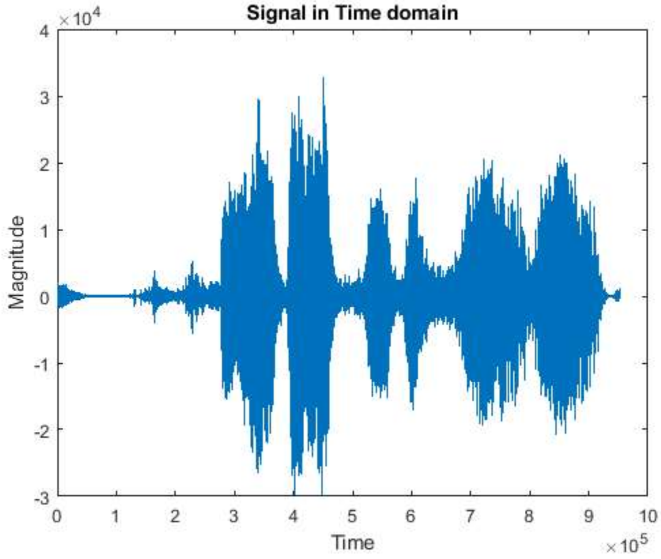
Project: Early detection and warning system for possible elephant movements in localities

Group ID: 2021.3.17

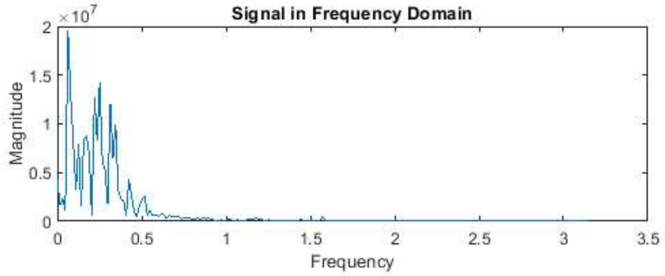
```
[audio_in,audio_freq_samp1] = audioread('Elephant_and_Bird_Sound.mpeg');
length_audio = length(audio_in);
df = audio_freq_samp1/length_audio;
```

```
clc
filename = 'Elephant_and_Bird_Sound.mpeg';
[y,Fs] = audioread(filename);
normal=y/max(y);
sound(y,Fs);
data = round(32767*normal);
figure(1)
plot(data),title('Signal in Time domain'),xlabel('Time'),ylabel('Magnitude');
```



Converting time domain signal to frequency domain

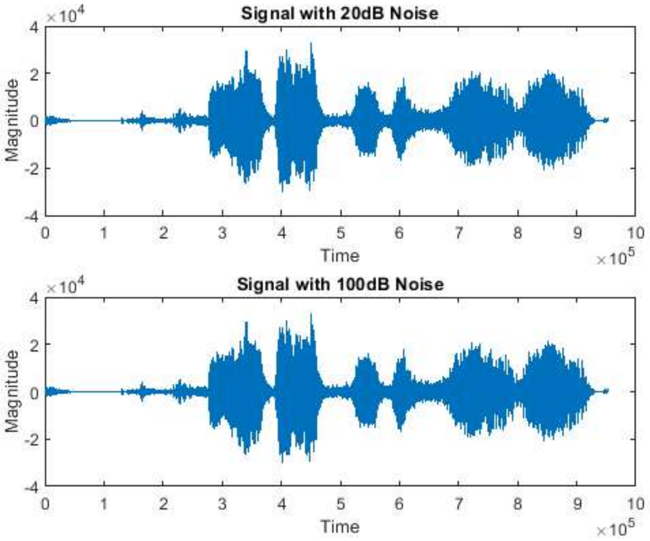
```
w=0:pi/200:pi;
[h, w]=freqz(data,1,w);
subplot(212),plot(w,abs(h)),title('Signal in Frequency Domain'),xlabel('Frequency'),ylabel('Magnitude');
```



