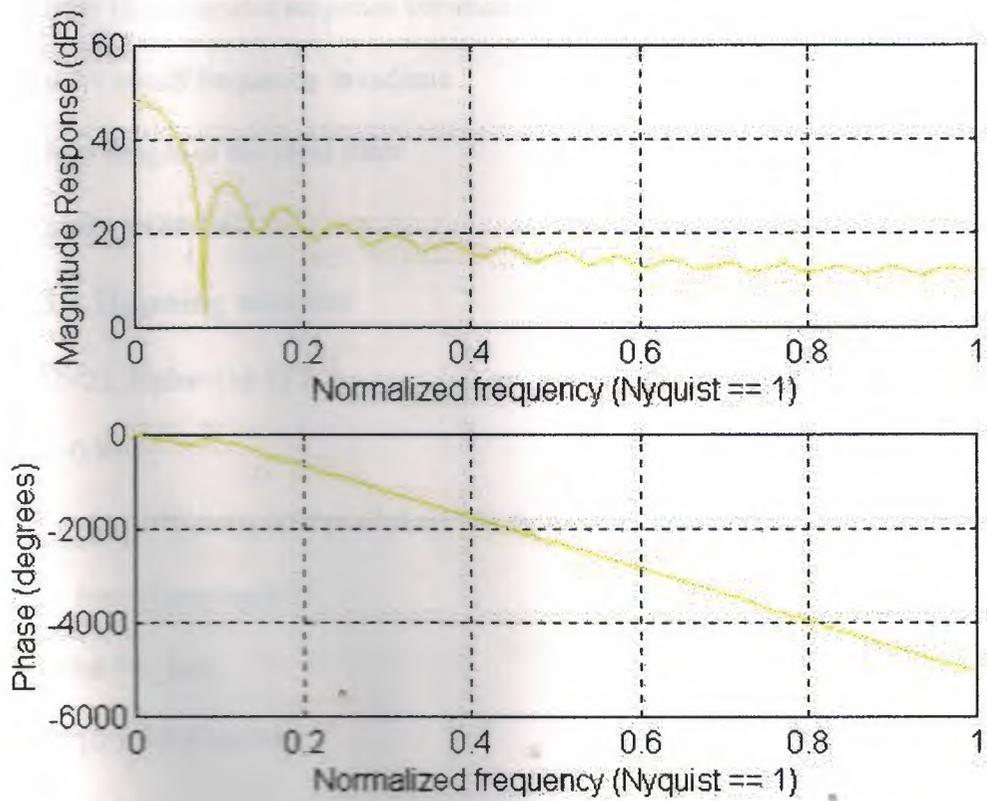
```
» h=[2,6,-1,5];
```

```
» M=length(h);
```

```

» L=M/2;
» d=2*[h(L:-1:1)];
» n=[1:L]; n=n-0.5;
» w=[0:1:30]*pi/30;
» Hr=sin(w*n)*d';
» freqz(Hr);

```



### 3.3 Window Design Techniques

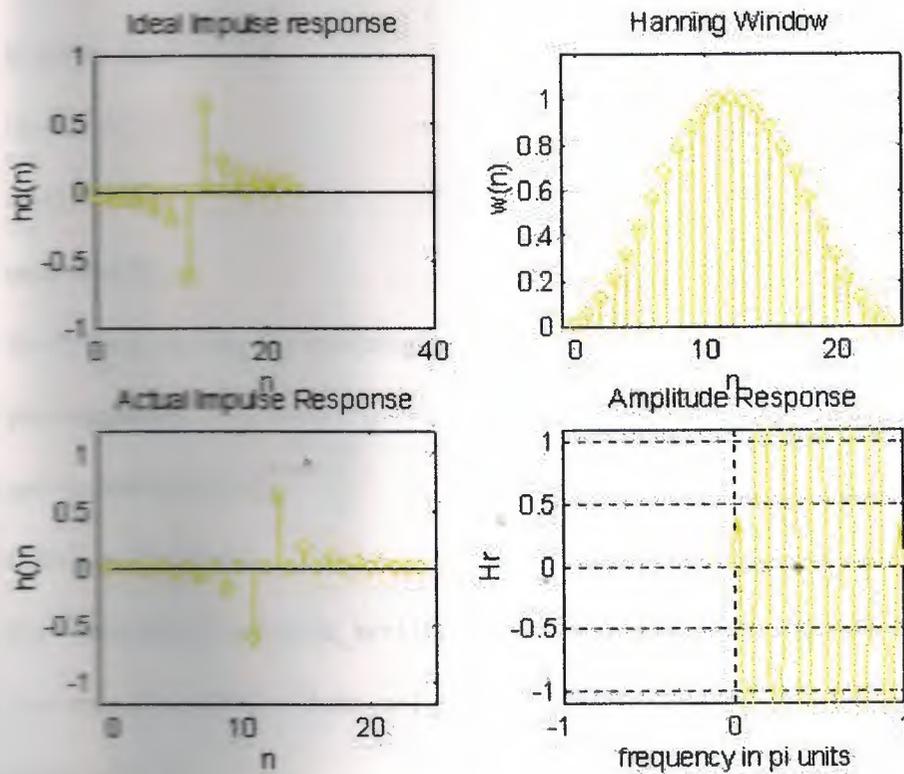
```
function hd = ideal_lp(wc,M);  
  
% Ideal lowpass filter computation  
%-----  
  
%[hd] = ideal_lp(wc,M)  
  
% hd = ideal impulse response between 0 to M-1  
  
% wc = cutoff frequency in radians  
  
% M = length of the ideal filter  
  
% alpha = (M-1)/2
```

