

## 8. Design of digital FIR Low Pass and High Pass filter using Hanning and Hamming window

### Theory

#### Finite Impulse Response (FIR) Filter

FIR filters are digital filters with finite impulse response. They are also known as non-recursive digital filters as they do not have the feedback (a recursive part of a filter), even though recursive algorithms can be used for FIR filter realization. Hence it is an all zero filter. Therefore, input and output difference equation for FIR filter is given by

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + \dots + b_{M-1}x(n-N+1)$$

Where  $b_0, b_1, b_2 \dots b_{M-1}$  are filter coefficients. FIR filters are particularly useful for applications where exact linear phase response is required. The FIR filter is generally implemented in a non-recursive way which guarantees a stable filter.

FIR filters can be designed using different methods, but most of them are based on ideal filter approximation. The objective is not to achieve ideal characteristics, as it is impossible anyway, but to achieve sufficiently good characteristics of a filter. The transfer function of FIR filter approaches the ideal as the filter order increases, thus increasing the complexity and amount of time needed for processing input samples of a signal being filtered.

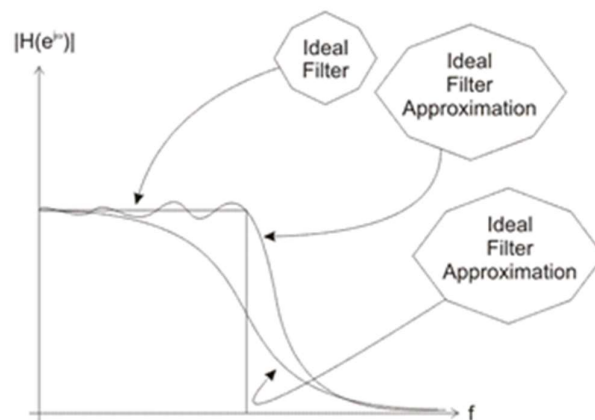


Fig-1

FIR filters can have linear phase characteristic, which is not like IIR filters. Obviously, in such cases when it is necessary to have a linear phase characteristic, FIR filters are the only option available. If the linear phase characteristic is not necessary, as is the case with processing speech signals, FIR filters are not good solution at all.

One of the drawbacks of FIR filters is a high order of designed filter. The order of FIR filter is remarkably higher compared to an IIR filter with the same frequency response. This is the reason why it is so important to use FIR filters only when the linear phase characteristic is very important.

A number of delay lines contained in a filter, i.e. a number of input samples that should be saved for the purpose of computing the output sample, determines the order of a filter. For example, if the filter is assumed to be of order 10, it means that it is necessary to save 10 input samples preceding the current sample. All eleven samples will affect the output sample of FIR filter

The transform function of a typical FIR filter can be expressed as a polynomial of a complex variable  $z^{-1}$ . All the poles of the transfer function are located at the origin. For this reason, FIR filters are guaranteed to be stable, whereas IIR filters have potential to become unstable

### Basic concepts and FIR filter specification

Most FIR filter design methods are based on ideal filter approximation. The resulting filter approximates the ideal characteristic as the filter order increases, thus making the filter and its implementation more complex.

Figure 2.a and 2.b illustrates a low-pass digital filter specification. The word specification actually refers to the frequency response specification

