EXPERIMENT – 7

DSP – LAB

Name – Aman Kumar
Roll no – EE22BTECH11006.

Aim:

The aim of the experiment is to analyse the impulse response and magnitude response of a High Pass Filter(HPF) & Low Pass Filter(LPF).

Theory:

Both HPF and LPF are types of linear filters used to modify the frequency content of a signal. HPF allows frequencies above a certain cutoff to pass through, while attenuating lower frequencies, whereas LPF permits frequencies below a cutoff while attenuating higher frequencies.

MATLAB CODE (HPF):

```
[d,r]= audioread('msmn1.wav');

fc = 1000;
N = 125;
[y,h_n] = highpassfilter(d, fc, r, N);
specgram(y,1024,r);
fvtool(h_n);
function [y,h_n] = highpassfilter(x, fc, r, N)
    wc = 2*pi*fc/r;
    h_d = zeros(1,N);
    for n = -(N - 1) / 2 : (N - 1) / 2
        if n == 0
            h_d(n + (N - 1) / 2 + 1) = 1 - wc/pi;
        else
            h_d(n + (N - 1) / 2 + 1) = sin(pi*n)/(pi*n) - sin(wc*n)/(pi*n);
        end
    end

    w_h = zeros(1,N);
    for k= 0:N-1
        w_h(k+1) = 0.54-0.46*cos(2*pi*k/(N-1));
    end
    h_n = h_d .* w_h;
    y = conv_calculator(x, h_n);
end

function ans_conv = conv_calculator(x, h)
    n = length(x);
    m = length(h);
    N = [x, zeros(1, m - n - 1)];
    M = [h, zeros(1, n - 1)];
```

```
    Y = zeros(1, m + n - 1);

    for i = 1:m + n - 1
        for j = 1:n
            if (i - j + 1 > 0 && i - j + 1 <= m)
                Y(i) = Y(i) + N(j) * M(i - j + 1);
            end
        end
    end
    ans_conv = Y;
end
```

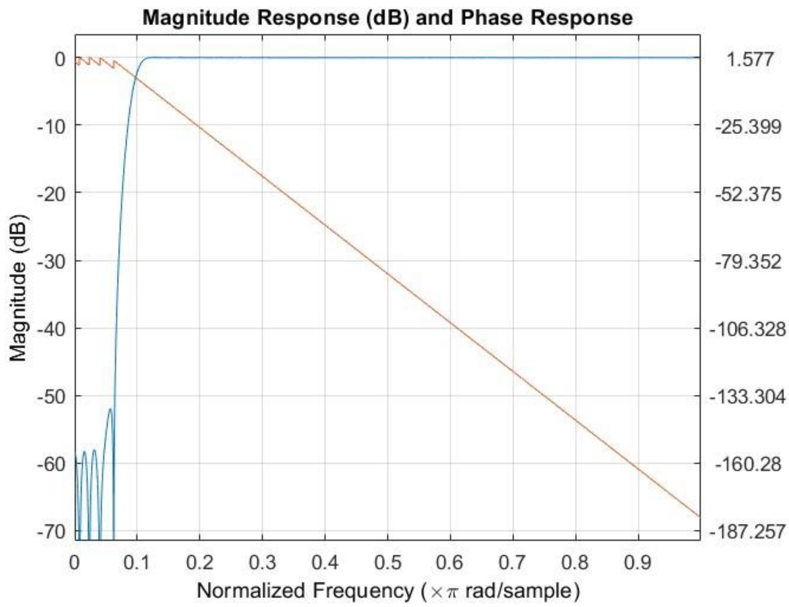
MATLAB CODE(LPF):

