DSP LAB Assignment

EE22BTECH11022 – GARIKAPATI SAI HARSHITH

Aim: To simulate a low pass and high pass filter to filter out the given music signal and to observe it in a spectrogram (of Matlab).

Explanation:

1. We have to convert the audio file of .wav or .mp3 format into data (array format) by using audioread function in matlab.

2. For the design of a **Low-Pass FIR filter**, a cutoff frequency f_c is specified, below which signals remain unchanged, and above which signals are attenuated. The ideal impulse response $h_d(n)$ of a low-pass FIR filter is given by a sinc function with a normalized cutoff frequency w_c. This ideal response is then multiplied by a chosen window function $w_H(n)$, typically the Hamming window, resulting in the final impulse response $h_{lp}(n)$.

$$h_d(n) = \begin{cases} \dfrac{\sin\,(w_c n)}{\pi n} & -\dfrac{N-1}{2} \le n \le \dfrac{N-1}{2} \\ \dfrac{w_c}{\pi} & n = 0 \end{cases}$$

Here, as f_c is 1000Hz. f_s of the audio file given is 22050. This indicates there are 22050 samples per second. Therefore, $[-11025{,}11025]$ indicates w is $[-\ \pi, \pi]$. To filter out samples from $[-1000{,}1000]$, w required is $\left[-\frac{\pi}{11.025}, \frac{\pi}{11.025}\right]$. So.

$$w_c = \frac{\pi f_c}{11025} = \frac{\pi}{11.025}$$

3. For the design of a **High-Pass FIR filter**, a cutoff frequency f_c is specified, below which signals remain unchanged, and above which signals are attenuated. The ideal impulse response $h_d(n)$ of a low-pass FIR filter is given by a sinc function with a normalized cutoff frequency w_c. This ideal response is then multiplied by a chosen window function $w_H(n)$, typically the Hamming window, resulting in the final impulse response $h_{lp}(n)$.

$$h_d(n) = \begin{cases} \dfrac{\sin\,(\pi n)}{\pi n} - \dfrac{\sin\,(w_c n)}{\pi n} & -\dfrac{N-1}{2} \le n \le \dfrac{N-1}{2} \\ \dfrac{\pi - w_c}{\pi} & n = 0 \end{cases}$$

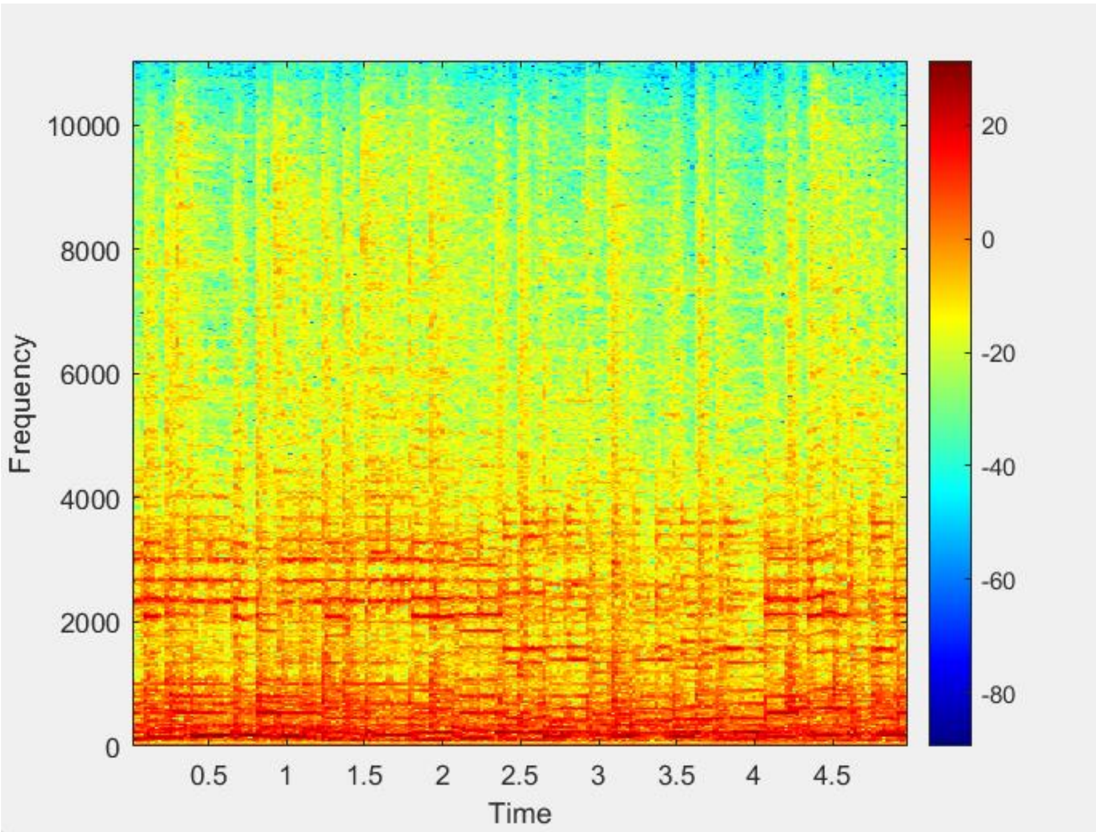
Here, as f_c is 5000Hz. f_s of the audio file given is 22050. This indicates there are 22050 samples per second. Therefore, $[-11025{,}11025]$ indicates w is $[-\ \pi, \pi]$. To filter out samples from $[-1000{,}1000]$, w required is $\left[-\frac{5\pi}{11.025}, \frac{5\pi}{11.025}\right]$. So.