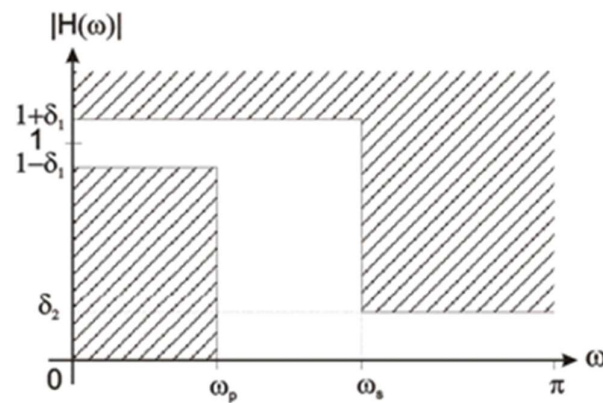


Figure 2.a: Low-pass digital filter specification

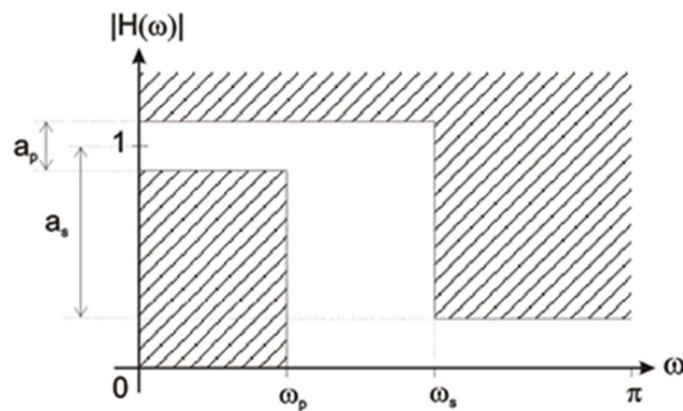


Figure 2.b: Low-pass digital filter specification

$\omega_p$  -normalized cut-off frequency in the passband;

$\omega_s$ - normalized cut-off frequency in the stopband;

$\delta_1$  -maximum ripples in the passband

$\delta_2$  - minimum attenuation in the stopband [dB]

$a_p$  - maximum ripples in the passband; and

$a_s$  - minimum attenuation in the stopband [dB].

$$a_p = 20 \log_{10} \left( \frac{1 + \delta_1}{1 - \delta_1} \right)$$

$$a_s = -20 \log_{10} \delta_2$$

Frequency normalization can be expressed as follows:

$$\omega = (2\pi f / f_s)$$

where:

$f_s$  - is a sampling frequency;

$f$  - is a frequency to normalize; and

$\omega$  - is normalized frequency.

Specifications for high-pass, band-pass and band-stop filters are defined almost the same way as those for low-pass filters. Figure 3.a and 3.b illustrates a high-pass filter specification

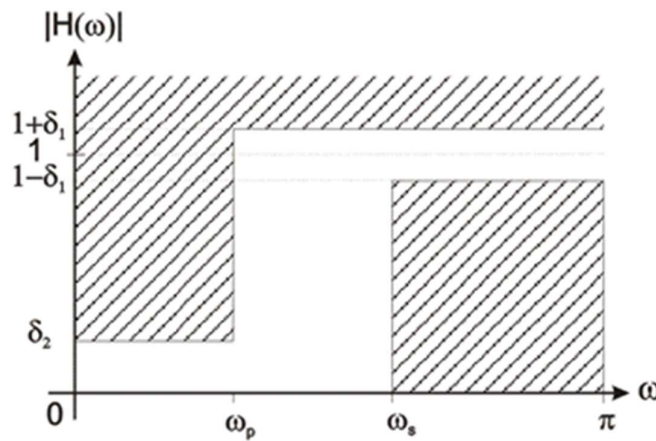


Figure 3.b: High-pass digital filter specification

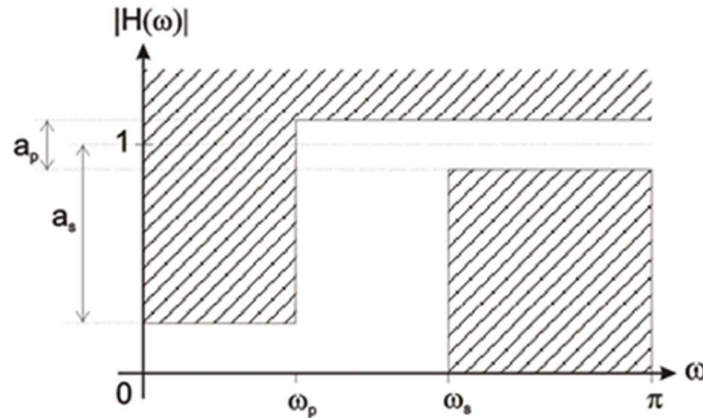


Figure 3.b: High-pass digital filter specification

Comparing these two figures 2.a, 2.b and 3.a, 3.b, it is obvious that low-pass and high-pass filters have similar specifications. The same values are defined in both cases with the difference that in the latter case the pass band is substituted by the stop band and vice versa.

### Finite impulse response (FIR) filter design methods

The filter design process starts with specifications and requirements of the desirable FIR filter. Which method is to be used in the filter design process depends on the filter specifications and implementation?

There are essentially three well-known methods for FIR filter design namely:

1. The window method.
2. The frequency sampling technique.
3. Optimal filter design methods

Each of the given methods has its advantages and disadvantages. Thus, it is very important to carefully choose the right method for FIR filter design. Due to its simplicity and efficiency, the window method is most commonly used method for designing filters. The sampling frequency method is easy to use, but filters designed this way have small attenuation in the stop band.

