جامعة أبوظبي
ABU DHABI UNIVERSITY

ABU DHABI UNIVERSITY

CEN 464 - DIGITAL SIGNAL PROCESSING

Lab Report 5
IIR & FIR Filter Design using MATLAB

Authors:
Muhammad Obaidullah 1030313

Supervisor:
Dr. Mohammed Assad Ghazal

Section 1

June 9, 2014

Contents

Abstract

In this lab we were introduced to different filter design using MATLAB toolboxes, complex features, and advanced functions. Advanced MATLAB tools such as fdatool GUI were used to design the Finite Impulse Response Filters. The windows which were used are Kaiser, Hamming and Hanning. The advantage of designing a filter using MATLAB is simplicity and very easy tuning of the filter parameters. With the click of few buttons, the filter parameters an be changed to get a whole new design and even the computational powers of MATLAB can help the user tune certain parameters to get the desired results. Finally, in this lab we will compare and contrast between all of these methods of filter design and give our results.

1 Objectives of the lab

- Understand and recognize the parameters needed for designing the FIR filter.
- Be able to design an Finite Impulse Response filter using windowing techniques.
- Be able to design and tune an Finite Impulse Response filter using MATLAB FDA Tool.
- Have a clear distinction between the FIR and IIR filter designs.

2 List of equipment used

- A Computer.
- MATLAB.
- Filter Design & Analysis Toolbox.

3 FIR Filter Design By Windowing Techniques

In the first part of the lab, we will be designing different filters by the traditional windowing techniques. The general steps are to first determine the specifications of the filter using the equations related to the filter and then inputting the information in the MATLAB via a code for simulating the output.

MATLAB provides several functions to easily design and implement window filters. A brief list is given below: [1]

- **boxcar(M)** returns the M-point rectangular window function in array w.
- **triang(M)** returns the M-point Bartlett (triangular) window function in array w.
- **hanning(M)** returns the M-point Hanning window function in array w.
- **hammimg(M)** returns the M-point Hamming window function in array w.
- **blackman(M)** returns the M-point Blackman window function in array w.

3.1 Hamming Window

3.1.1 Ideal Lowpass Impulse Response $h_d(n)$

To start designing the lowpass FIR filters, an Ideal low-pass impulse response should be obtained first.

```
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;
n = [0:1:(M-1)]
M = n     alpha + eps;  % add smallest number to avoid divide by zero
hd = sin(wc*M) ./ (pi*M);
```

3.1.2 To Display Frequency domain of a filter

This function returns the magnitude response in absolute as well as in relative dB scale, the phase response, and the group delay response.

```
function [db, mag, pha, grd, w] = freqz_m(b, a)
% Modified version of freqz sunroutine
% ------------------------------------------------------
% [db,mag,pha,grd,w] = freqz_m(b, a)
%  db = Relative magnitude in dB computed over 0 to pi radians
% mag = absolute magnitude computed over 0 to pi radians
% pha = Phase response in radians over 0 to pi radians
% grd = group delay over 0 to pi radians
%   w = 501 frequency samples between 0 to pi radians
%   b = numerator polynimial of H(z)   (for FIR: b=h)
%   a = denominator polynomial of H(z) (for FIR: a=[1])
%
[H,w] = freqz(b,a,1000,  whole  );
H = (H(1:1:501))   ;
w = (w(1:1:501))   ;
mag = abs(H);
db = 20*log10((mag+eps)/max(mag));
pha = angle(H);
grd = grpdelay(b,a,w);
% End of function
```

3.1.3 Given Specifications

$w_p = 0.3\pi, w_s = 0.4\pi, R_p = 0.25dB, A_s = 50dB$

We entered the following code and the result produced was saved.

```
wp = 0.3*pi; ws = 0.4*pi;
tr_width = ws-wp;
M = ceil(6.6*pi/tr_width)+1;
n = [0:1:M-1];
wc = (ws+wp)/2;
hd = ideal_lp(wc,M);
w_ham = (hamming(M))';
h = hd .* w_ham;
[db,mag,pha,grd,w] = freqz_m(h,[1]);
% plots
subplot(1,1,1)
subplot(2,2,1); stem(n,hd); title('Ideal Impulse Response')
axis([0 M-1 -0.1 0.3]); xlabel('n'); ylabel('hd(n)')
subplot(2,2,2); stem(n, w_ham); title('Hamming Window')
axis([0 M-1 0 1.1]); xlabel('n'); ylabel('w(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')
```

