Adama Science and Technology University

School of Electrical Engineering and Computing

ELECTRONICS AND COMMUNICATION DEPARTMENT

Digital Signal Processing Project

Title:- *Band Pass FIR Digital Filtering*

<u>Section 6</u>

Name	*ID*
1. Yohannes Melese	***UGR/17831/11***
2. Habtamu Tadesse	***UGR/17174/11***
3. Rediet Manyazewal	***UGR/17572/11***
4. Tamirat Tesfaye	***UGR/17691/11***
5. Belete Andargie	***UGR/16844/11***

Date of submission 21/05/2020

Submitted to:- Mr. Kirubel

What is Filter?

In signal processing, a filter is a device or process that removes some unwanted components or features from a signal. Filtering is a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal. Most often, this means removing some frequencies or frequency bands.
One way to classify filters is:-

- Analog Filter
- Digital Filter

Digital Filter

Digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. This is in contrast to the other major type of electronic filter, the analog filter, which is typically an electronic circuit operating on continuous-time analog signals.

A digital filter designed to pass signal components of certain frequencies without any distortion should have a frequency response of value equal to one at these frequencies and should have a frequency response of value equal to zero at all other frequencies to totally block signal components with those frequencies.

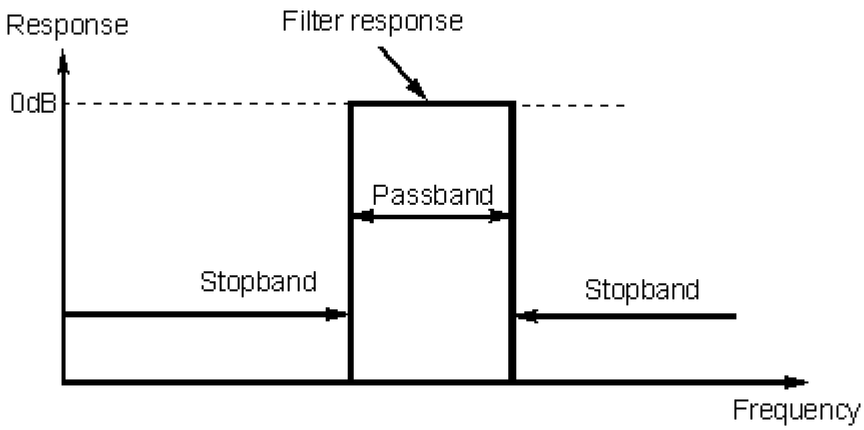
There are four common types of digital filters:-

- Low pass digital filter
- High pass digital filter
- Band pass digital filter
- Band stop digital filter

lets see what Band-Pass filter is

Band-pass Digital Filter

The ideal band pass digital filter has a unit magnitude response and zero phase response.
The filter passes the frequency components that are in a given frequency range and attenuates those frequencies outside the frequency range.



We use FIR digital filters to approximate this band pass filter having a linear phase response with finite impulse response. This can be achieved by designing techniques like windowing and frequency sampling.

Designing of Digital Filters

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

- **Specifications**: they are determined by the applications
- **Approximations**: once the specification are defined, we use various concepts and mathematics to come up with a filter description that approximates the given set of specifications.
- **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).

Q# Design an FIR band pass filter to filter the audio signal attached with the document. The specification of the required filter are:-