Out of these three techniques we are going to discuss about window method only

### Ideal filter approximation

The ideal filter frequency response is used when designing FIR filters using window functions. The objective is to compute the ideal filter samples. FIR filters have finite impulse response, which means the ideal filter frequency sampling must be performed in a finite number of points. As the ideal filter frequency response is infinite, it is easy to produce sampling errors. The error is less as the filter order increases. Figure 4.a and 4.b illustrates the transfer functions of two standard ideal filters

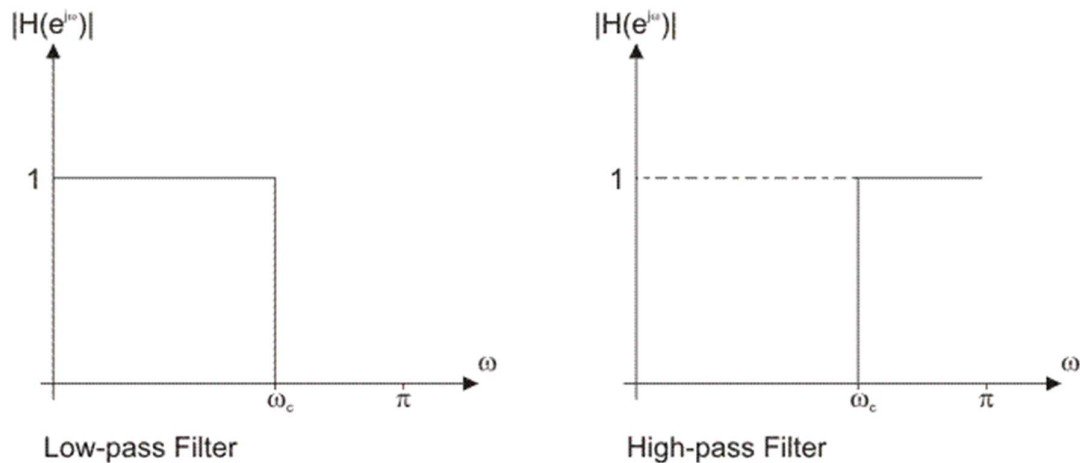


Figure 4.a and 4.b: Transfer functions of two standard ideal filters

The ideal filter frequency response can be computed via inverse Fourier transform. The two standard ideal filters frequency responses are contained in the table 1 below.

Type of filter	Frequency response $h_d[n]$
low-pass filter	$h_d[n] = \begin{cases} \frac{\sin[\omega_c(n-M)]}{\pi(n-M)}; & n \neq M \\ \frac{\omega_c}{\pi}; & n = M \end{cases}$
high-pass filter	$h_d[n] = \begin{cases} 1 - \frac{\omega_c}{\pi}; & n \neq M \\ -\frac{\sin(\omega_c(n-M))}{\pi(n-M)}; & n = M \end{cases}$

Table 1: The frequency responses of two standard ideal filters

The value of variable  $n$  ranges between 0 and  $N$ , where  $N$  is the filter order. A constant  $M$  can be expressed as  $M = N / 2$ . Equivalently,  $N$  can be expressed as  $N = 2M$

The constant  $M$  is an integer if the filter order  $N$  is even, which is not the case with odd order filters. If  $M$  is an integer (even filter order), the ideal filter frequency response is symmetric about its  $M$ th sample which is found via expression shown in the table 1 above. If  $M$  is not an integer, the ideal filter frequency response is still symmetric, but not about some frequency response sample

Since the variable  $n$  ranges between 0 and  $N$ , the ideal filter frequency response has  $N+1$  sample

If it is needed to find frequency response of a non-standard ideal filter, the expression for inverse Fourier transform must be used:

$$h_d[n] = 1/\pi \int_0^\pi e^{j\omega(n-M)} d\omega$$

Non-standard filters are rarely used. However, if there is a need to use some of them, the integral above must be computed via various numerical methods

### **FIR filter design using window functions**

The FIR filter design process via window functions can be split into several steps:

1. Defining filter specifications;
2. Specifying a window function according to the filter specifications;
3. Computing the filter order required for a given set of specifications;
4. Computing the window function coefficients;
5. Computing the ideal filter coefficients according to the filter order;
6. Computing FIR filter coefficients according to the obtained window function and ideal filter coefficients
7. If the resulting filter has too wide or too narrow transition region, it is necessary to change the filter order by increasing or decreasing it according to needs, and after that steps 4, 5 and 6 are iterated as many times as needed

The final objective of defining filter specifications is to find the desired normalized frequencies, transition width and stopband attenuation. The window function and filter order are both specified according to these parameters. Accordingly, the selected window function must satisfy the given specifications.