Adding noise to the signal

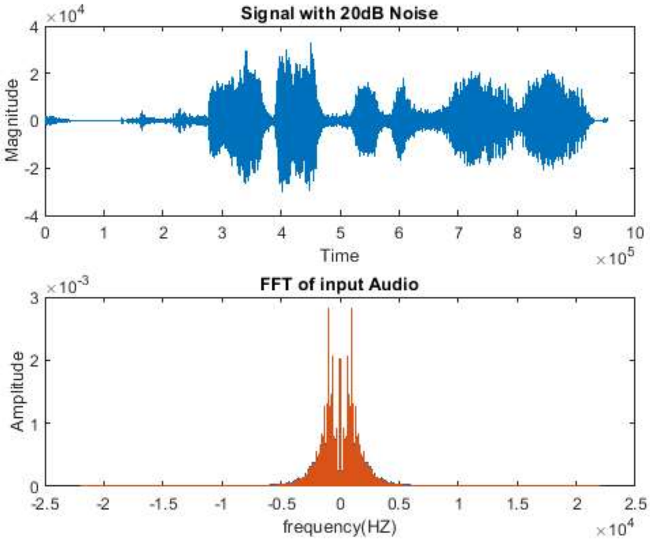
```
noise_SNR_1=awgn(data,20);
noise_SNR_2=awgn(data,100);
figure(2)
```

```
subplot(211),plot(noise_SNR_1),title('Signal with 20dB Noise'),xlabel('Time'),ylabel('Magnitude');
subplot(212),plot(noise_SNR_2),title('Signal with 100dB Noise'),xlabel('Time'),ylabel('Magnitude');
```



```
frequency_audio = -audio_freq_samp1/2:df:audio_freq_samp1/2-df;
```

```
FFT_audio_in = fftshift(fft(audio_in)/length(fft(audio_in)));
plot(frequency_audio,abs(FFT_audio_in));
title('FFT of input Audio');
xlabel('frequency(HZ)');
ylabel('Amplitude');
```

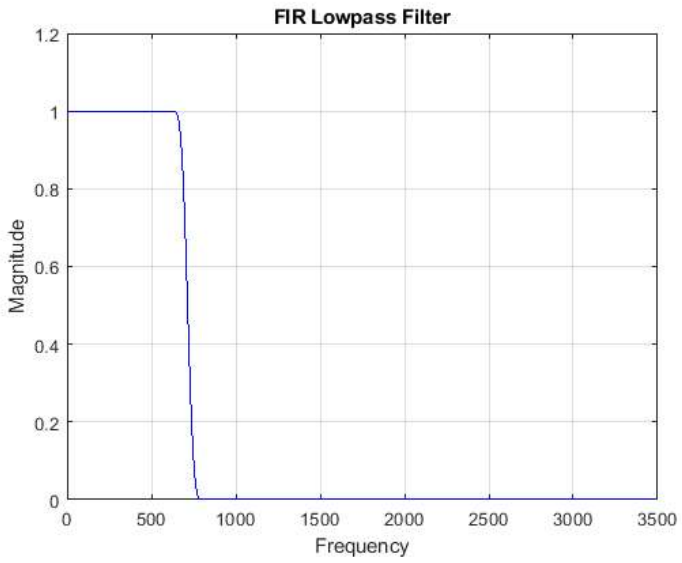


```
lower_threshold = 150;
upper_threshold = 2500;

val = abs(frequency_audio)<upper_threshold & abs(frequency_audio)>lower_threshold;
FFT_ins = FFT_audio_in(:,1);
FFT_voc = FFT_audio_in(:,1);


FFT_ins(val) = 0;
FFT_voc(~val) = 0;


FFT_a = ifftshift(FFT_audio_in);
FFT_a11 = ifftshift(FFT_ins);
```

```
FFT_a31 = ifftshift(FFT_voc);
s1 = ifft(FFT_a11*length(fft(audio_in)));
s3 = ifft(FFT_a31*length(fft(audio_in)));
```

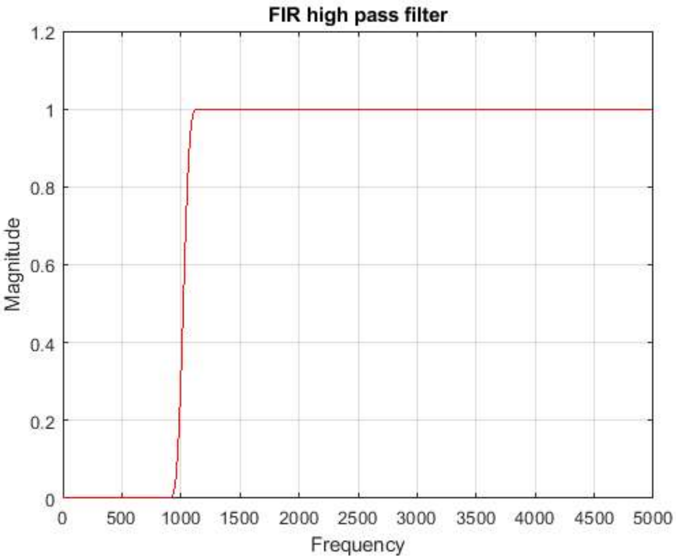
Lowpass Filter

```
F1=Fs/5;
F2=800;
F3=1000;
r1=10;
r2=80;
p1=1-10.^(-r1/20);
s1_=10.^(-r2/20);
FF=[F2 F3];
ma=[1 0];
v=[p1 s1_];
[A21,wA21,bt,~]=kaiserord(FF,ma,v,F1);
lowpassfilter=fir1(A21,wA21,kaiser(A21+1,bt));
[h,w]=freqz(lowpassfilter,1);
figure(3)
plot(w*Fs*(0.5/pi),abs(h),'b');
title('FIR Lowpass Filter'),xlabel('Frequency'),ylabel('Magnitude'), grid on;
```