$f_s = 13\,K\,Hz\,;$

$f_{c1} = 2000\,Hz\,;$

$f_{c2} = 6000\,Hz\,;$

$\Delta f = 286\,Hz\,;$

1. The frequency domain plot of the audio signal, X (K).

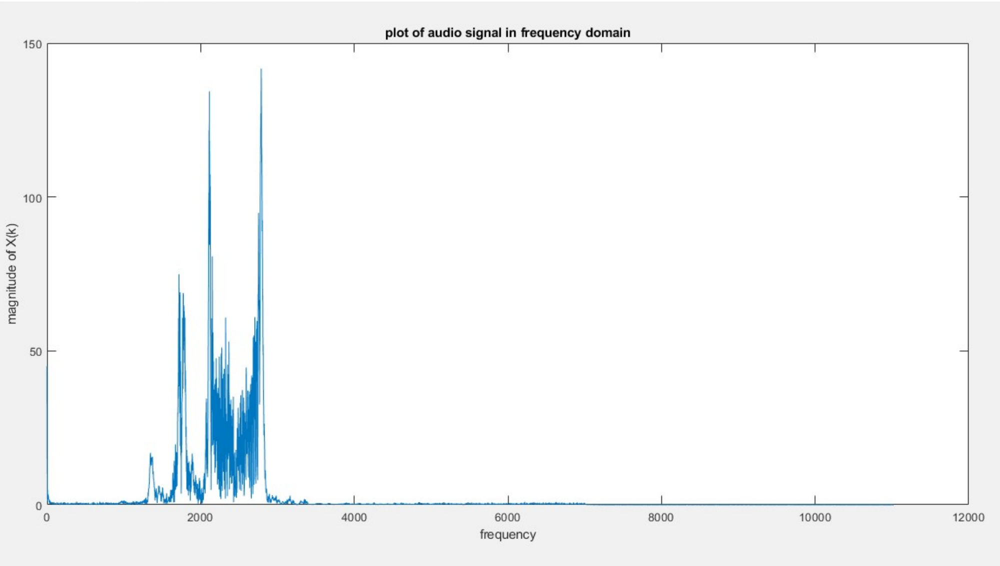
The given audio signal is in the working directory named “bird_chrip.wav”

The MATLAB code is:-

```
Fs = 13000;                    % sampling frequency

% loading the audio signal
[x, Fso] = audioread("bird_chirp.wav");
xf = linspace(0, Fso, Fs);
mf = abs(fft(x, Fs));

% ploting the figure
plot(xf(1:Fs/2), mf(1:Fs/2));
title("plot of audio signal in frequency domain");
xlabel("frequency")
ylabel("magnitude of X(k)")
```



2. The time domain and frequency domain ($H(e^{j\omega})$) plot of the windowing function.

The given Stop band attenuation factor is -70db this is best approximated by blackman window which has stop -74db

- Blackman window

This is a 2-order raised cosine window.

$$w(n)=\left[0.42-0.5\cos\left(\frac{2\pi n}{N-1}\right)+0.08\cos\left(\frac{4\pi n}{N-1}\right)\right]R_N(n)$$

$$W(\omega)\approx 0.42W_R(\omega)+0.25\left[W_R(\omega-\frac{2\pi}{N})+W_R(\omega+\frac{2\pi}{N})\right]$$
$$+0.04\left[W_R(\omega-\frac{4\pi}{N})+W_R(\omega+\frac{4\pi}{N})\right]$$
$$(N>>1)$$

But first we have to calculate the order of the filter and group delay constant alpha

$$\textbf{Fn} = \frac{\textbf{Fr}}{\textbf{Fs}}$$

where fn - normalized frequency
fr - normal frequency
fs – sampling frequency

We have the transition bandwidth Δf = 286hz
Δf in normalized form will be 286 / 13000 = 0.022
Δw = 2πΔf = 0.138
|As| = 70