#### 3.3.1 Hanning window

```
» M=25; alpha=(M-1)/2;  
» n=0:M-1;  
» hd=(2/pi)*((sin((pi/2)*(n-alpha)).^2)./(n-alpha)); hd(alpha+1)=0;  
» w_han=(hanning(M));  
» h=hd.*w_han;  
» w=[0:1:500]*pi/500;  
» L=alpha;  
» c=2*h;  
» Hr=sin(w*n)*c';  
» subplot(1,1,1);  
» subplot(2,2,1); stem(n,hd); title('Ideal Impulse response');
```



```
» xlabel('n'); ylabel('hd(n)');  
» subplot(2,2,1); stem(n,hd); title('Ideal Impulse response');  
» xlabel('n'); ylabel('hd(n)');  
» subplot(2,2,2); stem(n,w_han); title('Hanning Window')  
» axis([-1 M 0 1.2]); xlabel('n'); ylabel('w(n)')  
» subplot(2,2,3); stem(n,h); title('Actual Impulse Response')  
» axis([-1 M -1.2 1.2]); xlabel('n'); ylabel('h(n)')  
» subplot(2,2,4); plot(w/pi,Hr); title('Amplitude Response');grid  
» axis([-1 1 -1.1 1.1]); xlabel('frequency in pi units'); ylabel('Hr');
```



