National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering and Computer Science

Faculty Member: Dr. Ahmad Salman Dated: 24/03/2023

Semester: 6th Section: BEE 12C

EE-330 Digital Signal Processing

Lab 9: FIR Filter design using Windowing

Group Members

		PLO4 - CLO4		PLO5 - CLO5	PLO8 - CLO6	PLO9 - CLO7
Name	Reg. No	Viva / Quiz / Lab Performance	Analysis of data in Lab Report	Modern Tool Usage	Ethics and Safety	Individual and Teamwork
		5 Marks	5 Marks	5 Marks	5 Marks	5 Marks
Danial Ahmad	331388					
Muhammad Umer	345834					
Tariq Umar	334943					

1 Table of Contents

2 FIR Filter design using Windowing ... 3

2.1 Objectives ... 3

2.2 Introduction ... 3

2.3 Software ... 3

2.4 Lab Report Instructions ... 3

3 Lab Procedure ... 4

3.1 Lab Task 1 ... 4

3.2 Lab Task 2 ... 6

3.3 Lab Task 3 ... 7

4 Conclusion ... 9

2 FIR Filter design using Windowing

2.1 Objectives

The principal objective of this lab is to demonstrate FIR Filter design using windowing. In addition, linear phase filters will also be demonstrated

2.2 Introduction

A finite impulse response (FIR) filter is a digital filter whose impulse response has a finite duration. FIR filters are typically used in applications where sharp cutoffs are required, such as in audio filtering and image processing. There are many different methods for designing FIR filters. One common method is windowing. Windowing is a method of multiplying a desired frequency response by a window function. The window function is a function that has a finite duration and a zero-phase response. The product of the desired frequency response and the window function is the frequency response of the windowed filter. The windowed filter will have the same desired frequency response as the original filter, but it will have a different impulse response. Linear phase filters are FIR filters that have a constant phase response over their passband. Linear phase filters are often used in applications where phase distortion is to be minimized, such as in audio processing and speech recognition.

2.3 Software

MATLAB is a high-level programming language and numerical computing environment. Developed by MathWorks, it provides an interactive environment for numerical computation, visualization, and programming. MATLAB is widely used in various fields, including engineering, science, and finance, due to its capabilities for matrix and vector operations, implementation of algorithms, and creation of graphical representations of data. The objective of this lab is to provide a hands-on experience with the A-to-D sampling and the D-to-A reconstruction processes that are essential for digital image processing. We will also demonstrate a commonly used method of image zooming (reconstruction) that produces "poor" results, which will help illustrate the importance of understanding the underlying principles of digital image processing.

2.4 Lab Report Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

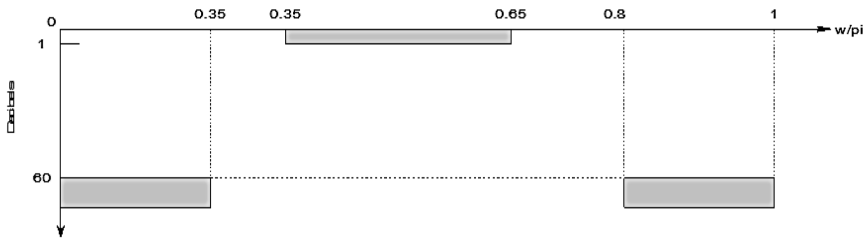
- Lab objectives
- MATLAB codes
- Results (graphs/tables) duly commented and discussed
- Conclusion

3 Lab Procedure

3.1 Lab Task 1

Design the following digital bandpass filter:

$$\begin{aligned} &\textit{lower stopband edge}: \quad w_{1s} = 0.2\pi, \quad A_s = 60dB \\ &\textit{lower passband edge}: \quad w_{1p} = 0.35\pi \quad R_p = 1dB \\ &\textit{upper passband edge}: \quad w_{2p} = 0.65\pi, \quad R_p = 1dB \\ &\textit{upper stopband edge}: \quad w_{2s} = 0.8\pi \quad A_s = 60dB \end{aligned}$$



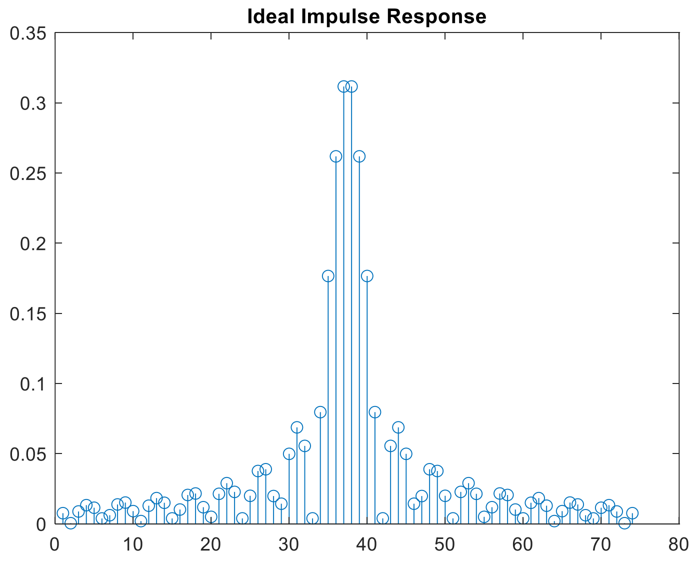
Follow the following steps:

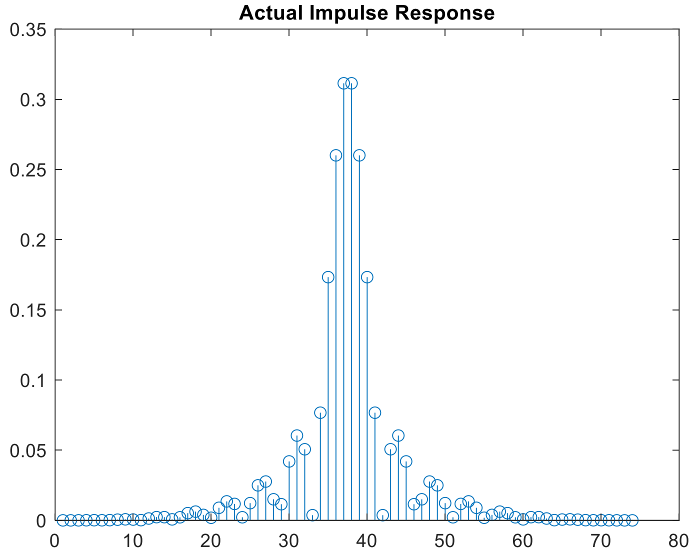
Find transition width

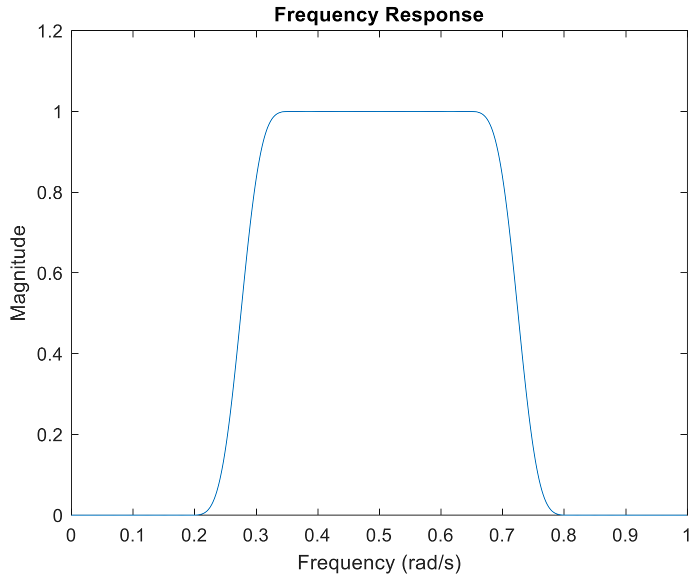
- Calculate M
- Create ideal band pass filter with two ideal low pass filter
- Implement appropriate window (use the table)
- Multiply window coefficient with your ideal band pass filter
- Plot the ideal impulse response
- Plot actual impulse response
- Plot frequency response

```
% Define the parameters
wp1 = 0.35*pi;
ws1 = 0.2*pi;
wp2 = 0.65*pi;
ws2 = 0.8*pi;
tranband = wp1 - ws1;

% Calculate the filter length
M = ceil((11*pi) / tranband);

% Calculate the cutoff frequencies
wc1 = (ws1+wp1)/2;
wc2 = (ws2+wp2)/2;
```

```
% Create the ideal low-pass filters
hd1 = ideal_lp(wc1, M);
hd2 = ideal_lp(wc2, M);

% Create the bandpass filter
hd = hd2 - hd1;

% Apply the Blackman window
wn = blackman(M)';
h = wn .* hd;

% Plot the ideal impulse response
figure
stem(abs(hd))
title("Ideal Impulse Response")

% Plot the actual impulse response
figure
stem(abs(h))
title("Actual Impulse Response")

% Plot the frequency response
figure
[H, w] = freqz(h, 1, 1024);
plot(w/pi, abs(H))
title("Frequency Response")
xlabel("Frequency (rad/s)")
ylabel("Magnitude")
```