The filter order can be approximated by

$$N = \frac{As - 8}{2.285.\Delta w}$$

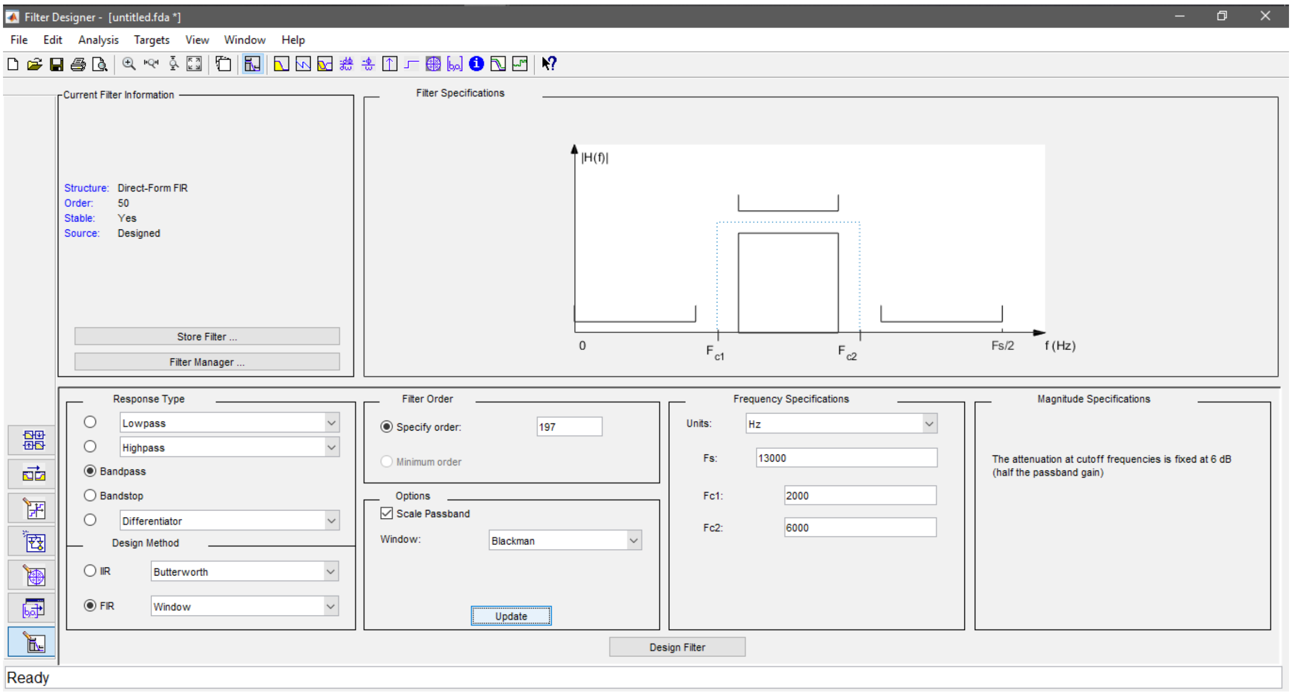
so N = 70 – 8 / 2.285 * 0.138 = 197
the group delay constant alpha calculated as