$$w_c = \frac{\pi f_c}{11025} = \frac{5\pi}{11.025}$$

4. Convolution of input and h_d(impulse response of filter) gives output of filter.

5. A spectrogram is a visual representation of the frequency content of a signal as it changes over time. It is a 2D plot where the horizontal axis represents time, the vertical axis represents frequency, and the intensity of the color or shading represents the magnitude or power of each frequency component at a particular time.

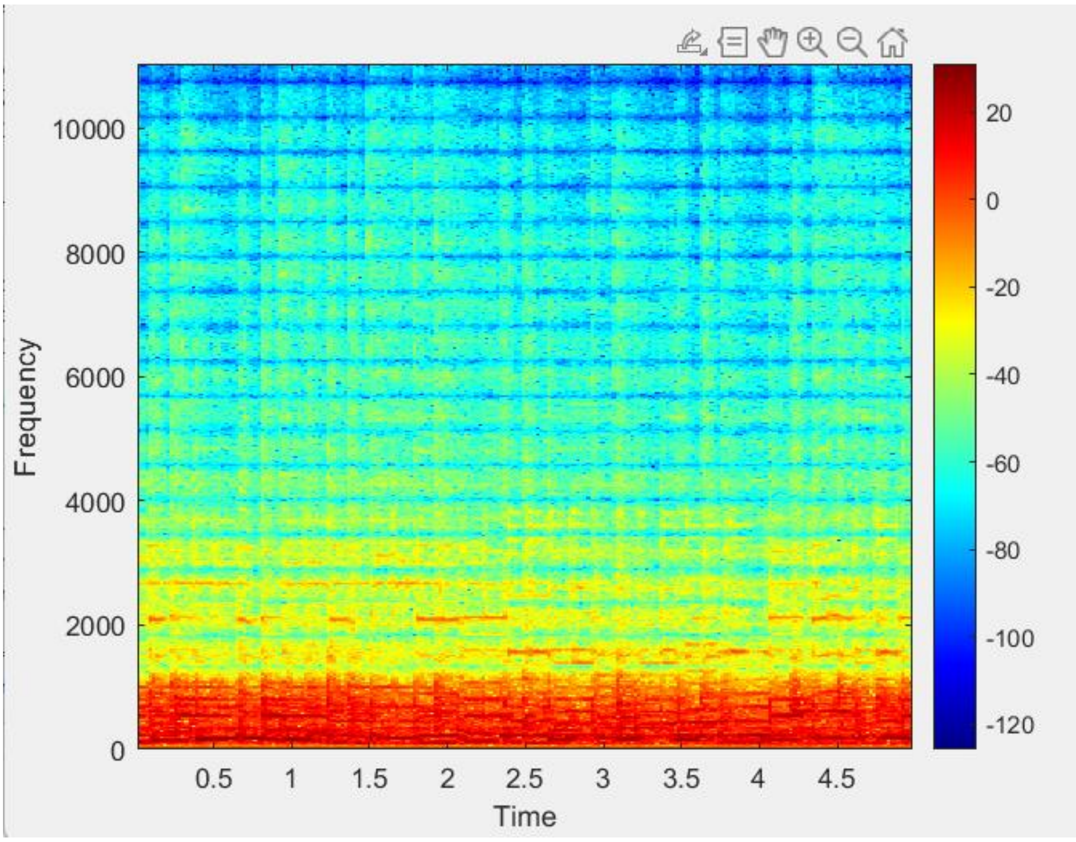
Input spectrogram plot:



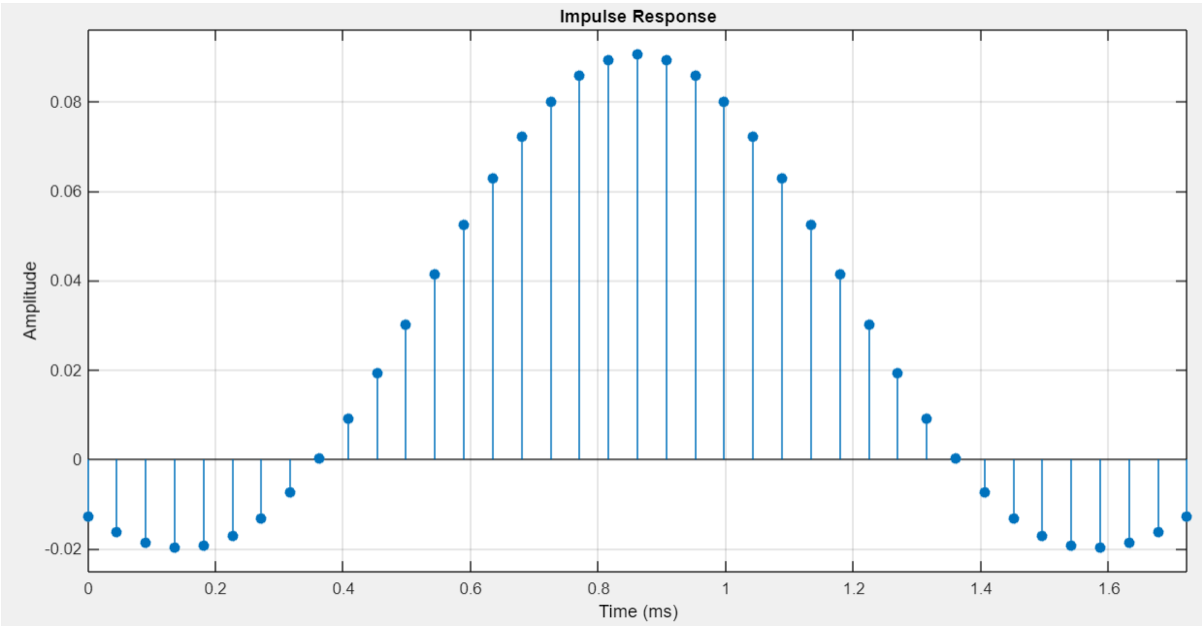
Low pass filter Code:

```
N = 39;
n = 0 : N-1;
[d,fs] = audioread('msmn1.wav');
wc = pi/(11.025);

hd = lowpas(wc,N,n);

x_d=convolve(d,hd);
x_d = x_d(20:end);
x_d = x_d(1:end-19);

specgram(x_d,1024,fs);
fvtool(hd, 1, 'Analysis', 'freq', 'Fs', fs);

function hd = lowpas(wc,N,n)
    hd = sin(wc * (n - (N-1)/2)) ./ (pi * (n - (N-1)/2));
    hd((N-1)/2 + 1) = wc/pi;
    % wh = 0.54 - 0.46 * cos(2*pi*n/(N-1));
    % h = hd .* wh;
end

function y = convolve(x, h)
    M = length(x);
    N = length(h);
    y = zeros(1, M + N - 1);

    for i = 1:M + N - 1
        for j = 1:M
            if i - j + 1 > 0 && i - j + 1 <= N
                y(i) = y(i) + x(j) * h(i - j+1);
            end
        end
    end
end
```