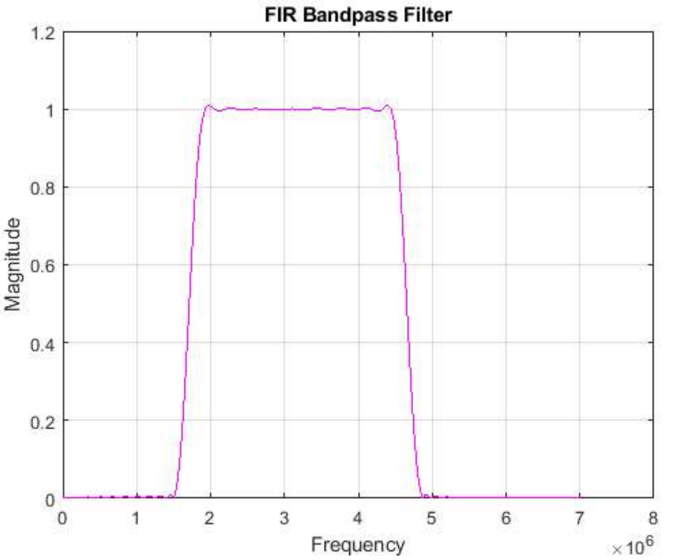
Highpass Filter

```
F2=3000;
F3=2000;
FF1 = [F2,F3];
[A23,wA23,bt,~]=kaiserord(FF1,ma,v,F1);
higpassfilter=fir1(A23,wA23,'high',kaiser(A23+1,bt));
[h,w]=freqz(higpassfilter,1);
figure(4)
plot(w*Fs*0.5/pi,abs(h),'r');
title('FIR high pass filter');
xlabel('Frequency'),ylabel('Magnitude'), grid on;
```



Bandpass Filter

```
F = [900 1250 2800 3050 ];
ma=[0 1 0];
devs=[0.01 0.05 0.01];
[n,wA23,bt,Yp]=kaiserord(F,ma,devs,F1);
n = n+rem(n,2);
bandpassfilter=fir1(n,wA23,'bandpass',kaiser(n+1,bt),'noscale');
[h,w]=freqz(bandpassfilter,1,1024,F1);
figure(5)
plot(w*Fs*0.5/pi,abs(h),'m');
title('FIR Bandpass Filter');
xlabel('Frequency'),ylabel('Magnitude'), grid on;
```



Background Sound

Signal Filtering Using Lowpass Filter

```
SFULF=fftfilt(lowpassfilter,s1);
```

Signal Filtering Using Highpass Filter

```
SFUHF=fftfilt(higpassfilter,s1);
```

Signal Filtering Using Bandpass Filter

```
SFUBF=fftfilt(bandpassfilter,s1);
```

Elephant Sound

Signal Filtering Using Lowpass Filter

```
SFULF_E=fftfilt(lowpassfilter,s3);
```

Signal Filtering Using Highpass Filter

```
SFUHF_E=fftfilt(higpassfilter,s3);
```

Signal Filtering Using Bandpass Filter

```
SFUBF_E=fftfilt(bandpassfilter,s3);
```

Audio File Write

```
sound(s1,audio_freq_samp1)
audiowrite('sound_background.wav',s1,audio_freq_samp1);
audiowrite('sound_Elephant.wav',s3,audio_freq_samp1);
```

Background Sound Audio File After Passing Low Pass Filter

```
audiowrite('sound_background_Passing_Low.wav',SFULF,audio_freq_samp1);
```

Background Sound Audio File After Passing High Pass Filter

```
audiowrite('sound_background_Passing_High.wav',SFUHF,audio_freq_samp1)
```

Background Sound Audio File After Passing Band Pass Filter

```
audiowrite('sound_background_Passing_Band.wav',SFUBF,audio_freq_samp1)
```

Elephant Sound Audio File After Passing Low Pass Filter

```
audiowrite('sound_Elephant_Passing_Low.wav',SFULF_E,audio_freq_samp1);
```