$$\alpha = (N-1) / 2$$

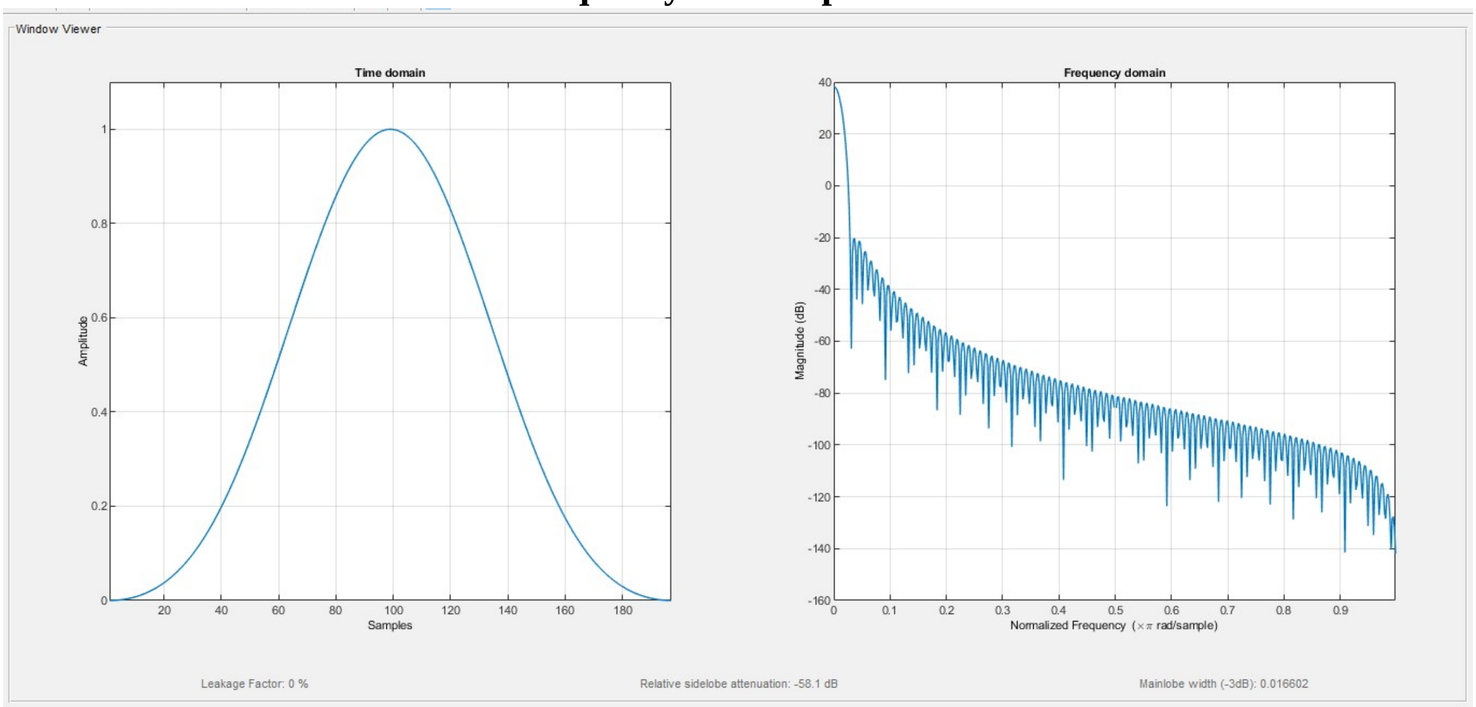
so α = (197-1) / 2 = 98

in MATLAB we have the command called ***filterDesigner*** which is used to design digital filters. We use this function to design our window and filter. We can have all types of plot like magnitude plot phase plot etc can be drawn by using this MATLAB tool.

By using filter order N = 197



The time domain and frequency domain plot of the window function is



3. Frequency domain plot of w(n)h(n) or equivalently (H($e^{j\omega}$)* W($e^{j\omega}$))

$$H_d(e^{j\omega}) = \begin{cases} 1\cdot e^{-j\alpha\omega}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}$$

$$h_d(n) = F^{-1}\left[H_d(e^{j\omega})\right] = \frac{1}{2\pi}\int_{\omega_c}^{\omega_c} e^{-j\alpha\omega} e^{j\omega n}\, d\omega = \frac{\omega_c}{\pi}\,\frac{\sin[\omega_c(n-\alpha)]}{\omega_c(n-\alpha)}$$

$$w(n) = R_N(n)$$

$$h(n) = h_d(n)w(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

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

$$h(n) = \begin{cases} \dfrac{\omega_c}{\pi}\,\dfrac{\sin\left[\omega_c\left(n-\frac{N-1}{2}\right)\right]}{\omega_c\left(n-\frac{N-1}{2}\right)}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

The function is designed for low pass filter but this can be modified and used it to design band pass filter

i.e $$\mathbf{h_d(n)} = \frac{\sin(w_{c1}(n-\alpha)) - \sin(W_{c2}(n-\alpha))}{\pi(n-\alpha)}$$

the MATLAB code to implement the above function and plot h(n) in time domain and frequency domain is