After this step, that is, when the window function is known, we can compute the filter order required for a given set of specifications.

When both the window function and filter order are known, it is possible to calculate the window function coefficients  $w[n]$  using the formula for the specified window function

After estimating the window function coefficients, it is necessary to find the ideal filter frequency samples. The final objective of this step is to obtain the coefficients  $h_d[n]$ . Two sequences  $w[n]$  and  $h_d[n]$  have the same number of elements.

The next step is to compute the frequency response of designed filter  $h[n]$  using the following expression:

$$h[n] = w[n] \cdot h_d[n]$$

Lastly, the transfer function of designed filter will be found by transforming impulse response via Fourier transform:

$$H(e^{j\omega}) = \sum_{n=0}^N h(n) \cdot e^{-jn\omega}$$

If the transition region of designed filter is wider than needed, it is necessary to increase the filter order, reestimate the window function coefficients and ideal filter frequency samples, multiply them in order to obtain the frequency response of designed filter and reestimate the transfer function as well. If the transition region is narrower than needed, the filter order can be decreased for the purpose of optimizing hardware and/or software resources. It is also necessary to reestimate the filter frequency coefficients after that. For the sake of precise estimates, the filter order should be decreased or increased by 1.

## **Window functions**

The window method is most commonly used method for designing FIR filters. The simplicity of design process makes this method very popular.

A window is a finite array consisting of coefficients selected to satisfy the desirable requirements. This chapter provides a few methods for estimating coefficients and basic characteristics of the window itself as well as the result filters designed using these coefficients. The point is to find these coefficients denoted by  $w[n]$ .

When designing digital FIR filters using window functions it is necessary to specify:

A window function to be used; and

The filter order according to the required specifications (selectivity and stop band attenuation).

These two requirements are interrelated. Each function is a kind of compromise between the two following requirements:

The higher the selectivity, i.e. the narrower the transition region; and

The higher suppression of undesirable spectrum, i.e. the higher the stop band attenuation.

Table 2 below contains all window functions mentioned in this chapter and briefly compares their selectivity and stop band attenuation

These two requirements are interrelated. Each function is a kind of compromise between the two following requirements:

The higher the selectivity, i.e. the narrower the transition region; and

The higher suppression of undesirable spectrum, i.e. the higher the stop band attenuation.

Table 2 below contains all window functions mentioned in this chapter and briefly compares their selectivity and stop band attenuation.

Special attention should be paid to the fact that minimum attenuation of window function and that of the filter designed using that function are different in most cases. The difference, i.e. additional attenuation occurs under the process of designing a filter using window functions. This affects the stop band attenuation to become additionally higher, which is very desirable.

However, a drawback of this method is that the minimum stop band attenuation is fixed for each function. The following concepts such as the main lobe, main lobe width, side lobes, transition region, minimum stopband attenuation of window function and minimum stopband attenuation of designed filter are described in more detail in Figure 5

