EE2801 - DSP LAB

Experiment – 7: Audio signal processing using HPF

Name: Anupama Kulshreshtha

Roll no.: EE22BTECH11009

Aim of the Experiment: To read the given audio file and filter it using HPF and LPF filters and then produce the specgram plots of input and output audio files using matlab.

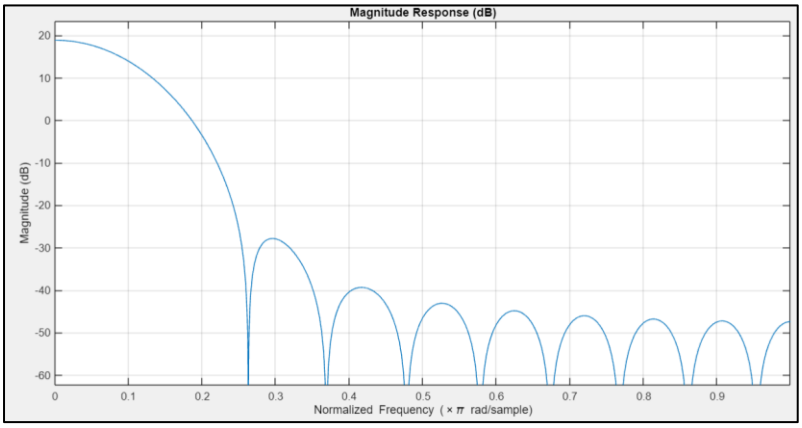
Matlab Code:

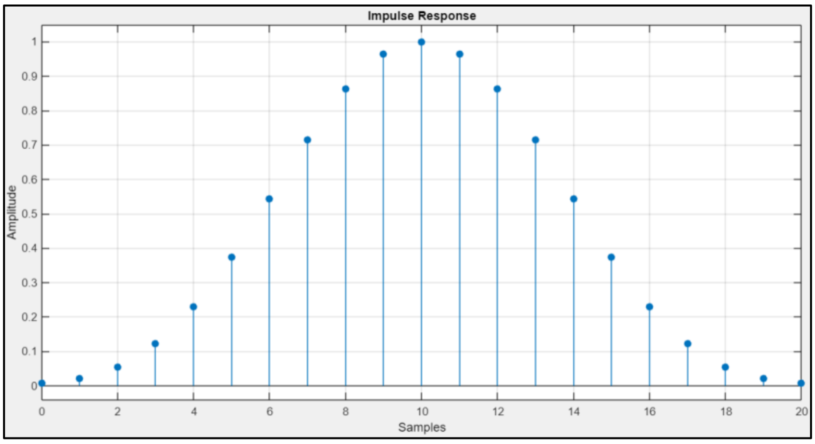
```
[d,r] = audioread('msmn1.wav');
wc = pi/11;
fs = 22000;
N = 21;
fc = (wc * fs) / (2 * pi);

h = LPF(fc,fs,N);
h1 = HPF(fc,fs,N);
y = myConvolution(d,h);
y1 = myConvolution(d,h1);
figure;
subplot(3,1,1);
specgram(d,1024,r);
subplot(3,1,2);
specgram(y,1024,r);
subplot(3,1,3);
specgram(y1,1024,r);
%soundsc(y,r);
fvtool(h)
fvtool(h1)

function h = LPF(fc, fs, N)
    wc = 2 * pi * fc / fs;
    hd = zeros(1,N);
    % Calculating impulse response
    for k = 1:N
        n = k - (N+1)/2;
        if n == 0
            hd(k) = 1;
        else
            hd(k) = sin(wc*n)/(wc*n);
        end
    end

    % Define Hamming window
    n1 = 0:N-1;
    WH = zeros(1, N);
    l = (n1 >= 0) & (n1 <= N-1);
    WH(l) = 0.54 - 0.46 * cos(2 * pi * n1(l) / (N-1));

    % Apply window to filter coefficients to get practical impulse response
    h = hd .* WH;
```

```
end

function hd = HPF(fc, fs, N)
    % Calculate the cutoff frequencies in radians
    wc = 2*pi*fc/fs;
    hd = zeros(1,N);
    % Calculating impulse response
    for k = 1:N
        n = k - (N+1)/2;
        if n == 0
            hd(k) = 1 - (wc/pi);
        else
            hd(k) = (sin(pi*n) - sin(wc*n))./(pi*n);
        end
    end
end


function result = myConvolution(x,h)
    % Lengths of the signals
    M = length(x);
    N = length(h);

    % Length of the result signal
    L = M + N - 1;

    % Initialize the result signal
    result = zeros(1, L);
    % Perform convolution
    for n = 1:L
        for k = max(1, n-N+1):min(n, M)
            result(n) = result(n) + x(k) * h(n-k+1);
        end
    end
end
```