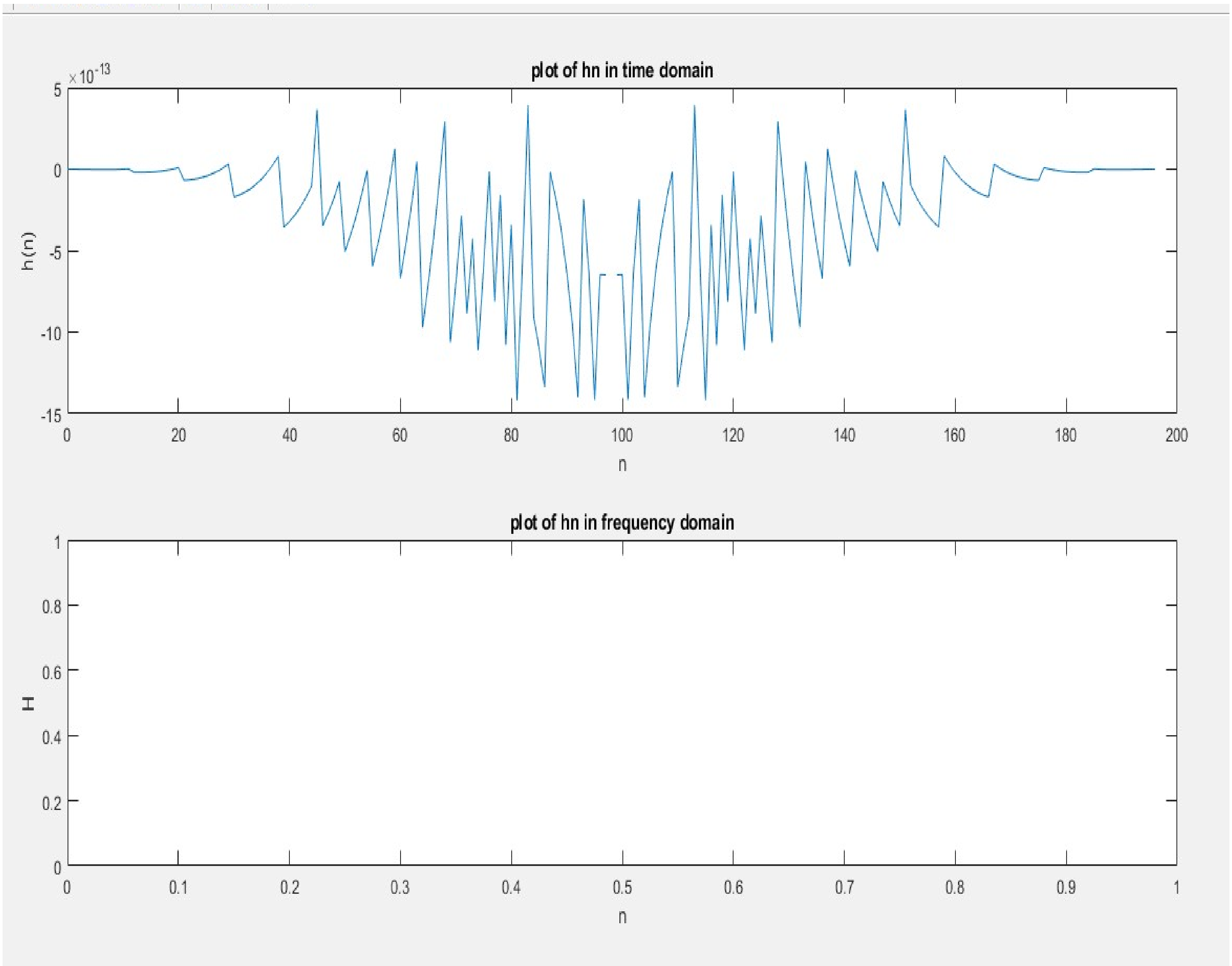
```
N = 197;              % order of the filter
aph = (N-1) / 2;      % the shift factor
Fs = 13000;           % sampling frequency
fc1 = 2000;           % low cutoff frequency
fc2 = 6000;           % higher cutoff frequency

n = [0:N-1];
wc1 = fc1*2*pi;
wc2 = fc2*2*pi;

hd = (sin(wc2*(n-aph)-sin(wc1*(n-aph)))) ./ (pi*(n-aph));   %hd(n)
wn = blackman(N);  % w(n)
hn = hd .* wn';    % h(n)
```

```
subplot(2, 1, 1)
plot(n, hn)
title("plot of hn in time domain")
xlabel("n")
ylabel("h(n)")

subplot(2, 1, 2)
h = freqz(hn,1,w);
plot(w/pi, abs(h));
title("plot of hn in frequency domain")
xlabel("n")
ylabel("H")
```



4. Use the designed filter to filter the audio signal.

After designing the filter we can use either convolution in time domain or multiplication in frequency domain.
But here in MATLAB we use ***fir1()*** function to generate the numerator and denominator which are used in the function ***filter()***

The MATLAB code is:-

```
% Create the window vector for the design algorithm.
N = 197;
fc1 = 2000;
fc2 = 6000;
Fs = 13000;
[x, Fso] = audioread("bird_chirp.wav");
win = blackman(N+1);
% Calculate the coefficients using the FIR1 function.
b  = fir1(N, [fc1 fc2]/(Fs/2), 'bandpass', win, 'scale');
y = filter(b, 1, x);
```

here y will store the filtered signal output in time domain.

In the files attached to the document there is a MATLAB function named ***filteringfunction(x)*** which accepts the original signal x as input and gives the result which is the filtered signal y

so alternatively we can use this function to filter the original signal

```
y =  filteringfunction(x)
```