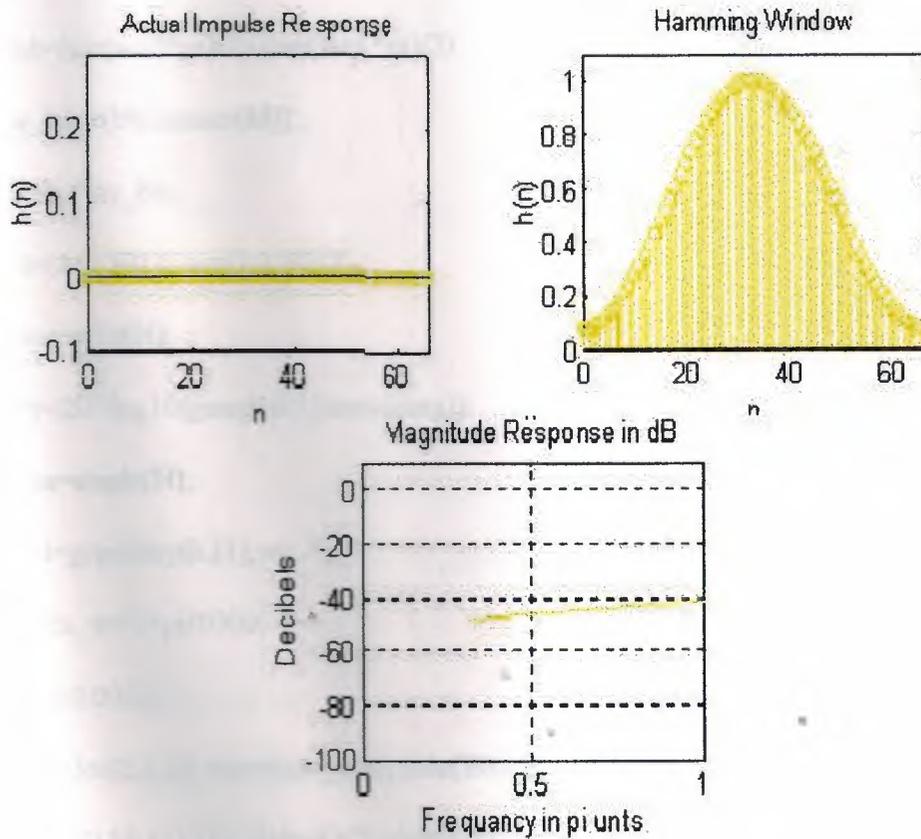
### 3.3.2 Hamming Window

```
» wp=0.2*pi; ws=0.3*pi;
» tr_width=ws-wp;
» M=ceil(6.6*pi/tr_width)+1;
» n=[0:1:M-1];
» wc=(ws+wp)/2;
» alpha=(M-1)/2;
» m=n-alpha+1;
» hd=sin(wc*m)/(pi*m);
» w_ham=(hamming(M))';
» h=hd*w_ham;
» H=[1:1:501];
» w=[1:1:501];
» mag=abs(H);
» db=20*log10((mag+1)/max(mag));
» pha=angle(H);
» grd=grpdelay(h,1,w);
» delta_w=2*pi/1000;
» Rp=-(min(db(1:1:wp/delta_w+1)));
» As=-round(max(db(ws/delta_w:1:501)));
» subplot(1,1,1)
» subplot(2,2,2); stem(n,w_ham); title('Hamming Window');
» axis([0 M-1 0 1.1]); xlabel('n'); ylabel('h(n)')
```

```

» subplot(2,2,3); stem(n,h); title('Actual Impulse Response')
» axis([0 M-1 -0.1 0.3]); xlabel('n'); ylabel('h(n)')
» subplot(2,2,4); plot(w/pi,db); title('Magnitude Response in dB');grid
» axis([0 1 -100 10]); xlabel('Frequency in pi units'); ylabel('Decibels')

```



### 3.3.3 Blackman window

```

» ws1=0.2*pi; wp1=0.35*pi;

```

```

» ws2=0.8*pi; wp2=0.65*pi;

» As=60;

» tr_width=min((wp1-ws1),(ws2-wp2));

» M=ceil(11*pi/tr_width)+1;

» n=[0:1:M-1];

» wc1=(ws1+wp1)/2; wc2=(wp2+ws2)/2;

» alpha=(M-1)/2;

» m=n-alpha+.1;

» hd=(sin(wc2*m)/2)-(sin(wc1*m)/2);

» w_bla=(blackman(M))';

» h=hd.*w_bla;

» H=(1:1:501)'; w=(1:1:501)';

» mag=abs(H);

» db=20*log10((mag+0.1)/max(mag));

» pha=angle(H);

» grd=grpdelay(h,[1],w);

» delta_w=2*pi/1000;

» Rp=0.0030;

» subplot(2,2,2); stem(n,w_bla); title('Blackman Window')

» axis([0 M-1 0 1]); xlabel('n'); ylabel('hd(n)')

» subplot(2,2,3); stem(n,h); title('Actual Impulse Response')

» axis([0 M-1 -0.6 0.8]); xlabel('n'); ylabel('h(n)')

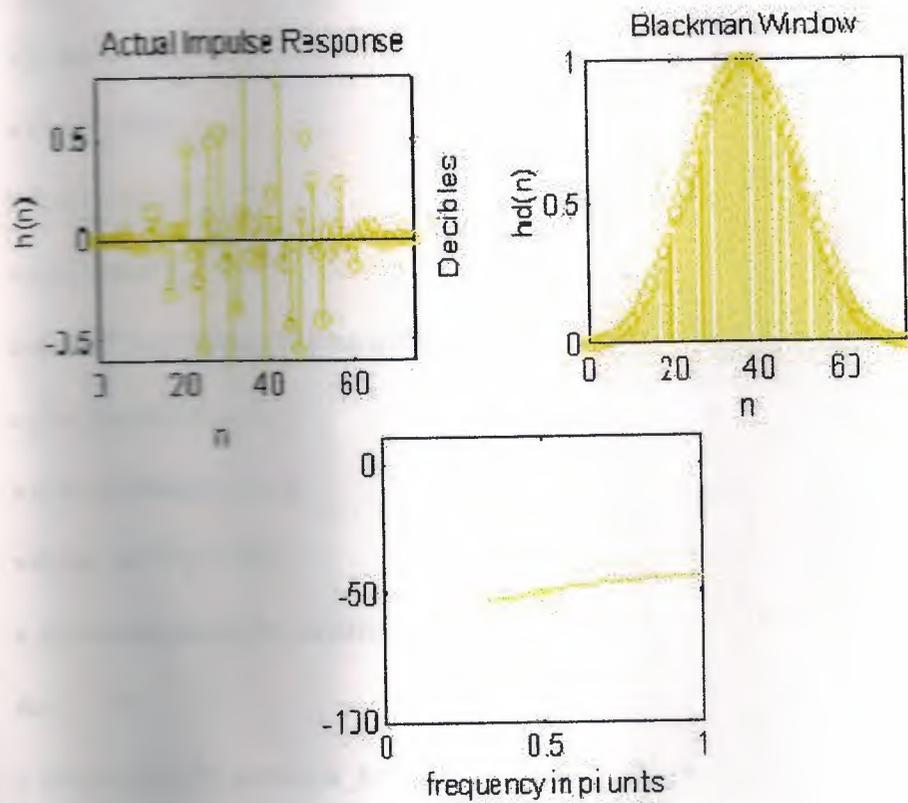
» subplot(2,2,4); plot(w/pi,db); axis([0 1 -150 10])