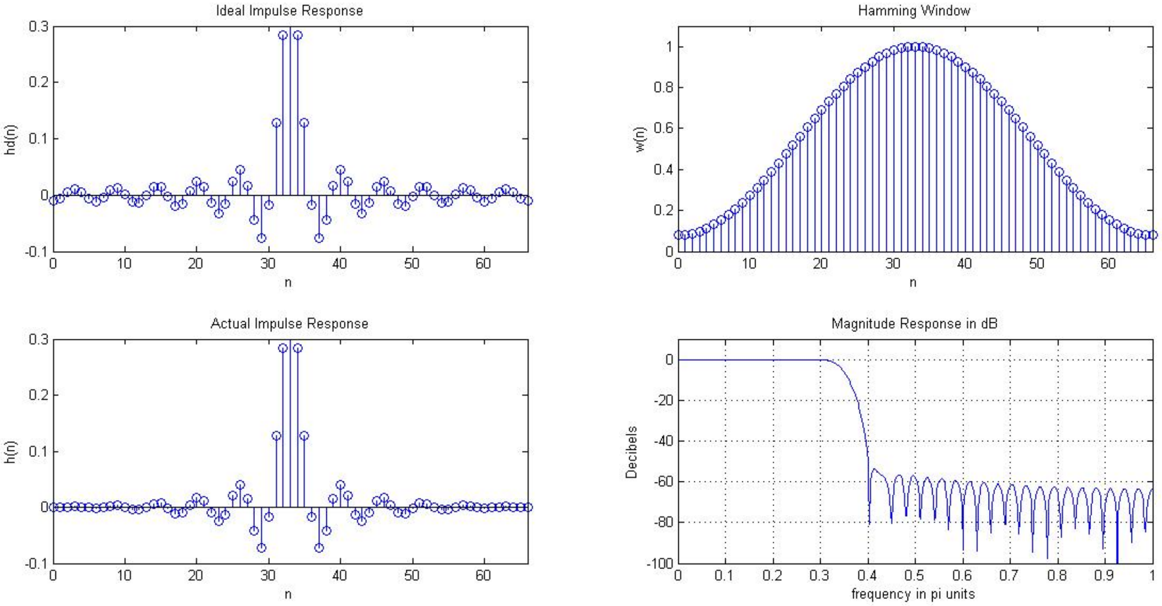
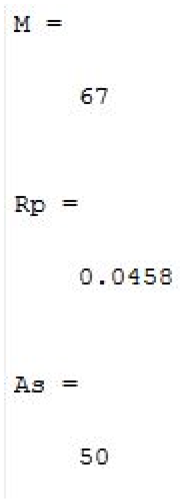


Figure 1: This figure was generated after running the code for given specifications. The figure shows the ideal filter impulse response, the hamming window applied filter impulse response, the actual hamming window in time domain and the magnitude response of the hamming window.

3.1.4 Finding parameters for the designed filter

Find the value of M, R_p, and A_s of the designed filter in the procedure

```
% For Determining size of filter. (Filter Taps)
M = ceil(6.6*pi/tr_width)+1;
% Print Size of Filter
M
delta_w = 2*pi/1000;
% Actual Passband Ripple
Rp = -(min(db(1:wp/delta_w+1)))
% Min Stopband attenuation
As = -round(max(db(ws/delta_w+1:501)))
```



```
M =

    67


Rp =

    0.0458


As =

    50
```

Figure 2: MATLAB calculated all the design parameters and the results is shown. Basically, the filter size (Filter Taps) will be 67, pass-band ripple will be 0.0458, and the minimum stop-band Attenuation will be -50dB.

3.2 Kaiser Window

3.2.1 Design the FIR high-pass filter using Kaiser window.

Given Specifications: $w_p = 0.3\pi, w_s = 0.4\pi, A_s = 50dB$

```
% Filter Specifications
wp = 0.3*pi;
ws = 0.4*pi;
tr_width = ws-wp;

% Finding Filter Essential Parameters
M = ceil((As-7.95)/(2.285*tr_width)+1) + 1;
n = [0:1:M-1];
wc = (ws+wp)/2;
hd = ideal_lp(pi, M)-ideal_lp(wc,M);

% Since As > 50
Beta = 0.1102*(As-8.7)

% Building the Filter
Kaiser_Window = (kaiser(M,Beta))';
```

```
h = hd .* Kaiser_Window;
[db,mag,pha,grd,w] = freqz_m(h,[1]);
delta_w = 2*pi/1000;
As = -floor(max(db(1:1:(wp/delta_w)+1)))

% Plotting the Results
subplot(2,2,1);
stem(n,hd);
title('Ideal Impulse Response')
axis([0 M-1 -0.5 0.5]);
xlabel('n');
ylabel('hd(n)')
subplot(2,2,2);
stem(n,Kaiser_Window);
title('Kaiser Window Graph')
axis([0 M-1 0 1.1]);
xlabel('n');
ylabel('w(n)')

subplot(2,2,3);
stem(n,h);
title('Actual Impulse Response');
axis([0 M-1 -0.5 0.5]);
xlabel('n');
ylabel('h(n)');
subplot(2,2,4);
plot(w/pi,db);
axis([0 1 -130 10]);
title('Magnitude Response in dB');
grid on;
```