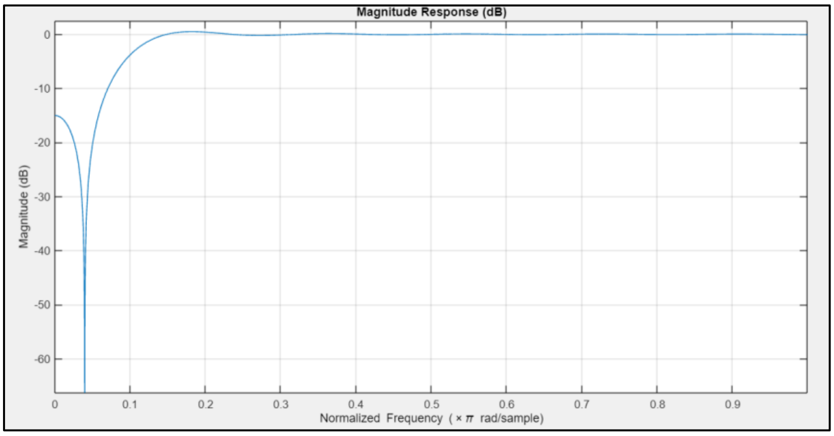
Impulse and Magnitude Plots:

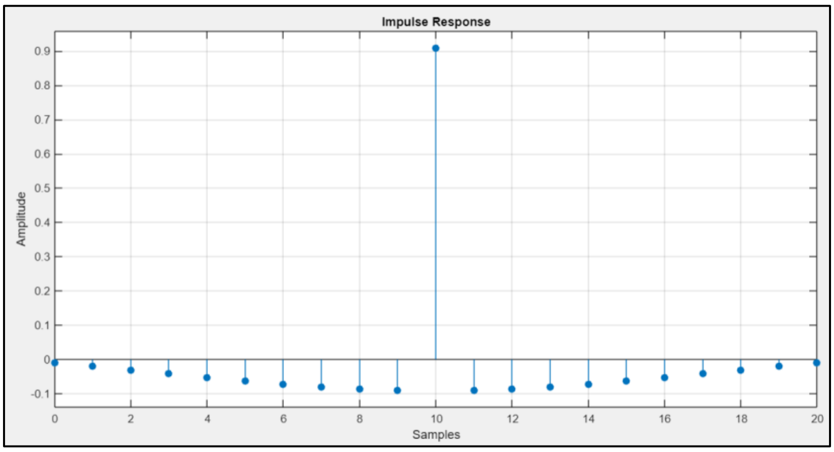
(a) LPF:





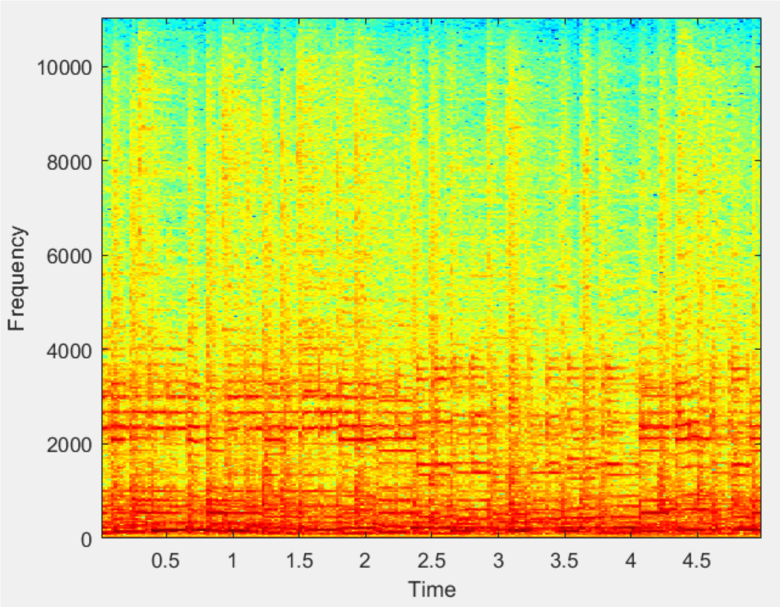
(b) HPF:



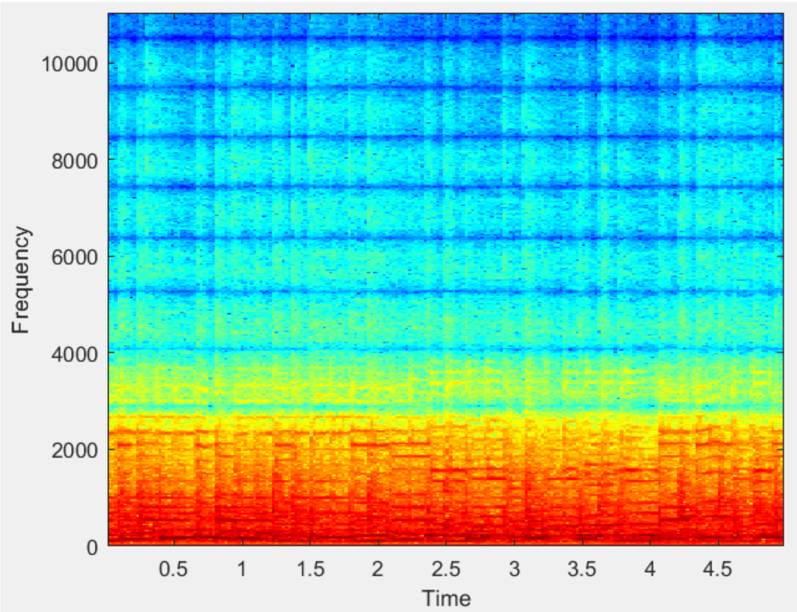


Specgram Plots:

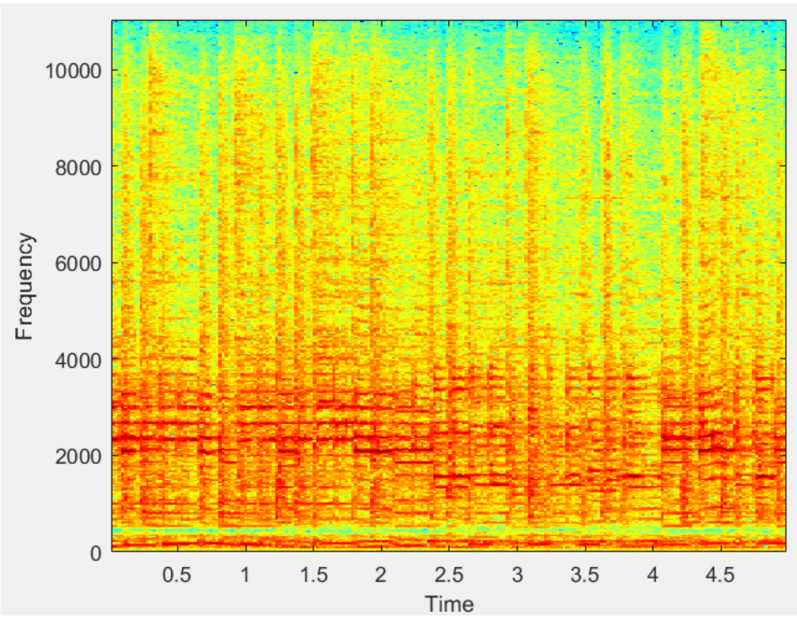
Input audio:



Output audio with LPF:



Output audio with BPF:



Observations:

- We use filtering in audio signal processing to remove unwanted frequency components from a signal.
- In the specgram plot obtained using HPF, there is a shift in the frequency content towards higher frequencies while compared to the input audio specgram plot.
- This indicates that the low-frequency components are attenuated or removed from the signal.
- Similar case is observed for LPF, but in this case high-frequency components are attenuated from the signal.
- Hence, the code and the specgram plots help us to visualize the effect of the HPF and LPF functions on the given audio signal.