```

```

» subplot(2,2,4); plot(w/pi,db); axis([0 1 -15 10])
» subplot(2,2,4); plot(w/pi,db); axis([0 1 -150 10])
» subplot(2,2,4); plot(w/pi,db); axis([0 1 -100 10])
» xlabel('frequency in pi units'); ylabel('Decibels')

```



### 3.3.4 Kaiser window

```

» wp=0.2*pi; ws=0.3*pi; As=50;
» tr_width=ws-wp;
» M=ceil((As-7.95)/(14.36*tr_width/(2*pi))+1)+1
M = 61
» n=[0:1:M-1];

```

```

» beta=0.1102*(As-8.7)

beta = 4.5513

» wc=(ws+wp)/2;

» hd=sin(wc*m)/(pi*m);

» w_kai=(kaiser(M,beta));

» h=hd*w_kai;

» H=[1:1:501];

» w=[1:1:501];

» mag=abs(H);

» db=20*log10((mag+1)/max(mag));

» pha=angle(H);

» grd=grpdelay(h,1,w);

» delta_w=2*pi/1000;

» As=-round(max(db(ws/delta_w+1:1:501)))

As = 52

» subplot(2,2,2); stem(n,w_kai); title('Kaiser Window')

» axis([0 M-1 0 1.1]); xlabel('n'); ylabel('h(n)')

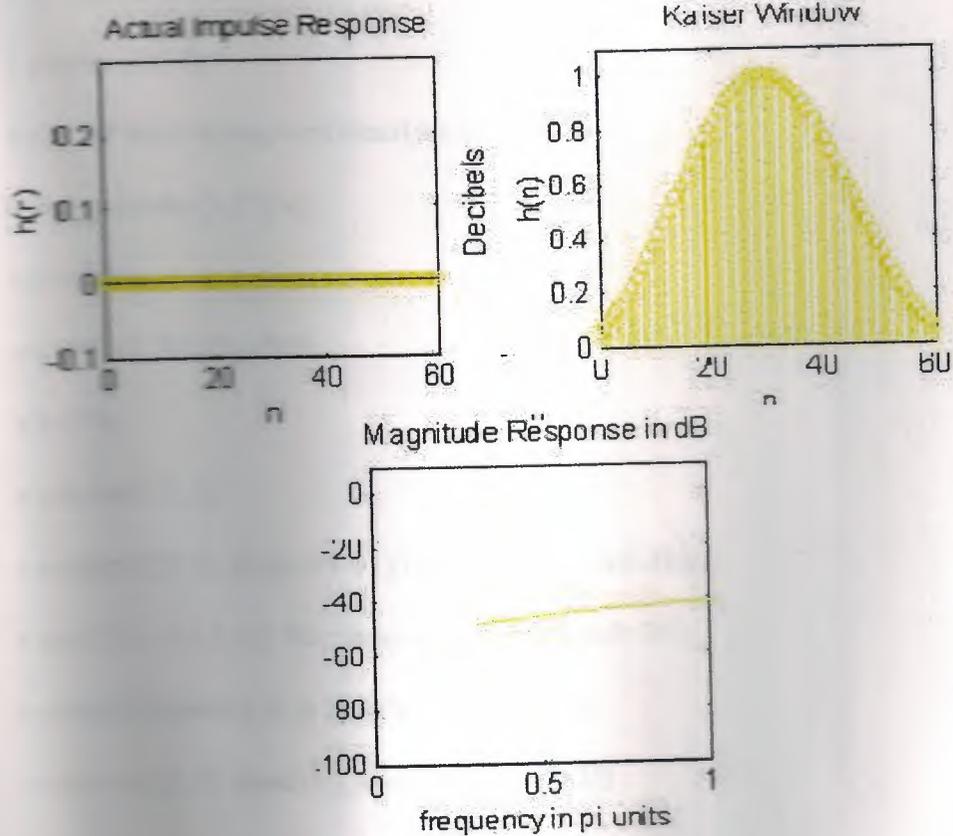
» subplot(2,2,3); stem(n,h); title('Actual Impulse Response')

» axis([0 M-1 -0.1 0.3]); xlabel('n'); ylabel('h(n)')

» subplot(2,2,4); plot(w/pi,db); title('Magnitude Response in dB')

» axis([0 1 -100 10]); xlabel('frequency in pi units'); ylabel('Decibels')