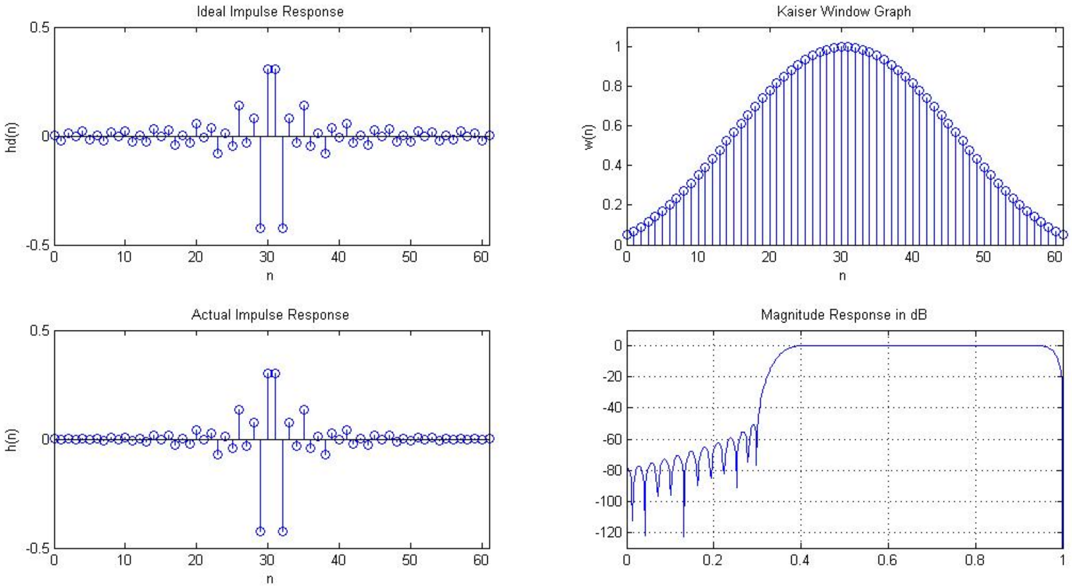


Figure 3: As we can see from the figure that the high pass Kaiser filter was designed successfully meeting the design specifications.

3.3 Band Reject Filter

3.3.1 The Code:

```
wp1=0.35*pi; ws1=0.2*pi;
wp2=0.65*pi; ws2=0.8*pi;
 %only one transition bandwidth
% allowed in window design
 tr_width=min(wp1-ws1,ws2-wp2);
  M=ceil(11*pi/tr_width)+1;
n=[0:M-1];
wc1=(ws1+wp1)/2; %ideal cutoff frequency 1
wc2=(ws2+wp2)/2; %ideal cutoff frequency 2
  hd=ideal_lp(pi,M)-ideal_lp(wc2,M)-ideal_lp(wc1,M);
w_blackman=(blackman(M))';
h=hd.*w_blackman;
  figure(1);stem(n,h);  title('h(n)')
 figure(2);freqz(h,[1])
```

3.3.2 Result:

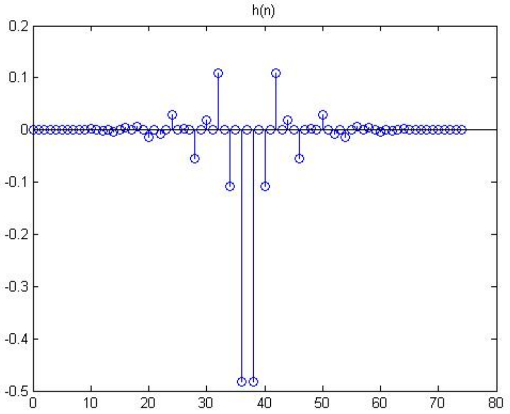


Figure 4: The impulse response of the band reject filter.

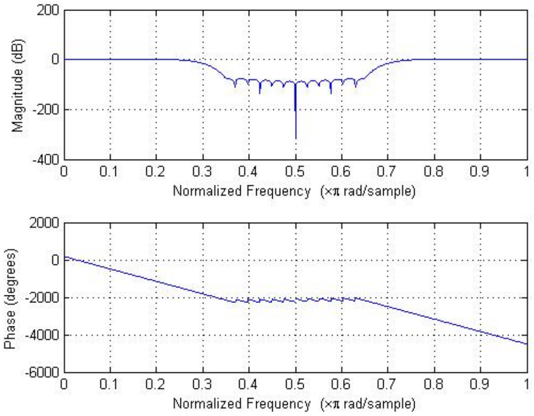


Figure 5: The magnitude response of the band reject filter and the phase response. We can clearly see that the phase is linear in pass-bands, which is acceptable.

3.4 Band Pass Filter

3.4.1 The Code:

```
wp1=0.35*pi;  ws1=0.2*pi;
wp2=0.65*pi;  ws2=0.8*pi;
 %only one transition bandwidth
% allowed in window design
tr_width=min(wp1-ws1,ws2-wp2);
 M=ceil(11*pi/tr_width)+1;
n=[0:M-1];
wc1=(ws1+wp1)/2; %ideal cutoff frequency 1
wc2=(ws2+wp2)/2; %ideal cutoff frequency 2
 hd= ideal_lp(wc2,M)-ideal_lp(wc1,M);
w_blackman=(blackman(M))';
h=hd.*w_blackman;
 figure(1);stem(n,h);  title('h(n)')
figure(2);freqz(h,[1])
```

3.4.2 Result:

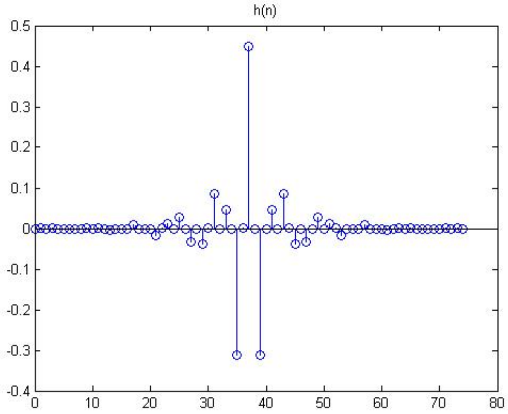


Figure 6: The impulse response of the band pass filter.