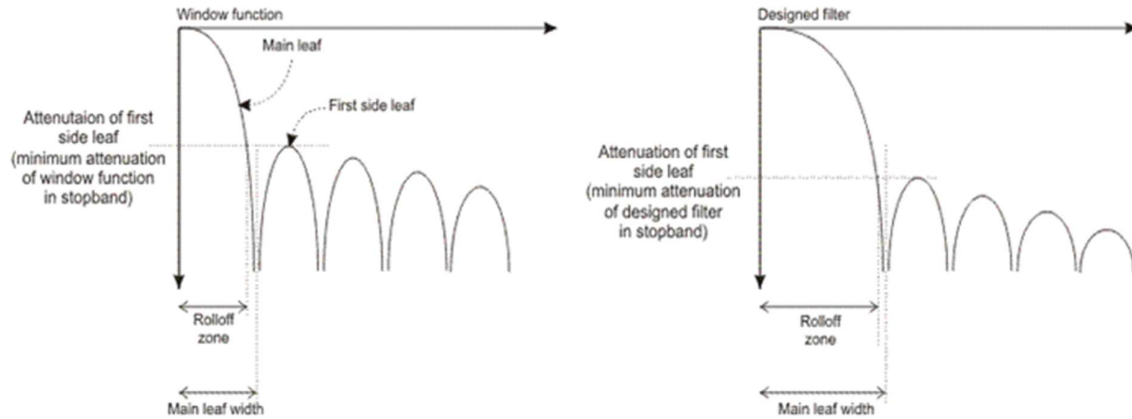


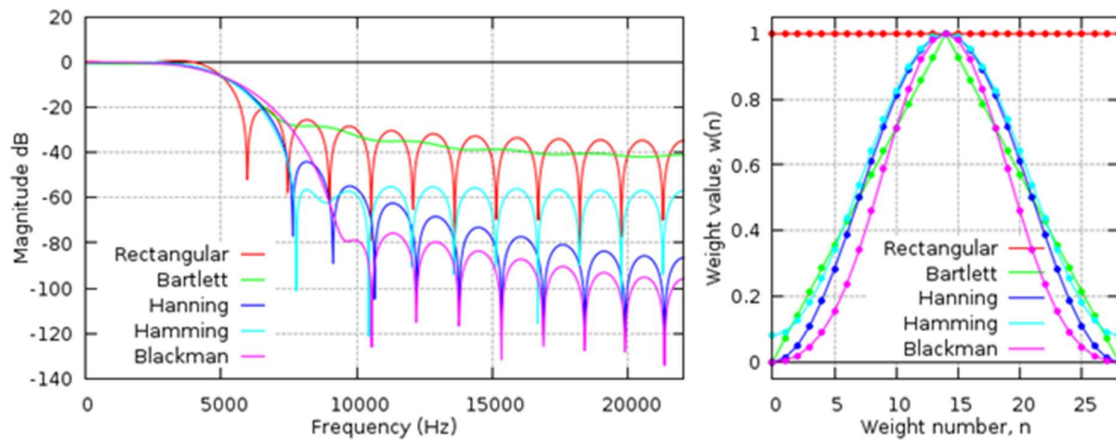
Figure 5: Main lobe, main lobe width, side lobes, transition region

As can be seen in the table 2 above, the stopband attenuation of these windows is not adjustable. It is only possible to affect the transition region by increasing the filter order. For this reason, it is preferable to start design process by specifying the appropriate window function on the basis of the stopband attenuation. It is most preferable to specify a window with the least stopband attenuation that satisfies the given requirements. This enables the designed filter to have the narrowest transition region

Name of Window	Time-domain sequence of window $w(n)$ , $0 \leq n \leq N-1$
Rectangular Window	1
Barlett (triangular) window	$1 - \frac{2 n - \frac{N-1}{2} }{N-1}$
Blackman Window	$0.42 - 0.5\cos\frac{2\pi n}{N-1} + 0.08\cos\frac{4\pi n}{N-1}$
Hamming Window	$0.54 - 0.46\cos\frac{2\pi n}{N-1}$
Hanning Window	$\frac{1}{2}(1 - \cos\frac{2\pi n}{N-1})$
Kaiser Window <i><math>I_0(x)</math> is the zeroth order Besselfunction.</i>	$\frac{I_0[\alpha\sqrt{(\frac{N-1}{2})^2 - (n - \frac{N-1}{2})^2}]}{I_0[\alpha(\frac{N-1}{2})]}$

### Frequency Response and Weight Values of different windows types





Design an ideal high pass filter with a frequency response

$$H_d e^{j\omega} = 1 \text{ for } \pi/4 \leq |\omega| \leq \pi$$

$= 0 \text{ for } |\omega| \leq \pi/4$ . Find the values of  $h(n)$  for  $N=11$ . Find  $H(z)$ . Plot the magnitude response using Hanning window and Hamming window.

Formula for Hanning window

The rectangular window (no window function):

$$w_R(n) = 1 \quad 0 \leq n \leq N-1$$

The triangular window:

$$w_{tri}(n) = 1 - \frac{|2n - N + 1|}{N-1}, \quad 0 \leq n \leq N-1$$

The Hamming window:

$$w_{hm}(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

The Hanning window:

$$w_{hn}(n) = 0.5 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

clear ;

clc ;

close ;

N=11;

U=6;

```

h_hann = window ( 'hn' ,N ) ;

for n = -5+ U :1:5+ U

if n ==6

hd ( n) =0.75;

else

hd ( n)=( sin ( %pi *( n - U ) ) -sin( %pi *( n - U ) /4 ) ) /( %pi *( n - U ) ) ;

end

h ( n ) = h_hann ( n ) * hd ( n ) ;

end

[ hzm , fr ]= firzag ( h ,256) ;

hzm_dB = 20* log10 ( hzm ) ./ max ( hzm ) ;

figure

plot (2* fr , hzm_dB )

a = gca ();

xlabel ( 'Frequency w*pi' );

ylabel ( 'Magnitude in dB' );

title ( 'Frequency Response of FIR HPF with N =11 Using Hanning Window' );

xgrid (2) ;

```

✓ In the below figure the hanning window coefficients are computed in  $h\_hann(-5)$ ,  $h\_hann(-4)$ ,  $h\_hann(-3)$ ,  $h\_hann(-2)$ ,  $h\_hann(-1)$ ,  $h\_hann(0)$ ,  $h\_hann(1)$ ,  $h\_hann(2)$ ,  $h\_hann(3)$ ,  $h\_hann(4)$ ,  $h\_hann(5)$  as a total of 11 values ( $N=11$ ).