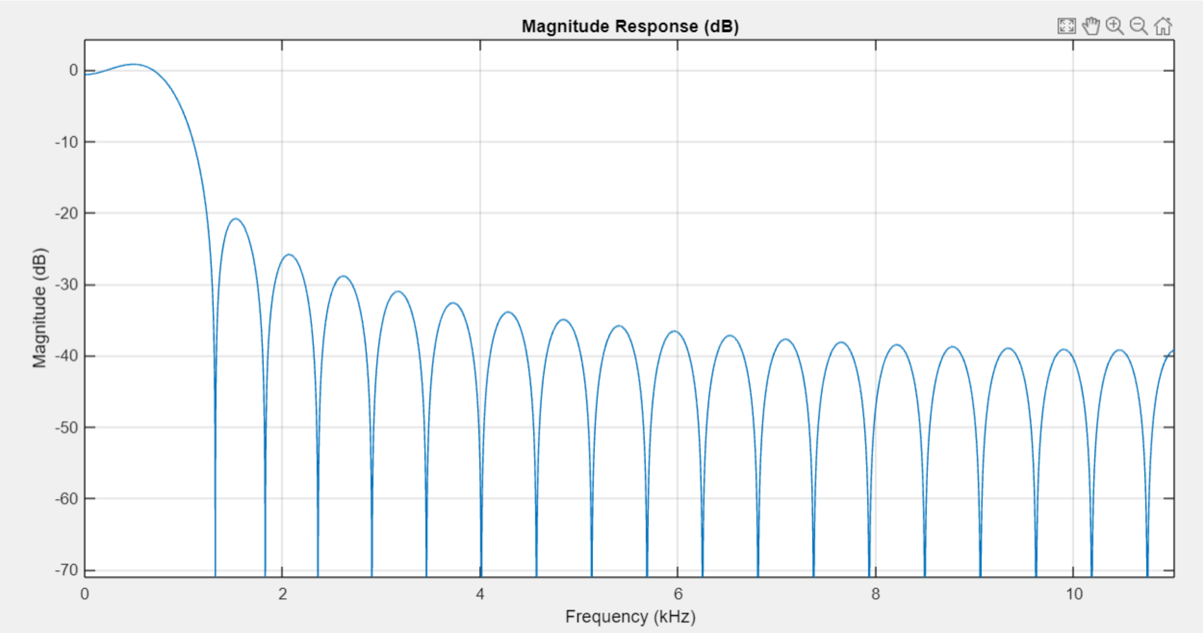
Output spectrogram plot:



Output impulse plot:



Output magnitude plot:

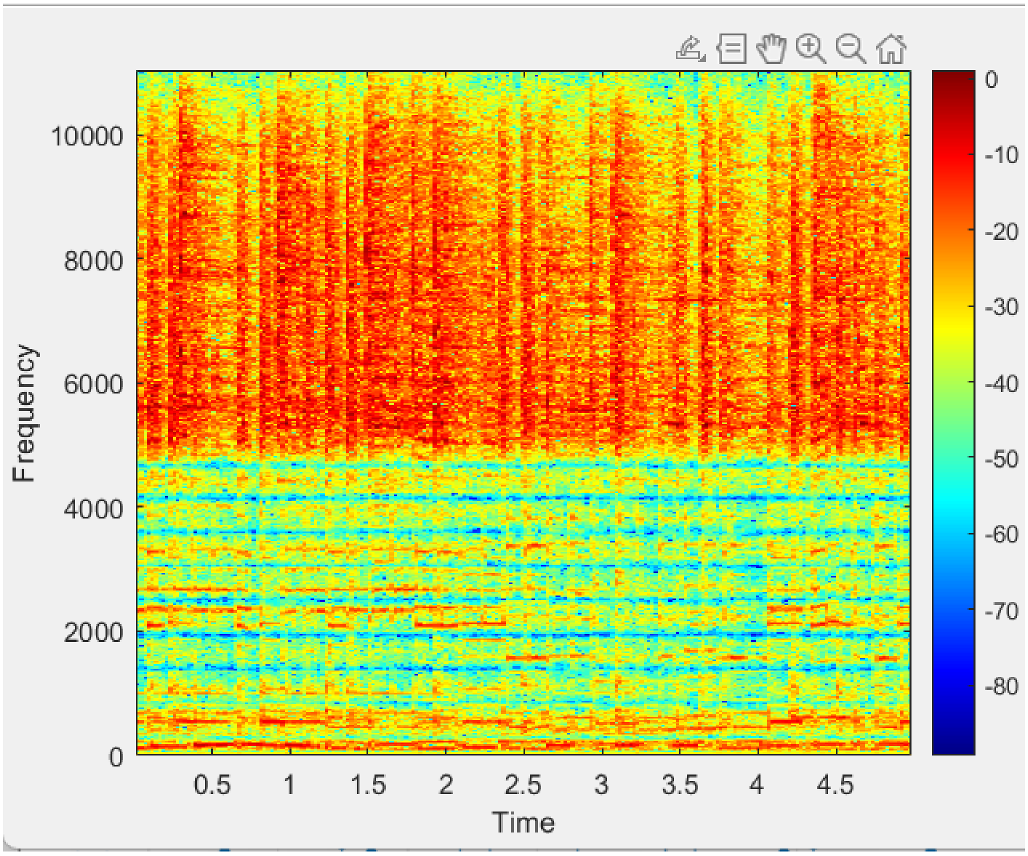


High pass filter code:

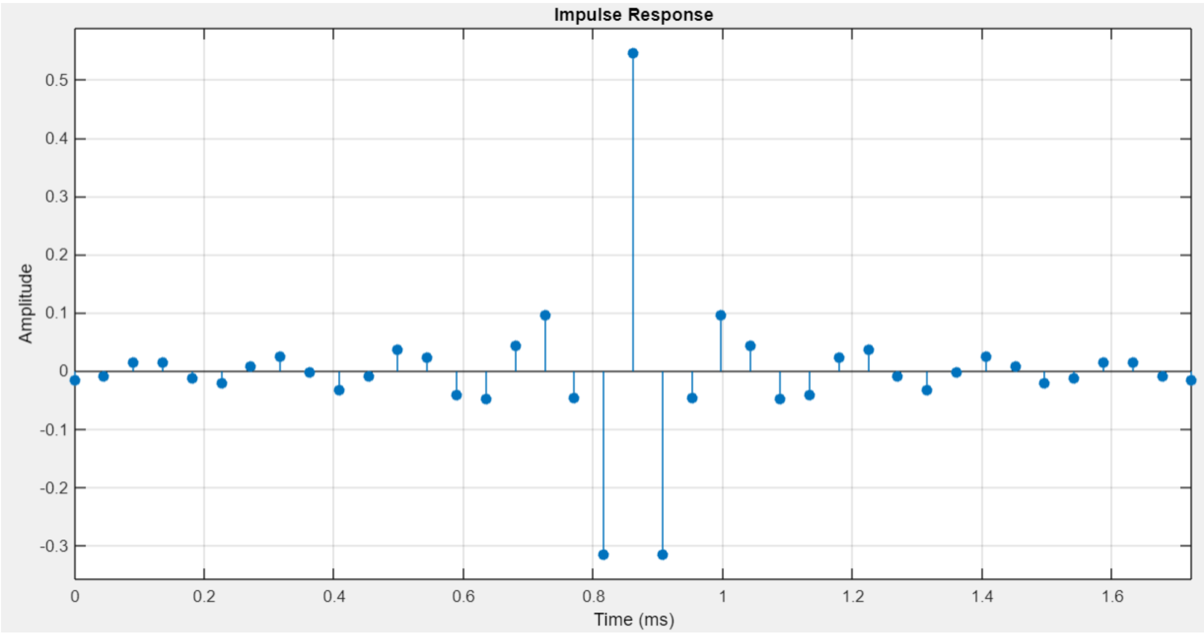
```
N = 39;
n = 0 : N-1;
[d,fs] = audioread('msmn1.wav');
wc = (5*pi)/(11.025);

hd = highpas(wc,N,n);

x_d=convolve(d,hd);
x_d = x_d(20:end);
x_d = x_d(1:end-19);

specgram(x_d,1024,fs);
fvtool(hd, 1, 'Analysis', 'freq', 'Fs', fs);

function hd = highpas(wc,N,n)
    hd = (sin(pi * (n - (N-1)/2)) ./ (pi * (n - (N-1)/2)))-(sin(wc * (n - (N-
1)/2)) ./ (pi * (n - (N-1)/2)));
    hd((N-1)/2 + 1) = 1-(wc/pi);
    % wh = 0.54 - 0.46 * cos(2*pi*n/(N-1));
    % h = hd .* wh;
end

function y = convolve(x, h)
    M = length(x);
    N = length(h);
    y = zeros(1, M + N - 1);

    for i = 1:M + N - 1
        for j = 1:M
            if i - j + 1 > 0 && i - j + 1 <= N
                y(i) = y(i) + x(j) * h(i - j+1);
            end
```

```
        end
    end
end
```