Elephant Sound Audio File After Passing High Pass Filter

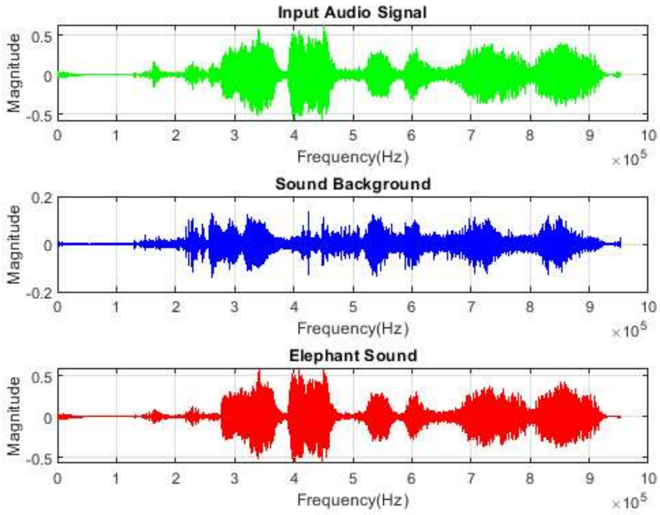
```
audiowrite('sound_Elephant_Passing_High.wav',SFUHF_E,audio_freq_samp1)
```

Elephant Sound Audio File After Passing Band Pass Filter

```
audiowrite('sound_Elephant_Passing_Band.wav',SFUBF_E,audio_freq_samp1)
```

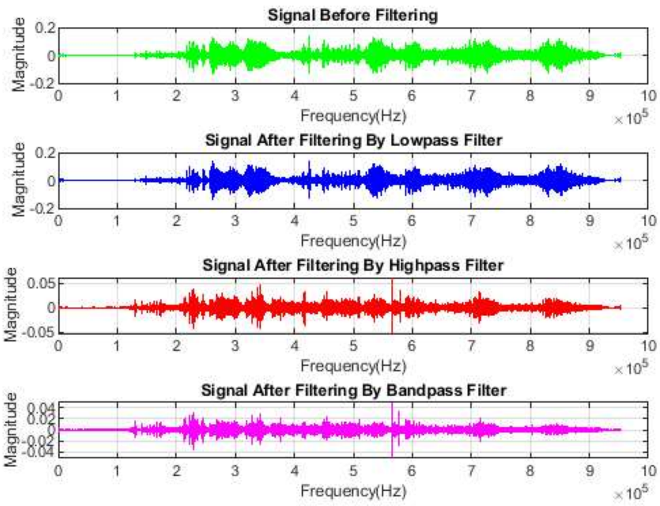
Graphical Comparison

```
figure(6)
subplot(311),plot(y,'g'),title('Input Audio Signal'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(312),plot(s1,'b'),title('Sound Background'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(313),plot(s3,'r'),title('Elephant Sound'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
```



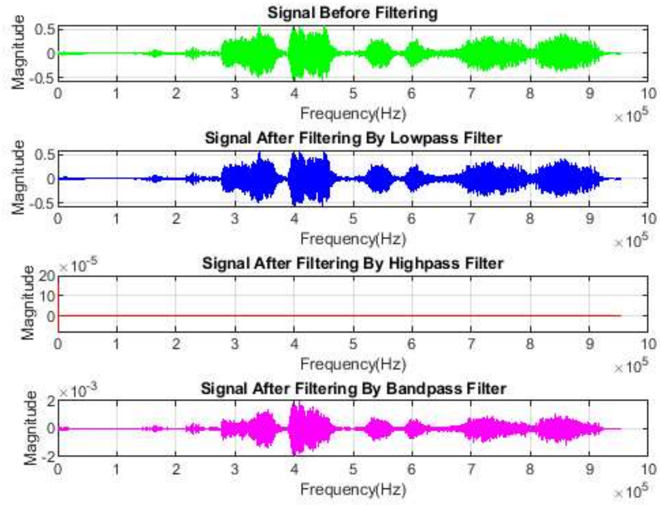
Background Sound

```
figure(7)
subplot(411),plot(s1,'g'),title('Signal Before Filtering'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(412),plot(SFULF,'b'),title('Signal After Filtering By Lowpass Filter'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(413),plot(SFUHF,'r'),title('Signal After Filtering By Highpass Filter'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(414),plot(SFUBF,'m'),title('Signal After Filtering By Bandpass Filter'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
```