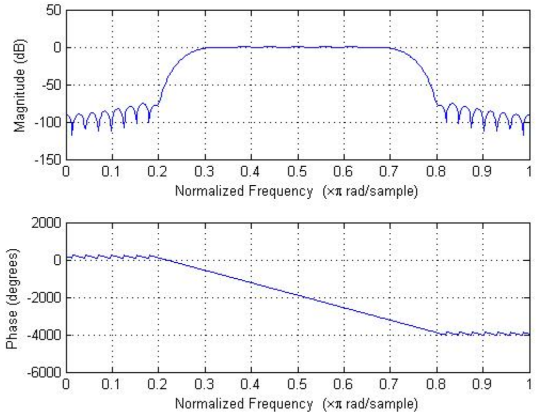


Figure 7: The magnitude response of the band pass filter and the phase response. We can clearly see that the phase is linear in pass-bands, which is acceptable.

4 FIR Filter Design By MATLAB FDA Tool

Filter Design and Analysis Toolbox in MATLAB is a very powerful but user-friendly GUI filter designer which can be used to design almost all popular filters very easily and conveniently. Different windowing techniques such as Hanning, Hamming, Bartlett, Rectangular, Blackman etc. can be applied to band-pas, band-reject, low-pass, high-pass etc. So all in all, it offers much flexibility when it comes to choosing the right filter and implementing it and it is a real time saver for a practicing engineer.

4.1 Specifications of the Desired Filter

- Low-pass Filter
- $f_{cut-off} = 10Khz$
- $F_{sampling} = 50Khz$
- $M = 30$

4.1.1 Rectangular Window

Indeed, the attenuation was -20dB in the stopband and the transition band was very sharp. The phase response is linear in the pass-band and the filter has a constant group delay, meaning that all frequencies get delayed by the same amount as we can see from the figure.

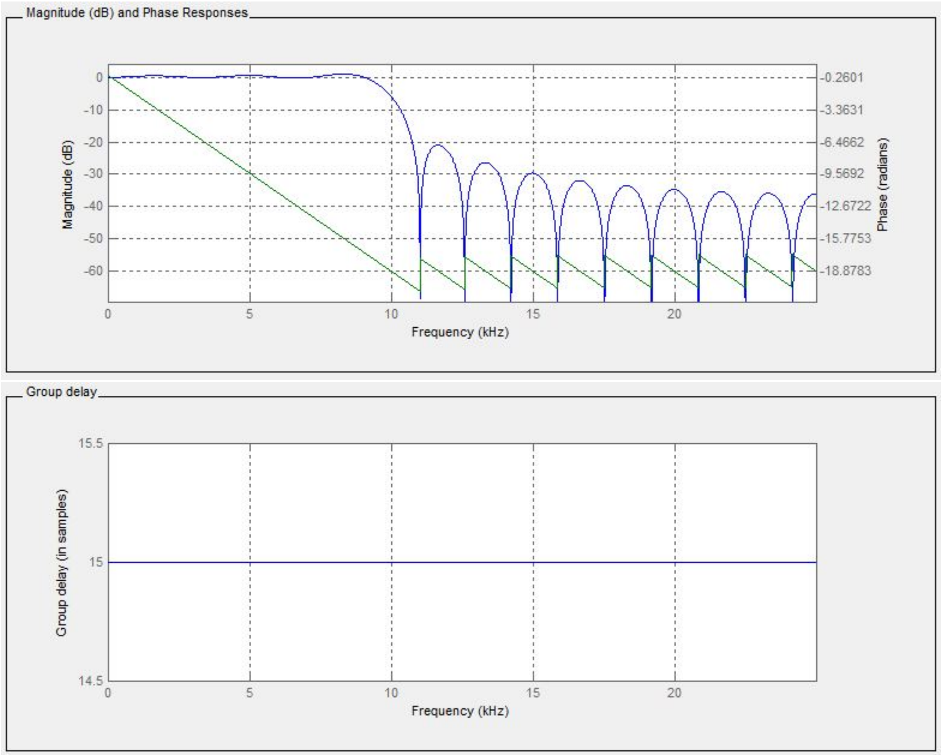


Figure 8: The magnitude response, phase response and group delay of the low pass filter designed using FDA Tool in MATLAB with rectangular window.