✓ And then the  $h_d(n)$  followed by  $h(n)$  (Multiplying  $h\_hann(n) * h_d(n)$ ).

```

--> N =11;

--> U =6;

--> h_hann = window ( 'hn',N )
h_hann =

    0.    0.0954915    0.3454915    0.6545085    0.9045085    1.    0.9045085    0.6545085    0.3454915    0.0954915    0.

--> for n = -5+ U :1:5+ U
> if n ==6
> hd ( n) =0.75;
> else
> hd ( n)=( sin ( %pi *( n - U ) ) -sin( %pi *( n - U ) /4 ) ) /( %pi *( n - U ) ) ;
> end
> h ( n ) = h_hann ( n ) * hd ( n ) ;
> end

--> [ hzm , fr ]= firzag ( h ,256) ;

-->

```

- ✓ “*frmag*” is used to find all the 256 points for the “ $\omega$ ” i.e.,  $h'e^{j\omega}$
- ✓ For Example, the value “0.0826497” corresponds to  $h'e^{j\omega}$  where  $\omega=0^\circ$  like wise its calculated for all 256 points of “ $\omega$ ”.
- ✓ Then its is converted in to dB values by “*hzm\_dB*”.
- ✓ Finally, the graph is plotted between “ $\omega$ ” and “*hzm\_dB*”.

```
--> [ hzm , fr ]= frmag ( h ,256)
hzm =

    column 1 to 15

0.0826497  0.0827782  0.0831637  0.0838059  0.0847044  0.0858585  0.0872675  0.0889303  0.0908457  0.0930125  0.0954291  0.0980939  0.101005  0.1041603  0.1075576

    column 16 to 30

0.1111947  0.115069  0.1191777  0.1235181  0.1280872  0.1328818  0.1378986  0.1431342  0.1485849  0.1542471  0.1601169  0.1661903  0.1724632  0.1789313  0.1855903

    column 31 to 45

0.1924357  0.1994628  0.2066667  0.2140435  0.2215874  0.2292936  0.2371571  0.2451728  0.2533353  0.2616393  0.2700795  0.2786504  0.2873464  0.296162  0.3050916

    column 46 to 60

0.3141295  0.3232701  0.3325074  0.3418359  0.3512496  0.3607429  0.3703098  0.3799446  0.3896414  0.3993944  0.4091977  0.4190456  0.4289322  0.4388518  0.4487987

    column 61 to 75

0.4587672  0.4687515  0.4787461  0.4887454  0.4987438  0.5087359  0.5187162  0.5286795  0.5386205  0.5485338  0.5584146  0.5682575  0.5780578  0.5878106  0.5975111

    column 76 to 90

0.6071545  0.6167364  0.6262522  0.6356976  0.6450683  0.6543602  0.6635692  0.6726914  0.6817231  0.6906605  0.6995001  0.7082384  0.7168723  0.7253984  0.7338138

    column 91 to 105
```

```

-> hzm_dB = 20* log10 ( hzm ) ./ max ( hzm )
hzm_dB =

    column 1 to 15

-21.52367 -21.510254 -21.470141 -21.403728 -21.311665 -21.19483 -21.05431 -20.891363 -20.707389 -20.503893 -20.282454 -20.044686 -19.792216 -19.526651 -19.249561

    column 16 to 30

-18.962457 -18.666782 -18.363896 -18.05507 -17.741485 -17.424229 -17.104296 -16.782592 -16.459935 -16.137061 -15.814627 -15.493221 -15.173361 -14.855505 -14.540055

    column 31 to 45

-14.22736 -13.917725 -13.611411 -13.308643 -13.009612 -12.714476 -12.42337 -12.136402 -11.853661 -11.575216 -11.301118 -11.031406 -10.766104 -10.505226 -10.248774

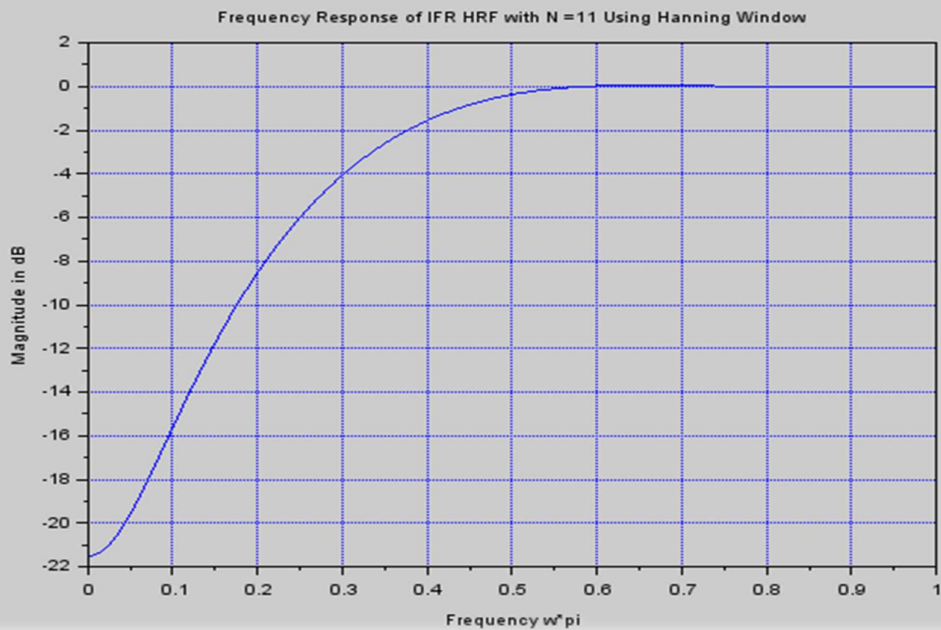
    column 46 to 60

-9.9967445 -9.7491237 -9.5058924 -9.2670258 -9.0324938 -8.8022625 -8.5762942 -8.3545483 -8.1369816 -7.9235489 -7.7142031 -7.5088959 -7.3075777 -7.1101982 -6.9167064

    column 61 to 75

-6.7270508 -6.5411797 -6.3590412 -6.1805835 -6.005755 -5.8345039 -5.6667791 -5.5025298 -5.3417053 -5.1842557 -5.0301315 -4.8792836 -4.7316637 -4.5872238 -4.4459167