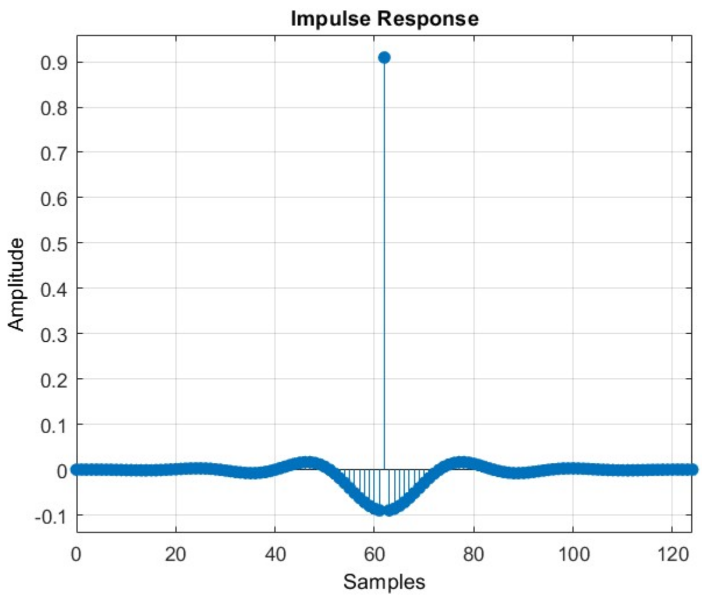
```
[d,r]= audioread('msmn1.wav');

fc = 1000;
N = 125;
[y,h_n] = lowpassfilter(d, fc, r, N);
specgram(y,1024,r);
fvtool(h_n);
function [y,h_n] = lowpassfilter(x, fc, r, N)
    wc = 2*pi*fc/r;
    h_d = zeros(1,N);
    for n = -(N - 1) / 2 : (N - 1) / 2
        if n == 0
            h_d(n + (N - 1) / 2 + 1) = wc/pi;
        else
            h_d(n + (N - 1) / 2 + 1) = sin(wc*n)/(pi*n);
        end
    end

    w_h = zeros(1,N);
    for k= 0:N-1
        w_h(k+1) = 0.54-0.46*cos(2*pi*k/(N-1));
    end
    h_n = h_d .* w_h;
    y = conv_calculator(x, h_n);
end

function ans_conv = conv_calculator(x, h)
    n = length(x);
    m = length(h);
    N = [x, zeros(1, m - n - 1)];
    M = [h, zeros(1, n - 1)];
    Y = zeros(1, m + n - 1);

    for i = 1:m + n - 1
        for j = 1:n
            if (i - j + 1 > 0 && i - j + 1 <= m)
                Y(i) = Y(i) + N(j) * M(i - j + 1);
            end
        end
    end
    ans_conv = Y;
end
```

In the above codes we are reading the audio signal then keeping cutoff frequency as 1000Hz and passing it through the high pass filter & low pass filter subsequently, also we are using hamming window function for windowing.

HPF plots:

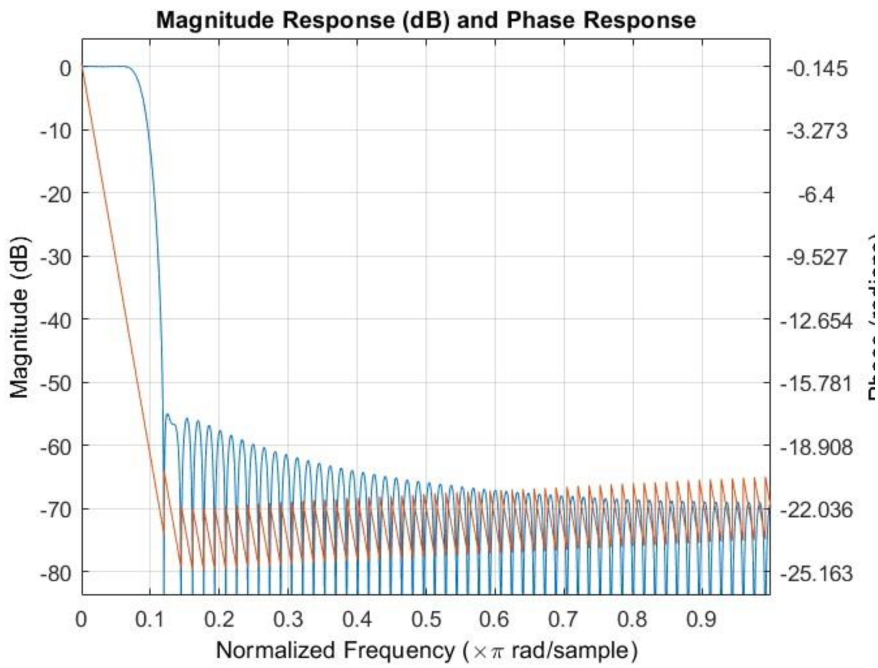
Magnitude Response, Phase response & Impulse response:

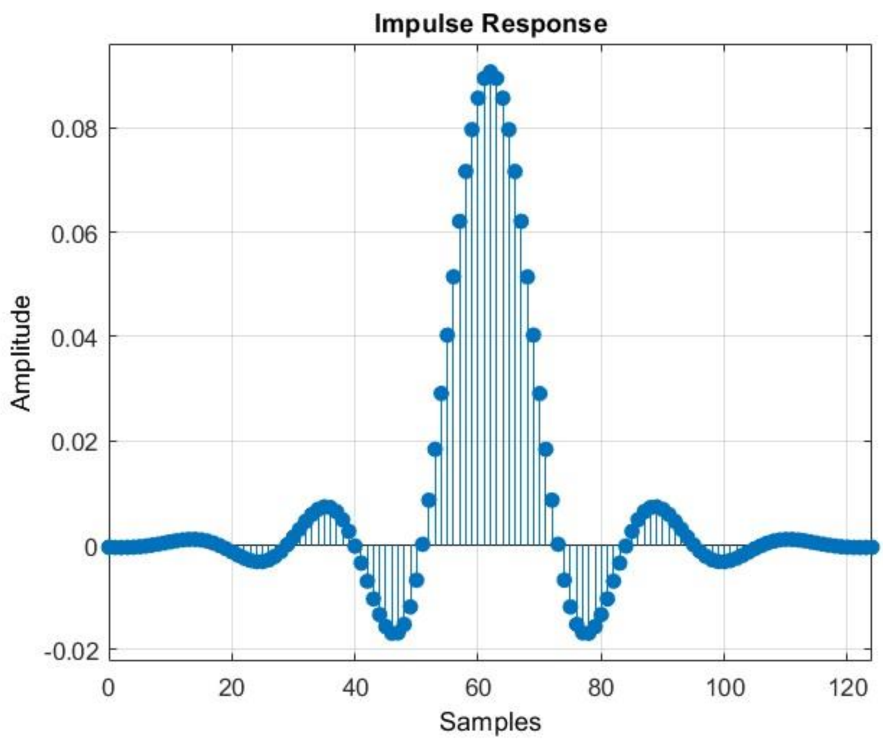




LPF Plots:

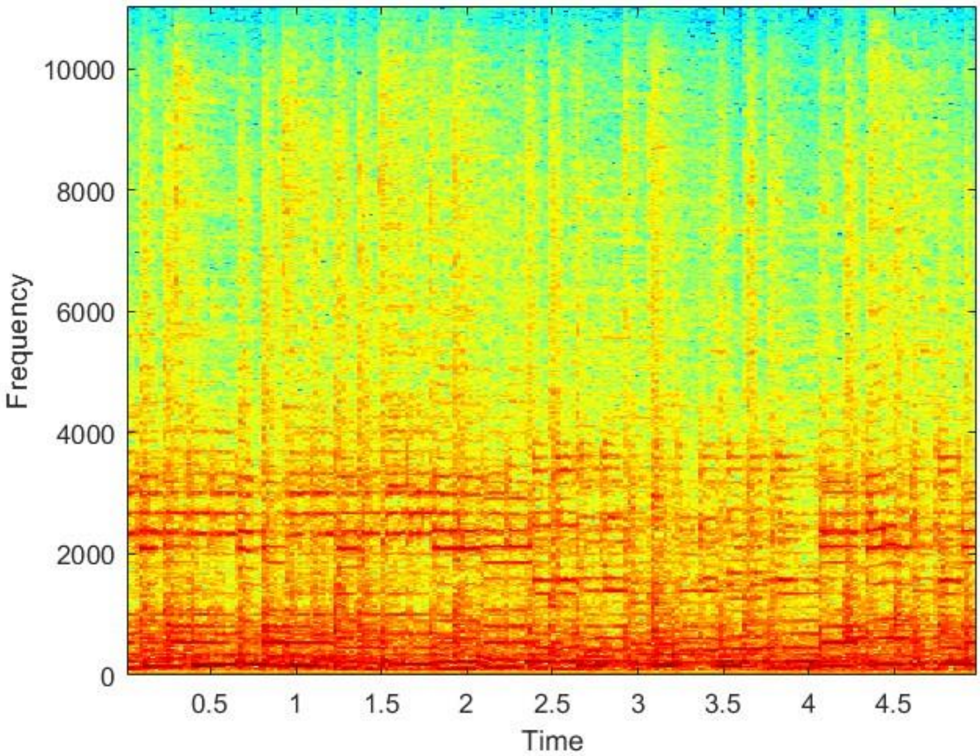
Magnitude, Phase Response & Impulse response:



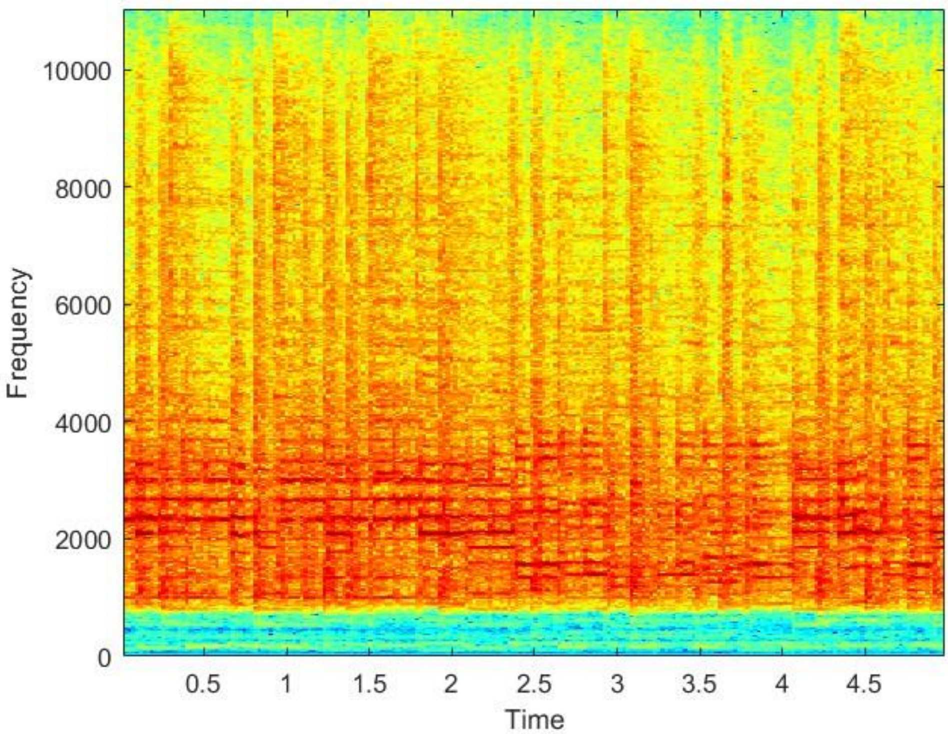


Specgram Plots:

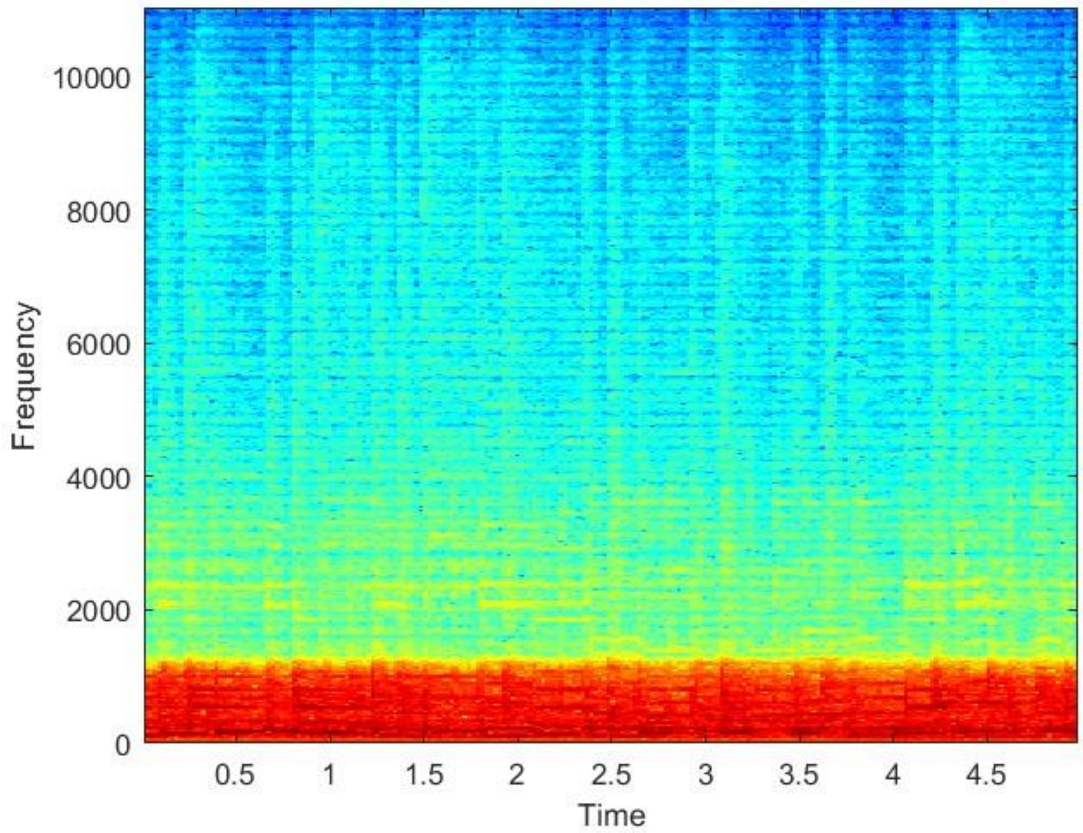
1. Original plot:



2. HPF:



3. LPF:



Observations:

1. Both the High Pass Filter (HPF) and Low Pass Filter (LPF) exhibit distinct characteristics in their impulse responses and frequency responses.

2. When applied to the input audio signal, the HPF emphasizes high-frequency components, while the LPF enhances low-frequency components. This is evident in the spectrograms of the filtered signals, where the HPF attenuates lower frequencies and the LPF attenuates higher frequencies.

3. Visualizing the frequency responses of both filters using the `fvtool` function provides insights into their magnitude and phase responses. The HPF exhibits significant attenuation in the lower frequencies, while the LPF shows attenuation in the higher frequencies, as expected based on their filter designs.

4. Comparing the original spectrogram with those of the filtered signals allows for a clear assessment of the filtering effects.