4.1.2 Hanning Window

Indeed, the attenuation was below -40dB in the stopband as expected of hanning window but the transition band was not sharp. The phase response is linear in the pass-band and the filter has a constant group delay, meaning that all frequencies get delayed by the same amount as we can see from the figure.

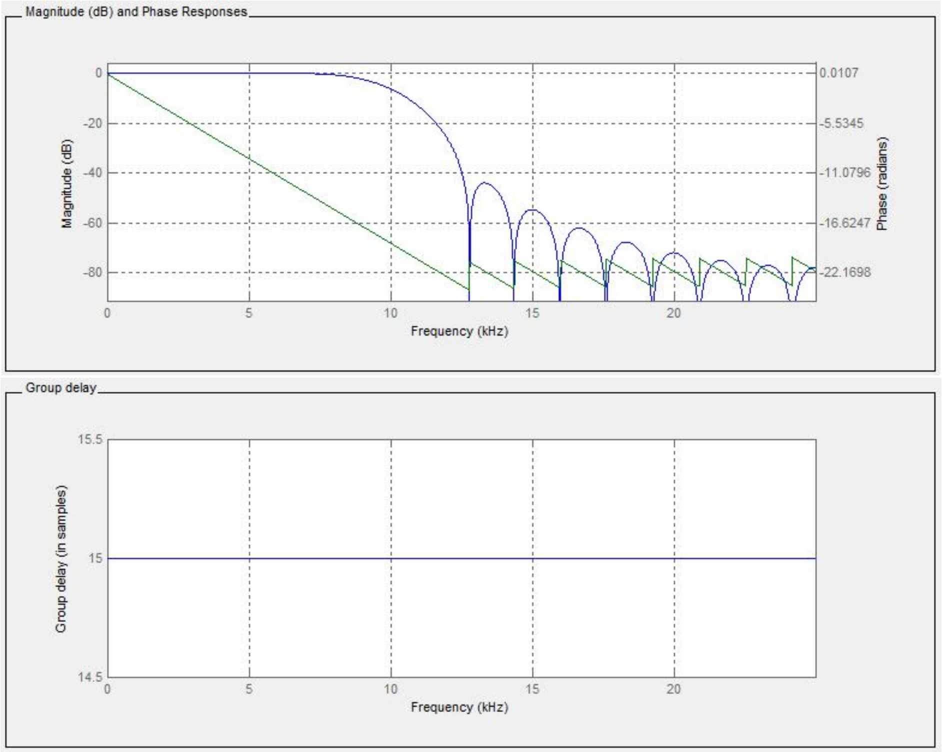


Figure 9: The magnitude response, phase response and group delay of the low pass filter designed using FDA Tool in MATLAB with hanning window.

4.1.3 Hamming Window

Indeed, the attenuation was below -60dB in the stopband as expected of hamming window but the transition band was even more smooth than rectangular or hanning. The phase response is linear in the pass-band and the filter has a constant group delay, meaning that all frequencies get delayed by the same amount as we can see from the figure.

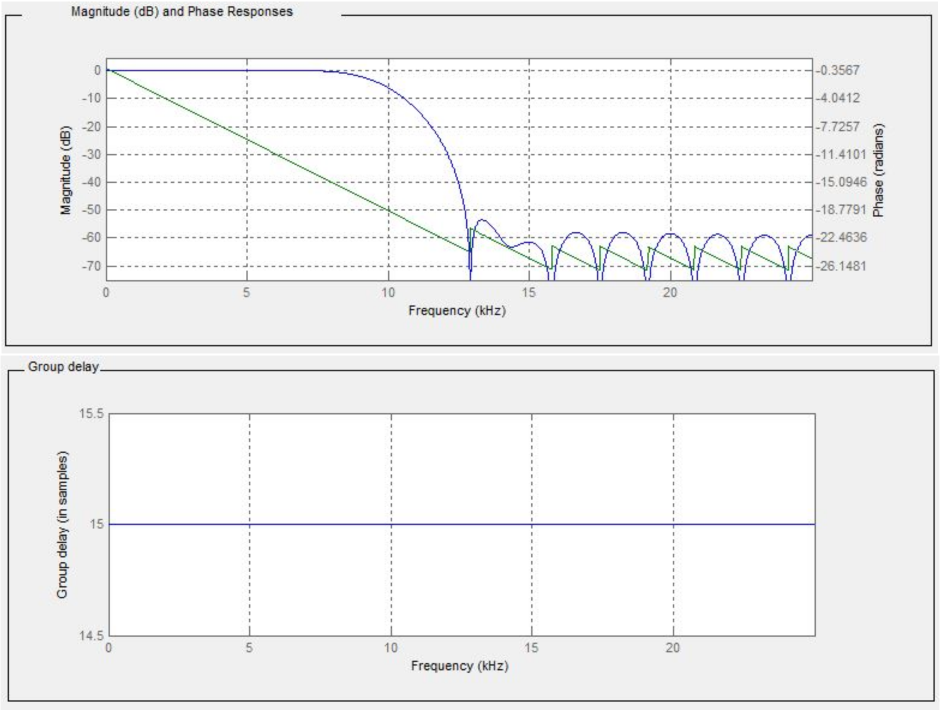


Figure 10: The magnitude response, phase response and group delay of the low pass filter designed using FDA Tool in MATLAB with hamming window.