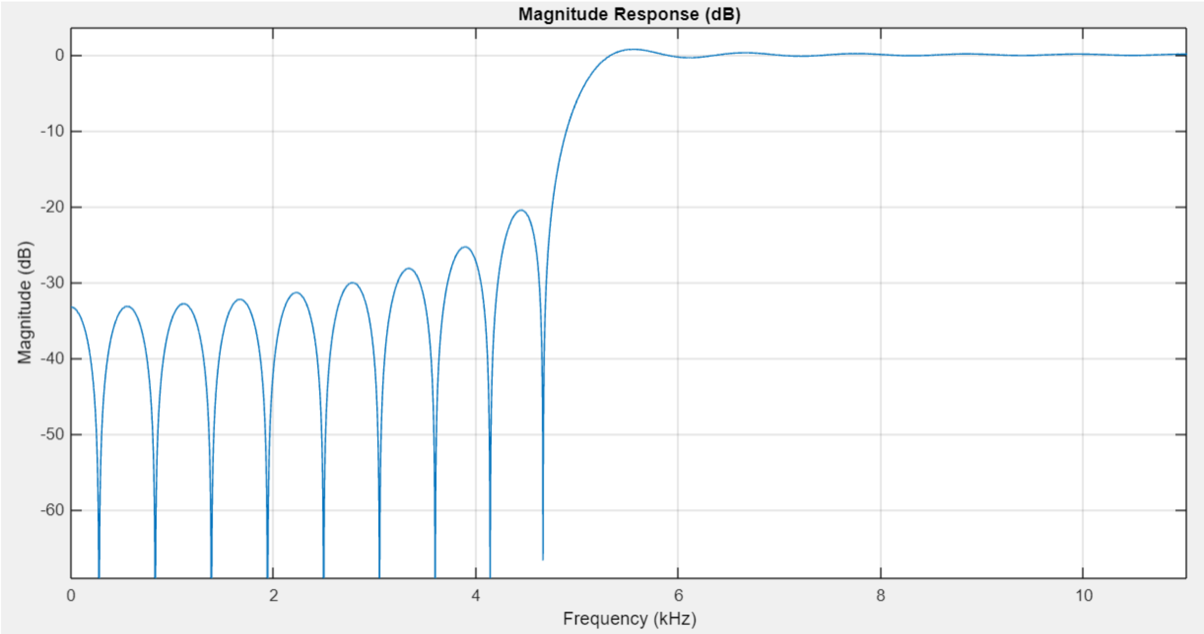
Output spectrogram plot:



Output impulse plot:



Output magnitude plot:



Observations:

1. For required w_c, given that cutoff frequency(in this audio file)

$$w_c = \frac{\pi f_c}{11025}$$

2. Impulse response of filter here is rectangular in frequency domain and it becomes $sinc$ function in time domain.

3. Spectrogram apply short time fourier transform at different time intervals and plot its frequency components wrt time which is nothing but windowing. There is no need of extra windowing.

4.In magnitude plots we can observe the cutoff frequency at '-6dB' bandwidth from max gain bandwidth for both filters .

5. We can observe the change in audio or filtered audio using sound function.

Conclusion:

Audio signal can filtered using rectangular filters(in time domain). The cutoff frequency gives on w_c, which gives required impulse response. Spectrogram helped me to observe the changes before and after filtering.