```



*Fig.1: Magnitude response of HPF using Hanning Window*

Similarly Using Hamming Window we can obtain the HPF response

```
clear ;

clc ;

close ;

N=11;

U=6;

h_hamm = window ( 'hm' ,N ) ;

for n = -5+ U :1:5+ U

if n ==6

hd ( n )=0.75;

else

hd ( n )=( sin ( %pi *( n - U ) ) -sin( %pi *( n - U ) /4 ) ) /( %pi *( n - U ) ) ;

end

h ( n ) = h_hamm ( n ) * hd ( n ) ;

end

[ hzm , fr ]= frmag ( h ,256 ) ;

hzm_dB = 20* log10 ( hzm ) ./ max ( hzm );

figure

plot (2* fr , hzm_dB )

a = gca () ;

xlabel ( 'Frequency w*pi' ) ;

ylabel ( 'Magnitude in dB' ) ;

title ( 'Frequency Response of FIR HRF with N=11 Using Hamming Window' ) ;

xgrid (2) ;
```

*Design an LPF filter with*

$$H_d e^{j\omega} = e^{-j3\omega} \text{ for } -\pi/4 \leq \omega \leq \pi$$

*= 0 for  $\pi/4 \leq |\omega| \leq \pi$ . Find the values of  $h(n)$  for  $N=7$ . Find  $H(z)$ .*

*Plot the magnitude response using Hamming window and Hanning window techniques.*