finally to save the filtered audio we use the ***audiowrite()*** function

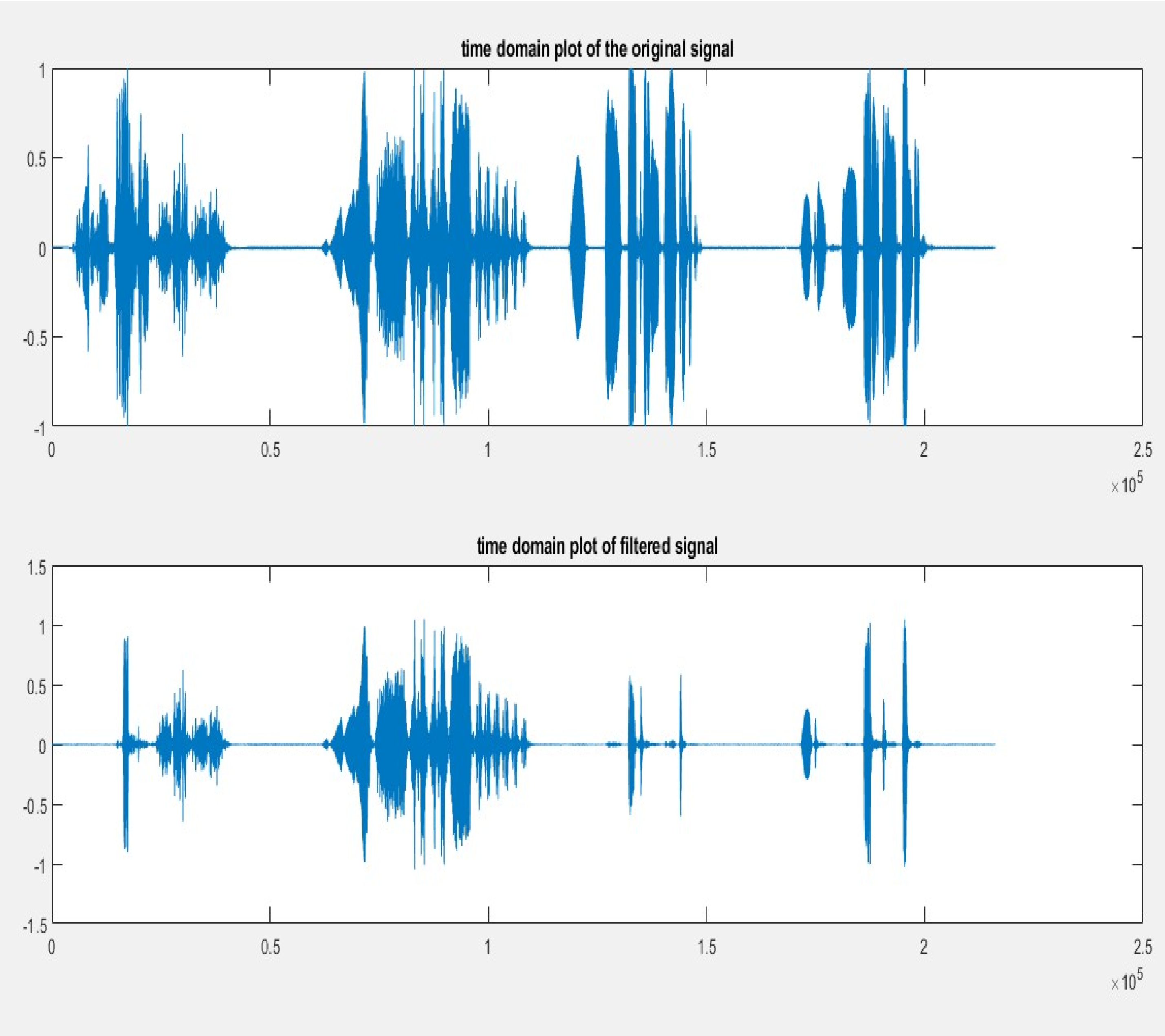
```
audiowrite('filtered.wav', y, Fso);
```

In the files attached the file named “filtered.wav” is the filtered audio signal.

5. Compare the original signal with the filtered (plot the frequency domain) one and comment on the difference. Listen to the two audio files and feel the difference.