4.1.4 Blackman Window

Indeed, the attenuation was below -80dB in the stopband as expected of blackman window but the transition band was even more smooth than all. The phase response is linear in the pass-band and the filter has a constant group delay, meaning that all frequencies get delayed by the same amount as we can see from the figure.

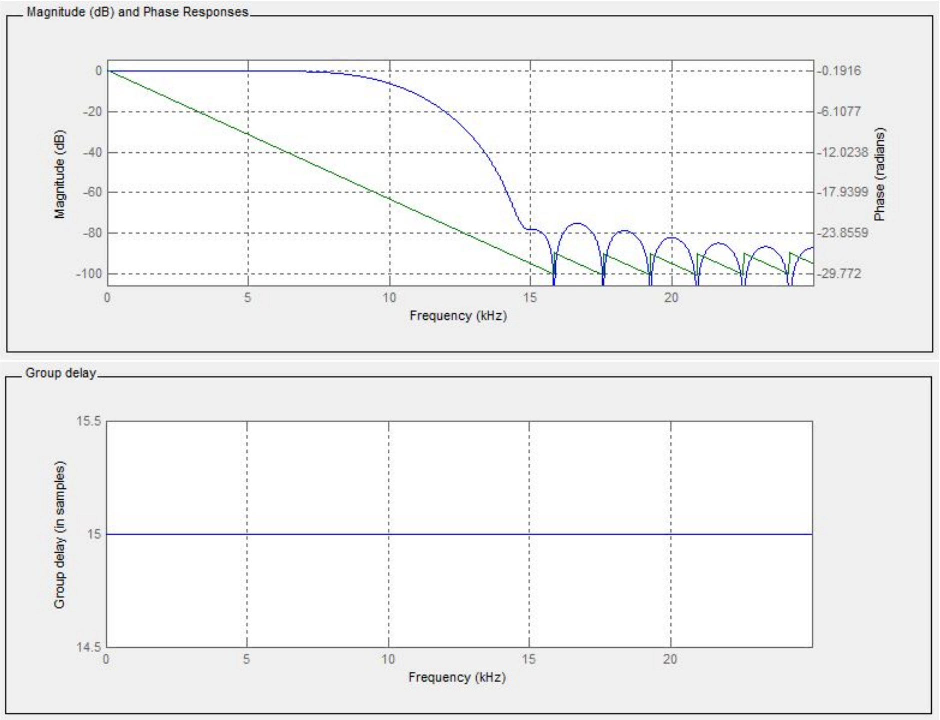


Figure 11: The magnitude response, phase response and group delay of the low pass filter designed using FDA Tool in MATLAB with blackman window.

5 IIR Filter Design By MATLAB FDA Tool

5.0.5 Butterworth Filter

Indeed, the attenuation was below -80dB at about 11KHz but the transition was very smooth. The phase response is almost linear in the pass-band but at 8KHz started to get non-linear. The filter does not have a constant group delay, meaning that all frequencies do not get delayed by the same amount as we can see from the figure. The frequency 10KHz experiences largest group delay of about 50 samples.

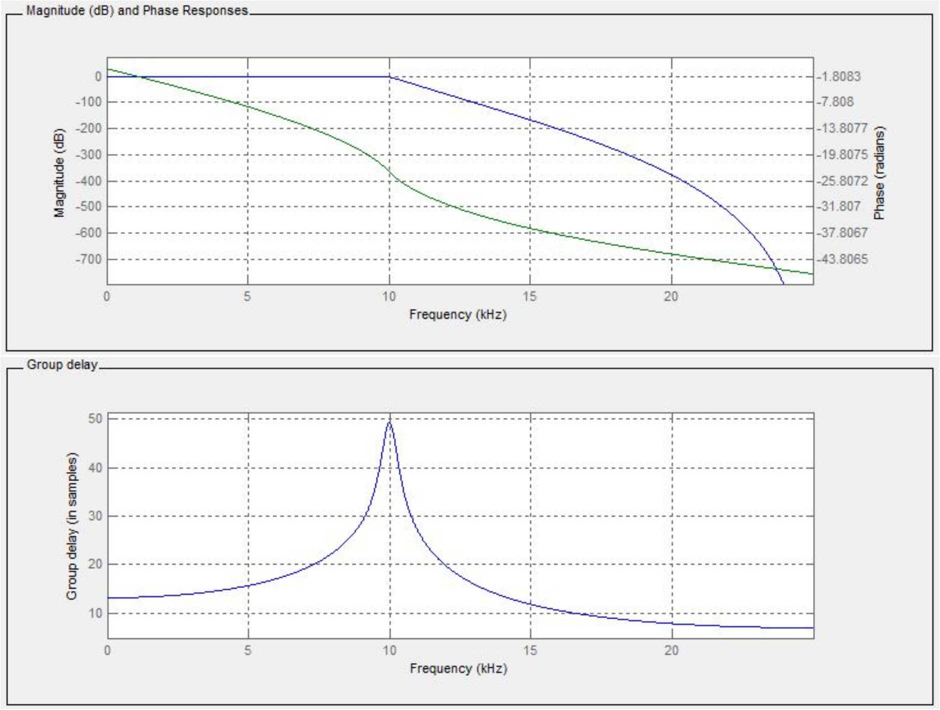


Figure 12: The magnitude response, phase response and group delay of the butterworth low pass filter designed using FDA Tool in MATLAB.