*Scilab code for both Hanning and Hamming window Techniques*

```
clear all;

clc ;

close ;

N =7;

alpha =3;

U =1;

h_hann = window ( ' hn ' ,N ) ; // (h_hamm = window ( ' hm ' ,N ) ; ) for Hamming window

for n =0+ U :1:6+ U

if n ==4

hd ( n ) =0.25;

else

hd ( n ) =( sin ( %pi * ( n -U - alpha ) /4 ) ) / ( %pi * ( n -U - alpha ) ) ;

end

h ( n ) = hd ( n ) * h_hann ( n ) ; // (h ( n ) = h_hamm ( n ) * hd ( n ) ; ) For Hamming window

end

[ hzm , fr ] = frmag ( h ,256 ) ;

hzm_dB = 20* log10 ( hzm ) ./ max ( hzm ) ;

figure

plot ( 2* fr , hzm_dB )

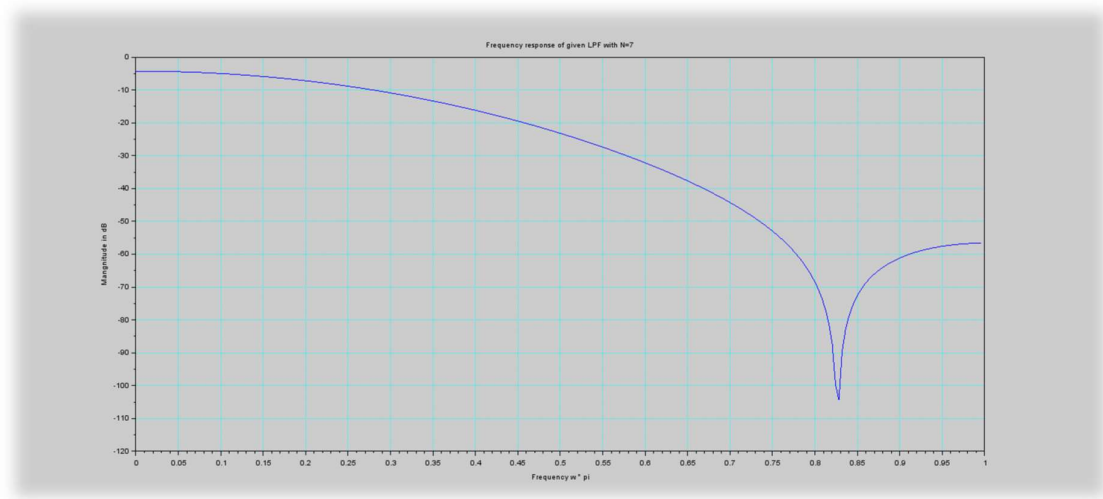
a = gca ( ) ;

xlabel ( 'Frequency w * pi' ) ;

ylabel ( 'Mangnitude in dB' ) ;
```

title ('Frequency response of given LPF with N=7') ;

xgrid (2) ;



*Fig.2: Magnitude response of LPF filter using Hamming Window*

**Pre-lab:** Design of digital FIR Band Pass Filter using Hanning and Hamming window

**Post-lab:** Design of digital FIR Band Stop filter using Hanning and Hamming window

**Lab 9:** Design of digital FIR filter using frequency sampling method

*Problem.1: Determine the filter coefficients  $h(n)$  obtained by sampling*

$$H_d e^{j\omega} = e^{-j(N-1)\omega/2} \text{ for } 0 \leq |\omega| \leq \pi/2$$

$$= 0 \text{ for } \pi/2 \leq |\omega| \leq \pi. \text{ For } N=7.$$

```
clear ;
clc ;
close ;
N=7;
U=1; // Zero Adjust
for n=0+U :1: N -1+ U
h ( n ) =(1+2* cos (2* %pi *( n -U -3) /7) ) / N
end
disp (h , "Filter Coefficients , h(n)=")
```

```
--> close ;
--> N =7;
--> U =1; // Z e r o A d j u s t
--> for n =0+ U :1: N -1+ U
> h ( n ) =(1+2* cos (2* %pi *( n -U -3) /7) ) / N
> end
h =
-0.1145625
h =
-0.1145625
0.0792797
h =
-0.1145625
0.0792797
0.3209971
h =
-0.1145625
0.0792797
0.3209971
0.4285714
```

It computes its coefficients like in the iteration: 1  $h(0)$   
iteration:2  $h(0)$  and  $h(1)$   
iteration:3  $h(0)$  ,  $h(1)$  and  $h(2)$  goes on till  $h(6)$

```
--> disp (h , "Filter Coefficients, h(n)=")
```

```
-0.1145625
0.0792797
0.3209971
0.4285714
0.3209971
0.0792797
-0.1145625
```

BY using "disp" comment we get all the 7 coefficients alone shown in the console.

```
"Filter Coefficients, h(n)="
```



### Pre-lab Question

1. Find the values of  $h(n)$  for  $N=11$ . (For problem.1)

### Post-lab question

1. What is the general formula to find the values of  $H(k)$ ,  $\theta(k)$  and  $h(n)$  using frequency sampling method?
2. List the steps in defining the “ $k$ ” values of frequency sampling method.