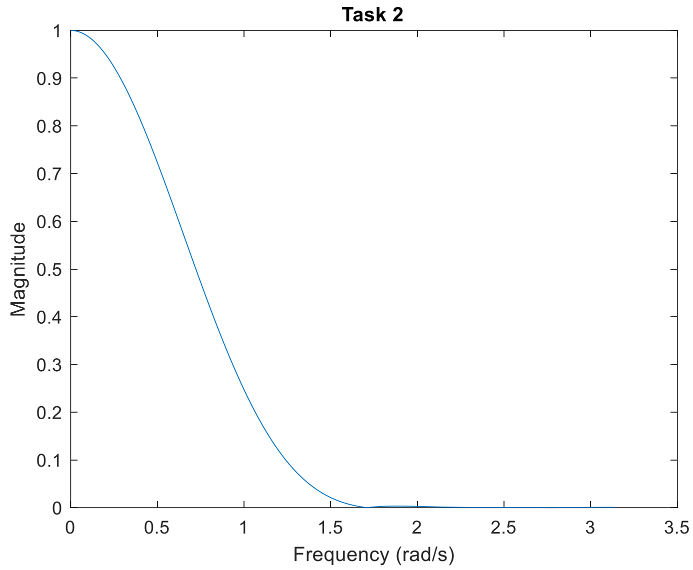


3.2 Lab Task 2

Design a lowpass FIR filter with a passband cutoff frequency of 1 kHz, a stopband edge at 4.3 kHz, and a sampling frequency of 10 kHz. We will use a Hamming window. The transition width is the difference between the stopband edge and the passband edge. Thus, the normalized transition width del f is the number of coefficients can then be estimated. We must now determine the normalized cutoff frequency for MATLAB.

```
% Define the parameters
fp = 1000;
fst = 4300;
Fs = 10000;
trans = fst - fp;

% Calculate the transition width
```

```
del_w = (2*pi*trans)/Fs;

% Calculate the filter length
M = ceil(6.6*pi/del_w);

% Create the filter
b = fir1(M, 1000/5000);

% Calculate the frequency response
[H, w] = freqz(b, 1, 1024);

% Plot the frequency response
figure
plot(w, abs(H))
xlabel("Frequency (rad/s)")
ylabel("Magnitude")
title("Task 2")
```



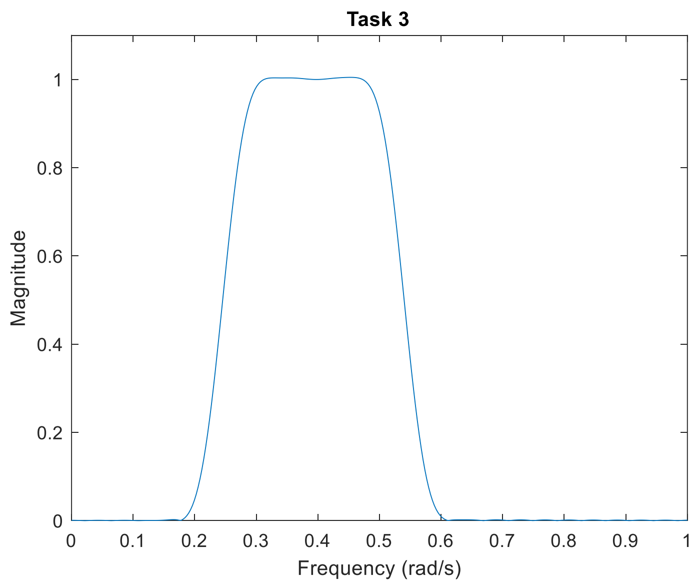
3.3 Lab Task 3

Design Bandpass Filter with following specification using fir1 function.

- Fs = 48 kHz
- Passband Cutoff Frequencies = 8 kHz & 16 kHz
- Stopband Edge Frequencies = 7 kHz & 17 kHz
- Hamming Window

```
% Define the parameters
fp1 = 8000;
fs1 = 7000;
```

```
fp2 = 16000;
fs2 = 17000;
Fs = 48000;
trans = fp1 - fs1;

fc1 = (fp1 + fs1)/2;
fc2 = (fp2 + fs2)/2;

% Calculate the transition width
del_w = (2*pi*trans)/Fs;

% Calculate the filter length
M = ceil(6.6*pi/del_w);

% Create the filter
b = fir1(M, [fc1/Fs/2, fc2/Fs/2]);

% Calculate the frequency response
[H, w] = freqz(b, 1, 1024);

% Plot the frequency response
figure
plot(w, abs(H))
xlabel("Frequency (rad/s)")
ylabel("Magnitude")
title("Task 3")
axis([0 1 0 (1+0.1)])
```



4 Conclusion

In this lab, we have demonstrated the design of FIR filters using windowing. We have seen that windowing is a simple and effective method for designing FIR filters. We have also seen that linear phase filters can be designed using windowing. The main advantage of windowing is that it is a very simple method to implement. The main disadvantage of windowing is that it can introduce some distortion into the frequency response. This distortion can be minimized by using a window function with a good stopband attenuation. In general, windowing is a good choice for designing FIR filters when simplicity and speed are important considerations. However, if high performance is required, other methods of designing FIR filters, such as least squares design, may be a better choice.