5.0.6 Chebyshev Type I Filter

Indeed, the attenuation was below -80dB at about 11KHz but the transition was more sudden than butterworth. The phase response is almost linear in the pass-band but at 7KHz started to get non-linear. The filter has a improvement in the group delay region from butterworth but does not have a constant group delay, meaning that all frequencies do not get delayed by the same amount as we can see from the figure. The frequency 10KHz experiences largest group delay of about 500 samples.

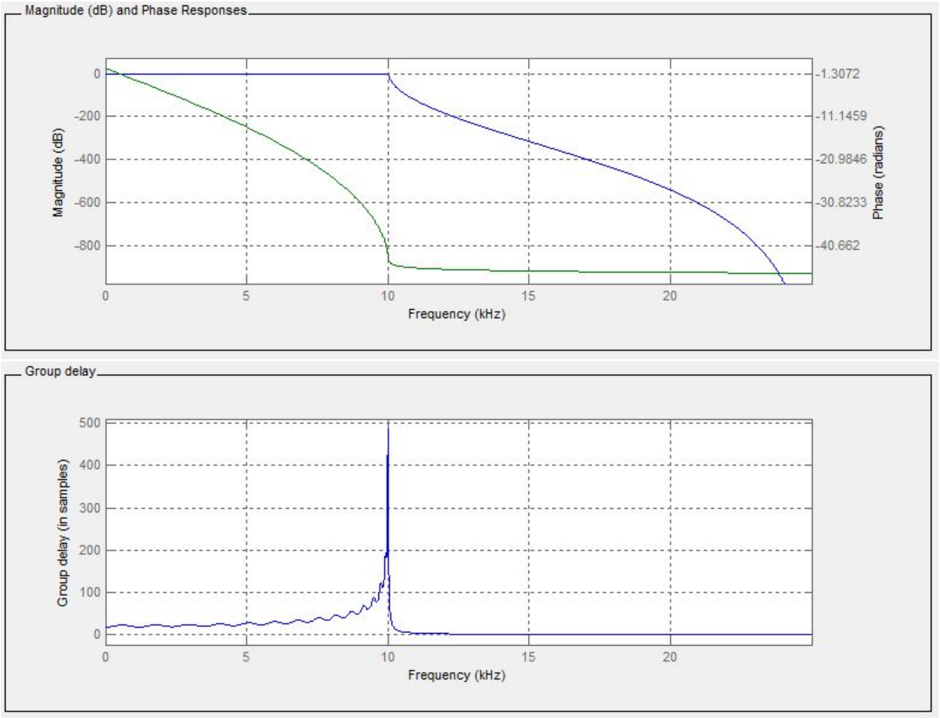


Figure 13: The magnitude response, phase response and group delay of the Chebyshev Type I low pass filter designed using FDA Tool in MATLAB.

5.0.7 Chebyshev Type II Filter

Indeed, the attenuation was below -80dB at about 10KHz but the transition was more sudden than butterworth and chebyshev type I. The phase response is almost linear in the pass-band but at 6KHz started to get non-linear. The filter has a non-constant group delay, meaning that all frequencies do not get delayed by the same amount as we can see from the figure. The frequency 9KHz experiences largest group delay of about 120 samples.