```



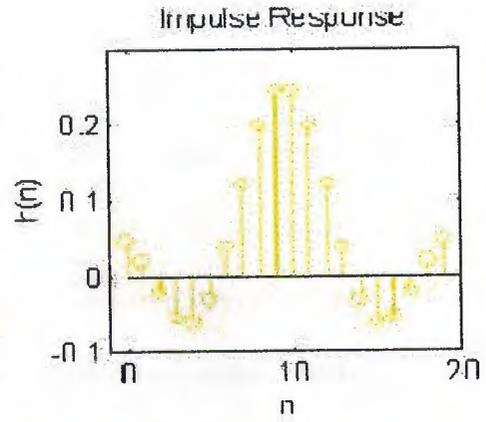
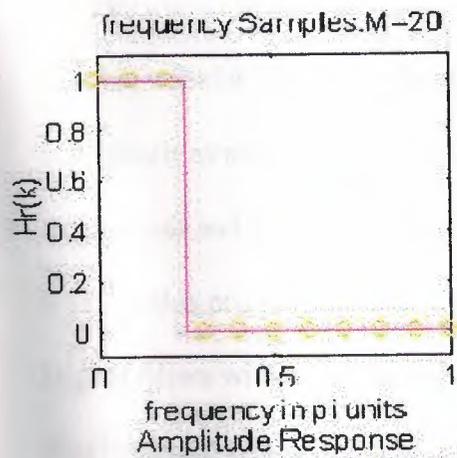
### 3.3.5 Naive window

- »  $M=20$ ;  $\alpha=(M-1)/2$ ;  $l=0:M-1$ ;  $wl=(2*\pi/M)*l$ ;
- »  $Hw=[1,1,1,zeros(1,15),1,1]$ ;
- »  $Hd=[1,1,0,0]$ ;  $wdl=[0,0.25,0.25,1]$ ;
- »  $k1=0:\text{floor}((M-1)/2)$ ;  $k2=\text{floor}((M-1)/2)+1:M-1$ ;
- »  $\text{angH}=[-\alpha*(2*\pi)/M*k1,\alpha*(2*\pi)/M*(M-k2)]$ ;
- »  $H=Hw.*\exp(j*\text{angH})$ ;