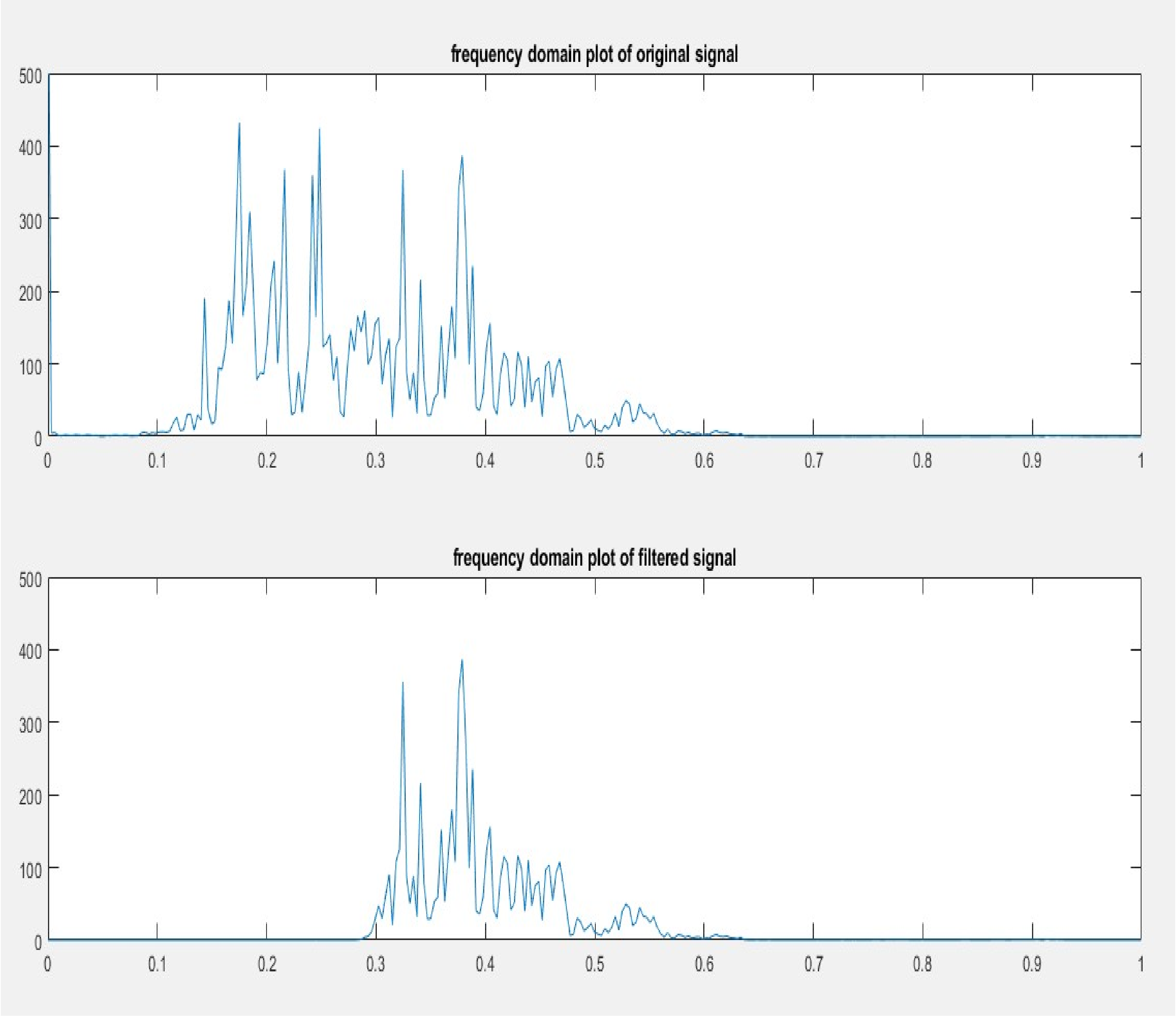
First lets us compare the original audio signal and the filtered audio signal in time domain.

```
subplot(2, 1, 1)
plot(x)
title('time domain plot of the original signal')
subplot(2, 1, 2)
title('time domain plot of filtered signal')
plot(y)
```



we can also plot both signals in frequency domain

```
subplot(2, 1, 1)
plot(w/pi, abs(freqz(x, 1, w)))
ylim([0 500])
title('frequency domain plot of original signal')
subplot(2, 1, 2)
plot(w/pi, abs(freqz(y, 1, w)))
ylim([0 500])
title('frequency domain plot of filtered signal')
```



<u>Conclusion</u>

First, in this project we were given an audio signal and filtered the signal in a ***passband*** FIR digital filter with the given specification. We used the windowing method to truncate the time domain impulse response function *hn* with a window named ‘***blackman***’ window.

Comparing the two audio signals i.e the original audio signal ‘*bird_chrip.wav*’ represented by x and the filtered audio signal ‘*filtered.wav’* represented by y, the original signal has both high frequency and low frequency signals which are disturbing for the hearing but in the filtered audio signal some of the frequencies chopped off from the original audio signal and makes less disturbing noises. This can clearly be seen from the frequency domain plot of the two signals.

So we can use bandpass digital FIR filters to pass a frequencies in a given range only.