Elephant Sound

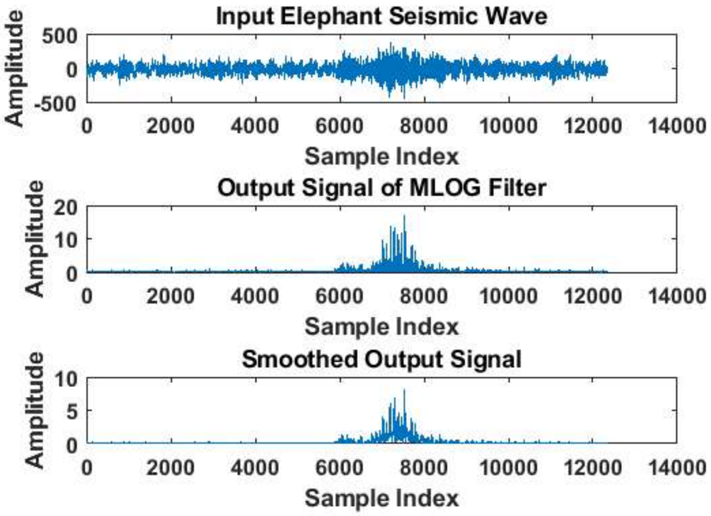
```
figure(8)
subplot(411),plot(s3,'g'),title('Signal Before Filtering'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(412),plot(SFULF_E,'b'),title('Signal After Filtering By Lowpass Filter'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(413),plot(SFUHF_E,'r'),title('Signal After Filtering By Highpass Filter'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
subplot(414),plot(SFUBF_E,'m'),title('Signal After Filtering By Bandpass Filter'),xlabel('Frequency(Hz)'),ylabel('Magnitude'), grid on;
```



Elephant step vibration detection in using MLOG

```
load x_Example
sigma = 2.5;
N = 10;

  [Gaussian_1D_2_Diff_Modified]=MLOG(sigma,N) /sigma;

Output = filter (Gaussian_1D_2_Diff_Modified,1,x);

Output = Output.^2;

Output_More_Smoothing = zeros(1,size(x,2));
m = 8;
k = 1;
count = 1;
 for j=1:k:(length(Output))-max([m k])
 Output_More_Smoothing(count)=(mean(Output(j:j+m-1)));
 count = count +1;
 end
```

```
figure(1)
subplot(3,1,1),plot(1:length(x),x)
xlabel('Sample Index','FontSize',14,'FontWeight','bold')
ylabel('Amplitude','FontSize',14,'FontWeight','bold')
title('Input Elephant Seismic Wave','FontSize',14,'FontWeight','bold')
set(gca,'fontsize',12,'FontWeight','bold')
subplot(3,1,2),plot(1:length(Output), Output)
xlabel('Sample Index','FontSize',14,'FontWeight','bold')
ylabel('Amplitude','FontSize',14,'FontWeight','bold')
title('Output Signal of MLOG Filter','FontSize',14,'FontWeight','bold')
set(gca,'fontsize',12,'FontWeight','bold')
subplot(3,1,3),plot(1:length(Output_More_Smoothing), Output_More_Smoothing)
xlabel('Sample Index','FontSize',14,'FontWeight','bold')
ylabel('Amplitude','FontSize',14,'FontWeight','bold')
title('Smoothed Output Signal','FontSize',14,'FontWeight','bold')
set(gca,'fontsize',12,'FontWeight','bold')
```



IoT System

```
channelInfo = 

  struct with fields:

            ChannelID:  1684946

            Author : mwa0000026055608

                 Name: 'Elephant Detection'

          Description: ''

             Latitude: 23.777176

            Longitude: 90.399452

              Created: 25-March-2022 17:50:32

              Updated: 27-March-2022 13:40:31

          LastEntryID: 35431

    FieldDescriptions: {1×6 cell}

             FieldIDs: [1 2 3 4 5 6]

                  URL: 'https://thingspeak.com/channels/1684946/api_keys?'
```

```
thingSpeakWrite(1684946,getdatasamples(s1),'Writekey','88QQLDEOD4I7ELH8',"Fields",1)
thingSpeakWrite(1684946,getdatasamples(s3),'Writekey','88QQLDEOD4I7ELH8',"Fields",2)
thingSpeakWrite(1684946,getdatasamples(SFULF),'Writekey','88QQLDEOD4I7ELH8',"Fields",3)
thingSpeakWrite(1684946,getdatasamples(SFUhF),'Writekey','88QQLDEOD4I7ELH8',"Fields",4)
thingSpeakWrite(1684946,getdatasamples(SFUbF),'Writekey','88QQLDEOD4I7ELH8',"Fields",5)
thingSpeakWrite(1684946,getdatasamples(staus),'Writekey','88QQLDEOD4I7ELH8',"Fields",6)
```