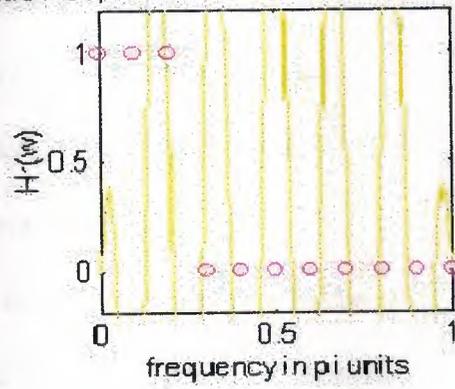
```

» h=real(iffit(H,M));
» mag=abs(H);
» w=(w(1:1:501))';
» pha=angle(H);
» db=20*log10((mag+eps)/max(mag));
» grd=grpdelay(h,[1],w);
» L=alpha;
» ww=[0:1:500]*pi/500;
» b=2*h;
» subplot(1,1,1);
» subplot(2,2,1); plot(wl(1:11)/pi,Hrs(1:11),'o',wdl,Hdr);
» axis([0,1,-0.1,1.1]); title('frequency Samples:M=20')
» xlabel('frequency in pi units'); ylabel('Hr(k)');
» subplot(2,2,2); stem(1,h); axis([-1,M,-0.1,0.3])
» title('Impulse Response'); xlabel('n'); ylabel('h(n)');
» subplot(2,2,3); plot(ww/pi,Hr,wl(1:11)/pi,Hrs(1:11),'o');
» axis([0,1,-0.2,1.2]); title('Amplitude Response');
» xlabel('frequency in pi units'); ylabel('Hr(w)');

```



Amplitude Response



## CONCLUSION

The digital filter design problem is concerned with the development of suitable transfer function meeting the frequency response specifications. The specifications are usually given in the term of pass band and stopband edge frequencies and allowable deviations from passband and stopband magnitude level. In this project the main objective was to obtain linear phase response. Digital filters with finite duration impulse response have both advantages and disadvantages compared to infinite duration impulse response (IIR) filters.

FIR filters have the following advantages:

- They can have exactly linear phase.
- They are always stable.
- The design methods are generally linear.
- They can be realized efficiently in hard ware.
- The filter startup transients have finite duration.

The primary disadvantage of FIR filter is that they often require a much higher filter order than IIR filters to achieve a given level of performance.

Correspondingly, the delay of these filters is often much greater than for an equal performance IIR filter.

## References

- [1] *Stilianos Antoniou* (1998), *Digital Filters Analysis and Design*, New York, McGraw-Hill Company.
- [2] *Kishan Sreeni*. (1997) *Digital Signal processing in Telecommunication*, San Jose, California. Telecom Solutions, Inc.,
- [3] *L.B.Jackson, J.F.Kaiser, and H.S.McDonald*, (1988) An Approach to the Implementation of Digital Filters, New Jersey, IEEE Trans., Audio Electroacoustics, pp. 413-421.
- [4] *S.K.Tewksbury*, (1973) Special Purpose Hardware Implementation of Digital Filter, NJ USA proc. IEEE int. Symp. Circuits syst., pp. 418-421.
- [5] *James McClellan, Roland Schafer and Mark Yoder*, (1998) *DSP First: A Multimedia Approach*, Georgia, the Georgia Institute of Technology.
- [6] *James McClellan, C.Sidney Burrus, Thomas W. Park and Alan V.Oppenheim*, (1998) *Computer Based Exercise for Signal Processing Using Matlab*, New York. Schafer/Schuessler.
- [7] *Ashek Ambaradar and Craig Borghesani*, (1998) *Mastering DSP Concepts Using Matlab*, Michigan.Terasoft, Inc
- [8] *E. Kamen and Bonnie Heck*, (1997) *Fundamentals of Signals and Systems: with Matlab*, Georgia, the Georgia Institute of Technology.
- [9] *Matlab Signal Processing tool box (User Guide)* (1999), New York, The Mathworks Inc.
- [10] Web site: <http://www.mathworks.com>

## Index

### A

Absolute Specifications (9)

### B

Basic Window Design Idea (23)

Bartlett Window (27)

Blackman Window (29)

Basic Idea F.S.D.T (32)

### C

Constraint on the Number of Extrema (38)

### D

Development of the Minimax Problem (35)

### F

Frequency Response (15)

Filter Designing Models (3)

FIR and IIR Filters (7)

### H

Hanning Window (27)

Hamming Window (28)

### I

Introduction to Filters (1)

### K

Kaiser Window (29)

### M

Matlab examples (42)

### O

Optimum design method (33)

Optimal Equiripple Design (34)

Technique

### P

Properties of Linear-Phase

FIR Filters (12)

Parks-McClellan algorithm (39)

### R

Rectangular Window (24)

### T

Theorem1: Alternation (39)

### W

Window Design Techniques (20)

### Z

Zero Locations (18)