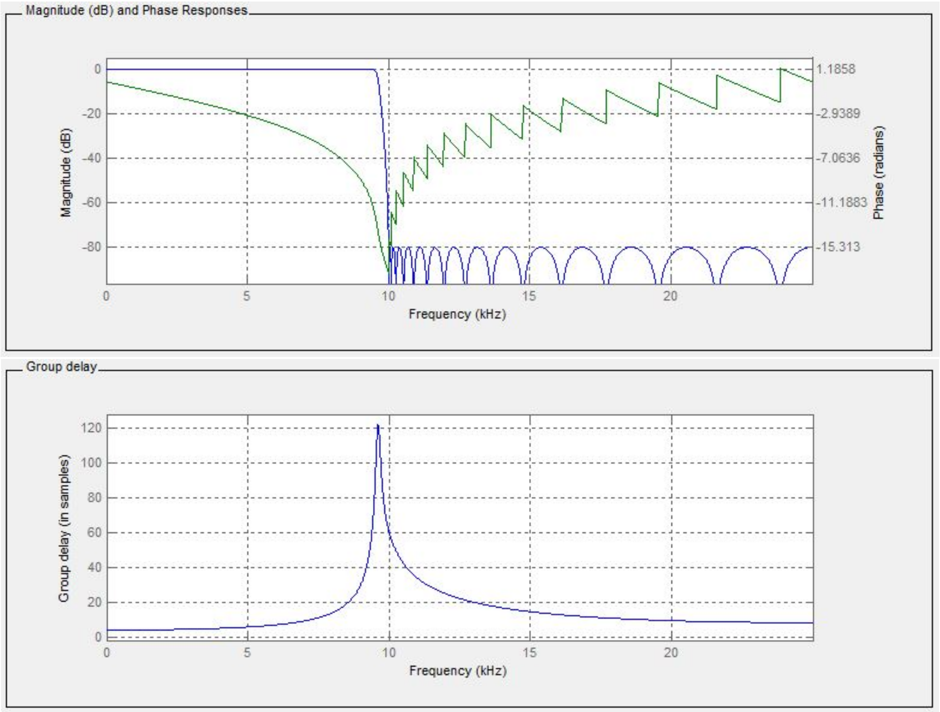


Figure 14: The magnitude response, phase response and group delay of the Chebyshev Type II low pass filter designed using FDA Tool in MATLAB.

5.0.8 Elliptic Filter

Indeed, the attenuation was below -80dB at 10Khz and the transition was very very sudden. The phase response is almost linear in the pass-band but at 7KHz started to get non-linear. The filter has a improvement in the group delay region from all other IIR filters, but has gigantic group delay of the frequency 10Khz. The frequency 10KHz experiences largest group delay of about 9000 samples.

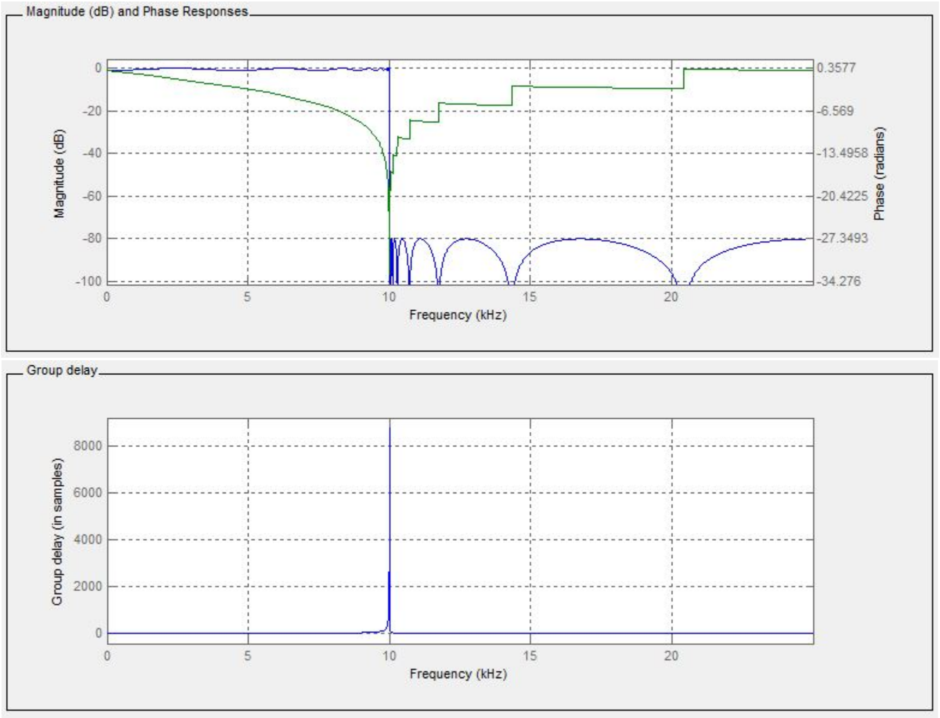


Figure 15: The magnitude response, phase response and group delay of the Elliptic low pass filter designed using FDA Tool in MATLAB.

6 Conclusion

All in all, there were many conclusions to draw from this extensive analysis of the filter designs.

- FIR filters have constant group delay while IIR filter do not usually have constant group delays.
- Butter worth filter has the least ripples in the pass-band or stop band from all other IIR filters but lacks in sudden transition from pass-band to stop-band or vice versa.
- Chebyshev I filter has ripples in pass-band while the Chebyshev II filter has ripples in stop band.
- Elliptic filter has the most sudden transition from the pass-band to stop-band and the most constant group delay but does not have a linear phase for frequencies near the transition band.
- Elliptic filters have ripples in both, the stop-band and the pass-band as well.

References

[1] Abu Dhabi University, *FIR Filter Design Using Window Techniques*, `http://blackboard.adu.ac.ae/bbcswebdav/pid-225732-dt-content-rid-1676568_1/xid-1676568_1` [9 June 2014].

[2] Allan Oppenheim *Discrete-Time Signal Processing*, 3rd Edition USA: PEARSON, 2010 [9 June 2014].

[3] Fourier Series and Transform, *Why Fourier transform*, `http://www.engr.uconn.edu/~lanbo/G377FFTYC.pdf` [9 June 2014].

[4] What is a Matrix?, *A Video Screen is One Type of Matrix*, `http://cycling74.com/docs/max5/tutorials/jit-tut/jitterwhatisamatrix.html` [9 June 2014].